

Воронежский государственный университет
Факультет прикладной математики, информатики и механики

**Математика,
информационные технологии,
приложения**

*Межвузовская научная конференция
молодых ученых и студентов*

Воронеж,
27 апреля 2022 г.

Воронеж
Издательско-полиграфический центр
«Научная книга»
2022

УДК 531(063)+51-7(063)
ББК 22.2я5+22.1я5
М34

Председатель программного и организационного комитета

А. И. Шашкин, д-р физ.-мат. наук, профессор, декан факультета прикладной математики, информатики и механики Воронежского государственного университета

Заместители председателя программного и организационного комитета:

С. Н. Медведев, канд. физ.-мат. наук, доцент
С. Ю. Болотова, канд. физ.-мат. наук, доцент

Члены программного и организационного комитета:

Г. В. Абрамов, д-р техн. наук, проф.; Т. В. Азарнова, д-р техн. наук, доц.;
Е. М. Аристова, канд. физ.-мат. наук, доц.; М. А. Артемов, д-р физ.-мат. наук, проф.;
И. Ф. Астахова, д-р техн. наук, проф.; Н. Б. Баева, канд. экон. наук, доц.;
Е. С. Барановский, канд. физ.-мат. наук, доц.; А. Г. Баскаков, д-р физ.-мат. наук, проф.;
Ю. В. Бондаренко, д-р техн. наук, доц.; И. Е. Воронина, д-р техн. наук, доц.;
О. Д. Горбенко, канд. физ.-мат. наук, доц.; В. Г. Задорожний, д-р физ.-мат. наук, проф.;
Н. А. Каплиева, канд. физ.-мат. наук, доц.; И. Л. Каширина, д-р техн. наук, доц.
С. Л. Кенин, канд. физ.-мат. наук; А. В. Ковалев, д-р физ.-мат. наук, проф.;
Н. В. Козлова, канд. техн. наук; В. Г. Курбатов, д-р физ.-мат. наук, проф.;
Т. М. Леденёва, д-р техн. наук, проф.; Л. Н. Ляхов, д-р физ.-мат. наук, проф.;
О. А. Медведева, канд. физ.-мат. наук, доц.; Н. В. Минаева, д-р физ.-мат. наук, доц.;
И. П. Половинкин, д-р физ.-мат. наук, доц.; Ю. К. Тимошенко, д-р физ.-мат. наук, доц.;
О. Ф. Ускова, канд. техн. наук, доц.

Математика, информационные технологии, приложения : сборник трудов Межву-
М34 зовской научной конференции молодых ученых и студентов, Воронеж, 27 апреля 2022 г. –
Воронеж : Научная книга, 2022. – 292 с.

ISBN 978-5-4446-****-*

Традиционно конференция является междисциплинарной. Важнейшая ее цель – ознакомление молодых специалистов с новыми веяниями в современной науке, а также обмен знаниями и достижениями в предложенных научных направлениях.

УДК 531(063)+51-7(063)
ББК 22.2я5+22.1я5

ISBN 978-5-4446-****-*

© ФГБОУ ВО ВГУ, 2022
© Издательско-полиграфический центр
«Научная книга», 2022

ОПРЕДЕЛЕНИЕ ДИАПАЗОНА УГЛОВ ДЛЯ ЧЕТЫРЕХЗВЕННОГО МАНИПУЛЯТОРА

А. Ю. Блиняева

Воронежский государственный университет

Введение

В наши дни в промышленности для автоматизации производства все чаще вводятся в эксплуатацию роботы-манипуляторы. Для того, чтобы их работа была более эффективной, для них необходимо решить прямую и обратную задачи кинематики. Прямая задача кинематики заключается в определении пространственного положения рабочего механизма по его заданной ориентации подвижных звеньев, а обратная задача, наоборот, заключается в нахождении параметров, которые необходимо задать звеньям манипулятора, для того, чтобы его рабочий механизм оказался в заданной точке.

1. Постановка задачи

Дан четырехзвенный манипулятор на плоскости. Для него задана начальная точка фиксации с координатами (x_0, y_0) , конечная точка схвата с координатами (x_4, y_4) , а также известны длины всех его звеньев l_i , $i = 1, 4$. Для данного манипулятора необходимо решить обратную задачу кинематики (ОЗК), т. е. найти расположение звеньев на плоскости, чтобы манипулятор был связным, крепился в начальной точке и заканчивался в конечной. Для того чтобы найти расположение звеньев на плоскости, необходимо для всех его звеньев найти конечные координаты (x_1, y_1) , (x_2, y_2) , (x_3, y_3) или же углы их поворота в сочленениях α_i , где $i = 1, 4$ (рис. 1).

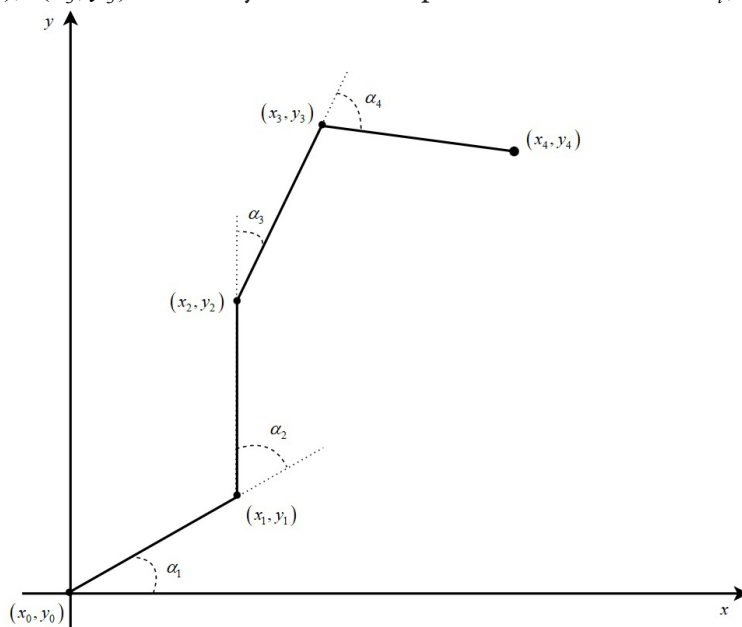


Рис. 1. Четырехзвенный манипулятор

В связи с тем, что одной из главных проблем при решении ОЗК является неоднозначность ее решения, для задачи также ставится и задача выбора оптимального варианта из множества, который будет удовлетворять определенному критерию.

2. Алгоритм построения звеньев манипулятора

Первым делом обозначим сумму длин всех звеньев манипулятора как d , а отрезок, соединяющий начальную точку фиксации и конечную точку схвата как OM . Далее вычислим длину этого отрезка и рассмотрим три случая, где:

- 1) $d < OM$;
- 2) $d = OM$;
- 3) $d > OM$.

Рассмотрение каждого из случаев необходимо для того, чтобы при построении манипулятора на плоскости выбрать наиболее подходящий диапазон углов отклонения его первых звеньев.

В первом случае, когда $d < OM$, решением ОЗК будет пустое множество, т. к. длина всех звеньев заведомо меньше расстояния до конечной точки схвата.

Во втором случае, когда $d = OM$, звенья манипулятора выстроятся в единую линию и это будет единственным возможным решением.

Третий случай, при котором $d > OM$, рассмотрим более подробно.

В точке O с координатами (x_0, y_0) построим окружность радиуса l_1 и через эту же точку проведем перпендикуляр к отрезку OM . После чего обозначим угол между осью Ox и проведенным перпендикуляром как φ (рис. 2). Данный угол будет необходим для вычисления диапазона угла отклонения первого звена манипулятора и его дальнейшего построения.

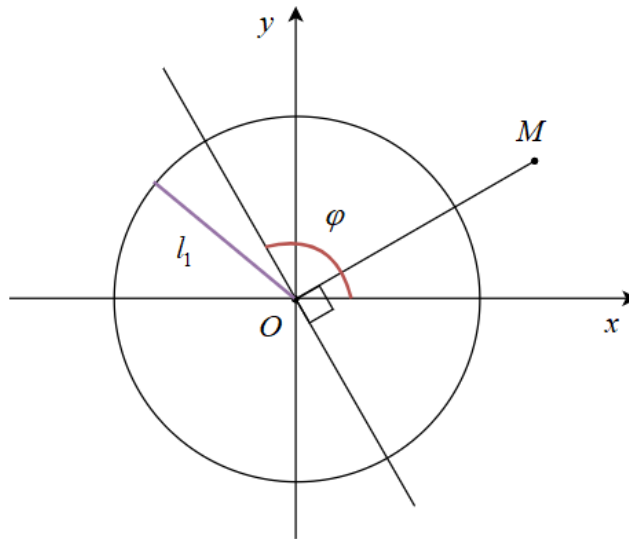


Рис. 2. Построение окружности радиуса l_1

Для удобства разделим пополам сумму длин всех звеньев d и рассмотрим случаи, когда:

- $OM \in \left[0, \frac{d}{2}\right]$;
- $OM \in \left[\frac{d}{2}, d\right]$.

Начнем с первого случая и рассмотрим его более подробно. Исходя из практического анализа, если расстояние до точки M попадает в промежуток $\left[0, \frac{d}{2}\right]$, то угол отклонения первого звена от перпендикуляра составит $\left[\varphi + \frac{\pi}{2}, \varphi - \frac{\pi}{4}\right]$ (рис 3). Если угол будет выходить за пределы данного промежутка, то при последовательном соединении точек схвата манипулятора его

звенья не будут образовывать выпуклый многоугольник. Заметим, что на выпуклых манипуляторах чаще выполняется критерий минимизации суммарного момента.

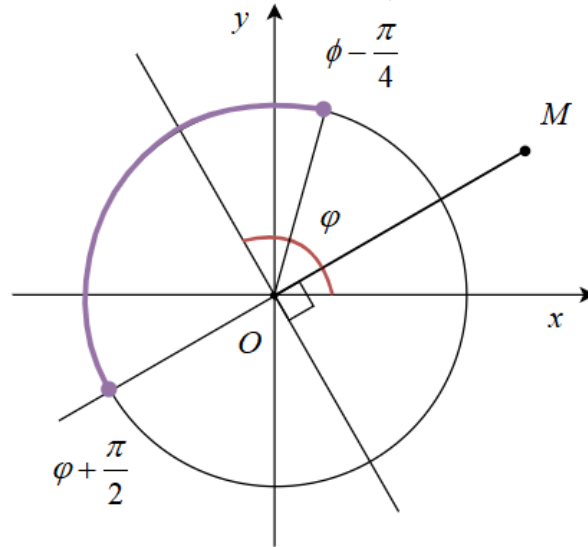


Рис. 3. Угол отклонения первого звена манипулятора в случае, когда $OM \in \left[0, \frac{d}{2}\right]$

Теперь рассмотрим второй случай, когда $OM \in \left[\frac{d}{2}, d\right]$. В предыдущей статье [1] была рассмотрена ситуация, при которой $OM = \frac{d}{2}$. Тогда угол отклонения первого звена от перпендикуляра оказался лежащим в промежутке $\left[\varphi - \frac{\pi}{4}; \varphi + \frac{\pi}{8}\right]$, а диапазон углов отклонения второго звена попал в интервал $\left(\varphi_1 - \frac{\pi}{2}, \varphi_1\right)$.

При постепенном приближении точки M к предельному значению, диапазон углов отклонения первого звена постепенно сужается и смещается к случаю, когда $OM = d$ и все звенья выстраиваются в одну линию.

После построения первого звена переходим к построению второго. От конечной точки первого звена с координатами (x_1, y_1) проводится отрезок до точки M и обозначается как O_1M . К нему проводится перпендикуляр и строится окружность радиуса l_2 с центром в точке O_1 . Угол между осью OX и вновь построенным перпендикуляром обозначается как φ_1 .

Таким образом происходит переход от построения четырехзвенного манипулятора к построению трехзвенного манипулятора.

В случае, когда $OM \in \left[0, \frac{d}{2}\right]$ диапазон угла отклонения второго звена, начиная с промежутка $\left[\varphi_1 - \frac{\pi}{2}, \varphi_1\right]$ начнет равномерно сужаться к значению угла φ_1 . В другом случае, когда $OM \in \left[\frac{d}{2}, d\right]$, наоборот, диапазон угла отклонения будет стремиться к значению $\varphi_1 - \frac{\pi}{2}$. Но при этом в обоих случаях полученный манипулятор необходимо будет проверять на выпуклость.

Построение третьего и четвертого звеньев манипулятора сводится к задаче пересечения двух окружностей, т. к. нам известны координаты центров данных окружностей $((x_2, y_2)$ и $(x_4, y_4))$ и длины их радиусов, равные l_3 и l_4 соответственно.

Заключение

В ходе практического анализа для четырехзвенного манипулятора на плоскости были рассмотрены два случая, при которых расстояние до точки M попадает либо в промежуток $\left[0, \frac{d}{2}\right]$, либо $\left[\frac{d}{2}, d\right]$. Для данных промежутков были установлены диапазоны углов отклонения для построения первого и второго звеньев. Построение третьего и четвертого звеньев будет решаться с помощью задачи пересечения двух окружностей, в ходе которой станет известно, существует ли решение поставленной ОЗК или не существует.

Литература

1. Блиняева, А. Ю. О задаче поиска диапазона углов отклонения звеньев манипулятора / А. Ю. Блиняева, С. Н. Медведев // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 13–15 декабря 2021 г. – Воронеж, 2022. – С. 155–159.
2. Медведев С. Н., Яковлев А. Ю., Корольков О. Г. Разработка метода определения присоединенных параметров манипулятора с вращающимися звеньями на основе генетических алгоритмов. [Электронный ресурс]: Научная электронная библиотека eLIBRARY.RU. Режим доступа: https://elibrary.ru/download/elibrary_36486344_89732949.pdf
3. Обзор методов решения обратной задачи кинематики. Режим доступа: http://www.artechsar.com/hp/task_ozk.shtml. – (Дата обращения: 12.04.2022).

Информация об авторах

Блиняева Александра Юрьевна — студент 2-го курса магистратуры кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: sasha.blinyaeva@gmail.com

ПРИМЕНЕНИЕ КЛАССОВ ТИПОВ, АЛГЕБРАИЧЕСКИХ СТРУКТУР И КОНЦЕПЦИЙ ТЕОРИИ КАТЕГОРИЙ В РЕАЛИЗАЦИИ ПОЛИМОРФНЫХ МЕТОДОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ SCALA

Я. А. Боль

Воронежский государственный университет

Введение

Теория категорий в частности и математика вообще часто используются как основа для написания более качественного кода и давно закрепили себя как основа многих функциональных языков, например, Haskell. Однако, практическое применение языка Haskell ограничивается, в основном, академическими кругами. В рамках данной работы будет рассмотрен более распространённый в коммерческих проектах функциональный (мультипарадигменный) язык Scala и его возможности для реализации наиболее часто используемых концепций теории категорий и общей алгебры для решения задач написания полиморфных функций и методов с использованием классов типов.

1. Реализация полиморфизма с помощью классов типов

1.1. Общие определения

Обобщённое программирование — парадигма программирования, кратко описать которую можно как написание алгоритмов (функций и методов), применимых к различным типам данных без изменения этих алгоритмов и типов. Основным понятием обобщённого программирования является **полиморфизм** — способность алгоритма работать на входных данных различных типов без необходимости модификации.

Классы типов — инструмент системы типов, позволяющий определить функциональность для отдельно взятого типа не путём наследования, модификации или расширения типа, но путём реализации независимой таблицы виртуальных методов для типа или множества типов.

1.2. Специфика реализации классов типов в языке Scala

Для реализации полиморфизма с помощью классов типов потребуется ознакомление со следующими основными концепциями.

Трейт (англ. trait) — термин языка Scala, обозначающий абстракцию, используемую для организации кода с целью облегчения обмена информацией о типах, значениях и методах. Трейты аналогичны понятию interface языка Java, но, в силу их функциональности, только начиная с версии Java 8.

Объект в объектно-ориентированном программировании — понятие, чаще всего обозначающее экземпляр класса. Так как язык Scala работает в окружении JVM, объекты являются неотъемлемой частью языка и многих его понятий. Так, например, конкретные реализации трейтов, обеспечивающие функциональность классов типов, являются объектами.

Имплицитные (неявные) параметры (англ. implicit) — параметры, которые не указываются явно при вызове метода, а передаются компилятором за счёт нахождения нужных значений по их типу в списке неявных параметров в области видимости, в которой произошёл вызов метода.

Также в реализациях классов типов в Scala достаточно часто используется синтаксический сахар **Context Bound** (примерно можно перевести как «ограничение контекста»). С его помощью значительно упрощаются сигнатуры многих методов, так как неявные параметры функции, имеющие типы высшего порядка, можно записать справа от обобщённого типа через двоеточия.

2. Алгебраические концепции и их реализация

2.1. Простейшие алгебры

Магма — простейшая алгебраическая структура, состоящая из множества M с бинарной операцией $M \times M \rightarrow M$, причём множество замкнуто относительно операции \times . Сама по себе магма не представляет особого интереса для изучения в силу своей простоты, однако, интересны её подвиды.

Полугруппа — магма, операция которой обладает свойством ассоциативности. То есть, для операции \times выполняется свойство:

$$(a \times b) \times c = a \times (b \times c), \quad \forall a, b, c \in M.$$

Моноид — полугруппа, имеющая нейтральный элемент $e \in M$ такой, что

$$m \times e = e \times m = m, \quad \forall m \in M.$$

То есть, при применении операции к нейтральному элементу и любому другому выбранному элементу множества получается не изменённый выбранный элемент.

2.2. Реализация концепций на языке программирования Scala

Полугруппа

Полугруппа (англ. semigroup) может быть реализована как трейт, являющийся основой класса типов, который позволяет совершать бинарную операцию над элементами неизвестного (любого) типа T , не нарушая при этом строгой типизации языка.

Так, для полугруппы основной трейт будет выглядеть следующим образом:

Листинг 1

```
trait Semigroup[T] {  
  def op(t1: T, t2: T): T  
}
```

Этот класс типов может быть реализован для любого типа T , для которого эта реализация будет иметь смысл. Например, для типа целых чисел реализация представлена в листинге 2.

Листинг 2

```
implicit val s: Semigroup[Int] = new Semigroup[Int] {  
  override def op(t1: Int, t2: Int): Int = t1 + t2  
}
```

Эта реализация выглядит слишком громоздкой, и её можно упростить, используя синтаксический сахар языка Scala (SAM, single abstract method), что позволяет сократить определение до одного обозначения операции:

Листинг 3

```
implicit val s: Semigroup[Int] = _ + _
```


Моноид

Моноид (англ. monoid) также может быть реализован как класс типов, однако, инструменты языка Scala позволяют упростить реализацию за счёт переиспользования кода с помощью наследования. Под наследованием понимается наследование классов (в значении объектно-ориентированного программирования в рамках JVM, не путать с классом типов).

Так как моноид по определению является полугруппой с нейтральным элементом, это определение можно отразить и в его реализации:

Листинг 4

```
trait Monoid[T] extends Semigroup[T] {  
  val neutral: T  
}
```

Здесь директива *extends* обозначает наследование полугруппы моноидом, и необходима только сигнатура нейтрального элемента.

В листинге 5 представлена реализация моноида для типа строк в рамках операции конкатенации (сложения).

Листинг 5

```
implicit object m extends Monoid[String] {  
  val neutral: String = «»  
  def op(t1: String, t2: String): String = t1 + t2  
}
```

Нейтральным элементом служит пустая строка, а операция представляет из себя конкатенацию строк (вызов метода сложения).

3. Концепции теории категорий и их реализация

3.1. Определения

Функтор — отображение $F : C \rightarrow D$ из категории C в категорию D .

Ковариантный функтор

Функтор является **ковариантным**, если:

1. Каждому морфизму $f : X \rightarrow Y$ категории C функтор сопоставляет морфизм $F(f) : F(X) \rightarrow F(Y)$ в категории D . Это сопоставление имеет следующие ограничения:

$$F(id_A) = id_{F(A)}$$

$$F(f \circ g) = F(f) \circ F(g)$$

2. Функтор сопоставляет каждому объекту X категории C объект Y категории D .

Контравариантный функтор

Контравариантный функтор определяется аналогично, однако, такой функтор сопоставляет морфизму $f : X \rightarrow Y$ категории C морфизм $F(f) : F(Y) \rightarrow F(X)$ в категории D . Контравариантные функторы также соблюдают ограничения на сохранение тождественных морфизмов и композицию функторов.

3.2. Реализация концепций на языке программирования Scala

Ковариантный функтор

Ковариантный функтор абстрагирует отображения для типов, часто используемые на практике. Определение ковариантного функтора:

Листинг 6

```
trait CovariantFunctor[F[_]] {  
  def map[A, B](a: F[A])(f: A => B): F[B]  
}
```

Тип F в определении функтора — абстракция над типами высшего порядка, для которых будет определено преобразование. Ими могут служить как классы типов, так и повсеместно используемые типы, например, коллекции и монады.

Так, например, можно определить ковариантный функтор для типа односвязного списка из стандартной библиотеки языка Scala:

Листинг 7

```
implicit object covariant extends CovariantFunctor[List] {  
  def map[A, B](a: List[A])(f: A => B): List[B] = a.map(f)  
}
```

Реализация такого функтора, очевидно, тривиальна в силу того, что в самой стандартной библиотеке уже определён метод `map` для преобразования списка с помощью отображения.

Контравариантный функтор

Контравариантный функтор значительно реже используется на практике, и его практическая польза может быть несколько более сложна для понимания. Тем не менее, в некоторых случаях применение контравариантного функтора может значительно упростить написание новых классов типов.

Дадим следующее определение:

Листинг 8

```
trait ContravariantFunctor[F[_]] {  
  def contraMap[A, B](a: F[A])(f: B => A): F[B]  
}
```

Как и в математическом определении, отличие от ковариантного функтора здесь заключается в направлении отображения f .

Рассмотрим класс типов, который позволяет сравнивать значения определённого типа. В стандартной библиотеке Scala уже существует специальный трейт `Ordering[T]` с методом `compare`, а также его реализации для основных типов, например, целочисленного типа `Int`. При решении задачи отсортировать список, например, людей по их возрасту в годах, можем ли мы переиспользовать уже существующий `Ordering[Int]`?

В решении такой задачи нам и поможет контравариантный функтор. Его определение для класса типов `Ordering` и пример использования в сортировке списка представлены в листинге 9.

Листинг 9

```
implicit object contravariant extends ContravariantFunctor[Ordering] {  
  def contraMap[A, B](a: Ordering[A])(f: B => A): Ordering[B] =  
    Ordering.by(f)(a)  
}  
case class Human(age: Int)  
implicit val ageOrdering: Ordering[Human] =  
  contravariant.contraMap(Ordering[Int])(_.age)  
println(List(Human(20), Human(10)).sorted)
```

Программа в листинге 9 напечатает список людей в корректном порядке (по возрастанию). **Замечание:** метод `sorted` ничего не знает о типе списка, а сравнивать элементы может за счёт того, что принимает неявный параметр типа `Ordering`, и использует предоставленную реализацию в ходе сортировки списка. Это — пример реализации полиморфизма с помощью классов типов.

4. Практические примеры

В качестве заключительных примеров решим задачи на написание абстрактных методов с простой логикой, на которых можно продемонстрировать гибкость полиморфизма с использованием классов типов.

4.1. Сложение элементов коллекции с самими собой

Хорошим примером применения продемонстрированных классов типов в реализации полиморфных методов может служить задача сложить каждый элемент какой-либо коллекции с самим собой. Очевидно, что коллекции представляют из себя множество типов высшего порядка, для которых могут быть определены ковариантные функторы; типы элементов коллекций, чаще всего, мономорфны, и в рамках такой задачи для них можно определить моноид. **Замечание:** при использовании классов типов стандартной практикой является использовать наименее «сильный» класс, то есть, с меньшим количеством неиспользуемых нами методов. В данном случае, так как мы используем операцию сложения и не используем нейтральный элемент, стоит определить полугруппу, а не моноид. Решение задачи и примеры применения представлены в листинге 10.

Листинг 10

```
def sumWithItself[F[_]: CovariantFunctor, T: Semigroup](a: F[T]) =  
  implicitly[CovariantFunctor[F]]  
    .map(a)(x => implicitly[Semigroup[T]].op(x, x))  
sumWithItself(List(1, 2, 3))  
sumWithItself(Vector(0.1, 0.2, 0.3))  
sumWithItself(Option(«String #1»))
```

Примеры использования включают в себя разные коллекции и разные типы данных, для которых определены указанные в сигнатуре метода (в «контекстных ограничениях») классы типов `CovariantFunctor` и `Semigroup`. Определения конкретных экземпляров классов типов тривиальны и опущены для краткости. Результаты вызовов метода представлены в листинге 11.

Листинг 11

```
List(2, 4, 6)  
Vector(0.2, 0.4, 0.6)  
Some(String #1String #1)
```

4.2. «Опустошение» коллекции

Рассмотрим задачу замены всех элементов в коллекции на некий «пустой» элемент. Так, для массива чисел [1, 2, 3] результатом алгоритма будет массив [0, 0, 0]. **Замечание:** с учётом практик функционального программирования, в данной задаче будет возвращаться не та же самая коллекция с изменениями, а копия коллекции такой же размерности, заполненная нейтральными элементами конкретного типа.

Для заполнения коллекции нейтральными элементами потребуется два контекстных ограничения: ковариантный функтор для типа высшего порядка коллекции и моноид для моно-

морфного типа элемента коллекции. Функция f , таким образом, будет отображением из всего входного множества в один элемент, нейтральный.

Решение задачи и примеры применения представлены в листинге 12.

Листинг 12

```
def empty[F[_]: CovariantFunctor, T: Monoid](a: F[T]) =  
  implicitly[CovariantFunctor[F]]  
    .map(a)(_ => implicitly[Monoid[T]].neutral)  
empty(List(1, 2, 3))  
empty(Vector(0.1, 0.2, 0.3))  
empty(Option(«String #1»))
```

Реализации классов типов опущены для краткости. Результаты вызовов метода представлены в листинге 13.

Листинг 13

```
List(0, 0, 0)  
Vector(0.0, 0.0, 0.0)  
Some()
```

В результате вызовов метода для разных типов коллекций и элементов осуществляется отображение элементов без потери информации о типах, то есть, сохраняются преимущества сильной типизации.

Заключение

В результате исследования были описаны и наглядно продемонстрированы реализации концепций из алгебры и теории категорий, а также их использование при написании полиморфных методов в рамках системы типов языка Scala как классов типов. Очевидным преимуществом их применения является возможность реализации полиморфизма не путём наследования типов, но путём реализации категорий типов.

Литература

1. The Scala Programming Language. – Режим доступа: <https://scala-lang.org>. – (Дата обращения: 01.04.2022).
2. How to make ad-hoc polymorphism less ad hoc. – Режим доступа: <https://users.csc.calpoly.edu/~akeen/courses/csc530/references/wadler.pdf>. – (Дата обращения: 01.04.2022).
3. Scala with Cats 2. – Режим доступа: <https://www.scalawithcats.com>. – (Дата обращения: 01.04.2022).
4. Monoid. – Режим доступа: <https://encyclopediaofmath.org/wiki/Monoid>. – (Дата обращения: 01.04.2022).
5. Category Theory. – Режим доступа: <https://plato.stanford.edu/entries/category-theory>. – (Дата обращения: 01.04.2022).

Боль Ян Александрович — магистрант 2-го года обучения кафедры МОЭВМ Воронежского государственного университета. E-mail: yan.bol.official@gmail.com

Борисенков Дмитрий Васильевич (научный руководитель) — канд. техн. наук, доцент кафедры МОЭВМ Воронежского государственного университета. E-mail: xuser@relex.ru

ПРОГНОЗИРОВАНИЕ СКЛАДСКИХ ЗАПАСОВ ДЛЯ ТОВАРОВ С ПРЕРЫВИСТЫМ СПРОСОМ МОДИФИЦИРОВАННЫМ БУТСТРЕП-МЕТОДОМ

Ю. В. Бондаренко, С. Ю. Будаева

Воронежский государственный университет

Введение

Материальные запасы (англ. *inventories*) в теории логистики представляют собой товарно-материальные ценности, имеющиеся в наличии в конкретный момент времени и ожидающие производственного или личного потребления. Запас играет роль буфера между потоками поставок (производства) и спроса и обеспечивает непрерывность потребления в случае колебания указанных потоков.

Управление запасами представляет собой одну из важнейших и ответственных функций в логистике. С одной стороны, наличие достаточного количества запасов способствует повышению качества обслуживания потребителей, обеспечивает стабильность и непрерывность процессов производства, распределения и сбыта. С другой стороны, запасы представляют собой «замороженные» оборотные денежные средства предприятия. Издержки хранения материальных запасов занимают второе место после производственных издержек [1].

В этой связи оптимизация запасов остается актуальной задачей руководства компании и представляет собой мощный инструмент, способствующий повышению эффективности производства, улучшению потребительских свойств продукции, росту конкурентоспособности предприятия.

1. Классификация моделей спроса

Задача своевременного удовлетворения потребностей клиентов, без создания при этом чрезмерных запасов товаров, продуктов, сырья, запасных частей и комплектующих решается в рамках модели управления запасами организации. Любая модель управления запасами, в конечном счете, отвечает на вопрос: когда и какое количество продукции заказывать или изготавливать, учитывая уже имеющееся на складе количество, текущий спрос на продукт и скорость доставки необходимого сырья поставщиками.

Выбор модели управления запасами существенно зависит от модели спроса [2], которая чаще всего определяется по временным рядам спроса. Существует несколько методик классификации спроса для поиска наиболее точного метода прогнозирования. Так, например, Синтетос и Бойлан в работе [3] на основе эмпирических данных предлагают использовать двумерную схему, в которой для определения предсказуемости спроса применяются два параметра:

- 1) Средний интервал спроса (англ. *Average Demand Interval — ADI*).
- 2) Квадрат коэффициента вариации (англ. *Coefficient of Variation — CV*).

Параметр *ADI* характеризует прерывистость спроса и определяется как среднее количество периодов времени между двумя последовательными потребностями. Параметр CV^2 указывает на изменчивость спроса и определяется как квадрат отношения стандартного отклонения данных о спросе к среднему значению спроса. Авторами рекомендуется использовать $ADI = 1,32$ и $CV^2 = 0,49$ в качестве пороговых значений. Основываясь на этих двух измерениях, модели спроса классифицируются по четырем различным категориям [4] (рис. 1).

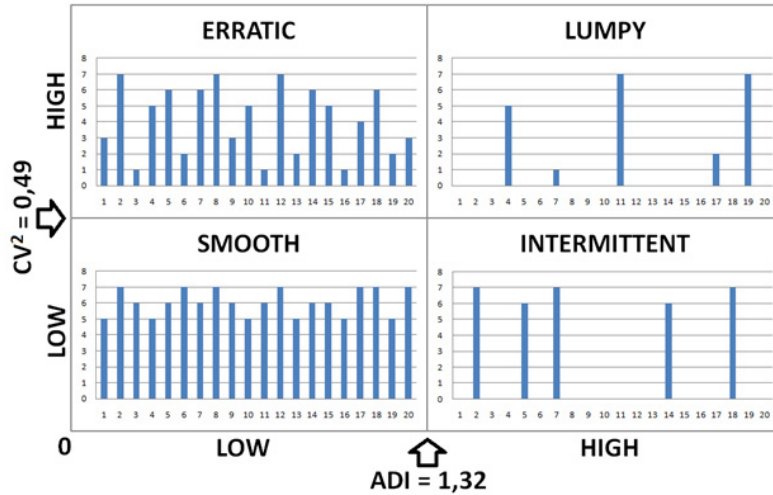


Рис. 1. Классификация моделей спроса по Синтетосу и Бойлану

Для регулярно потребляемых товаров (*smooth* и *erratic*, рис. 1) сначала строят прогноз продаж на основе прошлой статистики, а затем вычисляют страховой запас. При суммировании основного и страхового запасов получается необходимый запас, который с заданным уровнем сервиса (англ. *service level* — *SL*) обеспечит удовлетворение будущего спроса [5].

Исследования многих авторов подтверждают тот факт, что спрос, чаще всего, имеет случайный характер с большой долей нулевых значений, вариативностью интервалов между точками возникновения спроса, а также вариативностью его величины (*intermittent* и *lumpy*, рис. 1). Сложный характер временных рядов, вынуждает отказаться от использования традиционных методов прогнозирования: скользящего среднего, экспоненциального сглаживания, авторегрессии и других. Рекомендуется сразу определять оптимальный запас (англ. *optimal stock* — *OS*), не разделяя его на страховой запас и прогнозируемый спрос [5]. В настоящей статье будем рассматривать товары с прерывистым спросом.

2. Параметрические и непараметрические методы прогноза прерывистого спроса

Подходы к прогнозированию прерывистого спроса, в целом, можно разделить на *параметрические* и *непараметрические*. Первые предполагают, что будущий спрос может быть достаточно хорошо представлен некоторым статистическим распределением, например, нормальным или пуассоновским, с неизвестными параметрами — средним значением и дисперсией, которые могут быть вычислены на основе истории спроса. Вторые исходят из того, что статистическое распределение изначально неизвестно и определяют его параметры эмпирически по имеющимся данным. Многие параметрические методы представляют собой вариации метода Кростона [6], а непараметрические основаны на бутстреп-процедуре.

Метод Кростона предполагает разделение исходного прерывистого ряда данных на два — ряд из ненулевых значений спроса и ряд временных интервалов между ненулевыми значениями. После этого к каждому из получившихся рядов применяется простое экспоненциальное сглаживание с некоторым коэффициентом α . Если время между запросами обозначить через s_t , а размер спроса, наблюдаемый в период t через z_t , то метод Кростона описывается формулами (1) и (2):

$$\tilde{s}_t = \alpha \cdot s_t + (1 - \alpha)\tilde{s}_{t-1}, \quad (1)$$

$$\tilde{z}_t = \alpha \cdot z_t + (1 - \alpha)\tilde{z}_{t-1}. \quad (2)$$

Прогнозные значения размера спроса \tilde{z}_t и времени между ненулевыми значениями \tilde{s}_t обновляются только в периоды с положительным спросом. Для вычисления оценки спроса в единицу времени используют формулу (3):

$$\hat{Y}_t = \tilde{z}_t / \tilde{s}_t. \quad (3)$$

Метод Кростона основан на следующих предположениях:

- вероятность возникновения/отсутствия спроса имеет распределение Бернулли;
- для моментов времени, когда спрос больше нуля, вероятность для размера спроса имеет нормальное распределение.

Далеко не для всех товаров указанные предположения выполняются, что ограничивает применение метода Кростона. Существуют непараметрические методы, которые позволяют реконструировать эмпирическое распределение данных, что делает излишними предположения об их изначальном распределении. Часто для этой цели применяют бутстреп-процедуру (англ. *bootstrap*; в иной транслитерации: бутстрэп, бутстрап) — метод генерации повторных выборок.

Применительно к задаче прогнозирования запасов товаров с прерывистым спросом метод бутстрепинга (метод Виллемейна [7]) позволяет оценить, с какой вероятностью может быть продан тот или иной объем товара, то есть рассчитать накопительную вероятность возникновения спроса. Это позволяет определить размер запаса, достаточного для удовлетворения спроса с известным уровнем сервиса на период поставки (англ. *lead time* — *LT*).

Для этого из имеющегося временного ряда исторических наблюдений случайным образом извлекают несколько значений спроса, соответствующих периоду поставки, и суммируют их. Эту процедуру повторяют многократно (1000–10000 раз), а затем строят кумулятивную функцию распределения (англ. *Cumulative Distribution Function* — *CDF*) для полученных сумм выбранных значений. Оптимальный размер запаса (без разделения на основной и страховой) устанавливается на уровне, который обеспечит удовлетворение спроса с заданной доверительной вероятностью, равной выбранному уровню сервиса.

3. Метод Кростона

Рассмотрим два временных ряда, описывающие спрос на некоторые товары в течение трех лет помесечно (рис. 2).

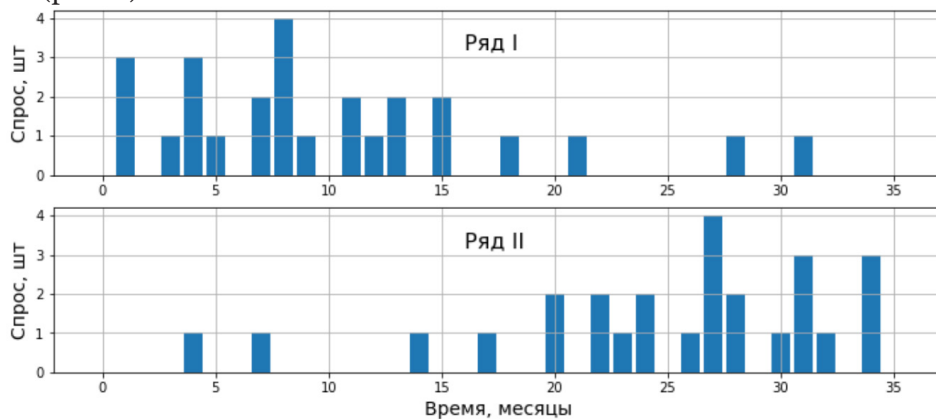


Рис. 2. Тестовые временные ряды I и II

Ряд II получен из ряда I зеркальным отражением «слева направо». Ряд I характерен для товаров, выходящих из употребления, ряд II, наоборот, для новых товаров. Оба ряда, естественно, имеют одинаковые статистические характеристики — среднее значение и дисперсию. Поскольку $ADI_I = ADI_{II} = 2,4$, а $CV_I^2 = CV_{II}^2 = 0,31$, то оба ряда, в соответствии с классификацией моделей спроса (рис. 1), относятся к рядам с прерывистым спросом.

Вычислим прогнозные значения размера спроса, времени между ненулевыми значениями спроса, а также среднее значение прогноза размера спроса для обоих рядов по методу Кросто-

на на период поставки два месяца ($LT = 2$) и коэффициентом сглаживания $\alpha = 0,2$. Для этого воспользуемся скриптом [8] на языке Python. Результаты работы скрипта представлены на рис. 3 и в табл. 1.

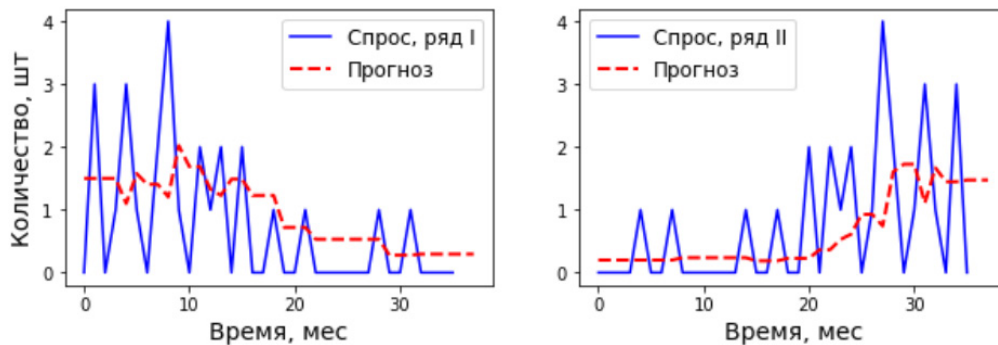


Рис. 3. Прогноз спроса по методу Кростона

Таблица 1

Результаты прогнозирования методом Кростона

Результат	Ряд I	Ряд II
Среднее значение прогноза спроса (шт./мес.)	0,29	1,47
Прогноз спроса (шт.)	1,11	2,25
Прогноз интервала возникновения спроса (мес.)	3,76	1,52

Методом классического бутстреппинга определим размер запаса, достаточного для удовлетворения спроса с уровнем сервиса, например, $SL = 90\%$ (обслуживаются девять из каждых десяти клиентов) на тот же период поставки $LT = 2$. Полученная гистограмма, показывающая частоты возникновения того или иного спроса для рядов I и II показана на рис. 4, причем интервалы с нулевым спросом представлены с наибольшей частотой.

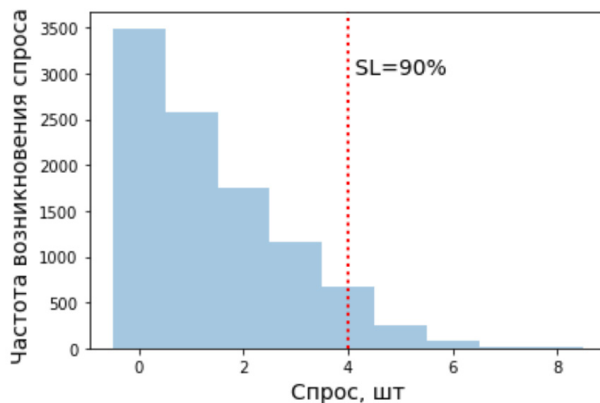


Рис. 4. Частоты возникновения спроса для рядов I и II по методу бутстрепа

В отличие от метода Кростона, полученные результаты для оптимального размера запаса оказались одинаковы для обоих рядов, поскольку классический бутстреп с равномерной выборкой не учитывает любую потенциальную автокорреляцию данных, характерную для временных рядов. В данном случае, с равной вероятностью были учтены события, произошедшие как три года назад, так и события месячной давности.

4. Бутстрепинг с неравномерной выборкой

Одним из вариантов решения проблемы, связанной с устареванием данных является ограничение длины временного ряда. Ряд длиной N разбивается на две части: длиной $(N - n)$ и n соответственно. Тогда при проведении процедуры бутстрепинга вероятность выборки элемента будет неравномерной: из первой части $p_1 = 0$, а из второй $p_2 = 1/n$. Недостатками данного подхода являются потеря наблюдений для модели прогнозирования, потенциально приводящая к уменьшению точности прогноза, и трудности выбора алгоритма для автоматического вычисления оптимального значения n .

Алгоритм, позволяющий придавать больший вес более поздним наблюдениям, применяется в основе метода экспоненциального сглаживания (англ. *Simple Exponential Smoothing* — *SES*) [9]. При этом прогнозы рассчитываются с использованием средневзвешенных значений, где наименьшие веса связаны с самыми ранними наблюдениями по формуле (4):

$$\tilde{y}_t = \alpha \cdot y_t + (1 - \alpha)\tilde{y}_{t-1}. \quad (4)$$

Здесь \tilde{y}_t — экспоненциальная средняя, y_t — текущее значение ряда, α — параметр сглаживания, характеризующий вес текущего, самого нового, наблюдения ($0 < \alpha < 1$).

Предлагается применить аналогичный подход и для бутстреп-процедуры. Каждой позиции i временного ряда длиной n необходимо присвоить свой вес w_i так, чтобы выполнялось условие (5):

$$\sum_{i=1}^n w_i = 1. \quad (5)$$

Значение веса w_i численно равно вероятности p_i с которой будет производиться выборка элемента временного ряда, располагающегося на i -й позиции при бутстрепинге. Более поздние наблюдения будут извлекаться в новую подвыборку чаще, чем более ранние. При $w_i = 1/n$ получаем классический равномерный бутстреп. Необходимо заметить, что предлагаемый подход нельзя применять при определении параметров (среднего, дисперсии, медианы) статистических распределений по имеющимся данным.

Последовательность значений w_i представляет собой геометрическую прогрессию. Значение первого члена w_1 находят из формулы суммы n первых членов прогрессии, учитывая, что в данном случае $S_n = 1$ (6):

$$S_n = \frac{w_1(g^n - 1)}{g - 1}. \quad (6)$$

Знаменатель прогрессии g предлагается выбирать эмпирически, по аналогии с выбором коэффициента сглаживания α в формуле (4) модели SES. Обычно, α берут из интервала $(0 \dots 1/3)$, априори $\alpha = 0,2$. В этом случае весовые коэффициенты представляются последовательностью, вычисляемой по формуле (7):

$$w_{i-1} = w_i(1 - \alpha). \quad (7)$$

Для выбранного значения $\alpha = 0,2$ было получено, соответствующее ему по смыслу, значение знаменателя прогрессии: $g = (1 - \alpha)^{-1} = (1 - 0,2)^{-1} = 1,25$.

Python-функция *Weight(n, g)* (листинг 1) вычисляет и возвращает массив весовых коэффициентов по заданным n (длина ряда) и g (знаменатель прогрессии).

Листинг 1

Функция *Weight(n, g)*

```
def Weight(n, g):  
    w = []  
    w.append((g - 1)/(g**n - 1))  
    for i in range(n - 1):  
        w.append(w[i] * g)  
    return w
```

Python-функция $Bootstrap(ts, weight, lead_time, SL)$ (листинг 2) вычисляет и возвращает значение оптимального запаса по заданным ts (временной ряд истории спроса), $weight$ (массив весовых коэффициентов), $lead_time$ (период поставки) и SL (требуемый уровень сервиса).

Листинг 2

Функция $Bootstrap(ts, weight, lead_time, SL)$

```
def Bootstrap(ts, weight, lead_time, SL):
    stock = []
    for i in range(10000):
        sample = numpy.random.choice(ts, size = lead_time, p = weight)
        stock.append(numpy.sum(sample))
    return numpy.percentile(stock, SL)
```

На основании полученных данных строится кумулятивная функция распределения, определяющая вероятность некоторого целого числа спроса x или любого числа ниже x , случайным образом выбранных из распределения для ряда I (рис. 5а) и ряда II (рис. 5б).

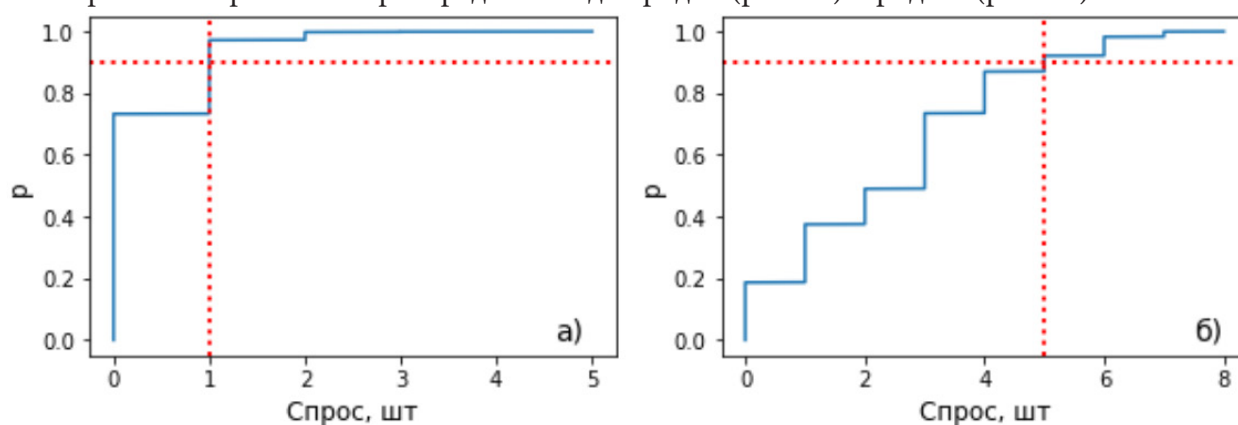


Рис. 5. Графики кумулятивной функции распределения

Полученные значения оптимального запаса $OS_I = 1$ и $OS_{II} = 5$ численно коррелируют с решениями по методу Кростона, и визуально с характером временных рядов (рис. 2), в отличие от решений классического бутстрепа ($OS_I = OS_{II} = 4$) с равномерной выборкой.

Заключение

Прогнозирование прерывистого спроса является сложной задачей. Предложенная модификация бутстреп-метода не позволяет точно указать время появления и значение предстоящего ненулевого заказа, но улучшает качество прогнозирования, особенно при нарастающем или падающем спросе. Метод позволяет получить удовлетворительный кумулятивный прогноз на длительный период времени и минимизировать затраты на содержание материальных запасов, сохраняя при этом требуемый уровень оказания услуг потребителю.

Литература

1. Гаджинский, А. М. Логистика: учебник для вузов / А. М. Гаджинский. – Москва : Издательско-торговая корпорация «Дашков и К^о», 2012. – 472 с.
2. Тяхтина, А. А. Модели управления запасами: учебно-методическое пособие / А. А. Тяхтина – Нижний Новгород : Изд-во Нижегородского университета, 2017. – 84 с.
3. Syntetos, A. On the categorization of demand patterns / A. Syntetos, J. Boylan // Journal of the Operational Research Society. – 2005. – № 56. – P. 495–503.

4. Demand classification: why forecastability matters. – Режим доступа: <https://medium.com/@frepple/demand-classification-why-forecastability-matters-2d65d2d67954>. – (Дата обращения: 04.04.2022).

5. Дроздов, П. А. Управление запасами в цепях поставок: учебно-методическое пособие / П. А. Дроздов. – Минск: ИБМТ БГУ, 2014. – 103 с.

6. Croston, J. D. Forecasting and stock control for intermittent demands / J. D. Croston // Operational Research Quarterly. – 1972. – № 23. – P. 289–304

7. Willemain, T. R. A new approach to forecasting intermittent demand for service parts inventories / T. R. Willemain, C. N. Smart, H. F. Schwarz // International Journal of Forecasting. – 2004. – Vol. 20. – P. 373–387.

8. Forecasting Intermittent Demand with the Croston Model. – Режим доступа: <https://towardsdatascience.com/croston-forecast-model-for-intermittent-demand-360287a17f5f>. – (Дата обращения: 05.04.2022).

9. Лукашин, Ю. П. Адаптивные методы краткосрочного прогнозирования временных рядов: учебное пособие / Ю. П. Лукашин – Москва : Финансы и статистика, 2003. – 416 с

Будаева Светлана Юрьевна — магистрант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: lanabudaeva777@gmail.com

Бондаренко Юлия Валентиновна (научный руководитель) — д-р техн. наук, доцент, проф. кафедры математических методов исследования операций Воронежского государственного университета. E-mail: bond.julia@yandex.ru

АВТОМАТИЗАЦИЯ РЕБАЛАНСИРОВКИ ПОРТФЕЛЯ ЦЕННЫХ БУМАГ КАК СПОСОБ ЗАЩИТЫ ОТ ФАКТОРОВ, ВЛИЯЮЩИХ НА ФОНДОВЫЙ РЫНОК

И. М. Бухольцев

Воронежский государственный университет

Введение

Одним из вариантов защиты своего портфеля ценных бумаг от внешних факторов рынка является ребалансировка. Однако, при проведении ребалансировки портфеля необходимо учитывать множество факторов, и тут нам поможет автоматизация. Приложение для автоматизации ребалансировки портфеля поможет проводить ее как неквалифицированным инвесторам, так и профессионалам.

1. Определение автоматизации

Портфель ценных бумаг — это инструмент инвестирования, с помощью которого инвестор наиболее эффективным способом добивается желаемого уровня доходности при заданном уровне риска.

У термина «*ребалансировка*» в инвестировании существует множество обозначений, таких как:

- 1) приведение фактического портфеля в соответствие с текущим стратегическим распределением активов из-за ценовых изменений в портфеле;
- 2) пересмотр весов целевых классов активов инвестора из-за изменений в его инвестиционных целях или ограничениях, или из-за изменений в его ожиданиях рынка капитала;
- 3) тактическое распределение активов.

Существует множество факторов рынка, которые способны повлиять на стоимость, доходность, риск, ликвидность и другие характеристики как отдельных активов в составе портфеля инвестора, так и их совокупности.

На изменение рынка оказывают влияние как внешние, так и внутренние факторы. К основным внутренним факторам относятся рост ВВП, развитие производства в промышленной отрасли, объем вложений в основной капитал, индекс цен, монетизация экономики, кредитная процентная ставка, остатки средств на банковских счетах, ставка рефинансирования, изменения денежного предложения [3].

Среди внешних факторов, оказывающих влияние на фондовый индекс (в нашем случае РТС) можно выделить следующие: курс рубля, международные резервы России, остаток торгового баланса [3].

Одним из методов защиты портфеля от их влияния является его ребалансировка.

По сути, ребалансировка — это регулярный или ситуационный пересмотр структуры инвестиционного портфеля. Данный метод помогает инвесторам восстанавливать баланс активов в портфеле, приводя его в норму в соответствии с заданными при формировании инвестиционной политикой и целями.

Однако, ребалансировка портфеля может быть довольно времязатратной даже для квалифицированного инвестора, не говоря уже о новичках в этой сфере. В данном случае нам поможет автоматизация данного процесса.

Для инвестирования создано множество инструментов, которые позволяют упростить работу с рынком ценных бумаг.

Автоматизация ребалансировки портфеля происходит с использованием соответствующего приложения, которое поможет любому инвестору значительно сократить время на данный процесс, а также выполняет многие другие функции. Как пример, программа позволяет видеть портфель и понимать изменения в портфеле, давать подсказки о проведении ребалансировки с учетом котировок и т. д.

2. Возможности автоматизации

Необходимость применения программы в целях ребалансировки портфеля обосновывается многими преимуществами данного процесса:

1. Наглядность для неквалифицированных инвесторов
2. Возможность проверить свои идеи заранее, без проведения дополнительных вычислений
3. Сокращение времени на ребалансировку
4. Возможность прогнозирования результата

и др.

Существует огромное количество неквалифицированных инвесторов — новичков в этой сфере, которые только начинают свою работу с ценными бумагами. Для таких людей наглядность процесса ребалансировки будет являться плюсом, так как это позволит гораздо проще работать со своим портфелем. Визуализация процесса и результата работы портфеля с помощью приложения позволяет проще осознавать свои возможности и результаты работы своего портфеля.

Еще одним преимуществом является сокращение времени на ребалансировку портфеля. При ребалансировке портфеля необходимо учитывать множество данных, таких как стоимость каждого актива в портфеле, комиссия брокеру, изменения во внешних и внутренних факторах рынка и т. д.

Так как при проведении ребалансировки предполагается простой компромисс: затраты на ребалансировку должны быть ниже затрат на отсутствие ребалансировки [2], то имеется необходимость в большом количестве расчётов. Безусловно, данные расчеты можно производить и без использования каких-либо средств автоматизации, однако такой способ подойдет далеко не каждому.

Начинающие инвесторы часто не разбираются в теориях рынка, не знают формул для расчетов и не имеют возможности либо желания учитывать все возможные факторы, которые участвуют в процессе ребалансировки портфеля. В таком случае именно автоматизация позволяет облегчить данный процесс и существенно ускорить его проведение для неквалифицированных инвесторов. Кроме того, стоит заметить, что даже квалифицированные инвесторы вынуждены тратить приличное количество времени на этот процесс. Хотя у каждого из них уже выработан свой способ для расчета и проведения ребалансировки портфеля, именно использование в этих целях специализированного приложения считаю более разумным, так как таких ребалансировок при нестабильной ситуации рынка может понадобиться большое количество, и в этих условиях значительно быстрее будет автоматизировать данный процесс.

Возможности автоматизации ребалансировки состоят не только в пересмотре портфеля и приведении его к целевому состоянию, но и в прогнозировании результата его работы. Не стоит забывать, что при ребалансировке в приложении можно указывать множество параметров, которые и будут влиять на результат. Таким образом можно смоделировать любую ситуацию на рынке, которую может представить себе инвестор, опираясь на свое воображения или исторические ситуации, которые уже происходили с рынком за его существование.

Эта же возможность позволяет проверить свои идеи относительно портфеля заранее, перед или после его изменения, добавив соответствующие данные для расчетов в приложение.

Для реализации каждого из преимуществ автоматизации может использоваться как одно приложение, включающее в себя все инструменты для работы с портфелем, так и множество разных приложений, не связанных между собой.

Приложений для инвестирования на текущий момент существует большое количество, в том числе и узкоспециализированных, для расчетов конкретных данных в конкретных ситуациях. Автоматизация расчетов может быть полезна как для квалифицированных инвесторов, так и для новичков в инвестировании. Для новичков особенно полезными могут быть такие функции, как подсказки в приложениях и инструкции с примерами, которые не придется самим искать на различных ресурсах интернета.

Заключение

Автоматизация ребалансировки портфеля поможет защититься от многих факторов, влияющих на рынок ценных бумаг. Конечно, предсказать заранее изменения в макроэкономике и различные корпоративные события, которые могут сильно повлиять на ваш портфель, очень сложно, однако с помощью автоматизации ребалансировки можно проверить устойчивость своего портфеля к воздействиям факторов на основе своих предположений и исторически похожих ситуаций.

Литература

1. Всё о распределении активов / Ричард Ферри; пер. с англ. Леонида Веригина. – М. : Манн, Иванов и Фербер, 2014. – 530 с.
2. Фундаментальные основы инновационного развития науки и образования: монография / Под общ. ред. Г. Ю. Гуляева – Пенза : МЦНС «Наука и Просвещение», 2021. – 150 с.
3. Факторы, влияющие на фондовый рынок. – Режим доступа: https://spravochnick.ru/rynok_cennyh_bumag/torgovlya_na_fondovom_rynke/factory_vliyayuschie_na_fondovyy_rynok – (Дата обращения: 22.04.2022).

Бухольцев Иван Михайлович — магистрант 2-го года обучения кафедры ERP-систем и бизнес процессов Воронежского государственного университета.
E-mail: mirror.my.shadow@gmail.com

Кенин Сергей Леонидович (научный руководитель) — канд. техн. наук, доцент, и. о. зав. кафедрой ERP-систем и бизнес процессов Воронежского государственного университета.
E-mail: kenin.sergey@yandex.ru

АЛГОРИТМ РАБОТЫ ИСКУССТВЕННОЙ ИММУННОЙ СИСТЕМЫ

Е. С. Вернер

Воронежский государственный университет

Введение

Актуальность темы данной работы определяется прежде всего тем, что в современном мире системы искусственного интеллекта внедряются во все сферы человеческой жизни. Так же разработка новых отечественных технологий оказалась важна как никогда. Практически во всех отраслях деятельности людей могут быть применены генетические алгоритмы, искусственные иммунные системы, системы распознавания образов, алгоритмы машинного обучения. Современная наука позволяет сочетать как традиционные математические методы, так и прикладные алгоритмы, пришедшие из смежных областей науки.

1. Постановка задачи и её актуальность

Магистерская работа посвящена построению искусственных иммунных систем и их практическому использованию в задачах распознавания образов и автономного управления. Целью работы является изучение принципов построения искусственных иммунных систем, а также разработка и исследование конкретной модели искусственной иммунной системы для распознавания объектов в задаче автономного управления морским транспортным средством. Достижение этих целей позволит избавить человека от рутинных задач, минимизировать возможность появления ошибок, а также позволит значительно сэкономить время. Из-за человеческого фактора происходит свыше 80 % происшествий на море, он является главной причиной убытков.

Для решения этой проблемы необходимо создать соответствующую математическую модель, при построении которой используются методы системного анализа и математического моделирования, методы и алгоритмы искусственных иммунных систем и генетических алгоритмов. В данной статье будут рассмотрены искусственные иммунные системы и принципы построения математических моделей.

2. Иммунные системы

2.1. Естественные иммунные системы

Естественная иммунная система представляет собой сложную систему, состоящую из нескольких частей с различной функциональностью. Иммунная система использует многоуровневую защиту против внешних враждебных антигенов, с помощью врожденных и приобретенных защитных механизмов. Основная роль иммунной системы состоит в классификации клеток и молекул на «свои» и «чужие». Большой интерес для информатики заключается в том, что она способна обрабатывать большие объемы данных, используя параллельный и распределенный характер вычислений.

В организме человека содержится большое количество иммунокомпетентных клеток, циркулирующих во всем теле. Клетки иммунной системы, которые обладают ключевой ролью в приобретённом иммунитете, относятся к лимфоцитам и отвечают за приобретённый иммуни-

тет. Когда в организм попадает антиген, лишь малая часть клеток иммунной системы способна к его распознаванию. Такое распознавание стимулирует процессы размножения и дифференцирования лимфоцитов, что приводит к увеличению числа идентичных клеток. Такой процесс, называемый размножением клона, формирует многочисленную популяцию специфичных к этому конкретному антигену клеток, что приводит к разрушению или нейтрализации антигена. После этого, часть образовавшихся клеток сохраняется для иммунной памяти. В результате, последующее воздействие этого или похожего антигена приводит к более быстрой иммунной реакции.

Иммунные системы имеют ряд свойств, полезных при решении прикладных задач, например:

- Распознавание, которое является одним из самых важных свойств, позволяет решать задачи классификации,
- Выделение особенностей, позволяющее более качественно решать поставленные задачи,
- Разнообразие, гарантирующее, что хотя бы один из лимфоцитов сможет отреагировать на антиген,
- Обучение, являющееся способностью создавать клоны лимфоцитов для более успешной борьбы с антигенами,
- Память, позволяющая поддерживать баланс между экономией ресурсов и функцией защиты,
- Распределённый поиск, позволяющий не сосредотачиваться на одном конкретном месте, а реагировать на появление антигена в любом фрагменте организма.

2.2. Искусственные иммунные системы

В отличие от естественных иммунных систем, искусственные иммунные системы являются достаточно упрощёнными и поэтому применимы только к конкретным видам задач.

Представим искусственную иммунную систему в следующем виде:

$$I = \langle \varphi, \psi, A, M, S \rangle,$$

где I — искусственная иммунная система,

φ — пространство лимфоцитов,

ψ — множество антигенов, любых объектов, которые рассматриваются как чужеродные или потенциально опасные, в нашей задаче роль антигенов будут играть изображения, объекты на которых мы будем классифицировать. При этом для оптимизации работы алгоритма изображения будут преобразованы в бинарные,

A — мера аффинности, ставящая каждому лимфоциту и антигену число из отрезка $[0,1]$, показывающее насколько хорошо лимфоцит реагирует на конкретный антиген,

M — оператор мутации, позволяющий улучшить свойства распознавания лимфоцита,

S — оператор селекции, позволяющий оставить в системе только лучшие лимфоциты.

2. Алгоритм создания искусственной иммунной системы

Алгоритм создания искусственной иммунной системы можно разделить на следующие этапы:

На первом шаге случайным образом создаётся заданное число лимфоцитов, образующих начальную иммунную систему $Im \subset \varphi$. При решении моей задачи лимфоциты будут представлены в виде деревьев выражений.

На втором шаге подаётся антиген $g \in \psi$, являющийся в нашем случае изображением, прошедшим обработку и ставшим бинарным, а для всех лимфоцитов $l \in Im$ текущей иммунной системы считается аффинность $a_l = A(l, g)$. Аффинность определяет целевую функцию, кото-

рая показывает, насколько данный лимфоцит приспособлен и реагирует на поданный антиген. Используются функции расстояния, между двумя точками n -мерного пространства, где одна точка — лимфоцит, другая — антиген. В качестве функции аффинности можно использовать евклидово расстояние, расстояние Хэмминга или расстояние Левенштейна, в случае если лимфоцит будет представлен в виде дерева (преобразование дерева в строку).

На третьем шаге находится наилучший лимфоцит $l^* = \arg \max(a_l)$, у которого показатель аффинности самый высокий, наиболее близкий к соответствующему антигену.

На четвёртом шаге к лимфоцитам, у которых значение аффинности выше, применяется оператор мутации, вносящий изменения в структуру или значение лимфоцита: $m = \{M(l), l \in Im\}$. Мутация для лимфоцитов, представляющих собой дерево выражений, может заключаться в замене случайно выбранного действия, изменении переменной, добавлении или удалении поддерева случайно выбранного узла.

На пятом шаге оператор селекции оставляет лимфоциты с наилучшими значениями аффинности из текущего набора лимфоцитов и множества мутировавших лимфоцитов, полученного на предыдущем шаге. При этом могут как создаваться копии лучших лимфоцитов, так и совершенно новые лимфоциты. Лимфоциты, которые не могут справиться со своей задачей, удаляются из рассматриваемой модели. Размер системы всегда остаётся постоянным.

На шестом шаге, если решение l^* удовлетворяет заданному критерию или достигнуто максимальное число итераций, происходит завершение выполнения алгоритма, в противном случае происходит возврат ко второму шагу. Таким образом получаем, что на каждом шаге аффинность лучших лимфоцитов не уменьшается, а только растёт, и если после определённого числа итераций аффинность не меняется, то алгоритм также прекращает работу, так как найдено оптимальное решение.

При этом для корректной работы системы следует так же построить обучающую выборку, а затем протестировать работу системы на других входных данных.

Заключение

В данной статье был описан алгоритм работы искусственной иммунной системы. Резюмируя всё вышесказанное, можно отметить, что иммунная система решает задачу оптимизации функции, которая представляет собой аффинность. При этом стоит заметить, что алгоритм обучения иммунной системы является алгоритмом обучения без учителя, что позволяет проводить обучение без какого-либо вмешательства экспериментатора. В дальнейшем это позволит системе самостоятельно обнаруживать угрозы, используя выявленные в процессе обучения закономерности. Программу по данному алгоритму предлагается писать на языке программирования python, так как он позволяет сократить время разработки и улучшить читаемость кода.

Благодарности

Выражаю благодарность своему научному руководителю Астаховой Ирине Фёдоровне, доктору технических наук, профессору кафедры математического обеспечения ЭВМ за помощь в исследованиях.

Литература

1. Искусственные иммунные системы и их применение / под ред. Д. Дасгупты. Пер. с англ. под ред. А. А. Романюхи. – М. : ФИЗМАТЛИТ, 2006. – 314 с.

2. *Ушаков, С. А.* Разработка и исследование алгоритмов решения задач распознавания на основе искусственных иммунных систем: автореферат дис. канд. техн. наук. – Воронеж : Издательский дом ВГУ, 2015. – 16 с.

Вернер Екатерина Сергеевна — студентка 1-го курса магистратуры кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: wernerammvsu@yandex.ru

Астахова Ирина Фёдоровна (научный руководитель) — д-р техн. наук, проф., профессор кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: astachova@list.ru

ИСПОЛЬЗОВАНИЕ ПИКТОГРАФИКОВ ЧЕРНОВА В КАЧЕСТВЕ ИНСТРУМЕНТА АНАЛИЗА МНОГОМЕРНЫХ ДАННЫХ

О. С. Воронова

Воронежский государственный университет

Введение

С каждым годом появляется всё больше задач, связанных с обработкой объёмной информации. Старые подходы, применявшиеся длительное время, демонстрируют свою неэффективность при работе с многочисленными параметрами.

Поскольку основной целью любого исследования является получение адекватной информации для анализа и принятие на её основе решения, то необходимо выбирать методы, направленные на достижение эффективных результатов. Достаточно часто для анализа различных данных используется визуализация. На данный момент графическое представление информации является одним из самых эффективных методов отображения данных в наглядной форме.

Под визуализацией можно определить создание графических образов, которые представляют собой максимально информативные формы, которые воспроизводят ключевые моменты исследуемой задачи. Существует достаточно большое количество способов для визуального отображения информации. Выбор зависит от области поставленной задачи и объёма данных, которые требуется отобразить в виде графических образов. Традиционно используют такие инструменты как графики и диаграммы. Однако, когда возникает необходимость в работе с тремя и более величинами, их эффективность близится к минимуму. В большинстве современных задач довольно часто происходит столкновение с данными, состоящими из множества параметров.

Многомерный анализ — один из наиболее действенных методов исследования, направленный на повышения качества оценки результатов исследования, который использует большое число характеристик. Применение графики в задачах, использующих данный инструмент в работе, повышает уровень восприятия полученной информации и увеличивает скорость передачи данных.

Интерактивная компьютерная графика способствовала возникновению одного из направлений в проблематике искусственного интеллекта, который получил название когнитивной компьютерной графики (способствующей познанию). Под данным термином подразумевается совокупность приёмов и методов представления условий задачи в виде некоторого образа, который позволяет получить подсказку для нахождения верного решения или увидеть его сразу [1]. Применение когнитивной графики значительно расширяет возможности, направленные на обработку объёмных данных, контроль состояния процесса анализа и минимальные временные затраты на обнаружение результатов.

Среди многообразия инструментов обработки многомерных величин выделяют пиктографики, в частности «лица Чернова» [2].

1. Визуализация многомерных данных с помощью «лиц Чернова»

«Лица Чернова» — это не просто один из способов визуального представления данных, но и мощный исследовательский инструмент разведочного анализа информации. Главная идея данного метода для работы с многочисленными величинами основана на способности фикс-

сировать сложные связи между многими переменными, если они проявляются в некоторой последовательности. выделяют среди пиктографиков как наиболее интересный тип, т.к. визуальное представление данных выполнено в виде человеческого лица, где важная информация отображена в виде формы и размеров отдельных черт, например, угол между бровями, ширина лица.

Для того, чтобы оценить многомерную информацию, генерируется ряд «лиц», по которым происходит оценка результатов в рамках задачи. Каждое такое изображение представляет собой массив из двадцати элементов, значение которых будет расположено в отрезке от 0 до 1. Каждое число придаёт соответствующий вид части рисунка. В качестве основных параметров принято использовать следующие величины, продемонстрированные на рис. 1.



Рис. 1. Параметры «лиц Чернова», используемые для многомерного анализа

Указанное количество параметров не является максимальным. В зависимости от типа рассматриваемого условия их число может быть увеличено до тридцати шести.

Схематическое представление пиктографиков в зависимости от различных входных данных представлено на рис. 2, но кроме параметров отдельных элементов рисунка можно использовать различную цветовую палитру для выявления характерных зависимостей у похожих частей. Пример того, как выглядят «лица Чернова» в цвете, созданные в среде разработки RStudio [3], показан на рис. 3.

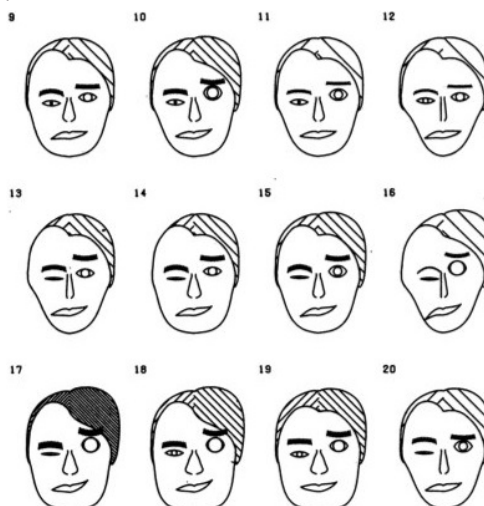


Рис. 2. Схематическое представление «лиц Чернова»



Рис. 3. Пример «лиц Чернова», сгенерированных в среде разработки RStudio

Одна из причин выбора такого изображения данных заключена в том, что для людей естественно смотреть на лица других. Поэтому результаты анализа получаются натуралистичными. В отличие от табличного представления, в них прослеживается лёгкость определения сходств, различий и объединения информации в группы.

Использование пиктографиков Чернова в задачах классификации и нахождения общих данных является эффективным инструментом, т. к. графическое представление позволяет легко распознавать граничные положения элементов «лица» для однозначного определения качественного результата.

Для реализации и изучения данного подхода существует несколько распространённых решений — программа Chern [4], разработанная в институте кибернетики Томского политехнического университета для визуализации многомерных медицинских данных с помощью «лиц Чернова», и язык программирования R [5].

2. Использование языка программирования R для работы с «лицами Чернова»

Для обработки данных и работы с графическим представлением информации существует язык программирования R [5, 6]. Основной способ работы с ним — командный интерпретатор. Несмотря на простоту и примитивность, за счёт большого числа библиотечных средств, этот язык реализует практически все универсальные статистические вычисления и создаёт качественные графические представления информации. Среди существующих расширений присутствуют те, которые позволяют обрабатывать многомерные данные и визуализировать «лица Чернова».

Для работы с языком программирования R существует среда разработки RStudio, обладающая понятным и простым интерфейсом. Чтобы сгенерировать «лица», необходимо воспользоваться методом, в котором задаются значения параметров, пример представлен в Листинге 1, или использовать готовые библиотеки с наборами данных.

Листинг 1

Структура и синтаксис для задания параметров лица

```
faces(xу, which.row, fill = FALSE, face.type = 1, nrow.plot, ncol.plot, scale = TRUE,
byrow = FALSE, main, labels, print.info = TRUE, na.rm = FALSE, ncolors = 20, col.nose =
rainbow(ncolors), col.eyes = rainbow(ncolors, start = 0.6, end = 0.85), col.hair = terrain.
colors(ncolors), col.face = heat.colors(ncolors), col.lips = rainbow(ncolors, start = 0,
end = 0.2), col.ears = rainbow(ncolors, start = 0, end = 0.2), plot.faces = TRUE)
```

Для работы с тренировочными данными выбраны восемнадцать параметров, целочисленная матрица $xу$, где строки представляют собой информацию об объектах, столбцы — названия характеристик. Если в таблице присутствуют недостающие параметры, то происходит дополнение или трансформация данных. Кроме этого указывается палитра цветов, используемая при изображении «лиц» и координаты их размещения на экране.

Классическим примером использования готовых наборов является классификация шестнадцати моделей автомобилей по их характеристикам. Пиктографики для данного примера продемонстрированы на рис. 4.

По результату можно сказать, что Cadillac Fleetwood и Lincoln Continental максимально похожи между собой, незначительные отличия имеются у моделей одних и тех же марок.

3. Результаты практического применения лиц Чернова

Для тестирования данного инструмента анализа многомерных данных были выбраны фотографии двадцати людей, которые были сгенерированы на сайте ThisPersonDoesNotExist [7]. Полученные графические объекты продемонстрированы на рис. 5.

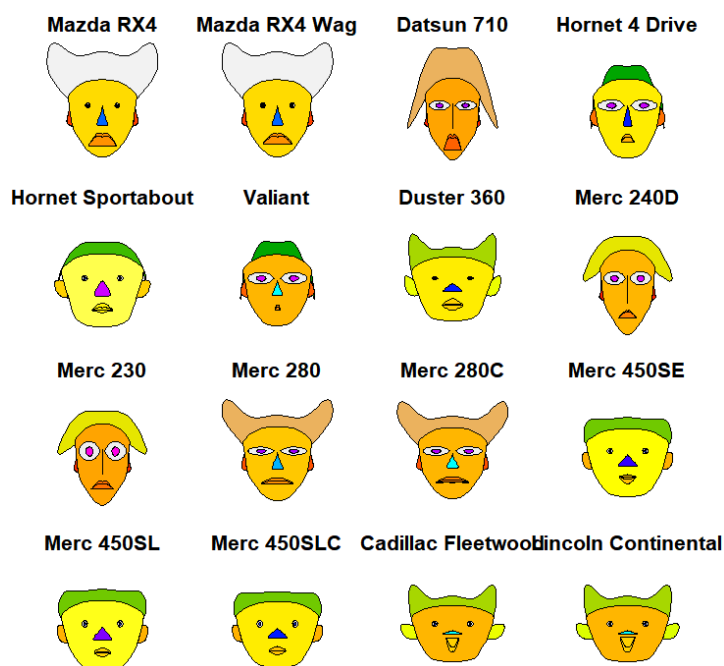


Рис. 4. Пример «лиц Чернова» для классификации моделей автомобилей



Рис. 5. Фотографии, сгенерированные для анализа с помощью «лиц Чернова»

В результате полученные изображения были обработаны и преобразованы в датасеты. В них присутствуют длины отдельных элементов лица, коэффициенты кривизны, значения смещений. На рис. 6 представлена часть таблицы данных.

face height	face width	face structure	height	mouth width	mouth smile
0.232	0.165	0.95	0.3	0.929	0.27
0.426	0.147	0.78	0.9	0.879	0.51
0.254	0.402	0.22	0.2	0.918	0.35
0.781	0.985	0.26	0.1	0.909	0.81
0.054	0.731	0.09	0.5	0.635	0.18
0.346	0.622	0.51	0.4	0.817	0.68
0.098	0.222	0.28	0.5	0.575	0.63
0.309	0.659	0.57	0.8	0.674	0.87
0.773	0.327	0.83	0.7	0.973	0.14
0.368	0.101	0.21	0.5	0.540	0.33
0.235	0.257	0.17	0.4	0.924	0.41
0.161	0.458	0.36	0.1	0.899	0.12

Рис. 6. Фрагмент датасета для генерации «лиц Чернова»

В результате выполнения метода для построения пиктографиков были получены результаты, продемонстрированные на рис. 7.

Учитывая, что люди, изображённые на сгенерированных фотографиях, не похожи между собой, практически одинаковых «лиц» не было получено. Однако между некоторыми прослеживаются общие черты.

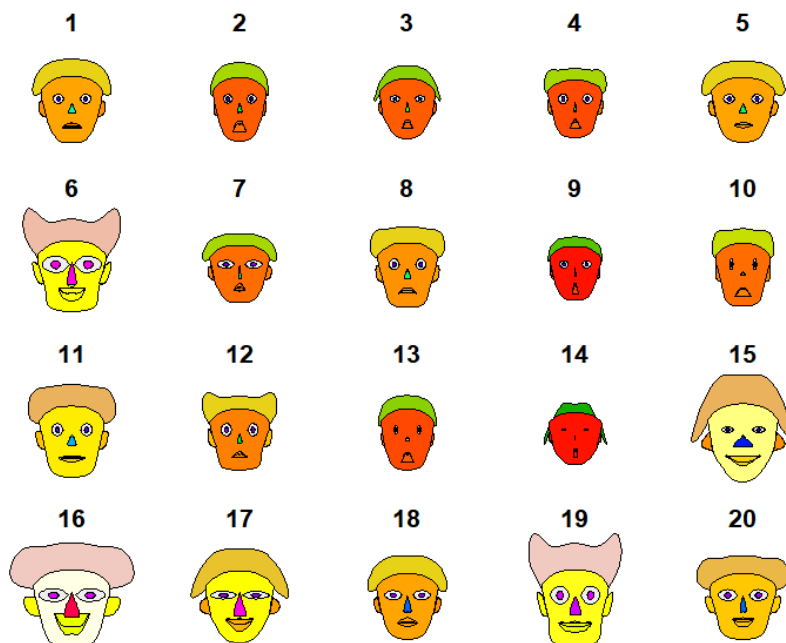


Рис. 7. «Лица Чернова», полученные на основе фотографий реальных людей

Заключение

На данном этапе работы в рамках исследования использования пиктографиков Чернова в качестве инструмента анализа многомерных данных выполнено изучение теоретического материала, выбран язык программирования для реализации и анализа данного инструмента. Среди достоинств рассматриваемого метода можно выделить возможность сформировать представление об объекте исследования в полном объёме, в отличие от графиков, где данные разбиваются по показателям, сравнение нескольких объектов для выделения отличительных и схожих признаков.

В результате были выполнены практические и тестовые задачи для погружения в данную область с помощью языка программирования R в среде разработки RStudio на базе стандартных библиотек.

В дальнейшем исследование данной темы продолжится на примере конкретной задачи анализа фотографий представителей факультета Прикладной Математики и Информатики для нахождения общих черт лица и генерации портрета классического студента ПММ.

Литература

1. *Зенкин, А. А.* Когнитивная компьютерная графика / А. А. Зенкин, под ред. Д. А. Поспелова. – Москва : Наука, 1991. – 192 с.
2. *Chernoff, H.* The Use of Faces to Represent Points in K-Dimensional Space Graphically / H. Chernoff // Journal of the American Statistical Association. – American Statistical Association Request Permissions, Jun., 1973. – Vol. 68, No. 342. – P. 361–368.
3. Официальный сайт RStudio. – Режим доступа: <https://www.r-studio.com/>. – (Дата обращения: 28.03.2022).
4. *Прокопьева, Р. О.* Визуализация многомерных данных с помощью пиктографиков «лиц Чернова» / Р. О. Прокопьева, И. А. Осадчая // Молодежь и современные информационные технологии : сб. тр. Междунар. научно-практической конф. студентов, аспирантов и молодых ученых (Томск, 13-16 ноября 2013 г.) : электронный ресурс. – Томск, 2013. – С. 501–503.

5. Наглядная статистика. Используем R! : учебник / А. Б. Шипунов [и др.]. – Москва : ДМК Пресс, 214. – 298 с.

6. Язык программирования R. – Режим доступа: <https://clck.ru/UVHAG/>. – (Дата обращения: 30.03.2022).

7. Сайт-генератор реалистичных лиц на основе генеративных нейронных сетей. – Режим доступа: <https://thispersondoesnotexist.com/>. – (Дата обращения: 30.03.2022).

Воронова Ольга Сергеевна — магистрант 1-го года обучения кафедры Математического обеспечения ЭВМ Воронежского государственного университета.

E-mail: olga.voronova99@yandex.ru

Болотова Светлана Юрьевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры МО ЭВМ Воронежского государственного университета.

E-mail: Bolotova.svetlana@gmail.com

КАЧЕСТВЕННОЕ ИССЛЕДОВАНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИКИ РАЗВИТИЯ МАЛОГО ПРЕДПРИЯТИЯ

С. Р. Дорофеева

Ижевский государственный технический университет имени М. Т. Калашникова

Введение

Математические модели в экономике можно разделить на классы по ряду признаков, относящихся к особенностям моделируемого объекта, цели моделирования и используемого инструментария. Процесс развития общества, и, как его составная часть, процесс развития производства, сопровождается экономическим ростом. В настоящее время проблема построения и исследования моделей роста поднялась на новый уровень в связи со становящейся все более актуальной концепцией устойчивого развития социально-экономического состояния общества. Предмет экономической динамики — исследование важнейших аспектов развития экономической системы. Для примера рассмотрим математическое моделирование и качественный анализ таких математических моделей в экономике, как модели динамики развития малого предприятия при различных вариантах государственной поддержке и кредитовании. Поскольку малое предпринимательство является важным фактором роста национальной экономики [1, 2], то рассмотренные модели получили широкое распространения.

1. Модели динамики развития малого предприятия

Развитие малого предприятия возможно как за счет внутренних источников (прибыль, амортизационные отчисления и т. д.), так и за счет внешних источников (государственные инвестиции, банковские кредиты) [3]. Как правило, внутренние источники развития не могут обеспечить серьезного финансового роста, позволяющего малому предприятию развиваться и конкурировать в сложных условиях становления рыночных отношений. В этой связи решающее значение в развитии малого предпринимательства приобретает государственная поддержка и различные механизмы кредитования [4, 5].

Рассмотрим модель динамики развития малого предприятия с участием внешних инвестиций как формы государственной поддержки.

Допущения модели:

1. Предприятие функционирует при неизменной технологии, что предполагает постоянство фондоотдачи.
2. Производственная деятельность описывается однофакторной линейной функцией.
3. Темп развития предприятия определяется динамикой основных производственных фондов.
4. Предприятие осуществляет налоговые выплаты по двум видам налогов: налог на объем выпуска и налог на прибыль.
5. Внутренняя инвестиционная стратегия предприятия определяется постоянной долей чистой прибыли, отчисляемой на реинвестирование.

Введем обозначения:

$Y(t)$ — выпуск продукции;

$K(t)$ — стоимость основных производственных фондов;

$I(t)$ — государственные инвестиции;

$D(t)$ — чистая прибыль;

$N(t)$ — налоговые отчисления.

Математическая модель динамики развития малого предприятия:

$$\frac{dK(t)}{dt} = \xi D(t) + I(t), \quad (1)$$

$$Y(t) = \alpha K(t), \quad (2)$$

$$D(t) = (1 - c)Y(t) - N(t), \quad (3)$$

$$N(t) = \eta_1 Y(t) + \eta_2 k(1 - \xi)D(t), \quad (4)$$

где α — фондоотдача; c — удельная себестоимость выпуска продукции; η_1, η_2 — ставки налогов на объем выпуска и на прибыль; ξ — доля чистой прибыли, отчисляемой на реинвестирование; k — доля реинвестирования, облагающаяся налогом.

Из (3) и (4) можно получить зависимость чистой прибыли от выпуска:

$$D(t) = \frac{(1 - c - \eta_1)Y(t)}{1 + \eta_2 k(1 - \xi)}. \quad (5)$$

Тогда динамику производственных фондов можно описать дифференциальным уравнением:

$$\frac{dK(t)}{dt} = aK(t) + I(t), \quad (6)$$

где

$$a = \frac{\xi(1 - c - \eta_1)}{1 + \eta_2 k(1 - \xi)} \alpha, \quad a > 0. \quad (7)$$

Рассмотрим следующие стратегии государственной поддержки:

- постоянная;
- линейная;
- экспоненциальная.

2. Модель господдержки при фиксированном объеме инвестирования

Рассмотрим модель государственной поддержки при фиксированном объеме инвестирования:

$$I(t) = I_0 = \text{const}. \quad (8)$$

Уравнение (6) приведет к задаче Коши:

$$\frac{dK(t)}{dt} = aK(t) + I_0, \quad K(0) = K_0. \quad (9)$$

Решение задачи (9):

$$K(t) = e^{at} \left(K_0 + \frac{I_0}{a} \right) - \frac{I_0}{a}. \quad (10)$$

Исследуем решение (10) задачи (9) на устойчивость. Изменим начальные условия: $\tilde{K}(0) = \tilde{K}_0$ ($|K_0 - \tilde{K}_0| < \delta$ при $\delta > 0$), тогда получим решение

$$\tilde{K}(t) = e^{at} \left(\tilde{K}_0 + \frac{I_0}{a} \right) - \frac{I_0}{a}.$$

Оценим разность:

$$\left| K(t) - \tilde{K}(t) \right| = \left| e^{at} \left(K_0 + \frac{I_0}{a} \right) - \frac{I_0}{a} - e^{at} \left(\tilde{K}_0 + \frac{I_0}{a} \right) + \frac{I_0}{a} \right| = |K_0 - \tilde{K}_0| e^{at}.$$

Следовательно, при $t \rightarrow \infty$ решение не устойчиво по Ляпунову. Поэтому данная модель не может описывать устойчивый рост малого предприятия в долгосрочном периоде.

Исследуем модель на устойчивость в промежутке времени $t \in [0, T]$. Для этого воспользуемся теоремой Коши о непрерывной зависимости решения от начальных условий [6]. Проверим выполнение условий теоремы:

1) поскольку функция $f(t, K) = aK(t) + I_0$ является линейной, то она непрерывна в области $\Omega = \{0 \leq t \leq T, K_0 \leq K \leq K(T)\}$;

2) производная функции $f(t, K)$ ограничена: $|f'_K| = |(aK(t) + I_0)'| = |a| \leq N$ в области $\Omega = \{0 \leq t \leq T, K_0 \leq K \leq K(T)\}$;

3) если $|K_0 - \tilde{K}_0| < \delta$, то $|K(t) - \tilde{K}(t)| = |K_0 - \tilde{K}_0| e^{at} < \varepsilon = \delta e^{aT}$ выполняется для любого t , такого что $|t_0 - t| < T$, $T < T_0 = \min \left\{ a, \frac{1}{N}, \frac{b}{M} \right\}$, где $M = \max_{(t,y) \in D} |f(t, y)|$.

Поскольку выполняются условия теоремы Коши, то можно сказать, что модель, задаваемая уравнением (9), устойчива на промежутке $t \in [0, T]$.

3. Модель господдержки при линейно возрастающем объеме инвестирования

Рассмотрим модель государственной поддержки при линейно возрастающем объеме инвестирования с постоянным темпом $\gamma > 0$ [7]:

$$I(t) = \gamma t. \quad (11)$$

Задача Коши (9) запишется следующим образом:

$$\frac{dK(t)}{dt} = aK(t) + \gamma t, \quad K(0) = K_0. \quad (12)$$

Дифференциальное уравнение из задачи (12) представляет собой линейное неоднородное дифференциальное уравнение с постоянными коэффициентами [8]. Решение данного уравнения, полученное с помощью подстановки Бернулли:

$$K(t) = e^{at} \left(-\frac{\gamma t e^{-at}}{a} - \frac{\gamma e^{-at}}{a^2} + C \right) = -\frac{\gamma}{a^2} (at + 1) + C e^{at}.$$

С учетом начального условия решение задачи Коши для случая линейно возрастающей финансовой поддержки со стороны государства имеет вид:

$$K(t) = \left(K_0 + \frac{\gamma}{a^2} \right) e^{at} - \frac{\gamma}{a^2} (at + 1). \quad (13)$$

Исследуем решение (13) задачи Коши (12) на устойчивость:

$$\tilde{K}(0) = \tilde{K}_0 \quad (|K_0 - \tilde{K}_0| < \delta \text{ при } \delta > 0),$$

$$|K(t) - \tilde{K}(t)| = \left| \left(K_0 + \frac{\gamma}{a^2} \right) e^{at} - \frac{\gamma}{a^2} (at + 1) - \left(\tilde{K}_0 + \frac{\gamma}{a^2} \right) e^{at} + \frac{\gamma}{a^2} (at + 1) \right| = |K_0 - \tilde{K}_0| e^{at}.$$

Следовательно, при $t \rightarrow \infty$ модель (12) не устойчива по Ляпунову.

Исследование на устойчивость в промежутке времени $t \in [0, T]$ по теореме Коши аналогично предыдущему случаю. Следовательно, модель, задаваемая уравнением (12), устойчива на промежутке $t \in [0, T]$.

4. Модель господдержки при экспоненциально возрастающем объеме инвестирования

Рассмотрим модель государственной поддержки при экспоненциально возрастающем объеме инвестирования со средним темпом $\beta > 0$ и с минимальным уровнем гарантированной государственной поддержки $I(0) = I_0$ [9]:

$$I(t) = I_0 e^{\beta t}. \quad (14)$$

Получим задачу Коши:

$$\frac{dK(t)}{dt} = aK(t) + I_0 e^{\beta t}, \quad K(0) = K_0. \quad (15)$$

Решение задачи Коши (15):

$$K(t) = K_0 e^{at} + \frac{I_0}{\beta - a} (e^{\beta t} - e^{at}). \quad (16)$$

Исследуем решение (16) на устойчивость:

$$\left| K(t) - \tilde{K}(t) \right| = \left| K_0 e^{at} + \frac{I_0}{\beta - a} (e^{\beta t} - e^{at}) - K_0 e^{at} - \frac{I_0}{\beta - a} (e^{\beta t} - e^{at}) \right| = |K_0 - \tilde{K}_0| e^{at}.$$

При $t \rightarrow \infty$ модель (15) не устойчива по Ляпунову. Исследование на устойчивость в промежутке времени $t \in [0, T]$ по теореме Коши аналогично первому случаю. Следовательно, модель, задаваемая уравнением (15), устойчива на промежутке $t \in [0, T]$.

5. Модель при условии одновременного кредитования

В случае отсутствия государственной поддержки ($I(t) = 0$) целесообразно рассмотреть ситуацию кредитования малого предприятия. Расходы по кредиту сказываются на показателях прибыли и себестоимости предприятия.

Пусть в момент времени $t = 0$ предоставляется одновременный кредит на период T в размере $B(0) = B_0$.

Тогда начальная стоимость основных фондов составит:

$$K(0) = K_0 + B_0. \quad (17)$$

Рассмотрим непрерывное начисление процентов по кредиту (r — ставка по кредиту), тогда размер долгового обязательства в момент времени t определяется по формуле:

$$B(t) = B_0 e^{rt}. \quad (18)$$

Поскольку осуществляется равномерное погашение долга с учетом начисления процентов, то расходы по кредиту определяются по формуле:

$$R(t) = \frac{B_0 e^{rT}}{T}. \quad (19)$$

Расходы по кредиту складывается из части основного долга (S) в момент времени t и процентов (s), выплачиваемых в этом периоде:

$$R(t) = \frac{B_0}{T} + \frac{B_0 (e^{rT} - 1)}{T} = S + s, \quad (20)$$

где $S = \frac{B_0}{T}$; $s = \frac{B_0 (e^{rT} - 1)}{T}$.

В случае внутренней инвестиционной стратегии величина S уменьшает чистую прибыль малого предприятия:

$$\frac{dK(t)}{dt} = \xi (D(t) - S). \quad (21)$$

Величина s увеличивает удельную себестоимость на величину $\frac{s}{Y(t)}$, тогда чистая прибыль (3) будет задаваться следующим выражением:

$$D(t) = \left(1 - c - \frac{s}{Y(t)} \right) Y(t) - N(t) = (1 - c)Y(t) - s - N(t). \quad (22)$$

Тогда с учетом уравнение (4) получим:

$$D(t) = (1-c)Y(t) - s - \eta_1 Y(t) - \eta_2 k(1-\xi)D(t), \quad (23)$$

откуда

$$D(t) = \frac{(1-c-\eta_1)Y(t) - s}{1 + \eta_2 k(1-\xi)}. \quad (24)$$

Динамика основных производственных фондов будет задаваться уравнением:

$$\frac{dK(t)}{dt} = \frac{\xi(1-c-\eta_1)}{1 + \eta_2 k(1-\xi)} \alpha K(t) - \xi \left(\frac{s}{1 + \eta_2 k(1-\xi)} + S \right) \quad (25)$$

или

$$\frac{dK(t)}{dt} = aK(t) - \xi(bs + S), \quad (26)$$

где $b = \left(\frac{1}{1 + \eta_2 k(1-\xi)} \right)$.

Если принять $I_0 = -\xi(bs + S)$, то решение дифференциального уравнения (26) будет аналогично решению уравнения (9) с начальными условиями (17):

Тогда решение дифференциального уравнения (9) имеет вид:

$$K(t) = e^{at} \left(K_0 + B_0 - \frac{\xi(bs + S)}{a} \right) + \frac{\xi(bs + S)}{a}. \quad (27)$$

Исследование на устойчивость решения (27) дифференциального уравнения (26) аналогично исследованию решения (10). Таким образом, решение (27) устойчиво на промежутке $t \in [0, T]$.

Исследование полученного соотношения свидетельствует о значительном влиянии константы a на темп роста системы. В свою очередь сама константа зависит от внутреннего экономического механизма рассматриваемого предприятия.

Важным вопросом является исследование доступности кредита для малого предприятия [10]. Возможность роста малого предприятия обеспечивается двумя условиями:

1. Необходимое условие: размер процентов, выплачиваемых в момент времени t , не должен превышать общей прибыли:

$$D(t) + N(t) = (1-c)Y(t) - s > 0. \quad (28)$$

2. Достаточное условие: размер чистой прибыли должен превышать долговые обязательства:

$$\frac{dK(t)}{dt} = \xi(D(t) - S) > 0 \text{ или } D(t) - S > 0. \quad (29)$$

Отношение долгового обязательства S к чистой прибыли $D(t)$ называется величиной доступности кредита и обозначается индикатором $\mu(t)$:

$$\mu(t) = \frac{S}{D(t)}. \quad (30)$$

Кредит в момент времени t является доступным при $\mu(t) \leq 1$, при $\mu(t) > 1$ — недоступным. Для малого предприятия выражение (29) определяет доступность кредитования:

$$\mu(t) = \frac{B_0/T}{\frac{(1-c-\eta_1)\alpha K(t) - s}{1 + \eta_2 k(1-\xi)}}. \quad (31)$$

Поскольку в знаменателе выражения (31) находится величина пропорциональная $K(t)$, то при достаточно быстром росте основных фондов малого предприятия обеспечивается доступность кредита $\mu < 1$.

На рис. 1 представлена динамика выпуска продукции условного малого предприятия за период в 10 лет при различных способах государственной поддержки и кредитования.

Из рис. 1 видно, что наиболее успешное развитие наблюдается у предприятия в случае экспоненциальной стратегии государственной поддержки.

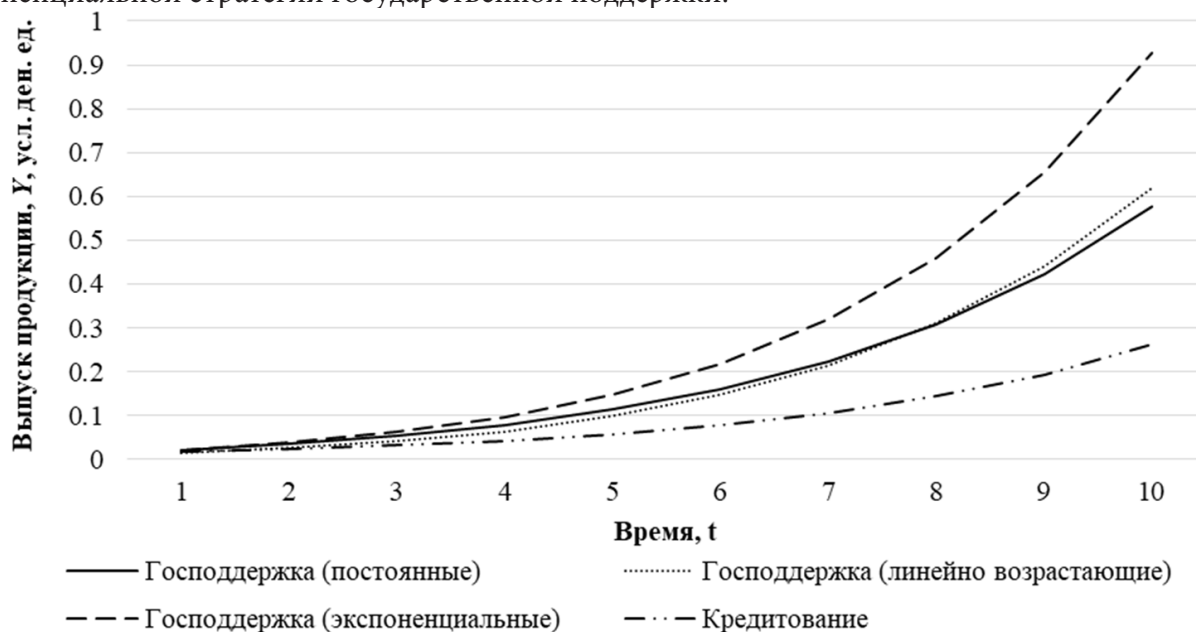


Рис. 1. Динамика выпуска продукции условного малого предприятия

Заключение

На микроуровне рассмотрены модели динамики развития малого предприятия при различных стратегиях государственной поддержки и механизмах кредитования. Данные модели развития малого предприятия неустойчивы в долгосрочном периоде, но могут адекватно описывать динамику развития предприятия на ограниченном промежутке времени.

Литература

1. Касаткина, Е. В. Математическая модель оптимального развития экономики предприятия / Е. В. Касаткина, Д. Д. Вавилова // Статистика и Экономика. – 2020. – Т. 17, № 2. – С. 72–81.
2. Кетова, К. В. Системный анализ деятельности предприятия и оптимальное управление его производственными факторами / К. В. Кетова, Е. В. Касаткина, Д. Д. Вавилова // Актуальные вопросы менеджмента современной организации : сб. тр. Междунар. студ. науч.-практ. конф. (Воронеж, 12–15 сентября 2016 г.) : электронный ресурс. – Ижевск, 2015. – С. 90–96.
3. Вавилова, Д. Д. Информационно-аналитическая система прогнозирования обобщающих показателей социально-экономического развития региона / Д. Д. Вавилова, Е. В. Касаткина // Проблемы управления. – 2015. – № 4. – С. 25–34.
4. Егорова, Н. Е. Применение дифференциальных уравнений для анализа динамики развития малых предприятий, использующих кредитно-инвестиционные ресурсы / Н. Е. Егорова, С. Р. Хачатрян // Экономика и математические методы. – 2006. – Т. 42, № 1. – С. 50–68.
5. Протасов, Д. Н. Развитие моделей кредитно-инвестиционных ресурсов промышленного предприятия : дис. ... канд. экон. наук. – Тамбов, 2009. – 139 с.
6. Кетова, К. В. Качественный анализ математических моделей динамических систем : учеб. пособие / К. В. Кетова, Е. В. Касаткина. – Ижевск : Изд-во ИжГТУ имени М. Т. Калашникова, 2013. – 116 с.

7. *Егорова, Н. Е.* Дифференциальный анализ развития малых предприятий, использующих кредитно-инвестиционный ресурс / Н. Е. Егорова, С. Р. Хачатрян, М. А. Маренный // Аудит и финансовый анализ. – 2000. – № 4. – С. 11–37.

8. *Пучков, В. Ф.* Разработка и применение математических моделей для решения задач управления экономическими системами : монография / В. Ф. Пучков, Г. В. Грацинская – Москва: Изд-во: Издательский дом «БИБЛИО-ГЛОБУС», 2015. – 416 с.

9. *Герасимов, Б. И.* Дифференциальные динамические модели : учеб. пособие / Б. И. Герасимов, Н. П. Пучков, Д. Н. Протасов. – Тамбов: Изд-во ГОУ ВПО ТГТУ, 2010. – 80 с.

10. *Лановая, О. Г.* Финансовый механизм кредитования малого бизнеса : дис. ... канд. экон. наук. – Москва, 2008.

Дорофеева Светлана Рудольфовна — магистрант 1-го года обучения кафедры «Автоматизированные системы обработки информации и управления» Ижевского государственного технического университета имени М. Т. Калашникова. E-mail: svetamien2018@gmail.com.

Касаткина Екатерина Васильевна (научный руководитель) — канд. физ.-мат. наук, доц., доцент кафедры «Автоматизированные системы обработки информации и управления» Ижевского государственного технического университета имени М. Т. Калашникова.

СТАТИСТИЧЕСКИЙ АНАЛИЗ ДИНАМИКИ ДЕМОГРАФИЧЕСКИХ ПОПУЛЯЦИЙ

А. Д. Дроботенко

Воронежский государственный университет

Введение

В настоящее время задача изучения и моделирования демографических процессов является достаточно актуальным направлением в демографическом прогнозировании. *Демографический прогноз* — это обоснованное с научной точки зрения предсказание будущей ситуации в демографии и основных показателей движения населения: численности, характеристик рождаемости и смертности, передвижки, структуры пола и возраста. Необходимость демографического прогнозирования тесно связана с задачами планирования важных социально-экономических процессов. Демографическая ситуация оказывает влияние на работу как отдельных направлений развития общества — производства и потребления товаров и услуг, работу государственных органов и коммерческих организаций, предприятий здравоохранения и образования; так и на системы в целом — политико-экономическую, научную, международную.

В рамках данной статьи будет рассмотрено применение математической модели в виде уравнений состояния для анализа демографического состава популяции. Также будет использован фильтр Калмана — Бьюси для работы в условиях неточных и неполных наблюдений — эффективный последовательный рекурсивный фильтр, который может быть применен для оценки вектора состояния заданной динамической системы при недостаточных и зашумленных показаниях, ведь оценка будет скорректирована в результате анализа каждой новой выборки на следующем шаге измерения.

1. Постановка задачи

Основная задача — это исследование применения математических моделей в виде уравнений состояния в совокупности с оптимальным фильтром Калмана — Бьюси для оценки демографического состава популяции в условиях неопределенности. ФКБ нашел активное применение в инженерных областях, однако его использованию в демографических системах препятствует ряд особенностей:

- неизвестность большого количества значимых факторов (экологических, социально-экономических и других);
- невозможность проведения достаточного количества экспериментов над исследуемой популяцией для более точного моделирования;
- большее количество ошибок наблюдения по сравнению с техническими системами и т. д.

Таким образом, в рамках данной работы будет проведен анализ алгоритма работы ФКБ, а также его возможности применения в демографии.

2. Решение задачи

2.1. Представление математической модели популяции

Для решения текущей задачи необходимо разделить исходную тестовую популяцию на группы по полу и возрасту, задав длительность временного интервала. Тогда математическую

определенные возрастные группы, достаточно положить соответствующие блоки матрицы C равными 1, остальные обнулить.

$v[k]$ — вектор выходных шумов наблюдения, который представляет собой последовательность случайных векторов со средним значением $Mv[k] = \bar{v}$ и ковариационной матрицей V_v . Он введен для отображения ошибок в определении параметров наблюдения.

С одной стороны, данная модель предоставляет возможность учета основных демографических факторов — рождаемости (α_i, γ_i) ; смертности (β_i, δ_i) ; миграции (U^F, U^M) . Однако, смоделированная демографическая система является нестационарной и стохастичной, что значительно усложняет решение задачи в условиях неопределенности.

Так, например, непосредственное решение системы уравнений состояния является корректным для прогноза состояния на последующий момент времени лишь в некоторой части случаев. Ошибки определения начального вектора состояния $X[0]$, матриц A, U^0 приводят к значительному уменьшению точности оценки и неограниченному возрастанию ошибок прогноза.

С целью устранения вышеописанных проблем необходимо выполнить замену $X[k]$ на $Y[k]$ как текущее наблюдение вектора $X[k]$ в предположении, что вектор $X[k]$ наблюдается полностью:

$$\widehat{X}[k+1] = A[k]Y[k] + U^0[k].$$

При такой модификации ошибки в начальном состоянии и в матрицах уже не приведут к значимым последствиям, так как на следующем шаге прогноз будет скорректирован. Однако возникает следующая проблема: потенциальная возможность наблюдения вектора $X[k]$ с ошибками при отсутствии фильтрации наблюдений.

Таким образом, задача состоит в нахождении статистически оптимальных оценок вектора состояния $\widehat{X}[k]$ (численности возрастных групп) по данным вектора наблюдений $Y[k]$.

2.2. Алгоритм фильтра Калмана — Бьюси

Алгоритм фильтрации Калмана — Бьюси (ФКБ) позволяет статистически оптимальным образом находить оценки состояния системы, описываемые вектором состояния популяции $X[k]$. Фильтр обеспечивает минимум среднеквадратической ошибки оценивания состояния системы. Главная идея ФКБ состоит в том, чтобы получить наилучшее приближение оценок неизвестных значений вектора состояния $X[k]$ по известным данным наблюдения $Y[k]$, а прогнозирование выдает оценки на один шаг вперед. Алгоритм состоит из двух повторяющихся этапов предсказания и корректировки.

1. На первом шаге выполняется прогноз состояния системы на основе заданных параметров в следующий момент времени, учитывая неточность измерения:

$$\widehat{X}^- [k+1] = A\widehat{X}[k] + \widehat{U}[k].$$

Знак «-» в верхнем индексе означает, что это предсказанное промежуточное значение, индекс k обозначает состояние системы в текущий момент времени, $k+1$ — в следующий. $\widehat{U}[k]$ — значения баланса миграции населения.

2. Вычисление матрицы коэффициентов усиления Калмана:

$$\Gamma[k] = AP[k]C\{C^T P[k]C + V_v\}^{-1},$$

$\Gamma[k]$ — матрица коэффициентов усиления обратной связи фильтра, $P[k]$ — заданная ковариационная матрица ошибки оценки состояния $X[k]$, C — матрица, определяющая отношение между измерениями и состоянием системы, V_v — ковариационная матрица выходных шумов.

3. Обновление оценки, учитывая поступление новых реальных данных и неточность их наблюдения:

$$Y[k] = CX[k] + v[k],$$

$$\widehat{X}[k+1] = \widehat{X}^- [k+1] + \Gamma[k] \{Y[k] - C\widehat{X}[k] - \bar{v}\},$$

$$\widehat{X}[0] = MX[0].$$

4. Обновление матрицы ковариации ошибки вектора состояния:

$$P[k+1] = AP[k]A^T + V_U - \Gamma[k]C^T P[k]A^T,$$

$$P[0] = P_0,$$

P_0 — заданная ковариационная матрица ошибки оценки начального состояния $X[0]$,

V_U — ковариационная матрица входных шумов.

3. Анализ результатов

Для анализа результатов использования модели в форме уравнений состояния и ФКБ в рамках задачи оценивания структурного состава популяции было разработано веб-приложение, которое функционирует в режиме моделирования различных тестовых данных. На первом шаге необходимо задать параметры для моделирования будущей популяции. На рис. 1 представлен интерфейс данного функционала. Система предоставляет возможность определения следующих атрибутов:

- количество наблюдаемых групп;
- максимальный коэффициент рождаемости;
- количество точных наблюдений;
- среднее количество особей в группе.

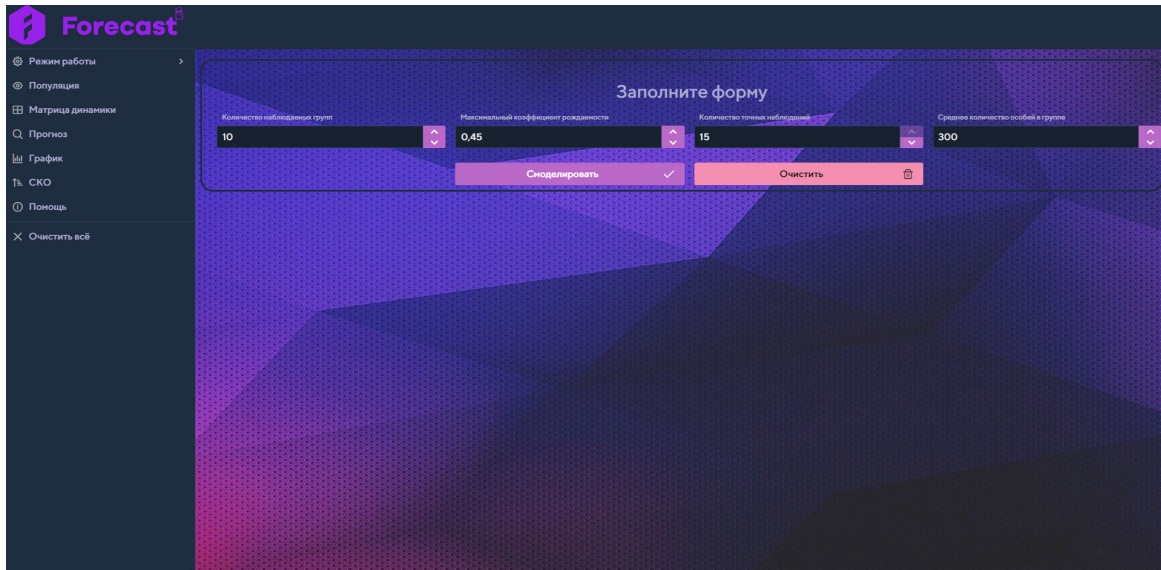


Рис. 1. Интерфейс моделирования популяции

Далее на вкладке «Популяция», представленной на рисунке 2, отображается матрица популяции в соответствии с заданными пользователем параметрами-атрибутами. Количество столбцов в таблице равно числу векторов состояний, а количество строк — числу возрастных групп, на которые разбивается исходная популяция.

Матрица динамики возрастного состава популяций приведена на рис. 3. Она находится на одноименной вкладке и представляет собой коэффициенты рождаемости и смертности, расположенные в первой строке и ниже главной диагонали соответственно.

На вкладке «Прогноз» в форме слева пользователь может задать параметры СКО коэффициентов рождаемости, смертности и шумов наблюдения. После нажатия на кнопку «Прогноз» в области справа система отображает вектора состояния, соответствующие действительному

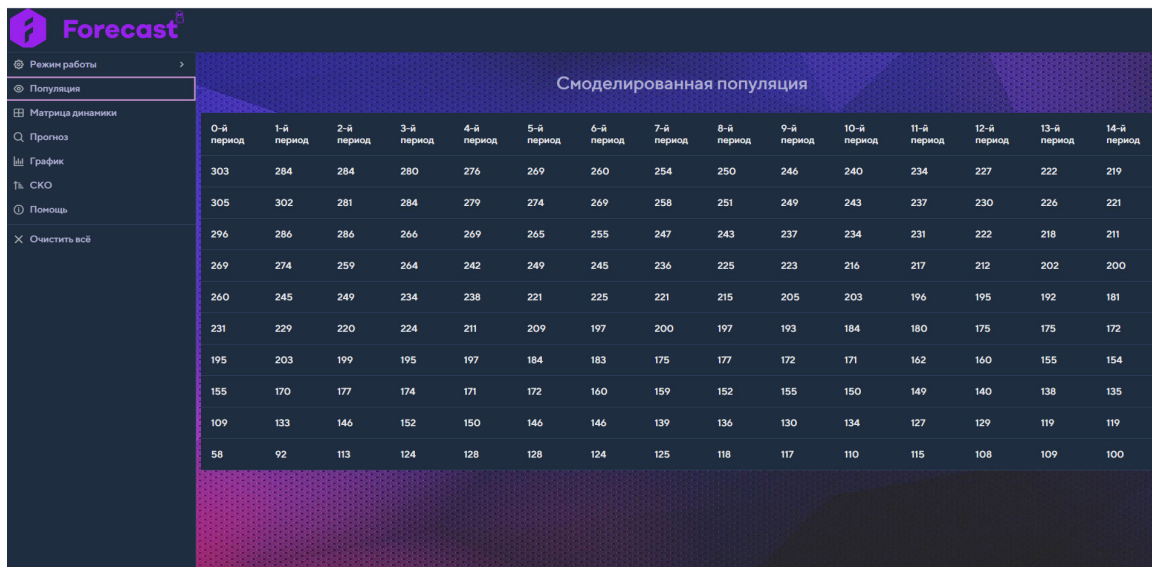


Рис. 2. Матрица смоделированной популяции

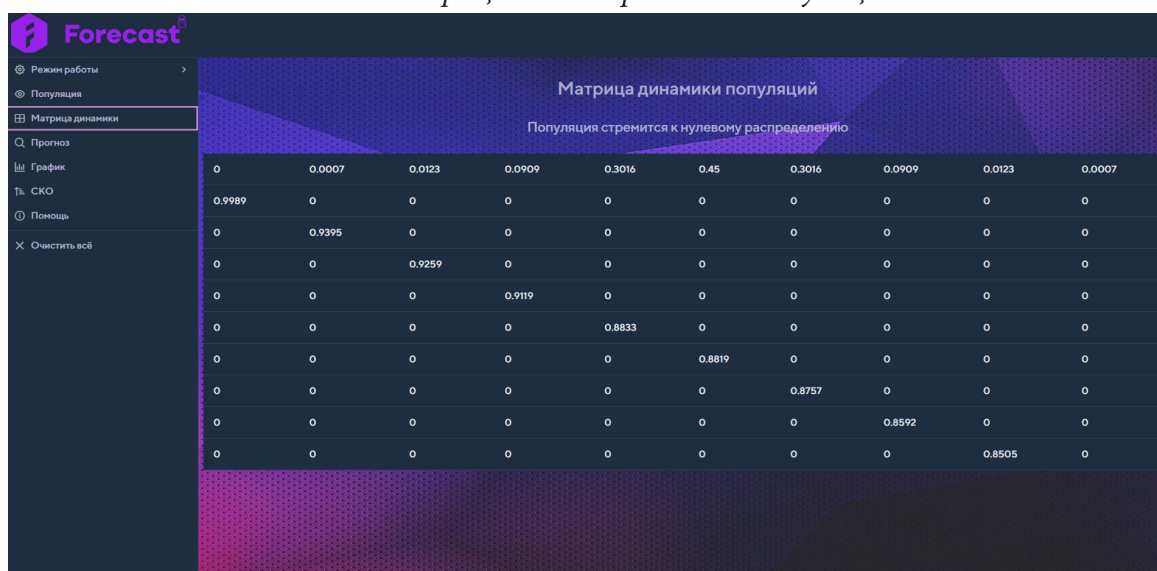


Рис. 3. Матрица динамики популяции

состоянию популяции, упрощенному прогнозу без фильтра и прогнозу с применением ФКБ. На данном этапе можно увидеть, что оценка с использованием ФКБ является наиболее приближенной к реальному состоянию популяции, чем оценка без использования фильтра. На рис. 4 приведен пример оценки численности популяции на девятый период в разбиении по возрастным группам.

Для удобства пользователя данные оценочные значения состояния демографической популяции также представлены в виде графика на рис. 5.

Заключение

В рамках данной работы была построена модель оценки структурного состава популяции в форме уравнений состояния с использованием ФКБ. Также было разработано веб-приложение с целью исследования работы фильтра Калмана — Бьюси и возможности его использования в демографической области. Оно показало перспективность применения построенной модели для изучения социально-демографических процессов.

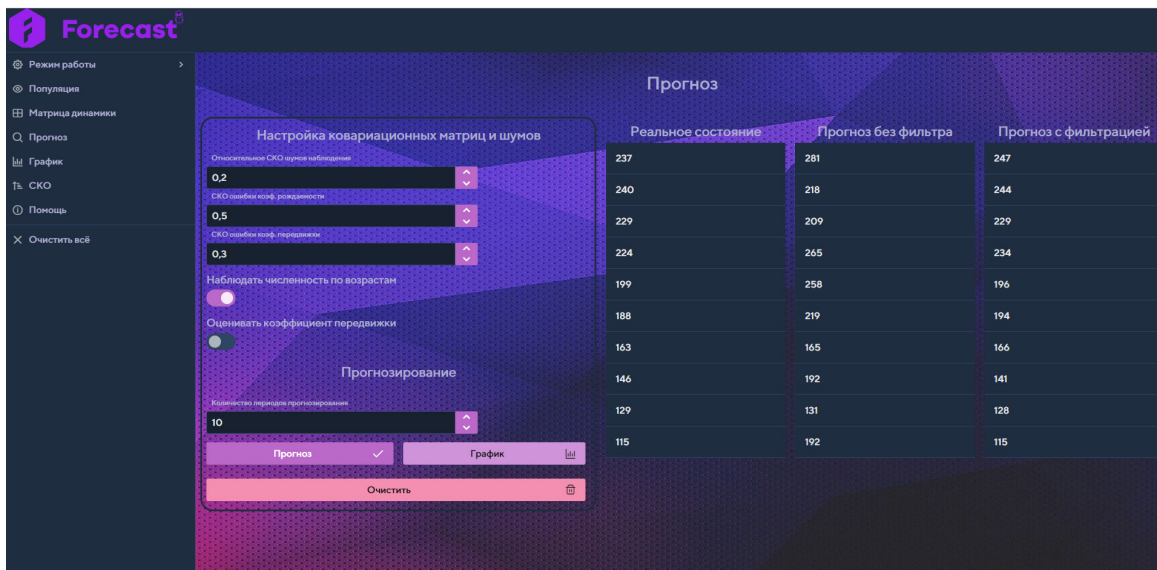


Рис. 4. Прогноз популяции

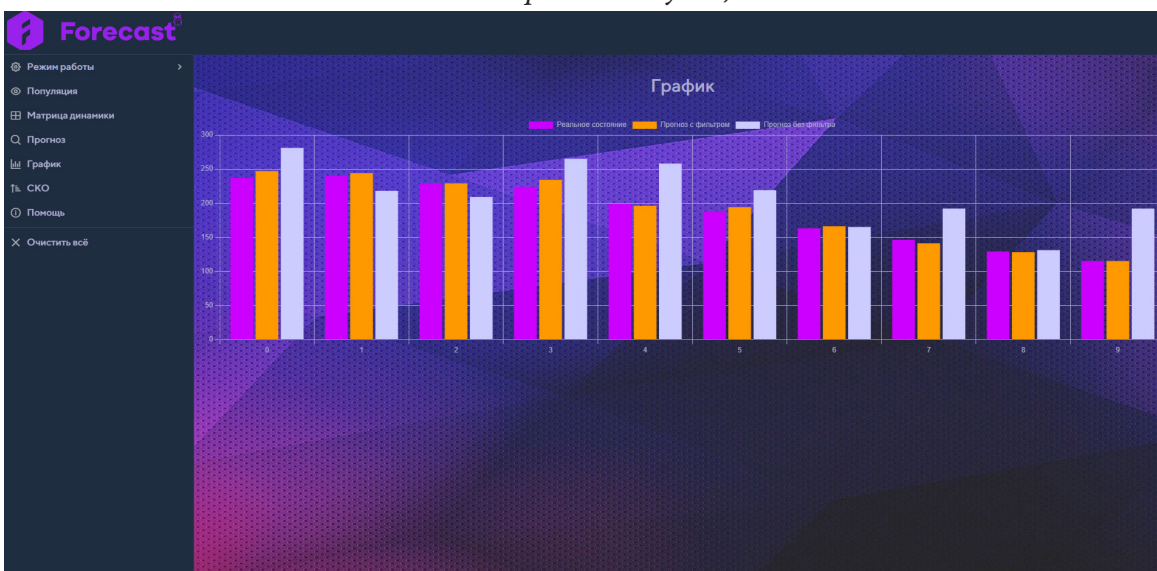


Рис. 5. График прогноза популяции

Литература

1. Рудалев, В. Г. Фильтрация и прогнозирование данных демографических наблюдений / В. Г. Рудалев, А. И. Кремер // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – Воронеж, 2014. – № 1. – С. 109–113.

2. Калман, Р. Е. Идентификация систем с шумами / Р. Е. Калман // Успехи мат. наук. – 1985. – Т. 40, вып. 4(244). – 1985. – С. 27–41.

3. Фильтр Калмана. – Режим доступа: <https://habr.com/ru/post/166693>. – (Дата обращения: 12.03.2022).

Дроботенко Александра Дмитриевна — магистрант 2-го курса кафедры ERP-систем и бизнес процессов Воронежского государственного университета.
E-mail: aleksandradrobotenko@mail.ru

Рудалев Валерий Геннадьевич — канд. физ.-мат. наук, доц., доцент кафедры ERP-систем и бизнес процессов Воронежского государственного университета. E-mail: rud_wl@mail.ru

МАТЕМАТИЧЕСКИЕ МЕТОДЫ ПОДДЕРЖКИ УПРАВЛЕНИЯ ФИНАНСАМИ ОБЪЕДИНЕНИЯ ЭКОНОМИЧЕСКИХ АГЕНТОВ ПРИ РЕАЛИЗАЦИИ СОВМЕСТНЫХ ПРОЕКТОВ

Е. И. Дыбов, Ю.В. Бондаренко

Воронежский государственный университет

Введение

В настоящее время проектная деятельность набирает все большую популярность в области экономики, бизнеса, строительства, IT — технологий, государственного управления, образования и мн. др. Многообразие форм реальных проектов порождает многообразие тех задач, решение которых необходимо для их эффективной реализации [1–3]. Одной из их разновидностей является класс проектов, реализуемых и одновременно финансируемых инициативной группой экономических агентов (индивидуумов, бизнес-единиц, организаций, предприятий). К числу таких проектов относятся, например, стартап проекты, научные разработки и т. п. Эффективное управление такими проектами требует решения целого ряда задач, среди которых важное место занимают вопросы эффективного управления финансами. Сложность решения поставленной задачи и ее актуальность требуют разработки инновационных механизмов решения, основанных на активном привлечении современных математических методов, моделей и информационных технологий.

Целью настоящей работы является разработка математических методов, моделей и программ, обеспечивающих поддержку эффективного управления финансами экономических агентов при реализации совместных проектов, включая формирование оптимального пакета проектов и согласованного распределения прибыли от его реализации.

Отличительная особенность предлагаемого в настоящей работе подхода состоит в том, что он не только обеспечивает согласование интересов агентов и устойчивость объединения, но и стимулирует экономических агентов к активному участию в проектах объединения.

Перейдем к описанию предлагаемого подхода.

1. Описание подхода

Будем рассматривать активную систему, включающую следующие элементы:

- центр принятия решений по управлению финансами (ЦПР);
- экономические агенты, число которых обозначим через n .

В качестве важного свойства агентов отметим их активность — способность формулировать собственные цели и определенная самостоятельность в принятии решений по их достижению. Центр принятия решений может иметь неформальный характер и формироваться из агентов — участников проекта.

Полагаем, что каждый агент A_i , где $i = 1, \dots, n$, располагает финансовыми средствами в объеме R^i , которые он готов инвестировать в реализацию проектов. При этом агент располагает набором собственных инициатив — проектов, которые он может предложить объединению.

Обладая активностью в принятии инвестиционных решений, каждый агент может:

- самостоятельно, собственными средствами, формировать и реализовывать собственные проекты;
- объединяясь с другими агентами системы, вкладывать финансовые средства в реализацию совместных проектов.

При этом отметим, что экономической целью каждого агента является получение наибольшей прибыли, что и делает выгодным его объединение с другими агентами.

Предлагаемый в настоящей работе *механизм управления финансовыми ресурсами экономических агентов при реализации совместных проектов* включает два укрупненных этапа:

Этап 1. Формирование оптимального пакета проектов, обеспечивающего получение максимальной совокупной прибыли объединения агентов.

Этап 2. Согласованное распределение прибыли между агентами, обеспечивающее компромисс интересов и стимулирование агентов к активному участию к инициализации и реализации проектов.

На первом этапе каждый агент A_i направляет управляющему центру свои предложения вариантов тех проектов, в реализации которых он заинтересован:

$$P^i = \{P_1^i, P_2^i, \dots, P_{Q_i}^i\},$$

где Q_i — количество проектов, P^i — множество проектов, $P_1^i, \dots, P_{Q_i}^i$ — перечень проектов агента A_i .

Полагаем, что каждый проект рассчитан на определенный промежуток времени (например, на один год). Для каждого проекта P_q^i ($i = 1, \dots, n$, $q = 1, \dots, Q_i$) приводится обоснование затрат финансов в объеме c_q^i , а также величины выгоды (прибыли) Π_{iq} , которую ожидается получить от реализации проекта q агента A_i . При этом собственных ресурсов агента может быть достаточно или недостаточно для самостоятельной реализации проектов множества P_i .

Таким образом, управляющий центр получает информацию о множестве проектов объединения $P = \bigcup_{i=1}^n P^i = \{P^1, P^2, \dots, P^n\} = \{P_1^1, P_2^1, \dots, P_{Q_1}^1, \dots, P_1^n, \dots, P_{Q_n}^n\}$, а также о доступных количествах финансового ресурса: $R = \sum_{i=1}^n R^i$.

Практическую реализацию первого этапа механизма (формирование оптимального пакета проектов объединения) предлагается осуществлять на основе решения оптимизационной задачи.

Введем в рассмотрение следующие бинарные переменные:

$$y_{iq} = \begin{cases} 1, & \text{если проект } P_q^i \text{ включается в пакет,} \\ 0, & \text{иначе.} \end{cases}$$

Математическая задача формирования оптимального пакета проектов системы имеет следующий вид:

$$f(y) = \sum_{i=1}^n \sum_{q=1}^{Q_i} \Pi_{iq} \cdot y_{iq} \rightarrow \max, \quad (1)$$

$$\begin{cases} \sum_{i=1}^n \sum_{q=1}^{Q_i} c_q^i \cdot y_{iq} \leq R, \\ y_{iq} \in \{0, 1\}, i = 1, \dots, n; q = 1, \dots, Q_i. \end{cases} \quad (2)$$

Задача (1)–(2) является задачей о ранце, в качестве решения которой может быть выбран метод ветвей и границ.

Решением задачи (1)–(2) является оптимальный вектор значений переменных $y^* = (y_{11}^*, \dots, y_{1Q_1}^*, \dots, y_{n1}^*, \dots, y_{nQ_n}^*)$. На основе полученного решения задачи формируется оптимальный пакет проектов объединения (системы) агентов $P^* = \{P_q^i \in P \mid y_{iq}^* = 1\}$, а также рассчитывается оптимальная прибыль объединения $f^* = f(y^*)$, которую планируется получить от реализации оптимального пакета проектов.

Вторым этапом механизма является построение правила и алгоритма распределения прибыли f^* между агентами объединения.

Обозначим через \tilde{f}_i размер прибыли, которую получает в результате распределения агент A_i ($i = 1, \dots, n$), $\tilde{f} = (\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_n)$ — искомый набор (вектор) распределения прибыли.

При этом очевидно, что $\tilde{f}_i \geq 0$ для всех $i = 1, \dots, n$ и $\sum_{i=1}^n \tilde{f}_i = f^*$.

Будем говорить, что распределение \tilde{f} обеспечивает компромисс интересов агентов системы, или является **согласованным**, если выполняется следующее **условие согласованности**: ни одному из объединений агентов не выгодно самостоятельно реализовывать собственные проекты, отделившись от остальных агентов системы.

Для формального описания условия согласованности рассмотрим множество индексов агентов системы: $I = \{1, 2, \dots, n\}$, а также всевозможные непустые подмножества этого множества (коалиции агентов), число которых $2^n - 1$. Подмножество, включающее агентов с номерами i_1, i_2, \dots, i_k обозначим как I_{i_1, i_2, \dots, i_k} . Пусть $f_{i_1, i_2, \dots, i_k}^*$ — прибыль, которую самостоятельно может получить коалиция агентов I_{i_1, i_2, \dots, i_k} . Ожидаемая величина прибыли может быть рассчитана на основе моделей, аналогичных (1)–(2), но решаемых для отдельного игрока или коалиции.

В рамках введенных обозначений **условие согласованности** формализовано можно представить в виде следующей системы неравенств:

$$\begin{cases} \tilde{f}_i \geq f_i^*, i = 1, \dots, n; \\ \tilde{f}_{i_1} + \tilde{f}_{i_2} + \dots + \tilde{f}_{i_k} \geq f_{i_1, i_2, \dots, i_k}^*, \forall I_{i_1, i_2, \dots, i_k} \subset I; \\ \tilde{f}_1 + \tilde{f}_2 + \dots + \tilde{f}_n = f^*. \end{cases} \quad (3)$$

Неравенства, выделенные в первую строку условий (3), в теории кооперативных игр носят название условия индивидуальной рациональности, а в последней строке — условия коллективной рациональности [4].

Система (3) может иметь множество решений, каждое из которых является согласованным распределением прибыли. Представляется логичным, что окончательное распределение прибыли должно обладать не только свойством согласованности, но и мотивировать агентов на активное участие и инициализацию проектов как настоящих, так и будущих. В основе формирования такого распределения прибыли предлагается использовать количественные показатели активности каждого агента в реализации проектов.

Алгоритм согласованного распределения прибыли экономических агентов

Шаг 1. ЦПР для каждого агента A_i формирует нижнюю $\underline{\Delta}_i$ и верхнюю $\overline{\Delta}_i$ границы стимулирующей надбавки Δ_i , где $\Delta_i \in [\underline{\Delta}_i, \overline{\Delta}_i]$.

Шаг 2. Эксперты ЦПР определяют α_i — количественный показатель активности агента A_i , $\alpha_i \geq 0$, $\sum_{i=1}^n \alpha_i = 1$.

Шаг 3. Решение задачи формирования оптимального вектора согласованного распределения прибыли агентов с учетом активности:

$$\sum_{i=1}^n \alpha_i \cdot \frac{\Delta_i - \underline{\Delta}_i}{\overline{\Delta}_i - \underline{\Delta}_i} \rightarrow \max, \quad (4)$$

$$\begin{cases} \tilde{f}_i \geq f_i^* + \Delta_i, i = 1, \dots, n; \\ \tilde{f}_{i_1} + \tilde{f}_{i_2} + \dots + \tilde{f}_{i_k} \geq f_{i_1 i_2 \dots i_k}^*, \quad \forall I_{i_1 i_2 \dots i_k} \subset I, \\ \tilde{f}_1 + \tilde{f}_2 + \dots + \tilde{f}_n = f^*, \\ \Delta_i \in [\underline{\Delta}_i, \overline{\Delta}_i]. \end{cases} \quad (5)$$

Поскольку задача (4)–(5) с переменными \tilde{f}_i, Δ_i (где $i = 1, \dots, n$) является задачей линейного программирования, то для ее решения может быть выбран симплекс-метод. Оптимальное решение задачи $\tilde{f}^* = (\tilde{f}_1^*, \tilde{f}_2^*, \dots, \tilde{f}_n^*)$ является согласованным распределением прибыли с учетом активности агентов.

2. Программная реализация

Для написания приложения был использован язык Java с использованием интегрированной среды разработки IntelliJ IDEA. Реализованный инструмент является десктопным приложением. Данный язык программирования имеет достаточно много преимуществ, что позволило ему приобрести высокую степень популярности среди программистов. К ним относят: независимость и универсальность; надежность и безопасность; объектно-ориентированный язык; автоматическое управление памятью; библиотеки и фреймворки; принцип обратной совместимости; простота синтаксиса; широкое распространение.

Диаграмма классов программы включает:

- Класс Controller, необходимый для реализации графического интерфейса;
- Connector — для подключения базы данных MySQL;
- SimplexMethods — для реализации симплекс-метода;
- UnionAgents — единичное сочетание агентов;
- Backpack — для реализации динамического рюкзака;
- Bank — для хранения всех возможных сочетаний агентов и их прибыли;
- Edit — для SQL-запросов;
- Project — для описания проекта агента;
- Agent — для описания агента;
- Application — главный класс.

Для реализации приложения использовались следующие библиотеки:

1. OpenJFX — библиотека для работы с JavaFX (графический интерфейс)
2. JUnit — библиотека для Unit тестирования
3. MySQL — библиотека для работы с БД MySQL
4. POI — библиотеки для работы с файлами для XML и XMLS
5. Commons — библиотека математических функций
6. Maven Plugins — библиотека плагинов
7. JetBrains — библиотека аннотаций

При запуске программы открывается окно, представленное на рис. 1.

Сначала нужно ввести всех агентов в формате «Имя агента, Бюджет» и нажать кнопку «Add» (рис. 2).

После заполнения всех агентов необходимо ввести их проекты в формате «Название проекта, Затраты на реализацию, Прибыль». Для этого сперва следует выбрать агента, к которому относится проект, затем ввести данные и нажать кнопку «Add» (рис. 3).

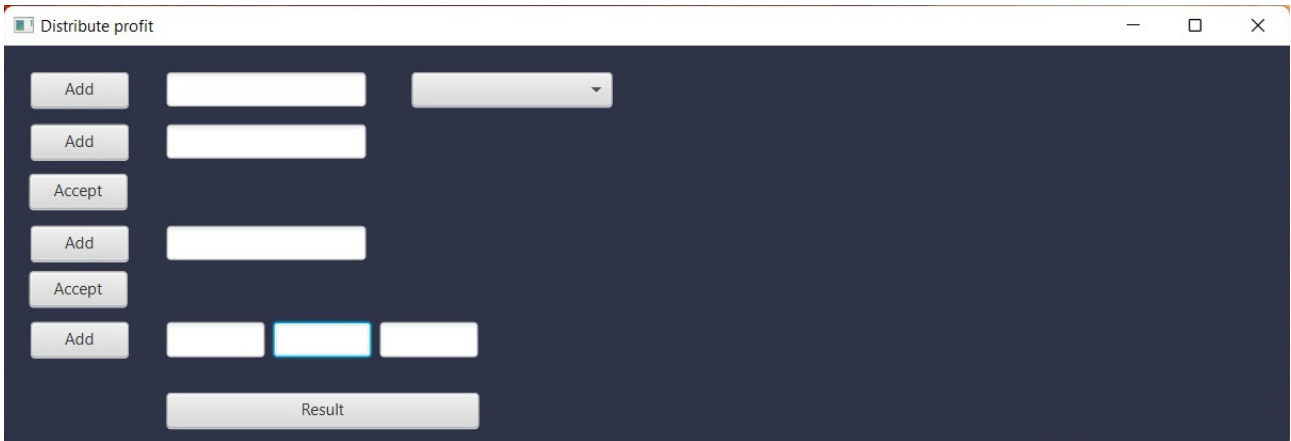


Рис. 1. Окно ввода данных

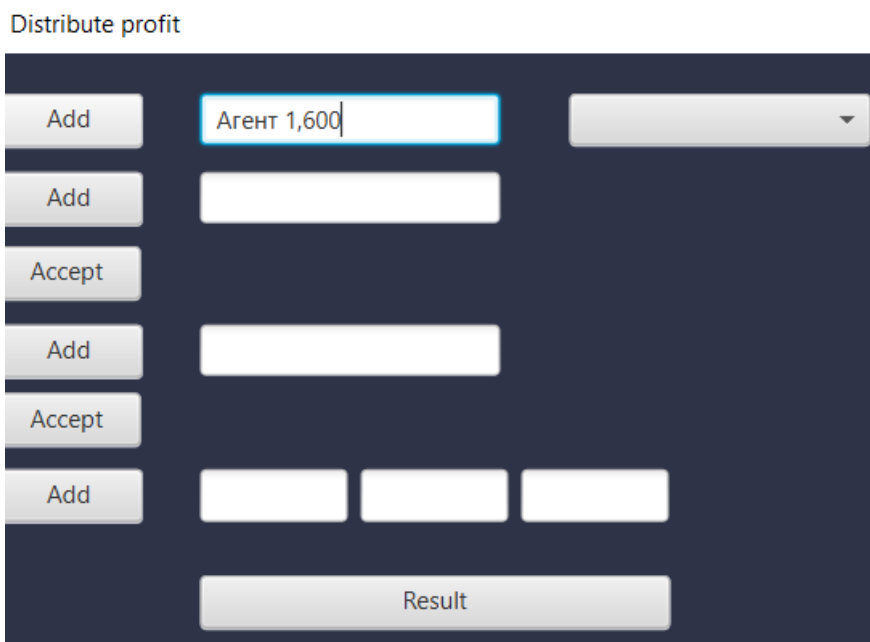


Рис. 2. Ввод данных агентов

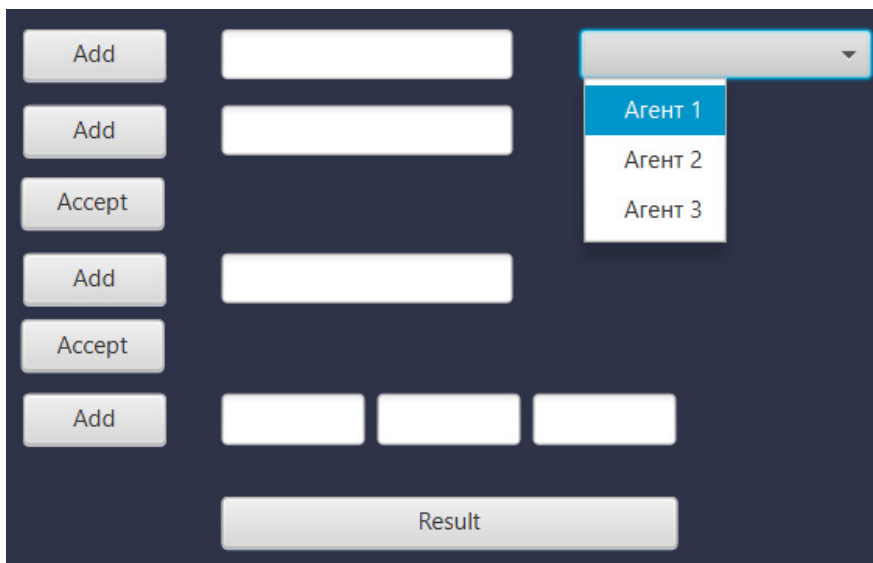


Рис. 3. Ввод данных проектов

После заполнения всех проектов необходимо нажать кнопку «Ассерпт», в результате чего откроется запись с результатами решения задачи формирования оптимального пакета проектов по всем сочетаниям агентов в формате «Наименование агента, Бюджет агента, Пакет проектов, Прибыль» (рис. 4).

Агенты: Агент 1 , Проекты: 11 14 , Оптимальная прибыль: 190.0
 Агенты: Агент 2 , Проекты: 21 23 , Оптимальная прибыль: 82.6
 Агенты: Агент 3 , Проекты: 33 34 35 , Оптимальная прибыль: 190.0
 Агенты: Агент 1 Агент 2 , Проекты: 11 13 14 21 23 , Оптимальная прибыль: 290.6
 Агенты: Агент 1 Агент 3 , Проекты: 11 12 14 34 35 , Оптимальная прибыль: 434.0
 Агенты: Агент 2 Агент 3 , Проекты: 21 22 34 35 , Оптимальная прибыль: 312.6
 Агенты: Агент 1 Агент 2 Агент 3 , Проекты: 11 12 13 14 21 22 23 24 34 , Оптимальная прибыль: 545.6

Рис. 4. Расчет оптимальных значений прибыли коалиций

Далее вводим показатели активности агентов (рис. 5).

Рис. 5. Ввод показателей активности экономических агентов

Окно ввода интервалов изменения стимулирующей надбавки представлено на рис. 6.

Рис. 6. Окно ввода интервалов изменения стимулирующей надбавки

После нажатия на кнопку «Result» выводится решение задачи — согласованное распределение прибыли с учетом активности агентов (рис. 7).

Решение задачи: 233; 111,6; 201

Рис. 7. Окно вывода результатов

Заключение

Настоящее исследование посвящено разработке математической и программной поддержки управления финансами объединения экономических агентов при реализации совместных проектов. Предлагаемый подход включает два укрупненных этапа. На первом этапе решается

задача формирования оптимального по критерию получения прибыли комплекта проектов, финансово доступных для реализации. Основу данного этапа составляет оптимизационная модель, относящаяся к задачам о ранце. На втором этапе вводится понятие согласованного распределения прибыли и предлагается оригинальный алгоритм его отыскания. Основу алгоритма составляет модифицированная модель отыскания С-ядра кооперативной игры. Программная реализация этапов позволяет осуществить необходимые расчеты и принять оперативное решение.

Литература

1. Болтон, П. Теория контрактов / П. Болтон, М. Деватрпоинт. – Москва : Издательский дом «Дело» РАНХиГС, 2019.– 800 с.
2. Кузин, Б. И. Методы и модели управления фирмой / Б. И. Кузин, В. Н. Юрьев, Г. М. Шахдинаров. – Санкт-Петербург : ПИТЕР, 2001.– 432 с.
3. Долгих, Е. Н. Формирование интегрированного пространства и интегрированной информационной среды инновационных хозяйствующих субъектов / Е. Л. Долгих, А. В. Семенихина // Экономические и гуманитарные науки. – 2019.– № 3(326).– С. 97–108.
4. Колокольцов, В. Н. Математическое моделирование многоагентных систем конкуренции и кооперации / В. Н. Колокольцов, О. А. Малафеев. – Санкт-Петербург : Издательство «Лань», 2012. – 624 с.

Дыбов Евгений Иванович — студент 4-го курса кафедры математических методов исследования операций Воронежского государственного университета. E-mail: ivan.dibov@gmail.com

Бондаренко Юлия Валентиновна (научный руководитель) — д-р техн. наук, доц., профессор кафедры математических методов исследования операций ВГУ.

КОЛИЧЕСТВО АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ, НЕОБХОДИМЫХ ДЛЯ ОБРАЩЕНИЯ ТРЕУГОЛЬНОЙ МАТРИЦЫ

Л. И. Еникеева

Воронежский государственный университет

Введение

Треугольные матрицы возникают, например, в результате применения прямого хода метода Гаусса [2], QR-алгоритма [3] и при матричном представлении причинных операторов [6]. Один из способов обращения треугольной матрицы — применение процедуры обратного хода метода Гаусса. В настоящей заметке обсуждается алгоритм вычисления обратной к треугольной матрице, основанный на применении явной формулы для элементов обратной. Преимущество этого алгоритма по сравнению с алгоритмом обратного хода метода Гаусса заключается в том, что он вычисляет диагонали обратной матрицы последовательно друг за другом. Это удобно в тех случаях, когда заранее известно [1, 4], что диагонали обратной матрицы быстро убывают и поэтому для приближенного нахождения обратной матрицы достаточно посчитать лишь небольшое число диагоналей.

В настоящей заметке вычисляется число арифметических действий, необходимых для обращения треугольной матрицы путем использования явной формулы для элементов обратной. Оказывается, использование явной формулы требует в точности такого же количества операций, что и обратный ход метода Гаусса.

1. Количество операций в обратном ходе метода Гаусса

В этом параграфе вычисляется количество арифметических операций, которые необходимо совершить для обращения треугольной матрицы методом Гаусса. Метод Гаусса — метод поиска матрицы \mathbf{T}^{-1} или, что равносильно, решения уравнения

$$\mathbf{T}\mathbf{T}^{-1} = \mathbf{I}$$

или в матричном виде

$$\begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n,1} & t_{n,2} & \cdots & t_{n,n} \end{pmatrix} \begin{pmatrix} t_{1,1}^{(-1)} & t_{1,2}^{(-1)} & \cdots & t_{1,n}^{(-1)} \\ t_{2,1}^{(-1)} & t_{2,2}^{(-1)} & \cdots & t_{2,n}^{(-1)} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n,1}^{(-1)} & t_{n,2}^{(-1)} & \cdots & t_{n,n}^{(-1)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

Он заключается в преобразовании расширенной матрицы

$$\mathbf{D} = \left(\begin{array}{cccc|cccc} t_{1,1} & t_{1,2} & \cdots & t_{1,n} & 1 & 0 & \cdots & 0 \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{n,1} & t_{n,2} & \cdots & t_{n,n} & 0 & 0 & \cdots & 1 \end{array} \right)$$

к виду

$$\left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ 0 & 1 & \cdots & 0 & b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{array} \right)$$

путем проведения только элементарных преобразований: перестановки строк, умножения строки на число, не равное нулю, и сложения строки с другой, умноженной на число. Как известно [5, с. 128], возникающая справа матрица \mathbf{B} совпадает с матрицей \mathbf{T}^{-1} .

Метод Гаусса состоит из двух этапов. Первый этап — прямой ход метода — приведение матрицы к ступенчатому виду, следующий за ним — обратный ход метода — приведение к диагональному виду. Для обращения треугольной матрицы прямой ход метода проводить не нужно, матрица уже является ступенчатой.

Подсчитаем число операций, требуемых для совершения обратного хода метода Гаусса. Будем производить элементарные преобразования над расширенной матрицей

$$\mathbf{D} = \left(\begin{array}{cccc|cccc} t_{1,1} & t_{1,2} & \cdots & t_{1,n} & 1 & 0 & \cdots & 0 \\ 0 & t_{2,2} & \cdots & t_{2,n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & t_{n,n} & 0 & 0 & \cdots & 1 \end{array} \right)$$

для получения единичного блока слева от черты. Для начала разделим каждую строку на первый ненулевой элемент, то есть, в нашем случае, диагональный. Это потребует в k -й строке $(n - k)$ операций слева от черты и одну операцию справа, итого делений:

$$\sum_{k=1}^n (n - k + 1) = \frac{n^2 + n}{2}.$$

Далее последовательно обнулим элементы, стоящие над главной диагональю, начиная с n -го столбца. При этом не будем считать вычитание нуля и вычитание из нулевых элементов операциями. Второе действие можно интерпретировать как операцию, которая меняет знак перед уже имеющимся числом. Назовем эту операцию ссылкой. Также учтем, что в действительности операции слева от черты не производятся — запоминая необходимые для расчетов значения элементов, на их месте записываем нули, не производя никаких вычислений. Для обнуления элементов над главной диагональю в k -м столбце для $k = 1, \dots, n$ будет произведено $(k - 1)(n - k + 1)$ умножений, $(k - 1)(n - k)$ вычитаний, и $(k - 1)$ ссылок, итого для обнуления нужных элементов во всех столбцах потребуется

$$\sum_{k=1}^n (k - 1)(n - k + 1) = \frac{n^3}{6} - \frac{n}{6}$$

умножений,

$$\sum_{k=1}^n (k - 1)(n - k) = \frac{n^3}{6} - \frac{n^2}{2} + \frac{n}{3}$$

вычитаний и

$$\sum_{k=1}^n (k - 1) = \frac{n^2}{2} - \frac{n}{2}$$

ссылок. Таким образом, для получения ответа потребуется всего

$$\frac{n^3}{3} + \frac{2n}{3}$$

арифметических операций и

$$\frac{n^2}{2} - \frac{n}{2}$$

ССЫЛОК.

2. Явная формула для обратной к треугольной матрице

В этом и следующем параграфах используются нижнетреугольные матрицы, как это принято в теории причинных операторов, а в предыдущем параграфе использовались верхнетреугольные матрицы. Понятно, что суть дела от этого не меняется.

Приведем формулу [3, теорема 11.1.3, с. 485] для обратной к нижнетреугольной матрице в обозначениях из [7, теорема 8].

Предложение 1. *Предположим, что \mathbf{A} — нижнетреугольная матрица:*

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & 0 & \cdots & 0 & 0 \\ a_{2,1} & a_{2,2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-1} & 0 \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-1} & a_{n,n} \end{pmatrix}, \quad (1)$$

причем ее диагональные элементы обратимы. Тогда матрица обратима, а обратная матрица

$$\mathbf{A}^{-1} = \begin{pmatrix} a_{1,1}^{(-1)} & 0 & \cdots & 0 & 0 \\ a_{2,1}^{(-1)} & a_{2,2}^{(-1)} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1,1}^{(-1)} & a_{n-1,2}^{(-1)} & \cdots & a_{n-1,n-1}^{(-1)} & 0 \\ a_{n,1}^{(-1)} & a_{n,2}^{(-1)} & \cdots & a_{n,n-1}^{(-1)} & a_{n,n}^{(-1)} \end{pmatrix}$$

состоит из элементов

$$a_{i,j}^{(-1)} = \sum_{i=i_1 > i_2 > \dots > i_m = j} (-1)^{m+1} a_{i_1, i_1}^{-1} a_{i_1, i_2}^{-1} a_{i_2, i_2}^{-1} a_{i_2, i_3}^{-1} \dots a_{i_{m-1}, i_m}^{-1} a_{i_m, i_m}^{-1}, \quad i \geq j. \quad (2)$$

(Здесь и далее под суммой $\sum_{i=i_1 > i_2 > \dots > i_m = j}$ в случае $i = j$ понимается одно слагаемое, например,

$\sum_{i=i_1 = j} (-1)^{1+1} a_{i_1, i_1}^{-1} = a_{i,i}^{-1}$.) В частности,

$$a_{i,i}^{(-1)} = a_{i,i}^{-1},$$

$$a_{i+1,i}^{(-1)} = -a_{i+1,i+1}^{-1} a_{i+1,i} a_{i,i}^{-1},$$

$$a_{i+2,i}^{(-1)} = -a_{i+2,i+2}^{-1} a_{i+2,i} a_{i,i}^{-1} + a_{i+2,i+2}^{-1} a_{i+2,i+1} a_{i+1,i+1}^{-1} a_{i+1,i} a_{i,i}^{-1}.$$

3. Модифицированная формула для элементов обратной матрицы

Заметим, что формула (2) состоит из однотипных выражений, вычисление которых многократно повторяется. Организуем последовательность вычислений так, чтобы их можно было не повторять.

Во-первых, будем вычислять диагонали матрицы \mathbf{A}^{-1} последовательно: сначала главную, затем — находящуюся под ней и так далее.

Во-вторых, будем вначале вычислять выражения (2) без первого сомножителя $a_{i,i}^{-1}$ (обозначим соответствующие числа через $c_{i,j}$):

$$c_{i,j} = \sum_{i=i_1 > i_2 > \dots > i_m = j} (-1)^{m+1} a_{i_1, i_2}^{-1} a_{i_2, i_2}^{-1} a_{i_2, i_3}^{-1} \dots a_{i_{m-1}, i_m}^{-1} a_{i_m, i_m}^{-1}, \quad i \geq j, \quad (3)$$

а первый сомножитель $a_{i,i}^{-1}$ добавим в самом конце. В частности, положим $c_{i,j} = 1$.

Заметим, что в формулах (2) и (3) элемент $a_{i,j}$ всегда умножается справа на $a_{j,j}^{-1}$. Вычислим заранее числа

$$d_{i,j} = a_{i,j} a_{j,j}^{-1}.$$

Тогда это действие не придется повторять. На это уйдет $n(n-1)/2$ умножений (делений).

Заметим, что при вычислении по формулам (2) и (3) элементов k -й (главную диагональ мы считаем нулевой) диагонали для всех индексов на этой диагонали выполняется равенство $i-j=k$. Поэтому элементы $(k+1)$ -й диагонали можно выразить через элементы предыдущих диагоналей:

$$c_{i,j} = -\sum_{i_1=j}^{i-1} a_{i,i_1} a_{i_1,i_1}^{-1} c_{i_1,j} = -\sum_{i_1=j}^{i-1} d_{i,i_1} c_{i_1,j}. \quad (4)$$

Докажем эту формулу:

$$\begin{aligned} -\sum_{i_1=j}^{i-1} d_{i,i_1} c_{i_1,j} &= -\sum_{i_1=j}^{i-1} a_{i,i_1} a_{i_1,i_1}^{-1} c_{i_1,j} = \\ &= -\sum_{i_1=j}^{i-1} a_{i,i_1} a_{i_1,i_1}^{-1} \sum_{i=i_0>i_1>i_2>\dots>i_m=j} (-1)^{m+1} a_{i_1,i_2} a_{i_2,i_2}^{-1} a_{i_2,i_3} \dots a_{i_{m-1},i_m} a_{i_m,i_m}^{-1} = \\ &= \sum_{i=i_0>i_1>i_2>\dots>i_m=j} (-1)^{m+2} a_{i_0,i_1} a_{i_1,i_1}^{-1} a_{i_1,i_2} a_{i_2,i_2}^{-1} a_{i_2,i_3} \dots a_{i_{m-1},i_m} a_{i_m,i_m}^{-1}. \end{aligned}$$

Вычисление одного числа $c_{i,j}$ по формуле (4) требует $i-j-1$ умножений (на одно меньше, чем число элементов на диагонали, потому что при вычислении $d_{i,j} c_{j,j}$ умножать на $c_{i,j} = 1$ не требуется) и $i-j-1$ сложений.

Вычисление чисел $c_{i,i}$, стоящих на главной диагонали, арифметических операций не требует, поскольку эти числа являются единицами. Вычисление первой поддиагонали также вычислений не требует, поскольку на ней стоят числа $-d_{i+1,i}$. Вычисление k -й поддиагонали (на ней $(n-k)$ чисел) требует $(n-k)(k-1)$ умножений и $(n-k)$ сложений. Таким образом, для всех диагоналей потребуется

$$\sum_{k=2}^{n-1} (n-k)(k-1) = \frac{1}{6}(n^3 - 3n^2 + 2n)$$

умножений,

$$\sum_{k=2}^{n-1} (n-k)(k-1) = \frac{1}{6}(n^3 - 3n^2 + 2n)$$

сложений, а всего операций —

$$\frac{1}{3}(n^3 - 3n^2 + 2n).$$

С учетом вычисления чисел $d_{i,j}$ потребуется

$$\frac{1}{2}n(n-1) + \sum_{k=2}^{n-1} (n-k)(k-1) = \frac{1}{6}n(n^2 - 1)$$

умножений (делений), столько же (как и раньше)

$$\sum_{k=2}^{n-1} (n-k)(k-1) = \frac{1}{6}(n^3 - 3n^2 + 2n)$$

сложений, а всего операций —

$$\frac{1}{6}n(2n^2 - 3n + 1).$$

В конце надо от чисел $c_{i,j}$ вернуться к числам $a_{i,j}^{(-1)}$:

$$a_{i,j}^{(-1)} = a_{i,i}^{-1} c_{i,j}.$$

Это потребует дополнительно $n(n+1)/2$ умножений (делений). С учетом этого потребуется

$$\frac{1}{6}n(n^2 + 3n + 2)$$

умножений, а всего операций —

$$\frac{1}{3}n(n^2 + 2).$$

В случае метода Гаусса число операций точно такое же:

$$\frac{1}{3}n(n^2 + 2).$$

Еще раз повторим, что преимущество описанного алгоритма по сравнению с алгоритмом обратного хода метода Гаусса заключается в том, что он вычисляет диагонали обратной матрицы последовательно друг за другом. Это оказывается удобным в тех случаях, когда заранее известно [1, 4], что диагонали обратной матрицы быстро убывают и поэтому для приближенного нахождения обратной матрицы достаточно посчитать лишь небольшое число диагоналей.

Литература

1. Баскаков, А. Г. Теорема Винера и асимптотические оценки элементов обратных матриц / А. Г. Баскаков // Функц. анализ и его прил. – 1990. – Т. 24, № 3. – С. 64–65.
2. Березин, И. Методы вычислений / И. С. Березин, Н. П. Жидков. – Второе изд. – Москва : Наука, 1962. – Т. 2. – 640 с.
3. Голуб, Д. Матричные вычисления / Дж. Голуб, Ч. Ван Лоун. – Москва : Мир, 1999. – 548 с.
4. Курбатов, В. Г. Об алгебрах разностных и интегральных операторов / В. Г. Курбатов // Функц. анализ и его прил. – 1990. – Т. 24, № 2. – С. 87–88.
5. Сборник задач по математике для вузов. Линейная алгебра и основы математического анализа / Под ред. А. В. Ефимова, Б. П. Демидовича. – Москва : Наука, 1993. – Т. 1. – 480 с.
6. Kurbatov, V. G. Functional differential operators and equations / V. G. Kurbatov. – Dordrecht : Kluwer Academic Publishers, 1999. – Vol. 473 of Mathematics and its Applications. – xx+433 p.
7. Kurbatov, V. G. Green's function of the problem of bounded solutions in the case of a block triangular coefficient / V. G. Kurbatov, I. V. Kurbatova // Operators and Matrices. – 2019. – Vol. 13, No 4. – P. 981–1001.

Еникеева Лада Ильясовна — студентка 2-го курса Воронежского государственного университета. E-mail: enikeili@mail.ru

Курбатов Виталий Геннадьевич (научный руководитель) — д-р физ.-мат. наук, проф., профессор кафедры системного анализа и управления Воронежского государственного университета. E-mail: kv51@inbox.ru

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, РЕАЛИЗУЮЩЕГО МЕТОДЫ ОБРАБОТКИ ИЗОБРАЖЕНИЙ ДЛЯ АНАЛИЗА ФАЦИЙ КРОВИ

А. В. Жидких

Воронежский государственный университет

Введение

Использование методов компьютерной обработки изображений для анализа фаций (высушенных капель биологической жидкости) крови позволяет частично, насколько это возможно, автоматизировать процесс формирования и распознавания отличительных признаков, которые могут помочь обнаружить те или иные патологические процессы. Основной целью данной статьи является описание программного обеспечения, разработанного для того, чтобы облегчить конечному пользователю задачу эффективного использования методов обработки изображений. Это достижимо созданием простого и понятного интерфейса, благодаря которому пользователю не было бы необходимости разбираться в коде программы, а также исключить возможность вмешательства в этот код.

1. Постановка задачи

Задача работы состоит в следующем: требуется объединить реализованные методы в единую модульную структуру, упростить вызов функций и реализовать безопасный вызов функций из консоли и написать справку, содержащую информацию о функциях и их параметрах.

2. Технические требования

Для использования реализованной программы необходимо:

1. Python 3.7.0+;
2. Pillow 5.2.0+, opencv 3.4.1+;
3. 4GB RAM

3. Описание методов и функций

3.1. Основное тело программы и инициализация

Основное тело программы состоит из вызова метода *restart()*, который будет подробно рассмотрен в следующем абзаце, цикла, который предоставляет пользователю вызов методов и функций из консоли, пока не будет вызван метод *close()*, и словаря со всеми доступными пользователю командами.

Описание методов и функций начинается с описания первого метода *restart()*, который автоматически выполняется при запуске программы и может быть вызван пользователем впоследствии. Метод запускает цикл *while(flag)*, с еще одним аналогичным вложенным циклом. Сначала пользователю предоставляется ввод относительного или абсолютного пути к проекту, по возможности исправляя ошибки пользователя. Затем папка проекта проверяется на наличие и создается, если таковой не имеется. Далее пользователю предоставляется возможность ввести имя файла с форматом, находящегося в этой папке. Если такого файла нет или

название введено неверно, то метод будет запрашивать имя файла снова и снова. Если папка была выбрана неверно, то пользователя может ввести команду *dir* для возвращения на предыдущий шаг, иными словами, вложенный цикл завершится. Если файл все-таки будет найден, то переменной *flag* будет присвоено значение *false* и оба цикла завершатся. Далее пользователю будет предложено ввести формат файла, по возможности исправляя ошибки, если будет необходимость сохранять результаты функций в другом формате. Также пользователь может ввести команду *same*, чтобы не вводить формат вручную, если такой необходимости нет. Далее метод выводит уведомление о загрузке, во время которой определяет все глобальные переменные, необходимое для других функций, а именно: исходное изображение, черно-белое исходное изображение, размер изображения. Если ширина и высота изображения разные, то метод выводит уведомление, что некоторые функции будут недоступны.

3.2. Функции и методы, доступные пользователю

В начале будут описаны небольшие методы и функции, которые не связаны напрямую с обработкой изображений, но предоставляют некоторые инструменты взаимодействия с программой и изображениями.

Функция *resize(percent, show=1, save=0)* — функция изменения размера изображения. Параметр *show* определяет, будет ли выведен результат в отдельном окне. Этот параметр присутствует в большинстве функций обработки изображений. Параметр *save* определяет, будет ли результат сохранен под именем *imagename_percent_percent*. Этот параметр так же присутствует в большинстве функций обработки изображений. Параметр *percent* определяет, в каком процентном соотношении будет исходное изображение с новым изображением.

Метод *open()* открывает папку с проектом в проводнике.

Метод *close()* меняет условие основного цикла программы на *false*.

Функция *helpme(command='general')* при отсутствии параметра *command* выводит список всех команд, доступных пользователю. Если же параметр *command* указан, то функция выводит более подробную справку по конкретной команде с описанием всех параметров и сути команды.

Далее будут описаны функции, связанные непосредственно с обработкой изображений.

Функция *contours(ft, st=255, show=1, save=0)* осуществляет поиск контуров на изображении по заданным значениям нижней и верхней границы порога (параметры *ft* и *st* соответственно) [1].

Функция *wavelet(lvl, show=1, save=0)* осуществляет упрощенное вейвлет преобразование, где параметр *lvl* определяет степень декомпозиции. [1]

Функция *hough(threshold, dilation, Ex, show=1, save=0)* осуществляет поиск и отрисовку прямолинейных структур на изображении с помощью комбинации классических трансформаций Хафа и Харриса, а также дополнительных вычислений. Параметр *threshold* определяет порог для предобработки изображения. Параметры *dilation* и *Ex* определяют количество соответствующих морфологических операций для предобработки. Также присутствуют скрытые от пользователей параметры, такие как *savelines*, который записывает координаты линий в файл, и *editmode*, который рисует на каждой линии ее номер и позволяет их удалять. Эти дополнительные возможности нужны прежде всего для отладки функции. Сложно представить, будет ли изображение достаточно обработано для эффективного применения алгоритма после доступной пользователю обработки, поэтому использование этой функции не рекомендуется, так как она требует доработки. [1]

Функция *iminfo(filename="characteristics.txt", print=0)* рассчитывает такие характеристики изображения, как его энергию, энтропию, контрастность, корреляцию и гомогенность и со-

хранят их в файл. Параметр *print* определяет, будут ли эти характеристики также выведены и в консоль [1].

Функция *points(threshold, size=Size/20)* с помощью SURF рассчитывает локальные точки интереса изображения, найденные соответственно порогу, указанному в параметре *threshold*, а затем вырезает их соответственно параметру *size*, после чего применяю к каждому из них функцию *iminfo*. Все результаты сохраняются в подпапку проекта *./localpoints/*. Функция выведет предупреждение, если точек слишком много, и запросит либо увеличить порог, либо подтвердить решение. Не рекомендуется применять эту функцию, если точек больше 100. Функция требует оптимизации и доработки [2].

4. Описание интерфейса

Интерфейс приложения представляет собой консоль. Интерпретация функций происходит с помощью функции *eval()*. Эта функция используется для парсинга строковой переменной как выражения Python и затем его выполнения. Параметры этой функции следующие: *expression* — обязательный строковый параметр, который анализируется и выполняется как выражение Python; *globals* — словарь, используемый для указания методов, доступных для выполнения; *locals* — словарь, используемый для указания локальных переменных и методов, доступных функции *eval()*.

Опасность функции *eval()* заключается в том, что пользователю по умолчанию доступны все встроенные методы и функции. Для того, чтобы этого избежать, необходимо указать в словаре *'__builtins__'*: *None*, либо перечислить доступные пользователю встроенные методы и функции после двоеточия. Также необходимо указать в словаре остальные методы и функции, которые будут разрешены пользователю.

При запуске программы запускается метод инициализации *restart()*, который предлагает пользователю выбрать папку проекта, а затем изображение в ней, после чего запросит ввод формата, в котором будут сохраняться результирующие изображения. Далее пользователю предлагается ввести команду, а также выводится совет воспользоваться функцией *helpme()*, чтобы получить список команд и доступную справку по ним, методом *close()*, чтобы закрыть программу и методом *open()*, чтобы открыть папку с проектом.

В процессе работы каждого вызванного метода или функции выводятся сообщения о начале их выполнения, конце их выполнения и уведомление, что работа программы приостановлена. Выполнение программы приостанавливается, когда открыто отдельное окно с просмотром результата работы функции, если при вызове функции пользователь указал в ее параметрах вывод результата на экран.

Выход из программы осуществляется закрытием консоли, закрытием процесса *python.exe* через консоль или вводом команды *close()*.

5. Пример работы программы

Программа была запущена из командной строки с помощью команды *python*. С помощью команды *cd* была выбрана папка с файлом программы. Было активировано окружение с помощью команды *conda activate base*. В процессе инициализации была создана папка “Test” в директории программы, куда было помещено изображение фации крови “test.tif”, которое имеет следующие свойства: цветное изображение с глубиной цвета 24, с шириной и высотой по 1729 пикселей. Исходное изображение можно наблюдать на рис. 1.

Была выведена команда *helpme()*, которая выводит список всех команд и их параметров. Далее были применены команды *hough()* (поиск линейных структур) и *points()* (поиск локальных точек интереса и их анализ). Все введенные команды и сам интерфейс можно видеть на рис. 2.

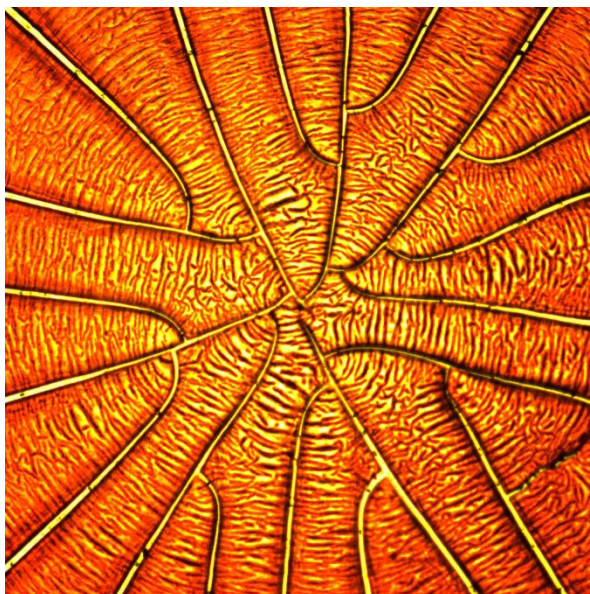


Рис. 1. Исходное изображение test.tif

```
Администратор: C:\Windows\system32\cmd.exe
(base) C:\Users\Alex>cd C:\Users\Alex\Desktop\Test\

(base) C:\Users\Alex\Desktop\Test>python wavelet.py
////////////////////////////////////
Sample Text
Input directory
Test
Input file name with file format. To change directory type: dir
test.tif
Input save format. If it is the same as previous format, you can type: same
same
Loading...
Input a command. To close application type close(). To see command list type helpme()
helpme()

To get info on certain function type help('command_name'). Here is the functions you can use:
restart()
open()
close()
resize(percent, show=1, save=0)
contours(ft, st=255, show=1, save=0)
wavelet(lvl, show=1, save=0)
hough(threshold, dilation, Ex, show=1, save=0)
iminfo(filename='Characteristics.txt',Print = 0)
points(threshold, size=Size/20)

Input a command. To close application type close(). To see command list type helpme()
hough(80, 2, 1, 1, 1)

Input a command. To close application type close(). To see command list type helpme()
points(30000, 80)

Initiating search of points of interest...
Analyzing point 1
Analyzing point 2
Analyzing point 3
Analyzing point 4
Analyzing point 5
Analyzing point 6
Search of points of interest done! You can find the results in your_folder/localpoints/
Input a command. To close application type close(). To see command list type helpme()
close()

See you later!

(base) C:\Users\Alex\Desktop\Test>
```

Рис. 2. Интерфейс программы

Результаты работы `hough()` и `points()` были выведены на экран и сохранены в папке проекта. На рис. 3а. представлено изображение с найденными линейными структурами. На рис. 3б. представлен основной вывод поиска и анализа локальных точек интереса. Также в папке с проектом была создана директория с результатами анализа каждой точки.



Рис. 3а. Результат работы `hough()`

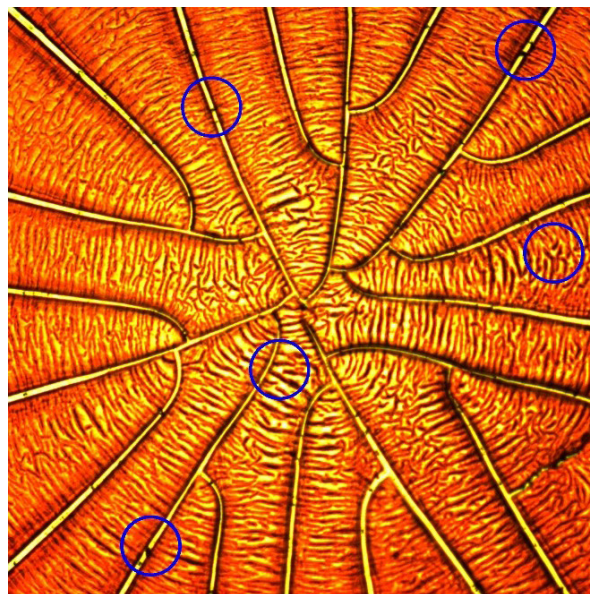


Рис. 3б. Результат работы `points()`

Заключение

На текущий момент программа требует продолжения тестирования и длительную поддержку. В процессе написания этой статьи были найдены и исправлены ошибки, которые породили другие ошибки, которые также были исправлены. Запланирована доработка алгоритмов, которая сделала бы их более универсальными и менее зависимыми от входных данных. Также в планах написание алгоритмов, которые бы сравнивали два изображения.

Литература

1. Жидких, А. В. Применение методов обработки изображений для анализа фаций крови до и после применения аппарата Cell-Saver / А. В. Жидких, С. Н. Медведев // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 13-15 декабря 2021 г. – Воронеж, 2022. – С. 415-420.
2. Introduction to SURF (Speeded-Up Robust Features). – Режим доступа: <https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e>. – Дата обращения: (20.04.2022)
3. Шабалин, В. В. Биофизически механизмы формирования твердофазных структур биологических жидкостей человека: автореф. дис. ... д-ра биологических наук: 03.01.02 / Шабалин Владимир Владимирович ; ФГБУИФ. – Санкт-Петербург, 2018. – 48 с.
4. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – 3-е изд, испр. и доп. – М. : Техносфера, 2012. – 1104 с.

Жидких Алексей Валерьевич — магистрант 2-го года обучения кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: barsoval12@gmail.com

МЕТОД АНАЛИЗА СТОХАСТИЧЕСКОЙ ГРАНИЦЫ ПРИ ОЦЕНКЕ ЭФФЕКТИВНОСТИ ПРОИЗВОДСТВЕННОГО ПОТЕНЦИАЛА

Н. Н. Карякин, И. Н. Щепина

Воронежский государственный университет

Введение

Производственный потенциал — термин, который используют для характеристики возможности производителя. Оценка производственного потенциала предполагает описание возможных результатов производственной деятельности, которые обычно измеряются как объем произведенной продукции либо величина дохода, полученная производителем продукции от ее реализации. При этом объемы применяемых ресурсов зачастую используется исключительно как источник для получения результата производственной деятельности, а в случае, когда объем использованных ресурсов является ключевой характеристикой производства, говорят о ресурсном потенциале.

Неоклассическая теория производства полагает, что максимально возможный результат производственной функции детерминирован объемом ресурсов, а производственный потенциал, соответствующий данному представлению, принято называть детерминированной производственной функцией [1]. При попытках эмпирически оценить параметры такой функции пришло понимание того, что случайные воздействия носят объективный характер, что означает, что при фиксированных объемах ресурсов могут быть получены различные производственные результаты.

На данном этапе развитие детерминистского подхода пришло к появлению и развитию стохастической производственной функции, для оценки параметров которой применялся метод наименьших квадратов и его модификации, которые позволяли предполагать о том, что случайные воздействия приводят к случайным (симметричным) отклонениям в ту или иную сторону от ожидаемых результатов. Однако такие оценки являются справедливыми в теоретическом отношении лишь для эффективного производства, а в иных случаях предположения о симметричных отклонениях распределения остатков эконометрической модели выполняться не будут. В связи с данным фактом традиционные подходы не позволяли получить теоретически обоснованные эмпирические оценки параметров производственной функции и эффективности производства.

1. Стохастическая граница

1.1. Зарождение модели

Концепция стохастической границы зародилась в семидесятых годах прошлого столетия. Она позволила получить ответ на важный вопрос: является ли производство эффективным? Иными словами, модель позволяла зафиксировать наличие эффективности или ее отсутствия при наблюдении за производственными объектами. Как следствие, стали появляться методы измерения эффекта снижения эффективности производства, которые позволяли провести соответствие с оценками технологической эффективности. Кроме того, родилась новая модель, которая позволяла описывать результаты эффективного производства — стохастическая граничная производственная функция. Её преимущество заключалось в том, что эмпирическая

оценка ее параметров не требует априорных предположений об уровне эффективности производства, позволяя при этом учитывать и дифференцировать результаты исследования производственных факторов и последствия случайных воздействий, которые приводят к снижению эффективности, а также прочих случайных воздействий, существующих в условиях эффективного производства.

1.2. Концепция стохастической границы

В основу концепции стохастической границы положено разделение всех совокупностей производственных факторов на две части: основные и сопутствующие. Результат воздействия основных факторов детерминирован их объемом, а сопутствующие, в свою очередь, оказывают случайное воздействие на результаты производственной деятельности. При этом вся совокупность сопутствующих факторов, в свою очередь, так же делится на две группы. Первые характеризуются сбалансированным влиянием на производственный процесс, то есть могут как повысить, так и снизить его объем. Результат воздействия же этой группы принято называть «белый шум». Примером может служить солнечная активность, атмосферное давление, то есть неуправляемые факторы, совокупное воздействие которых характеризуется нормальным распределением. С другой стороны, вторая группа факторов включает в себя параметры, систематическое воздействие которых снижают эффективность производства. К ним можно отнести отсутствие мотивации персонала, недостаточную квалификацию. Такие факторы обычно называются факторами эффективности.

2. Анализ стохастической границы

2.1. Метод SFA

Метод стохастического анализа SFA (Stochastic Frontier Analysis) был предложен в 1977 г. Aigner, Lovell and Schmit [2]. В его основе лежит оценивание стохастической производственной функции или функции затрат со случайной ошибкой. При этом предполагается, что ошибка содержит две независимые компоненты: несистематическую ошибку v_i с нормальным законом распределения и нулевым математическим ожиданием и систематическую ошибку u_i , характеризующую неэффективность производства, выраженную как усеченное в нуле нормальное распределение.

В общем виде модель может быть записана в следующем виде:

$$\ln Y_i = \sum_{m=1}^M \beta_m \ln X_{mi} + v_i - u_i, \quad i = \overline{1, n}.$$

Здесь Y_i — зависимая переменная, X_{mi} — факторы, β_m — параметры модели. Для идентификации параметров используется метод максимального правдоподобия.

Одним из главных преимуществ метода SFA является то, что он оценивает «истинную» границу производства, а не среднее среди всех. Кроме того, измеренное изменение продуктивности будет представлено только изменением в технологиях и инновациях, а не комбинацией изменения эффективности и изменения технологий, как это происходит в случае оценки неграничными моделями.

2.2. Сравнение с DEA

Метод DEA (Data Envelopment Analysis) применяется к определению технической эффективности наряду с методом SFA. Их различие состоит в том, что для DEA применяется детер-

минированный подход для определения граничной производительности, в отличие от стохастического для SFA.

Метод SFA имеет как преимущества, так и недостатки перед DEA. Преимуществом служит фактор того, что SFA учитывает в себе случайные ошибки измерения. Говоря же о недостатках, для SFA необходимо выбирать функциональную форму для производственного функционала и для формализации вида ошибки, чего не требует DEA.

Наглядно видеть различия между результатами работы моделей можно на рис. 1.

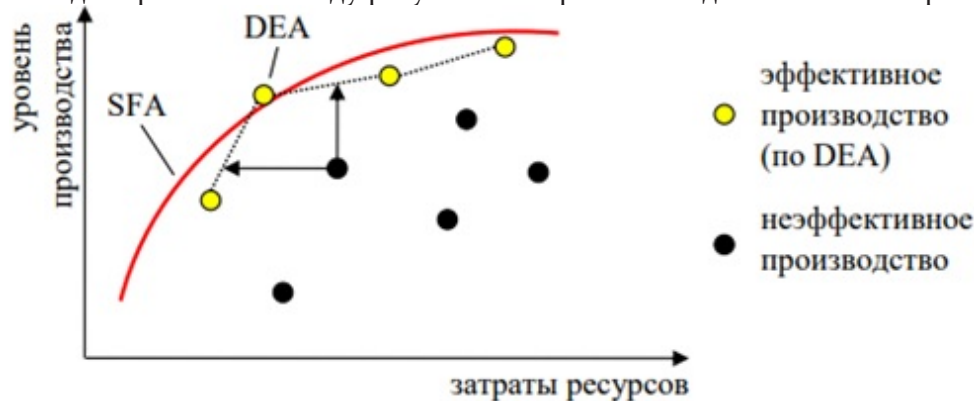


Рис. 1. Графическая иллюстрация задач DEA и SFA

В DEA граничный производственный уровень образует ломаную линию, проходящую через наиболее эффективные объекты. В SFA он представлен аналитической моделью, причем за счет случайных факторов отдельные объекты могут и оказаться за его границами.

Заключение

Модель SFA на сегодняшний день применяется во многих сферах экономики и стала популярной благодаря своей гибкости, наличию большого количества модификаций и возможности адекватно применить экономические концепции к реалиям. Одной из наиболее популярных целей использования данного метода в России является оценка эффективности регионов, их инновационной активности, а также при принятии инвестиционных решений.

Литература

1. Айвазян, С. А. Моделирование производственного потенциала на основе концепции стохастической границы: Методология, результаты эмпирического анализа / С. А. Айвазян, М. Ю. Афанасьев. – Москва : Изд-во Крсанд, 2014. – 352 с.
2. Малахов, Д. И. Методы оценки показателя эффективности в моделях стохастической производственной границы / Д. И. Малахов, Н. П. Пильник // Экономический журнал ВШЭ. – 2013. – С. 692–718.
3. Щетинин, Е. И. Стохастическая граница производственных возможностей и факторы технической эффективности предприятий российской обрабатывающей промышленности: автореф. дис. на соиск. учен. степ. канд. экономических наук (08.00.13) / Щетинин Евгений Игоревич; Национальный исследовательский университет «Высшая школа экономики». – Москва – 2016. – 29 с.
4. Семенычев, В. К. Эволюция технической эффективности российских регионов в 2011–2016 годы: метод стохастического анализа SFA с пространственными эффектами / В. К. Семенычев, Г. А. Хмелева, В. Н. Кожухова. – Самара, 2018. – 28 с. (Препр. / ФГБОУВО «Самарский государственный университет»)

Карякин Никита Николаевич — студент 4-го курса кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: ni098754@gmail.com

Щепина Ирина Наумовна (научный руководитель) — д-р экон. наук, доц., доцент кафедры математических методов исследования операций Воронежского государственного университета.

ПРИМЕНЕНИЕ АЛГОРИТМОВ НЕЙРОННЫХ СЕТЕЙ ДЛЯ АНАЛИЗА ЭМОЦИОНАЛЬНОЙ ОКРАСКИ ТЕКСТА

О. А. Каширских

Воронежский государственный университет

Введение

Современный мир нельзя представить без чтения и обработки информации. Объём информации, который человек получает постоянно, огромен. В то же время всё более популярным и используемым становится делегирование различных задач компьютерной технике, в том числе и нейронным сетям. Всеобщее увлечение нейросетевыми технологиями и глубинным обучением не обошло стороной и компьютерную лингвистику — автоматическую обработку текстов на естественном языке. Повышенное внимание лингвистов к нейронным сетям обусловлено несколькими причинами. Применение нейронных сетей, во-первых, существенным образом повышает качество решения некоторых стандартных задач классификации текстов и последовательностей, во-вторых, снижает трудоемкость при работе непосредственно с текстами, в-третьих, позволяет решать новые задачи.

В связи с этим представляется актуальным решение задачи анализа эмоциональной окраски текста с помощью нейронных сетей.

1. Предварительная обработка данных

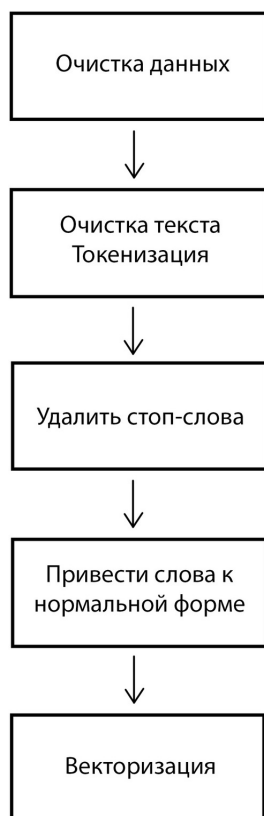


Рис. 1. Этапы предварительной обработки данных

Предварительная обработка и очистка данных должны проводиться до того, как набор данных будет использоваться для обучения модели. Необработанные данные зачастую искажены и ненадежны, и в них могут быть пропущены значения. Использование таких данных при моделировании может приводить к неверным результатам.

Реальные данные собираются для последующей обработки из разных источников и процессов. Они могут содержать ошибки и повреждения, негативно влияющие на качество набора данных. Типичные проблемы с качеством данных являются:

1. Неполнота: данные не содержат атрибутов, или в них пропущены значения.
2. Шум: данные содержат ошибочные записи или выбросы.
3. Несогласованность: данные содержат конфликтующие между собой записи или расхождения.

Предварительную обработку данных можно разделить на несколько частей. На рис. 1 представлена схема предварительной обработки данных.

К главным этапам предварительной обработки данных относятся:

1. Очистка данных: заполнение отсутствующих значений, обнаружение и удаление шума данных и выбросов.

2. Очистка текста — удаление внедренных символов, которые могут нарушать выравнивание данных.

3. Токенизация — это процесс разбиения текста на более мелкие части.

4. Удалить стоп-слова. Стоп-слова — это слова, которые могут иметь важное значение в человеческом общении, но не имеют смысла для машин.

5. Привести слова к нормальной форме. В процессе нормализации все формы слова приводятся к единому представлению. Есть два основных подхода к нормализации: стемминг и лемматизация. В случае стемминга выделяется основа слова, дополнив которую можно получить слова-потомки. Такой метод сработает на приведенном примере. Однако это наивный подход — стемминг просто обрезает строку, отбрасывая окончание. Такой метод не обнаружит связь между *feel* и *felt*. Лемматизация стремится решить указанную проблему, используя структуру данных, в которой все формы слова связываются с его простейшей формой — леммой.

6. Векторизация — преобразование токена в числовой массив, который представляет его свойства. В контексте задачи вектор уникален для каждого токена. Векторные представления токенов используются для оценки сходства слов, классификации текстов и т. д.

Все эти шаги служат для уменьшения шума, присущего любому обычному тексту, и повышения точности результатов классификатора. Для решения указанных задач есть несколько отличных библиотек, например, NLTK, TextBlob и spaCy.

2. Проектирование нейронной сети

Проектирование нейронной сети можно разделить на следующие этапы:

1. Разделение данных на обучающую и тестовую выборки (наборы данных).
2. Выбор архитектуры модели.
3. Использование обучающих данных для настройки параметров модели.
4. Использование тестовых данных, чтобы оценить качество обучения модели.
5. Использование обученной модели на новых, ранее не рассматриваемых входных данных для создания прогнозов.

Для решения задачи была выбрана сверточная нейронная сеть. Сверточная нейронная сеть обычно представляет собой чередование сверточных слоев, субдискретизирующих слоев и при наличии полносвязных слоев на выходе. Все три вида слоев могут чередоваться в произвольном порядке. В сверточном слое нейроны, которые используют одни и те же веса, объединяются в карты признаков, а каждый нейрон карты признаков связан с частью нейронов предыдущего слоя. При вычислении сети получается, что каждый нейрон выполняет свертку некоторой области предыдущего слоя. На рис. 2 архитектура сверточной нейронной сети.

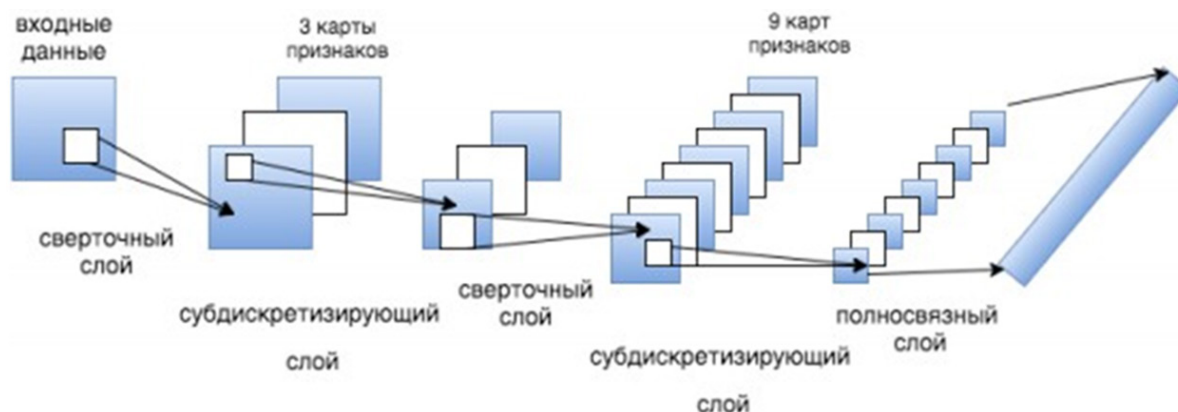


Рис. 2. Архитектура сверточной нейронной сети

Для обучения модели были взяты отзывы с портала «Кинопоиск». Проектирование сверточной нейронной сети производилось на языке программирования «Python». Для построения сверточной нейронной сети использовалась библиотека Keras, которая выступает высокоуровневой надстройкой над TensorFlow, CNTK и Theano. Входными данными является матрица с фиксированной высотой n , где каждая строка представляет собой векторное отображение токена в признаковое пространство размерности k . Для формирования признакового пространства использовался инструмент дистрибутивной семантики Word2Vec.

На первом этапе входная матрица обрабатывается слоями свертки.

В разработанной архитектуре использовались фильтры с высотой $h = (2, 3, 4, 5)$, которые предназначены для параллельной обработки биграмм, триграмм, 4-грамм и 5-грамм соответственно.

В нейронную сеть были добавлены по 10 сверточных слоев для каждой высоты фильтра, функция активации — ReLU.

После обработки слоями свертки, карты признаков поступали на слои субдискретизации, где к ним применялась операция 1-max-pooling, тем самым извлекая наиболее значимые n -граммы из текста. На следующем этапе происходило объединение в общий вектор признаков (слой объединения), который подавался в скрытый полносвязный слой с 30 нейронами. На последнем этапе итоговая карта признаков подавалась на выходной слой нейронной сети с сигмоидальной функцией активации.

Поскольку нейронные сети склонны к переобучению, после embedding-слоя и перед скрытым полносвязным слоем был добавлен dropout-регуляризацию с вероятностью выброса вершины $p = 0.2$.

Модель была сконфигурирована с функцией оптимизации Adam (Adaptive Moment Estimation) и бинарной кросс-энтропией в качестве функции ошибок. Качество работы классификатора оценивал в критериях макро-усредненных точности, полноты и f -меры.

3. Оценка прогресса обучения модели

В ходе оценки обучения были использованы следующие метрики:

Истинно положительные (true positives, TP) — число отзывов, которые модель правильно предсказала как положительные.

Ложноположительные (false positives, FP) — число отзывов, которые модель неверно предсказала как положительные, хотя на самом деле они были негативными.

Истинно отрицательные (true negatives, TN) — число отзывов, которые модель правильно предсказала как негативные.

Ложноотрицательные (false negatives, FN) — число отзывов, которые модель неверно предсказала как негативные, хотя на самом деле они были положительными.

Поскольку модель для каждой метки возвращает оценку от 0 до 1, положительный или отрицательный результат определяется на основе этой оценки. На основе четырех описанных статистических данных вычислены две метрики: точность и полнота. Эти метрики являются показателями эффективности модели классификации:

Точность (precision) — отношение истинно положительных результатов ко всем элементам, отмеченным моделью как положительные (истинные и ложные срабатывания). Точность 1.0 означает, что каждый отзыв, отмеченный моделью как положительный, действительно относится к положительному классу.

Полнота (recall) — это отношение истинно положительных отзывов ко всем фактическим положительным отзывам, то есть количество истинно положительных отзывов, деленных на суммарное количество истинно положительных и ложноотрицательных отзывов.

Ещё более популярной метрикой является F1-мера — среднее гармоническое точности и полноты.

Наивысший показатель $F1=76.80\%$ на валидационной выборке был достигнут на третьей эпохе обучения. Качество работы обученной модели на тестовых данных составило $F1=78.1\%$.

На рис. 3 представлены оценки точности обучения на тестовой выборке.

Метка класса	Точность	Полнота	F_1	Количество объектов
Negative	0.78194	0.78243	0.78218	22457
Positive	0.78089	0.78040	0.78064	22313
avg / total	0.78142	0.78142	0.78142	44770

Рис. 3. Оценки точности обучения

Заключение

В статье были рассмотрены этапы анализа текста с помощью нейронных сетей, подробно расписаны стадии предобработки текстов, оценки точности обучения для нейронных сетей. В качестве примера анализа текста с помощью нейронных сетей была спроектирована и продемонстрирована сверточная нейронная сеть на языке Python, анализирующая отзывы на фильмы.

Литература

1. Бенгфорт, Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорт, Р. Билбро, Т. Охеда. – СПб. : Питер, 2019. – 368 с.
2. Макмахан, Б. Знакомство с PyTorch: глубокое обучение при обработке естественного языка / Б. Макмахан, Д. Рао. – СПб.: Питер, 2020. – 258 с.;
4. Грушевицкая, Т. Г. Культурология : учебник для вузов / Т. Г. Грушевицкая, А. П. Садохин. – 3-е изд., перераб. и доп. – Москва : ЮНИТИ-ДАНА, 2008. – 687 с.
3. Круглов, В. В. Искусственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. – Москва : Горячая Линия-Телеком, 2002. – 382 с.
4. Ясницкий, Л. Н. Введение в искусственный интеллект / Л. Н. Ясницкий – 2-е изд., испр. – Москва : Академия, 2008. – 176 с.
5. Фаустова, К. И. Нейронные сети: применение сегодня и перспективы развития / К. И. Фаустова // Территория науки. – 2017. – № 3. – С. 89–91.

Каширских Ольга Александровна — студентка 2 курса магистратуры направления «Прикладная математика и информатика» Воронежского государственного университета. E-mail: olya_kashirskih@mail.ru

Ляликова Виктория Геннадиевна (научный руководитель) — канд. тех. наук, доцент кафедры ERP-систем и бизнес-процессов Воронежского государственного университета. E-mail: vikalg@yandex.ru

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДВИЖЕНИЯ СЫРЬЯ В МЕЖВИТКОВОМ ПРОСТРАНСТВЕ ШНЕКОВОГО ОБОРУДОВАНИЯ

М. В. Копылов¹, Е. А. Татаренков², Р. С. Дюбин², М. В. Голиков²

¹Воронежский государственный университет инженерных технологий

²ВУНЦ ВВС «ВВА им. проф. Н. Е. Жуковского и Ю. А. Гагарина»

Введение

Для сохранения всех ценных веществ в растительных маслах, полученным методом холодного прессования, необходимо правильно подобрать рациональные параметры для его извлечения.

Рассмотрим общую схему процесса получения масла (рис. 1). Сырье через приемник 1 поступает в рабочее пространство шнекового пресса, где в области I происходит сжатие сырья с уплотнением мятки, что приводит к выделению масла, которое в области II фильтруется через разделительную перегородку 2 под действием разности давлений в рабочем пространстве и окружающей среде. Отработанный жмых удаляется через специальные каналы [2].

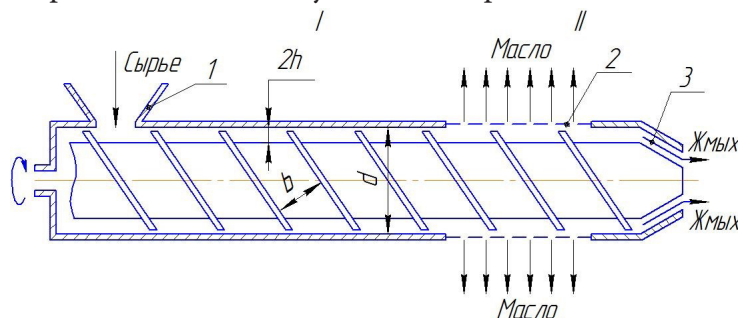


Рис. 1. Схема шнекового пресса

Таким образом, рабочую область можно разбить на две последовательные зоны. Задача области 1 состоит в максимальном увеличении давления в сырье с ограничением температурного режима. Задача области 2 состоит в максимальном извлечении масла. Хотя в действительности винт червячного пресса вращается, картину течения легче представить, если, наоборот, рассматривать вращающимся вокруг неподвижного шнека его наружный цилиндр (обечайку) [3].

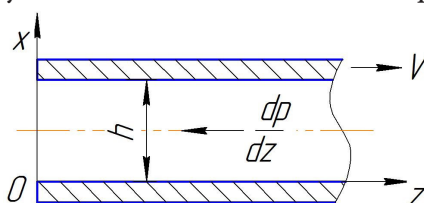


Рис. 2. Схема течения в шнековом канале

Если кривизна кольцевого канала невелика, то система эквивалентна течению между параллельными пластинами, одна из которых покоится, а другая движется со скоростью, равной величине продольной составляющей линейной скорости вращения на периферии цилиндра (рис. 2). Скорость будет иметь направление, противоположное продольному предмету давления dp/dz .

В связи с этим необходимо определить профиль скорости на давление в конце области I, которые и будут являться исходными данными для анализа процесса фильтрации в области II.

Будем считать сырье в канале шнека несжимаемой, тогда выделив некий элементарный объем, сечение которого показано на рис. 3, действие объемного давления на него будет сказываться в том, что мятка деформируется с выделением некоторого количества масла. В итоге производной знание давления в конце области I будет и информация о массовой концентрации масла в сырье.

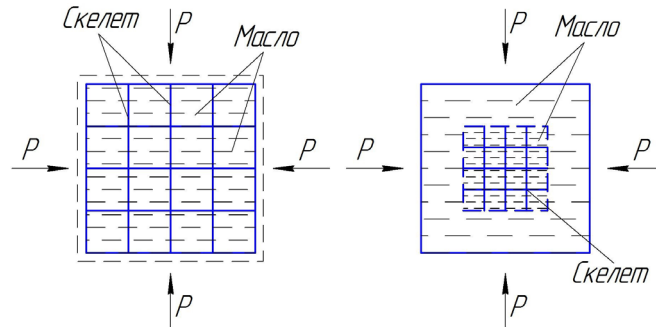


Рис. 3. Схема перераспределения масла в мятке

Эта информация должна быть достаточной для определения закономерностей фильтрации в области II, для которой возможно применение конвективно-диффузионных представлений для расчета кинетики трансформации поля концентрации.

Как показывает практика отжима, изменение температуры происходит на незначительную величину, то есть порядка 20 °С. Этот факт позволяет принять в области I режим изотермического течения. Учитывая таким, что скорости течения среды в канале шнека невысоки, то течение можно считать ламинарным.

Основная часть

В основе построения математических моделей гидродинамики при ламинарном изотермическом течении различных сред лежат уравнения, вытекающие из законов сохранения вещества и количества движения [1]:

$$\rho \cdot \frac{D\bar{v}}{Dt} = -\nabla p - [\nabla \cdot \tau] + \rho \bar{g}; \quad (1)$$

$$\frac{D\rho}{Dt} = -\rho(\nabla \cdot \bar{v}); \quad (2)$$

$$\nabla^2 p = -\rho \nabla \cdot (\bar{v} \nabla) \bar{v} - \nabla \cdot [\nabla \cdot \tau], \quad (3)$$

где ρ — плотность жидкости; \bar{v} — вектор скорости; p — давление; \bar{g} — вектор ускорения силы тяжести; τ — тензор напряжений; $D/Dt = \partial/\partial t + (\bar{v} \nabla)$ — оператор субстанциональной; ∇ — градиент скалярного поля.

Уравнения (1) и (2) в компонентном виде для несжимаемой среды в декартовой системе координат при условии отсутствия действие внешних объемных сил несущественным ($g_x = g_y = g_z = 0$), имеет вид:

$$\rho \left(\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} + v_z \frac{\partial v_x}{\partial z} \right) = -\frac{\partial p}{\partial x} + \eta \left\{ 2 \frac{\partial}{\partial x} \left(\varphi \frac{\partial v_x}{\partial x} \right) + \frac{\partial}{\partial z} \left[\varphi \left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z} \right) \right] \right\}; \quad (4)$$

$$\rho \left(\frac{\partial v_z}{\partial t} + v_x \frac{\partial v_z}{\partial x} + v_z \frac{\partial v_z}{\partial z} \right) = -\frac{\partial p}{\partial z} + \eta \left\{ \frac{\partial}{\partial x} \left[\varphi \left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z} \right) \right] + 2 \frac{\partial}{\partial z} \left(\varphi \frac{\partial v_z}{\partial z} \right) \right\}; \quad (5)$$

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z} = 0, \quad (6)$$

где

$$\varphi = \left[\left\{ 2 \left[\left(\frac{\partial v_x}{\partial x} \right)^2 + \left(\frac{\partial v_y}{\partial z} \right)^2 \right] + \left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z} \right)^2 \right\} / \dot{\gamma}_0^2 \right]^{\frac{n-1}{2}}. \quad (7)$$

где t, x, y, z — текущее время и локальные декартовы координаты; v_x, v_y, v_z — компоненты вектора скорости \bar{v} ; g_x, g_y, g_z — проекции вектора ускорения силы тяжести на оси декартовой системы координат; $\tau_{\alpha, \beta}$ — компоненты тензора напряжений τ ($\alpha = xVyVz, \beta = xVyVz$).

Решая уравнения (4)–(7) получим:

$$V(X) = \begin{cases} \frac{n}{1+n} \left(\frac{dP}{dZ} \right)^{\frac{1}{n}} \left[(X - X^*)^{\frac{1+n}{n}} - X^{*\frac{1+n}{n}} \right], & \text{при } 0 \leq X \leq X^*; \\ V_0 + \frac{n}{1+n} \left(\frac{dP}{dZ} \right)^{\frac{1}{n}} \left[(X - X^*)^{\frac{1+n}{n}} - (1 - X^*)^{\frac{1+n}{n}} \right], & \text{при } X^* \leq X \leq 1. \end{cases} \quad (8)$$

Найдем среднюю скорость движения в шнековом канале как:

$$\begin{aligned} \bar{V} &= \int_0^{X^*} V_1(X) dX + \int_{X^*}^1 V_2(X) dX = \\ &= V_0(1 - X^*) - \frac{n}{1+2n} \left(\frac{dP}{dZ} \right)^{\frac{1}{n}} X^{*\frac{1+2n}{n}} + \frac{n}{1+n} \left(\frac{dP}{dZ} \right)^{\frac{1}{n}} (1 - X^*)^{\frac{1+n}{n}} \left[\frac{1+3n}{1+2n} (1 - X^*) - 1 \right]. \end{aligned} \quad (9)$$

Заключение

Таким образом, в ходе математического моделирования были получены зависимости, которые легли в создание системы автоматизированного проектирования САПР «Маслопресс», позволяющей проводить расчет конструктивных параметров основных узлов прессующего оборудования.

Литература

- 1 Берд, Р. Явления переноса [Текст] / Р. Берд, В. Стьюарт, Е. Лайтфут. – М. : Химия, 1974. – 688 с.
- 2 Карташов, Л. П. Параметрический и структурный синтез технологических объектов на основе системного подхода и математического моделирования / Л. П. Карташов, Т. М. Зубкова. – Екатеринбург : УрО РАН, 2009. – 225 с.
- 3 Василенко, В. Н. Создание ресурсосберегающей технологии комплексной переработки масличного и зернового сырья / В. Н. Василенко, Е. А. Татаренков, М. В. Копылов, Е. В. Бабич // Материалы международной научно-практической конференции «Современные направления теоретических и прикладных исследований 2010». В 8т. Т. 5. – Одесса : Одесс. национ. морской унив, 2010. – С. 5–7.

Дюбин Михаил Викторович — студент 2-го курса ВУНЦ ВВС ВВА, факультет авиационное вооружение. E-mail: tea_86@mail.ru

Голиков Иван Петрович — студент 2-го курса ВУНЦ ВВС ВВА, факультет авиационное вооружение. E-mail: tea_86@mail.ru

МОДИФИКАЦИИ ПРИЛОЖЕНИЯ ПО ПРОВЕДЕНИЮ ИНТЕЛЛЕКТУАЛЬНЫХ ВИКТОРИН

И. Д. Коток

Воронежский государственный университет

Введение

Два тяжелых года под ужасной коронавирусной инфекцией показали человечеству, что будущее невозможно без интернета — технологий. Многие процессы, которые ранее и невозможно было представить, оказалось можно выполнять дистанционно из дома, без задействования офисов и специальных сооружений. Многие компании, проработав в такой системе два года с начала эпидемии в 2020 году, поняли, что дистанционная работа выгодна не только экономически, но и более продуктивна в некоторых моментах. Так, некоторые организации решили, даже после послабления коронавирусных ограничений, оставить часть своих сотрудников на удаленном режиме работы.

Студенческие объединения университета тоже получили подобный опыт, когда организация очных мероприятий фактически стала невозможной из-за сложной эпидемиологической обстановки в регионе. Они стали уходить в социальные сети и искать возможности общения со своей аудиторией в новом формате, более интенсивно. В 2020 году было написано приложение для проведения интеллектуальных викторин, его частично использовали при проведении своих мероприятий активисты студенческого совета факультета ПММ. Сейчас приложение находится на доработке. Планируется добавить видеосвязь между участниками игры, чтобы максимально приблизить их к реальной версии игры «Своя игра». Также, сейчас идет речь о том, чтобы добавить новые возможности для проведения ежегодных олимпиад по программированию для студентов первого курса воронежских университетов. Помимо этого, для организаторов, как турниров, так и олимпиад, интересен вопрос о переходе на более актуальную версию .NET Framework, которая позволит расширить возможности программного продукта [1–2].

1. Диаграмма прецедентов

Были разработаны диаграммы прецедентов для предыдущей версии приложения, которые продемонстрированы далее. Стоит отметить, что доработка этих диаграмм будет являться основополагающей частью дальнейшей доработки программного продукта [3–4].

Данная диаграмма помогает разобраться, как правильно и точно распределить обязанности между различными пользователями разрабатываемого приложения, чтобы облегчить и ускорить решение поставленной задачи на этапе разработки.

Для данного программного обеспечения были взяты следующие сценарии для двух акторов, один из них является авторизованный пользователь нашего приложения, другой администратор этого продукта.

Для актора «Пользователь» был составлен сценарий, который продемонстрирован на рис. 1. Актор имеет следующие основные прецеденты:

- участие в турнире, который включает прецедент регистрация команды;
- работа с личным кабинетом;
- создание пакета вопросов;

- отправка вопроса;
- просмотр информации ресурса, который, в свою очередь, включил в себя просмотр новостной ленты и просмотр одобренных вопросов и пакетов.



Рис. 1. Диаграмма прецедентов для актора «Пользователь»

Для актора «Администратор» был составлен похожий сценарий, но обладающий большим количеством возможностей. Данная диаграмма представлена на рис. 2.

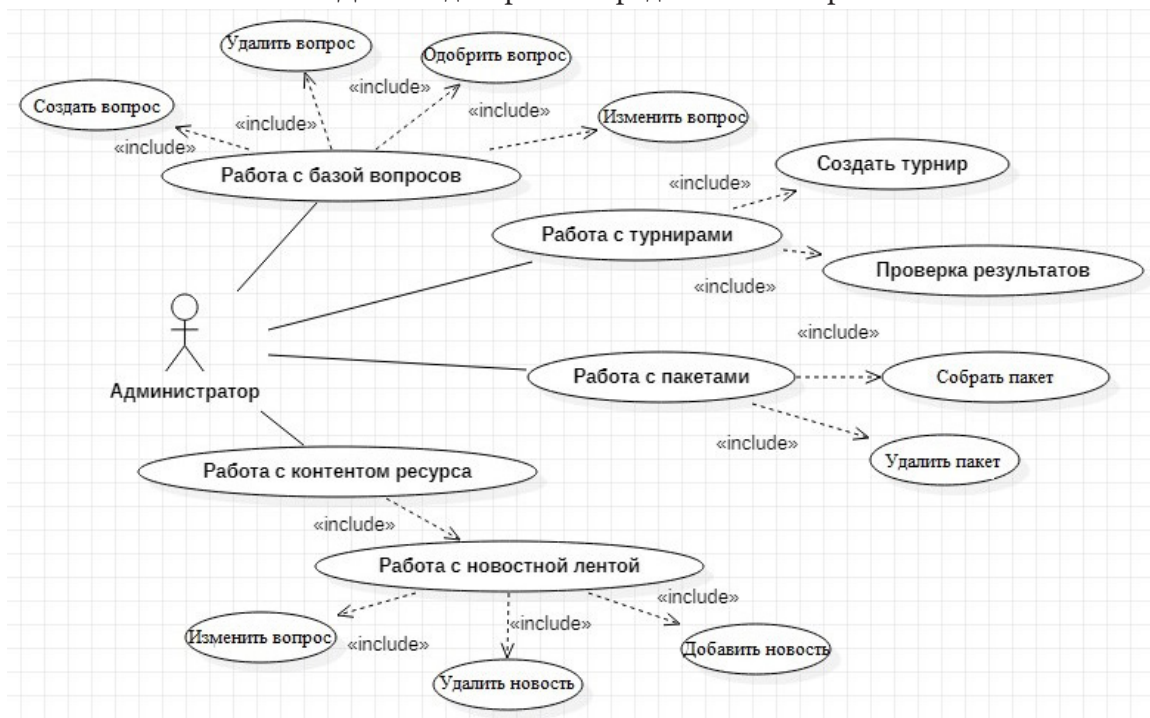


Рис. 2. Диаграмма прецедентов для актора «Администратор»

Актор имеет следующие прецеденты:

- работа с базой вопросов, который, в свою очередь, включает в себя прецеденты для взаимодействия с вопросами;
- работа с контентом ресурса, который включает в себя работу с новостной лентой. Этот прецедент имеет включения для работы с публикацией новостей;

- работа с пакетами, который также имеет включения;
- прецедент, который отвечает за работу с турнирами, имеющий два включения для проверки результатов и создания нового турнира.

Диаграммы для новой части приложения, которая будет включать в себя работу с олимпиадами, будет иметь схожую структуру, отличаться наличием возможностей работы с базой олимпиады.

Заключение

По результатам работы за 2021 год были определены основные моменты доработки приложения, оценены риски при его разработке, решено какими функциональными возможностями оно должно обладать по окончании разработки данного программного продукта [5–6]. Сформирована группа студентов активистов, которой интересно потенциальное развитие проекта, его продвижение. Также хотелось бы отметить новую особенность приложения, такую, как проведение олимпиад онлайн, которое поможет облегчить работу организационного комитета. Помимо этого, приложение позволит вести статистику по участникам и начать заниматься их классификацией.

Также было решено, что приложение будет переработано с использованием более современного фреймворка для создания веб-приложений на платформе .NET — ASP.NET Core. Это поможет облегчить и упростить разработку приложения в дальнейшем, а также позволит расширить его функциональные возможности, используя все современные возможности платформы.

Литература

1. *Албахари Джозеф, Албахари Бен. С# 7.0 Справочник. Полное описание языка.: Пер. с англ. – СПб. : ООО «Диалектика», 2019. – 1024с.*
2. .NET Documentation | Microsoft Docs. – Режим доступа: <https://docs.microsoft.com/en-us/dotnet/>
3. *Коток И. Д. Разработка программного обеспечения для проведения интеллектуальных компьютерных игр студенческим советом факультета прикладной математики, информатики и механики // Актуальные проблемы прикладной математики, информатики и механики: сб. тр. Междунар. науч.-техн. конф. (Воронеж, 11–13 ноября 2019 г.). – Воронеж: Издательство «Научно-исследовательские публикации», 2020. – С. 443–444.*
4. *Коток И. Д. Разработка программного обеспечения для проведения интеллектуальных викторин // Информационные технологии в образовательном процессе вуза и школы: материалы XIV Всероссийской научно-практической конференции (Воронеж, 25 марта 2020 г.). – Воронеж: Воронежский государственный педагогический университет, 2020. – С. 171–175.*
5. *Коток И. Д. Интернет-технологии в активизации работы студенческого совета факультета прикладной математики, информатики и механики // Математика, информационные технологии, приложения: сборник трудов Межвузовской научной конференции молодых ученых и студентов, Воронеж, 27 апреля 2020 г. – Воронеж : Научная книга, 2020. – С. 123–126.*
6. *Коток И. Д. Разработка программного обеспечения для работы студенческого объединения в сложной эпидемиологической обстановке / Актуальные проблемы прикладной математики, информатики и механики: сб. тр. Междунар. науч.-техн. конф. (Воронеж, 7–9 декабря 2020 г.). – Воронеж : Издательство «Научно-исследовательские публикации», 2021. – С. 545–551.*

Коток Игорь Дмитриевич — магистрант 2 курса кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики Воронежского государственного университета. E-mail: tigariskot@gmail.com.

Ускова Ольга Фёдоровна (научный руководитель) — канд. техн. наук, профессор кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики Воронежского государственного университета. E-mail: nat-uskova@mail.ru

ЭФФЕКТИВНОСТЬ ПРИМЕНЕНИЯ НЕЙРОННОЙ СЕТИ В ЗАДАЧЕ КЛАССИФИКАЦИИ КРОВОПОДТЁКОВ

В.Д. Крыжановский

Воронежский государственный университет

Введение

Судебная экспертиза — это сложная система разнородных научных оснований, методов и методик проведения исследований, включающая в себя также нормативно-правовое регулирование. На определённом этапе развития технологий судебной экспертизы возникла потребность усовершенствовать инструментарий профильных специалистов, в частности, посредством разработки и внедрения различных информационных и экспертных систем [1].

В современной практике в качестве инструмента установления реальной картины произошедших событий и прогнозирования вероятных действий при заданных условиях широко используются вычислительные устройства, специализированный математический аппарат и программное обеспечение. В частности, исторически первые успешные попытки создания информационных систем лежат в области почерковедения, портретно-идентификационных исследованиях, судебно-автотехнической экспертизы, предназначенной для прогнозирования возможности уклонения от ДТП, наезда на пешеходов или иное внезапно возникшее препятствие, а также позволяющих рассчитать скорость и траекторию движения транспортного средства.

Исходные данные для информационных и экспертных систем формируются соответствующими специалистами на основании материалов дела, натуральных коллекций и описательных фондов, а также иных источников. Результаты работы информационных и экспертных систем изучаются, оцениваются и, в случае достоверности, заверяются подписью эксперта. В случае некоторых экспертиз допустимо применение квалифицированной электронной подписи (КЭП).

При применении информационных и экспертных систем в области судебно-медицинской экспертизы можно выделить три основных пути:

- автоматизация исследования вещественных доказательств и иных материалов и документов, представляющих ценность в контексте расследования правонарушения;
- создание диалоговых систем;
- математическое моделирование элементов экспертного исследования.

Одной из задач, стоящих перед центрами судебно-медицинской экспертизы (СМЭ) является определение давности получения травм потерпевшими. На текущий момент нет информационных и экспертных систем подобного рода, получивших широкое распространение.

1. Постановка задачи

Требуется разработать модуль экспертной системы для классификации кровоподтёков, подобных приведённым на фотографии, по четырём категориям давности: одни сутки (рис. 1), двое–пятеро (рис. 2), пять–десять (рис. 3) и десять–пятнадцать суток (рис. 4). Обычно документальная фиксация предполагает использование линейки эксперта: рассматриваемый объект, в нашем случае, кровоподтёк, должен быть расположен рядом с ней. В силу необходимости автоматического определения области расположения повреждения для определения координат объекта распознавания принято решение использовать нейронную сеть [2–4].

Входными данными системы являются фотографии. Изначальный набор изображений разделён на тестовую и обучающую выборку в отношении один к четырём. Обученная нейронная сеть на входе принимает изображение, а на выходе делает оценку вероятности принадлежности изображения к каждому из пяти классов.



Рис. 1. Кровоподтёк суточной давности



Рис. 2. Кровоподтёк, полученный 2–5 суток назад



Рис. 3. Кровоподтёк, полученный 5–10 суток назад



Рис. 4. Кровоподтёк, полученный 10–15 суток назад

2. Выбор технологий

Изначально предполагалось, что модуль будет только определять цвет раны, и на основании этого будет выноситься предварительный вывод о сроке, прошедшем с момента получения повреждения. Одна из технологий базируется на использовании библиотеки алгоритмов компьютерного зрения OpenCV после решения задачи детекции с помощью нейронной сети, обработке её гауссовым размытием и, с использованием метод k-средних, определения цвета в формате RGB с последующей классификацией. По мере уточнения требований к системе от этой технологии пришлось отказаться в силу того, что это не обеспечивало достаточной точности распознавания из-за различия в цветах кожи.

Для распознавания изображений была выбрана библиотека TensorFlow [1], API Keras и модель нейронной сети Sequential.

TensorFlow — это библиотека с открытым исходным кодом, созданная для Python командой Google Brain. TensorFlow объединяет множество различных алгоритмов и моделей воедино, позволяя пользователю реализовать глубокие нейронные сети для использования в таких задачах, как распознавание и классификация изображений, а также обработка естественного языка.

Keras — это высокоуровневый API (интерфейс прикладного программирования), который может использовать функции TensorFlow. Keras был разработан с учетом удобства использования и модульности в качестве своих руководящих принципов. С практической точки зрения Keras делает реализацию многих мощных функций TensorFlow максимально простой, и он настроен на работу с Python без каких-либо серьезных модификаций или конфигураций.

Sequential — модель нейронной сети, предполагающая последовательное расположение слоёв в линейном стеке.

Полносвязные нейронные сети получают входные данные и преобразовывают их, пропуская через набор скрытых слоёв. Каждый слой состоит из определённого количества нейронов, каждый из которых связан со всеми нейронами предыдущего слоя и не имеет связей с другими нейронами своего слоя. Последний полносвязный слой называется выходным слоем и в задачах классификации представляет собой распределение оценок по классам.

Обычные нейронные сети плохо масштабируются для больших изображений. Например, в наборе данных CIFAR-10, изображения размером $32 \times 32 \times 3$ (32 пикселя высота, 32 пикселя ширина и 3 цветовых канала) для обработки изображений каждый нейрон в первом скрытом слое полносвязной нейронной сети должен иметь $32 \times 32 \times 3 = 3072$ веса. Это значение не слишком велико, однако вполне очевидно, что полносвязные слои потребуют слишком больших затрат для больших изображений. Например, изображение $256 \times 256 \times 3$, приведёт к тому, что количество весов станет равным $256 \times 256 \times 3 = 196\,608$. Более того, нам понадобится не один подобный нейрон, поэтому общее количество весов быстро начнёт расти. Таким образом, можно сделать вывод, что полносвязность для обработки изображений чрезмерна, кроме того, большое количество параметров быстро приведут сеть к переобучению.

Свёрточные нейронные сети используют тот факт, что входные данные — это изображения, что позволяет образовать более чувствительную архитектуру к подобному типу данных. В частности, в отличие от обычных нейронных сетей, слои в свёрточной нейронной сети располагают нейроны в 3 измерениях — ширине, высоте, глубине. Например, входные изображения из набора данных CIFAR-10 являются входными данными в 3D-представлении, размерность которых равна $32 \times 32 \times 3$ (ширина, высота, глубина). В случае свёрточной нейронной сети, нейроны в одном слое будут связаны с небольшим количеством нейронов в предыдущем слое, вместо того чтобы быть связанными со всеми предыдущими нейронами слоя.

На левой стороне рис. 3 изображена стандартная 3-х слойная нейронная сеть. На правой стороне рис. 3 изображена свёрточная нейронная сеть располагает свои нейроны в 3-х измерениях (ширине, высоте, глубине), как это показано на одном из слоёв. Каждый слой свёрточной нейронной сети преобразует 3D-представление входных данных в 3D-представление выходных данных в виде нейронов активации. В этом примере красный входной слой содержит изображение, поэтому его размеры будут равны размерам изображения, а глубина будет равна 3 (три канала — red, green, blue).

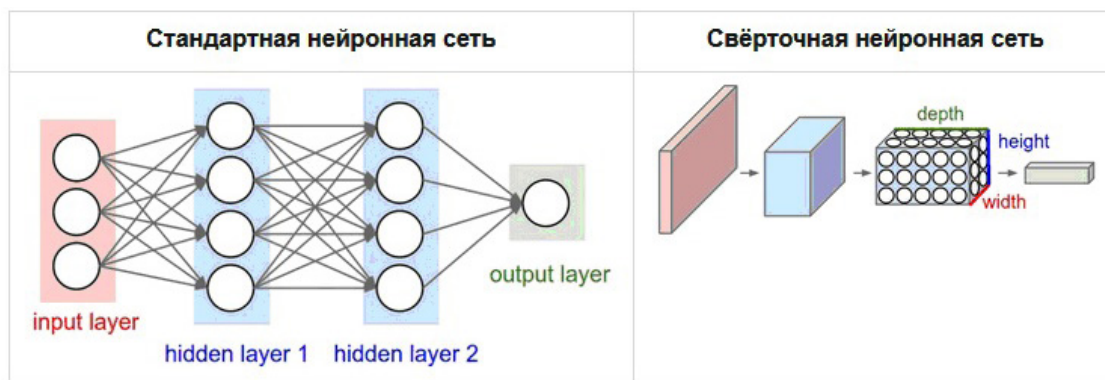


Рис. 5. Визуализация сравнения сетей

3. Разработка алгоритма решения задачи

Предварительно была произведена операция аугментации [2] имеющихся данных для увеличения количества изображений, в частности, посредством операции отражения, поворота и растяжения. Потеря «обрезанных» краёв изображения в данном случае не критична, поскольку важная информация извлекается из центральной части изображения. Кроме того, исходный датасет имеет высокое разрешение, что негативно скажется на процессе обучения нейронной сети. Данная проблема была решена посредством сжатия изображений.

Разметка нужных объектов — в нашем случае кровоподтеков — проводилась с помощью свободно распространяемой программы с открытым исходным кодом LabelImg.

В качестве ядра свёртки было выбрано (3,3), как ядро для операции макспулинга — (2,2). В модель нейронной сети добавлены свёрточные слои Conv2D с функцией активации ReLu, содержащий 32 нейрона, слой для операции макспулинга и слой Dropout с аргументом 0,25 для решения проблемы переобучения. За ними располагается слой Conv2D с функцией активации ReLu, содержащий 64 нейрона, ещё один слой MaxPool2D (для макспулинга) и Dropout. После этого полученные данные выравниваются слоем Flatten и передаются на слой Dense из 256 нейронов с функцией активации ReLu, за которым расположен ещё один слой Dense из четырёх нейронов с функцией активации softmax. Обучение проводилось за 100 эпох, отношение тестовых изображений к обучающим выбрано как один к четырём. Как функция потерь выбрана функция категориальной перекрёстной энтропии, параметр оптимизации — стохастический градиентный спуск, а метрика — «ассигасу».

Таблица 1

Характеристики слоёв нейронной сети

Слой	Выходная форма	Количество весов
Conv2D	(None, 254, 254, 32)	896
MaxPooling2D	(None, 127, 127, 32)	0
Dropout	(None, 127, 127, 32)	0
Conv2D	(None, 125, 125, 64)	18496
MaxPooling2D	(None, 62, 62, 64)	0
Dropout	(None, 62, 62, 64)	0
Flatten	(None, 246016)	0
Dense	(None, 256)	62980352
Dense	(None, 5)	1285

Заключение

Разработан модуль экспертной системы СМЭ для классификации кровоподтёков в зависимости от давности нанесения травмы. Эффективность работы полученной сети близка к 90 %.

Литература

1. Проблемы информационного и математического обеспечения экспертных исследований в целях решения задач судебной экспертизы / под ред. Н. С. Полевого. – М., 2000. – 159 с.
2. Организация и использование нейронных сетей (методы и технологии) / Под общ. ред. В.Б. Новосельцева. – Томск : Изд-во НТЛ, 2006. – 128 с.
3. Круглов, В. В. Гибридные нейронные сети / В. В. Круглов, В. В. Борисов. – Москва : 2005. – 137 с.

4. Уоссермен, Ф. Нейрокомпьютерная техника: теория и практика / Ф. Уоссермен. – Москва : Мир, 1992. – 380 с.

5. Документация Keras [Электронный ресурс] : база данных. – Режим доступа : <https://keras.io/guides/>

6. Глубокое обучение в биологии и медицине : издание / Б. Рамсундар [и др.] ; пер. с англ. В. С. Яценкова. – Москва : LVR Пресс, 2020. – 200 с.

Крыжановский Владислав Дмитриевич — магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета.

E-mail: vlad0kryzh@gmail.com

Горбенко Олег Данилович (научный руководитель) — канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.

АНАЛИЗ АУДИОДАНЫХ. РАСПОЗНАВАНИЕ МЕЛОДИИ ПРИ ПОМОЩИ ИНС-МОДЕЛИ ОБРАБОТКИ ДАННЫХ

М. Ю. Лихачев, И. В. Замятин

Воронежский государственный университет

Введение

Музыка — важная часть жизни любого человека, одно из древних видов искусства. В настоящее время она продолжает развиваться, в том числе при помощи современных технологий. В сети Интернет размещено большое количество онлайн-приложений и платформ, предназначенных для работы с оцифрованной звуковой информацией (музыкой). В данную категорию входят такие типы систем, как музыкальные проигрыватели (iTunes, Google Play Music, Яндекс.Музыка, Spotify), системы распознавания и поиска музыки (Shazam), а также платформы для распространения пользовательской авторской музыки (SoundCloud, Bandcamp).

Цель настоящей работы: разработать, смоделировать и спроектировать модель решения распознавания мелодии.

С этой целью были решены следующие задачи:

1. Проведен анализ аудиопризнаков, выделены и изучены наиболее значимые и подходящие аудиопризнаки.
2. Спроектирована модель решения задачи распознавания мелодии.
3. Реализована тестовая модель решения задачи распознавания мелодии при помощи искусственной нейронной сетевой (ИНС) модели обработки данных.

Объектом работы является изучение и анализ аудиопризнаков и решение задачи распознавания мелодии.

Предмет работы: аудиоанализ и тестовая искусственная нейронная сетевая (ИНС) модель для обработки изученных данных.

Положения, выносимые на защиту:

1. Аудиоанализ.
2. Тестовая ИНС-модель распознавания мелодии.

1. Анализ предметной области

1.1. Аудиоанализ

Аудиоанализ — область, включающая автоматическое распознавание речи (ASR), цифровую обработку сигналов, а также классификацию, тегирование и генерацию музыки — представляет собой развивающийся поддомен приложений глубокого обучения [1].

1.2. Музыка

Мелодия — последовательность музыкальных звуков (нот), принадлежащих к какой-нибудь гамме или ладу (гармонии) — определенного набора нот.

Нота — графическое обозначение музыкальных звуков. Также используется как синоним к музыкальным звукам. Является одним из основных символов современной музыкальной нотации. Вариации в изображении ноты, в сочетании с другими символами позволяют показать такие характеристики звука как: высоту, длительность и порядок исполнения по отношению к другим звукам.

2. Аудиоанализ

2.1. Обработка цифрового аудиосигнала

Звуковые волны оцифровываются путем выборки из дискретных интервалов, известных как частота дискретизации (как правило, 44,1 кГц для аудио с CD-качеством, то есть 44 100 семплов в секунду). Каждый семпл представляет собой амплитуду волны в определенном временном интервале, где глубина в битах (или динамический диапазон сигнала) определяет, насколько детализированным будет семпл (обычно 16 бит, т.е. семпл может варьироваться от 65 536 значений амплитуды).

Семплинг — преобразование непрерывного сигнала в серию дискретных значений (Audio Times Series).

Частота дискретизации (Sample Rate) — это количество семплов за определенный фиксированный промежуток времени. Высокая частота дискретизации приводит к меньшей потере информации, но к большим вычислительным затратам.

Звук представлен в форме аудиосигнала с такими параметрами, как частота, полоса пропускания, децибел и т. д. Типичный аудиосигнал можно выразить в качестве функции амплитуды и времени.

Некоторые устройства могут улавливать эти звуки и представлять их в машиночитаемом формате. Процесс обработки звука включает извлечение акустических характеристик, относящихся к поставленной задаче, за которыми следуют схемы принятия решений, которые включают обнаружение, классификацию и объединение знаний.

2.2. Аудиопризнаки

Спектрограмма — визуальный способ представления уровня или громкости сигнала во времени на различных частотах, присутствующих в форме волны. Обычно изображается в виде тепловой карты. Данные преобразуются в кратковременное преобразование Фурье. С его помощью можно определить амплитуду различных частот, воспроизводимых в данный момент времени аудиосигнала. Каждый аудиосигнал состоит из множества признаков.

Спектральные (частотные) признаки получаются путем преобразования временного сигнала в частотную область с помощью преобразования Фурье. К ним относятся частота основного тона, частотные компоненты, спектральный центроид, спектральный поток, спектральная плотность, спектральный спад и т. д.

Спектральный центроид (Spectral Centroid) указывает, на какой частоте сосредоточена энергия спектра или, другими словами, указывает, где расположен «центр масс» для звука. Схож со средневзвешенным значением:

$$f_c = \frac{\sum_k S(k)f(k)}{\sum_k S(k)},$$

где $S(k)$ — спектральная величина элемента разрешения k , а $f(k)$ — частота элемента k .

Спектральный спад (Spectral Roloff) — мера формы сигнала, представляющая собой частоту, в которой высокие частоты снижаются до 0. Чтобы получить ее, нужно рассчитать долю элементов в спектре мощности, где 85 % ее мощности находится на более низких частотах.

Мел-частотные спектральные коэффициенты (MFCC) — представляют собой небольшой набор признаков (обычно около 10–20), которые кратко описывают общую форму спектральной огибающей. Они моделируют характеристики человеческого голоса.

Цветность (признак или вектор цветности, хромаграмма) (Chroma) обычно представлен вектором признаков из 12 элементов, в котором указано количество энергии каждого высотного класса (Pitch class) {C, C#, D, D#, E, F, F#, G, G#, A, A#, B} в сигнале. Используется для описания меры сходства между музыкальными произведениями. [1]

3. Проектирование модели решения

Для решения задачи распознавания мелодий необходимо сгенерировать данные. Полученные данные необходимо сохранить с аудиофайлах, из которых будут определенным образом получены данные для ИНС-модели. После обработки данных необходимо будет обработать полученный результат. Таким образом, решение задачи разделено на следующие этапы (иллюстрация приведена в соответствии с рис. 1):

1. Генерация тестовых данных (Data Generating);
2. Предварительная обработка (Preprocessing);
3. Обработка при помощи ИНС-модели (ANN Processing);
4. Обработка полученного ИНС-моделью результата (Postprocessing).

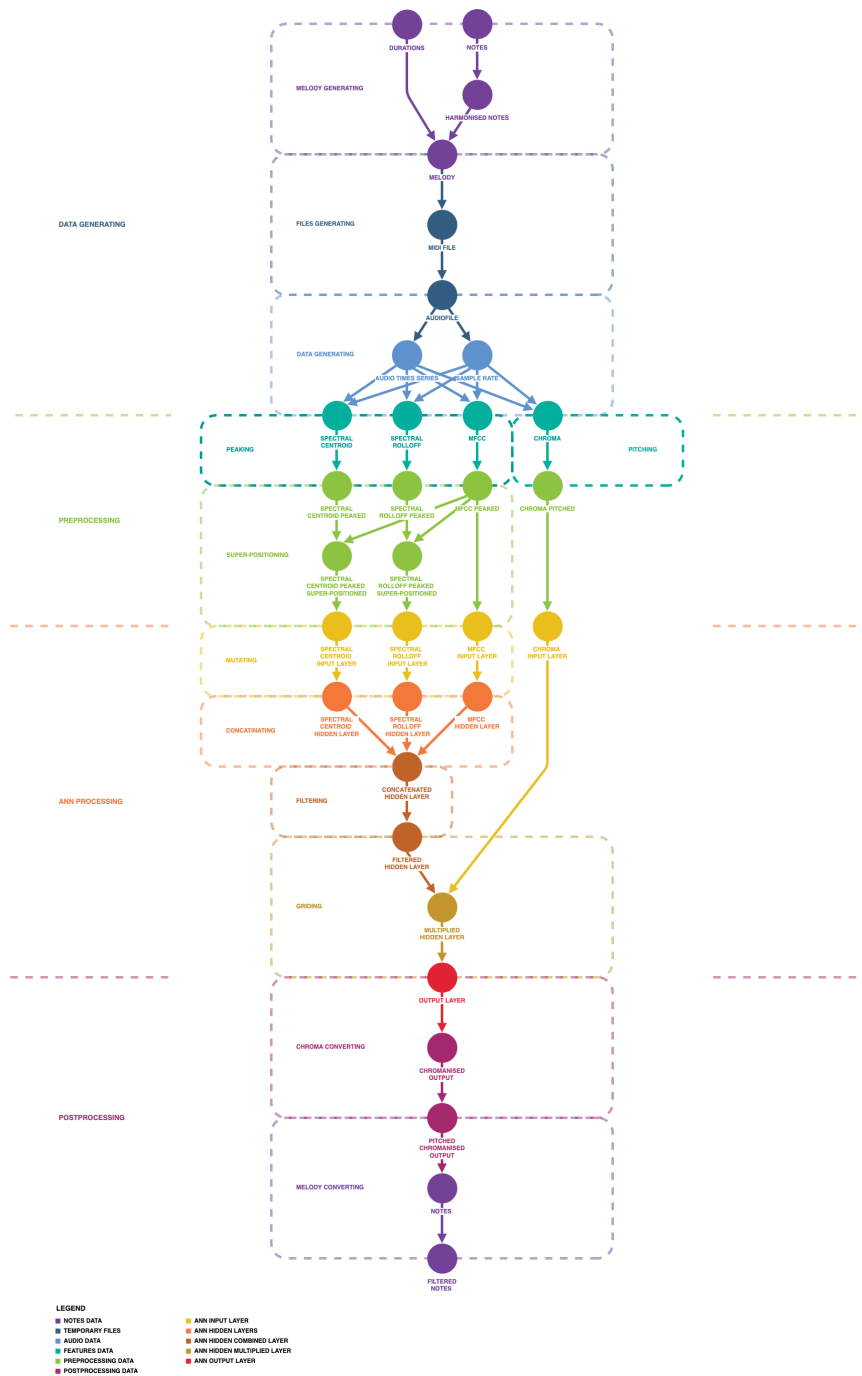


Рис 1. Общая модель обработки данных

3.1. Генерация тестовых данных

Состоит из следующих этапов (иллюстрация приведена в соответствии с рис. 2):

1. Генерация мелодий (Melody Generating) — генерация набора высот нот (Notes), приведение их в гармонию (Harmonised Notes), и длительностей (Durations), и объединение гармонизированных нот и длительностей в мелодий (Melody);
2. Генерация файлов (Files Generating) — генерация MIDI-файлов из мелодии и последующая его конвертация в аудиофайлы (WAV-файл) при помощи виртуального синтезатора;
3. Получение аудиоданных из аудиофайла (Data Generating) — чтение аудиоданных (Audio Times Series) и частоты дискретизации (Sample Rate), и получение вышеописанных аудиопризнаков (Features Data).

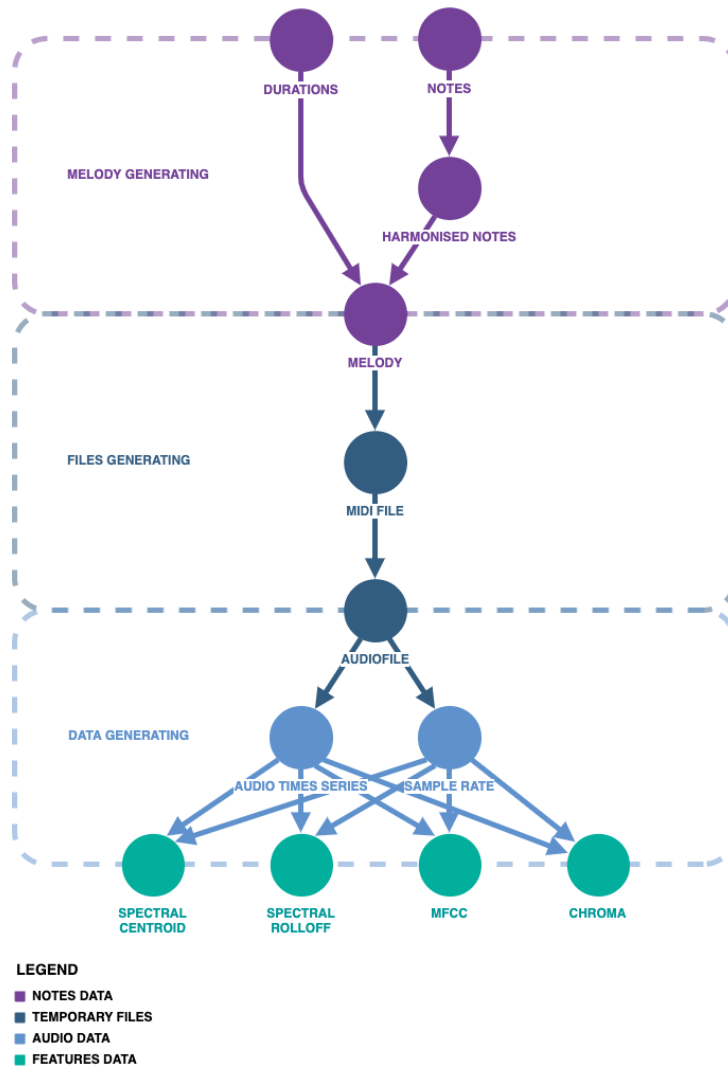


Рис 2. Модель генерации данных

3.2. Предварительная обработка данных

Состоит из следующих этапов (иллюстрация приведена в соответствии с рис. 3):

1. Получение локальных максимумов спектральных и мел-частотных характеристик (Peaking) и получение класса высот из хромограммы с наибольшей энергией (Pitching);
2. Синхронизация спектральных и мел-частотных локальных максимумов данных путем сдвига спектральных относительно мел-частотных (Super-positioning).

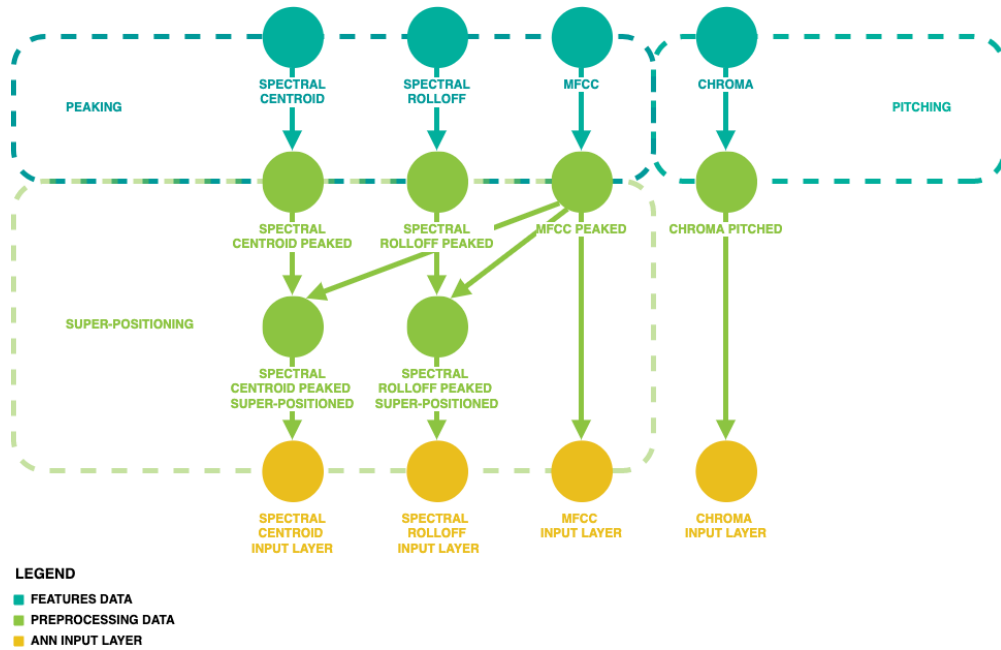


Рис 3. Модель предварительной обработки данных

3.3. Обработка данных при помощи ИНС-модели

Состоит из следующих этапов (иллюстрация приведена в соответствии с рис. 4):

1. Искажение входных данных (Mutating) — обработка данных слоем входных нейронов и слоев линейных нейронов [2];

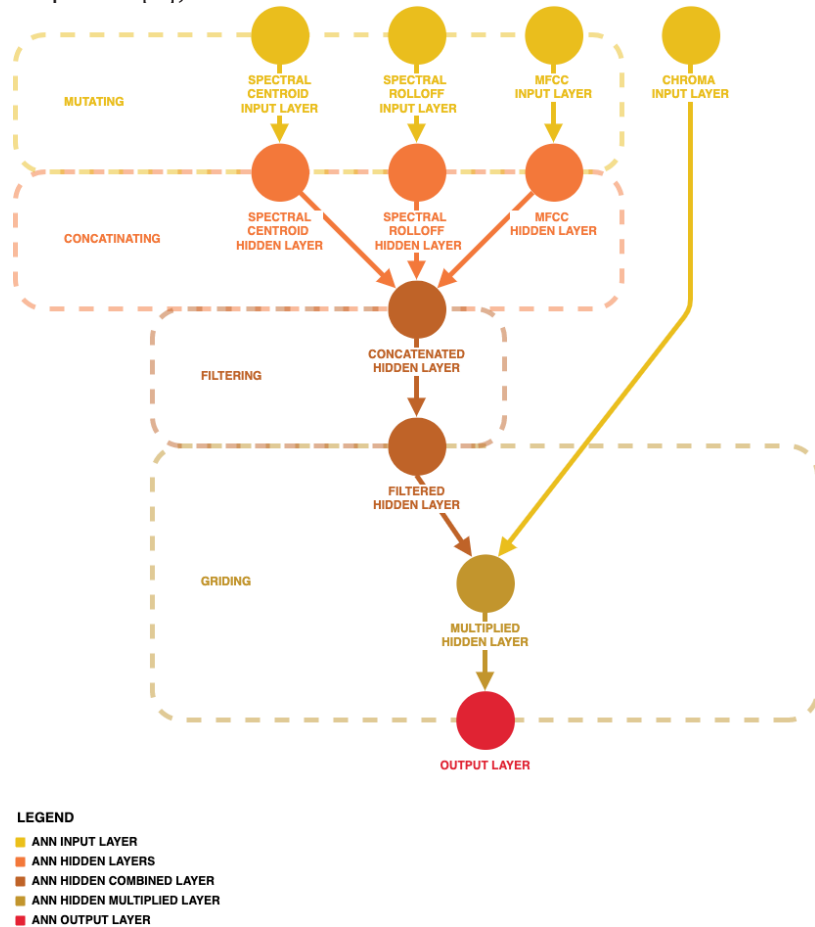


Рис 4. Модель обработки данных при помощи ИНС-модели

2. Объединение обработанных спектральных и мел-частотных характеристик (Concatinating) — суммирование и объединение всех вышеописанных характеристик;
3. Фильтрация данных (Filtering) — приведение данных в бинарный вид и отсеивание значений ниже среднего при помощи слоя линейных нейронов с бинарной ступенчатой функцией активации [3];
4. Наложение полученной в результате конкатенации «сетки» спектральных и мел-частотных характеристик на предварительно обработанный вектор цветности (Gridding).

3.4. Обработка данных, полученных в результате работы ИНС-модели

Состоит из следующих этапов (иллюстрация приведена в соответствии с рис. 5):

1. Построение хромаграммы из полученных ИНС-моделью данных (Chroma Converting);
2. Получение класса высот из хромаграммы, преобразование их в последовательность нот, и фильтрация идущих подряд нот (Melody Converting).

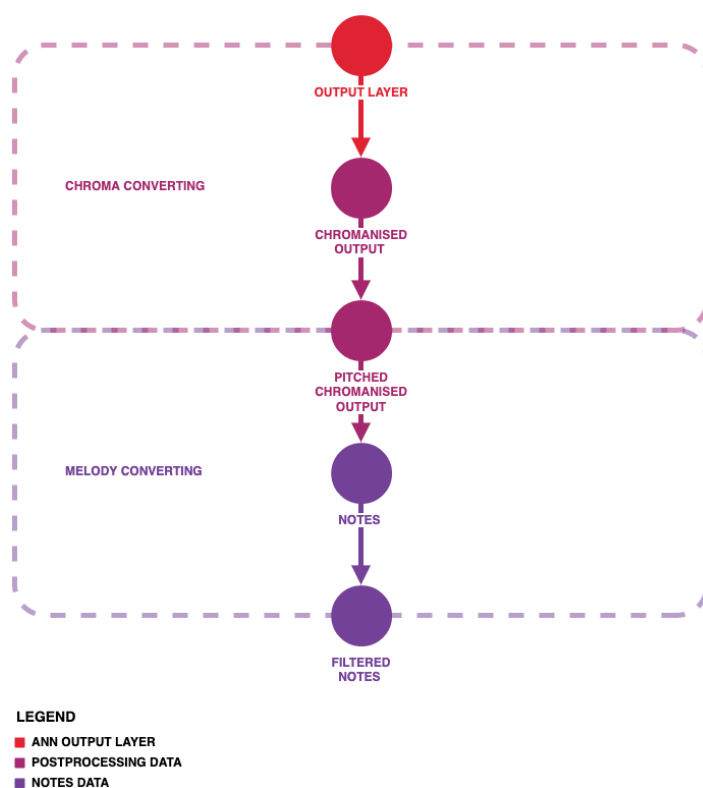


Рис 5. Модель обработки результатов ИНС-модели

4. Анализ результатов тестирования модели решения

В результате проведения тестирования модели решения на автоматически сгенерированных данных было установлено, что средняя метрика сравнения полученных наборов нот меняется в пределах 82 % – 85 % при средней метрики сравнения размера данных наборов нот в пределах 84 % – 89 %. Данный результат свидетельствует о хорошей предсказательной способности полученной модели обработки данных, и неточность объясняется тем, что в результате обработок теряется примерно 11 % – 16 % изначально исполненных нот.

Заключение

Результатом исследования стали анализ аудиоданных и реализация тестовой ИНС-модели для разработанного решения задачи распознавания мелодии. На основе данного анализа изучены методы, особенности, преимущества и проблематика полученного решения вышеописанной задачи на основе реализованной ИНС-модели.

Литература

1. Nagesh Singh Chauhan: Audio Data Analysis Using Deep Learning with Python (Part 1). – Режим доступа: <https://levelup.gitconnected.com/audio-data-analysis-using-deep-learning-part-1-7f6e08803f60>. – (Дата обращения: 19.04.2022).
2. *Rashid, T.* Make Your Own Neural Network: A Gentle Journey Through the Mathematics of Neural Networks, and Making Your Own Using the Python Computer Language / T. Rashid. – California : CreateSpace Independent Publishing Platform, 2016. – 222 с.
3. *Haykin, S. S.* Neural Networks: A Comprehensive Foundation / S. S. Haykin. – 2-nd ed. – New Jersey : Prentice Hall, 2016. – 906 с.

Лихачев Максим Юрьевич — магистрант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: max.likhachev@icloud.com

Замятин Игорь Викторович (научный руководитель) — канд. физ.-мат. наук, доцент кафедры математических методов исследования операций Воронежского государственного университета.

ПОСТРОЕНИЕ ИНТЕГРАЦИОННОЙ МОДЕЛИ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

М. А. Ломакин

Воронежский государственный университет

Введение

В современных проектах, особенно в крупных компаниях, вопрос регрессионного и санитарного тестирования (от английского *sanity testing* — тестирование работоспособности) часто сводится к созданию огромного списка автоматических тестов. И вопрос анализа данного списка как никогда становится актуальным, ведь сам процесс может занимать значительное количество времени и вероятность создания повторных дефектов тоже является высокой.

Еще одной особенностью работы с крупными проектами, особенно на международном уровне, является производственное тестирование, часто на окружениях заказчика. В большинстве случаев анализ таких ошибок может занимать колоссальное количество времени, в основном из-за того, что доступ к данным реальных пользователей находится под множеством проверок подлинности личности. В таких условиях затраты на обработку дубликатов или «не ошибок» очень велики, поэтому лучше иметь способ оценивания вероятности данного события.

Отсюда и вытекает необходимость в построении интеграционной модели для автоматизированного тестирования ошибок в рамках регрессионного и санитарного тестирования на окружениях заказчика с использованием нейронных сетей.

1. Практическая значимость

Практическая значимость проведенного исследования состоит в изучении тенденций рынка в области автоматизированного тестирования ошибок, выявления их преимуществ и недостатков, а также анализ на возможные непокрытые области тестирования.

В ходе работы рассмотрены существующие аналоги систем для автоматизированного тестирования ошибок, созданных в ходе запуска регрессионных автотестов, выявлены и проанализированы их преимущества и недостатки.

В результате данного исследования сформулирована принципиально новая модель учитывающая недостатки своих конкурентов на рынке, а так же дающая возможность идти в новом направлении — производственном тестировании. Главной целью данной модели является минимизация однотипной работы инженера по контролю качества в выявлении дубликатов ошибок. А также в отказе созданных ошибок из-за проходящих в момент тестирования работ с окружением, ошибок в тестовых данных и т. п.

2. Обзор существующих аналогов

На данный момент на рынке представлено всего пара решений в данной области. Одной из них является *Unified Logfile Analyzer* (сокращенно *UFL*) — система разработанная компанией «Сбербанк» в противовес существующему на тот момент решению *ReportPortal*.

Из преимуществ можно выделить, что оба продукта предоставляют богатую информацию об автотестах, такие как тексты ошибок, скриншоты, файлы логирования, и информация из

базы данных, что значительно улучшает качество анализа. Также отдельное внимание уделено отчетности и статистике.

Из различий стоит сказать, что решения используют различные алгоритмы анализа файлов логирования, а так же различные интеграции с базами данных, для UFL — Oracle, а для ReportPortal — MongoDB.

Проанализировав представленные на рынке аналоги, и исходя из потребностей, возникающих при работе с производственным тестированием, можно сделать вывод о необходимости построения модели, способной подключаться к окружению заказчика, с возможностью интеграции в распространенные системы отслеживания ошибок (HP ALM, Jira и т.п.). Данная модель с помощью нейронных сетей будет оценивать вероятность того, является ли ошибка дубликатом, ошибкой окружения или ошибкой тест данных.

3. Описание интеграционной модели

Исходя из представленных выше требований, описываемую модель можно будет представить в виде множества модулей, каждый из которых будет отвечать за отдельную потребность. Такой вид модели позволит упростить поддержку, обновление и ее повторное использование на разных проектах.

Таким образом, можно выделить следующие модули:

1. Модуль подключения к окружению заказчика;
2. Модуль интеграции с системой отслеживания ошибок;
3. Модуль выгрузки информации об ошибках (файлы логирования, обфусцированные данные из базы данных и т.д.);
4. Модуль текстового анализа;
5. Модуль анализа состояния окружения;
6. Модуль анализа схожести ошибок;
7. Модуль анализа данных;
8. Результирующий модуль.

Таким образом, мы получили абстрактную модель, реализация которой может позволить сэкономить большое количество часов поиска и анализа однотипных ошибок, что позволит упростить работу с производственными ошибками.

Заключение

В ходе проделанной работы был проведен анализ рынка приложений для автоматизированного тестирования ошибок, были выявлены преимущества и недостатки представленных аналогов, которые в последствии помогли сформулировать принципиально новую интеграционную модель автоматизированного тестирования с использованием нейронных сетей. Данный продукт позволит сэкономить время и силы при работе с производственными ошибками.

В связи с тем, что данная модель является колоссальной, но в тоже время модульной, было принято решение, что в будущем она будет разрабатываться по нисходящей методологии проектирования, где результаты нижестоящих элементов будут заменены на финальные значения.

Таким образом, в будущем планируется разработать и обучить модель нейронной сети для оценки ошибок на основе таких внешних факторов как: состояние окружения, наличие ошибок в файлах логирования, состояния данных и существовании похожих ошибок.

В финальной версии приложения количество аргументов, а также модулей, может измениться исходя из потребностей при тестировании продукта заказчика, но основные тенденции будут оставаться неизменными — необходимость в упрощении и ускорении работы с производственными ошибками.

Литература

1. *Аксенов, С. В.* Организация и использование нейронных сетей (методы и технологии) / С. В. Аксенов, под ред. В. Б. Новосельцев. – Томск: Изд-во НТЛ, 2006. – 128 с.
2. *Шолле, Ф.* Глубокое обучение на Python / Ф. Шолле – СПб. : Питер, 2018. – 400 с.
3. *Будума, Н.* Основы глубокого обучения. Создание алгоритмов для искусственного интеллекта следующего поколения / Н. Будума, Н. Локашо; пер. с англ. А. Коробейникова; [науч.ред. А. Созыкин]. – М. : Манн, Иванов и Фербер, 2020. – 304 с.
4. *Савин, Р.* Тестирование Дот Ком, или пособие по жестокому обращению с багами в интернет-стартапах / Р. Савин – М. : Дело, 2007. – 312 с.
5. *Ли Дж. Коупленд* A Practitioner's Guide to Software Test Design / *Ли Дж. Коупленд.* – Artech House Publishers, 2003. – 300 с.

Ломакин Максим Алексеевич — магистрант 2-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: lomak.max@mail.ru

Каплиева Наталья Алексеевна (научный руководитель) — канд. физ-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.

МОДЕЛИРОВАНИЕ ДИФФУЗИИ ИННОВАЦИЙ НА ПРИМЕРЕ МЕЖСТРАНОВОГО АНАЛИЗА РАСПРОСТРАНЕНИЯ ИНТЕРНЕТА

Д. Ю. Медведева, И. Н. Щепина

Воронежский государственный университет

Введение

Инновации являются движущей силой экономического роста, и на их скорость влияет доступность нового продукта. Слово «диффузия» стало хорошо известной коннотацией в контексте распространения инноваций. Распространение инноваций обеспечивает концептуальную парадигму для понимания процесса распространения и социальных изменений. Изучение диффузии инноваций важно, поскольку инновации без диффузии не выживают и не оказывают влияния на общество или экономику. Распространение необходимо в процессе инновационного роста и технологических изменений. Существуют 2 подхода к распространению инноваций. Первый подход основан на социально-экономических факторах микроуровня. Этот пространственный и традиционный подход начинается с предположения, что процесс распространения аналогичен распространению видов и/или болезней среди населения и зависит от социальной и экономической структуры. Второй подход изучает закономерности распространения инновации на макроуровне. Эти подходы можно лучше понять с помощью диффузионного моделирования.

1. Классические модели диффузии инноваций

1.1. Описание математических моделей диффузии, используемых для анализа распространения Интернета

Модели распространения инноваций широко применялись для объяснения того, как инновация распространяется в социальной системе через определенные каналы связи во времени и пространстве. В настоящее время известно множество моделей диффузии, которые используются при анализе инновационных процессов. Все модели отражают S-образные кривые и показывают три фазы внедрения новых продуктов: привлечение первых потребителей (медленный рост); резкий рост, когда все больше и больше людей вступают в контакт с инновацией; насыщение (замедление роста).

Рассмотрим наиболее часто используемые модели: модель Колемана (Coleman), модель Фишера-Прая (Fisher-Pry), модель Басса (Bass) и модель Гомперца (Dixon).

1.2. Модель Колемана

Модель Колемана является «инновационной» моделью (потребители ориентируются на свои предпочтения и рыночные объемы, предлагаемой новой технологии). Модель описывается следующим дифференциальным уравнением:

$$\frac{df(t)}{dt} = a(1 - f(t)), \quad (1)$$

где $f(t)$ — доля распространения данной инновации, a — коэффициент инновации.

Аналитическое решение уравнения (1) имеет вид:

$$f(t) = 1 - \beta e^{-at}. \quad (2)$$

Линеаризуем данное уравнение:

$$\ln(1 - f(t)) = \ln \beta - at. \quad (3)$$

Введем замену:

$$y = \ln(1 - f(t)); \beta_0 = \ln \beta; \beta_1 = -a. \quad (4)$$

Отсюда получаем уравнение линейной регрессии:

$$y = \beta_0 + \beta_1 t, \quad (5)$$

где $y = \beta_0 + \beta_1 t$.

Сумма квадратов отклонений считается по следующей формуле:

$$S = \sum_{t=1}^n (1 - \beta e^{-at} - f(t))^2. \quad (6)$$

1.3. Модель Фишера-Прая

Модель Фишера-Прая является «имитационной» моделью, т. е. потребители ориентируются не только на свои предпочтения, но и долю потребителей уже использующий данную технологию. Модель описывается следующим дифференциальным уравнением:

$$\frac{df(t)}{dt} = bf(t) \frac{(1 - f(t))}{1}, \quad (7)$$

где $f(t)$ — доля распространения данной инновации.

Аналитическое решение уравнения (7) имеет вид:

$$f(t) = \frac{1}{1 + \beta e^{-bt}}. \quad (8)$$

Линеаризуем данное уравнение:

$$\ln\left(\frac{1}{f(t)} - 1\right) = \ln \beta - bt. \quad (9)$$

Введем замену:

$$y = \ln\left(\frac{1}{f(t)} - 1\right); \beta_0 = \ln \beta; \beta_1 = -b. \quad (10)$$

Отсюда получаем уравнение линейной регрессии:

$$y = \beta_0 + \beta_1 t, \quad (11)$$

где $\beta = e^{\beta_0}$, $b = -\beta_1$.

Сумма квадратов отклонений считается по следующей формуле:

$$S = \sum_{t=1}^n \left(\frac{1}{1 + \beta e^{-bt}} - f(t)\right)^2. \quad (12)$$

1.4. Модель Басса

Модель Басса, являющаяся по сути обобщением моделей Колемана и Фишера-Прая, выражается следующим дифференциальным уравнением:

$$\frac{df(t)}{dt} = a(1 - f(t)) + bf(t) \frac{(1 - f(t))}{1}, \quad (13)$$

где a — коэффициент «инновации», b — коэффициент «имитации», $f(t)$ — доля распространения данной инновации.

Аналитическое решение имеет вид:

$$f(t) = \frac{1 - e^{-kt}}{1 + qe^{-kt}}, k = a + b, q = \frac{b}{a}. \quad (14)$$

Модель не линеаризуема, поэтому будем искать параметры модели, решая нелинейную оптимизационную задачу (15), с помощью встроенного в Excel пакета «поиск решений»:

$$S = \sum_{t=1}^n \left[\left(\frac{1 - e^{-kt}}{1 + qe^{-kt}} \right) - f(t) \right]^2 \rightarrow \min$$

$$K = a + b;$$

$$q = \frac{b}{a}; \quad (15)$$

$$0 \leq a \leq 1;$$

$$0 \leq b \leq 1.$$

1.5. Модель Гомперца

Модель, предложенная Диксоном, в основе своей имеет S-образную кривую Гомперца и поэтому чаще встречается в литературе как модель Гомперца. Она описывается следующим дифференциальным уравнением:

$$\frac{df(t)}{dt} = bf(t) \ln \left(\frac{f(t)}{M} \right), \quad (16)$$

где M — потенциальный размер рынка.

С начальным условием $f(0) = 0$. Решением является функция вида Гомперца:

$$f(t) = M \exp[-a \exp(-bt)]. \quad (17)$$

Для модели Гомперца значения параметров вычисляются, аналогично модели Басса, путем решения нелинейном оптимизационной задачи:

$$S = \sum_{t=1}^n [M \exp[-a \exp(-bt)] - f(t)]^2 \rightarrow \min,$$

$$a \geq 0;$$

$$b \geq 0;$$

$$M \geq 0. \quad (18)$$

2. Моделирование распространения Интернета

В работе анализировалось распространение Интернета в различных странах мира. Для анализа пользовалась данными с 1993 по 2017 год [15, 16], содержащие долю населения, пользующегося Интернетом при условии, что все население страны бралось за 100 %.

Для анализа были выбраны 7 стран, в которых распространение Интернета отличается по скорости, уровню распространения и периодам массового распространения. Траектории распространения представлены на рис. 1. Выбор данных стран был сделан по следующим основаниям. Швецию и Бельгию выбрали, потому что это страны — лидеры в продвижении инноваций с максимальным количеством пользователей Интернета. Эстонию — как страну Восточной Европы, которая также является лидером по распространению Интернета. Японию — как развитую восточную страну с высоким уровнем цифрового развития. Китай — как страну с высоким потенциалом для роста цифрового развития. И Индию — как страну с низким уровнем цифрового развития, но большим потенциалом для роста. И, конечно, в список стран

включена Россия, т. к. важно проанализировать траекторию распространения Интернета в нашей стране и оценить перспективы ближайшего будущего. Анализ проводился в 2 этапа.

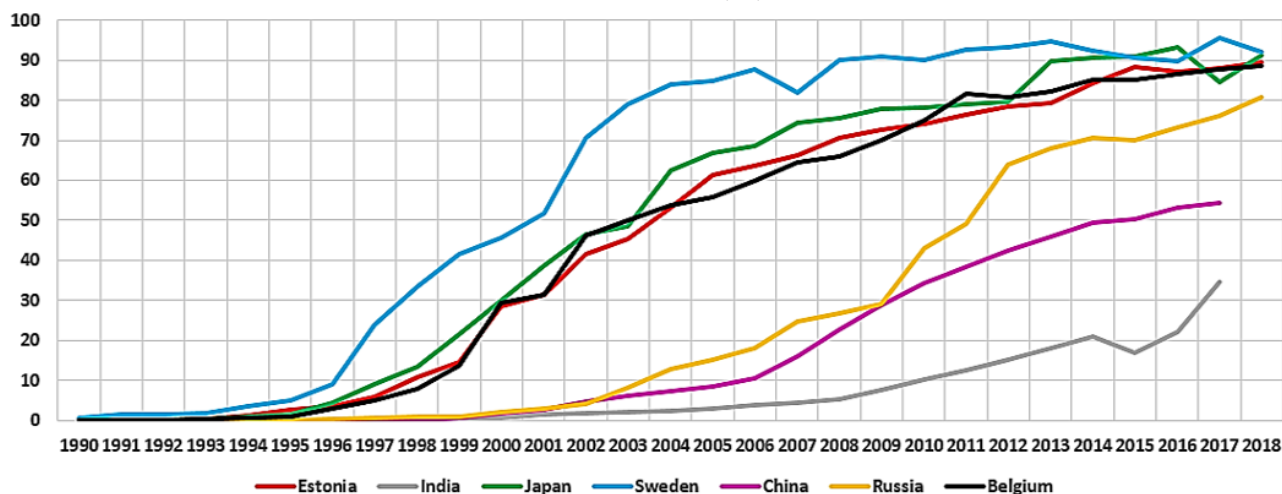


Рис. 1. Траектории распространения Интернета по 7 странам

Этап 1. Анализ проводился за период с 1993 по 2015 год по 7 странам, составлялся прогноз по моделям до 2020 года. Качество модели оценивалось по минимуму критерия — сумма квадратов отклонений модельных значений от реальных данных. Точность постпрогноза оценивалась по отклонению исходных данных за 2016, 2017 и 2018 годы от значений, полученных по моделям. Для каждой страны выбирались модели, которые наилучшим образом отражали распространение и имели наибольшую точность прогноза.

Этап 2. По выбранным моделям для каждой страны вновь вычислялись значения параметров по данным за период с 1993 по 2017 год и составлялся прогноз до 2022 года. Проверка качества модели оценивалась аналогично этапу 1. Также для всех стран, кроме Индии и Китая полученные результаты прогнозирования за 2018 год сравнивались по отклонениям с реальными данными.

Был проведен сравнительный анализ результатов моделирования распространения Интернета в разных странах мира, полученных по моделям Колемана (МК), Фишера-Прая (МФП), Басса (МБ) и Гомперца (МГ). В табл. 1 представлены прогнозные значения по распространению Интернета до 2022 года, полученные по наиболее адекватным моделям в результате 2-х этапного анализа.

Таблица 1

Прогнозы распространения Интернета по странам

Год	Эстония (Сред. знач. по модели Б. и модели Гом.)	Индия (Модель Гомперца)	Япония (Сред. знач. по модели Б. и модели Гом.)	Швеция (Сред. знач. по модели Б. и модели Гом.)	Китай (Модель Гомперца)	Россия (Модель Гомперца)	Бельгия (Сред. знач. по модели Б. и модели Гом.)
2018	0,9012	0,3198	0,9275	0,9947	0,5713	0,8212	0,9006
2019	0,9084	0,3524	0,9324	0,9959	0,5853	0,8480	0,9079
2020	0,9143	0,3852	0,9363	0,9969	0,5969	0,8712	0,9139
2021	0,9192	0,4178	0,9395	0,9977	0,6065	0,8910	0,9189
2022	0,9232	0,4500	0,9420	0,9982	0,6144	0,9079	0,9231

Полученные прогнозы показывают, что при сложившийся на сегодняшний день тенденции, к 2022 году практически все население Швеции будет иметь доступ к Интернету, в Японии этот показатель будет составлять 94 %, в Бельгии и Эстонии по 92 %, в России — 91 %, в Китае — 61 %, а в Индии доступ к Интернету будут иметь меньше половины жителей — 45 %.

После проведения данного исследования был разработан программный комплекс, позволяющий выбирать страну, предварительно загрузив данные по ней, и по представленным моделям оценить распространение Интернета и построить прогноз, получив результаты моделирования в табличном и графическом виде (рис. 2). Сравнение результатов моделирования проводится по показателю S , т. е. выбирается модель с минимальным значением суммы квадратов отклонений модельных данных от реальных.

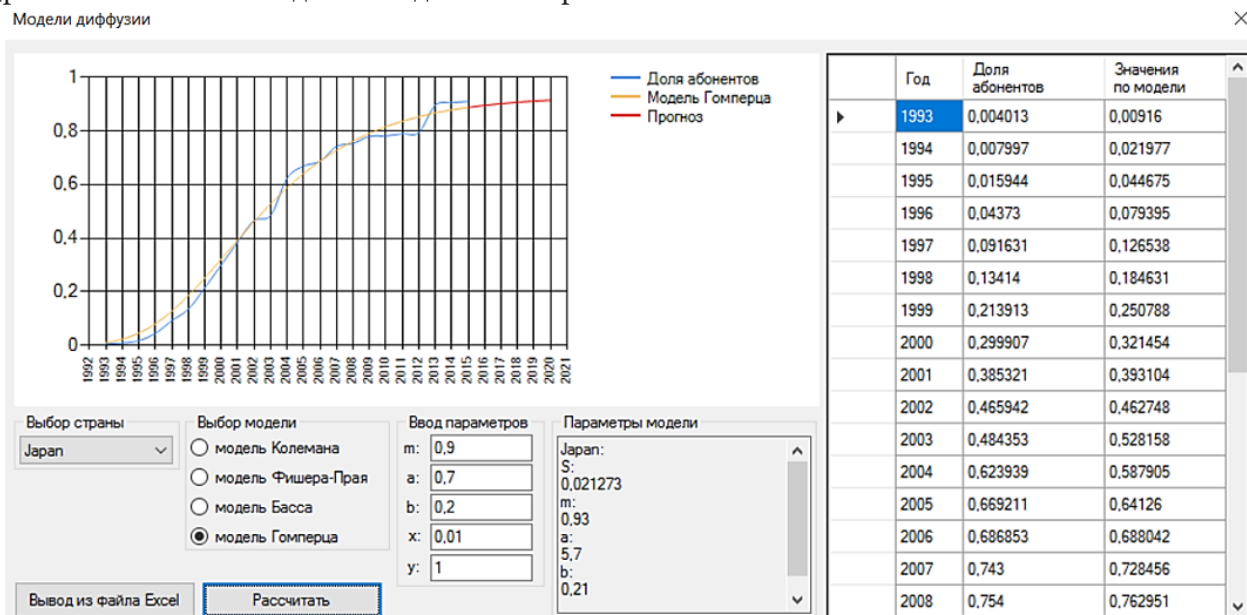


Рис. 2. Демонстрация программного комплекса

На данный момент известны данные о проникновении Интернета в страны мира до 2020 года (для некоторых стран до 2021 года) [17]. Таким образом, мы можем сравнить правильность полученных нами прогнозов (табл. 2). Результаты моделирования продемонстрировали достаточно точную аппроксимацию реальных данных и допустимые отклонения для постпрогноза (наибольшие отклонения не превосходят 10 %), что позволяет подтвердить адекватность использования диффузионных моделей для описания процессов распространения инновационных технологий.

Таблица 2

Прогнозы распространения Интернета по странам

Год	Эстония	Индия	Япония	Швеция	Китай	Россия	Бельгия
	Прогноз/ Реальные данные	Прогноз/ Реальные данные	Прогноз/ Реальные данные	Прогноз/ Реальные данные	Прогноз/ Реальные данные	Прогноз/ Реальные данные	Прогноз/ Реальные данные
2018	0,9012/ 0,918	0,3198/ 0,348	0,9275/ 0,925	0,9947/ 0,962	0,5713/ 0,593	0,8212/ 0,732	0,9006/ 0,901
2019	0,9084/ 0,935	0,3524/ 0,383	0,9324/ 0,932	0,9959/ 0,965	0,5853/ 0,601	0,8480/ 0,745	0,9079/ 0,915
2020	0,9143/ 0,962	0,3852/ 0,406	0,9363/ 0,935	0,9969/ 0,968	0,5969/ 0,628	0,8712/ 0,797	0,9139/ 0,939
2021	0,9192	0,4178	0,9395/ 0,945	0,9977	0,6065	0,8910	0,9189

Заключение

Диффузионные модели эпидемического подхода позволяют достаточно хорошо отражать долгосрочную тенденцию для распространения глобальных инноваций. Однако рассмотренные модели являются дескриптивными и не включают экономические и институциональные параметры, оказывающие влияние на процесс распространения инноваций. Хотя модели достаточно хорошо отражают реальные процессы, но при смене тенденций распространения требуется выбор временного периода для настройки параметров модели. Этот период может оказаться небольшим, что ухудшает прогностические возможности модели. Например, вынужденная самоизоляция и перевод большинства видов общения в онлайн в настоящее время может способствовать более быстрому увеличению доли пользователей Интернета в странах, где его распространение еще не достигло максимума возможных пользователей. Поэтому для выявления источников, влияющих на скорость распространения инновационных технологий и построения более обоснованных прогнозов необходимо использовать модели диффузии, включающие экономические и институциональные факторы.

Литература

1. Шумпетер, Й. Теория экономического развития: исследование предпринимательской прибыли, капитала, кредита, процента и цикла конъюнктуры / Й. Шумпетер; пер. В. С. Автономов, М. С. Любский, А. Ю. Чепуренко. – Москва : Прогресс, 1982. – 456 с.
2. Пригожин, А. И. Нововведение: стимулы и перспективы / А. И. Пригожин. – Москва : Политиздат, 1998 г. – 81 с.
3. Rogers, E. M. Diffusion of Innovations. (5th ed.) / E. M. Rogers. – New York : Free Press, 2003. – 208 p.
4. Bass, F. M. A New Product Growth Model For Consumer Durables / F. M. Bass. – Management Science, 1969. – 227 p.
5. Hagerstrand, T. Diffusion of Innovation as The Arial Process / T. Hagerstrand. – Management Science, 1954. – 85 p.
6. Романченко, С. В. Новшества, нововведения, инновации: определения и сущность / С. В. Романченко // Молодой ученый. – 2012. – № 4 (39). – 168 с.
7. Азгальдов, Г. Г. Интеллектуальная собственность, инновации и квалиметрия / Г. Г. Азгальдов, А. В. Костин // Экономические стратегии. – 2008. – 184 с.
8. Земцов, С. П. Моделирование диффузии инноваций и типология регионов России на примере сотовой связи / С. П. Земцов, В. Л. Бабурин // Известия Российской академии наук. Серия географическая, 2017. – 56 с.
9. Барановский, С. Ю. Теория моделирования диффузии инноваций / С. Ю. Барановский, А. В. Пузыревская // Наука и инновации. 2018. – 78 с.
10. Лапшов, В. В. Интегрированная система инновационного аудита «ИСИА» / В. В. Лапшов // Становление, развитие и перспективы оценочной деятельности в России. Тез. докл. I-й междунар. конфер., 25–26 ноября 2008. – Москва, 2008.
11. Щепина, И. Н. Моделирование многокомпонентного замещения продуктов и технологий / И. Н. Щепина // Диссертация на соискание ученой степени кандидата экономических наук. – Москва, 1991. – 167 с.
12. Концепция инновационной политики Российской Федерации на 1998-2000 гг. (утверждена Постановлением Правительства РФ от 24 июля 1998 г. № 832).
13. Руководство Осло. Рекомендации по сбору и анализу данных по инновациям. Третье издание. – Москва, 2010. – 107 с.

14. Oslo Manual. Guidelines for Collecting and Interpreting Innovation Data (3rd edition) // A Joint Publication of OECD and Eurostat. OECD/EC, 2005. – 84 p.

15. UN data. A world information. – Режим доступа: <http://data.un.org/Data.aspx?q=internet&d=ITU&f=ind1Code%3aI99H> (дата обращения: 02.05.2020).

16. Международный союз электросвязи. – Режим доступа: <https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx> (дата обращения: 02.05.2020).

17. World Internet Users and Population Stats 2022. – Режим доступа: <https://www.internetworldstats.com/stats.htm> (дата обращения: 22.04.2022).

Медведева Дарья Юрьевна — магистрант 2-го года обучения кафедры Математических методов исследования операций Воронежского государственного университета.
E-mail: MedvedevaDY@yandex.ru

Щепина Ирина Наумовна (научный руководитель) — д-р экон. наук, доц., доцент кафедры математических методов исследования операций Воронежского государственного университета.

МОДЕЛЬ BERT ДЛЯ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

В. И. Минаков, С. В. Золотарёв

Воронежский государственный университет

Введение

Данная работа посвящена модели языкового представления BERT, который, в отличие от других моделей языкового представления, предназначен для предварительного подготовки представлений из неразмеченного текста, путем обработки контекста на всех уровнях. В результате чего такая модель может быть настроена всего с одним дополнительным выходным слоем для создания современных моделей, которые будут использоваться для широких кругов задач (ответы на вопросы, поддержка диалога с клиентом). Предварительно обученные языковые модели (PLM) нацелены на изучение универсальных языковых представлений путем выполнения обучающих задач с самоконтролем на масштабный корпус. Покажем, что предварительно обученные представления уменьшают потребность во многих сложных архитектурах для конкретных задач. BERT — это первая модель представления на основе тонкой настройки, которая обеспечивает высочайшую производительность в большом наборе задач на уровне предложений и токенов, превосходя многие архитектуры для конкретных задач.

1. Архитектура модели BERT

1.1. Описание проблемы

Одной из основных причин хорошей работы BERT в различных задачах обработки естественного языка было использование так называемого *semi-supervised learning*. Это означает, что модель обучается для конкретной задачи, которая позволяет ей понимать языковые шаблоны. После обучения модель имеет возможности обработки языка, которые можно использовать для расширения возможностей других моделей, которые используются для обучения с учителем.

В данной структуре есть два этапа: *pre-training* (предварительное обучение) и *fine-tuning* (настройка). Во время предварительного обучения модель использует неразмеченные данные в качестве главного источника обучения. Для точной настройки модель BERT сначала инициализируется с предварительно обученными параметрами, и все параметры настраиваются с использованием помеченных данных из последующих задач. Каждая нижестоящая задача имеет отдельные точно настроенные модели, даже если они инициализируются с одними и теми же предварительно обученными параметрами.

На рис. 1.1 представлена подробная схема двух этапов.

Помимо выходных слоев, одни и те же данные используются как для предварительной подготовки, так и для тонкой настройки. Одни и те же параметры предварительно обученной модели используются для инициализации моделей для различных последующих задач. При тонкой настройке все параметры уточняются.

Перед подачей последовательностей слов в BERT 15 % слов в каждой последовательности заменяются токеном [MASK]. Затем модель пытается предсказать исходное значение замаскированных слов на основе контекста, обеспечиваемого другими, не замаскированными словами в последовательности. С технической точки зрения, предсказание выходных слов требует

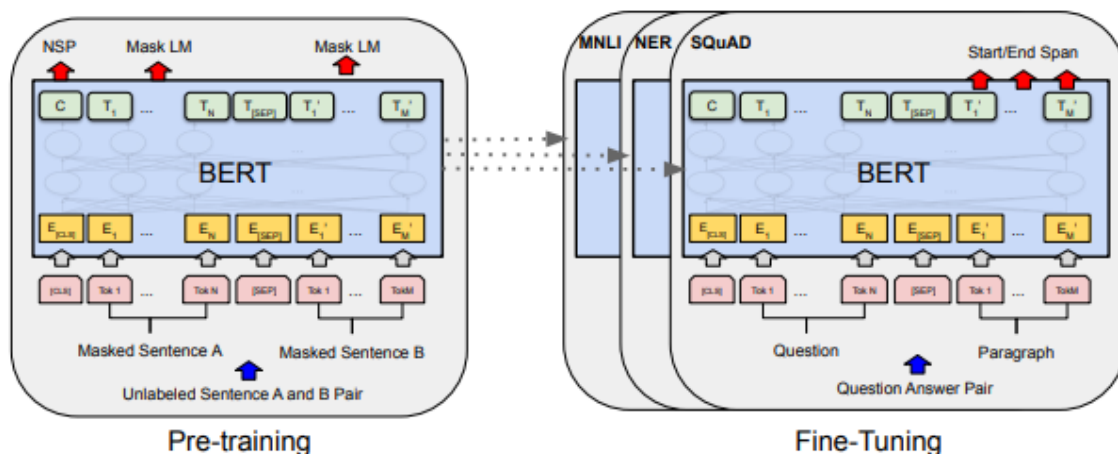


Рис. 1.1. Подробная схема этапов BERT

добавление слоя классификации поверх вывода. Затем полученные выходные векторы умножаются на матрицу вложения и преобразовывают их в размерность словаря. Последним этапом является расчет вероятности каждого слова в словаре с помощью softmax. Функция потерь BERT учитывает только предсказание замаскированных значений и игнорирует предсказание не замаскированных слов. Как следствие, модель сходится медленнее, чем направленные модели, что компенсируется ее повышенной осведомленностью о контексте (рис. 1.2).

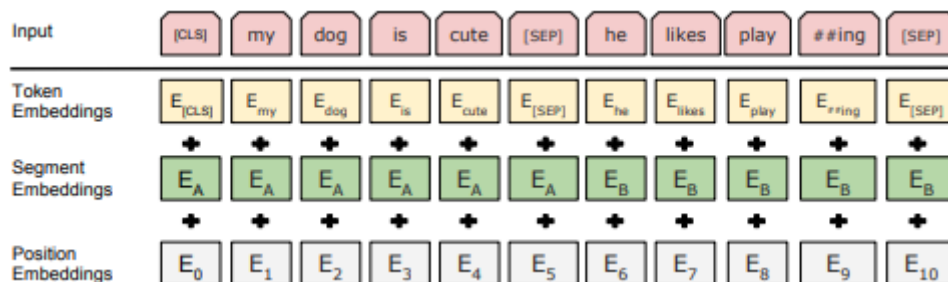


Рис. 1.2. Входное представление BERT

Таким образом, дана текстовая последовательность:

$$X = [CLS, x_1, x_2, \dots, x_L, SEP] \quad (1)$$

С L токенами, а языковая модель f_{LM} производит контекстуальное представление слова:

$$f_{LM}(X) = [h_{CLS}, h_1, h_2, \dots, h_L, h_{SEP}] \quad (2)$$

Для конкретной задачи функция f_H дополнительно использует $f_{LM}(x)$ и генерирует прогноз как $f_H(h_{CLS})$ для задач классификации последовательностей.

1.2. Предварительное обучение модели

Чтобы найти данную функцию, необходимо взять стартовый набор параметров. Возьмем алгоритм сегментации подслов WordPiece, содержащий 30000 токенов, который используется для задач обработки естественного языка. Первый токен последовательности всегда является токеном CLS (токен специальной классификации). Окончательное состояние, соответствующее этому токеному, используется как совокупное представление последовательности для классификации задачи. Прежде всего, разделяем эти пары специальным токеном SEP и добавляем настройку к каждому токеному, который указывает, принадлежит ли он к предложению A или к предложению B.

Интуитивно разумно полагать, что глубокая двунаправленная модель более эффективна, чем модель left-right. К сожалению, стандартные условные языковые модели можно обучать только слева направо или справа налево, поскольку такой подход позволит каждому слову косвенно «видеть себя», а модель может тривиально предсказать целевое слово в многоуровневом контексте.

На рис. 1.3 показано входное представление BERT. Входные вложения представляют собой сумму вложений токенов, вложений сегментации и вложений позиций.

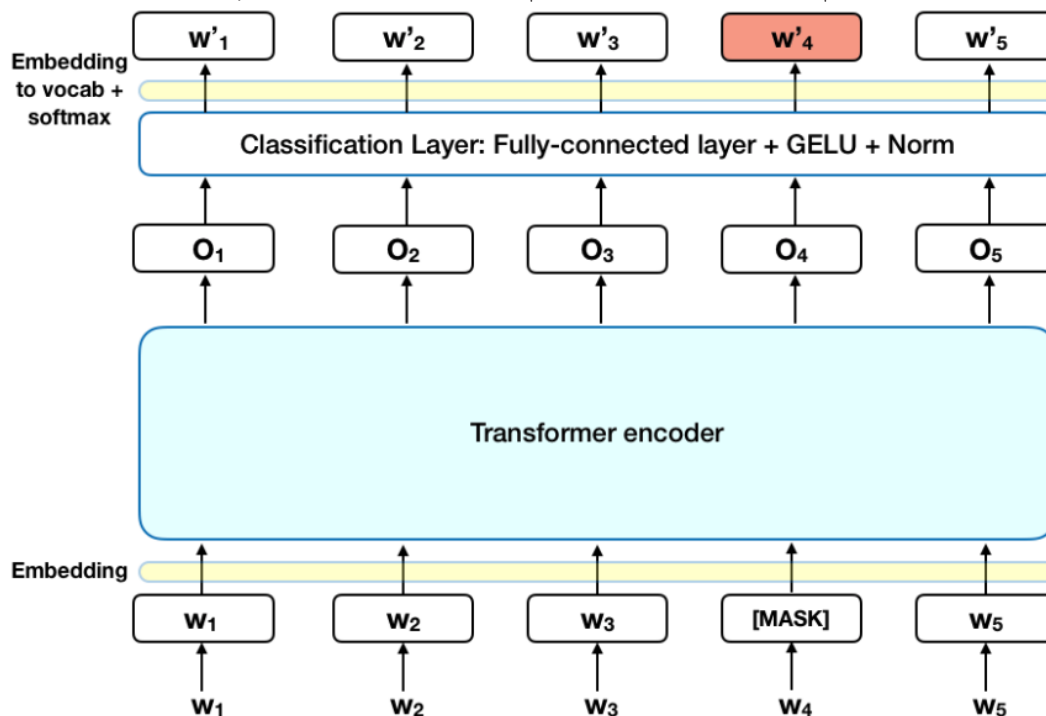


Рис. 1.3. Входное представление BERT

Чтобы модель различала два предложения при обучении, перед входом в модель ввод обрабатывается с помощью токенов [CLS] и [SEP], которые вставляются в начало и конец каждого предложения соответственно. Затем позиционное вложение, указывающее на предложение А или В добавляется к каждому токenu, чтобы указать его положение в последовательности. При обучении модели BERT, *Masked LM* и *Next Sentence Prediction* обучаются вместе с целью минимизировать комбинированную функцию потерь двух стратегий.

1.3. Fine-training (тонкая настройка)

Механизм настройки в *Transformer* позволяет BERT моделировать многие последующие задачи — независимо от того, включают ли они один текст или текстовые пары — путем замены соответствующих входов и выходов. Для приложений, использующих текстовые пары, общий шаблон состоит в том, чтобы независимо кодировать текстовые пары. Вместо этого BERT использует механизм для объединения этих двух этапов, поскольку кодирование сцепленной текстовой пары включает двунаправленную связь между двумя предложениями.

Это не единственный способ использования BERT. Существует возможность использовать предварительно обученный BERT для создания контекстуализированных вложений слов. Затем передавать эти вложения в существующую модель — процесс, который, дает результаты, не сильно отстающие от точной настройки BERT для такой задачи, как распознавание именованных объектов.

2. Результаты работы

В соответствии с использованием рекомендованных параметров были получены следующие результаты сгенерированных данных (рис. 2.1).

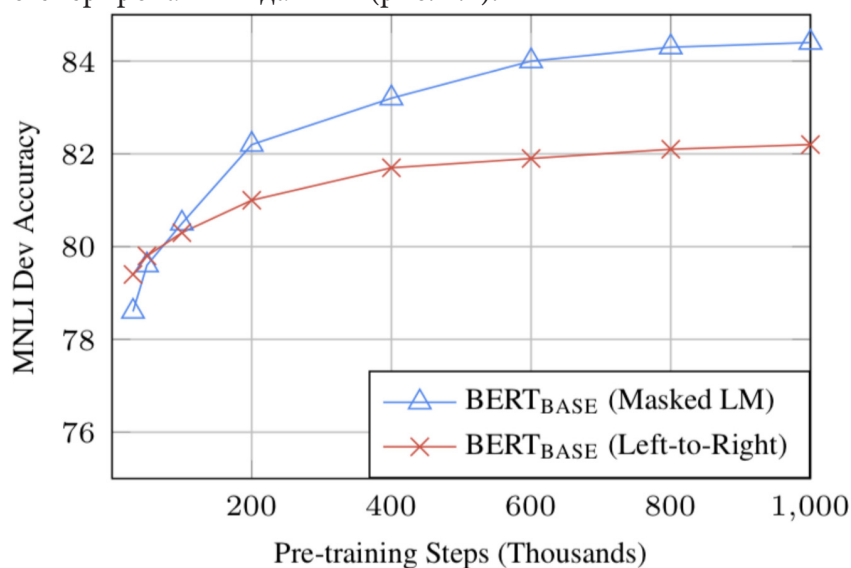


Рис. 2.1. Результаты pre-training BERT

Двунаправленный подход BERT (MLM) сходится медленнее, чем подходы слева направо (поскольку в каждой партии прогнозируется только 15 % слов), но двунаправленное обучение по-прежнему превосходит обучение слева направо после небольшого количества предварительных шагов. При достаточном количестве данных для обучения больше шагов обучения и более высокая точность. Например, в задаче MNL точность улучшается на 1,0 % при обучении на 1 млн шагов (размер пакета 128 000 слов) по сравнению с 500 000 шагов с тем же размером пакета.

Заключение

Таким образом, BERT совершил скачок в области интеллектуальной обработки естественного языка и закрепил превосходство использования предварительно обученных моделей представления текста на огромных наборах данных в качестве универсальной основы для построения интеллектуальных алгоритмов решения конкретных задач. Основной вклад заключается в дальнейшем обобщении этих результатов для глубоких двунаправленных архитектур, позволяющих одной и той же предварительно обученной модели успешно решать широкий набор задач обработки естественного языка.

Литература

1. Bert: Pre-training of deep bidirectional transformers / Jacob Devlin. [et al.] // arXiv:1810.04805v2 [cs]. – 16 p. – URL: <https://arxiv.org/pdf/1810.04805> (дата обращения: 1.04. 2022).
2. Bert: A Review of Applications in Natural Language Processing and Understanding / M. V. Koroteev // arXiv:2103.11943v1 [cs]. – 2021. – 18 p. – URL: <https://arxiv.org/ftp/arxiv/papers/2103/2103.11943.pdf> (дата обращения: 01.04. 2022).

Минаков Виталий Игоревич — магистрант 2-го года обучения кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: mindev97@gmail.com

Золотарёв Сергей Владимирович (научный руководитель) — канд. физ.-мат. наук, преподаватель кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета. E-mail: szolotarev82@gmail.com

РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ: ОПРЕДЕЛЕНИЕ, ЗАДАЧИ, АРХИТЕКТУРА

Н. В. Муратова

Воронежский государственный университет

Введение

Рекомендательная система (РС) в общем смысле представляет собой набор программных средств, технологий и инструментов, нацеленных на предсказание интересов пользователей и составление рекомендаций, которые будут им интересны.

Примерами рекомендаций могут служить: книги, музыка, фильмы, новости, товары и т. д. Для обозначения того, что система рекомендует пользователям, чаще всего используется термин «элемент» или «объект».

В своей простейшей форме рекомендации предлагаются в виде ранжированных списков элементов. При выполнении этого ранжирования рекомендательные системы пытаются предсказать, какие продукты или услуги являются наиболее подходящими, исходя из предпочтений и ограничений пользователя. Для выполнения такой вычислительной задачи рекомендательные системы собирают у пользователей их предпочтения, которые либо явно выражены, например, в виде рейтингов продуктов, либо выводятся путем интерпретации действий пользователя. Например, рекомендательная система может рассматривать переход на страницу определенного продукта как неявный признак предпочтения элементов, показанных на этой странице.

Развитие РС началось с довольно простого наблюдения: люди часто полагаются на рекомендации других при принятии рутинных, повседневных решений. Например, при выборе книги для чтения принято полагаться на рекомендации сверстников; работодатели рассчитывают на рекомендательные письма при принятии решения о приеме на работу; и при выборе фильма для просмотра люди, как правило, читают и полагаются на обзоры фильмов, написанные кинокритиком и опубликованные в газете, которую они читают.

Как самостоятельная область исследований данное направление появилось в середине 1990-х гг. В последние годы резко возрос интерес к рекомендательным системам, о чем свидетельствуют следующие факты:

1. Рекомендательные системы играют важную роль на таких интернет-сайтах, как Яндекс, Amazon.com, YouTube, Netflix, Tripadvisor, IMDb. Более того, многие компании в настоящее время разрабатывают и внедряют РС как часть услуг, которые они предоставляют своим подписчикам. Например, Netflix (сервис онлайн-просмотра фильмов) провела соревнования по разработке алгоритма, позволяющего существенно улучшить производительность своей рекомендательной системы.

2. Существуют специальные конференции и семинары, связанные с рекомендательными системами. Например, ACM Recommender Systems (RecSys) – главная международная конференция, на которой ежегодно представляются новые результаты исследований, методов и технологий и определяются тренды в области рекомендательных систем. Кроме того, сессии, посвященные РС, часто включаются в более традиционные конференции по базам данных, информационным системам и адаптивным системам.

1. Постановка задачи

Существует два способа *постановки задачи* для рекомендательных систем.

1. Прогнозирование. Первый подход заключается в прогнозировании значения рейтинга для комбинации пользователь-элемент. Предполагается, что доступны обучающие данные, указывающие предпочтения пользователя в отношении элементов. Для m пользователей и n элементов это соответствует неполной матрице размера $m \times n$, где заданные (или наблюдаемые) значения используются для обучения. Отсутствующие (или ненаблюдаемые) значения прогнозируются с использованием этой обучающей модели. Эта проблема также называется **проблемой завершения матрицы**, так как задана не полностью определенная матрица значений, а остальные значения предсказываются алгоритмом обучения.

2. Ранжирование. На практике нет необходимости предсказывать оценки пользователей по конкретным элементам, чтобы давать рекомендации пользователям. Скорее, продавец может порекомендовать топ- k товаров для конкретного пользователя или определить топ- k пользователей, на которых следует ориентироваться для конкретного товара. Определение первых k элементов более распространено, чем определение первых k пользователей, хотя методы в этих двух случаях полностью аналогичны. Эта проблема также называется **проблемой рекомендаций топ- k** .

Во втором случае абсолютные значения прогнозируемых рейтингов не важны. Первая формулировка является более общей, поскольку решения для второго случая могут быть получены путем решения первой формулировки для различных комбинаций пользовательских элементов и последующего ранжирования прогнозов. Однако во многих случаях проще и естественнее разработать методы непосредственного решения ранжирующей версии задачи.

В общем виде постановка задачи может быть представлена следующим образом.

Введем следующие обозначения:

$U = \{u_1, u_2, \dots, u_n\}$ — множество пользователей,

$P = \{p_1, p_2, \dots, p_n\}$ — множество исследуемых продуктов (объектов),

$R = (r_{ij})$ — матрица рейтингов размера $m \times n$, где $i \in 1, \dots, n$, $j \in 1, \dots, m$.

Возможные значения матрицы определяются типом рейтинга и могут быть числами в промежутке от 1 (не нравится) до 5 (нравится), либо значениями 0 и 1 (понравилось/не понравилось). Если пользователь i не оценил элемент j , то соответствующий элемент матрицы r_{ij} остается пустым.

Матрицу R можно представить в виде таблицы, где строки соответствуют пользователям, а столбцы — объектам (см. табл. 1).

Таблица 1

Рейтинг продуктов пользователей системы

	Продукт 1	Продукт 2	Продукт 3	Продукт 4	Продукт 5
Пользователь 1	1	2	5	1	?
Пользователь 2	4	3	4	2	5
Пользователь 3	5	1	?	3	2
Пользователь 4	?	3	3	2	5

Для каждого пользователя необходимо заполнить пропущенные значения матрицы, таким образом предсказать рейтинг для неоцененных объектов. Среди полученных результатов оценок необходимо выбрать только релевантные, т. е. удовлетворяющие предпочтениям пользователя, объекты.

Обозначим:

\hat{r}_{ij} — прогнозируемый рейтинг для пользователя i относительно продукта j ,

N — количество рекомендуемых пользователю продуктов,

$(p_{i1}, p_{i2}, \dots, p_{iN})$ — N -мерный вектор рекомендаций.

Тогда для заданного пользователя u_i задача рекомендательной системы сводится к нахождению N -мерного вектора $(p_{i1}, p_{i2}, \dots, p_{iN})$, где p_{ik} — неоцененные пользователем объекты в матрице рейтингов R , $k \in N$, таким образом, чтобы полученные продукты наиболее точно удовлетворяли предпочтениям пользователя, т. е. прогнозируемый рейтинг \hat{r}_{ik} имел наибольшее значение.

2. Классификация рекомендательных систем

Рекомендательные системы работают с двумя видами информации:

- *Данные о характеристиках:* информация о товарах (ключевые слова, категории и т. д.) и пользователях (предпочтения, профили и т. д.).

- *Данные об оценках пользователей объектам:* рейтинги, количество покупок и т. д.

Исходя из этого, различают три класса алгоритмов, используемых в рекомендательных системах:

- Системы на основе контента, в которых используется информация о характеристиках.

- Системы коллаборативной фильтрации, основанные на взаимодействии пользователя с элементом.

- Гибридные системы, которые объединяют оба типа информации с целью избежать проблем, возникающих при работе только с одним видом.

2.1. Системы, основанные на контентной фильтрации

Данные системы составляют рекомендации, используя ключевые характеристики элементов и профиль пользователя. Профиль пользователя обычно заполняется на основе взаимодействия с системой или путем прямого опроса пользователей об их интересах.

Как правило, каждому элементу со своими характеристиками сопоставляется числовой вектор в некотором заданном пространстве. Затем, используя меру подобия, можно группировать элементы в кластеры и предлагать их пользователям. Функция подобия может быть простой функцией, такой как скалярное произведение или евклидово расстояние, или более сложной.

Преимущества:

1. Не требуется хранить большие данные. Достаточно только задать описание ключевых характеристик объектов и пользователя.

2. Сравнительная простота в реализации.

Недостатки:

1. Одна из возникающих проблем — выдача очевидных рекомендаций из-за чрезмерной специализации (пользователя А интересуют только категории В, С и D, и система не может рекомендовать товары вне этих категорий, даже если они могут быть ему интересны).

2. Еще одна распространенная проблема заключается в том, что новые пользователи не имеют определенного профиля, если только у них явно не запрашивают информацию. Эта проблема также известна под названием «холодный старт». Тем не менее, добавить новые элементы в систему относительно просто. Нужно убедиться, что им назначена группа в соответствии с их характеристиками.

2.2. Системы коллаборативной фильтрации

Коллаборативная фильтрация в настоящее время является одним из наиболее часто используемых подходов и обычно дает лучшие результаты, чем рекомендации на основе кон-

тента. Некоторые примеры ее использования можно найти в таких известных системах, как Яндекс.Музыка, Кинопоиск, Youtube.

Такие виды систем используют взаимодействия с пользователем для фильтрации интересных элементов. Они основаны на принципе, согласно которому похожие пользователи предпочитают похожие товары, а похожие товары предпочитаются похожими пользователями.

Набор взаимодействий пользователя с системой представляется с помощью матрицы, где каждая запись $a(i, j)$ означает взаимодействие между пользователем i и элементом j .

Существует несколько классов методов для решения задачи прогнозирования по заданной матрице: фильтрация на основе памяти и на основе моделей.

Коллаборативная фильтрация на основе памяти

Существует два подхода: первый идентифицирует *кластеры пользователей* и использует взаимодействия одного конкретного пользователя для прогнозирования взаимодействий других подобных пользователей. Второй подход идентифицирует *кластеры элементов*, которые были оценены пользователем А, и использует их для прогнозирования взаимодействия пользователя А с другим, но похожим элементом В.

Эти методы обычно сталкиваются с серьезными проблемами с большими разреженными матрицами, поскольку количество пользовательских элементов взаимодействия могут быть слишком низкими для создания высококачественных кластеров.

Так же, как и в системах, основанных на контентной фильтрации, коллаборативная фильтрация сталкивается с проблемой холодного старта. Чтобы составить рекомендации для новых пользователей, необходимо предварительно собрать данные о взаимодействии пользователя с элементами системы.

Кроме того, одной из значительных проблем данного метода является его плохая масштабируемость: создание новой рекомендации может занять много времени для больших систем, так как размерность матрицы будет достаточно большой.

В большинстве алгоритмов рекомендаций необходимо быть предельно осторожным, чтобы избежать эффекта перенасыщения для популярных товаров и не создать для пользователей так называемую «область ограничения информации». Другими словами, нельзя, чтобы система рекомендовала только популярные товары, а также, чтобы пользователи получали рекомендации только для товаров, очень близких к тому, который им уже понравился. Эти проблемы могут возникать в большинстве алгоритмов рекомендаций, но данное утверждение особенно верно для алгоритмов совместной фильтрации, основанных на памяти. При отсутствии модели «регуляризации» такого рода явления могут усиливаться и наблюдаться чаще.

Коллаборативная фильтрация на основе модели

Данный класс методов основан на методах машинного обучения и интеллектуального анализа данных. Его цель состоит в том, чтобы обучить модели строить прогнозы. Например, можно использовать существующие взаимодействия пользователя с элементом, чтобы научить модель прогнозировать несколько основных элементов, которые могут понравиться пользователю больше всего. Одним из преимуществ этих методов является то, что они могут рекомендовать большее количество элементов большему количеству пользователей по сравнению с другими методами, такими как фильтрация на основе памяти.

Коллаборативная фильтрация на основе модели полагается только на информацию о взаимодействии пользователя с элементом и предполагают наличие скрытой модели, которая должна объяснять эти взаимодействия.

Одним из способов получения такой скрытой модели является *матричная факторизация*. Алгоритм матричной факторизации состоит в разложении достаточно большой разреженной матрицы взаимодействия пользователя и элемента на произведение двух меньших и плотных матриц: матрицу «пользователь-фактор» (содержащую представления пользователей) и матрицу «элемент-фактор» (содержащую представления элементов). Таким образом, взаимодействие между пользователем и элементом может быть получено путем вычисления скалярного произведения соответствующих плотных векторов (матриц) в заданном пространстве.

Заключение

Рекомендательные системы являются одним из наиболее используемых методов машинного обучения и находят применение во многих областях, от экономики до Интернета вещей. В данной статье приведен обзор основных характеристик рекомендательных систем: введено определение, сформулированы варианты постановки задачи, рассмотрена классификация рекомендательных систем. Для каждого класса (коллоборативные рекомендательные системы, системы на основе содержимого) выявлены их преимущества и недостатки. Основными проблемами в данной области являются: сильная разреженность данных, возможность быстрой адаптации модели к изменению данных, составление разнообразных рекомендаций, «холодный старт» системы. Для преодоления существующих недостатков чаще всего используются гибридные рекомендательные системы.

Литература

1. Фальк, К. Рекомендательные системы на практике / пер. с англ. Д. М. Павлова. – Москва : ДМК Пресс, 2020. – 448 с.
2. Ricci, F. Recommender Systems. Handbook / F. Ricci, L. Rokach, B. Shapira, Kantor P. – NY : Springer, 2015 – 1008 p.
3. Aggarwal, C. C. Recommender Systems. The textbook / C. C. Aggarwal. – NY : Springer, 2016 – 518 p.
4. Кутянин, А. Р. Рекомендательные системы: обзор основных постановок и результатов / А. Р. Кутянин // Интеллектуальные системы. Теория и приложения. – 2017. – том 21, выпуск 4. – С. 18–30.
5. Нефедова, Ю. С. Архитектура гибридной рекомендательной системы GEFEST / Ю. С. Нефедова // Системы и средства информатики. – 2012. – том 22, выпуск 2. – С. 176–196.
6. Barot, M. Fuzzy Logic Based Recommender System / M. Barot, K. H. Wandra, S. B. Patel // International Journal of Research and Scientific Innovation. – 2017. – volume IV, issue VIS. – P. 71–80.
7. Introduction to recommender systems. – Режим доступа: <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>. – (Дата обращения: 18.03.2022).

Муратова Наталья Владимировна — аспирант 2 курса кафедры математического и прикладного анализа Воронежского государственного университета. E-mail: nat.muratova@gmail.com

Шашкин Александр Иванович (научный руководитель) — профессор, доктор физико-математических наук, заведующий кафедрой математического и прикладного анализа Воронежского государственного университета. E-mail: Shashkin@kvm.vsu.ru

МАТЕМАТИЧЕСКАЯ И ПРОГРАММНАЯ ПОДДЕРЖКА ФОРМИРОВАНИЯ ОПТИМАЛЬНОГО КОМПЛЕКТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ УЧЕБНОГО ПРОЦЕССА ВУЗА В УСЛОВИЯХ ИМПОРТОЗАМЕЩЕНИЯ

М. С. Наумов, Ю. В. Бондаренко

Воронежский государственный университет

Введение

Стремительное развитие информационных технологий и их интенсивное проникновение во все процессы и сферы жизнедеятельности, ставят перед руководством высших учебных заведений новые задачи, связанные с подготовкой высококвалифицированных кадров для цифровой трансформации общества. Платформой успешного их решения является наличие в вузе базы современного программного обеспечения, способного покрыть растущие информационные потребности учебного процесса и обеспечить достижение высокого уровня развития цифровых компетенций обучающихся. Вместе с тем, в условиях ограниченности финансовых ресурсов, разнообразия рынка ИТ-технологий и его бурного развития, формирование и регулярное обновление базы программного обеспечения (ПО) является комплексной задачей, успешность решения которой во многом зависит от способности руководства вуза эффективно согласовать актуальные потребности цифровой экономики и с возможностями и интересами участников учебного процесса [1–3]. Современные реалии, связанные с введением санкций на зарубежное программное обеспечение, стали дополнительным вызовом для информационного обеспечения вузов страны, добавляя новые жесткие ограничения на номенклатуру ПО и требуя от руководства вузов обновления ПО в направлении исключения санкционного ПО и перехода на отечественные технологии. В условиях импортозамещения задача принятия решений руководством вуза по формированию оптимального пакета отечественного ПО, является остро актуальной.

Целью настоящего исследования является разработка и программная реализация математического подхода, позволяющего, в сложных условиях импортозамещения, осуществить действенную поддержку руководству вуза в оперативном принятии решений по формированию комплекта программного обеспечения учебного процесса.

Математическую основу предлагаемого подхода составляет модель формирования оптимального комплекта ПО, обеспечивающего покрытие потребностей учебных дисциплин с учетом предпочтений и интересов участников учебного процесса и требованиями сведения к минимуму использование зарубежных технологий.

Перейдем к описанию предлагаемого подхода.

1. Описание подхода

В настоящем исследовании без ограничения общности будем рассматривать некоторое структурное подразделение вуза (например, факультет). Полагаем, что исходя из рабочих программ учебных дисциплин, преподаваемых на факультете, возможно определить укрупненный перечень задач или тем, решение и усвоение которых требует использование программного обеспечения. Например, поиск обратной матрицы, решение системы уравнений, вычисление интегралов, построение моделей системной динамики, регрессионный анализ, нейронные сети и т. д. Число таких задач (а оно может быть достаточно велико) обозначим через n , а порядковой номер задачи через j , где $j = 1, \dots, n$.

Предположим, что для включения в учебный процесс может быть рассмотрено m программных продуктов (ПП), i — порядковый номер ПО, где $i = 1, \dots, m$. Множество индексов программных продуктов $M = \{1, 2, \dots, m\}$ разобьем на два непересекающихся подмножества $M = M_0 \cup M_1$, где M_0 — множество индексов отечественного ПО, M_1 — множество индексов разрешенного зарубежного ПО.

Каждый конкретный программный продукт имеет определенный функционал и может быть использован как инструмент решения некоторого круга учебных задач. Для формального описания соответствия программного продукта учебным задачам введем в рассмотрение матрицу покрытий программного обеспечения учебных задач.

Матрица покрытий A имеет размерность $m \times n$, а ее элементы формируются по следующему правилу:

$$a_{ij} = \begin{cases} 1, & \text{если задача } j \text{ может быть решена с помощью ПО } i, \\ 0, & \text{иначе.} \end{cases}$$

Каждая программа j имеет определенную стоимость p_j , в которую может быть включена стоимость лицензии, установки, обслуживания, консультаций специалистов и т. д. Вместе с тем, на закупку ПО в конкретный момент времени выделен определенный финансовый бюджет. Обозначим через y и Y соответственно минимальный и максимальный объем денежных средств, которые могут быть потрачены на закупку ПО.

Будем учитывать, что количество закупаемых программ может варьироваться от значения c до C .

Естественно, что каждый программный продукт обладает некоторым набором характеристик, каждая из которых является более или менее важной для организации качественного учебного процесса. Набор характеристик, по которым предлагается осуществлять сравнение комплектов ПО должен учитывать интересы участников учебного процесса — преподавателей, студентов, системных администраторов и др. В числе частных критериев оценки могут быть выбраны следующие: количество в комплекте зарубежных ПП, стоимость, сложность освоения, длительность обучения студентов работе, наличие мануала, дружелюбность интерфейса, качество и полнота решения задач, готовность студентов к освоению в рамках часов, отведенных в рабочей программе дисциплины, универсальность, распространенность в практической деятельности компаний, частота обновлений, объем занимаемой памяти и т. д.

Частные критерии могут быть как количественными, так и качественными. Для формирования количественных значений последних привлекаются эксперты из числа заинтересованных лиц и применяются методы обработки экспертной информации (например, метод парных сравнений) [4]. На вопросе формирования таких оценок в настоящем исследовании останавливаться не будем, полагая, что выбраны L частных критериев оценки ПО:

$$F_1, F_2, \dots, F_L.$$

Каждый критерий представляет собой некоторую функцию, позволяющую осуществить сравнение альтернатив — ПП и направленность — максимум или минимум.

Задача формирования оптимального комплекта ПО состоит в определении такого набора ПП, который обеспечивает достижение максимального (минимального) значения каждого частного критерия и при этом покрывает все потребности учебного процесса, доступен по бюджету и удовлетворяет ограничениям на количество.

Для формального описания задачи введем в рассмотрение m булевых переменных $x_i \in \{0, 1\}$. Тогда каждому комплекту ПО ставится в соответствие вектор $x = (x_1, x_2, \dots, x_m)$, состоящий из 0 и 1. При этом, если $x_i = 1$, то i -й программный продукт включается в комплект, а если $x_i = 0$, то нет.

С учетом введенных обозначений математическая форма записи задачи формирования оптимального комплекта ПО имеет следующий вид:

$$\begin{aligned}
F_1(x) &\rightarrow \text{extr}, \\
F_2(x) &\rightarrow \text{extr}, \\
&\dots \\
F_L(x) &\rightarrow \text{extr},
\end{aligned}
\tag{1}$$

$$\left\{ \begin{array}{l}
\sum_{i=1}^m a_{ij} \cdot x_i \geq 1, j = 1, \dots, n, \\
y \leq \sum_{i=1}^m x_i \leq Y, \\
c \leq \sum_{i=1}^m x_i \leq C, \\
x_i \in \{0, 1\}.
\end{array} \right.
\tag{2}$$

В условиях импортозамещения одним из критериев задачи является минимизация количества зарубежного ПО:

$$\sum_{i \in M_1} x_i \rightarrow \min.$$

Задача (1)–(2) относится к классу целочисленных задач многокритериальной оптимизации [5], решением которой являются эффективные (Парето-оптимальные) точки. Методы решения таких задач основываются на дополнительной информации от лица, принимающего решения, на основании которой возможно осуществить выбор единственного решения.

При решении задачи (1)–(2) полагаем, что лицо, принимающее решение, может количественно определить веса важности каждого частного критерия: $\alpha_1, \dots, \alpha_L$, где $\alpha_i \geq 0$, $\sum_{l=1}^L \alpha_l = 1$.

Тогда задачу (1)–(2) можно свести к задаче однокритериальной оптимизации с функцией цели, представляющей взвешенную сумму частных критериев:

$$\sum_{l=1}^L \alpha_l F_l^n(x) \rightarrow \max,
\tag{3}$$

где $F_l^n(x)$ — нормированный частный критерий с индексом l , $l = 1, \dots, L$.

Задачу однокритериальной оптимизации (2)–(3), и все частные задачи, необходимые для нормировки критериев, предлагается решать методом Гомори.

2. Программная реализация

Решение представленной задачи реализовано на языке программирования Python версии 3.9 в среде PyCharm. Готовый программный продукт представляет собой десктопное приложение, работающее вне среды разработки на ОС Windows / Linux / MacOS.

В качестве характеристик для программных продуктов были выбраны следующие: сложность освоения, наличие мануала, язык интерфейса, статус санкций для продукта, вес приложения, последний год обновления приложения, востребованность в образовании, оценка пользователей, цена. Перечень программного обеспечения и значения характеристик хранятся в виде таблицы Excel (рис. 1).

Пользователь имеет возможность выбора частных критериев, среди которых:

- максимизация суммарной оценки привлекательности для учебного процесса программных продуктов комплекта;
- минимизация среднего порога знаний студентов, необходимого для адаптации их к использованию программных продуктов;

Название программы	Сложность освоения	Наличие мануала	Есть русский язык	Под санкциями	Вес ПО	Последний год обновлений	Востребованность ПО в образовании	Оценка пользователей	Цена
Matlab	Высокая	Да	Нет	Да	3788,8	2021	Высокая	3,5	175850,5
Mathcad	Высокая	Да	Да	Нет	24,1	2020	Высокая	4	134400
IBM SPSS Statistics	Средняя	Нет	Да	Нет	971,5	2018	Высокая	3,5	96530,7
Mathematica	Высокая	Да	Нет	Возможно	1,6	2019	Высокая	4	14966
Minitab	Средняя	Нет	Нет	Нет	362,1	2018	Средняя	4	209000
MathType	Средняя	Да	Нет	Возможно	40	2017	Высокая	4	4592
Graph	Средняя	Нет	Да	Нет	9,6	2017	Низкая	4	0
Maple	Высокая	Да	Нет	Возможно	48,9	2017	Высокая	4	116734,8
Derive	Низкая	Нет	Нет	Возможно	3	2017	Низкая	4	18707,5
Scilab	Низкая	Да	Да	Возможно	352	2019	Средняя	4	0
PASW	Средняя	Да	Нет	Да	308,9	2017	Низкая	4	45101
GeoGebra	Низкая	Нет	Да	Нет	93,9	2021	Средняя	3,5	0
GnuPlot	Низкая	Нет	Нет	Нет	52,4	2019	Низкая	4	0
Gretl	Низкая	Нет	Да	Нет	62,7	2021	Низкая	4	0
Mathomatic	Высокая	Да	Нет	Да	0,340820313	2019	Средняя	4	0
Microsoft Matemati	Низкая	Да	Да	Да	36	2019	Высокая	4	0
Smath Studio	Низкая	Нет	Да	Нет	2,2	2019	Средняя	4	0
Maxima	Средняя	Нет	Нет	Нет	255	2020	Низкая	4	0
GeoEnZo	Низкая	Нет	Нет	Возможно	3,3	2020	Низкая	4	0

Рис. 1. Таблица характеристик ПО

- минимизация количества зарубежных программных продуктов;
- минимизация суммарной стоимости программных продуктов.

Экран выбора файлов (excel таблицы) частных и интегральных критериев представлен на рис. 2.

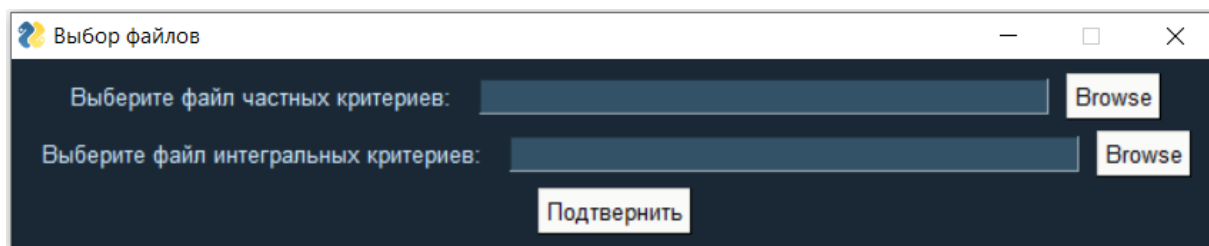


Рис. 2. Экран выбора файлов частных и интегральных критериев

На экране пользователь может: по нажатию на кнопку «Browse» указать через системный обозреватель путь к требуемому файлу; при фокусе в поле ввода пути к файлу — вручную указать путь к файлу; по нажатию на кнопку «Подтвердить» — проверка валидации.

Если валидация прошла успешно — пользователь переходит на экран настройки первичных фильтров (рис. 3).

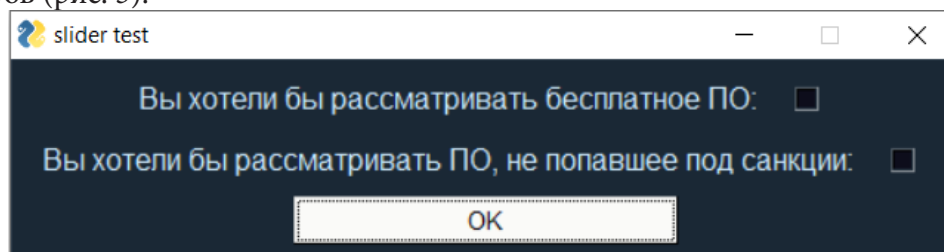


Рис. 3. Экран настройки первичных фильтров

На экране пользователь может выбрать способы фильтрации данных (фильтр по бесплатным программным продуктам, исключая фильтр по санкционным программным продуктам). По нажатию на кнопку «ОК» — пользователь переходит на экран настройки важности критериев (рис. 4).

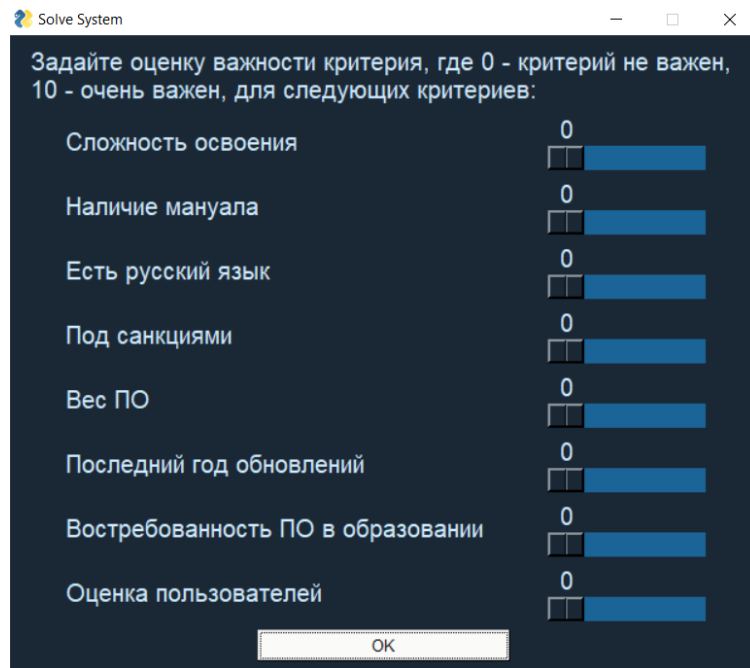


Рис. 4. Экран настройки важности критериев

Экран ввода данных для решения задачи формирования оптимального комплекта ПО представлен на рис. 5.

На экране пользователь может ознакомиться со списком программных продуктов, прошедших и не прошедших на этап формирования оптимального пакета; может выбрать чек-боксами целевые функции для формирования пакета.

Результаты работы программы — оптимальный пакет ПО представляется в окне вывода и при желании может быть сохранен (рис. 6).

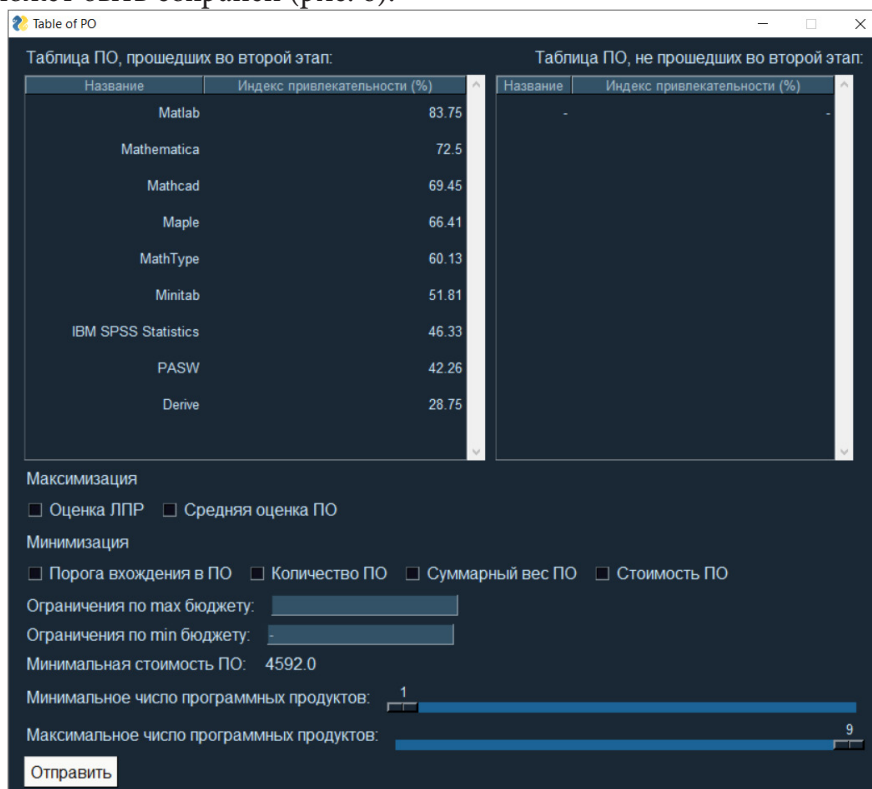
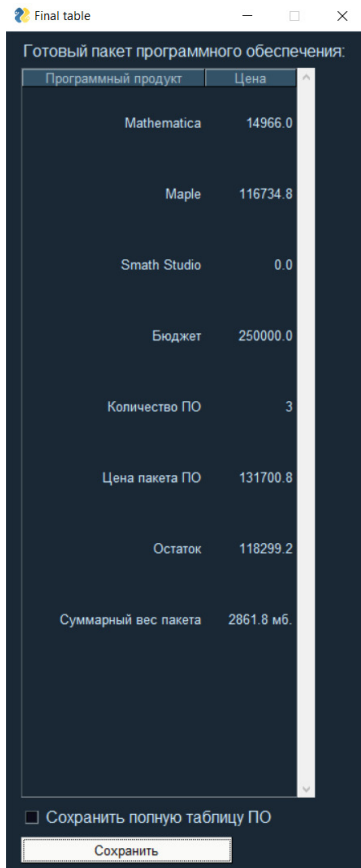


Рис. 5. Экран ввода данных задачи формирования оптимального комплекта ПО

Заключение



Программный продукт	Цена
Mathematica	14966.0
Maple	116734.8
Smath Studio	0.0
Бюджет	250000.0
Количество ПО	3
Цена пакета ПО	131700.8
Остаток	118299.2
Суммарный вес пакета	2861.8 мб.

Сохранить полную таблицу ПО

Сохранить

Рис. 6. Окно вывода результатов

Настоящее исследование посвящено разработке математической и программной поддержке формирования оптимального комплекта ПО для обеспечения учебного процесса в вузе, которая приобрела особую актуальность в условиях санкционных ограничений и исключительную важность для успешности процессов импортозамещения. Основу математического блока составляет задача формирования оптимального комплекта ПО, представляющая задачу дискретной оптимизации при многих критериях. Для осуществления оперативных расчетов разработан программный продукт, позволяющий в удобной для пользователя формы вводить исходных данные и получать результаты расчетов.

Дальнейшее направление исследования связано с разработкой многовариантного алгоритма, позволяющего сделать процесс формирования ПО адаптируемым к различным условиям.

Отметим, что разработанный подход является универсальным и допускает расширение не только на образовательные организации.

Литература

1. Бобровских, А. И. Формирование оптимального комплекта программного обеспечения в образовательных организациях / А. И. Бобровских, Ю. В. Бондаренко // Вестник Южно-Уральского университета. Сер. Компьютерные технологии, управление, радиоэлектроника. – 2021. – Т. 21, № 4. – С. 5–18.
2. Глухенький, А. Н. Цифровизация образовательного процесса в вузе: реальность и перспективы / А. Н. Глухенький, Н.М. Глухенькая // Перспективы науки. – 2020. – № 11(134). – С. 95–97.
3. Гулина, О. В. Информационно-образовательное пространство экономического вуза в условиях цифровизации / О. В. Гулина // Цифровая трансформация. – 2021. – № 2. – С. 5–12.
4. Грешилов, А. Н. Математические методы принятия решений / А. Н. Грешилов. – Москва: Издательство ВГТУ им. Н. Э. Баумана, 2014. – 647 с.
5. Баева, Н. Б. Теория и практика векторной оптимизации / Н. Б. Баева, Ю. В. Бондаренко, Т. В. Азарнова, И. Л. Каширина. – Воронеж : Издательский дом ВГУ, 2017. – 82 с.

Наумов Максим Сергеевич — студент 4-го курса кафедры Математических методов исследования операций Воронежского государственного университета. E-mail: makser2000@mail.ru

Бондаренко Юлия Валентиновна (научный руководитель) – д-р техн. наук, доц., профессор кафедры Математических методов исследования операций Воронежского государственного университета. E-mail: bond.julia@mail.ru

МОБИЛЬНЫЙ БЭК-ОФИС «Z-OFFICE»

А. С. Овчинникова, Г. В. Абрамов

Воронежский государственный университет

Введение

В настоящее время невозможно представить свою жизнь без телефона, и, соответственно, без ряда мобильных приложений, используемых пользователем. Мобильные устройства решают множество задач — от самых простейших до более сложных. Развитие мобильных приложений не стоит на месте, эта отрасль развивается стремительными темпами и с каждым днем достигает огромных результатов.

В связи с этим, большой популярностью среди индивидуальных предпринимателей и компаний пользуются мобильные приложения для получения аутсорсинговых услуг в сфере: бухгалтерии; маркетинга; экономического и финансового анализа и т. д.

1. Постановка задачи

Цифровой бэк-офис «Z-office», позволяет пользователям с помощью личного кабинета на сайте, получать аутсорсинговые услуги в сфере: бухгалтерии; маркетинга; экономического и финансового анализа; бизнес-планирования; оптимизации налогообложения; кадрового делопроизводства, юридического консультирования и формирования документов.

Бэк-офис «Z-Office» представляет собой информационно-аналитическую систему, позволяющую провести цифровизацию следующих направлений деятельности организации:

- формирование финансовой отчетности организации и ее отправка в государственные органы;
- выполнение работ по оптимизации системы налогообложения в организации;
- проведение экономического и финансового анализа организации;
- бизнес-планирование и формирование бизнес-модели;
- бюджетирование деятельности организации и т. д.

На данном этапе разработки приложение должно иметь функциональные возможности для выполнения некоторых задач виртуального бэк-офиса, таких как:

- аутентификация в приложении, использующая различные роли;
- сохранение архивных задач для определенных специалистов в хранилище;
- возможность обмениваться сообщениями с некоторым числом специалистов;
- создание и отображение таблиц и диаграмм по различным показателям, сформированных по различным приоритетам.

Развитие проекта потребует включение в приложение функциональных экранов для постановки и реализации задач в сферах: экономики, финансов, маркетингового анализа, чтобы пользователь посредством приложения через соответствующие экраны мог ставить задачи специалистам, отправлять документы, скачивать сформированные документы и отчеты, осуществлять мониторинг собственного расчетного счета, обладать аналитическими данными о своей финансово-хозяйственной деятельности.

2. Структура приложения

Бэк-офис «Z-office» представляет собой мобильное приложение с личным кабинетом с определенными функциональными возможностями (рис. 1) для пользователей сервиса.

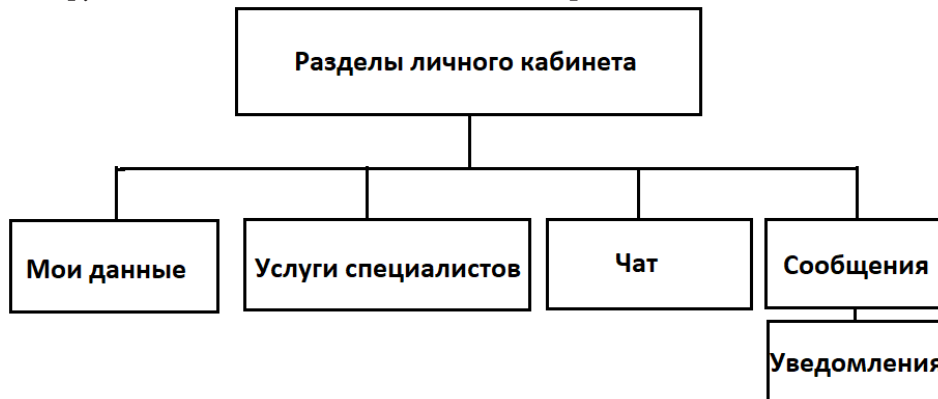


Рис. 1. Структура личного кабинета

В качестве проектной архитектуры для мобильного приложения бэк-офиса «Z-office» была выбрана клиент-серверная архитектура (рис. 2), где сервер баз данных, файловый сервер и другие представляют собой отдельный уровень, результаты работы которого использует сервер приложений. Все обращения клиентов к базе данных происходят через промежуточное программное обеспечение, которое находится на сервере приложений. Вследствие этого, повышается гибкость работы и производительность.

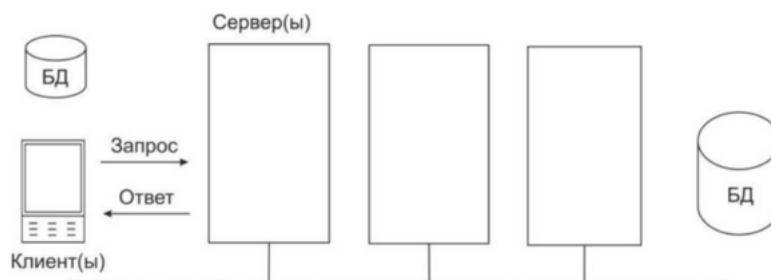


Рис. 2. Клиент-серверная архитектура

3. Программные средства

Данное мобильное приложение разрабатывается под операционную систему Android (версии 7.0 и выше). На данный момент в приложении используется клиент-серверная архитектура. Средой разработки данного проекта является Android Studio. К преимуществам данной среды разработки относят:

- Универсальность.
- Расширенный редактор для работы с макетами.
- Доступность данной среды для любой операционной системы.
- Большой набор функций.
- Поддержка таких языков, как: C++, Java, Kotlin.

К недостаткам данной среды разработки относят то, что для запуска эмулятора требуется высокопроизводительная аппаратная основа ПК.

В связи с этим в данном проекте было выбрано два устройства эмуляции: Google Pixel (версия Android: 7.1.1) и Google Pixel 3A (версия Android: 8.0). В качестве языков программи-

рования были выбраны Java и Kotlin и их библиотеки. Java — строго типизированный объектно-ориентированный язык программирования. Язык активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной. Для такой компиляции используется дополнительный инструмент, а именно Android SDK (Software Development Kit), разработанный компанией Google.

Достоинствами языка являются:

- Достаточно простой и удобный синтаксис
- Независимость от платформы
- Автоматическое управление памятью
- Стабильность

Помимо этого, в качестве базы данных была выбрана Firebase, Figma для дизайна интерфейса приложения и Git в качестве системы контроля версий. Система контроля версий является программным обеспечением, которое упрощает работу с изменяющейся информацией. Эта возможность хранить некоторое количество вариантов одного и того же документа и при необходимости возвращаться к предыдущим его версиям, а также определять, кто и когда совершил то или иное изменение. При изменении последней версии проекта разработчик создает новую версию с помощью commit в локальном репозитории. В том случае, если работа над каким-то модулем была завершена, то она отправляется на удаленный репозиторий.

4. Разработка Android-приложения

На данном этапе работы приложение содержит несколько экранов и выполняет ряд задач, как, например, авторизация пользователей с их ролями в рамках одного предприятия. Благодаря удобству облачной базы данных Firebase регистрация происходит быстро и просто, а разработчику нет необходимости делать разметку для формы регистрации. Она позволяет пользователям хранить и получать сохраненную информацию, а также может помочь с регистрацией и авторизацией пользователей, хранением сессий (авторизованные пользователи), медиафайлов к которым с легкостью предоставляет доступ благодаря Cloud Storage. В данном проекте используется эта база данных для авторизации пользователей (рис. 3).

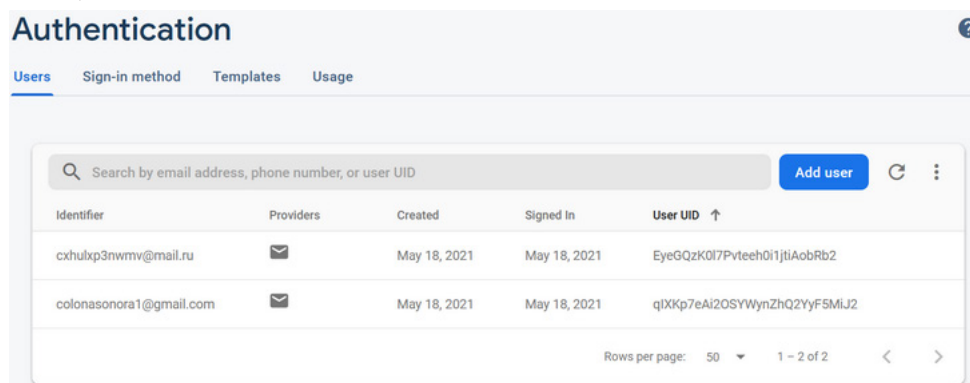


Рис. 3. Аутентификация в Firebase

После входа/регистрации, пользователь оказывается в главном меню приложения (рис. 4).

Данный экран обладает несколькими функциональными областями: переходы на разделы — чат, оповещения, информация, настройки, техподдержка (все они объединены элементом GridLayout, который упрощает формат расположения кнопок посредством сетки); также есть кнопки «Задачи», «Аналитика», «Отчеты» и «Счет».

Функциональный экран «Задачи» также обладает некоторыми важными функциями. Для постановки задачи с помощью кнопки «Выбрать специалиста» и выпадающего списка определяется специалиста, которому ставится задача (рис. 5). В разделе «описание задачи» в текстовом формате описывается срочная задача и при нажатии кнопки «отправить задание» отправляет специалисту, а если возникает необходимость прикрепить файлы к данной задаче (на данный момент в приложении возможно только прикрепить файл в формате «jpg» или «png»), то это возможно сделать, нажав на соответствующую кнопку. Ниже представлена форма постановки задачи.

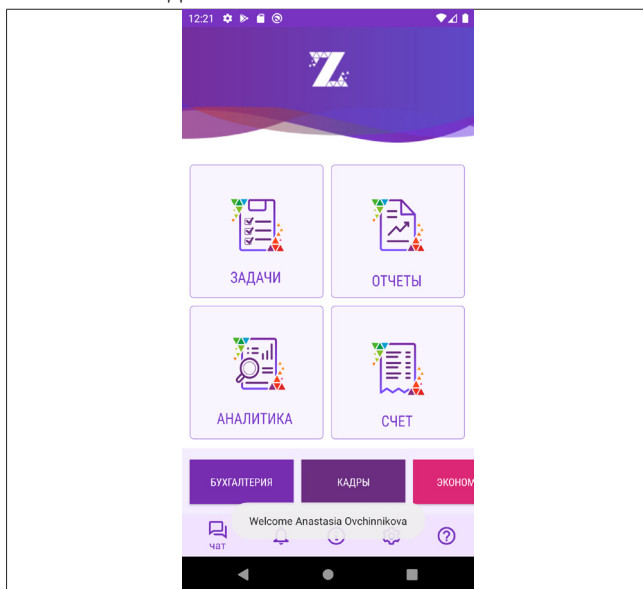


Рис. 4. Главное меню

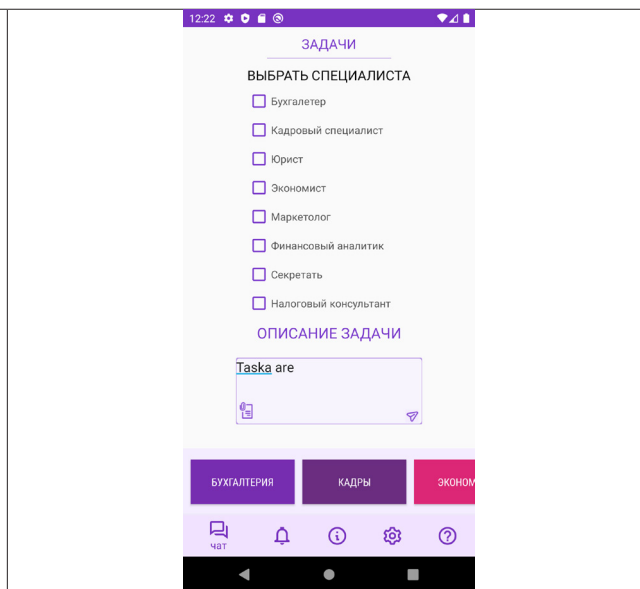


Рис. 5. Описание задачи

Для сохранения архивных задач пользователя была также создана база данных с помощью Firebase (рис. 6).

ExtraInfo		
id_extra	integer	
id_task	integer	
pic	blob	
doc	blob	
Add field		

Task		
id_task	integer	
title	varchar(255)	
note	varchar(255)	
created_ts	integer	
Add field		

Рис. 6. Таблица для архивных задач

В приведенной выше таблице хранится вся информация о задачах, которые были направлены конкретным специалистам — так, к примеру, это не только номер задачи, но и ее содержание вместе с датой создания задачи. Для корректной работы базы данных Firebase необходимо не только указать файл с ней, но и указать необходимые плагины для ее обработки.

После того, как была создана и отправлена хотя бы одна задача, пользователь получает доступ к архиву, то есть ко всем документам и задачам, которые были им отправлены. Все архивные задачи используют пролистывающийся список, и при нажатии на любую из них, возможен повторный просмотр.

Также важной особенностью приложения является возможность получить более наглядные данные с помощью функции «Аналитика». На данной области (рис. 7) существуют графики со значениями, показатели которых формируются за счет расчетных таблиц специалистов и отправки этих данных в приложение. На данном этапе возможен просмотр графиков с прибылью и выручкой предприятия за последние полгода работы.

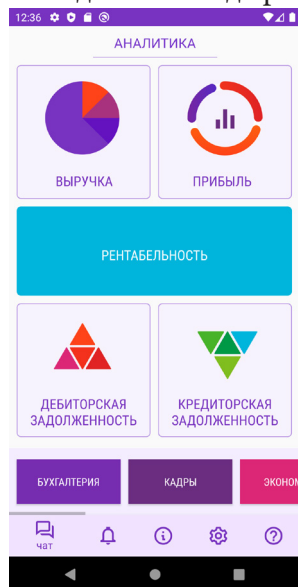


Рис. 7. Аналитика

Важной особенностью приложения является использование метода анализа иерархий в рамках анализа предприятия. Данная возможность представляет собой достаточно простой инструмент для решения некоторых проблем предприятия путем обоснования принятия решений в условиях многокритериальности и определенности. Так, к примеру, выбрав несколько поставщиков одного товара на специальном экране, можно определить, услугами которого из них стоит воспользоваться, предварительно изменив индексы приоритетов, в том случае, если они изменятся, программа пересчитает вектора приоритетов и выведет новый результат.

Помимо этого, в приложении также реализована возможность чата со специалистами. Реализация системы обмена сообщениями на Android подразумевает под собой создание разметки основного экрана (который представляет собой чат-комнату). В интерфейсе присутствуют такие элементы как:

- Список, который отображает все сообщения группового чата в хронологическом порядке;
- Поле ввода для нового сообщения;
- Кнопка для отправки сообщения в чат.

Чтобы сохранять сообщения чата в базе данных Firebase в реальном времени, необходимо создать для них модель. После того как она будет создана, можно будет легко добавлять новые сообщения в базу данных Firebase в реальном времени. Также необходим макет для создания сообщений чата, который содержит информацию о каждом отправленном сообщении - отправитель, дата отправки и само сообщение.

Для настройки выхода из чата необходимо переопределить метод метода класса AuthUI для выхода пользователя. При выходе из чатов приложение автоматически закрывается.

Для сохранения чатов в базе данных была создана модель, которая имеет три представления, и для их заполнения нужны три поля. Для этого был создан класс ChatMessage.java, и чтобы модель была совместима с FirebaseUI были добавлены get- и set- методы.

Заключение

В процессе работы был разработан прототип приложения бэк-офиса «Z-office» с несколькими экранами, каждый из которых обладает своими функциональными возможностями. При этом изучена работа подобных приложений, язык программирования Java и его отдельные библиотеки, разработка приложений для Android-устройств, а также основы работы с базой данных Firebase.

В перспективе планируется реализовать полностью необходимые функциональные возможности для пользователя, который через соответствующие экраны может ставить задачи специалистам, отправлять документы, скачивать сформированные документы и отчеты, осуществлять мониторинг собственного расчетного счета, обладать аналитическими данными о своей финансово-хозяйственной деятельности.

Литература

1. FireBase – Краткое руководство. – Режим доступа: <https://coderlessons.com/tutorials/veb-razrabotka/izuchite-firebase/firebase-kratkoe-rukovodstvo>. – (Дата обращения: 31.03.2022).
2. Шилдт, Г. Java 8: руководство для начинающих / Г. Шилдт. – 6-е изд., перераб. и доп. – Москва : Вильямс, 2015. – 720 с.
3. Android Application Development. – Режим доступа: https://www.fandroid.info/category/androidid_development_tutorials/ – (Дата обращения: 12.04.2022).
4. Java official documentation. – Режим доступа: <https://docs.oracle.com/en/java/> – (Дата обращения: 14.04.2022).
5. SQLite – Краткое руководство. – Режим доступа: <https://coderlessons.com/tutorials/bazy-dannykh/vyuchit-sqlite/sqlite-kratkoe-rukovodstvo> – (Дата обращения: 20.04.2022).
6. Основы Material Design. – Режим доступа: <https://material.io/design?hl=ru> – (Дата обращения: 22.04.2022).

Овчинникова Анастасия Сергеевна — магистрант 1-го года обучения кафедры Математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: an.s.ovchinnikova@mail.ru

Абрамов Геннадий Владимирович (научный руководитель) — д-р техн. наук, проф., заведующий кафедрой математического обеспечения ЭВМ Воронежского государственного университета.

ЧИСЛЕННО-АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ТЕПЛОВОЙ ОБРАБОТКИ СЕМЯН МАСЛИЧНЫХ КУЛЬТУР ВЫСОКОТЕМПЕРАТУРНЫМ ТЕПЛОНОСИТЕЛЕМ

К. В. Орешин, А. А. Шевцов, Н. А. Сердюкова

*Военный учебно-научный центр Военно-воздушных сил
«Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина»*

Введение

В последнее время использование рапсового масла получило широкое распространение в пищевой и химической промышленности. Однако использование рапса в качестве перспективного источника сырья замедляется в связи со сложностью кинематических закономерностей тепловой обработки семян при сушке и дальнейшей переработке семян. Определение полей температур и влагосодержаний в единичных семенах масличных культур, принципиально важным является разработка модели тепломассопереноса, которая позволила бы проводить анализ множества альтернативных вариантов аппаратурно-технологического оформления тепловой обработки и выбирать из них оптимальные с точки зрения энергетических затрат и показателей качества высушиваемого продукта экспериментально выполнить крайне сложно.

На данный момент при тепловой обработке зерновых культур отдается предпочтение высокотемпературным теплоносителям обладающим существенными преимуществами [1–3]. Для получения более точных результатов при решении задач оптимальными параметрами в процессе сушки возникает необходимость в разработке новых методов аналитического решения А. В. Лыкова. Температурные режимы при сушке семян рапса, с одной стороны, должны быть ограничены областью допустимых технологических свойств зерна, а с другой стороны, обеспечить снижение удельных теплоэнергетических затрат за счет увеличения производительности процесса обработки.

Основная часть

Сформулированная на основе дифференциальных уравнений А. В. Лыкова математическая модель процесса сушки семян рапса учитывает обобщенный закон перемещения влаги во влажном зерне в виде пара и жидкости [4]. Для ее решения приняты следующие упрощающие допущения: пренебрегается теплопередачей за счёт теплопроводности; не учитывается усадка и градиент давления; единичное зерно представляется в форме шара; теплофизические параметры зависят от температуры и влажности семян; начальное распределение температуры и влагосодержания в частице продукта постоянны.

С учетом принятых допущений уравнения нестационарного тепломассопереноса в сферической системе координат представлены в виде.

$$c_m \rho_0 \frac{\partial \theta}{\partial t} = \lambda \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{2\partial \theta}{x \partial x} \right) + \varepsilon r_0 \rho_0 \frac{\partial u}{\partial t},$$

$$\frac{\partial u}{\partial t} = a_m \left(\frac{\partial^2 u}{\partial x^2} + \frac{2\partial u}{x \partial x} \right) + a_m \delta \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{2\partial \theta}{x \partial x} \right), \quad (1)$$

с граничными условиями третьего рода:

$$\begin{aligned}
-\lambda \frac{\partial \theta}{\partial x} \Big|_{x=R} + \alpha (\theta|_{x=R} - \theta_c) - (1 - \varepsilon) r_0 \rho_0 \beta (u|_{x=R} - u_c) &= 0, \\
a_m \frac{\partial u}{\partial x} \Big|_{x=R} + a_m \delta \frac{\partial \theta}{\partial r} \Big|_{x=R} + \beta (u|_{x=R} - u_p) &= 0,
\end{aligned} \tag{2}$$

и начальными условиями:

$$\theta(x, 0) = \theta_0, \quad u(x, 0) = u_0, \tag{3}$$

где θ_0 , θ — начальная и текущая температура семян, К, θ_c — температура теплоносителя, К; u , u_p , u_0 — соответственно текущее, равновесное и начальное влагосодержание семян, кг/кг; ε — критерий фазового превращения, величина безразмерная, характеризующая долю влаги, перемещающейся в виде пара; r_0 — удельная теплота парообразования, кДж/кг; a_m — коэффициент диффузии влаги, м²/с; δ — термоградиентный коэффициент, 1/К; α — коэффициент теплообмена, Вт/м² К; β — коэффициент массоотдачи, м/с; t — текущее время тепловой обработки, с.

После замены переменных: $r = x/R$ — безразмерная пространственная координата; $T = (\theta - \theta_0) / (\theta_c - \theta_0)$, $U = u / u_0$ — соответственно безразмерные температура и влагосодержание семян; $\tau = t / t_k$ — безразмерное время, где t_k — конечное время тепловой обработки, получена безразмерная форма уравнений (1) и (2):

$$\begin{aligned}
\frac{\partial T}{\partial \tau} &= A_{11} \left(\frac{\partial^2 T}{\partial r^2} + \frac{2\partial T}{r\partial r} \right) + A_{12} \left(\frac{\partial^2 U}{\partial r^2} + \frac{2\partial U}{r\partial r} \right), \\
\frac{\partial U}{\partial \tau} &= A_{21} \left(\frac{\partial^2 T}{\partial r^2} + \frac{2\partial T}{r\partial r} \right) + A_{22} \left(\frac{\partial^2 U}{\partial r^2} + \frac{2\partial U}{r\partial r} \right), \\
r &\in [0, 1], \quad \tau \in [0, 1],
\end{aligned} \tag{4}$$

с граничными условиями:

$$\begin{aligned}
-\frac{\partial T(r, \tau)}{\partial r} \Big|_{r=1} + a_1 \left[1 - T(r, \tau) \Big|_{r=1} \right] - a_2 \left[U(r, \tau) \Big|_{r=1} - \frac{u_p}{u_0} \right] &= 0, \\
\frac{\partial U(r, \tau)}{\partial r} \Big|_{r=1} + b_1 \left[1 - T(r, \tau) \Big|_{r=1} \right] + b_2 \left[U(r, \tau) \Big|_{r=1} - \frac{u_p}{u_0} \right] &= 0,
\end{aligned} \tag{5}$$

с начальными условиями:

$$T(r, 0) = \theta_0, \quad U(r, 0) = 1, \tag{6}$$

и условием ограниченности решения:

$$|T, U| < \infty, \tag{7}$$

где комплексы критериев определяются уравнениями: $A_{11} = 1 + \varepsilon K_o L_u P_n$, $A_{12} = \varepsilon K_o L_u$, $A_{21} = L_u P_n$, $A_{22} = L_u$, $a_1 = Bi_q$, $a_2 = (1 - \varepsilon) K_o L_u Bi_m$, $b_1 = P_n Bi_q$, $b_2 = Bi_m (1 - \varepsilon) K_o L_u P_n$, а используемые критерии имеют вид: $K_o = r_0 u_0 / c_q (\theta_c - \theta_0)$ — Коссовича; $L_u = a_m / a$ — Лыкова; $P_n = \delta (\theta_c - \theta_0) / u_0$ — Поснова; $F_0 = at / R^2$ — число Фурье; теплообменный и массообменный критерии Био соответственно $Bi_q = \alpha R / \lambda$, $Bi_m = \beta R / a_m$.

После замены неизвестных функций:

$$T = \frac{Z(\tau, r)}{r}, \quad U = \frac{W(\tau, r)}{r}, \quad (Z, W) \in C(0 \leq r \leq 1), \tag{8}$$

система (2.19) относительно Z и W приняла более простую форму:

$$\begin{aligned}\frac{\partial Z}{\partial t} &= A_{11} \frac{\partial^2 Z}{\partial r^2} + A_{12} \frac{\partial^2 W}{\partial r^2}, \\ \frac{\partial W}{\partial t} &= A_{21} \frac{\partial^2 W}{\partial r^2} + A_{22} \frac{\partial^2 W}{\partial r^2},\end{aligned}\quad (9)$$

с граничными и начальными условиями:

$$\begin{aligned}-\left. \frac{\partial Z(r, \tau)}{\partial r} \right|_{r=1} + a_1 \left[1 - Z(r, t) \Big|_{r=1} \right] - a_2 \left[W(r, t) \Big|_{r=1} - \frac{u_p}{u_0} \right] &= 0, \\ \left. \frac{\partial W(r, \tau)}{\partial r} \right|_{r=1} + b_1 \left[1 - Z(r, t) \Big|_{r=1} \right] + b_2 \left[W(r, t) \Big|_{r=1} - \frac{u_p}{u_0} \right] &= 0,\end{aligned}\quad (10)$$

$$Z(0, r) = 0, \quad W(0, r) = r. \quad (11)$$

Так как функции T и U при $r \rightarrow 0$ ограничены по условию (7), то из (8) следует:

$$Z(\tau, r) \Big|_{r=0} = W(\tau, r) \Big|_{r=0} = 0, \quad (12)$$

В этом случае задача сводилась к нахождению решения системы (9), удовлетворяющего граничным условиям (10), (12) и начальным условиям (11).

Предполагалось, что решение на сферической границе зерна при $r = 1$ представлено функциями:

$$Z \Big|_{r=1} = \varphi(\tau), \quad W \Big|_{r=1} = \psi(\tau), \quad (13)$$

Для замены сложной формы граничных условий на более простые и удобные условия предложено использовать метод разложения неизвестных функций $\varphi(\tau)$, $\psi(\tau)$ в модифицированные ряды Фурье [5]. При этом задача моделирования заключалась в том, чтобы найти решение системы (9) с начальным условием (11) и граничными условиями (12), где неизвестные функции (13) следует определить так, чтобы выполнялись граничные условия (10).

При такой постановке решение задачи представлено следующими модифицированными рядами Фурье:

$$Z = M_z + \sum_{m=1}^{\infty} Z_m(\tau) \sin(m\pi r), \quad W = M_w + \sum_{m=1}^{\infty} W_m(\tau) \sin(m\pi r), \quad (14)$$

где зависимости M_z и M_w имеют вид:

$$\begin{aligned}M_z &= \varphi(\tau)r + \varphi_0(\tau) \left(\frac{r^2}{2} - \frac{r^3}{6} - \frac{r}{3} \right) + \varphi_1(\tau) \left(\frac{r^3}{6} - \frac{r}{6} \right), \\ M_w &= \psi(\tau)r + \psi_0(\tau) \left(\frac{r^2}{2} - \frac{r^3}{6} - \frac{r}{3} \right) + \psi_1(\tau) \left(\frac{r^3}{6} - \frac{r}{6} \right),\end{aligned}\quad (15)$$

Функции M_z и M_w обеспечивают равномерность сходимости разложений (14) внутри отрезка $r \in [0, 1]$ и на его границах вместе со вторыми частными производными по радиусу r включительно [6]. Выражения для Z и W в (14) вместе со вторыми частными производными при $r = 0$ и $r = 1$ обращаются в тождества. Это свойство позволило почленно дважды дифференцировать разложения (14) и подставлять их в дифференциальные уравнения (9), начальные условия (11) и граничные условия (12). Сходимость модифицированных рядов Фурье (14) имеет порядок $(\pi m)^{-5}$, где m — порядковый номер слагаемого в суммах системы (10).

В силу высокой скорости сходимости рядов Фурье (14) достаточно было ограничиться двумя первыми слагаемыми, при этом приближённое решение было получено в виде зависимостей:

$$T = \frac{1}{r} [M_z + Z_1(\tau) \sin(m\pi r)],$$

$$U = \frac{1}{r} [M_w + W_1(\tau) \sin(m\pi r)]. \quad (16)$$

Выполнена идентификация параметров модели по экспериментальным данным процесса сушки зерна рапса в барабанной сушилке. Численный эксперимент с учетом экспериментальных данных [8] обеспечил наилучшую сходимостью расчетных и экспериментальных данных с погрешностью моделирования 12,5 %.

Рассчитанные значения показателей средней ошибки аппроксимации и коэффициента детерминации для температуры не превысили значения 4 % и 0,957, а для влагосодержания — 2 % и 0,968 соответственно.

Заключение

Таким образом, математическая модель А.В. Лыкова с учетом сделанных допущений и применением *модифицированных рядов Фурье* для нахождения полей температур и влагосодержаний в единичном зерне адекватно описывает реальный процесс тепловой обработки масличных культур сферической формы и позволяет получить приближенное решение с заданной точностью.

Литература

1. *Остриков, А. Н.* Математическое моделирование процесса сушки пищевого растительного сырья перегретым паром / А. Н. Остриков, С. А. Шевцов // Изв. вузов. Пищевая технология. – 2013. – № 1. – С. 83–87.
2. *Муштаев, В. И.* Сушка дисперсных материалов / В. И. Муштаев, В.М. Ульянов. – М. : Химия, 1988. – 352 с.
3. *Остриков, А. Н.* Численно-аналитическое решение трехмерной модели нестационарного теплопереноса в процессе конвективной сушки пищевого растительного сырья / А. Н. Остриков, С. А. Шевцов, И. Н. Столяров // Изв. вузов. Пищевая технология. – 2014. – № 2–3. – С. 116–120.
4. *Лыков, А. В.* Тепломассообмен / А. В. Лыков : Справочник. – М. : Энергия, 1978. – 480 с.
5. *Чернышов, А. Д.* Метод быстрых разложений для решения нелинейных дифференциальных уравнений / А. Д. Чернышов // Журнал вычислительной математики и математической физики. – 2014. – Т. 54, № 1. – С. 11–21.
6. *Чернышов, А. Д.* Улучшение дифференцируемости решений краевых задач механики в форме обобщенных рядов Фурье с помощью граничных функций / А. Д. Чернышов // Изв. РАН. Механика твердого тела. – 2010. – № 1. – С. 151–162.
7. *Шевцов, А. А.* Компьютерное моделирование теплофизических характеристик семян рапса методом нестационарного теплового режима / А. А. Шевцов, В. В. Ткач, Н. А. Сердюкова // В сборнике: НАУКА, Образование и инновации в современном мире. Материалы национальной научно-практической конференции. – 2018. – С. 199–204.
8. *Бритиков, Д. А.* Разработка ресурсосберегающих процессов сушки зерна злаковых и масленичных культур с использованием теплонаносных технологий: дис. докт. техн. наук. – Воронеж, 2013. – С. 64–67.

Орешин Константин Вячеславович — курсант 5-го курса Военного учебно-научного центра Военно-воздушных сил «Военно-воздушной академии имени профессора Н. Е. Жуковского и Ю. А. Гагарина». E-mail: nata29m@mail.ru

Шевцов Александр Анатольевич (научный руководитель) — д-р техн. наук, проф., профессор кафедры общепрофессиональных дисциплин Военного учебно-научного центра Военно-воздушных сил «Военно-воздушной академии имени профессора Н. Е. Жуковского и Ю. А. Гагарина».

Сердюкова Наталья Алексеевна (научный руководитель) — канд. техн. наук, преподаватель кафедры общепрофессиональных дисциплин Военного учебно-научного центра Военно-воздушных сил «Военно-воздушной академии имени профессора Н. Е. Жуковского и Ю. А. Гагарина».

ПОСТАНОВКА ЗАДАЧИ МОДЕЛИРОВАНИЯ ПРОЦЕССА УПРАВЛЕНИЯ ФУНКЦИОНИРОВАНИЕМ ГОРОДСКИХ СИСТЕМ ВОДОСНАБЖЕНИЯ

С. А. Палкина

Воронежский государственный университет

Введение

Важность изучения проблем потокораспределения в городских системах водоснабжения (СВ) определяется практической значимостью получаемых решений. Итоговые прогнозы позволяют формировать планы проведения реконструкции и модернизации систем, мероприятий проводимых в случае возникновения аварийных ситуаций, ремонтов, а также обеспечения оптимальных режимов функционирования, в том числе при подключении новых потребителей. Решение таких задач опытным путем является трудоемким и ведёт к необоснованным тратам ресурсов, в том числе и времени. Более того, оно не позволяет в полной мере с большой точностью моделировать ситуации, для которых выполняется расчет (эксперимент). Активно внедряемым в проектных институтах является метод имитационного моделирования. Главное его достоинство — наглядность процесса моделирования. Основным недостатком является сложность подбора программного обеспечения, которое максимально точно позволит провести эксперимент. В связи с этим, основным методом получения прогнозов является аналитический метод. Хорошая изученность, апробирование на большом количестве задач, прозрачность расчетов и минимальное ресурсное обеспечение являются его главными достоинствами. Недостатки — сложность расчетов. В данной статье рассматривается один из подходов, решения задачи моделирования процесса управления функционированием системы подачи и распределения воды (СПРВ) при использовании аналитического метода.

1. Задача моделирования процесса управления функционированием системы подачи и распределения воды

Рассмотрим классическую задачу моделирования процесса управления, заключающуюся в расчете оптимальной нагрузки потоков водоснабжения городских систем. Пример системы подачи и распределения воды представлен на рис. 1.

Моделирование процесса управления — это задача по определению расходов водоснабжения на участках сети при заданном прогнозе потребления ресурсов сети. Процесс моделирования состояния представим в виде следующих этапов: постановка задачи и формирование модели, выбор алгоритма и расчет по нему, интерпретация результатов.

Постановка задачи. Сложность этапа связана с выбором объекта расчета: рассматриваем сеть в целом или отдельно взятый сегмент. Важность выбора связана с тем, что указываемое в модели число линейных элементов сетей соизмеримо с числом потребителей. При рассмотрении даже одного из районов города их количество исчисляется несколькими десятками тысяч. В масштабах города их будет более сотни тысяч. Описание сетей (сегментов сети) подразумевает описание большого количества элементов и их связей. Получаем, что масштаб объекта определяет размерность модели. В зависимости от типа задачи подбирается модель для описания.

Выбор алгоритма расчета. Методы и алгоритмы решения рассматриваемого типа задач на сегодняшний день хорошо изучены и вычислительной сложности не представляют. Однако при решении задачи важно прогноз получить и сделать это как можно оперативнее и точнее.

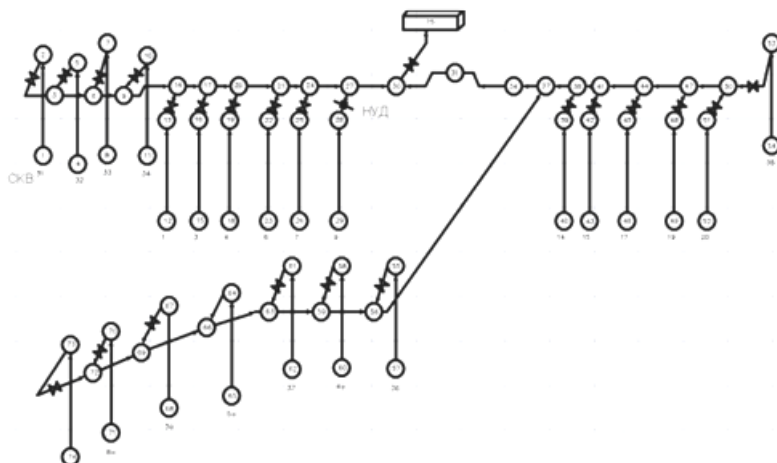


Рис. 1. Сегмент городской системы подачи и распределения воды

Это определяет значимость выбора алгоритма, позволяющего получить решение при имеющемся наборе входных данных и заданных ограничениях. Алгоритм зависит от модели, выбранной для описания задачи. Так например, если при использовании метода наименьших квадратов, который является базовым в рассматриваемых задачах, требуется задание модели в виде системы трех типов уравнений (цепные уравнения, узловые балансовые уравнения и нормальные уравнения). Более того, метод является итерационным. Следовательно, требуется подбор такого математического инструментария, который решал бы и системы разнотипных уравнений и реализовывал бы итерационный метод. При этом оперативность получения решения достигается за счет качества программной реализации предлагаемого алгоритма. И здесь следует учитывать, что реализация отдельно взятого алгоритма не позволяет учитывать необходимость унификации решения для описания сети (всякий раз необходимо будет строить свою уникальную систему уравнений) и не любой математический пакет может одинаково хорошо реализовывать и итерационные методы, и корректно работать со всеми типами систем уравнений.

Расчет по модели. Исходными данными задачи являются параметры системы и прогноз водоснабжения потребителей. Выходными данными – параметры настройки узлов сети и прогнозируемые расходы. Сопоставление получаемых расходов водопотребления с прогнозируемыми позволяет судить о том, что алгоритм позволяет обеспечить заданную точность вычислений.

Интерпретация результатов. Использование специализированных математических пакетов позволяет получать решение с очень высокой точностью, при этом и требует повышенного ресурсообеспечения. В сопоставлении с погрешностью измерения отдельных параметров оборудования СВ и точности предоставляемых исходных данных, такая точность является очень высокой, что не всегда оправдано.

2. Алгоритм моделирования процесса управления функционированием СВ

Принимая во внимание особенности каждого этапа, предлагается задачу моделирования процесса управления функционированием системы водоснабжения разбить на три подзадачи:

- 1) постановки задачи,
- 2) выбора математического аппарата,
- 3) программная реализация расчета по модели.

Согласно современной теории гидравлики задачи потокораспределения для управления водоснабжением разбиваются на две подзадачи: определения начального приближения и моделирования процесса управления водоснабжением. В свою очередь, задача моделирования

процесса управления водоснабжением разбивается на задачи определения допустимости сети (определения существования решения) и задачи поиска решения (оптимизации параметров сети). Поэтому в алгоритм, приведенный на рис. 2, добавим описанные выше подзадачи.

Сложность для реализации вычислительного алгоритма вызывается необходимостью контроля того, что вычислительный процесс не выйдет за пределы области решения задачи. Это определяет необходимость добавления в программную реализацию сервиса визуализации процессов моделирования и обработчиков событий выхода из области возможных решений. Соответственно, в алгоритм добавляются новые шаги по контролю процесса расчета. С учетом того, что программное обеспечение должно быть универсальным, то в реализации следует реализовывать сервис поддержки различных режимов ввода данных и визуализации полученного решения. Рекомендуемый алгоритм моделирования процесса управления функционированием системы водоснабжения представлен на рис. 2.

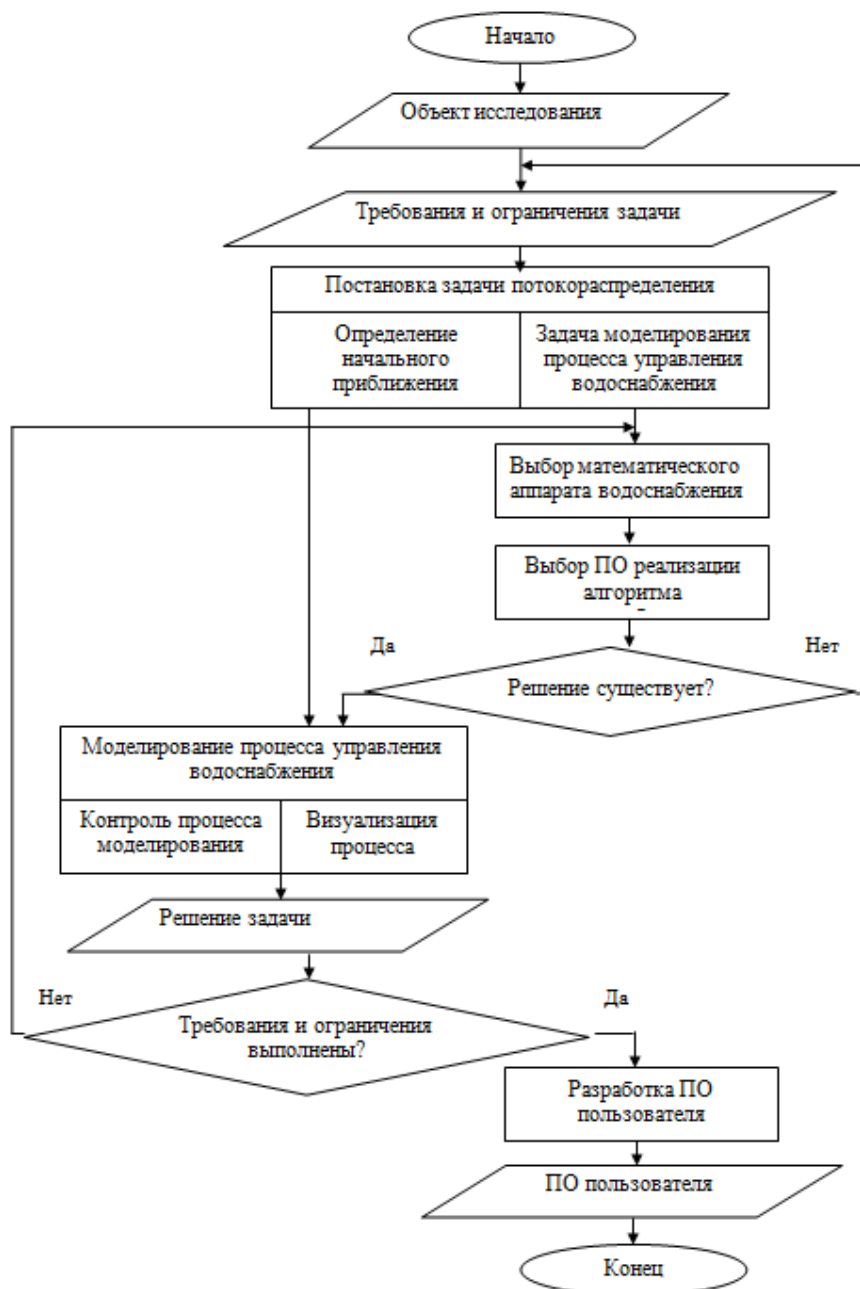


Рис. 2. Алгоритм моделирования процесса управления функционированием системы водоснабжения

Заключение

Предложенный подход к решению задачи моделирования процесса управления функционированием системы подачи и распределения воды позволяет обеспечить высокое качество реализации на всех уровнях: от разработки алгоритма до его программной реализации. Значимость правильных и быстрых расчетов в анализе систем водоснабжения, помогает избежать ряда ситуаций, которые часто несут за собой непредвиденно большие траты. Верные расчеты нагрузок, произведенные с учетом особенностей объекта и применяемых технологий, позволяют оптимально расходовать природные, финансовые и человеческие ресурсы.

Алгоритм, предложенный в работе позволит разработать ПО с учетом как технических, так и математических аспектов моделирования процесса управления функционированием систем подачи и распределения воды. В свою очередь, это позволит получать прогнозы с минимальной затратой времени и требуемой точностью.

Литература

1. Лукьяненко В. И., Мартыненко Г. Н., Панов М. Я. Оперативное управление функционированием городских систем газоснабжения // Физико-технические проблемы энергетики, экологии и энергоресурсосбережения. Труды научно-технической конференции молодых ученых, аспирантов и студентов: сб. науч. тр. – Воронеж, ВГТУ, 2012. – С. 89–94.

2. Панов М. Я. Модели управления функционированием систем подачи и распределения воды [Текст] : монография / М. Я. Панов, Ю. Ф. Петров, В. И. Щербаков ; М-во образования и науки РФ, Федеральное гос. бюджетное образовательное учреждение высш. проф. образования «Воронежский гос. архитектурно-строительный ун-т». – Воронеж : Воронежский ГАСУ, 2013. – 271 с.

3. Использование системы Matlab для моделирования процесса управления функционированием систем газоснабжения / М. Я. Панов М.Я [и др.] // Известия высших учебных заведений. Технология текстильной промышленности. 2017. – № 1 (367). – С. 239–243.

Палкина София Александровна — студентка 3-го курса кафедры механики и компьютерного моделирования Воронежского государственного университета. E-mail: mail.soffya@mail.ru

Минаева Надежда Витальевна (научный руководитель) — д-р физ.-мат. наук, профессор кафедры механики и компьютерного моделирования Воронежского государственного университета. E-mail: nminaeva@yandex.ru

РАСПРЕДЕЛЕНИЕ ПОЛЯРИЗАЦИИ В СФЕРИЧЕСКОЙ СЕГНЕТОЭЛЕКТРИЧЕСКОЙ НАНОЧАСТИЦЕ

И. А. Панков, Д. Р. Юнда, А. В. Шуба

*Военный учебно-научный центр Военно-воздушных сил
«Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина»*

Введение

Зачастую при моделировании физических свойств наноразмерных сегнетоэлектрических образцов с ограниченной геометрией или нанокомпозитов на основе диэлектрической матрицы с сегнетоактивными включениями используется их аппроксимация точечными диполями [1]. При этом совершенно не учитывается распределение поляризации по объему нанообъектов и их взаимодействие с окружающей матрицей. Для учета этих факторов часто применяют численные методы решения, например, метод конечных элементов [2] или метод молекулярной динамики [3]. Однако с одной стороны для проведения расчетов с достаточно большим количеством частиц требуются серьезные вычислительные ресурсы, с другой — не ясны пределы применимости таких результатов и вытекающие отсюда слабые предсказательные возможности поведения физических свойств данных неоднородных материалов. Следует отметить, что для продвижения аналитически в решении подобного рода задач часто используют вариационный метод [4], основанный на выборе пробной функции поляризации с неизвестной амплитудой, подлежащей определению путем минимизации свободной энергии системы. Но такой выбор координатной зависимости поляризации несет в себе некий произвольный характер и может быть обоснован, например, исходя из наименьшего значения свободной энергии для соответствующей пробной функции, найденной путем перебора гладких функций с максимумом в центре частицы. Ясно, что такой перебор пробных функций с последующим нахождением вариационного параметра не может гарантировать установление истинной зависимости равновесного распределения поляризации в наночастице.

1. Постановка задачи

В данной статье предлагается численно-аналитический метод определения координатной зависимости параметра порядка внутри сегнетоэлектрической частицы с ограниченной геометрией путем кусочно-параболической аппроксимации ее свободной энергии.

В качестве иллюстрации данного метода рассмотрим отдельную химически однородную гранулу со сферической симметрией радиуса R . Свободная энергия такой частицы, имеющей фазовый переход второго рода, в рамках теории Ландау — Гинзбурга [5] имеет вид

$$F = \int_V \left\{ F_\infty + \frac{\kappa}{2} (\nabla P)^2 - \mathbf{E}_{ext} \mathbf{P} \right\} dV + \int_S \frac{\alpha_s}{2} \mathbf{P}^2 dS, \quad (1)$$

где $\mathbf{P} = \{0, 0, P(\rho)\}$ — вектор поляризации, играющий роль параметра порядка; \mathbf{E}_{ext} — вектор напряженности внешнего электрического поля; $\kappa \approx a^2$ корреляционная постоянная, где a — межатомное расстояние; α_s — коэффициент при квадратичном слагаемом в поверхностной энергии; F_∞ — однородная часть свободной энергии, равная

$$F_\infty = -\frac{\alpha}{2} \bar{P}^2 + \frac{\beta}{4} \bar{P}^4, \quad (2)$$

где $\alpha = \alpha_0(T_C - T)$, β — коэффициенты в разложении свободной энергии вблизи температуры Кюри T_C объёмного образца. Из условия минимума функционала (1) получаем уравнение равновесия на поляризацию

$$\kappa \Delta P + \alpha P - \beta P^3 + E_{ext} = 0 \quad (3)$$

с граничным условием

$$\kappa \nabla P + \alpha_s P \Big|_{\rho=R} = 0. \quad (4)$$

Решить нелинейную краевую задачу (3), (4) аналитически не представляется возможным. Для ее упрощения с сохранением нелинейности представим однородную часть свободной энергии (2) в виде двух парабол $F^I(P)$ и $F^{II}(P)$ (рис. 1), полученными разложением функции (2) в ряд по поляризации вблизи ее экстремумов $P = 0$ и $P = P_\infty$:

$$\begin{cases} F^I(P) = F_0 + \alpha(P - P_\infty)^2 - E_{ext}P, & |P| > P^*, \\ F^{II}(P) = -\frac{\alpha}{2}P^2 - E_{ext}P, & |P| \leq P^*, \end{cases} \quad (5)$$

где P_∞ — спонтанная поляризация объёмного образца, P^* — точка соединения парабол.

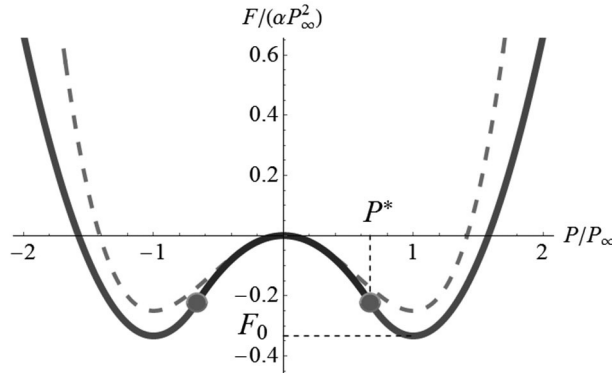


Рис. 1. Аппроксимация свободной энергии системой парабол (5), штриховая кривая — точный вид свободной энергии (2)

В выражении (5) параметры F_0 и P^* определяем из условий гладкой «сшивки» парабол:

$$\begin{cases} F^I(P^*) = F^{II}(P^*), \\ \left. \frac{\partial F^I}{\partial P} \right|_{P=P^*} = \left. \frac{\partial F^{II}}{\partial P} \right|_{P=P^*}, \end{cases}$$

откуда

$$\begin{cases} P^* = 2P_\infty / 3, \\ F_0 = -\alpha P^2 / 3. \end{cases} \quad (6)$$

Тогда нелинейное уравнение равновесия (3) для функционала (1) при замене выражения (2) кусочной функцией (5) в сферических координатах эквивалентно линейным неоднородным дифференциальным уравнениям для соответствующих областей частицы:

$$\kappa \left(\frac{d^2 P^I}{d\rho^2} + \frac{2}{\rho} \frac{dP^I}{d\rho} \right) - 2\alpha(P^I - P_\infty) + E_{ext} = 0, \quad \rho \leq \rho^*; \quad (7)$$

$$\kappa \left(\frac{d^2 P^{II}}{d\rho^2} + \frac{2}{\rho} \frac{dP^{II}}{d\rho} \right) + \alpha P^{II} + E_{ext} = 0, \quad \rho^* < \rho \leq R, \quad (8)$$

где ρ^* — абсцисса соединения функций P^I и P^{II} , отвечающая P^* . Отнеся все величины с размерностью длины к межатомному расстоянию a , параметр порядка P и внешнее электрическое поле E_{ext} — к поляризации объемного кристалла P_∞ , запишем общее решение уравнений (7), (8) в безразмерных переменных:

$$\tilde{P}^I(\tilde{\rho}) = C_1 \frac{I_{1/2}(\sqrt{2\alpha}\tilde{\rho})}{\sqrt{\tilde{\rho}}} + 1 + \frac{\tilde{E}_{ext}}{2\alpha}, \quad \tilde{\rho} \leq \tilde{\rho}^*; \quad (9)$$

$$\tilde{P}^{II}(\tilde{\rho}) = C_2 \frac{J_{1/2}(\sqrt{\alpha}\tilde{\rho})}{\sqrt{\tilde{\rho}}} + C_3 \frac{N_{1/2}(\sqrt{\alpha}\tilde{\rho})}{\sqrt{\tilde{\rho}}} - \frac{\tilde{E}_{ext}}{\alpha}, \quad \tilde{\rho}^* < \tilde{\rho} \leq \tilde{R}, \quad (10)$$

где $I_{1/2}(x)$ — модифицированная функция Бесселя первого рода (функция Инфельда) порядка $1/2$; $J_{1/2}(x)$ и $N_{1/2}(x)$ — функции Бесселя и Неймана порядка $1/2$ соответственно [6]. C_1 , C_2 , C_3 — константы интегрирования. В функции $\tilde{P}^I(\tilde{\rho})$ вторая константа интегрирования была принята равной нулю по причине естественной ограниченности поляризации в центре частицы. Постоянные C_1 , C_2 , C_3 и точка $\tilde{\rho}^*$ определяем из условий гладкой «сшивки» функций (9), (10) с учетом граничного условия (4) для функции (10). Полная система перечисленных соотношений в безразмерных переменных имеет вид

$$\begin{cases} \tilde{P}^I(\tilde{\rho}^*) = 2/3, \\ \tilde{P}^{II}(\tilde{\rho}^*) = 2/3, \\ \left. \frac{d^2 \tilde{P}^I}{d\tilde{\rho}^2} \right|_{\tilde{\rho}=\tilde{\rho}^*} = \left. \frac{d^2 \tilde{P}^{II}}{d\tilde{\rho}^2} \right|_{\tilde{\rho}=\tilde{\rho}^*}, \\ \left. \tilde{\alpha}_s \tilde{P}^{II} + \frac{d^2 \tilde{P}^{II}}{d\tilde{\rho}^2} \right|_{\tilde{\rho}=\tilde{R}} = 0. \end{cases} \quad (11)$$

2. Решение задачи и анализ результатов

Численно решив систему трансцендентных уравнений (11), и, подставив найденные значения C_1 , C_2 , C_3 , $\tilde{\rho}^*$ в выражения (9), (10), получим аналитические зависимости для распределения поляризации в центральном и пограничном слое гранулы (рис. 2).

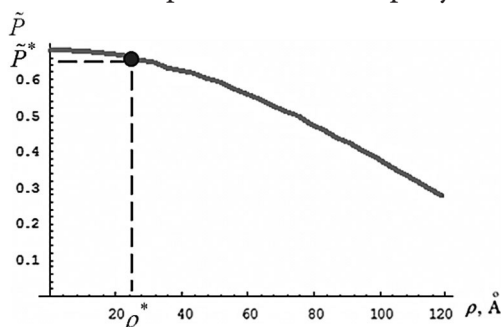


Рис. 2. Распределение поляризации в сегнетоэлектрической наночастице вблизи температуры фазового перехода

Здесь и далее для проведения расчетов использованы параметры триглицинсульфата (ТГС): $T_C = 332$ К, $\alpha_0 = 3.92 \cdot 10^{-3}$ К $^{-1}$ [7]. Рис. 3 изображает распределение поляризации в зависимости от параметров частицы — размера, типа «закрепления» поляризации на границе, задаваемого параметром α_s , температуры. С усилением «закрепления» поляризации на гра-

нице частицы с ростом параметра α_s повышается вклад энергии поверхности в свободную энергию (1). Для его уменьшения требуется снижение скачка поляризации на границе частицы за счет роста градиента поляризации в приповерхностном слое гранулы, что повышает энергию неоднородности (градиентный член в выражении (1)). Конкуренция этих двух энергий и приводит к равновесному распределению поляризации по сечению гранулы вблизи температуры фазового перехода T_p (рис. 3а). С уменьшением размера частицы (рис. 3б) при постоянном «закреплении» поляризации на ее границе ($\alpha_s = a$) для снижения скорости изменения поляризации, определяющей энергию неоднородности, система повышает энергию поверхности (скачок поляризации растет), что в целом энергетически более выгодно.

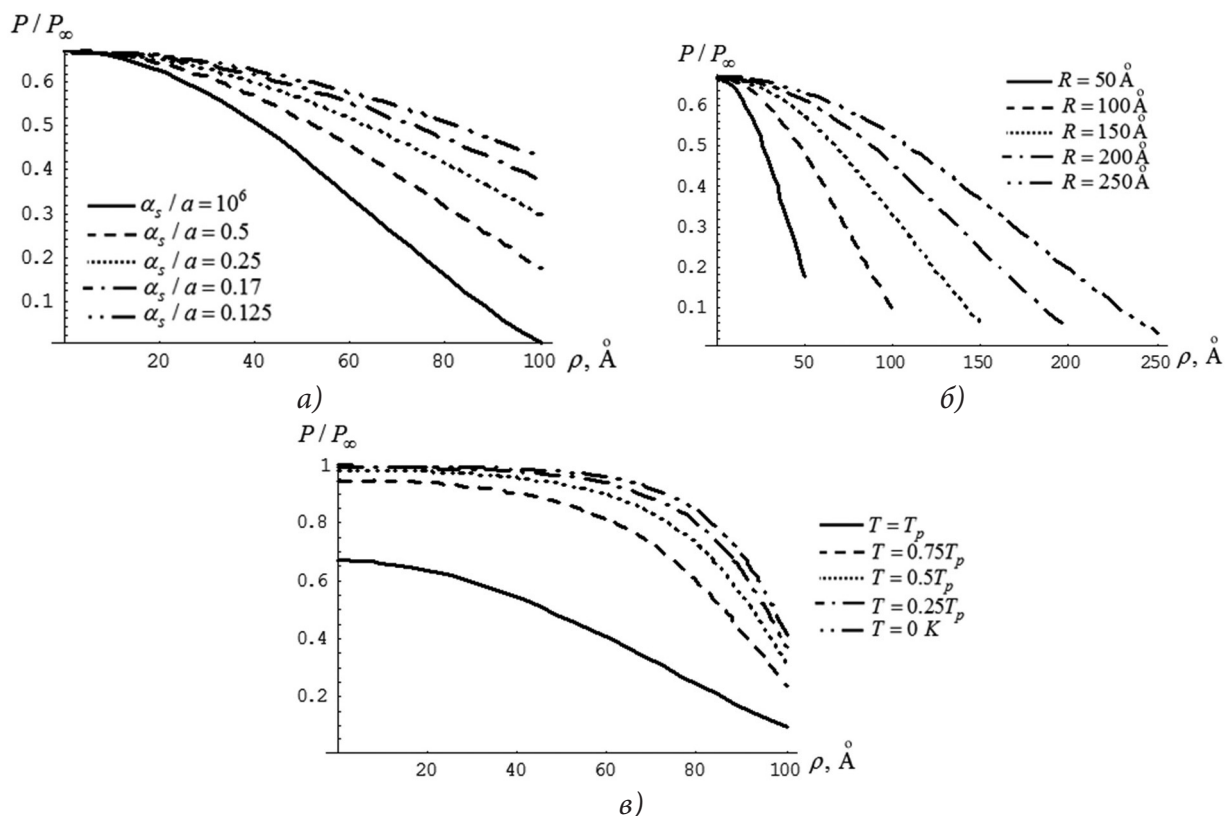


Рис. 3. Профили поляризации в сегнетоэлектрической частице:
 а) $R = 100 \text{\AA}$, $T \approx T_p$; б) $\alpha_s = a$, $T \approx T_p$; в) $R = 100 \text{\AA}$, $\alpha_s = a$

При охлаждении гранулы объемная часть свободной энергии (1) увеличивается за счет коэффициента α в выражении (2) и поляризации энергетически выгодно увеличиваться (рис. 3в). С физической точки зрения снижение тепловых флуктуаций в кристалле повышает электрическое взаимодействие в соседних диполях, что и приводит к росту поляризации.

Установленная координатная зависимость поляризации в сегнетоэлектрической частице позволяет определить усредненное по объему значение поляризации \mathcal{P} и свободной энергии \mathcal{F} частицы (рис. 4), где $P_\infty = \sqrt{\alpha/\beta}$, $F_\infty = -\alpha/(4\beta)$ — соответственно поляризация и свободная энергия единицы объема неограниченного однородного образца.

Заключение

В настоящей работе рассмотрен численно-аналитический метод расчета координатной зависимости поляризации в сегнетоэлектрической частице с фазовым переходом второго рода, основанный на кусочно-параболической аппроксимации ее свободной энергии.

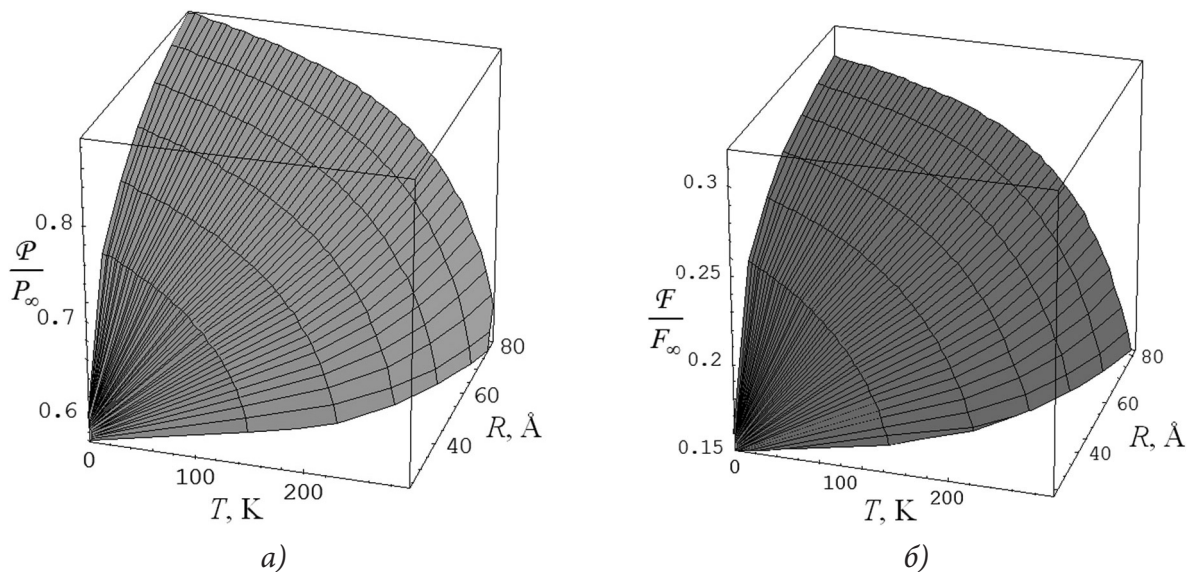


Рис. 4. Зависимости нормированных средних значений поляризации \mathcal{P} / P_{∞} (а) и свободной энергии \mathcal{F} / F_{∞} (б) частицы от её радиуса R и температуры

На следующем этапе исследований планируется расчет температурных зависимостей теплоемкости и диэлектрической проницаемости сегнетоэлектрических гранул в зависимости от их параметров для исследования влияния размерных и поверхностных факторов на их физические свойства.

Литература

1. Пирозерский, А. Л. Влияние геометрии сетки пор на фазовый переход в сегнетоэлектрике, заполняющем пористую матрицу / А. Л. Пирозерский, Е. В. Чарная, С. Тien // Физика твердого тела. – 2007. – Т. 49, Вып. 2. – С. 327–330.
2. Бубликов, К. В. Влияние металла на характеристики поперечных мод гибридных волн в слоистой структуре феррит-сегнетоэлектрик / К. В. Бубликов, А. В. Садовников, Е. Н. Бегинин, Ю. П. Шараевский, С. А. Никитов // Письма в журнал технической физики. – 2016. – Т. 42, Вып. 12. – С. 88–96.
3. Редьков, А. В. Моделирование процесса идентификации наномасштабных пленок на подложке методом молекулярной динамики / А. В. Редьков, А. В. Осипов, С. А. Кукушкин // Письма в журнал технической физики. – 2016. – Т. 42, Вып. 12. – С. 64–72.
4. Федорова, В. Ю. Функция распределения электронной плотности для металлических наночастиц в рамках теории функционала плотности / В. Ю. Федорова // Политехнический молодежный журнал. – 2018. – № 8. – С. 1–8.
5. Ландау, Л. Д. Теоретическая физика. Т.8. Электродинамика сплошных сред / Л. Д. Ландау, Е. М. Лифшиц. – Москва : Физматлит, 2005. – 656 с.
6. Бейтмен, Г. Высшие трансцендентные функции. Т. 2. Функции Бесселя, функции параболического цилиндра, ортогональные многочлены / Г. Бейтмен, А. Эрдейи. – Москва : Наука, 1966. – 297 с.
7. Цедрик, М. С. Физические свойства кристаллов семейства триглицинсульфата (в зависимости от условий выращивания) / М. С. Цедрик. – Минск : Наука и техника, 1986. – 216 с.

Панков Илья Андреевич — курсант 1-го курса кафедры эксплуатации комплексов авиационного вооружения (и прицельных систем) ВУНЦ ВВС «ВВА».

Юнда Даниил Романович — курсант 1-го курса кафедры эксплуатации комплексов авиационного вооружения (и прицельных систем) ВУНЦ ВВС «ВВА».

Шуба Андрей Витальевич (научный руководитель) — канд. физ.-мат. наук, доц., профессор кафедры математики ВУНЦ ВВС «ВВА». E-mail: shandvit@rambler.ru

МОДЕЛЬ ПОТОКА ИДЕАЛЬНОЙ ЖИДКОСТИ В ОКРЕСТНОСТИ ЦИЛИНДРА С ОТВЕРСТИЯМИ

Я. Д. Перепелов

*Военный учебно-научный центр Военно-воздушных сил
«Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина»*

Введение

Исследуется модельная задача о плоском течении идеальной жидкости в окрестности цилиндра малого радиуса с расположенными на его боковой поверхности несколькими отверстиями. Для данной задачи строится математическая модель плоского течения идеальной жидкости в круге с заданным, осесимметрично распределённым, расходом на внешней границе круга, и дискретно распределёнными стоками в круге.

1. Постановка задачи

Будем часть стоков располагать с постоянным окружным (угловым) шагом, и некоторым шагом по радиусу так, что все эти стоки расположены на окружностях. Однако, начиная с некоторой окружности, из-за конечных размеров стоков, их невозможно расположить так, чтобы угловой шаг совпадал с предыдущим. Таким образом, технологически невозможно осуществить осе симметричность течения в круге. Предполагается достижение равномерности расхода по площади круга путем размещения в точках M_k не просто стоков, а нескольких стоков (одного, двух, трёх и т. д.) расположенных на окружности малого радиуса ρ . На рис. 1 представлены схемы расположения локальных стоков, которые образуют сложный сток k -го порядка.

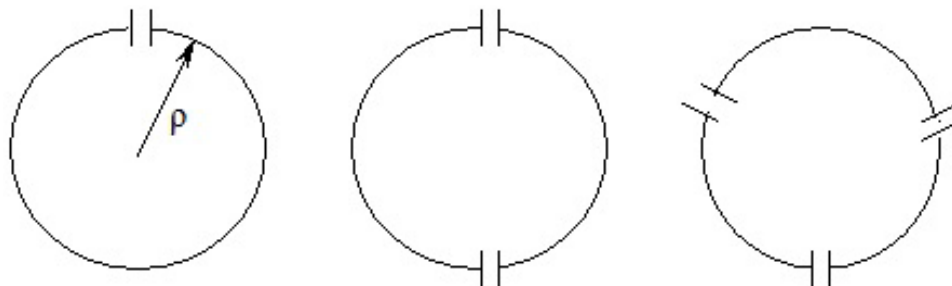


Рис. 1. Схематическое расположение нескольких стоков на окружности

Несимметричность обтекания цилиндра достигается наличием циркуляции вокруг цилиндра за счет несимметричного расположения локальных стоков относительно набегающего потока, что позволяет поворачивать поток за рядом цилиндров, и добиваться центр симметричности течения.

Математическая модель

Плоское течение идеальной жидкости в круге со многими жесткими контурами может быть описано теорией функции комплексных переменных для комплексного потенциала: $W = \Phi + i\Psi$, где Φ и Ψ являются гармоническими функциями удовлетворяющие уравнению Лапласа: $\Delta\Phi = 0$, $\Delta\Psi = 0$, и граничным условиям не протекания на жестких контурах.

Вследствие многосвязности области течения решение такой задачи не представляется возможным, поэтому сформулируем некоторые гипотезы, которые позволят построить приближённые решения или алгоритм определения положения и разворота сложных стоков:

- 1) мощность локальных стоков: $q = \frac{Q}{k \cdot N}$ существенно меньше мощности источника Q , т. е. $\frac{1}{k \cdot N} \ll 1$. (Здесь N — число сложных стоков, k — число локальных стоков в сложном стоке);
- 2) сложные стоки не влияют друг на друга так, что все сложные стоки, расположенные на некоторой окружности, обтекаются одинаковым потоком с некоторой скоростью;
- 3) сложные стоки, расположенные вниз по течению, не влияют на стоки, расположенные вверх по течению;
- 4) множество сложных стоков, расположенных на окружности, изменяет поле скоростей за ними за счёт уменьшения расхода, равномерно распределённого по окружности, и равномерно распределённой циркуляции, за счёт разворота сложных стоков.

С учетом принятых допущений выражение для комплексного потенциала течения принимает вид:

$$W = W_{ист} + \sum W_{сток} + \sum W_{дип} + \sum W_{вихря}, \quad (1)$$

где $W_{ист}$ — комплексный потенциал осе симметричного течения, обусловленного равномерно распределёнными источниками на окружности радиуса R ; $W_{сток}$ — комплексный потенциал стоков интенсивности q на окружности радиуса ρ , с центром в точках M_k ; $W_{дип}$ — комплексный потенциал диполей в точке M_k , обуславливающих наличие нулевых линий тока в виде окружностей радиуса ρ , с центром в точках M_k , которые можно отождествить с жесткими контурами; $W_{вихря}$ — комплексный потенциал вихрей, обусловленный несимметричностью обтекания сложных стоков на окружности радиуса ρ .

Комплексный потенциал стока в центре круга (распределённые источники по дуге круга):

$$W_{ист} = -\frac{Q}{2\pi} \ln z. \quad (2)$$

Комплексный потенциал стока в $\begin{pmatrix} j \\ mn \end{pmatrix}$ -й точке:

$$W_{mn}^j = -\frac{q_{mn}^j}{2\pi} \ln(z - z_{mn}^j). \quad (3)$$

Комплексный потенциал вихря в цилиндре:

$$W_{вихря} = -\frac{i\Gamma_{mn}}{2\pi} \ln(z - z_{mn}). \quad (4)$$

Комплексный потенциал диполя:

$$W_{дип} = -\frac{M_{mn} e^{i\alpha_{mn}}}{2\pi(z - z_{mn})}. \quad (5)$$

Из [1, 2] известна связь между моментом диполя, скоростью набегающего потока и радиусом цилиндра в точке M_{mn} :

$$M_{mn} = 2\pi\rho^2 |U_{mn}|. \quad (6)$$

При несимметричном обтекании цилиндра с одним локальным стоком, часть жидкости уйдёт по наискратчайшей траектории в локальный сток, и тем самым вокруг цилиндра образуется циркуляция:

$$\Gamma_{m+1n} = \frac{q_{m+1n}}{S} \rho \left(\frac{\pi}{2} - \theta_{m+1n} - \chi_{n+1n} \right), \quad (7)$$

где S — площадь поперечного сечения локального стока.

Рассмотрим комплексный потенциал течения вблизи стока в точке M_k . Подставляя в (1), (2)–(5), с учетом (6), (7) получим для сложного стока с одним локальным стоком выражение

$$W = -\frac{Q}{2\pi} \ln z - \frac{\rho^2 |U_m| e^{i(\gamma_{mn} + \chi_{mn})}}{(r - r_m) e^{i(\gamma - \gamma_n)}} - \frac{q_{mn}}{2\pi} \ln \left[(r - r_m) e^{i(\gamma - \gamma_n)} + \rho e^{i\theta_{mn}} \right] - \frac{i q_{mn} \rho \left(\frac{\pi}{2} - \theta_{mn} - \chi_{mn} \right)}{2\pi S} \ln \left((r - r_m) e^{i(\gamma - \gamma_n)} \right).$$

Разложим комплексный потенциал локального стока в ряд по малому параметру ρ , ограничиваясь первыми двумя членами ряда, получим комплексный потенциал течения вблизи сложного стока:

$$W = -\frac{Q}{2\pi} \ln z - \frac{\rho^2 |U_m| e^{i(\gamma_{mn} + \chi_{mn})}}{(r - r_m) e^{i(\gamma - \gamma_n)}} - \frac{q_{mn}}{2\pi} \ln \left[(r - r_m) e^{i(\gamma - \gamma_n)} \right] - \frac{\rho q_{mn} e^{i\theta_{mn}}}{2\pi (r - r_m) e^{i(\gamma - \gamma_n)}} - \frac{i \rho q_{mn} \left(\frac{\pi}{2} - \theta_{mn} - \chi_{mn} \right)}{2\pi S} \ln \left((r - r_m) e^{i(\gamma - \gamma_n)} \right).$$

Аналогично можно получить выражения для комплексного потенциала в случаях 2-х и 3-х локальных стоков.

В случаи нерегулярного расположения сложных стоков для создания радиального течения за цилиндром необходимо развернуть сложный сток на некоторый угол θ (рис. 2). Угол χ ориентации скорости течения жидкости в окрестности сложного стока может быть найден из выражения для функции тока, заданной в цилиндрических координатах:

$$\operatorname{tg} \chi = \frac{\partial \Psi}{\partial r}, \quad \Psi = U \left(r - \frac{\rho^2}{r} \right) \sin \gamma. \quad (8)$$

Дифференцируя (8) по радиусу, получим:

$$\operatorname{tg} \chi_{\bar{m}+1n} = U_{\bar{m}n} \left(1 + \frac{\rho^2}{r_{\bar{m}+1n}^2} \right) \sin \gamma.$$

Угол θ , поворота сложного стока определяется из условия поворота потока таким образом, чтобы критическая точка C лежала на радиус-векторе. Для этого воспользуемся выражением для расположения критической точки C на контуре цилиндра при обтекании его набегающим потоком с циркуляцией Γ [1, 2] (рис. 2).

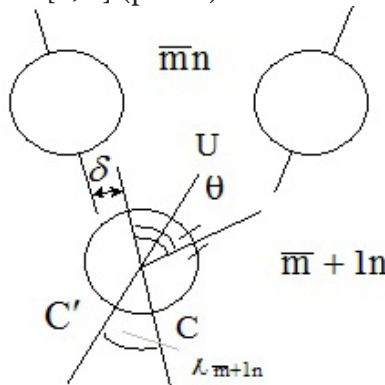


Рис. 2. Схематическое изображение нерегулярного обтекания цилиндра

Координаты критических точек на контуре цилиндра определяются выражением:

$$z_{1,2} = \frac{i\Gamma}{4\pi U} \pm \sqrt{-\left(\frac{\Gamma}{4\pi U} \right)^2 + \rho^2} = \xi \pm i\eta, \\ \Gamma < 4\pi U \rho.$$

Из геометрии расположения критических точек на контуре можно записать:

$$\operatorname{tg}(\pi - \chi) = -\operatorname{tg} \chi = \frac{2\Gamma}{\sqrt{(4\pi\rho U)^2 - \Gamma^2}}. \quad (9)$$

Система уравнений (7), (9) является нелинейной системой относительно Γ и θ , и после исключения Γ можно найти угол ориентации локального стока.

Предположим, что на некотором слое $m+1$ происходит сдвиг равномерной окружной сетки на линейную величину δ (рис. 2). Тогда в $m+1$ слое происходит поворот потока на угловую величину χ_{m+1} , где

$$\chi \approx \frac{\delta \left(\frac{r\Delta\gamma}{2} - \delta \right)}{\Delta r^2}, \quad (\delta < \Delta r). \quad (10)$$

Подставим выражение для циркуляции (7) в (9) получим для угла θ :

$$\theta = \frac{4\pi \operatorname{tg} \chi}{\sin \theta \sqrt{4 + \operatorname{tg}^2 \chi}} + \chi, \quad (11)$$

где χ определяется из выражения (10).

Уравнение (11) является нелинейным уравнением относительно θ , для решения которого воспользуемся численным методом.

Перепишем уравнение (11) в более удобной форме:

$$f(\theta) = (\theta - \chi) \sin \theta - \frac{4\pi \operatorname{tg} \chi}{\sqrt{4 + \operatorname{tg}^2 \chi}} = 0. \quad (12)$$

Для решения уравнения (12) воспользуемся методом Ньютона:

$$\theta_{n+1} = \theta_n - \frac{f(\theta_n)}{f'(\theta_n)}, \quad f'(\theta_n) \neq 0, \quad n = 0, 1, 2, \dots$$

Аналогичные вычисления проведены для случая 2-х и 3-х локальных стоков.

Результаты численного решения уравнения (12) для θ при $k=1$ и аналогичных уравнений для θ в случае $k=2, 3$ представлены на рис. 3.

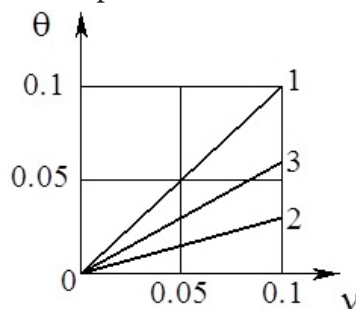


Рис. 3. График зависимости угла поворота сложного стока от угла ориентации скорости течения жидкости в окрестности сложного стока

Заключение

Как следует из графиков $\theta_1, \theta_2, \theta_3$ на рис. 3 для поворота потока до радиального необходимо разворачивать сложный сток с одним отверстием на меньший угол, чем с двумя, тремя и т. д. Следовательно, наибольшее влияние на поворот потока оказывают сложные стоки с нечетным числом локальных стоков по сравнению с четным.

Литература

1. *Кочин, Н. Е.* Теоретическая гидромеханика / Н. Е. Кочин, И. А. Кибель, Н. В. Розе. – Москва : Физматгиз, 1963. – 584 с.
2. *Милн-Томсон, Л.* Теоретическая гидродинамика / Л. Милн-Томсон – Москва : Мир, 1964. – 655 с.

Перепелов Ярослав Денисович — курсант 2-го курса ВУНЦ «ВВС ВВА им. проф. Н. Е. Жуковского и Ю. А. Гагарина» (г. Воронеж)

Фролов Александр Леонидович (научный руководитель) — канд. физ.-мат. наук, доц., профессор кафедры математики ВУНЦ «ВВС ВВА им. проф. Н. Е. Жуковского и Ю. А. Гагарина» (г. Воронеж). E-mail: al-frol@yandex.ru

ПРОВЕДЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ ДЛЯ УВЕЛИЧЕНИЯ ТОЧНОСТИ МОДЕЛИ МАШИННОГО ОБУЧЕНИЯ

Д. А. Петрова, И. Л. Каширина

Воронежский государственный университет

Введение

Интеллектуальный анализ данных — это способ анализа данных, предназначенный для поиска ранее неизвестных закономерностей в больших массивах информации. В основном используется отраслями, обслуживающими потребителей, в том числе в сфере розничной торговли, в финансах и маркетинге. Например, у Сбера существует сервис «Сбор Аналитика», который предоставляет данные по отраслям рынка или территориям на основе анализа денежных потоков населения, продаж товаров и услуг и прочих параметров. Его могут использовать как компании, так и госорганы, чтобы оценить потенциал развития региона.

Торговым сетям добыча данных позволяет анализировать покупательские корзины, чтобы улучшать рекламу, создавать запасы товаров на складах и планировать, как их разложить на витринах, открывать новые магазины и выявлять потребности разных категорий клиентов.

В нашем случае, мы будем рассматривать анализ данных сервиса виртуальных досок Avito. Сегодня виртуальные доски объявлений доступны каждому, у кого есть доступ к сети Интернет. Они могут иметь как узкую специализацию, принимая объявления только конкретных тематик (одежда, мебель и так далее), так и широкую направленность, охватывая большое количество разделов и освещая практически все сферы деятельности. В настоящий момент в сети существует свыше десяти тысяч различных сайтов для объявлений и все они отличаются по размеру, функционалу и способу взаимодействия с пользователями. Однако их суть сводится к одному: чтобы разместить объявление, каждый человек может зайти на сайт и заполнить соответствующую форму, указав свои контакты, текст объявления, при необходимости, уточнив некоторые условия или пожелания, а также приложить изображения для лучшего понимания сути.

Цель предлагаемого исследования заключается в анализе и выявлении дополнительных полезных признаков из сырого набора данных и построении эффективной модели регрессии для прогнозирования вероятности продажи товара на виртуальной доске объявлений. В рамках исследования описываются исходные данные, анализируются зависимости целевой переменной и на основе интеллектуального анализа данных формируются дополнительные признаки, с учетом которых строится несколько моделей машинного обучения на основе градиентного бустинга деревьев решений для прогнозирования спроса, определяется наиболее эффективная модель и проводится анализ ее результатов. Описанные задачи реализованы на языке программирования Python с помощью различных его библиотек на платформе для интерактивных вычислений Jupyter Notebook.

1. Исходные данные

В рамках исследования используются данные, предоставленные крупнейшим интернет-сервисом для размещения объявлений в России — Avito в рамках соревнования, где предлагается построить модель, способную наиболее точно спрогнозировать спрос на товар объявления на основе его полного описания (заголовков, описание, изображения и так далее), географического контекста и исторического спроса для похожих объявлений в аналогичном контексте [1].

Всего исходная выборка содержит 1503424 уникальных объявлений, каждое из которых описывается признаками, приведенными в табл. 1.

Таблица 1

Признаки объявлений

Тип признаков	Наименования признаков
Вещественные	Цена, количество объявлений продавца, дата объявления, рейтинг изображения
Категориальные	Номер продукта, номер пользователя, регион, город, категория продукта, подкатегория продукта, параметр 1, параметр 2, параметр 3, тип продавца
Текстовые	Заголовок объявления, описание объявления

Рейтинг изображения — это показатель качества и информативности изображения в объявлении, который определяет компания Avito. Признаки «Параметр 1», «Параметр 2» и «Параметр 3» — опциональные критерии для дополнительного, уточняющего описания предмета объявления (тип товара, его габариты или размеры, модельный ряд). Для каждой подкатегории продукта они разные, также могут встречаться пустые значения (в случае, если пользователь не стал указывать параметр или товар вовсе не описывается какими-либо измерениями).

Целевая переменная — вероятность сделки, описывает шанс продажи товара, описанного в заданном объявлении. Таким образом, значение прогноза будет варьироваться в промежутке от нуля до единицы. Описательная статистика целевой переменной приведена в табл. 2.

Таблица 2

Описательная статистика целевой переменной

Характеристика	Значение
Количество	1503424
Среднее значение	0.139
Стандартное отклонение	0.260
Минимум	0
Нижний квартиль	0
Медиана	0
Верхний квартиль	0.151
Максимум	1

Визуальный анализ данных проводился с помощью библиотек Matplotlib и Seaborn [2]. Для наглядности была построена гистограмма распределения целевой переменной (рис. 1). С помощью нее можно отметить, что распределение целевой переменной преобладает в значениях, близких к нулю, имеет заметную асимметрию с правосторонней скошенностью и некоторый «всплеск» значений около 0.8.



Рис. 1. Гистограмма распределения целевой переменной

2. Интеллектуальный анализ данных

2.1. Извлечение признаков из числовых, категориальных и текстовых данных

Для извлечения новых признаков из уже имеющихся необходимо выдвинуть гипотезы, какой дополнительный признак может потенциально повлиять на целевую переменную.

Первая гипотеза: «Влияет ли дата на спрос?». Предположим, что да, ведь покупательская способность посетителей сервиса может зависеть от дня месяца, когда они получают начисление заработной платы или от их свободного времени: они могут заходить чаще и планировать свои покупки в выходные. В таком случае, для дальнейшего анализа и проверки выдвинутой гипотезы из даты объявления вычлняются еще несколько отдельных признаков — день недели и день месяца. На рис. 2 приведен график средней вероятности сделки по каждому дню месяца. Очень маленькие значения у 30 и 31 дня обусловлены спецификой предоставленного набора данных — в этот день было выставлено очень мало объявлений и спрос на них был невелик.

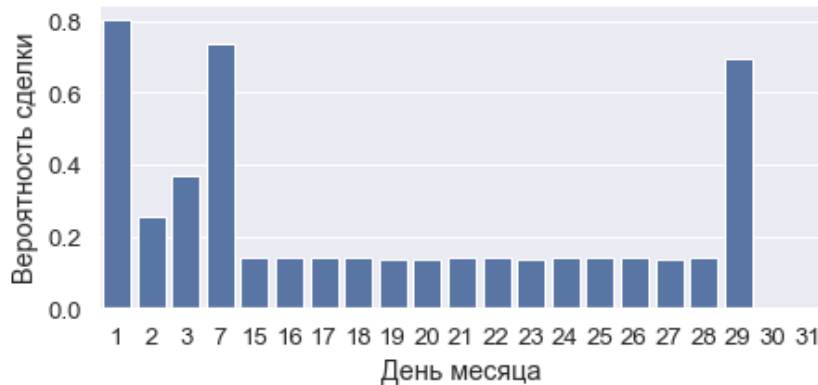


Рис. 2. Средняя вероятность сделки по каждому дню месяца

Вторая гипотеза звучит так «Если все товары одного типа имеют высокий спрос, то и новый товар, принадлежащий к данному типу, имеет больше шансов на продажу». Рассчитаем рейтинг продаж предметов объявления по его категориям, подкатегориям и параметрам объявления.

Средний рейтинг продаж для категориального признака — это среднее арифметическое значений целевой переменной (вероятности продать товар) по данному категориальному признаку.

На рис. 3–4 приведены графики зависимости средней вероятности сделки от категории товара и от его первого параметра. Исходя из иллюстраций можно сделать вывод, что наиболее успешными категориями являются услуги, причем первый параметр нам указывает на то, что это услуги ремонта и обслуживания техники, транспортные услуги и услуги няни, сиделки.

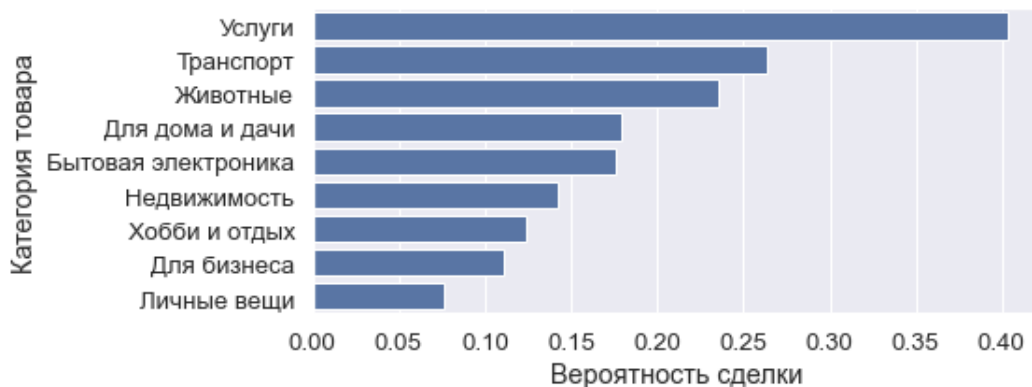


Рис. 3. Средняя вероятность сделки по категориям предмета объявления


```

def color_analysis(img):
    # получаем цветовую палитру изображения
    palette = defaultdict(int)
    for pixel in img.getdata():
        palette[pixel] += 1

    # сортируем цвета, присутствующие в изображении
    sorted_x = sorted(palette.items(), key=operator.itemgetter(1), reverse = True)
    light_shade, dark_shade, shade_count, pixel_limit = 0, 0, 0, 25
    for i, x in enumerate(sorted_x[:pixel_limit]):
        if all(xx <= 20 for xx in x[0][:3]): ## тусклый пиксель: слишком темный
            dark_shade += x[1]
        if all(xx >= 240 for xx in x[0][:3]): ## яркий пиксель: слишком светлый
            light_shade += x[1]
        shade_count += x[1]

    # вычисляем процентное соотношение тусклых/ярких пикселей ко всем рассмотренным
    light_percent = round((float(light_shade)/shade_count)*100, 2)
    dark_percent = round((float(dark_shade)/shade_count)*100, 2)
    return light_percent, dark_percent

```

Рис. 6. Вычисление процентного соотношения тусклых и светлых пикселей к общему их количеству

Пример неинформативного изображения по причине его чрезмерной тусклости приведен на рис. 7. Процентное соотношение темных пикселей ко всем в данном случае равно 98.12 процента.



Рис. 7. Пример самого темного изображения, вычисленного с помощью функции анализа яркости изображения

Некоторые изображения могут не содержать вариаций пикселей и быть полностью однородными. Средняя ширина пикселя — это мера, которая указывает количество контуров, присутствующих в изображении, то есть количество границ между отличающимися цветами в изображении. Если это число окажется очень низким, то изображение, скорее всего, является однородным (объект изображения не имеет четких форм или слишком зашумлен) и может не отражать правильный контент.

Для вычисления данного показателя используется метод Canny Detection библиотеки skimage. Он позволяет снизить количество шума в изображении до такого уровня, что мы сможем видеть лишь контуры объекта изображения (рис. 8). Далее, подсчитав соотношение количества светлых пикселей контура к общему количеству пикселей изображения, получаем

меру однородности изображения. Приведенное изображение по итогам расчетов имеет низкую среднюю ширину пикселя, что означает высокую степень однородности изображения или его зашумленности.

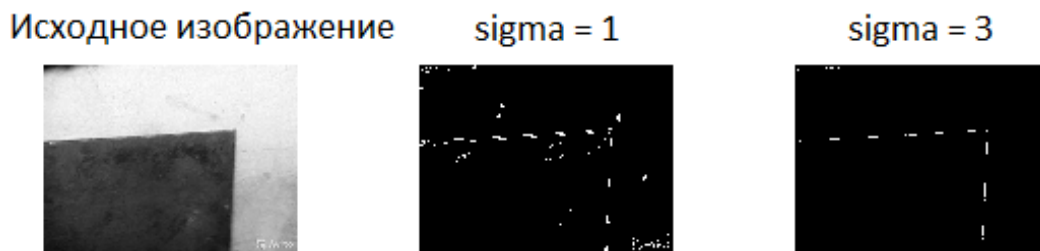


Рис. 8. Вычисление контура изображения с помощью метода Canny Detection

Цвета, используемые в изображениях, играют важную роль в привлечении внимания пользователей. Средний цвет изображения — еще один показатель, потенциально влияющий на успешность сделки. По цвету можно понять, насколько изображение соответствует описанию товара в объявлении, а также учесть моду на конкретный цвет при прогнозировании спроса. На рис. 9 показан программный код, позволяющий вычислить процентное соотношение среднего цвета картинке. Вычисляется среднее значение по всей картинке каждого из RGB цветов (красного, зеленого, синего), а затем делится на максимальное значение насыщенности пикселя (255 единиц). В итоге получаем три столбца, для красного, зеленого и синего цвета, обозначающие их среднее значение для всего изображения. Чем ближе коэффициент среднего цвета изображения к 1, тем больше его присутствие на изображении.

```
def get_average_color(img):
    path = images_path + img
    img = cv2.imread(path)
    average_color = [img[:, :, i].mean() for i in range(img.shape[-1])]
    return average_color

features['average_color'] = features['image'].apply(get_average_color)

features['average_red'] = features['average_color'].apply(lambda x: x[0] / 255)
features['average_green'] = features['average_color'].apply(lambda x: x[1] / 255)
features['average_blue'] = features['average_color'].apply(lambda x: x[2] / 255)
```

Рис. 9. Вычисление среднего процентного соотношения RGB цветов

Слишком большие или слишком маленькие изображения могут быть не очень хороши для привлечения внимания. Пользователи могут пропустить просмотр изображения очень большого или очень маленького размера. Выведем новые признаки: размер изображения, его ширину и высоту.

Размытость изображения существенно ухудшает качество картинки. Она становится менее информативной. Для измерения степени размытости изображения можно использовать дисперсию фильтра Лапласа [3]. В этом методе один канал изображения свернут с фильтром лапласиана. Необходимо взять один канал изображения (предположительно в градациях се-

рого) и свернуть его со следующим ядром 3×3 :
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Затем необходимо вычислить дисперсию (квадрат стандартного отклонения) результата. Если дисперсия падает ниже заранее определенного порога, изображение считается размытым; в противном случае изображение не будет размытым. Чем больше полученное значение,

тем менее размытым выглядит изображение. Будем считать размытым изображения со значениями показателя размытости ниже 80 единиц. Пример размытого изображения представлен на рис. 10.



Рис. 10. Изображение с размытостью в 75.916 единиц

3. Построение моделей и оценка важности признаков

Для исследования важности признаков и построения качественной модели прогнозирования были реализованы XGBoost, CatBoost и LightGBM модели машинного обучения на основе градиентного бустинга деревьев решений [4].

Оценка точности модели проводилась с помощью средней квадратичной ошибки прогнозирования (RMSE — Root Mean Squared Error) — это корень от квадрата ошибки.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \text{ где } n \text{ — количество объектов, } y_i \text{ — фактическое значение целевой переменной объекта } t, \hat{y}_i \text{ — предсказанное значение целевой переменной объекта } t.$$

Данную метрику легко интерпретировать, поскольку она имеет те же единицы измерения, что и целевая переменная исходного набора данных. Также она оперирует меньшими величинами по абсолютному значению, что может быть полезно для вычисления на компьютере [5].

Результаты построенных моделей приведены в табл. 3. Наименьшую ошибку показала модель CatBoost.

Таблица 3

Точность построенных моделей

Модель	Показатель RMSE без дополнительных признаков	Показатель RMSE с дополнительными признаками
XGBoost	0.232	0.232
LightGBM	0.239	0.236
CatBoost	0.224	0.222

Заключение

В ходе данного исследования были описаны исходные данные, проанализированы зависимости целевой переменной и на основе интеллектуального анализа данных сформирова-

ны дополнительные признаки, построено несколько моделей машинного обучения на основе градиентного бустинга деревьев решений для прогнозирования спроса, определена наиболее эффективная модель и проведен анализ ее результатов. В табл. 4 представлены признаки, полученные с помощью интеллектуального анализа данных.

Таблица 4

Итоги интеллектуального анализа данных

Исходные признаки	Дополнительные признаки
Дата объявления	День месяца, день недели
Регион, город, категория и подкатегория продукта, параметры 1–3, тип продавца	Рейтинги продаж по регионам, городам, категориям и подкатегориям продукта, параметрам 1-3, типам продавца
Заголовок и описание объявления	Количество слов в заголовке и описании, длина заголовка и описания объявления, наличие описания объявления, количество специфических символов
Изображение	Наличие изображения, степень тусклости и яркости изображения, средняя ширина пикселя, среднее значение синего, зеленого и красного цветов, степень размытости изображения, размер изображения, длина и ширина изображения

Всего сформировано 38 новых столбцов признаков. Определить полезность полученных признаков, то есть оценить их влияние на целевую переменную поможет метод `get_feature_importance` [6], позволяющий вывести важность признаков для конкретной модели машинного обучения.

Из всех новых созданных признаков наиболее успешными оказались: рейтинг продаж по первому параметру, рейтинг продаж по третьему параметру, рейтинг продаж по второму параметру, количество слов в заголовке, рейтинг продаж по городу, рейтинг продаж по категории продукта, длина заголовка (причем в обратной зависимости, в отличие от признака «количество слов в заголовке»), количество символов точки в описании объявления, среднее процентное соотношение зеленого цвета в изображении.

Это наиболее полезные признаки, они помогают оценить эффективность описания продукта в объявлении и ответить на вопрос: какие факторы влияют на статистику продаж и каким образом?

В результате качественной обработки данных, выделения новых признаков и подбора гиперпараметров модели лучше всего показала себя модель CatBoost с наилучшей в рамках исследования точностью $RMSE = 0.222$.

Литература

1. Avito Demand Prediction Challenge. – Available at: <https://www.kaggle.com/c/avito-demand-prediction> (accessed 15 March 2022)
2. Плас, Дж. В. Python для сложных задач: наука о данных и машинное обучение. / Дж. В. Плас. Пер. с англ. – Санкт-Петербург: Питер, 2018. – 576 с.
3. Pech Pacheco, J. L. Diatom autofocusing in brightfield microscopy: A comparative study / Pech Pacheco J. L. [etc] // Pattern Recognition, 2000. Proceedings. 15th International Conference on Volume: 3 (February 2000) – Barcelona: Instituto de Optica (CSIC), 2000. – P. 103 – 107.
4. Мюллер, А. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными / А. Мюллер, С. Гвидо. Пер. с англ. – Москва : Альпина Паблишерз, 2017. – 393 с.

5. Шолле, Ф. Глубокое обучение на Python / Ф. Шолле. Пер. с англ. – Санкт-Петербург: Питер, 2018. – 400 с.

6. Interpretable Machine Learning. – Available at: <https://towardsdatascience.com/interpretable-machine-learning-1dec0f2f3e6b> (accessed 28 March 2022)

Петрова Дарья Александровна — магистрант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: darya_petrova@mail.ru

Каширина Ирина Леонидовна (научный руководитель) — д-р техн. наук, проф., профессор кафедры математических методов исследования операций Воронежского государственного университета. E-mail: kash.irina@mail.ru

СРАВНИТЕЛЬНОЕ ИССЛЕДОВАНИЕ АЛГОРИТМОВ МУРАВЬИНОЙ КОЛОНИИ И ПЧЕЛИНОГО РОЯ ДЛЯ ПОСТРОЕНИЯ ОПТИМАЛЬНОГО КОЛЬЦЕВОГО МАРШРУТА

И. И. Петухов

Воронежский государственный университет

Введение

Задача построения кольцевого маршрута, также известная как задача коммивояжера, является важной, а также очень популярной задачей транспортной логистики [1]. Ее суть заключается в построении маршрута, который бы позволил торговцу посетить каждый город единственный раз и вернуться туда, где начался его маршрут. Критерием оптимальности маршрута обычно считается минимальная дистанция или минимальное время пройденного пути. В настоящее время популярны природные алгоритмы оптимизации, которые можно применять для решения поставленной задачи, к ним относят алгоритм муравьиной колонии и алгоритм пчелиного роя.

1. Описание алгоритма муравьиной колонии для решения задачи коммивояжера.

Пусть заданы n городов, для любых двух городов i и j известно целочисленное расстояние $d_{ij} > 0$ между ними. Требуется найти самый короткий замкнутый путь, проходящий через все города ровно по одному разу. Граф, представляющий эту задачу, является полносвязным и взвешенным по ребрам (вес ребра равен его длине d_{ij}).

Чтобы построить муравьиный алгоритм для задачи коммивояжера, берется колония из N муравьев и помещается в одну из вершин графа, которую будем называть стартовой вершиной. Все ребра графа помечаются одинаковым значением феромона $u_{ij} = u_0$. После этого муравьи начинают перемещаться по графу. Каждый муравей в процессе своего движения по графу хранит так называемый список запретов X , в который помещаются номера всех городов (вершин), в которых этот муравей уже побывал. При выборе следующей вершины для посещения муравьи опираются на значения концентрации феромона на исходящих ребрах и на значения длин этих ребер. Пусть k муравей находится в i вершине, а его запретный список X_k еще не является до конца заполненным (посещены еще не все вершины). Тогда вероятность перемещения в j вершину определяется формулой.

$$p_{ij}^k = \begin{cases} \frac{u_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l \notin X_k} u_{il}^\alpha \eta_{il}^\beta}, & j \notin X_k \\ 0, & j \in X_k \end{cases}$$

где α и β — параметры, управляющие относительной важностью между феромонной информацией u_{ij} и эвристической информацией

$$\eta_{ij} = \frac{1}{d_{ij}}.$$

Если же у k муравья не осталось непосещённых вершин, то он, уже не используя вероятностного выбора, просто возвращается в стартовую вершину. После того, как все муравьи вернутся домой, сделав ровно по n перемещений, производится обновление значений концентрации феромона на ребрах графа. Каждый муравей обновляет только те ребра, по которым он перемещался. Концентрация феромона на ребре (i, j) пересчитывается по формуле

$$u_{ij} \leftarrow pu_{ij} + \sum_{k=1}^N \Delta u_{ij}^k$$

где p — коэффициент испарения феромона

$$\Delta u_{ij}^k = \begin{cases} \frac{Q}{D_k}, & \text{проходил} \\ 0, & \text{иначе} \end{cases}$$

Q — некоторая фиксированная величина. Таким образом, чем короче путь k муравья, тем больше феромона он оставит на ребрах, по которым он ходил.

Описанные действия (перемещение по графу и обновлений феромонов) представляют собой одну итерацию муравьиного алгоритма. Итерации повторяются до тех пор, пока не окажется выполненным какой-нибудь из критериев останова алгоритма — исчерпано число итераций, достигнута нужная точность, получен единственный путь (алгоритм сошелся к некоторому решению) [2].

2. Описание алгоритма пчелиного роя для решения задачи коммивояжера.

Данный алгоритм моделирует поведение пчел в естественной среде. Идея пчелиного алгоритма заключается в том, что все пчёлы на каждом шаге будут выбирать как элитные участки для исследования, так и участки в окрестности элитных, что позволит, во-первых, разнообразить популяцию решений на последующих итерациях, во-вторых, увеличить вероятность обнаружения близких к оптимальным решениям.

Источник нектара характеризуется значимостью, определяемой различными параметрами. Фуражиры закреплены за источниками нектара. Количество всех пчел в этих участках больше, чем на остальных. Среднее количество разведчиков в рое составляет 5–10 %. Вернувшись в улей, пчелы обмениваются информацией посредством танцев на так называемой закрытой площадке для танцев.

Пусть решение представляет собой, как и в генетическом алгоритме, вектор N . Областью поиска нектара для пчел будет являться пространство поиска решений, размерностью $n!$ (количество всех возможных перестановок вектора N). Расположение источника нектара характеризуется конкретной перестановкой N , решением. Таким образом, координатами источника является решение N . Количество нектара на источнике обратно пропорционально целевой функции. Участок имеет размеры, где размер – количество решений, «близких» к N . Близость между векторами определяется значением расстояния Хемминга между ними. К примеру, решения $\{5,2,7,3,4,1,6\}$ и $\{5,4,7,3,2,1,6\}$ являются «близкими», т.е. в пространстве поиска они располагаются рядом, находятся на одном «участке» [3].

Словесное описание алгоритма пчёл.

- 1) Генерация участков для поиска нектара.
- 2) Оценка полезности участков.
- 3) Выбор участков для поиска в их окрестности.
- 4) Отправка фуражиров.
- 5) Поиск в окрестностях источников нектара.
- 6) Отправка пчёл-разведчиков.
- 7) Случайный поиск.
- 8) Оценка полезности новых участков.
- 9) Если условие останова не выполняется, то п. 2.
- 10) Конец работы алгоритма.

Таким образом, ключевой операцией алгоритма пчёл является совместное исследование перспективных областей и их окрестностей. В конце работы алгоритма популяция решений

будет состоять из двух частей: пчёлы с лучшими значениями ЦФ элитных участков, а также группы рабочих пчёл со случайными значениями ЦФ. Зависимость временной сложности пчелиного алгоритма от числа вершин — $O(n^2)$.

3. Практическое исследование

Экспериментальные исследования проводились на различных графах, размерность которых варьировала от 5 до 100. Все рассматриваемые графы являются полностью связными. Для проведения исследований была создана программа на ЭВМ.

Основные характеристики ЭВМ, на которой производились эксперименты:

- Intel(R) Xeon(R) CPU E5-2689 0 2.60GH
- ОЗУ — 32 ГБ
- Операционная система — Windows 10

Для нахождения точного решения был применен алгоритм полного перебора. Полученные результаты кратчайшего пути занесены в табл. 1.

Таблица 1

Результаты работы алгоритма полного перебора

5 вершин	10 вершин	50 вершин
846	956	1402

Для первого теста условием останова является истечение 10 секунд работы алгоритма. Результаты найденного пути занесены в табл. 2.

Таблица 2

Результаты работы алгоритмов за 10 секунд

Алгоритм	5 вершин	10 вершин	50 вершин	100 вершин
Муравьиный	846	956	1543	2067
Пчелиный	846	979	1674	2179

Для второго теста условием останова является истечение 60 секунд работы алгоритма. Результаты найденного пути занесены в табл. 3.

Таблица 3

Результаты работы алгоритмов за 60 секунд

Алгоритм	5 вершин	10 вершин	50 вершин	100 вершин
Муравьиный	846	956	1402	1844
Пчелиный	846	956	1434	1964

Заключение

После проведения анализа полученных результатов сравнения построения оптимальных маршрутов, полученных муравьиным и пчелиным алгоритмами, можно сделать следующие выводы:

1) с увеличением времени работы алгоритмов, полученный результат все больше приближается к значению кратчайшего пути;

2) в точности нахождения минимального пути и скорости работа алгоритм пчелиного роя уступает алгоритму муравьиной колонии.

Литература

1. *Мудров В. И.* Задача о коммивояжере. – М. : Знание, 1969. – 62 с.
2. *Штовба С. Д.* Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях. – 2003. – № 4. – С. 70–75.
3. *МакКоннелл Дж.* Основы современных алгоритмов. – М. : Техносфера, 2004. – 368 с.

Петухов Илья Ильич — магистрант 2-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: ilia.petukhov2015@yandex.ru

Борисенков Дмитрий Васильевич (научный руководитель) — канд. техн. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: xuser@relex.ru

РАЗРАБОТКА СИСТЕМЫ ХРАНИЛИЩА ДОКУМЕНТОВ С ИСПОЛЬЗОВАНИЕМ КОМПЬЮТЕРНОГО ЗРЕНИЯ

А. И. Попов

Воронежский государственный университет

Введение

В настоящее время переход к безбумажному документообороту многих предприятий осуществляется на уровне передачи скан-версий документов по телекоммуникационным каналам связи. Большинство организаций не в состоянии полностью отказаться от использования печатных версий документации. Промежуточным этапом перехода к тотальному безбумажному документообороту может послужить программное обеспечение для систематизации хранения цифровых версий реальных бумажных документов. Внесение данных для возможности их фильтрации и выгрузки требует указания реквизитов для хранения в информационной базе. Для ускорения данного процесса необходима его автоматизация. Динамичное развитие технологий нейронных сетей в сфере компьютерного зрения позволит решить данную проблему.

1. Постановка задачи

Задача заключается в разработке автоматизированной системы учета документации, включающей следующий функционал:

- добавление вручную новых типов документов;
- реализация автоматического распознавания реквизитов со сканов для определенных типов документов;
- сжатие документов при добавлении;
- возможность определения хранимых реквизитов в базе по каждому типу документа;
- множественное распознавание документов из одного файла (при условии наличия в файле только распознаваемых типов документов);
- поиск похожих документов при добавлении нового документа;
- актуализация хранимой версии документа;
- контекстный поиск внутри содержимого файлов;
- редактор для формализации распознавания документов, чтобы иметь возможность актуализировать часть приложения по распознаванию без вмешательства разработчиков.

2. Анализ задачи

В рамках поставленных условий требуется решить следующие задачи компьютерного зрения:

- предварительная обработка изображений с целью определения верности ориентации документа, удаления шума, улучшения контрастности и масштабирования;
- идентификация типа документа;
- обнаружение необходимых для данного типа документа реквизитов в результате сегментации участков изображения;
- оптическое распознавание значений найденных реквизитов.

3. Реализация

Анализ подходов к решению большинства перечисленных задач позволил сделать вывод, что проблем идентификации, сегментации и детектирования нейронными сетями можно значительно ускорить и качественно улучшить с помощью формализованного пользователем набора правил, особенно в ситуации строго определённой документации.

3.1. Формализация правил поиска

Для однозначной идентификации типа документа, а также реквизитов, необходимых для хранения в информационной базе определены следующие возможности задания правил поиска областей изображения и лексического анализа:

- указание положения реквизита по горизонтали в случае неизменности местонахождения реквизита путем выделения области на образце документа;
- определение типа возможных распознаваемых данных в заданной ячейке, таких как строка, число, дата;
- задание масок извлекаемых данных в виде регулярных выражений [1];
- назначение ключевых реквизитов, необходимых для точной интерпретации типа документа;
- задание относительного расположения области поиска информации в виде якоря, привязанного к другим реквизитам.

3.2. Распознавание

Пользовательский интерфейс создан на языках JavaScript и TypeScript с помощью библиотеки React. В качестве основного архитектурного подхода для построения пользовательского интерфейса выбрана Flux-архитектура, сочетающаяся с реактивным программированием и построенная на однонаправленных потоках данных.

Хранение данных приложения осуществляется в реляционной базе данных под управлением СУБД PostgreSQL.

Основная логика приложения содержится в его серверной части. Большая часть обработки данных производится в модуле, написанном на языке Java с использованием фреймворка Spring. Модуль с распознаванием отсканированной версии документа реализован на языке Python [3]. Серверное приложение реализовано в архитектурном стиле REST. Получение и выдача данных возможны при обращении к конечным точкам по их адресам по протоколу HTTP. Форматом обмена данных выбран JSON.

Скан-версии документов изначально являются довольно хорошо подготовленными данными для анализа. Тем не менее перед распознаванием они проходят предварительную обработку. Поиск необходимого угла поворота изображения происходит по следующему алгоритму:

- редуцирование размера изображения;
- бинаризация цветовой схемы;
- поиск контуров;
- применение преобразования Хафа для параметрической идентификации прямых линий описываемых уравнением $y = mx + b$. Для каждой прямой вычисляется значение двух параметров m и b . При этом в элементах аккумуляторного массива накапливаются значения, указывающие на вероятность наличия на изображении прямой $y = mx + b$, где M и B соответствуют дискретным значениям m и b ;
- определение угла наклона;
- выравнивание изображения с помощью аффинных преобразований [2].

Идентификация типа документа, загруженного в систему, начинается с ранжирования сверточной нейронной сетью принадлежности уменьшенной миниатюры документа к определенному типу [5]. В данной ситуации решается задача мульти-классовой классификации и определяется вероятность принадлежности каждому классу (типу документа) [4]. В выходном уровне сети используется активационная функция softmax. Для принятия решений используется логарифмическая функция потерь (кросс-энтропия). Сеть дообучается по мере внесения данных (в случае ручного указания пользователем отличного типа документа от того, который определен автоматически), так как, по сути, наращивается обучающая выборка. Полный алгоритм распознавания представлен на рис. 1.

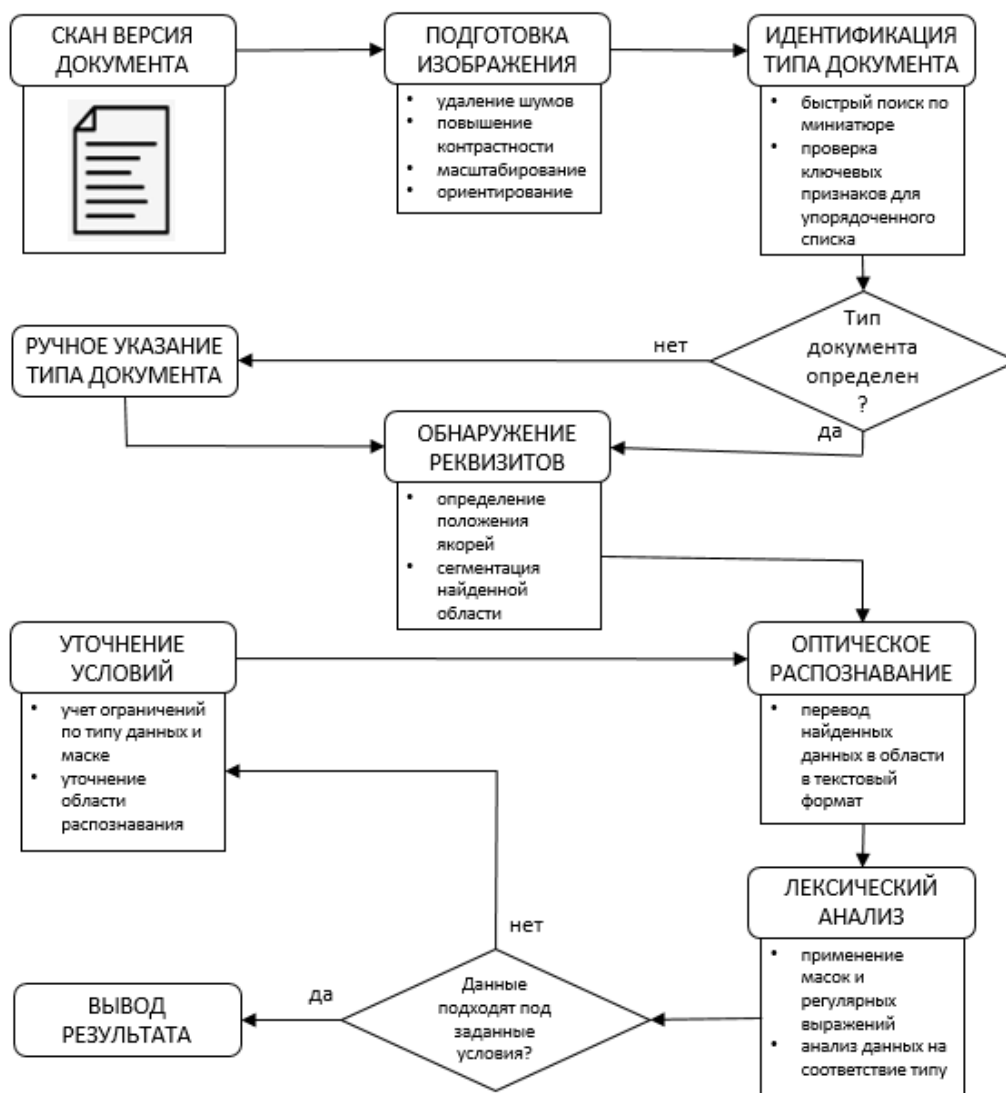


Рис. 1. Алгоритм распознавания документа

Заключение

Результаты данной работы показывают, что симбиоз нейронных сетей и возможность задания строго формализованных пользователями правил приносят существенный эффект в плане ускорения и точности распознавания. Конечно, есть вероятность негативного эффекта по причине человеческой ошибки, однако возможность неограниченного масштабирования приложения перевешивает данный фактор.

Литература

1. *Леденева, Т. М.* О различных реализациях механизма нечеткого логического вывода / Т. М. Леденева, В. В. Кашко // Актуальные проблемы прикладной математики, информатики и механики: сб. тр. Междунар. науч.-техн. конф. (Воронеж, 12–15 сентября 2016 г.): электронный ресурс. – Воронеж, 2016. – С. 391–395.
2. *Никулин, Е. А.* Компьютерная геометрия и алгоритмы машинной графики / Е. А. Никулин. – СПб. : БХВ-Петербург, 2003.
3. Santanu Pattanayak. Pro Deep Learning with TensorFlow: A Mathematical Approach to Advanced Artificial Intelligence in Python / Santanu Pattanayak. – Apress, 2017.
4. *Van de Sande, K.E.A.* Evaluating color descriptors for object and scene recognition / K. E. A. Van de Sande, T. Gevers, C. G. M. Snoek. – IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010
5. *Мазуров, В. Д.* Математические методы распознавания образов: учебное пособие / В. Д. Мазуров. – 2-е изд., перераб. и доп. – Екатеринбург : Изд-во Урал. ун-та, 2010. – 101 с.

Попов Алексей Игоревич — магистрант 2-го года обучения кафедры Вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: altdirection@outlook.com

Леденёва Татьяна Михайловна (научный руководитель) — д-р техн. наук, проф., заведующий кафедрой вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: tm-ledeneva@yandex.ru

БЛОКЧЕЙН

М. Ю. Пукась, С. В. Золотарев

Воронежский государственный университет

Введение

Децентрализация — это разделение процессов централизованной власти в целом на отдельные. Технология блокчейн — это базовая технология, которая обеспечивает децентрализацию и дает всем пользователям возможность стать частью узла для будущих транзакций. Используя технологию блокчейн, можно решать проблемы надежности сети, а также ухода от мега-корпораций, зависимость от которых слишком сильна в настоящее время.

1. Блокчейн

1.1. Технические характеристики

В централизованной системе данные могут быть изменены центральным органом власти. Но в сети блокчейн ни один из блоков не может быть изменен (защищен от несанкционированного доступа) после его создания из-за распределенной модели. В распределенной системе передача информации представленная в виде электронных денежных средств, происходит в системах биткойнов. Узлы в сети биткойн могут передавать и подписывать цифровым способом свои права на эти данные всем узлам, а блокчейн публично фиксирует транзакцию. Таким образом, всем узлам в сети разрешено проверять действительность транзакций. Эти детали поддерживались независимо в распределенной группе узлов. Наряду с этим механизмом криптографические значения или примитивы становятся защищенными от несанкционированного доступа, тк сама технология представляет собой особую базу данных, записи в которой линейно упорядочены, а допустимые запросы к ней могут быть лишь двух видов — запросы на чтение записей (любым желающим) и на добавление записи в конец базы (пользователем, выполнившим определённые условия). Данный механизм используется во многих секторах, а также доступен для многих приложений.

Защищенные от несанкционированного доступа блоки были внедрены распределенным способом. Это позволяет узлам записывать транзакции в сети таким образом, что никакие такие транзакции в сети не могут быть изменены. Эта идея была реализована в концепции криптовалют электронных денег, таких как биткойн. Биткойн был первой криптовалютой, основанной на блокчейне. Информация, хранящаяся в блокчейне биткойна, представляет собой электронные деньги и привязана к цифровому адресу. Узлы в блокчейне биткойна используют цифровые подписи и передают данные другим узлам в сети блокчейна. Записанные данные доступны прозрачно для всех узлов в сети для проверки транзакций. Наряду с криптографическим механизмом блокчейн защищен от несанкционированного доступа для узлов, которые пытаются изменить или подделать данные. Блокчейн создает доверие в среде, где невозможно легко идентифицировать все узлы.

Основные характеристики:

- Распределен — блокчейн распределен. Это позволяет узлам в блокчейне присоединяться к блокчейн-сети и защищает от злоумышленников.
- Безопасность — блокчейн защищен криптографией, hash функцией, так что транзакции защищены от несанкционированного доступа.

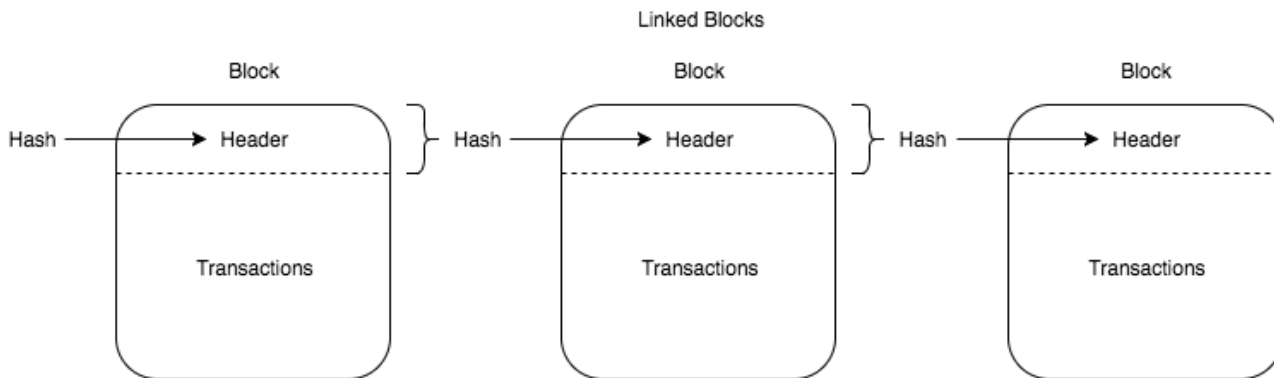


Рис. 1. Блокчейн архитектура

- Общий доступ — блоки являются общими для всех узлов в блокчейн-сети. Это предоставляет всем узлам прозрачный доступ.

1.2. Архитектура

Алгоритмом защиты данных в технологии блокчейна является защищенная хэш-функция (SHA), которая имеет 256 бит. Большинство компьютеров в настоящее время используют этот алгоритм в аппаратном обеспечении для более высоких вычислений.

Свойства хэш-функций:

- Устойчивость или сопротивление к прообразам. Входной хэш не может быть найден для входных данных, чем ниже вероятность нахождения входных данных, тем более защищённая хэш-функция.
- Устойчивость к столкновениям или сопротивление к второму прообразу. Это означает, что не удалось найти два входных хэша для одних и тех же выходных данных.

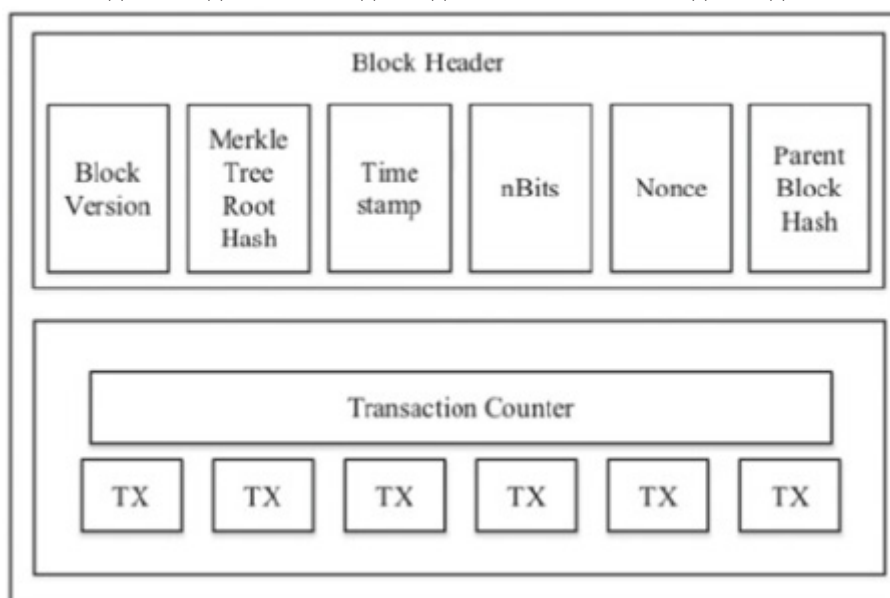


Рис. 2. Архитектура блока

Каждая транзакция в сети блокчейн хранится в цифровом регистре, называемом блоками в сети блокчейн. Блок состоит из данных обо всех транзакциях, которые произошли в распределенной сети. Копия данных одного блока совместно используется всеми другими узлами в сети с помощью согласованных протоколов. Блок содержит одноразовое значение (хэш-значение), временную метку каждой транзакции.

Заголовок блока в блокчейне содержит следующие данные:

- хэш-значение предыдущего блока;
- временная метка;
- размер блока.

Цифровые подписи являются ключами для обеспечения безопасности данных в технологии блокчейн, где они работают по алгоритму SHA-256. В сети блокчейн подписанная и проверенная транзакция распределяется между всеми узлами сети. Состоит из двух этапов, это этап проверки и этап подписания. Например, два узла в сети блокчейна договариваются в соответствии с протоколом и отправляют данные друг другу. Эти данные подписываются и шифруются пользователем, который их отправляет — это этап подписания. Затем эти данные проверяются пользователем, который их получает — это этап проверки. Таким образом, данные надежно защищены от несанкционированного доступа.

Форкинг. При использовании технологии блокчейн изменения и обновление данных на некоторое время станут затруднительными. В блокчейн-сетях без разрешений из-за жесткой конкуренции и увеличения числа узлов, согласованных протоколами, это становится намного сложнее. Некоторые изменения в сети блокчейна и в данных называются форкингом, как показано на рис. 3. Он делится на два типа: мягкое и жесткое разветвление. При жестком разветвлении возврат узлов назад несовместим, и блоки не будут обновлены. В разрешенной блокчейн-сети, поскольку все узлы подпадают под общий протокол и все узлы известны друг другу, нет возможности на разветвление.

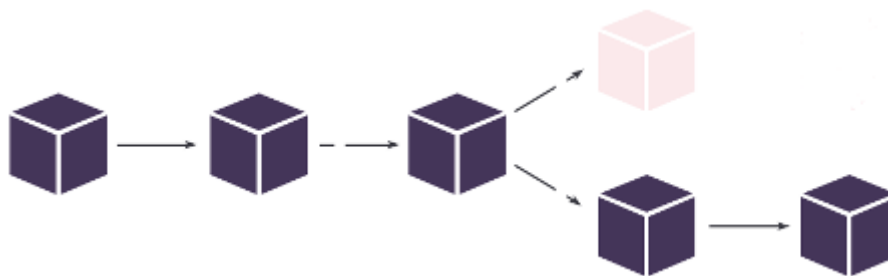


Рис. 3. Форкинг

Форкинг. При использовании технологии блокчейн изменения и обновление данных на некоторое время станут затруднительными. В блокчейн-сетях без разрешений из-за жесткой конкуренции и увеличения числа узлов, согласованных протоколами, это становится намного сложнее. Некоторые изменения в сети блокчейна и в данных называются форкингом, как показано

1.2. Распределенные приложения в блокчейне

Существует три типа технологии блокчейн:

- публичный;
- приватный;
- консорциум.

Публичная блокчейн-платформа, на которой каждый узел в сети блокчейн может получить доступ к данным и должен предоставить подтверждение работы для доступа к тем же данным. Биткойн является лучшим примером публичного блокчейна, где он представляет собой децентрализованную сеть. Например Ethereum помогает запускать смарт-контракты, а также позволяет пользователям создавать dapps.

В приватной сети блокчейна доступ к данным разрешен только авторизованным узлам. Подобно централизованным серверам, только владелец блока имеет разрешение предостав-

лять доступ к другим узлам в частной сети блокчейна. Совместное программное обеспечение предоставляется компанией Eris industries, использующей эту технологию блокчейна. Операции, основанные на финансовом учреждении, включая расчеты, обеспечивались этим частным блокчейном. Для финансовых транзакций MultiChain предоставляет распределенную базу данных с открытым исходным кодом.

Блокчейн консорциума представляет собой комбинацию как публичного, так и частного блокчейна. Сотрудничая с другими сетями, пользователи создают различные децентрализованные приложения для многих вариантов использования. Процесс согласования контролируется заранее выбранным набором узлов

Заключение

В данной статье был представлен обзор технологии блокчейн. Таким образом, блокчейн — это высокоуровневая защищенная технология для совместного использования транзакций как в публичных, так и в частных сетях, которая очень хорошо подходит для разработки децентрализованных приложений и может стать будущим веба 3.0.

Литература

1. Лоран Лелу. Блокчейн от А до Я / Лелу Лоран. – 1-е изд., перераб. и доп. – Москва : Эксмо, 2017. – 190 с.
2. Melanie Swan. Blockchain: blueprint for a new economy / Swan Melanie; O'Reilly Media; 1. Edition: Kindle, 2015. – 152 с.
3. Habr Habr. – Режим доступа: <https://habr.com/ru/company/flant/blog/516314/>– (Дата обращения: 01.04.2022).

Пукась Мария Юрьевна — магистрант 2-го года обучения кафедры ПОиАИС Воронежского государственного университета. E-mail: marielli1.618@gmail.com

Золотарев Сергей Владимирович (научный руководитель) — канд. физ.-мат. наук, преподаватель кафедры программного обеспечения и администрирования информационных систем Воронежского государственного университета.

ОБЗОР МЕТОДОВ МОДЕЛИРОВАНИЯ РАСПРОСТРАНЕНИЯ ЗАБОЛЕВАНИЙ

Ж. А. Святоха

Воронежский государственный университет

Введение

В настоящее время в условиях эпидемии в результате распространения коронавируса (COVID-19) остро стоит задача прогнозирования размеров потенциального ущерба, сроков пика и окончания распространения эпидемии, а также оценки эффективности возможных управленческих решений, направленных на предотвращение распространения эпидемии, своевременной организации противоэпидемических мероприятий. В этих условиях важными составляющими для эффективного принятия решений являются оперативный анализ и прогноз эпидемиологической ситуации, которые могут быть обеспечены благодаря совершенствованию информационно-аналитического обеспечения.

Математическое моделирование достаточно давно получило признание как важный инструмент в эпидемиологии. Модели позволяют определить численные закономерности эпидемического процесса, просчитывать возможные сценарии развития эпидемий и определять их исходы, сравнивать эффективность различных стратегий профилактических и противоэпидемических мероприятий. Выбор подхода к моделированию зависит от задач, стоящих перед исследователем.

Одним из инструментов построения моделей являются мультиагентные технологии. Мультиагентная модель состоит из популяции агентов, окружения и набора правил, регулирующих поведение агента, что «приближает» модель к реальной ситуации. Агент-ориентированные модели могут использоваться для моделирования нелинейных отношений между агентами, с более детальным проектированием межличностных взаимодействий, и тем самым решать более широкий спектр исследовательских задач, чем при использовании традиционных аналитических подходов. Даже простые агентные модели способны демонстрировать сложные взаимодействия поведения агентов и предоставлять ценную информацию о динамике реальной системы.

В данной статье будет произведено рассмотрение вариантов построения моделей имитации распространения болезней.

1. Виды моделей

Математические модели традиционно делятся на аналитические и имитационные.

Аналитические модели представляют собой уравнения или системы уравнений, записанные в виде алгебраических, интегральных, дифференциальных, конечно-разностных и иных соотношений и логических условий. Такая модель, как правило, статическая и подходит лишь для очень простых и сильно идеализированных задач и объектов, которые имеют мало общего с реальной (сложной) действительностью, но обладают высокой общностью. Данный тип моделей обычно применяют для описания фундаментальных свойств объектов. В аналитических моделях можно использовать широкий арсенал математических методов, что часто позволяет найти оптимальное решение и иногда провести анализ чувствительности. Сложные объекты редко удаётся описать аналитически.

Альтернативой аналитическим моделям являются имитационные модели (динамические). Основное отличие имитационных моделей от аналитических состоит в том, что вместо ана-

литического описания взаимосвязей между входами и выходами исследуемой системы строят алгоритм, отображающий последовательность развития процессов внутри исследуемого объекта, а затем «проигрывают» поведение объекта на компьютере. Имитационные модели используются, когда объект моделирования настолько сложен, что хорошо описать его поведение математическими уравнениями невозможно или очень затруднительно.

Также, имитационное моделирование позволяет разбивать большую модель на части или объекты, которыми можно оперировать по отдельности, создавая другие, более простые или, наоборот, более сложные модели. Имитационная модель в отличие от аналитической представляет собой не законченную систему уравнений, а развернутую схему с детально описанной структурой и поведением изучаемого объекта.

При имитационном моделировании воспроизводится алгоритм функционирования системы во времени — поведение системы, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания, что позволяет по исходным данным получить сведения о состояниях процесса в определенные моменты времени, дающие возможность оценить характеристики системы. Имитационные модели позволяют достаточно просто учитывать такие факторы, как наличие дискретных и непрерывных элементов, нелинейные характеристики элементов системы, многочисленные случайные воздействия и другие, которые часто создают трудности при аналитических исследованиях. Имитационное моделирование тяготеет к объектно-ориентированному представлению, которое естественным образом описывает объекты, их состояние, поведение, а также взаимодействие между ними. В настоящее время имитационное моделирование — наиболее эффективный метод исследования систем, а часто и единственный практически доступный метод получения информации о поведении системы, особенно на этапе ее проектирования.

Исходя из всего выше сказанного, при рассмотрении различных вариантов построения мультиагентных моделей необходимо сначала изучить аналитические модели. Рассмотрим их более подробно.

2. Аналитические модели

2.1. Модель SIR

Чаще всего для моделирования распространения инфекционных заболеваний используется классическая модель SIR, которую разработали британские ученые Андерсон Кермак и Уильям Маккендрик. Описание данной модели заключается в следующем: население в модели делится на три группы в зависимости от статуса: восприимчивый к заболеванию, заражённый и выздоровевший (Susceptible — Infected — Recovered).

Система уравнений SIR:

$$\begin{cases} \frac{dS}{dt} = -\frac{\beta IS}{N}, \\ \frac{dI}{dt} = \frac{\beta IS}{N} - \gamma I, \\ \frac{dR}{dt} = \gamma I, \end{cases}$$

где $S(t)$ — численность восприимчивых индивидов в момент времени t ;

$I(t)$ — численность инфицированных индивидов в момент времени t ;

$R(t)$ — численность переболевших индивидов в момент времени t ;

β — коэффициент интенсивности контактов индивидов с последующим инфицированием;

γ — коэффициент интенсивности выздоровления инфицированных индивидов.

У данной модели есть минусы. SIR-модель перестает работать в случае необходимости учитывать неоднородность популяции (например, различную плотность населения в разных районах), разные пути передачи инфекции и факторы случайности, значимые в малых популяциях и на начальной фазе распространения заболевания.

SIR-модель стала основой для других моделей. Далее стали создаваться ее различные вариации.

2.2. Модель SEIR

SEIR (Susceptible — Exposed — Infected — Recovered). Здесь к трём состояниям в SIR-модели добавляется еще одно — Exposed (зараженный, находящийся в инкубационном периоде).

Описывается системой уравнений:

$$\begin{cases} \frac{dS}{dt} = \mu N - \mu S - \beta \frac{I}{N} S, \\ \frac{dE}{dt} = \beta \frac{I}{N} S - (\mu + a) E, \\ \frac{dI}{dt} = a E - (\gamma + \mu) I, \\ \frac{dR}{dt} = \gamma I - \mu R, \end{cases}$$

где μ — уровень смертности;

a — величина, обратная среднему инкубационному периоду заболевания;

$E(t)$ — численность индивидов, то есть носителей заболевания в момент времени t .

Иначе говоря, в каждый момент времени каждый индивидуум с определенной вероятностью может заразиться, через некоторое время — заболеть, а затем поправиться либо погибнуть.

2.3. Модель SIS

Модель «восприимчивые — инфицированные — восприимчивые» применима при анализе распространения заболеваний, к которым не вырабатывается иммунитет, например гриппа и ОРВИ. Она описывается следующей системой уравнений:

$$\begin{cases} \frac{dS}{dt} = -\frac{\beta SI}{N} + \gamma I, \\ \frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I. \end{cases}$$

Вместе первое и второе уравнение означают, что число здоровых и больных в сумме не меняется, а число заражений пропорционально числу контактов здоровых и больных.

Второе уравнение описывает изменение числа заболевших в единицу времени, которое пропорционально числу заражений (числу контактов здоровых и инфицированных индивидуумов) за вычетом числа выздоровлений.

2.4. Модель MSEIR

Эта модель, построенная для заболевания с инкубационным периодом и учитывающая иммунитет детей, приобретенный внутриутробно, — одна из самых сложных для анализа в силу наличия большого числа независимых параметров. Система уравнений для нее выглядит следующим образом:

$$\begin{cases} \frac{dM}{dT} = B - \delta M - \mu M, \\ \frac{dS}{dT} = \delta M - \beta SI - \mu S, \\ \frac{dE}{dT} = \beta SI - (\varepsilon + \mu) E, \\ \frac{dI}{dT} = \varepsilon E - (\gamma + \mu) I, \\ \frac{dR}{dT} = \gamma I - \mu R, \end{cases}$$

где $M(t)$ — индивиды с пассивным иммунитетом, защищённые материнскими антителами.

От рассмотренных выше моделей эта система уравнений отличается тем, что учитывает рождение детей, вероятность заражения которых растёт со временем по мере утраты ими иммунитета, приобретенного внутриутробно. Эти зависимости описаны в первых двух уравнениях системы.

Приобретенный внутриутробно иммунитет может быть не у всех появившихся на свет детей, но вакцинацией можно охватить сто процентов младенцев. Введение в математическую модель этого параметра приводит к качественному изменению картины развития эпидемий.

Система уравнений для этой модели будет следующей:

$$\begin{cases} \frac{dS}{dt} = \mu N(1-P) - \mu S - \beta \frac{I}{N} S, \\ \frac{dI}{dt} = \beta \frac{I}{N} S - (\mu + \gamma) I, \\ \frac{dV}{dt} = \mu NP - \mu V, \end{cases}$$

где P — доля привитых младенцев, причем $0 < P < 1$.

Первые два уравнения повторяют модель SIR с учетом того, что вероятность заражения привитых детей равна нулю, а значит, вероятность заражения равна вероятности, что ребенок не привит, и, в свою очередь, равна $1 - P$.

Последнее уравнение учитывает смертность от других причин и позволяет рассчитать полную численность популяции.

2.5. Модель SEIR-HCD

Математическая модель распространения Covid-19 была предложена в работе [3]. Система имеет следующий вид:

$$\begin{cases} \frac{dS}{dt} = -\frac{5-a(t-\tau)}{5} \left(\frac{a_I S(t) I(t)}{N(t)} + \frac{a_E S(t) E(t)}{N(t)} \right) + \gamma R(t), \\ \frac{dE}{dt} = \frac{5-a(t-\tau)}{5} \left(\frac{a_I S(t) I(t)}{N(t)} + \frac{a_E S(t) E(t)}{N(t)} \right) - (k+p) E(t), \\ \frac{dI}{dt} = k E(t) - (\beta + \nu) I(t), \\ \frac{dR}{dt} = \beta I(t) + \rho E(t) - \gamma R(t) + \varepsilon_{HR} H(t), \end{cases}$$

$$\begin{cases} \frac{dH}{dt} = \nu I(t) + \varepsilon_{CH} C(t) - (\varepsilon_{HC} + \varepsilon_{HR}) H(t), \\ \frac{dC}{dt} = \varepsilon_{HC} H(t) - (\varepsilon_{CH} + \mu) C(t), \\ \frac{dD}{dt} = \mu C(t), \end{cases}$$

где $C(t)$ — численность индивидов, находящихся в критическом состоянии, требующих подключения к ИВЛ в момент времени t ;

$D(t)$ — численность летальных случаев в момент времени t ;

$a(t)$ — индекс самоизоляции;

τ — латентный период (запаздывание выделения вирионов или заразности);

a_I — параметр заражения между инфицированием и восприимчивым населением, который связан с заразительностью вируса и социальными факторами;

a_E — параметр заражения между бессимптомной и восприимчивой группами населения;

k — частота появления симптомов в открытых случаях, что приводит к переходу от бессимптомной к инфицированной популяции;

ρ — скорость восстановления выявленных случаев (выявленные случаи с выздоровлением без симптомов);

β — скорость выздоровления зараженных случаев;

γ — скорость повторного заражения Обратная величина уровня иммунитета вируса;

ν — доля госпитализированных случаев, находящихся в критическом состоянии и требующих подключения аппарата ИВЛ;

ε_{HR} — вероятность выздоровления индивидуумов, находящихся в тяжелом состоянии;

ε_{HC} — доля госпитализированных случаев, находящихся в критическом состоянии и требующих подключения к ИВЛ;

ε_{CH} — вероятность отключения аппарата ИВЛ у пациента;

μ — смертность в результате Covid-19;

E_0 — начальное количество бессимптомных инфицированных;

R_0 — начальное количество вылеченных индивидуумов.

2.6. Модель SEIR-D

Математическая модель распространения Covid-19. Система уравнений:

$$\begin{cases} \frac{dS}{dt} = -c(t-\tau) \left(\frac{a_I S(t) I(t)}{N(t)} + \frac{a_E S(t) E(t)}{N(t)} \right) + \gamma R(t), \\ \frac{dE}{dt} = c(t-\tau) \left(\frac{a_I S(t) I(t)}{N(t)} + \frac{a_E S(t) E(t)}{N(t)} \right) - (k + \rho) E(t), \\ \frac{dI}{dt} = kE(t) - \beta I(t) - \mu I(t), \\ \frac{dR}{dt} = \beta I(t) + \rho E(t) - \gamma R(t), \\ \frac{dD}{dt} = \mu I(t), \end{cases}$$

где c^{isol} — коэффициент влияния индекса самоизоляции на заражаемость.

3. Имитационные модели

Рассмотренные выше традиционные модели, опирающиеся на дифференциальные уравнения, описываемые в виде изменения численности индивидов в различных состояниях теряют свою эффективность при вводе новых параметров. Их сложность в этом случае растет экспоненциально. Решением этой проблемы служат агентные модели. Их также называют имитационными, индивидуально-ориентированными. Одним из преимуществ использования моделей на основе агентов по сравнению с моделями, описанными уравнениями, является то, что первые обеспечивают большую гибкость и свободу с точки зрения дизайна моделирования. В данных моделях описываются правила поведения каждого организма-агента, а ход эпидемического процесса рассматривается как сумма действий и взаимодействий индивидов. То есть моделирование распространения эпидемии с использованием агентного подхода отслеживает прогрессирование заболевания у каждого человека и отслеживает контакты каждого индивида с другими людьми. Важной особенностью является то, что агенты различаются по различным параметрам таким как: возраст, пол, состояние здоровья, поведению. Благодаря такой индивидуальности модели на основе агентов хорошо подходят для эпидемиологического моделирования.

Хотя эти модели эффективно отслеживают прогрессирование заболевания у каждого человека и отслеживают контакты каждого человека с другими в соответствующей системе (социальной или природной), исследователи начали добавлять физические инфраструктуры, такие как дорожные сети, и реальную географическую среду для учета более сложные взаимодействия между восприимчивыми и инфицированными людьми.

В качестве основы для агентных моделей часто берутся аналитические модели, описанные выше. Например, рассматривается модель, в которой агент переходит между состояниями: чувствителен к инфекции, заражен и инфекционен, иммунен к инфекции (модель SIR). А далее данная модель усложняется введенными параметрами — условиями, которые переводят агента из состояния в состояние. Таким образом, модель легко привести к имитации различных болезней, учитывая их особенности.

Существует множество методологий для агентного моделирования. Модели делятся на общие и нацеленные на конкретное заболевание. В зависимости от изменения параметров в общих моделях можно симулировать распространение различных заболеваний среди населения. Такие модели часто используются для изучения будущих вспышек.

Согласно источнику [4] обычно существует четыре основных компонента модели: болезнь, общество, транспорт и окружающая среда. Моделирование общества отражает поведение населения. Транспорт моделирует перемещение агентов в окружающей среде. Среда предполагает пространства, в котором взаимодействуют агенты. Компонент болезнь описывает процесс передачи и прогрессирования болезни у человека. Нельзя не отметить, что разделение на данные компоненты является условным, так как компоненты сильно влияют друг на друга.

Рассмотрим компоненты по отдельности.

Общество. В этом компоненте необходимо определить будет ли общество иметь группы. Например, школы, рабочие места и так далее. Иногда для моделирования реального общества разработчики даже берут перепись населения, как это делал Скворцов [2]. При этом отмечается, что разработчики моделей часто сталкиваются с проблемой недостатка данных. Также на данном этапе решается масштаб модели, сколько агентов будет входить в нее. Будет ли производится моделирование страны, города, села и так далее. При этом маленькие общества будут иметь нереалистичность, а большие потребуют большего вычислительного времени.

Транспорт. Симуляции различными способами моделируют движение агентов. Это может быть перемещение между различными состояниями (дом, работа, друзья), перемещения по расписанию и так далее. Сложные модели даже используют географические пункты и

составляют движения между ними, используют моделирование общественного транспорта. В данном компоненте также решается будет ли больной брать больничный или садится на карантин, так как данные вопросы также отвечают за перемещение агента. Кроме явных параметров, влияющих на выбор транспортного маршрута, также могут иметься и скрытые параметры. Например, в работе [5] рассмотрено предположение о том, что кроме качественных характеристик есть еще и некоторые скрытые переменные. Предполагается, что деятельность индивида зависит от последовательно принятых им решений. Выбор варианта зависит от его привлекательности для агента. При этом строится функция привлекательности варианта. Введение в данную функцию случайной составляющей позволяет производить адекватное моделирование принятия решения.

Среда. Разработка среды может быть различной сложности, как обычная сетка, так и разделение на школы, места работы и другие места, представляющие город. Модели, рассматривающие болезни, у которых гораздо больше факторов заражения, имеют в компоненте среды погодные условия и другие факторы окружающей среды.

Болезнь. Компонент, в котором рассматриваются индивидуальные характеристики модели. При его разработке необходимо детально рассмотреть моделируемую болезнь, изучить медицинские источники, статистику заражения. Вводятся различные характеристики болезни такие как вероятности передачи, длительность инкубационных периодов и так далее.

Конечно, любое расширение по каждому из компонентов увеличивает затраты ресурсов модели и приводят к трудностям при ее запуске, но при этом повышает ее точность и развивает прогнозирование.

Заключение

В результате проделанной работы были изучены принципы проектирования моделей прогнозирования различных заболеваний, рассмотрены как классические методы построения, так и имитационные. Имитационные методы более предпочтительны в качестве инструмента прогнозирования. В настоящее время во многих случаях имитационные модели строятся не вместо аналитических, а параллельно с ними, поскольку они относительно просты для создания и позволяют исследовать такие параметры реальных систем, которые невозможно отобразить в аналитических моделях. Комбинированное использование аналитических и имитационных методов позволяет сочетать достоинства обоих подходов. При построении комбинированных (аналитико-имитационных) моделей производится предварительная декомпозиция процесса функционирования объекта на составляющие подпроцессы, и для тех из них, где это возможно, используются аналитические модели, а для остальных подпроцессов строятся имитационные модели. Такой подход дает возможность охватить качественно новые классы систем, которые не могут быть исследованы с использованием аналитического или имитационного моделирования в отдельности. Также многие имитационные модели строятся, опираясь на аналитические. Они используют подобное разбиение на классы, систему переходов состояний, а также в последствие могут проверяться с помощью опорной модели. Таким образом, рассмотрение обоих видов моделей важно при изучении моделирования болезней.

Литература

1. Кондратьев, М. А. Методы прогнозирования и модели распространения заболеваний / М. А. Кондратьев // Компьютерные исследования и моделирование. – 2013. – № 5. – С. 863–882.
2. Epidemic modelling: Validation of agent-based simulation by using simple mathematical models / Skvortsov [и др] // International Congress on Simulation and Modelling. – 2007. – С. 657–662.

3. Epidemic analysis of COVID-19 Outbreak and Counter-Measures in France / E. Unlu [и др] // medRxiv. – 2020. Режим доступа: <https://www.medrxiv.org/content/10.1101/2020.04.27.20079962v1>. – (Дата обращения: 15.03.2022).

4. Hunter E. [и др] A taxonomy for agent-based models in human infectious disease epidemiology // Journal of Artificial Societies and Social Simulation. – 2017. – № 3. – Режим доступа: <http://jass.soc.surrey.ac.uk/20/3/2.html>. – (Дата обращения: 10.03.2022).

5. McFadden D.L. Mixed MNL Models for Discrete Response / McFadden D.L., Train K. // Journal of Applied Econometrics. – 2000. – № 5. – С. 447–470. Режим доступа: [https://onlinelibrary.wiley.com/doi/10.1002/1099-1255\(200009/10\)15:5%3C447::AID-JAE570%3E3.0.CO;2-1](https://onlinelibrary.wiley.com/doi/10.1002/1099-1255(200009/10)15:5%3C447::AID-JAE570%3E3.0.CO;2-1). – (Дата обращения: 10.03.2022).

Святоха Жанна Александровна — магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: swyatoha.janna@yandex.ru

Каплиева Наталья Алексеевна (научный руководитель) — канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.

ПОСТАНОВКА ОПТИМИЗАЦИОННОЙ ЗАДАЧИ ФОРМИРОВАНИЯ АССОРТИМЕНТА СКЛАДА

А. Д. Сотников

*Военный учебно-научный центр Военно-воздушных сил
«Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина»*

Введение

При формировании ассортимента военного склада важно оптимизировать процессы закупки требуемого оборудования таким образом, чтобы обеспечить не только наличие оборудования, соответствующего по количеству и качеству запрашиваемым, но и возможность альтернативного выбора поставщика в условиях конкурентного спроса с другой. В статье рассматривается постановка оптимизационной задачи формирования заявки склада при возможности закупки однотипных видов оборудования у одного или разных поставщиков.

1. Постановка оптимизационной задачи

1.1. Формирование матриц соответствия

Рассмотрим задачу обеспечения объекта оборудованием, максимально отвечающим потребностям потребителя. Предположим, что можно закупить однотипное оборудование у различных производителей (поставщиков), которых будем различать по номерам, то есть $N = \{n\}$ — множество поставщиков. Задача формирования закупки включает в себя максимальное соответствие техническим требованиям потребителя, т. е. соответствие производственному процессу. Заявки на поставку поступают от $M = \{m\}$ — складов в систему принятия решения. Значит, соответствие n -го поставщика m -му складу можно представить матрицей соответствия

$$(\alpha_{nm}) = \begin{pmatrix} \alpha_{11} & \dots & \alpha_{1m} & \dots & \alpha_{1M} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \alpha_{n1} & \dots & \alpha_{nm} & \dots & \alpha_{nM} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \alpha_{N1} & \dots & \alpha_{Nm} & \dots & \alpha_{NM} \end{pmatrix}, \quad n = \overline{1, N}, m = \overline{1, M} \quad (1)$$

Методика выбора функции принадлежности предложена в работах [1–2].

Функция принадлежности задается таким образом, чтобы обеспечить полное или частичное соответствие критериям выбора по всем параметрам, то есть равенство нулю означает, что есть полное несоответствие хотя бы по одному частному параметру и данный выбор не приемлем.

1.2. Набор текста

Рассмотрим теперь формирование предложения от потребителя поставщику. Пусть \underline{N} предприятий-поставщиков выпускают K типов однородного оборудования, где $n = \overline{1, N}$,

$k = \overline{1, K}$. Тогда матрица степени соответствия предложения по закупке k -го типа товара n -му поставщику выглядит так

$$(\beta_{nk}) = \begin{pmatrix} \beta_{11} & \dots & \beta_{1k} & \dots & \beta_{1K} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \beta_{n1} & \dots & \beta_{nk} & \dots & \beta_{nK} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \beta_{N1} & \dots & \beta_{Nk} & \dots & \beta_{NK} \end{pmatrix}, \quad n = \overline{1, N}, k = \overline{1, K} \quad (2)$$

Данные, содержащиеся в матрицах (1) и (2) показывают взаимосвязь между требованиями по закупке и поставке оборудования требуемого назначения.

Предположим, что заявка на поставку по каждому типу оборудования формируется с учетом наличного количества у поставщиков. Будем рассматривать M складов $m = 1, \dots, M$, подающих заявки на требуемую закупку. Все склады должны получить оборудование, соответствующее предъявляемым техническим требованиям в количестве, не превышающем имеющееся у предприятия-поставщика. Тогда в матрице соответствия (1) каждый член $\alpha_{km} \in [0, 1]$ вычисляется, как значение степени соответствия оборудования k -го типа (у всех поставщиков) нуждам потребителей.

Обозначим x_{mk} — количество оборудования k -го типа, указанное в заявке m -го склада, причем данное количество оборудования может быть куплено у как у одного поставщика, так и у нескольких. Тогда общее количество требуемого товара заранее определено, то есть $\sum_{m=1}^M \sum_{k=1}^K x_{mk} = \sum_{m=1}^M v_m$, где v_m — где количество требуемого оборудования в заявке m -го склада.

Задача состоит в том, чтобы определить максимально возможное в закупке количество товара x_{mk} , соответствующего заявленным качественным и количественным требованиям. Критерий максимизации выберем в виде нормированной аддитивной свертки [3]:

$$\frac{\sum_{m=1}^M \sum_{k=1}^K \alpha_{mk} x_{mk}}{\sum_{m=1}^M \sum_{k=1}^K x_{mk}} \rightarrow \max, \quad (3)$$

или, убрав нелинейность в знаменателе

$$\frac{\sum_{m=1}^M \sum_{k=1}^K \alpha_{mk} x_{mk}}{\sum_{m=1}^M v_m} \rightarrow \max.$$

При этом ограничения выбираем таким образом, чтобы «насытить» потребности склада и продать имеющееся в наличии количество товара. Получили задачу линейного программирования:

$$\frac{1}{\sum_{m=1}^M v_m} \sum_{k=1}^K \sum_{m=1}^M \alpha_{mk} x_{mk} \rightarrow \max; \quad (4)$$

$$\sum_{k=1}^K x_{mk} = v_m, \quad m = 1, \dots, M; \quad (5)$$

$$\sum_{m=1}^M x_{mk} = p_k, \quad k = 1, \dots, K; \quad (6)$$

$$\sum_{k=1}^K p_k = \sum_{m=1}^M v_m; \quad (7)$$

$$x_{mk} \in \{0; Z^+\}; \quad (8)$$

где p_k — количество закупаемого оборудования данного типа.

Чтобы задача была замкнутой потребуем, чтобы потребность каждого склада в сумме совпадала с количеством закупаемого оборудования, т. е. $\sum_{m=1}^M v_m = \sum_{k=1}^K p_k$.

Выполнение ограничений (5–7) обеспечивается в случае незамкнутой задачи линейного программирования введением фиктивных складов.

1.3. Постановка коммерческого предложения

Предположим теперь, что коммерческое предложение каждому поставщику формируется с учетом имеющегося ассортимента. Будем рассматривать N поставщиков $n = 1, \dots, N$, которые могут предложить однотипные товары. Каждый поставщик может поставлять оборудование наиболее соответствующее n -му коммерческому предложению по k -му типу оборудования, где $\beta_{kn} \in [0, 1]$ — численная оценка степени соответствия.

Количество k -го типа оборудования соответствующего n -му коммерческому требованию обозначим y_{kn} .

Аналогично (4–8) получим задачу линейного программирования, в которой в качестве критерия возьмем нормированную аддитивную свертку:

$$\frac{\sum_{n=1}^N \sum_{k=1}^K \beta_{kn} y_{kn}}{\sum_{n=1}^N \sum_{k=1}^K y_{kn}} \rightarrow \max \quad (9)$$

$$\sum_{k=1}^K y_{kn} = w_n, \quad n = 1, \dots, N; \quad (10)$$

$$\sum_{n=1}^N y_{kn} = p'_k, \quad k = 1, \dots, K; \quad (11)$$

$$\sum_{k=1}^K p'_k = \sum_{n=1}^N w_n; \quad (12)$$

$$y_{kn} \in \{0; Z^+\}; \quad n = 1, \dots, N, \quad k = 1, \dots, K, \quad (13)$$

где $\sum_{n=1}^N \sum_{k=1}^K y_{kn} = \sum_{n=1}^N w_n$ — суммарное количество оборудования у поставщика; p'_k — количество оборудования, поставляемого соответствующим поставщиком.

Чтобы обеспечить согласованность поставленных оптимизационных задач потребуем, чтобы количество приобретаемых и распределяемых товаров совпадало, т. е. $\sum_{n=1}^N y_{kn} = \sum_{m=1}^M x_{mk}$, $k = 1, \dots, K$.

Задачи (4)–(8) и (9)–(13) объединим, используя подход [2–3] с весовым коэффициентом λ , $\lambda \in [0, 1]$, регулирующий доминирование (либо равнозначность) коммерческих и технических требований. При $\lambda = 0,5$ получаем задачу с равнозначными требованиями. Коэффициент λ задается на основе экспертной оценки.

Установление рационального соответствия между тремя множествами:

- множество заменяемых, исходя из технической задачи, типов оборудования;
- множество поставщиков, предлагающих требуемое оборудование;
- множество складов (потребителей), закупающих товары;

сводится к определению значений переменных x_{mk} и y_{kn} таким образом, чтобы склады получили оборудование с максимальным соответствием предъявляемым техническим требованиям, а поставщики поставили оборудование с минимальными отклонениями (максимальным соответствием) от коммерческих предложений.

Окончательно модель выглядит следующим образом [4]:

$$\lambda \frac{1}{\sum_{m=1}^M v_m} \sum_{k=1}^K \sum_{m=1}^M \alpha_{mk} x_{mk} + (1-\lambda) \frac{\sum_{n=1}^N \sum_{k=1}^K \beta_{kn} y_{kn}}{\sum_{n=1}^N w_n} \rightarrow \max; \quad (14)$$

$$\sum_{k=1}^K x_{mk} = v_m, \quad m = 1, \dots, M; \quad (15)$$

$$\sum_{k=1}^K y_{kn} = w_n, \quad n = 1, \dots, N; \quad (16)$$

$$\sum_{n=1}^N y_{kn} = \sum_{m=1}^M x_{mk}, \quad k = 1, \dots, K. \quad (17)$$

$$x_{mk}; y_{kn} \in \{0; Z^+\}; \quad \forall m, n, k. \quad (18)$$

Заключение

Целочисленное решение задачи (14)–(18) позволяет определить оптимальное количество продаваемого и закупаемого оборудования, наиболее соответствующего техническим характеристикам (закупка) и коммерческим требованиям (продажа). Решение оптимизационной задачи позволяет оптимально формировать ассортимент склада при покупке однотипного оборудования у разных поставщиков.

Оптимизационная задача (14)–(18) относится к классу задач линейного программирования и соответственно может быть решена симплекс-методом [5].

Литература

1. Шапкин, А. С. Математические методы и модели исследования операций / А. С. Шапкин, Н. П. Мазаева. – М.: Дашков и К, 2005. – 400 с.
2. Матвеев, М. Г. Решение задачи линейного программирования с нечеткими параметрами / М. Г. Матвеев // Современная экономика, проблемы и решения. – 2012. – №1 (25). С. 170-178.
3. Ногин, В. Д. Принятие решений в многокритериальной среде: количественный подход / В. Д. Ногин – М.: ФИЗМАТЛИТ, 2004. – 176 с.
4. Будяков, А. Н. Решение задачи выбора как задачи целочисленного линейного программирования / А. Н. Будяков, К. Г. Гетманова, М. Г. Матвеев // XVIII Международной научно-практической конференции «Информатика: проблемы, методология, технология», Воронеж: ВГУ, 2018.
5. Вентцель, Е. С. Исследование операций / Е. С. Вентцель. – М.: Советское радио, 1972. – 552 с.

Сотников Андрей Дмитриевич — студент 3-го курса ВУНЦ ВВС ВВА, ф-т тыловое обеспечение, andreysotnikow32@gmail.com

Сумина Рита Семеновна (научный руководитель) — канд. техн. наук, доц., доцент кафедры МИПА (ВГУ), кафедры математики (ВУНЦ ВВС ВВА).

АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ К РАЗРАБОТКЕ ПРИЛОЖЕНИЙ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ НА ПРИМЕРЕ ПОМОЩНИКА АВТОЛЮБИТЕЛЮ

С. В. Стратиенко

Воронежский государственный университет

Введение

Современные направления информационных технологии каждый день стремительно развиваются и набирают популярность. Цифровые устройства, такие как смартфон или планшет, с каждым днём обретают новые способы применения в повседневной жизни. Одно из интересных и набирающих популярность возможностей использования мобильных устройств в современной реальности являются средства дополненной реальности.

Дополненная реальность (от англ. «augmented reality, AR» — расширенная реальность) — это технология дополнения реального (видимого) объекта с помощью виртуальной (цифровой) информации, к примеру, в форме текста, графики, аудио [1]. Ключевая задача AR — расширение возможностей взаимодействия пользователя и окружающего его мира.

Современный рынок содержит в себе огромное разнообразие товаров. Каждый продавец стремится продать как можно выгоднее свою продукцию. В то же время пользователю необходимо получить максимум возможностей от совершенной покупки, и для этого ему необходимо изучить сопроводительные документы. Большая часть инструкций к различным устройствам существует в виде бумажных книжек, документов формата PDF или в виде статей на интернет форумах. К сожалению, это не всегда удачные форматы для получения информации, т.к. изучение руководств по эксплуатации приводит к столкновению с сложными терминами, отсутствию ответов на появляющиеся вопросы, плохое качество сопроводительных графических и фотографических данных. Если человек в ходе работы с продуктом не смог в нём разобраться, то это говорит о том, что он не получил оплаченную услугу.

Следующая важная неприятность, с которой можно столкнуться при отсутствии полных и детальных сведений, возможна при столкновении с неожиданными поломками и неисправностями. Решение проблемы требует больших временных затрат, направленных на поиск ответов. Однако не всегда удаётся получить из ограниченной информации от службы поддержки, в инструкциях, в сети интернет то, что действительно необходимо.

Данная проблема в настоящее время является одной из глобальных в мире. Так как информационные носители служат для изучения всех возможностей объектов, то возникает необходимость в поиске наиболее подходящих решений.

Так как мобильные устройства стали неотъемлемой частью жизни современного человека, то решением поставленной проблемы будет выступать приложение, ориентированное на помощь в получении знаний и навыков эксплуатации предметов.

1. Анализ задачи

В настоящее время достаточно сложно найти человека, у которого есть машина и отсутствует высокотехнологичное мобильное устройство. Для водителя смартфон является одним из главных помощников. Приложения для навигации, с использованием максимально детализированных карт местности, информацией о пробках, светофорах, камерах. Виртуальный сборник правил дорожного движения. Программы для проверки штрафов и оплаты платных

парковок. Приложения для возможности заправиться, не выходя из машины. Существует огромное множество программ, облегчающих жизнь современным автомобилистам.

Производители автомобилей используют максимально большое число появляющихся новинок на рынке информационных технологий, учитывая потребности клиентов. Но программ, которые направлены на создание электронного руководства пользователя с использованием технологии дополненной реальности, очень мало. Одна из известных разработок — Hyundai Virtual Guide [2]. Его основная цель заключается в возможности просмотра обучающего материала об элементах машины при направлении камеры на тот или иной объект. На дисплее мобильного устройства будут отображаться предметы реального мира с дополнительными виртуальными изображениями и подсказками. Плюс такой программы заключается в замене бумажного руководства по эксплуатации электронным.

Но у этого программного продукта были выявлены следующие недостатки [2]:

- У аппаратного и программного обеспечения приложения отсутствует возможность работать с русским языком.
- База данных моделей автомобилей, с которыми будет работать приложение, крайне мала.
- В России отсутствуют последние выпущенные и продающиеся модели автомобилей, которые существуют в информационной базе программы.
- Для того, чтобы найти необходимую информацию в меню, необходимо совершить много действий.
- Во время сканирования фотографии для отображения необходимых виртуальных элементов и подсказок, приложение сильно нагружает устройство, на котором в данный момент времени оно запущено.

В результате анализа данной программы был получен следующий результат. Применение дополненной реальности может помочь сократить временные затраты на поиск требуемых данных. Но уже существующее решение имеет ряд минусов.

Так как сама идея использования технологии AR [1] для создания наиболее удобного и познавательного ресурса информации об эксплуатации автомобиля актуальна в настоящее время, то возникает необходимость в создании аналога данного приложения, учитывая все недостатки существующего решения.

Основная цель данной работы заключается в анализе существующих подходов к разработке приложения дополненной реальности на примере помощника автолюбителя, реализации некоторых способов и получении вывода об использованных технологиях. Т.к. в настоящее время существует большое количество технологий, необходимо выбрать подходящие для реализации поставленной задачи, наиболее доступные и понятные платформы и библиотеки. Также немало важный фактор выбора – скорость разработки, использование функциональных возможностей мобильного устройства.

2. Анализ существующих решений

Первый этап, который необходимо выполнить перед началом анализа существующих подходов к разработке приложений дополненной реальности на примере помощника автолюбителя – теоретический обзор и исследование основ данной области.

Необходимо точно определить предметную область, в рамках которой необходимо решить поставленные задачи, т. к. от этого будет зависеть выбор подходящего типа AR-приложения. В зависимости от различных критериев проекты дополненной реальности распределены по группам [3]. Классификация представлена в табл. 1.

Для реализации приложения помощника автолюбителя необходимо выбрать следующие типы: интерактивный (в приложении предполагается возможность выбора марки и модели

Классификация AR-приложений

Критерий классификации	Тип приложения дополненной реальности	Комментарий
Взаимодействие с пользователем	Автономные	Не рассчитаны на взаимодействие с пользователем
	Интерактивные	Предполагают взаимодействие с пользователем (ввод информации, выбор параметров и свойств и т. д.)
Степень мобильности системы	Стационарные	Не предполагают перемещения, работа в рамках одной области
	Мобильные	Предполагают перемещение в пространстве, взаимодействие с различными объектами
Способ получения информации	Геопозиционные	Использование информации, полученной с помощью датчиков GPS, компаса
	Оптические	Получение информации с помощью видеокамеры устройства

автомобиля), мобильный (взаимодействие с различными частями и деталями машины), оптический (для получения информации используется камера мобильного устройства).

Создание проекта требует использования специальных инструментов. Выбор зависит от сложности моделируемого графического контента.

Среда разработки Unity [4] считается одной из оптимальных для начала знакомства с разработкой приложения дополненной реальности. Среди её преимуществ можно выделить доступность, многие моменты создаются с помощью редакторов, что упрощает написание кода, широкий выбор возможностей и SDK для создания проекта высокого качества.

Следующая среда — OpenSceneGraph [5]. Это открытое ПО, которое предназначено для создания виртуальной реальности. К плюсам данной среды относятся: поддержка большого количества 2D и 3D форматов.

ARCore [6] от компании Google, которая позволят создавать приложения дополненной реальности. Одно из ключевых преимуществ — использование трёх технологий для качественного представления виртуальной информации в реальном мире. К ним относится отслеживание движения для понимания положения мобильного устройства в пространстве, восприятие окружающей среды для оценки размеров и местоположения поверхностей, оценка освещённости для контроля уровня освещения.

Для разработки первого прототипа помощника автолюбителя был выбран Unity [4]. Выбор обосновывается немало важными преимуществами – скорость разработки достаточно высокая, возможностью использования на одной платформе различных библиотек с обширными функциональными возможностями, а также наличие большого числа понятной документации.

Ключевая роль в качественной реализации приложения дополненной реальности — выбор специализированной библиотеки, которая удовлетворяет всем требованиям. Программное обеспечение должно соответствовать основному требованию программы – распознавание предметов и отображение соответствующих им объектов. Наиболее популярные библиотеки, обладающие требуемыми функциональными возможностями представлены в табл. 2.

В результате анализа возможностей библиотек, представленных в табл. 2, для реализации приложения дополненной реальности были выбраны Vuforia [7], EasyAR Basic [12] (бесплат-

Классификация AR-библиотек

Название библиотеки	Лицензия	Поддерживаемые платформы	Комментарий
Vuforia [7]	Платная Бесплатная	Android, iOS, Unity	Распознавание нескольких объектов одновременно, распознавание 2D и 3D форматов, сканирование реального объекта для его идентификации, отображение дополнительных элементов с использованием OpenGL
ARToolkit [8]	Бесплатная	Android, iOS, Windows, Linux, Mac OS X	Распознавание 2D формата, отображение дополнительных элементов с использованием OpenGL
Wikitude [9]	Платная	Android, iOS, Google Glass, Titanium, Unity	Распознавание 2D и 3D форматов, определение местоположения объектов, сканирование реального объекта для его идентификации, анимация 3D моделей
LayAR [10]	Платная	iOS, Android, BlackBerry	Отображение дополнительных элементов в зависимости от положения пользователя и объектов
Kudan [11]	Платная	Android, iOS, Unity	Отображение дополнительных элементов в зависимости от положения пользователя и объектов, безмаркерное отслеживание предметов
EasyAR [12]	Платная Бесплатная	Android, iOS, UWP, Windows, OS X, Unity	Распознавание 3D объектов, восприятие окружающей среды, облачное распознавание, облачное развёртывание приложений

ная лицензия программного обеспечения) и демонстрационные версии Wikitude [9] и Kudan [11] для тестирования и ознакомления с разнообразием инструментов. Для того, чтобы данные SDK изучить на платформе Unity, необходимо предварительно ознакомиться с инструкцией по их установке [13].

3. Анализ выбранных программного обеспечения

Для определения оптимального варианта, который подходит для создания полноценного прототипа приложения дополненной реальности с целью распознавания реальных объектов и отображением соответствующих им объектов на примере помощника автолюбителя необходимо провести сравнительный анализ по следующим критериям.

Одно из обязательных требований, предъявляемых к функциональным возможностям программы, — отслеживание объектов. Для качественного и точного отображения необходимых подсказок соответствующей детали автомобиля необходимо, чтобы приложение умело находить необходимые предметы с помощью камеры мобильного устройства. Некоторые AR SDK для повышения уровня распознавания требуют, чтобы настоящий объект имел некоторую текстуру или объём. Пример качественного отслеживания объекта и отображения необходимой подсказки для него представлен на рис. 1 [14].



Рис. 1. Пример качественного отслеживания объекта и отображения правильной подсказки

Следующий критерий — анализ и обработка полученных с помощью камеры данных. Так как в автомобиле присутствует множество деталей, которые необходимо внести в базу приложения, в каталоге должны присутствовать файлы с подсказками. При этом должно соблюдаться правило — для одного реального объекта существует один виртуальный предмет. Исключением в данной ситуации будут детали, которые используются в автомобиле не в единственном числе, например, колёса.

Обязательное требование, предъявляемое к приложению, — быстрый поиск в базе данных подсказки, которая соответствует обработанным с камеры данным. Если виртуальный объект найден, то происходит подготовка для его вывода на экран устройства. Далее реальный объект дополняется подсказкой, и пользователь может изучить полученные данные. Иначе — программа ничего не отображает на мобильном устройстве.

Чтобы понять сложность разработки программы, необходимо оценить сложность и время, затраченное на установку SDK в выбранную платформу, подготовку данных (фотографии реальных объектов, соответствующие им подсказки), сопоставление пар настоящих и виртуальных предметов, поиск ошибок, сборку и тестирование приложения.

Предварительно были созданы прототипы приложения, с использованием упомянутых библиотек. Результат анализа всех выбранных программных средств по указанным критериям представлен в сравнительной табл. 3.

Таблица 3

Анализ программного обеспечения

SDK	Отслеживание объектов	Анализ и обработка данных камеры	Поиск соответствующего объекта в базе данных	Временные затраты, сложность разработки
Vuforia [7]	Объекты отслеживаются качественно. Если предмета нет в базе, программа ничего не выводит.	Объекты отслеживаются качественно. Если предмета нет в базе, программа ничего не выводит.	Быстрый поиск виртуальных объектов в базе данных .	Установка и настройка SDK проходит быстро, трудности возникали при обработке фотографий реальных объектов и соответствующих им виртуальных подсказок. При сопоставлении пар проблем не возникло. Длительная сборка приложения. На проект (с учётом всех этапов) было затрачено 1.5 месяца.

EasyAR Basic [12]	Объекты отслеживаются качественно. Если предмета нет в базе, программа ничего не выводит.	Объекты отслеживаются качественно. Если предмета нет в базе, программа ничего не выводит.	Быстрый поиск виртуальных объектов в базе данных.	Установка и настройка SDK проходит быстро. Подготовка данных (фотографии реальных объектов и подсказки), их обработка проходили без проблем. Сопоставление пар происходило без трудностей. При тестировании проблем не возникало. Сборка приложения длительная. На проект (с учётом всех этапов) был затрачен месяц.
Wikitude [9]	Объекты отслеживаются качественно. Если предмета нет в базе, программа ничего не выводит.	Объекты отслеживаются качественно. Если предмета нет в базе, программа ничего не выводит.	Быстрый поиск виртуальных объектов в базе данных.	Установка и настройка SDK проходит быстро. Подготовка данных (фотографии реальных объектов и подсказки), их обработка проходили без проблем. Сопоставление пар происходило без трудностей. При тестировании проблем не возникало. Длительная сборка, в результате которой в программе присутствует водяной знак (связано с использованием демонстрационной версии). На проект (с учётом всех этапов) было затрачено почти два месяца.
Kudan [11]	Объекты отслеживаются качественно. Если предмета нет в базе, программа ничего не выводит.	Объекты отслеживаются качественно. Если предмета нет в базе, программа ничего не выводит.	Быстрый поиск виртуальных объектов в базе данных.	Установка и настройка SDK проходили со сбоем, что приводило к возникновению проблем с демонстрационной версией. Подготовка данных (фотографии реальных объектов и подсказки), их обработка проходили без проблем. Сопоставление пар происходило без трудностей. При тестировании возникали сбои. Сборка не осуществлялась, т.к. её нельзя выполнить в бесплатной версии SDK. На проект (с учётом всех этапов) было затрачено 2.5 месяца.

Для анализа были использованы демонстрационные версии SDK Kudan [11] и Wikitude [9]. При работе с первым программным обеспечением возникали многочисленные сбои, что затрудняло работу с данной SDK. Кроме этого отсутствие возможности собрать проект в готовое приложение затрудняет полноценный анализ данной библиотеки. Wikitude имеет широкие функциональные возможности, но в результате будет получена программа с водяными знаками, которые могут вызвать затруднение при восприятии подсказок и реальных объектов. Использование данных SDK для создания полноценного прототипа приложения дополненной реальности на примере помощника автолюбителя не самый лучший вариант.

Заключение

В результате анализа существующих подходов к разработке приложений дополненной реальности на примере помощника автолюбителя были выявлены преимущества и недостатки популярных и распространённых платформ. Создание полноценного прототипа программы, с использованием Unity совместно с следующими SDK Vuforia, EasyAR Basic наиболее оптимальный вариант. При работе с SDK Kudan возможны столкновения с такими трудностями, как сбои при установке, так и отсутствие бесплатной лицензионной версии для полноценного использования. Мощный инструмент для создания AR-программ в среде Unity Wikitude недоступен обычным пользователям в связи с высокой стоимостью. SDK Kudan обладает рядом недостатков, таких как частые сбои в работе, трудности с установкой, а также отсутствие в бесплатной версии возможности полноценно собрать итоговый проект.

В результате исследования были получены следующие выводы. Для создания полноценного прототипа приложения дополненной реальности с целью распознавания реальных объектов и отображением соответствующих им объектов на примере помощника автолюбителя достаточно использовать платформу Unity и SDK EasyAR Basic. Данная версия библиотеки имеет бесплатный доступ, широкие возможности, понятные и простые инструкции, большую базу примеров.

Литература

1. Дополненная реальность. – Режим доступа: <https://clck.ru/9bsuk>. – (Дата обращения: 24.03.2022).
2. Hyundai Virtual Guide Introduces Augmented Reality to the Owner's Manual. – Режим доступа: <https://www.hyundai.com/worldwide/en/company/newsroom/hyundai-virtual-guide-introduces-augmented-reality-to-the-owner%2527s-manual-0000005199>. – (Дата обращения: 27.03.2022).
3. Классификация и перспективные направления использования технологии дополненной реальности / Б. С. Яковлев, С. И. Пусто // Известия ТулГУ. Технические науки. – 2013. – № 3. – С. 484–492.
4. Официальный сайт Unity. – Режим доступа: <https://unity.com/ru>. – (Дата обращения: 29.03.2022).
5. Официальный веб-сайт проекта OpenSceneGraph для высокопроизводительной графики. – Режим доступа: <http://www.openscenegraph.org/>. – (Дата обращения: 29.03.2022).
6. Официальный сайт ARCore. – Режим доступа: <https://arvr.google.com/core/>. – (Дата обращения: 30.03.2022)
7. Vuforia engine developer portal. – Режим доступа: <https://developer.vuforia.com/>. – (Дата обращения: 2.04.2022).
8. Официальный сайт ARToolkitX. – Режим доступа: <http://www.artoolkitx.org/>. – (Дата обращения: 2.04.2022).

9. Официальный сайт wiktitude. – Режим доступа: <https://www.wiktitude.com/>. – (Дата обращения: 5.04.2022).
10. Официальный сайт LayAR. – Режим доступа: <https://www.layar.com/>. – (Дата обращения: 5.04.2022).
11. Официальный сайт Kudan. – Режим доступа: <http://www.kudan.io/>. – (Дата обращения: 6.04.2022).
12. Официальный сайт EasyAR. – Режим доступа: <http://www.easyar.com/>. – (Дата обращения: 6.04.2022).
13. Официальная документация Unity. Руководство по интеграции SDK. – Режим доступа: <https://docs.unity3d.com/Manual/UnityAnalyticsBasicSDK.html/>. – (Дата обращения: 7.04.2022).
14. Стратиенко, С. В. Исследование подходов к разработке приложений дополненной реальности на примере помощника автолюбителя / С. В. Стратиенко // Актуальные проблемы прикладной математики, информатики и механики : сб. тр. Междунар. науч.-техн. конф. (Воронеж, 13–15 декабря 2021 г.) : электронный ресурс. – Воронеж, 2022. – С. 208–214.

Стратиенко Сергей Владимирович – магистрант 2-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: sergstrateg@yandex.ru

Болотова Светлана Юрьевна (научный руководитель) – канд. физ.-мат. наук, доц., доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета.
E-mail: [Bolotova.svetlana@gmail.com](mailto:Volotova.svetlana@gmail.com)

МОДЕРНИЗАЦИЯ LDA-АЛГОРИТМА В ЗАДАЧАХ ИНФОРМАЦИОННОГО ПОИСКА

А. И. Строгонов

Воронежский государственный университет

Введение

Человек с древних времен занимался поиском новых закономерностей окружающего мира. После наступления XXI века и возрастающей глобализацией посредством сети Интернет потребность быстро находить интересующую информацию возросла в разы. Но, зачастую, многие источники знаний содержат нерелевантную и неактуальную информацию, а, зачастую, и содержащую ложную и провокационную информацию. Проблема состоит еще и в том, что пользователю, если он не является экспертом в данной предметной области трудно различать данную информацию. Таким образом, необходимо совершенствовать алгоритмы поисковых систем с акцентом на качество информации.

В системах обработки естественного языка существует много моделей текста, но наиболее подходящей для построения алгоритма, подходящего под поставленные цели, является LDA-модель. К сожалению, при рассмотрении LDA-модели корпуса текстов вводится много допущений, а также достаточное количество приближенных оценок, что повышает вероятность как недообучения, так и переобучения создаваемых прикладных алгоритмов в области информационного поиска.

В данной работе рассматривается существование, единственность и сходимости данного модернизированного алгоритма. Статья состоит из нескольких частей: введения, описания недостатков LDA-модели, доказательства существования модернизированного алгоритма и вывода, содержащего открытые вопросы в данной исследовательской области.

1. Модернизированный LDA-алгоритм

1.1. Описание недостатков классического LDA-алгоритма

Идеей LDA модели является последовательное разделение всех документов на документы, принадлежащие каким-либо темам, распределенным по статистике Дирихле, в то время как документы и слова внутри них распределены по вероятности. Существенным недостатком является вероятностная структура данного процесса. То есть мы пытаемся винеровским процессом приблизить марковский процесс, то есть отследить его эволюцию. К сожалению, данное приближение не является строго математическим строго точным. Основной проблемой данного алгоритма является его неустойчивость, то есть необходимо довольно долго подбирать аргументы соответствующих функций. В результате чего алгоритм либо недообучается, либо, что еще хуже переобучается [2].

1.2. Недостатки LDA-алгоритма

Переобучение — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке. Переобучение возникает при использовании избыточно сложных моделей [3].

Возникает вопрос выявления этапа, благодаря которому модель оказывается переобученной. Из определения складывается ощущение, что данный эффект должен регистрироваться на этапе построения вычислительной модели. Но, как мы знаем, что нейросетевые модели лишь описывают (моделируют) пространство с заданной точностью. Таким образом, мы приходим к проблеме топологической конструкции пространства, что в терминах анализа данных, является этапом очистки данных.

Соответственно, прикладной целью данного исследования является поиск инструмента или способа, позволяющего выявлять данные аномалии в данных с помощью продвинутых методов современной математики. Стратегия исследования будет заключаться в попытке решить данную задачу при помощи теории меры и мартингалов, а затем представить вычислительную модель на дискретных многообразиях с возможным использованием спектральных методов.

1.3. Топологические свойства LDA-модели

Как известно, статистические и вероятностные модели опираются на теорию интеграла, меры и топологии. Предлагаемый авторами подход состоит в последовательном определении топологических характеристик заданного пространства, конструировании меры, а затем перехода к интегральному представлению. Обычно в промышленной практике делается ровно наоборот — сначала вычисляются вероятностные характеристики, а затем, на основе полученных данных делается вывод о составе исследуемого набора данных. Проблема данного подхода заключается именно в том, что далеко не все процессы вокруг нас имеют строго вероятностную природу, как, например, в теоретической механике, гидродинамике и астрономии. Квантовая механика и образованные от нее науки, к сожалению, несут проблему квантования полей многообразий. Естественно, логично ожидать подобных проблем и здесь. Можно заключить, что они будут наблюдаться исключительно на этапах обработки данных и оценке качества модели.

В литературе [4] и статьях [5–7] минимальной является модель TF-IDF, построенная на векторах слов. Модель LDA построена на том же скалярном произведении, но с предположениями, упрощающими ситуацию. Сконструируем данный алгоритм.

Евклидовым векторным пространством называется вещественное векторное пространство V с выделенной на нём симметричной билинейной формой $(x, y) \mapsto (x | y)$ такой, что соответствующая квадратичная форма $x \mapsto (x | x)$ положительно определена. Это пара $V, (*, *)$, где V — векторное пространство над \mathbb{R} , $(*, *)$ — фиксированная симметричная билинейная форма на V .

Свойства скалярного произведения:

- 1) $(x | y) = (y | x), \forall x, y \in V,$
- 2) $(\alpha x + \beta y | z) = \alpha(x | z) + \beta(y | z), \forall \alpha, \beta \in \mathbb{R},$
- 3) $(x | x) > 0 \quad \forall x \neq 0.$

Алгебраические правила над множеством слов мы уже определили. Необходимо «пополнить» пространство интегральными правилами, чтобы можно было корректно определять интегральные и вероятностные характеристики. Для этого построим топологию пространства. Из описания задачи также легко понять, что речь идет о тихоновских вполне регулярных пространствах. Понятно, что можно улучшить результат введя более строгую норму, например расстояние Левенштейна, но каждое уточнение метрики требует соответственно настроенного источника данных.

Вполне регулярное (тихоновское) пространство — такое хаусдорфово пространство X , если для каждой точки $x \in X$ и каждой ее окрестности U найдется непрерывная функция $f: X \rightarrow [0, 1]$ с $f(x) = 1$ и $f(y) = 0$ при всех $y \notin U$.

Согласно теореме Стоуна — Вейерштрасса, мы понимаем, что продолжения существуют.

Но, в отличие от традиционной задачи функционального анализа наша задача состоит не только в доказательстве единственности приближения и его сходимости, а также и в поиске «наилучшего» приближения, а также правильного выбора нормы для оценки модели на заключительном этапе моделирования.

В связи с этим, необходимо аккуратно правильно выбрать метрику для оценки, а также минимизировать ошибку, возникающую на каждом этапе обучения. Возможной теорией, помогающей решить данную проблему, будет теория аппроксимативно-компактных пространств Ефимова — Стечкина.

1.4. Применение чебышёвских множеств в NLP задачах

Задача наилучшего приближения ставилась еще в XIX веке П. Л. Чебышёвым.

Величина наилучшего приближения — расстояние от заданного элемента x линейного нормированного пространства X до заданного непустого множества $M \subset X$ до заданного непустого множества называется величина

$$\rho(x, M) := \inf_{x \in M} \|x - y\|. \quad (1)$$

Как мы понимаем, нас интересуют аппроксимативные свойства, то есть свойства существования, единственности и устойчивости элементов наилучшего приближения. Множество таких точек существует и называются такие множества *солнечными* или *чебышёвскими*.

Множество существования если для каждого x множество $P_M x$ её ближайших элементов непусто (соответственно, пусто или одноточечно).

$$P_M x := \{y \in M \mid \rho(x, y) = \|x - y\|\}. \quad (2)$$

В случае конечномерного X верно и обратное утверждение: любое замкнутое непустое множество является множеством существования.

На данный момент теория аппроксимации формализована достаточно хорошо и на данный момент можно ознакомиться с ней в работах Н. В. Ефимова и С. Б. Стечкина [Статья Ефимова-Стечкина, 10], а также и в [9]. С помощью развития данной теории можно выявлять и подбирать точки, подозрительные на точки, которые могут быть наилучшим образом приближены.

Затем по данным точкам можно построить алгоритм Дейкстры, и, ориентируясь на функцию потерь получить наилучшее решение.

1.5. Проблема выбора метрики и минимизация функции ошибки

Пусть $X \in \mathbb{R}^d$, $Y \in \mathbb{R}$.

Таким образом, можно переформулировать задачу следующим способом: необходимо по выборке $z = z^{(m)}$ состоящей из векторов $z^i = (x^i, y^i)$ распределенных по неизвестной нам мере ρ построить функцию $f_z : X \rightarrow Y$ минимизирующую функцию ошибки (не всегда среднеквадратичную).

Отсюда получаются следующие проблемы:

- 1) как правильно сконструировать меру для выбранного набора данных?
- 2) как корректно определить функционал ошибки для данной задачи?

В статье [10] авторы конструируют вероятностную меру, заранее оговариваясь, что она, вообще говоря, не определена корректно. Но, известно, что $Z = X \times Y$.

Опишем более подробно данное пространство с точки зрения теории меры. Но, что можно точно сказать, что задана на нашем классе множеств аддитивная функция:

$$\mu\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n \mu(A_i), \quad (3)$$

Для всех n и всех таких конечных наборов попарно непересекающихся множеств $A_i \in \mathcal{A}$, что $\bigcup_{i=1}^n A_i \in \mathcal{A}$. \mathcal{A} — такая алгебра множеств, что все пересечения, объединения и дополнения множеств принадлежат этому множеству.

Можно также сказать, что мера *счетно-порождена*, так как σ -алгебра \mathcal{A} порождена счетным набором множеств.

В результате, сконструирована мера Лебега.

Пусть I — куб в \mathbb{R}^n вида $[a, b]^n$, \mathcal{A}_0 — алгебра конечных объединений параллелепипедов из I с ребрами, параллельными координатным осям (что наводит на мысль использовать OLAP-кубы и колоночные базы данных на этапе предварительной обработки данных). По доказанному, данная мера счетно-аддитивна. Назовем данную конструкцию мерой Лебега и обозначим ее λ_n на σ -алгебре $\mathcal{L}_n(I)$. Значит, наше пространство измеримо по Жордану. Если предположить убывание функции распределения меры μ , то получим меру Лебега — Стильеса, которая и будет вероятностной. Без такого предположения очень трудно доказать существование интегральных оценок на пространстве Ефимова — Стечкина.

Для того, чтобы была возможность построения производных и интегралов необходимо ввести еще несколько определений. Назовем измеримое множество (в данном случае, по Лебегу — Стильесу) E *атомом* \mathcal{L} , если $\mu(E) > 0$ и нет измеримых множеств A с $0 < \mu(A) < \mu(E)$. Мера без атомов называется, соответственно, *безатомической*. В соответствии с предположением 9.1.11 из [9] существует функция распределения вероятностей.

Итак, в качестве заключения, можно сказать, что в модели TF-IDF для обработки естественного языка имеет смысл применять методы дискретного топологического анализа данных, в том числе и при выборе метрик. Как понятно из предыдущих выкладок понятно, что данные пространства называются аппроксимативно-компактными пространствами Ефимова — Стечкина.

Заключение

На основе топологического анализа аппроксимативно-компактных пространств предлагается алгоритм, который поможет находить похожие темы еще быстрее:

- 1) очистить данные, посредством алгоритмов кластеризации, основанных также на дискретных метриках (например, UMAP алгоритм, [11]);
- 2) разделить набор данных на несколько множеств, применить теорию чебышёвских множеств и выбрать наиболее ярких «представителей-солнц» данных множеств существования;
- 3) провести оценку качества модели, используя также дискретные метрики, например R^2 скорректированного коэффициента детерминации;
- 4) сравнить получившиеся несколько моделей и выбрать лучшую модель с помощью, например, критерия Аккаике.

Подводя итог, можно сказать, что в случае поисковых алгоритмов лучше всего использовать методы дискретной оптимизации, в то время как для алгоритмов спам-фильтрации оптимальным являются алгоритмы, основанные на вероятностных моделях, так как множество спам-писем, вообще говоря, неизвестно, то его без ограничения общности можно принять бесконечным.

Стоит отметить, что структура аппроксимативно-компактного пространства Ефимова — Стечкина сохраняется и при online-обучении, то есть при непрерывном поступлении новых данных. Подтверждение этому высказыванию можно найти в теореме о представлении Хана.

Наиболее характерным и явным минусом данного подхода, является крайне выраженная зависимость от вычислительной мощности сервера, на котором проводятся данные вычисления. Возможным способом снижения времени обучения без потери точности является использование видеокарт, ПЛИС (FPGA), а также создаваемые на данный момент квантовые компьютеры.

В дальнейшем планируется построение такого алгоритма на уже изученных достаточно хорошо источниках данных для проверки данной гипотезы.

Литература

1. *Zrnc, A.* Users' ability to perceive misinformation: An information quality assessment approach / Aljaž Zrnc, Marko Poženel, Dejan Lavbič // *Information Processing & Management.* – 2022.– № 59.– С. 102739.

2. *Rinner, B.* Evaluating KNN, LDA and QDA classification for embedded online feature fusion / Bernhard Rinner 2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing. – 2008. – С. 85–90.

3. Переобучение. – Режим доступа: <https://ru.wikipedia.org/wiki/Переобучение>. – (Дата обращения: 19.04.2022).

4. *Hobson, L.* Natural Language Processing in Action: Understanding, analyzing and generating text with Python: книга / Hobson Lane, Cole Howard, Hannes Max Napke – New York : Manning, 2019. – 1126 с.

5. *Blei, D. M.* Latent dirichlet allocation / Blei D M, Ng A Y and Jordan M I // *The Journal of Machine Learning.* – 2003.– № 3.– С. 993–1022.

6. *Griffiths, Th.* Finding scientific topics / Thomas Griffiths and Mark Steyvers // *PNAS.* – 2004.

7. Stochastic variational inference / Matthew D. Hoffman [и др.] // *The Journal of Machine Learning.* – 2013. – № 14. – С. 1303–1347.

8. *Blei, D. M.* Variational inference for Dirichlet process mixtures / Blei D M, Jordan M I // *Bayesian Analysis.* – 2006. – Т. 1.

9. *Богачев, В. И.* Экономика Основы теории меры. Том 2: книга / В. И. Богачев. – Москва-Ижевск : Регулярная и хаотическая динамика, 2003. – 576 с.

10. *Ефимов Н. В.* Некоторые свойства чебышевских множеств / Н. В. Ефимов, С. Б. Стечкин // *Докл. АН СССР.* – 1958. – Т. 118, № 1. – С. 17–19.

11. *McInnes, L.* UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction / Leland McInnes, John Healy, James Melville // *Computer Science.* – 2018.– № 3.

Строгонов Андрей Игоревич — студент 2-го курса магистратуры кафедры математического обеспечения ЭВМ Воронежского государственного университета.

E-mail: strogonovandrej@hotmail.com

КРАТКИЙ ОБЗОР ПРОТОКОЛОВ HTTP/1.1 И HTTP/2 И СРАВНЕНИЕ ИХ ПРОИЗВОДИТЕЛЬНОСТИ

В. Г. Строгонов

Воронежский государственный университет

Введение

Hypertext Transfer Protocol, или HTTP — это протокол прикладного уровня, который с момента изобретения (в 1989 году) является стандартом для связи во Всемирной паутине. С момента выпуска HTTP/1.1 в 1997 году и до недавнего времени в протокол было внесено не так уж много изменений. Но в 2015 году появилась его новая версия HTTP/2, в которой предлагается несколько способов снижения задержки, особенно при работе с мобильными платформами, а также с графикой и видео, интенсивно использующими сервер. С тех пор протокол HTTP/2 становится все более популярным: по некоторым оценкам, его поддерживают около трети всех веб-сайтов в мире. В этом переменчивом ландшафте веб-разработчики должны понимать технические различия между HTTP/1.1 и HTTP/2 и использовать их преимущества. Это позволяет принимать обоснованные и эффективные решения.

В рамках данной работы проводится обзор протоколов HTTP/1.1 и HTTP/2, рассматриваются их главные отличия, а также проводится анализ производительности.

1. Обзор протоколов HTTP/1.1 и HTTP/2

1.1. HTTP/1.1

Разработанный Тимоти Бернерсом-Ли в 1989 году в качестве стандарта связи для Всемирной паутины, HTTP — это протокол прикладного уровня, который обеспечивает обмен информацией между клиентским компьютером и локальным или удаленным веб-сервером (рис. 1). В этом процессе клиент отправляет текстовый запрос на сервер, вызывая метод (GET или POST). В ответ сервер отправляет клиенту ресурс, например, HTML-страницу.

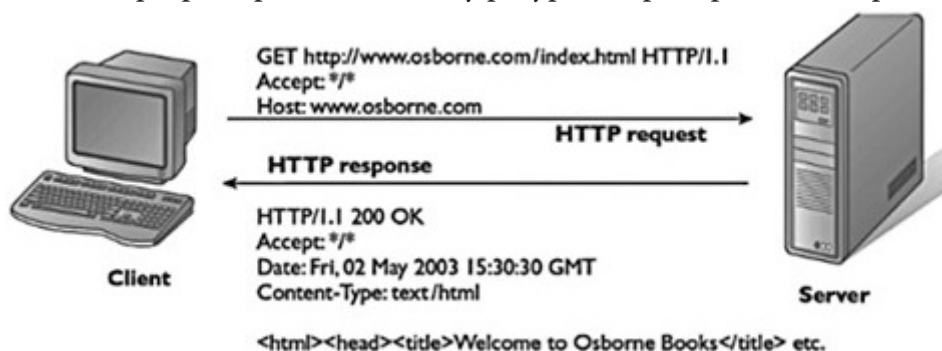


Рис. 1. Обмен информации между клиентом и сервером

В протоколе HTTP/1.1 появились следующие улучшения:

- Новые HTTP-методы PUT, PATCH, HEAD, OPTIONS, DELETE.
- Идентификация хостов. В протоколе HTTP/1.1 заголовок Host стал обязательным.
- Постоянные соединения. В HTTP/1.1 появились постоянные соединения, т. е. соединения, которые по умолчанию не закрывались, оставаясь открытыми для нескольких последо-

вательных запросов. Чтобы закрыть соединение, нужно было при запросе добавить заголовок Connection: close. Клиенты обычно посылали этот заголовок в последнем запросе к серверу, чтобы безопасно закрыть соединение.

- Поточковая передача данных, при которой клиент может в рамках соединения посылать множественные запросы к серверу, не ожидая ответов, а сервер посылает ответы в той же последовательности, в которой получены запросы.

HTTP/1.1 появился в 1999 и пробыл стандартом долгие годы. И, хотя он и был намного лучше своего предшественника, со временем начал устаревать. Веб постоянно растёт и меняется, и с каждым днём загрузка веб-страниц требует всё больших ресурсов. Сегодня стандартной веб-странице приходится открывать более 30 соединений. HTTP/1.1 поддерживает лишь одно внешнее соединение в любой момент времени. Разработчики пытались исправить это потоковой передачей данных, однако это не решало задачу полностью. Возникла проблема блокировки начала очереди — когда медленный или большой запрос блокировал все последующие (ведь они выполнялись в порядке очереди).

1.2. HTTP/2

HTTP/2 был разработан в 2015 году компанией Google с целью снижения задержки загрузки веб-страниц такими методами, как сжатие, мультиплексирование и приоритизация. С самого начала многие браузеры (включая Chrome, Opera, Internet Explorer и Safari) поддерживали эту попытку стандартизации. Частично благодаря этой поддержке с 2015 года наблюдается высокий уровень внедрения протокола, особенно среди новых сайтов.

Главные отличия от HTTP/1.1:

- В отличие от HTTP/1.1, в котором все запросы и ответы хранятся в простом текстовом формате, HTTP/2 использует двоичный уровень кадрирования для инкапсуляции всех сообщений в двоичном формате, при этом сохраняя семантику HTTP (методы, заголовки).

- Мультиплексирование — передача нескольких асинхронных HTTP-запросов по одному TCP-соединению.

- Сжатие заголовков.

- Добавлена технология Server Push, которая позволяет HTTP/2-совместимому серверу отправлять ресурсы HTTP/2-совместимому клиенту до того, как клиент запросит их.

- Приоритизация запросов.

- Безопасность.

Сейчас HTTP-сообщения состоят из одного или более фреймов. Для метаданных используется фрейм HEADERS, для основных данных — фрейм DATA. Каждый запрос и ответ HTTP/2 получает уникальный идентификатор потока и разделяется на фреймы. Фреймы представляют собой просто бинарные части данных. Коллекция фреймов называется потоком. Каждый фрейм содержит идентификатор потока, показывающий, к какому потоку он принадлежит, а также каждый фрейм содержит общий заголовок.

Поскольку HTTP/2 является бинарным протоколом, использующим для запросов и ответов фреймы и потоки, как было упомянуто выше, все потоки посылаются в едином TCP-соединении, без создания дополнительных. Сервер, в свою очередь, отвечает аналогичным асинхронным образом. Это значит, что ответ не имеет порядка, и клиент использует идентификатор потока, чтобы понять, к какому потоку принадлежит тот или иной пакет. Это решает проблему блокировки начала очереди — клиенту не придётся простаивать, ожидая обработки длинного запроса, ведь во время ожидания могут обрабатываться остальные запросы.

2. Сравнение производительности

Для замеров использовалось следующее оборудование:

- Сервер Oracle WebLogic Version: 12.2.1.3.0 с поддержкой HTTP/1.1 и HTTP/2. В платформу WebLogic Suite входят сервер приложений Jakarta EE, портал, интеграционные продукты, средства для разработки приложений и JRockit — собственная JVM компании. Данный сервер использовался для получения запросов и отправки ответов.

- Компьютер с установленной программой Insomnia, для отправки запросов. Insomnia - это простой, но мощный REST API-клиент с управлением файлами cookie, переменными среды, генерацией кода и аутентификацией

2.1. Замер времени выполнения Get запроса

Для замера времени было сделано 10,100 и 1000 Get запросов без Body используя протоколы HTTP/1.1 и HTTP/2 и посчитано среднее время от отправки запроса до получения полного ответа от сервера. Результаты замера приведены в табл. 1–3.

Таблица 1

Замер времени выполнения 10 Get запросов

Протокол	Среднее затраченное время (с)
HTTP/2	4,219
HTTP/1.1	6,621

Таблица 2

Замер времени выполнения 100 Get запросов

Протокол	Среднее затраченное время (с)
HTTP/2	39,376
HTTP/1.1	57,064

Таблица 3

Замер времени выполнения 1000 Get запросов

Протокол	Среднее затраченное время (с)
HTTP/2	262,941
HTTP/1.1	385,502

В результате выполнения данного замера было выявлено, что Get запросы с использованием протокола HTTP/2 в 1.5 раза быстрее чем при использовании HTTP1.1.

2.2. Замер времени выполнения Post запроса

Для замера времени было сделано 10,100 и 1000 Post запросов с Body используя протоколы HTTP/1.1 и HTTP/2 и посчитано среднее время от отправки запроса до получения полного ответа от сервера. Результаты замера приведены в табл. 4–6.

Таблица 4

Замер времени выполнения 10 Post запросов

Протокол	Среднее затраченное время (с)
HTTP/2	17.746
HTTP/1.1	26.776

Таблица 5

Замер времени выполнения 100 Post запросов

Протокол	Среднее затраченное время (с)
HTTP/2	154,345
HTTP/1.1	231,571

Таблица 6

Замер времени выполнения 1000 Post запросов

Протокол	Среднее затраченное время (с)
HTTP/2	782,133
HTTP/1.1	993,492

В результате выполнения данного замера было выявлено, что Post запросы с использованием протокола HTTP/2 также в 1.5 раза быстрее чем при использовании HTTP/1.1.

Заключение

В данной работе были кратко рассмотрены протоколы HTTP/1.1 и HTTP/2, а также было проведено сравнение их производительности. Анализируя результат можно сказать, что новая версия протокола HTTP имеет большой прирост в скорости работы, за счет использования бинарного протокола и мультиплексирования. Нет никаких сомнений, что для реализации веб-приложений необходимо использовать протокол HTTP/2, так как новые функции обеспечивают более высокий уровень контроля, который можно использовать для оптимизации производительности.

Литература

1. HTTP/1.1 vs HTTP/2: What's the Difference?. – Режим доступа свободный. – <https://www.digitalocean.com/community/tutorials/http-1-1-vs-http-2-what-s-the-difference>. – (Дата обращения: 01.04.2022).
2. HTTP/2 vs HTTP/1. – Режим доступа свободный. – <https://www.thewebmaster.com/hosting/articles/what-is-http2-and-how-does-it-compare-to-http1-1/>. – (Дата обращения: 01.04.2022).
3. Путь к HTTP/2. – Режим доступа свободный. – <https://habr.com/ru/post/308846/>. – (Дата обращения: 05.04.2022).
4. В ЧЕМ РАЗНИЦА МЕЖДУ HTTP/1.1 И HTTP/2?. – Режим доступа свободный. – <https://clck.ru/frPCz>. – (Дата обращения: 05.04.2022).

Строгонов Вячеслав Геннадьевич — магистрант 2-го курса кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: 432032@mail.ru

Борисенков Дмитрий Васильевич (научный руководитель) — канд. техн. наук, доцент кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: xuser@relex.ru

МОДЕЛИРОВАНИЕ ТРАНСПОРТНЫХ СИСТЕМ В СРЕДЕ ANYLOGIC СРЕДСТВАМИ БИБЛИОТЕКИ ДОРОЖНОГО ДВИЖЕНИЯ

Е. А. Струкова

Воронежский государственный университет

Введение

Имитационное моделирование является одним из основных подходов к исследованию транспортных систем. Настоящая работа посвящена описанию возможностей среды имитационного моделирования AnyLogic [1–2] при моделировании транспортных систем, а также её модуля под названием Библиотека дорожного движения [2–3]. В работе описываются компоненты и методы работы с данной библиотекой, проводится анализ их применения при имитационном моделировании различных транспортных систем. Приводится пример моделирования системы, в которой имеет место явление парадокса Браеса [4].

1. Библиотека дорожного движения, общие сведения и механизмы работы

Среди инструментов, предназначенных для имитационного моделирования, каждый имеет определённую сферу применения. В настоящей работе рассматривается среда AnyLogic [1–2], которая помимо элементов, имеющих у каждой модели (параметры, события, агенты и т.п.), имеет и специализированные объекты, вынесенные в отдельные библиотеки. В данной статье речь пойдёт о Библиотеке дорожного движения [2–3], предназначенной для моделирования транспортных систем различных конфигураций. Она предоставляет возможность существенно упростить процесс моделирования дорожной сети, а также сделать модель и наблюдаемые при её «жизни» процессы более наглядными. Стоит отметить, что в данной библиотеке реализованы правила и механизмы дорожного движения, что позволяет автомобилям, передвигающимся по моделируемой сети, «понимать», как правильно осуществлять это движения. Сюда входит распознавание дорожных знаков, разметки, выбор нужной для поворота полосы, схема манёвра перестроения, в общем, всё то, что необходимо знать для достижения пункта назначения.

Библиотека дорожного движения используется для решения таких задач, как моделирование дорог и автомагистралей, оценка пропускной способности и загруженности дорог, вычисление оптимального времени фаз светофоров, внедрение пешеходных переходов, парковок и общественных зданий в дорожную сеть и т.д. Для выполнения каждой из перечисленных задач в библиотеке определены соответствующие компоненты и их взаимодействие с передвигающимися по сети автомобилями.

2. Описание основных компонентов, входящих в библиотеку

В данном разделе описываются некоторые компоненты, входящие в библиотеку и необходимые для моделирования практически каждой транспортной системы. Таких элементов пять, причём изменение характеристик каждого из них может существенно изменить транспортную сеть, её пропускную способность, а также возможности движения по ней (например, с помощью изменения количества полос на дороге).

К основным компонентам Библиотеки дорожного движения относятся:

Дорога. Элемент, без которого не обходится ни одна транспортная система, причём для моделирования может использоваться как одна дорога, так и несколько. Основными параметрами данного элемента являются:

- **Направление движения.** Движение по дороге может быть как односторонним, так и двусторонним. Стрелками на дороге обозначено направление движения по ней.
- **Количество полос.** Для дороги может быть указано количество полос для каждого указанного направления движения. С помощью этой возможности можно моделировать, например, на данный момент не построенные реальные дороги, параллельно проводя анализ того, сколько полос можно выделить под каждое направление.
- **Ширина полосы.** Параметр, обычно анализируемый вместе с количеством полос. Такое исследование крайне важно при строительстве на местности, где есть строгое ограничение на ширину дороги, например, из-за наличия домов или каких-то других важных построек.

Перекрёсток. Элемент транспортной системы, образованный пересечением двух или более дорог. Основным компонентом любого перекрёстка являются соединители, с помощью которых можно указать разрешённые направления движения автомобилей на данном перекрёстке. По умолчанию устанавливаются все направления, которые разрешены правилами дорожного движения, но при моделировании реальных систем некоторые направления могут быть удалены (запрещены) из-за наличия каких-то индивидуальных для сети условий, например, запрещающих знаков или переходов с двусторонней дороги на одностороннюю.

Стоп-линия. Не является обязательным элементом сети, однако используется в большинстве из них. Стоп-линия может использоваться для разных целей в зависимости от конкретных ситуаций. Данный элемент можно задать в качестве пункта назначения автомобилей, а также использовать его в совокупности с дополнительными элементами. Например, если стоп-линия находится перед перекрёстком, то её можно использовать в соединении со светофором — для контроля над движением. Кроме того, с помощью встроенных функций для данного элемента, можно симитировать наличие дорожных знаков у стоп-линии таких, как «Уступи дорогу» и «Ограничение скорости». Для второго знака необходимо будет указать максимальное допустимое значение скорости, которого автомобили будут придерживаться от стоп-линии и до следующего перекрёстка, если не встретят других ограничений или отмены уже существующего.

Парковка. Элемент, предназначенный для моделирования парковочных мест в один ряд (если нужно несколько, то можно совместить два элемента данного типа), который обязательно должен примыкать к уже созданной дороге. Парковка может быть указана в качестве пункта назначения автомобилей. В качестве основных параметров для элементов этого типа задаётся тип парковки (параллельный или перпендикулярный), количество парковочных мест и длина парковочного места.

Парковку можно использовать в качестве отдельного элемента исследования. Допустим, есть дорога и необходимо проанализировать, каких размеров нужно будет построить парковку в зависимости от пропускной способности дороги, а также каким образом расположить на парковке автомобили; понять, достаточно ли одного ряда парковочных мест, или придётся выделить место под несколько. Для такого исследования нет необходимости строить большую дорожную сеть, достаточно лишь отдельно взятого участка дороги.

Автобусная остановка. Кроме передвижения личных автомобилей Библиотекой дорожного движения предусмотрена также возможность перемещения по транспортной сети общественного транспорта, в частности, автобусов. Для них же в библиотеке создан такой элемент как «остановка», который обязан примыкать к уже созданной дороге. Автобусная остановка также, как, например, стоп-линия, может быть указана в качестве пункта назначения не только для автобусов, но и для обычных автомобилей. Основным параметром для остановки считается длина. Участок дороги с остановкой может быть смоделирован для изучения того,

где её удобнее было бы расположить, а в совокупности с моделированием пешеходного потока, можно было бы вывести наиболее оптимальную периодичность выхода автобусов на маршрут и количество этих автобусов, чтобы сократить время ожидания.

В качестве примера можно привести схему транспортной сети, которую можно собрать с помощью Библиотеки дорожного движения (рис. 1). Представленная сеть может быть использована при моделировании транспортной системы, в которой имеет место явление парадокса Браеса [4].

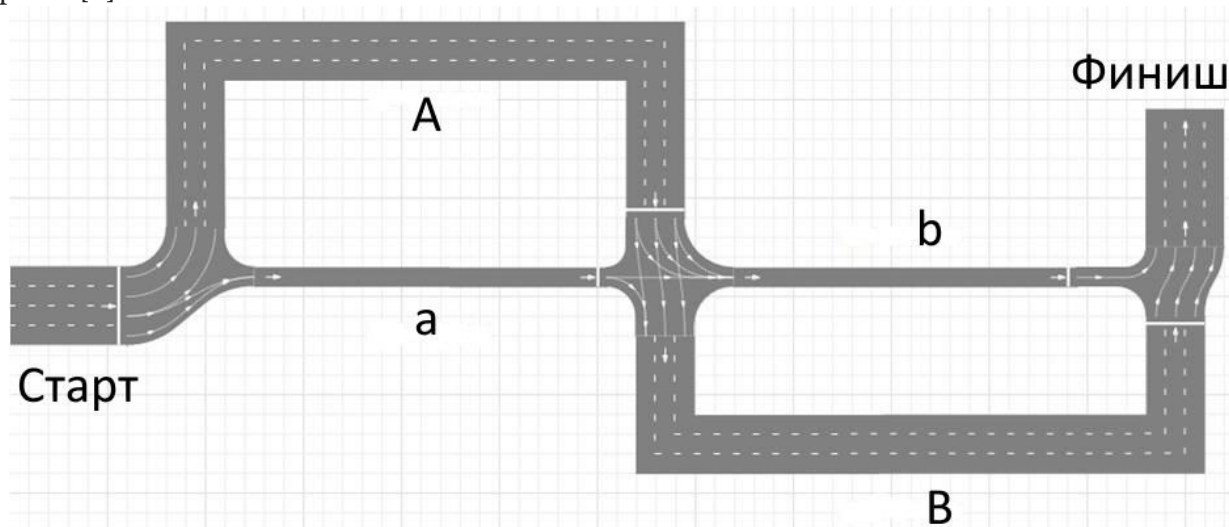


Рис. 1. Пример транспортной сети, собранной из компонентов Библиотеки дорожного движения

3. Организация транспортной сети

Для моделирования транспортной системы после добавления элементов, относящихся к рассмотренным в предыдущем разделе типам, их необходимо логически связать между собой. Это можно сделать с помощью сценариев движения потока машин и построения диаграммы дорожной сети. Пример соединения этих сценариев для сети, приведённой в разделе 2, представлен на рис. 2. В данном разделе рассматриваются основные сценарии движения.

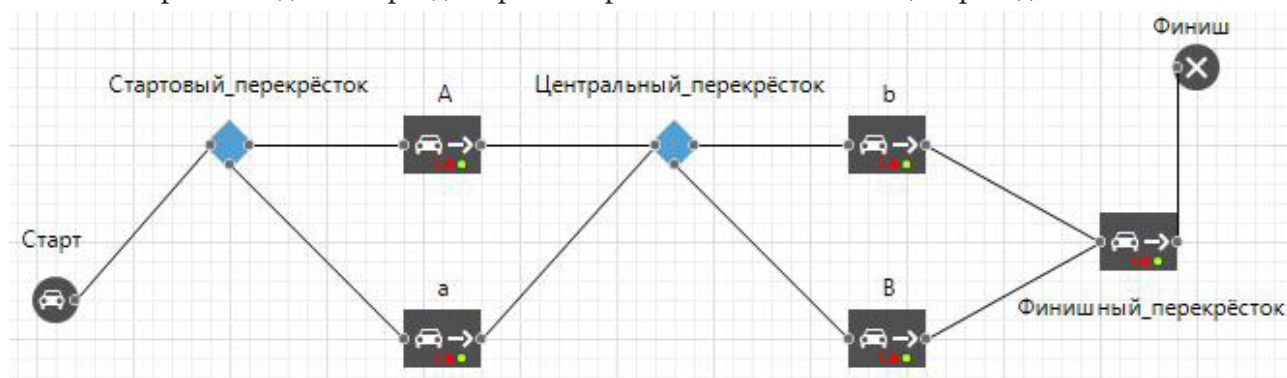


Рис. 2. Пример диаграммы дорожного движения, созданной в Библиотеке дорожного движения

CarSource. Блок отвечает за генерацию автомобилей в указанном месте дорожной сети, если это возможно. В качестве места появления автомобиля указывается либо дорога (какой-то её сегмент), либо парковка. Если указана дорога, то необходимо также выбрать, должен ли автомобиль появиться на полосе основного или встречного движения, а также на какой

именно полосе (можно выбрать случайную или какую-то конкретную). При моделировании реальных транспортных систем, естественно, этих данных не достаточно. Часто моделируются именно небольшие участки дорог, и поэтому для отображения близкой к действительности картины необходимо также указать значение такого параметра как «Прибывают согласно». Обычно для этого параметра выбирают «Интенсивности» или «Времени прибытия», причём в первом случае фиксируется, сколько автомобилей и за какое время должно появиться, а во втором случае — время между прибытием двух автомобилей.

CarMoveTo. Блок, отвечающий за передвижения автомобиля к указанному пункту назначения, если путь к нему существует. Таким пунктом назначения могут стать все базовые элементы, перечисленные в предыдущем разделе, кроме перекрёстка. Если путь до цели не будет найден, то автомобиль будет действовать так, как указано при выходе через порт `outWayNotFound`. Иначе автомобиль выполнить действия, соответствующие особенностям пункта назначения, например, если им является автобусная остановка или парковка, то автомобиль попытается занять свободное место. После выполнения описанного действия объект будет или переведён в следующий блок диаграммы движения `CarMoveTo`, или удалён из неё в принципе с помощью блока `CarDispose`.

CarDispose. Блок, отвечающий за удаление автомобилей из транспортной системы. При следовании этого блока после `CarMoveTo` по достижении пункта назначения автомобиль будет автоматически удалён из системы. Также можно осуществить это удаление при достижении парковки или автобусной остановки.

RoadNetworkDescriptor. Необязательный блок, с помощью которого можно получить доступ к управлению всем транспортом, находящимся в одной транспортной системе. Данный блок обладает следующими основными параметрами для настройки:

- Карта пробок на дорогах. Для наглядного представления и анализа можно настроить окраску участков дорожной сети в соответствии со средней скоростью передвижения по ним. Участок можно окрасить в зелёный цвет, если скорость выше установленного порога, например, выше 60 км/ч, или в красный – если скорость ниже установленного порога, например, 10 км/ч. С помощью этих данных можно сделать выводы о конкретных участках и предположить, какие изменения стоит внести, чтобы устранить эти пробки.

- Действия. Для более детального контроля над движением автомобиля можно расписать его действие в конкретных ситуациях таких, как вход в сеть, выход из неё, въезд на дорогу и перекрёсток и съезд с них, смена полосы и возобновление движения. Например, если для задачи требуется проанализировать то, сколько раз автомобиль сменит ту полосу, на которой он появился, тогда в пункте «Смена полосы» можно написать код, который бы переключал счётчик.

4. Интеграция классов, переменных и сценариев в библиотеку

В данном разделе рассматривается применение некоторых элементов среды `AnyLogic` при работе с Библиотекой дорожного движения.

Интеграция классов позволяет использовать при моделировании автомобиля, относящиеся к различным типам. Благодаря этому возможно, например, добавлять в систему легковые автомобили, грузовые, общественный транспорт и др.

По умолчанию при моделировании используется встроенный в библиотеку класс. Для придания отдельным автомобилям определённого набора характеристик можно создать пользовательский класс. При создании класса необходимо указать его имя, длину, начальную и предпочитаемую скорости, а также максимальные значения ускорения и торможения. Кроме того, можно изменить визуальное представление автомобиля, чтобы отличать объекты данного класса в общем потоке автомобилей и выбрать его двумерную или трёхмерную визуализацию.

Переменные и сбор статистики. Целью имитационного моделирования обычно является анализ моделируемой системы, её поведения и вариантов дальнейшей модернизации. Для такого анализа могут быть использованы переменные и сбор статистики, осуществляемый с их помощью. Выводы о состояниях системы можно сделать, например, основываясь на фиксации времени проезда определённого участка дорожной сети или сети в целом, время проезда конкретного перекрёстка и т.д. Есть также возможность проанализировать взаимодействие автомобилей с парковкой, сохраняя значение времени ожидания появления на ней свободного места. На основе этих данных можно сделать вывод о недостатке количества выделенных парковочных мест. Подобных счётчиков в одной модели может быть много и для различных целей, это зависит исключительно от целей рассмотрения конкретной задачи.

После того, как статистика с помощью счётчиков собрана, библиотекой дорожного движения предоставляется возможность создания на их основе отчёта, визуализирующего полученные данные. К примеру, могут быть построены диаграммы распределения времени проезда (рис. 3) по определённым участкам дорожной сети или распределение количества автомобилей на тех, которые ждали освобождения места на парковке и тех, кто занял свободное место сразу после приезда. Есть также отдельный класс задач по оптимизации, для которого крайне важен сбор статистики — настройка режима работы светофоров, как одного, так и нескольких вдоль одного маршрута. В таких задачах необходимо фиксировать время проезда первого светофора и всех последующих, если таковые имеются, и проводить анализ того, образуются ли пробки по ходу движения по данному маршруту, что будет, если уменьшить или увеличить время каждой фазы светофоров и т. д.



Рис. 3. Пример диаграммы распределения времени проезда, созданной в среде AnyLogic

5. Выводы о достоинствах и недостатках использования среды AnyLogic при моделировании транспортных систем

В данном разделе рассматриваются достоинства и недостатки использования среды AnyLogic и Библиотеки дорожного движения при имитационном моделировании транспортных систем.

Преимущества реализации имитационных моделей инструментами среды AnyLogic обусловлены удобством их использования [4]. К этим преимуществам относятся:

- понятный интерфейс, включающий в себя возможность визуализации моделируемой транспортной системы;
- наличие большого количества библиотечных компонентов, которые избавляют пользователя от необходимости самостоятельно реализовывать происходящие в транспортной системе процессы;

- множество инструментов, предназначенных для сбора статистической информации: среда предоставляет возможность строить различные графики, гистограммы и диаграммы на основе тех данных, которые необходимо структурировать, а затем проанализировать;

- возможность применения пользовательских классов, позволяющих моделировать конкретные транспортные системы.

К недостаткам рассматриваемого подхода относятся:

- достаточно высокие требования к ресурсам компьютера;

- наличие ограничения по времени в бесплатной версии ПО: время работы системы ограничено одним часом, что затрудняет получение абсолютно полных сведений о времени проезда большого количества автомобилей;

- недостаточная гибкость некоторых компонентов библиотеки.

В итоге можно сделать вывод о том, что использование Библиотеки дорожного движения целесообразно при моделировании отдельных перекрёстков и небольших участков дорожной сети, тогда как моделирование крупных транспортных систем требует, как минимум, дополнительных ресурсов, а как максимум – подключения дополнительных инструментов.

Заключение

В настоящей работе были рассмотрены и проанализированы средства среды AnyLogic, используемые при имитационном моделировании транспортных систем, приведён пример их использования при моделировании системы, в которой имеет место явление парадокса Браеса.

Работа выполнена под руководством канд. физ.-мат. наук, доцента кафедры вычислительной математики и прикладных информационных технологий ВГУ Королькова Олега Геннадьевича.

Литература

1. AnyLogic. – Режим доступа: <https://www.anylogic.ru/>.

2. Карпов, Ю. Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю. Г. Карпов. – Санкт-Петербург : БХВ-Петербург, 2005. – 400 с.

3. Обзор новой Библиотеки дорожного движения. – Режим доступа: <https://www.anylogic.ru/blog/obzor-novoy-biblioteki-dorozhnogo-dvizheniya/>.

4. Корольков, О. Г. О методах моделирования транспортных систем, демонстрирующих явление парадокса Браеса / О. Г. Корольков, Е. А. Струкова, А. А. Перфильева // Актуальные проблемы прикладной математики, информатики и механики: сборник трудов Международной научной конференции. – Воронеж, 2022. С. 530–536.

Струкова Екатерина Алексеевна — магистрант 1-го года обучения кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: katerina.strukowa@yandex.ru

ДИСКРЕТНАЯ МОДЕЛЬ ЗАДАЧИ УПРАВЛЕНИЯ ЗАПАСАМИ

П. И. Суркова, Е. П. Белоусова

Воронежский государственный университет

Введение

Дискретные процессы управления запасами приобретают все большее значение в теории управления. Это связано с тем, что большинство задач экономического планирования описываются разностными уравнениями, так как на практике чаще всего информация о состоянии процесса поступает в фиксированные моменты времени, и управление этим процессом осуществляется в дискретные промежутки времени.

Достаточно трудно построить универсальную обобщенную модель управления запасами, которая учитывала бы все возможные условия, наблюдаемые в реальных системах — например, колебания спроса в разные промежутки времени или возникновение необходимости создания страховых резервов. В этой работе описана задача приведения линейной дискретной системы из заданного начального состояния в заданное конечное состояние при наличии квадратичного критерия качества.

1. Начальные условия

Рассмотрим дискретную модель задачи управления запасами продукции на складе. Пусть n — количество видов продукции, m — количество видов используемых ресурсов (сырья), N — количество рассматриваемых периодов времени, $k = k_0$ — момент начала процесса. Далее возьмем

$$x(k) = \{x_1(k), \dots, x_n(k)\}, k = k_0, k_0 + 1, \dots, N - 1$$

— вектор количества товаров всех видов, имеющих на складе к концу k -го периода,

$$u(k) = \{u_1(k), \dots, u_m(k)\}, k = k_0, k_0 + 1, \dots, N - 1$$

— вектор ресурсов, затрачиваемых на производство единицы продукции,

$$w(k) = \{w_1(k), \dots, w_n(k)\}, k = k_0, k_0 + 1, \dots, N - 1$$

— вектор количества товаров, поставленных со склада в k -й период времени. Таким образом, уравнения, описывающие процесс, имеют вид:

$$x(k+1) = x(k) + Bu(k) - w(k), k = k_0, k_0 + 1, \dots, N - 1,$$

где $B_{n \times m}$ — технологическая матрица.

Также, задана взаимосвязь между количеством поставленных со склада товаров и количеством имеющих в наличии товаров:

$$w_i(k) = a_i(k)x_i(k), 0 \leq a_i(k) \leq 1.$$

В данной формуле коэффициенты $a_i(k)$ показывают, какая часть i -го товара будет поставлена в следующий период k .

Таким образом, уравнения, описывающие процесс, могут быть записаны в матричном виде:

$$x(k+1) = A(k)x(k) + Bu(k), k = k_0, k_0 + 1, \dots, N - 1,$$

$$\text{где } A_{n \times n}(k) = \begin{bmatrix} 1 - a_1(k) & 0 & \dots & 0 \\ 0 & 1 - a_2(k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 - a_n(k) \end{bmatrix}.$$

Кроме того, известны $x(k_0) = x_0$ — начальное состояние системы, и $x(N) = x_1$ — конечное состояние системы. Также, качество работы системы за период N оценивается квадратичным критерием:

$$I(u) = \sum_{k=k_0}^{N-1} u^T(k)u(k).$$

2. Постановка задачи

В данной работе требуется найти оптимальное программное управление $u^0(k)$, приводящее систему из начального состояния $x(k_0) = x_0$ в конечное состояние $x(N) = x_1$, а также, зная управление на каждом шаге, найти промежуточные состояния системы $x(k)$, $k = k_0 + 1, k_0 + 2, \dots, N - 1$. При этом управление $u^0(k)$ требуется выбрать так, чтобы величина критерия качества $I(u^0)$ была минимальной, то есть должно выполняться неравенство:

$$I(u^0) = \sum_{k=k_0}^{N-1} u^{0T}(k)u^0(k) \leq I(u) = \sum_{k=k_0}^{N-1} u^T(k)u(k).$$

3. Решение задачи

Введем обозначения:

$$c(k) = x_1 - A^{N-k}x(k),$$

$$S(k) = A^{N-k-1}B,$$

$$D(k) = \sum_{i=k}^{N-1} S(i)S^T(i).$$

Тогда оптимальное программное управление на k -м шаге будет иметь вид:

$$u^0(k) = S^T(k)D^+(k)c(k), \quad k = k_0, k_0 + 1, \dots, N - 1,$$

где $D^+(k) = \sum_{i=1}^r \lambda_i^{-1} v_i v_i^T$, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$ — это собственные значения матрицы $D(k)$, а v_1, v_2, \dots, v_r — это ортонормированные собственные векторы, отвечающие соответствующим собственным значениям.

Промежуточные состояния системы, будут зависеть от $u^0(k)$ и, как было сказано ранее, будут иметь вид:

$$x(k+1) = A(k)x(k) + Bu(k), \quad k = k_0, k_0 + 1, \dots, N - 1.$$

Стоит отметить, что реально для планирования будут использоваться близкие к найденным оптимальным управлениям величины $u(k)$, так как мы немного жертвуем минимальной величиной критерия качества в пользу целочисленных векторов $x(k)$.

4. Программное решение

Рассмотрим задачу, имеющую следующие условия. Количество видов товаров $n = 3$, количество видов затрачиваемых ресурсов $m = 2$, количество рассматриваемых периодов времени (месяцев) $N = 5$. Исходное состояние системы $x(k_0) = x_0 = (100, 50, 150)$, конечное состояние системы $x(N) = x_1 = (1000, 1800, 2000)$, $a_i(k)$, доли реализуемой продукции, отвечают табл. 1.

Также, задана технологическая матрица затрачиваемых ресурсов:

$$B = \begin{bmatrix} 0,1 & 0,05 \\ 0,04 & 0,1 \\ 0,3 & 0,1 \end{bmatrix}.$$

Таблица 1

$a_i(k)$	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$
$a_1(k)$	0,5	0,65	0,8	0,85	0,9
$a_2(k)$	0,5	0,6	0,7	0,8	0,85
$a_3(k)$	0,6	0,7	0,8	0,9	0,95

Требуется определить величины ресурсов $u(k)$, удовлетворяющие минимальному критерию качества, а также векторы количества товаров на конец k -го периода — $x(k)$ для такой организации системы, чтобы к концу N -го периода имелось x_i единиц продукции каждого вида соответственно.

Для решения поставленной задачи была использована система компьютерной алгебры Wolfram Mathematica. Код для решения данной задачи представлен в листинге 1.

Листинг 1

```

n=3; (* количество видов продукции *)
m=2; (* количество видов ресурсов *)
N1=5; (* количество периодов времени *)
k0=0; (* начальный момент времени *)
k1=1;
k2=2;
k3=3;
k4=4;
k5=N1; (* конечный момент времени *)
x[k0]={10, 50, 150}; (* исходное состояние системы *)
x[N1]={1000, 1800, 2000}; (* конечное состояние системы *)
(* матрицы коэффициентов ai(k) для каждого k *)
A[k0]={{0.5,0,0},{0,0.5,0},{0,0,0.4}};
A[k1]={{0.35,0,0},{0,0.4,0},{0,0,0.3}};
A[k2]={{0.2,0,0},{0,0.3,0},{0,0,0.2}};
A[k3]={{0.15,0,0},{0,0.2,0},{0,0,0.1}};
A[k4]={{0.1,0,0},{0,0.15,0},{0,0,0.05}};
E1={{1,0,0},{0,1,0},{0,0,1}};
B={{0.1,0.05},{0.04,0.1},{0.3,0.1}}; (* технологическая матрица *)
s[k_]:=x[N1]-((A[k])^(N1-k)).x[k]; (* вспомогательные векторы s(k) *)
s[k_]:=A[k]^(N1-k-1).B; (* вспомогательные матрицы S(k) *)
s[k0];
s[k1];
s[k2];
s[k3];
s[k4]=E1.B;
sT[k_]:=Transpose[s[k]]; (* транспонируем матрицы S(k) *)
sT[k0];
sT[k1];
sT[k2];
sT[k3];
sT[k4]=Transpose[E1.B];
ss[k_]:=s[k].sT[k];
ss[k4] = s[k4].sT[k4]; (* нахождение вспомогательных матриц D(k) *)
d[k0]=ss[k0]+ss[k1]+ss[k2]+ss[k3]+ss[k4] ;
d[k1]=ss[k1]+ss[k2]+ss[k3]+ss[k4];
d[k2]=ss[k2]+ss[k3]+ss[k4];
d[k3]=ss[k3]+ss[k4];
d[k4]=ss[k4];
(* нахождение собственных значений и собственных векторов для построения матриц D+(k) *)
sys1[k_]:=Eigensystem[d[k]];

```



```

sys1[k0];
sys1[k1];
sys1[k2];
sys1[k3];
sys1[k4];
(* построение матриц D+(k) *)
l1[k_] := sys1[k][[1,1]];
l2[k_] := sys1[k][[1,2]];
v1[k_] := sys1[k][[2,1]];
v2[k_] := sys1[k][[2,2]];
v1T[k_] := Transpose[Outer[Times,v1[k],v1[k]]];
v2T[k_] := Transpose[Outer[Times,v2[k],v2[k]]];
dp[k_] := ((l1[k])^(-1))*v1T[k]+((l2[k])^(-1))*v2T[k];
(* последовательное нахождение и вывод u(k) и x(k) *)
u[k_] := sT[k].dp[k].c[k]
Print[«u[k0] = »,u[k0]]
x[k1]=A[k0].x[k0]+B.u[k0];
Print[«x[k1] = », x[k1]]
Print[«»]
Print[«u[k1] = »,u[k1]]
x[k2]=A[k1].x[k1]+B.u[k1];
Print[«x[k2] = », x[k2]]
Print[«»]
Print[«u[k2] = »,u[k2]]
x[k3]=A[k2].x[k2]+B.u[k2];
Print[«x[k3] = », x[k3]]
Print[«»]
Print[«u[k3] = »,u[k3]]
x[k4]=A[k3].x[k3]+B.u[k3];
Print[«x[k4] = », x[k4]]
Print[«»]
Print[«u[k4] = »,u[k4]]
x[k5]=A[k4].x[k4]+B.u[k4];
Print[«x[k5] = », x[k5]]

```

Полученные результаты можно увидеть на рис. 1.

```

u[k0] = {389.679, 1148.14}
x[k1] = {101.375, 155.401, 291.717}

u[k1] = {336.926, 1149.46}
x[k2] = {126.647, 190.583, 303.539}

u[k2] = {403.54, 1590.51}
x[k3] = {145.209, 232.367, 340.821}

u[k3] = {962.025, 3604.9}
x[k4] = {298.229, 445.444, 683.179}

u[k4] = {907.386, 17023.3}
x[k5] = {971.724, 1805.44, 2008.7}

```

Рис. 1. Результаты программных вычислений

Стоит отметить, что решение не идеально — $x[k5]$ является лишь приближением к конечному вектору x_1 . Как уже было сказано ранее, в реальных задачах нам придется пожертвовать минимальным критерием качества при найденных $u(k)$ в пользу целочисленных элементов векторов $x(k)$.

Заключение

Рассмотрена задача приведения линейной дискретной системы в заданное конечное состояние x_1 при наличии квадратичного критерия качества. Построен алгоритм нахождения оптимального программного управления $u^0(k)$.

Стоит отметить, что задачи управления запасами среди группы экономических являются одними из наиболее сложных в математическом смысле. Эта особенность открывает широкие просторы для создания различных моделей, в зависимости от конкретных факторов и параметров системы.

Литература

1. Сазанова, Л. А. Синтез оптимального управления в линейных дискретных системах / Л. А. Сазанова // Труды института математики и механики УрО РАН. – 2000. – Т. 6, № 1-2. – С. 477–496.
2. Сазанова, Л. А. Дискретная модель управления запасами как задача оптимального управления / Л. А. Сазанова // Вестник Воронежского государственного университета. Серия: Экономика и управление. – 2017. – № 3. – С. 184–187.
3. Пропой, А. И. Элементы теории оптимальных дискретных процессов / А. И. Пропой. – М. : Наука, 1973. – 256 с.
4. Красовский, Н. Н. Теория управления движением : учебник / Н. Н. Красовский. – М. : Наука, 1968. – 476 с.
5. Воеводин, В. В. Линейная алгебра / В. В. Воеводин. – М. : Наука, 1980. – 400 с.

Суркова Полина Игоревна — студентка 4-го курса кафедры системного анализа и управления Воронежского государственного университета. E-mail: surkova.pi@yandex.ru

Белоусова Елена Петровна (научный руководитель) — канд. физ.-мат. наук, доцент кафедры системного анализа и управления Воронежского государственного университета. E-mail: e.p.belousova@gmail.com

ПОСТРОЕНИЕ РЕЙТИНГА ОБЪЕКТОВ

А. И. Сухоручкина, Е. М. Аристова

Воронежский государственный университет

Введение

Рейтинг — числовой или порядковый показатель, отражающий важность или значимость определённого объекта или явления. Одной из наиболее популярных форм предоставления информации являются рейтинги, которые характеризуют относительную значимость того или иного объекта или явления по сравнению с другими аналогичными объектами или явлениями. По сути, рейтинг представляет собой определенное место (позицию) анализируемого объекта среди аналогичного, определенное место расстановки всех объектов рейтинга, от более весомого — к менее весомому. В данной статье будет рассмотрено построение рейтинга автомобильных салонов с тремя показателями на основе различных методов для построения рейтинга, будет проведен анализ.

Постановка задачи построения рейтинга объектов

Рассмотрим задачу: необходимо выбрать лучший из пяти автосалонов для покупки автомобиля, учитывая их оценки по трем показателям. В качестве показателей выделим «расположение салона», «работу персонала» и «его репутацию».

Пусть для каждого из m объектов множества $X = \{1, 2, \dots, m\}$ заданы n числовых неотрицательных значений показателей (характеристик), т. е. задана матрица $A = (a_{ij})$, $i = \overline{1, m}$, $j = \overline{1, n}$.

В этой матрице каждому объекту $i \in X$ соответствует строка с номером i , а каждому показателю — столбец с номером j .

В матрице A нет ни одной строки с нулевыми значениями показателей, так как рассмотрение таких объектов не имеет смысла. В матрице A также отсутствуют столбцы, все элементы которых равны нулю.

Предполагается, что большие значения показателей более предпочтительны по отношению к меньшим их значениям. Характеристикам соответствуют весовые коэффициенты $\vartheta_j \geq 0$, $j = \overline{1, n}$, значения которых нормируются. Нормированные значения весовых коэффициентов

имеют вид $w_j = \frac{\vartheta_j}{\sum_{j=1}^n \vartheta_j}$, при этом $\sum_{j=1}^n w_j = 1$.

Вычисляются новые значения показателей с учетом весовых коэффициентов, $b_{ij} = a_{ij}w_j$. Значения показателей b_{ij} также нормируются: определяются $b_j = \max_{1 \leq i \leq m} b_{ij}$, нормированные значения x_{ij} показателей b_{ij} с учетом весов w_j имеют вид $x_{ij} = \frac{b_{ij}}{b_j}$, $i = \overline{1, m}$, $j = \overline{1, n}$. Полученные значения x_{ij} удовлетворяют условиям $0 \leq x_{ij} \leq 1$ и не имеют размерности. Отсутствие размерности делает осмысленными математические и логические операции над разными по содержанию показателями.

Рассматриваемая задача относится к классу многокритериальных задач, если в качестве критериев рассматривать показатели. При этом многокритериальная задача может быть сформулирована как задача, в которой мы стремимся придать каждому критерию максимально

возможное значение. С формальной точки зрения представленная задача — это задача кластеризации с некоторыми дополнительными условиями.

Задача кластеризации состоит в разделении исследуемого множества объектов на группы «похожих» объектов, называемых *кластерами*.

Задача кластеризации заключается в следующем. Имеется обучающая выборка $X^l = \{x_1, \dots, x_l\} \subset X$ и функция расстояния между объектами $\rho(x, x')$. Алгоритм кластеризации — это функция $a: X \rightarrow Y$ которая любому объекту $x \in X$ ставит в соответствие метку кластер $y \in Y$. Множество меток Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

Решение задачи кластеризации принципиально неоднозначно, и тому есть несколько причин. Во-первых, не существует однозначно наилучшего критерия качества кластеризации. Известен целый ряд достаточно разумных критериев, а также ряд алгоритмов, не имеющих чётко выраженного критерия, но осуществляющих достаточно разумную кластеризацию «по построению». Все они могут давать разные результаты. Во-вторых, число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием. В-третьих, результат кластеризации существенно зависит от метрики ρ , выбор которой, как правило, также субъективен и определяется экспертом.

Для решения задачи кластеризации рассмотрим несколько алгоритмов.

Алгоритм метода «идеальной точки»

1. Определяется «идеальная точка», т. е. лучшие параметры, с которыми сравниваются параметры рассматриваемых проектов. Такими идеальными параметрами могут быть лучшие значения параметров у всех рассматриваемых проектов или значения, определяемые экспертами $x_j = \max_{1 \leq i \leq m} x_{ij}$. Вектор $x = (x_1, \dots, x_n)$ соответствует «идеальному», гипотетическому, несуществующему объекту.

2. Число множеств $p > 1$. Зафиксируем в качестве расстояния одно из значений L_i . Вычислим $l = \min_i L_i$, $L = \max_i L$ и шаг $h = \frac{L-l}{p}$.

3. Распределим исследуемые объекты по отрезкам $[l + (i-1)h, l + ih]$, $i = \overline{1, p}$.

4. Определяются «расстояния» между значениями. Расстояния между значениями можно определить по формуле: $L_i = \max_{j \in J} |x_j - x_{ij}|$.

5. Затем проекты ранжируются по возрастанию расстояния до «идеальной точки» L_i . Лучшим считается проект, у которого расстояние до «идеальной точки» является минимальным.

Тогда в первый отрезок попадет объект, наиболее близкий к «идеальному», а в последний с номером p — наиболее далекий от него. Объекты, расстояния которых до «идеального», находятся в интервале с номером k , имеют рейтинг $q_k = k$ и образуют множество X_k .

Алгоритм линейной свертки показателей

1. Каждому объекту $i \in X$ ставится в соответствие линейная свертка его показателей $L_i = \sum_{j \in J} x_{ij}$, вычисляемая с применением значений весовых коэффициентов $b_{ij} = a_{ij} w_j$. Отметим, что при использовании данного алгоритма число множеств $p > 1$ задано.

2. Обозначим $l = \min_i L_i$, $L = \max_i L_i$ и вычислим их значения. Далее вычислим шаг $h = \frac{L-l}{p}$.

3. Распределим исследуемые объекты по отрезкам $[l + (i-1)h, l + ih]$, $i = \overline{1, p}$ в соответствии со значениями свертки L_i .

Тогда объект с минимальным значением L_i попадет в первый из отрезков, а с максимальным — в последний отрезок с номером p . Объекты, значения свертки которых находятся в

интервале с номером k , имеют рейтинг $q_k = p - k + 1$ и образуют множество X_{p-k+1} (чем выше рейтинг множества объектов, тем меньше его номер).

Метод анализа иерархий Саати

Метод состоит из следующих этапов:

1. Построение иерархии.
2. Получение оценок объектов, весовых коэффициентов критериев и подцелей для всех уровней иерархии.

Для получения оценок качественных критериев Саати рекомендует использовать парные сравнения. Объекты попарно сравниваются по специальной 9-балльной шкале. Результаты заносятся в матрицу парных сравнений [5].

3. Оценка однородности иерархии

$ИО = (\alpha_{\max} - n) / (n - 1)$ — индекс однородности, где α_{\max} — максимальное собственное значение матрицы; n — порядок матрицы.

$ОО = ИО / M(ИО)$ — отношение однородности, $M(ИО)$ — среднее значение (математическое ожидание) индекса однородности случайным образом составленной матрицы парных сравнений, которое основано на экспериментальных данных. Если для матрицы парных сравнений отношение однородности $ОО$ больше 0,1, то это говорит о нарушении логичности суждений эксперта. Поэтому эксперту предлагается пересмотреть данные, использованные для построения матрицы [5].

4. Расчет интегральных оценок объектов

Интегральная оценка объекта определяется как взвешенная сумма значений подцелей (критериев) второго уровня, которые, в свою очередь, вычисляются как взвешенные суммы значений критериев третьего уровня и так далее вплоть до получения значений критериев нижнего уровня иерархии [5].

Вычислительный эксперимент

Рассмотрим задачу про автосалоны. Пусть требуется выбрать лучший из пяти автосалонов для покупки автомобиля, учитывая их оценки по трем показателям («расположение салона» (А), «работа персонала» (Б) и «его репутация» (В)). Значения критериев (показателей) для всех альтернатив (салонов) приведены в табл. 1 по четырехбалльной системе оценок. Поставленную задачу будем рассматривать как задачу определения рейтинга объектов. В роли объектов выступают салоны (множество $S = (s_1, \dots, s_5)$, где s_i — i -й салон).

Оценки поставлены экспертами. Исходная шкала у всех показателей общая — {отл, хор, удовл, неудовл}. Оценки показателей для каждого салона, записанные в порядке их перечисления, образуют его векторную оценку.

Таблица 1

Значения показателей для салонов

Салон (номер по списку)	Оценки		
	А	Б	В
1	отл	хор	удовл
2	удовл	хор	отл
3	отл	неуд	удовл
4	отл	отл	хор
5	отл	удовл	неуд

Обозначаем показатели, как «отл» — 5, «хор» — 4, «удовл» — 3, «неудовл» — 2. Значения показателей представлены в таблице (табл. 2).

Таблица 2

Значения показателей

Варианты	Показатели		
	К1	К2	К3
s1	5	4	3
s2	3	4	5
s3	5	2	3
s4	5	5	4
s5	5	3	2

Матрица A_{ij} имеет вид:

$$A_{ij} = \begin{array}{c|c|c} 5 & 4 & 3 \\ \hline 3 & 4 & 5 \\ \hline 5 & 2 & 3 \\ \hline 5 & 5 & 4 \\ \hline 5 & 3 & 2 \end{array} .$$

Так как в задаче все показатели равноценны, то $w_j = \frac{1}{3}$. Новые значения показателей имеют вид:

$$B_{ij} = \begin{array}{c|c|c} \frac{5}{3} & \frac{4}{3} & 1 \\ \hline 1 & \frac{4}{3} & \frac{5}{3} \\ \hline \frac{5}{3} & \frac{2}{3} & 1 \\ \hline \frac{5}{3} & \frac{5}{3} & \frac{4}{3} \\ \hline \frac{5}{3} & 1 & \frac{2}{3} \end{array} .$$

Значения показателей нормируются по формуле $b_j = \max_{1 \leq i \leq m} b_{ij} : b_1 = \frac{5}{3}, b_2 = \frac{5}{3}, b_3 = \frac{5}{3}$.

Нормированные значения показателей b_{ij} с учетом весов w_j представим в виде матрицы

$$X_{ij} = \begin{array}{c|c|c} 1 & \frac{4}{5} & \frac{3}{5} \\ \hline \frac{3}{5} & \frac{4}{5} & 1 \\ \hline 1 & \frac{2}{5} & \frac{3}{5} \\ \hline 1 & 1 & \frac{4}{5} \\ \hline 1 & \frac{3}{5} & \frac{2}{5} \end{array} .$$

Видно, что полученные значения x_{ij} , $i = \overline{1,5}$, $j = \overline{1,3}$ удовлетворяют условиям $0 \leq x_{ij} \leq 1$. Для построения рейтинга объектов будем использовать вышеописанные алгоритмы.

Рассмотрим вначале *алгоритм идеальной точки*.

Пусть число множеств $p = 5$. В каждом из столбцов матрицы со значениями показателей x_{ij} выберем максимальный элемент $x_j = \max_{1 \leq i \leq 5} x_{ij}$.

Вектор $x = (x_1, x_2, x_3) = (1, 1, 1)$ соответствует «идеальному», гипотетическому, несуществующему объекту. Для каждого из объектов $i \in X$ определим расстояние до «идеального» объекта по формуле $L_i = \max_{j \in J} |x_j - x_{ij}|$: $L_1 = \frac{2}{5}$, $L_2 = \frac{2}{5}$, $L_3 = \frac{3}{5}$, $L_4 = \frac{1}{5}$, $L_5 = \frac{3}{5}$.

Согласно алгоритму, $l = \min_i L_i = \frac{1}{5}$, $L = \max_i L_i = \frac{3}{5}$, $h = \frac{L-l}{p} = \frac{2}{25}$. После распределения исследуемых объектов по отрезкам, получим следующую принадлежность к кластерам (всего пять множеств-кластеров I_1, I_2, I_3, I_4, I_5) — табл. 3.

Таблица 3
Распределение по множествам (кластерам)
с помощью алгоритма идеальной точки

Объект	Множество
1	I_3
2	I_3
3	I_5
4	I_1
5	I_5

Таким образом, первый и второй объекты имеют рейтинг, равный 3, третий и пятый объект имеет рейтинг, равный 5, а четвертый объект — рейтинг, равный 1. Следовательно, используя алгоритм идеальной точки, получим, что для покупки автомобиля будет выбран четвертый салон.

Рассмотрим теперь *алгоритм линейной свертки критериев*.

Пусть число множеств $p = 5$. Каждому объекту $i \in X$ ставится в соответствие линейная свертка его показателей $L_i = \sum_{j \in J} w_j x_{ij}$, вычисляемая с применением значений весовых коэффициентов w_j , $j = \overline{1,3}$: $L_1 = \frac{12}{5}$, $L_2 = \frac{12}{5}$, $L_3 = \frac{10}{5}$, $L_4 = \frac{14}{5}$, $L_5 = \frac{10}{5}$.

Согласно алгоритму $l = \min_i L_i = \frac{10}{5}$, $L = \max_i L_i = \frac{14}{5}$, $h = \frac{L-l}{p} = \frac{4}{25}$. После распределения исследуемых объектов по отрезкам, получим следующую принадлежность к кластерам (всего пять множеств-кластеров I_1, I_2, I_3, I_4, I_5) — табл. 4.

Таблица 4
Распределение по множествам (кластерам)
с помощью алгоритма линейной свертки критериев

Объект	Множество
1	I_4
2	I_4
3	I_1
4	I_5
5	I_1

Таким образом, первый и второй объекты имеют рейтинг, равный $q_5 = 5 - 4 + 1 = 2$, третий и пятый объекты имеют рейтинг, равный $q_4 = 5 - 1 + 1 = 5$, а четвертый объект – рейтинг, равный $q_1 = 5 - 5 + 1 = 1$. Следовательно, используя алгоритм линейной свертки критериев, получим, что лучшим салоном для покупки автомобиля будет выбран четвертый салон.

Наконец, рассмотрим *метод Саати*.

Построим иерархию. Рассмотрим метод по трем критериям: расположение салона, работа персонала и репутация (рис. 1).

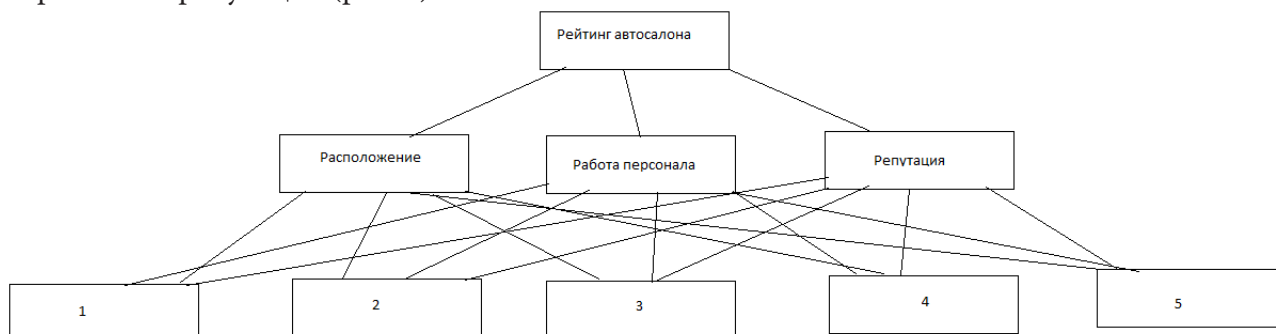


Рис. 1. Иерархия объектов

Рассчитаем весовые коэффициенты (табл. 5).

Таблица 5

Весовые коэффициенты

Название критерия	Расположение салона	Работа персонала	Репутация салона
Расположение салона	1	3	$\frac{1}{5}$
Работа персонала	$\frac{1}{3}$	1	$\frac{1}{7}$
Репутация салона	5	7	1

Вычислим весовые коэффициенты с помощью строчных сумм (табл. 6).

Таблица 6

Весовые коэффициенты с помощью строчных сумм

Критерий	Расположение	Работа персонала	Репутация	Строчные суммы	Нормированные значения весовых коэффициентов	Обозначение весовых коэффициентов
Расположение салона	1	3	$\frac{1}{5}$	4,2	0,224	w_1
Работа персонала	$\frac{1}{3}$	1	$\frac{1}{7}$	1,47	0,078	w_2
Репутация салона	5	7	1	13	0,696	w_3

Сравним автосалоны по критерию «Расположение салона» (табл. 7).

Таблица 7

Сравнение салонов по критерию «Расположение салона»

№ автосалона	№ автосалона					Строчная сумма	Нормированное значение критерия	Обозначение нормированного критерия
	1	2	3	4	5			
1	1	3	7	7	7	25	0,34	k_{11}
2	$\frac{1}{3}$	1	7	7	7	22,3	0,30	k_{12}
3	$\frac{1}{7}$	$\frac{1}{7}$	1	7	7	15,2	0,21	k_{13}
4	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	1	7	8,42	0,11	k_{14}
5	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	1	1,57	0,02	k_{15}
Сумма	1,7	4,42	15,2	22,1	29	72,61	1	

$$\alpha_{\max} = 8,38; \text{ ИО} = (\alpha_{\max} - n) / (n - 1) = 0,84; \text{ ОО} = \text{ИО} / M(\text{ИО}) = 0,75$$

Таблица 8

Сравнение салонов по критерию «Работа персонала»

№ автосалона	№ автосалона					Строчная сумма	Нормированное значение критерия	Обозначение нормированного критерия
	1	2	3	4	5			
1	1	6	4	7	5	23	0,36	k_{21}
2	$\frac{1}{6}$	1	4	$\frac{1}{7}$	5	10,3	0,16	k_{22}
3	$\frac{1}{4}$	$\frac{1}{4}$	1	$\frac{1}{7}$	$\frac{1}{5}$	1,84	0,02	k_{23}
4	$\frac{1}{7}$	7	7	1	7	22,1	0,34	k_{24}
5	$\frac{1}{5}$	$\frac{1}{5}$	5	$\frac{1}{7}$	1	6,54	0,10	k_{25}
Сумма	1,75	14,45	21	8,42	18,2	63,83	1	

$$\alpha_{\max} = 8,36; \text{ ИО} = (\alpha_{\max} - n) / (n - 1) = 0,84; \text{ ОО} = \text{ИО} / M(\text{ИО}) = 0,75$$

Таблица 9

Сравнение салонов по критерию «Репутация салона»

№ автосалона	№ автосалона					Строчная сумма	Нормированное значение критерия	Обозначение нормированного критерия
	1	2	3	4	5			
1	1	7	5	6	3	22	0,39	k_{21}
2	$\frac{1}{7}$	1	7	6	3	17,1	0,30	k_{22}
3	$\frac{1}{5}$	$\frac{1}{7}$	1	$\frac{1}{6}$	3	4,50	0,08	k_{23}
4	$\frac{1}{6}$	$\frac{1}{6}$	6	1	3	10,3	0,18	k_{24}
5	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	1	2,33	0,04	k_{25}
Сумма	1.84	8.64	19.3	13.5	13	56.31	1	

$$\alpha_{\max} = 7,91; \text{ ИО} = (\alpha_{\max} - n) / (n - 1) = 0,72; \text{ ОО} = \text{ИО} / M(\text{ИО}) = 0,65$$

Проведем расчет интегральных оценок (табл. 10).

Таблица 10

Расчет интегральных оценок

№ автосалона	Значение интегральной оценки	Рейтинг автосалона
1	0.37	1
2	0.29	2
3	0.10	4
4	0.18	3
5	0.04	5

Сравнивая полученные значения, можно определить рейтинг всех автосалонов. Высокий рейтинг будет соответствовать наибольшему значению. В приведенном примере, согласно исходным данным, наибольший приоритет оказался у первого автосалона.

Достоинством этого метода (в отличие от других экспертных) является возможность оценивать сразу и качественные, и количественные характеристики посредством перехода к безразмерным показателям.

С помощью этого метода можно осуществлять поиск оптимального решения в любой ситуации, так как он позволяет сравнивать все факторы одновременно, определяя значимость путем сравнения каждого с каждым.

Заключение

Рассмотренные в работе подходы могут применяться при решении задач ранжирования объектов в различных сферах деятельности, например, в задачах определения рейтинга выс-

ших учебных заведений, определения лучшего банка для вложения денежных средств. В данной статье был рассмотрен пример построения рейтинга автосалонов с помощью алгоритмов идеальной точки, линейной свертки и метода Саати. В результате работы был проведен анализ, получены приоритеты.

Литература

1. *Рыцарева, Е.* Рейтинг и «рейтинги» / Е. Рыцарева. – Москва : Эксперт, 2006. – № 37. – 177 с.
2. *Мандель, И. Д.* Кластерный анализ / И. Д. Мандель. – Москва: Финансы и статистика, 1988. – 176 с.
3. *Аристова, Е. М.* Определение рейтинга объектов / Е. М. Аристова // Вестник ВГУ, Серия: Системный Анализ и информационные технологии. – Воронеж, 2014. – № 2. – С. 51–56.
4. *Карминский, А. М.* Энциклопедия рейтингов: экономика, общество, спорт: монография / А. М. Карминский, А. А. Полозов. – 2-е изд., перераб. и доп. – Москва : ИД «ФОРУМ», ИНФРА-М, 2019. – 446 с.
5. Методика построения рейтингов доцент Соловьева Ю. А. – Режим доступа: <https://present5.com/metodika-postroeniya-rejtingov-docent-soloveva-yu-a/>.

Сухоручкина Алена Игоревна — студентка 4-го курса кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: aloyna.sukhoruchkina69@gmail.com

Аристова Екатерина Михайловна (научный руководитель) — канд. физ.-мат. наук, доц., доцент кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: pmim@yandex.ru

ОСНОВЫ ФОТОГРАММЕТРИИ

А. С. Сырых

Воронежский государственный университет

Введение

Перенос объектов из реального мира в виртуальный всегда был трудной задачей. Одним из возможных способов это сделать — воссоздать трехмерный объект в виртуальном мире по серии его снимков.

Получение трехмерной модели по серии изображений из реального мира является актуальной задачей и используется во многих областях нашей жизни. Так, например, в киноиндустрии этот процесс используется для создания цифровых дублеров. А в компьютерных играх перенос реальной модели может сэкономить время и деньги компании, а также привнести реализм в игру. Архитекторы могут планировать постройки прямо на месте выезда, а криминалисты — фиксировать места преступлений и исследовать их детальнее, находясь, как будто на месте преступления. Терапевты и другие медицинские работники могут сканировать части тела пациента для индивидуальной настройки протезов или слуховых аппаратов. В области культурного наследия перенос реального объекта в мир виртуальный также нашел свое применение. Фрагменты и памятники можно сохранить навсегда и использовать их для реконструкции в будущем, либо при создании виртуальных музеев. Таким образом, во многих областях нашей жизни требуется перенос объекта из реального мира в мир виртуальный.

Процесс получения модели из серии снимков называется фотограмметрией [1]. Фотограмметрия — это технология, которая позволяет создавать 3D-модели реальных объектов на основе фотографий этого объекта.

1. Базовый алгоритм фотограмметрии

Работу фотограмметрии можно разделить на три шага.

- 1) получение снимков объекта;
- 2) получение 3D модели из серии фотографий;
- 3) пост-обработка получившейся 3D модели.

1.1. Получение снимков объекта

На первом шаге подготавливаются снимки объекта. При отсутствии фотограмметрической установки лучше всего делать фотографии последовательно вокруг объекта. Сначала необходимо начать с круга под низким углом, затем сделать еще один под более высоким углом, чтобы захватить самые верхние поверхности. Также необходимо, чтобы перекрытие между соседними изображениями составляло не менее 50 %; 60–80 % — идеальный вариант. И, наконец, нужно сделать несколько дополнительных снимков тех участков, которые содержат важные, мелкие или сложные детали. Также важным моментом является постоянное освещение при съемке и отсутствие на фоне быть движущихся объектов. Обычно 40–50 фотографий достаточно для небольшого объекта.

1.2. Получение 3D модели из серии фотографий

Этот шаг, в свою очередь, можно разделить на четыре дополнительных шага:

- 1) сопоставление изображений;
- 2) сопоставление соответствий;
- 3) получение плотного облака точек;
- 4) триангуляция плотного облака точек.

Рассмотрим каждый этап подробнее.

1.2.1. Сопоставление изображений

Цель этого шага — найти изображения, которые относятся к одним и тем же участкам сцены. В результате этого шага получается компактный дескриптор изображения, который позволяет эффективно вычислять расстояние между всеми ключевыми точками изображений.

Одним из наиболее распространенных методов создания дескриптора изображения является метод словарного дерева [2]. Передавая в него все извлеченные дескрипторы признаков, дерево производит классификацию, сравнивая дескрипторы изображения с теми, которые находятся в каждом узле. Каждый дескриптор признака оказывается в одном листе, который может быть сохранен индексом этого листа в дереве. Дескриптор изображения в таком случае представляется коллекцией индексов используемых листьев.

1.2.2. Сопоставление соответствий

На данном этапе рассматриваются пары изображений (А и В) и ищутся наиболее близкие изображения. Цель этого шага — найти все пары-кандидаты и сопоставить все признаки между ними.

Сначала выполняется фотометрическое сопоставление набора дескрипторов двух входных изображений. Для каждого признака на изображении А благодаря дереву дескрипторов получается список признаков-кандидатов на изображении В. Поскольку пространство дескрипторов не является линейным и четко определенным, нельзя полагаться на абсолютные значения расстояния, чтобы узнать, является ли совпадение достоверным или нет. Чтобы удалить плохих кандидатов, делается предположение, что на другом изображении есть только одно достоверное совпадение. Поэтому для каждого дескриптора признака на первом изображении ищутся два ближайших дескриптора с использованием относительного порога между ними. Таким образом, получается список кандидатов на совпадение признаков, основанный только на фотометрическом критерии. Поиск двух ближайших дескрипторов на втором изображении для каждого признака требует больших вычислительных затрат при использовании метода перебора, но существует множество оптимизированных алгоритмов. Наиболее распространенными из них является метод ближайшего соседа [3] (метод k-ближайших соседей [3]) или более сбалансированные по скорости и качеству работы методы: SIFT [4], SURF [5], алгоритм Лукаса-Канаде [6].

Затем используется положение соответствий на изображениях для геометрической фильтрации с использованием эпиполярной геометрии для обнаружения выбросов под названием RANSAC (RANdom SAmple Consensus) [7]. Случайным образом выбирается небольшой набор соответствий и для каждой из виртуальных камер, из которой были получены изображения А и В вычисляется фундаментальная (или сущностная) матрица, которая описывает ориентацию камеры в пространстве (позиция и углы поворота).

Также на этом этапе с помощью алгоритма Bundle Adjustment [8] происходит удаление малоинформативных камер и минимизация ошибки перепроецирования для улучшения качества выходной модели.

1.2.3. Получение плотного облака точек

На этом шаге, на основании известного соответствия между изображениями и фундаментальными матрицами, полученных в пунктах 1.2.1 и 1.2.2, происходит построение плотного облака точек.

Для каждого изображения выбирается N лучших (ближайших) камер. После этого происходит рассмотрение параллельных плоскостей, основанных на пересечении оптической оси камер с точками выбранных соседних камер. Этот процесс рассмотрения плоскостей создает объем с множеством точек-кандидатов на точку глубины. Сходство оценивается для всех точек-кандидатов путем вычисления средней нормализованной кросс-корреляции (ZNCC) небольшого участка основного изображения, проецируемого на другую камеру. Таким образом, создается объем сходства. Для каждого соседнего изображения накапливается сходств в этом объеме. Однако получившийся объем очень шумный. В связи с этим, применяется шаг фильтрации по осям X и Y , который накапливает локальные значения, резко снижающие оценку изолированных высоких значений. Наконец, происходит выбор локальных минимумов и заменяется индекс выбранной плоскости значением глубины, хранящимся в карте глубины. Эта карта глубины имеет артефакты полосчатости, поскольку она основана на оригинальном выборе значений глубины.

Все эти карты глубины могут быть вычислены независимо друг от друга. Затем происходит этап фильтрации для обеспечения согласованности между несколькими камерами и удаления артефактов полосчатости.

1.2.4. Триангуляция плотного облака точек

Цель этого этапа — создать трехмерную модель из плотного облака точек, полученную из прошлого этапа.

Сначала объединяются все карты глубины, полученные в пункте 1.2.3 в глобальное восьмеричное дерево (octree) [9], где близкие значения глубины (относительного некоторого значения) объединяются в узлы восьмеричного дерева.

Затем выполняется трехмерная тетраэдрация. Одним из популярных методов тетраэдрации является тетраэдрация Делоне. Затем следует сложная процедура для вычисления весов ячеек и весов ребер [10, 12].

Для оптимального разрезания объема применяется теорема Форда-Фалкерсона [11]. Этот срез представляет собой поверхность сетки. Наконец, к сетке применяется фильтрация по Лапласиану для удаления локальных артефактов.

2. Пост обработка получившейся модели

После работы второго шага алгоритма на выходе получается зашумленная модель. Она может иметь большое число полигонов, и такую модель нельзя использовать в компьютерных играх. Все это подводит к тому, что модель необходимо обработать. При обработке у нее уменьшают число полигонов (убирая малозначимые) и убирают все ненужные части.

3. Обзор существующих программных продуктов

Существует множество программных обеспечений для фотограмметрии. Все они отличаются по цене, скорости работы и качеству выходной модели. Следует понимать, что все эти программные продукты, так или иначе, используют последовательность действий, описанную

в пункте 1. Различие заключается в том, как именно они производят поиск соответствий, извлечение признаков и построение плотного облака точек, с их последующей триангуляцией.

В табл. 1 представлены основные программные продукты, которые зарекомендовали себя с течением времени.

Таблица 1

Сравнение основных алгоритмов фотограмметрии

	Качество выходной модели	Скорость работы	Открытость технологии	Цена
3DF Zephyr	4/5	3/5	0/5	5/5
Reality Capture	5/5	5/5	2/5	2/5
Agisoft Metashape	4/5	2/5	1/5	4/5
Autodesk Recap	5/5	3/5	1/5	1/5

Здесь оценка производилась по 5-ти бальной шкале, где 0 — соответствует самому низкому показателю, а 5 — самому высокому.

Zephyr не является быстрым решением. Сканирование по умолчанию занимает около двух часов, а для высококачественного сканирования может потребоваться и 8-10 часов.

RealityCapture — отличное решение, обеспечивающее очень точные результаты сканирования. От других больше всего его отличает скорость обработки. Для механических объектов, которые включают плоские поверхности, RealityCapture имеет тенденцию давать несколько зашумленные результаты и для компенсации предлагает инструмент сглаживания. Алгоритмы лучше всего подходят для сканирования органических объектов без плоских элементов.

При сканировании механических объектов с плоскими поверхностями Agisoft Metashape дает менее шумные результаты, чем RealityCapture, но при этом на выходе может получиться менее детализированная модель, хотя при этом расчет может занимать в три раза больше времени. У этого продукта также больше всего проблем с направленным освещением, что может привести к значительным дырам в модели.

Программное обеспечение ReCap от Autodesk является самым сложным в своем классе и изначально основано на приложении, которое Autodesk приобрели в начале 2000-х годов под названием RealViz Image Modeler. Алгоритмы фотограмметрии Autodesk практически не имеют аналогов в отрасли и оказались лучшими в предоставлении сканов, пригодных для 3D-печати. Функция автоматической очистки является практически универсальной.

Заключение

Перенос модели из реального мира в виртуальный оставалась и остается сложной и актуальной задачей. Однако фотограмметрия помогает решить эту задачу.

Рассмотренный ранее фотограмметрический алгоритм можно разделить на три шага, где одним из основных и интересных остается второй шаг (получение 3D модели из серии фотографий).

Существует множество программных продуктов, которые позволяют построить модель по серии фотографий. Все они отличаются по скорости и качеству работы, однако используют одинаковый алгоритм, с некоторой модификацией отдельных его компонентов.

Например, в качестве сопоставления соответствий можно использовать SIFT, SURF или алгоритм Лукаса-Канаде.

А при триангуляции можно использовать тетраэдрацию Делоне, с последующим отсечением внутренней плоскости тетраэдра, либо же применить простую триангуляцию Делоне.

Не стоит забывать и про дальнейшую обработку модели, после ее получения из серии фотографий. Поскольку после второго шага алгоритма она получается шумной и с большим числом полигон.

Также, одним из ключевых моментов остается правильное создание серии фотографий объекта. Важно уделить внимание перекрытию фотографий. Обычно перекрытие должно составлять не менее 50 %, 70–80 % — идеальный случай.

Литература

1. What is Photogrammetry? – Режим доступа: <https://gisgeography.com/what-is-photogrammetry/>. – (Дата обращения: 20.04.2022).
2. David Nister. Scalable Recognition with a Vocabulary Tree / David Nister, Henrik Stewenius // CVPR – 2006.
3. A Simple Introduction to K-Nearest Neighbors Algorithm. – Режим доступа: <https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e#:~:text=K%20Nearest%20Neighbour%20is%20a,how%20its%20neighbours%20are%20classified.> – (Дата обращения: 20.04.2022).
4. Introduction to SIFT(Scale Invariant Feature Transform). – Режим доступа: <https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>. – (Дата обращения: 20.04.2022).
5. Introduction to SURF (Speeded-Up Robust Features). – Режим доступа: <https://medium.com/data-breach/introduction-to-surf-speeded-up-robust-features-c7396d6e7c4e>. – (Дата обращения: 20.04.2022).
6. Обучение алгоритму оптического потока Лукаса-Канаде. – Режим доступа: <https://russianblogs.com/article/1784133186/>. – (Дата обращения: 20.04.2022).
7. Konstantinos G. Derpanis Overview of the RANSAC Algorithm / Konstantinos G. Derpanis. – 2010.
8. An Efficient Planar Bundle Adjustment Algorithm. – Режим доступа: <https://arxiv.org/abs/2006.00187>. – (Дата обращения: 20.04.2022).
9. Octree. – Режим доступа: <https://www.geeksforgeeks.org/octree-insertion-and-searching/>. – (Дата обращения: 20.04.2022).
10. Jancosek Michal. Exploiting Visibility Information in Surface Reconstruction to Preserve Weakly Supported Surfaces / Michal Jancosek, Tomas Pajdla // CVPR – 2014.
11. Boykov Yuri. Multi-view reconstruction preserving weakly-supported surfaces / Yuri Boykov, Vladimir Kolmogorov // CVPR – 2004.
12. Jancosek Michal. Multi-view reconstruction preserving weakly-supported surfaces / Michal Jancosek, Tomas Pajdla // CVPR – 2011.

Сырых Александр Сергеевич — магистрант 1-го года обучения кафедры МО ЭВМ Воронежского государственного университета. E-mail: zsyryhz@mail.ru

ДАТЧИК ОПТИЧЕСКОЙ ПЛОТНОСТИ

О. А. Ткачев, С. А. Чернопятова, З. П. Чернецкий, И. Н. Науменко

Военный учебно-научный центр Военно-воздушных сил
«Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина»

Введение

В статье рассматривается конструкция, алгоритмы работы и структура программного обеспечения датчика оптической плотности для исследования дымовоздушной смеси. Предложено оригинальное конструктивное решение проблемы загрязнения оптических поверхностей датчика в процессе работы. Приводятся результаты реального применения датчика и алгоритмы его автоматической калибровки.

При исследовании дымовоздушной смеси актуальна задача измерения концентрации смеси. Причем концентрацию необходимо измерять в реальном масштабе времени в непрерывном потоке. Лабораторные методы анализа являются долгими по времени и довольно дорогостоящими. Одним из наиболее эффективных путей решения данной задачи является фотометрический метод. Он основан на анализе интенсивности светового потока проходящего через измеряемую среду.

Существующие конструкции приборов типа «ИОПД-24», «LASD-1-2», «ИКВЧ» и т.п. имеют сложную конструкцию, большие габариты, довольно высокую стоимость, кроме того оптические поверхности их сенсоров при длительном контакте с дымовоздушной смесью могут покрываться загрязняющими веществами, что существенно снижает точность измерения. Все эти факторы ограничивают применение существующих приборов-фотометров при исследованиях процессов дымогенерации. Предлагаемая конструкция позволяет успешно решить большинство из этих проблем.

Датчик оптической плотности дымовоздушной смеси состоит из выносного измерительного блока, размещаемого в трубопроводе подачи дыма, микропроцессорного блока выполняющего обработку полученных сигналов и передачу результатов на персональный компьютер (ПК) с установленным специальным программным обеспечением (ПО) [1].

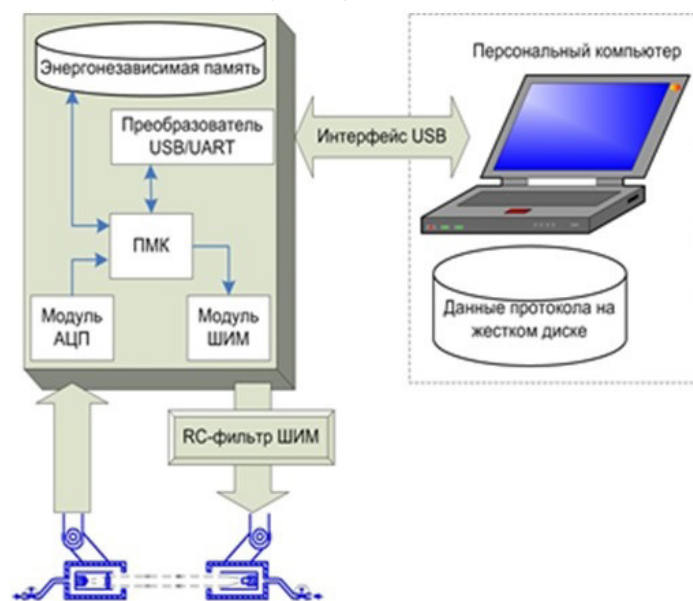


Рис. 1. Структурная схема датчика

Поскольку предлагаемый датчик используется, прежде всего, в исследовательских целях, то для снижения стоимости в нем не предусмотрены собственные элементы индикации (табло, дисплей и т. п.), а также отсутствуют органы ручного управления (тумблер, кнопки и т. п.) все настройки и конфигурирование прибора выполняется с ПК при помощи специального ПО. Структурная схема и внешний вид датчика приведены на рис. 1.

Экранная форма главного окна программы работы с датчиком (рис. 2) включает в себя область отображения данных (график), индикаторы величины текуще-

го значения оптической плотности и уровня мощности на который включен световой излучатель (лазерный модуль), а также панель настроек.



Рис. 2. Экранная форма главного окна программы работы датчика

Интерфейс программы даёт возможность пользователю плавно изменять мощность лазерного излучателя, обеспечивая тем самым широкий динамический диапазон измерения. В процессе работы датчика данные о текущей оптической плотности исследуемой среды представляются в графическом виде, а также сохраняются в текстовый файл. При этом имеется возможность их экспорта для дальнейшей обработки в стандартные математические пакеты, типа MS Excel, MathCAD, MathLAB и т.п.

Измерительный преобразователь (рис. 3) представляет собой расположенные друг напротив друга излучатель 2 и приемник 3 светового пучка. В качестве излучателя применяется лазерный модуль, а в качестве приемника – фотодиод.

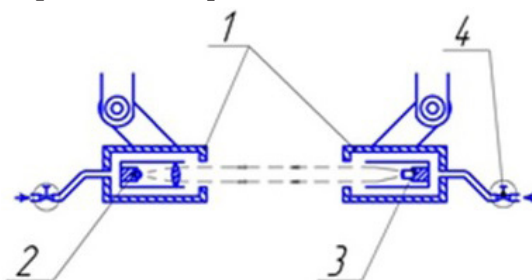


Рис. 3. Устройство первичного преобразователя

В процессе продолжительной работы датчика в дымовоздушной среде возникает проблема осаждения копильных веществ на оптических поверхностях [2]. Для предотвращения данного процесса, излучатель и приемник заключены в цилиндрические корпуса 1 с отверстиями для прохода светового пучка. В данные корпуса с небольшим расходом, ограниченным дросселем 4, подается воздух, создавая в них небольшое избыточное давление и препятствуя тем самым попаданию копильных веществ во внутреннее пространство. При этом оптические поверхности излучателя и приемника остаются чистыми на протяжении всего периода эксплуатации прибора [3].

Заключение

Интерфейс программы даёт возможность пользователю плавно изменять мощность лазерного излучателя, обеспечивая тем самым широкий динамический диапазон измерения. В процессе работы датчика данные о текущей оптической плотности исследуемой среды представляются в графическом виде, а также сохраняются в текстовый файл. При этом имеется возможность их экспорта для дальнейшей обработки в стандартные математические пакеты, типа MS Excel, MathCAD, MathLAB и т. п.

Литература

1. *Рогов, И. А.* Электрофизические, оптические и акустические характеристики пищевых продуктов [Текст] / И. А. Рогов, В. Я. Адаменко, С. В. Некрутман и др.; под ред. И. А. Рогова. – М. : Легкая и пищевая пром-сть, 1981. – 287 с.
2. *Рогов, И. А.* Электрофизические методы обработки пищевых продуктов [Текст]. – М. : Агропромиздат, 1988. – 487 с.
3. Smoking technology for the 80-s. Anonymous // Int. Flavours and Food Addit. – 1978. – No 6. – P. 265–266.

Ткачев Олег Александрович — канд. тех. наук, преподаватель кафедры общепрофессиональных дисциплин ВУНЦ ВВС «ВВА им. проф. Н. Е. Жуковского и Ю. А. Гагарина»
E-mail: tkachinyan@bk.ru.

Чернопятова Светлана Александровна — канд. тех. наук, преподаватель кафедры общепрофессиональных дисциплин ВУНЦ ВВС «ВВА им. проф. Н. Е. Жуковского и Ю. А. Гагарина».
E-mail: svvetkas@gmail.com

Чернецкий Захар Павлович — младший сержант 3-го курса ВУНЦ ВВС «ВВА им. проф. Н. Е. Жуковского и Ю. А. Гагарина»

Науменко Игорь Николаевич — курсант 1-го курса ВУНЦ ВВС «ВВА им. проф. Н. Е. Жуковского и Ю. А. Гагарина»

АРХИТЕКТУРА И ФУНКЦИОНАЛЬНОСТЬ CRM-РЕШЕНИЙ ДЛЯ МАЛОГО БИЗНЕСА

Д. И. Третьякова

Воронежский государственный университет

Введение

Самым ценным ресурсом любой компании являются клиенты. В современных экономических условиях важно максимально сократить отток потребителей и повысить их прибыльность. Хорошее знание клиента позволяет коренным образом изменить бизнес-модель: от неэффективной стратегии поиска новых потребителей под предлагаемые решения перейти к разработке решений для существующих клиентов с учетом их запросов и потребностей бизнеса. Персонализированное обслуживание позволяет повышать лояльность клиентов, а значит, повышать ценность клиентской базы как одного из ресурсов компании.

Использование CRM-систем в компаниях малого бизнеса оправдано в случае наличия большой клиентской базы и плана коммуникаций, отличающегося разнородностью задач. В этом случае, большинство процессов персонала удастся автоматизировать, а качество предоставления сервиса повысить [1]. При этом, существующие на рынке CRM-решения, либо являются универсальными и не решают задач, специфичных для отдельно взятых предприятий, либо обладают избыточной функциональностью, за которую клиент не готов платить. Таким образом, малый бизнес нуждается в решениях, которые будут максимально адаптированы под спектр оказываемых ими услуг, а стоимость ПО будет соразмерна доходам компаний малого бизнеса. Это и делает тему работы актуальной [2].

1. Сравнительный анализ функциональности существующих решений и постановка цели исследования

Цель статьи заключается в изучении варианта архитектуры CRM-решения, который позволит реализовать базовый функционал по работе с клиентами аутсорсинговых компаний. В ряд основных задач входит анализ функциональности CRM-решений, которые могут быть реализованы в аутсорсинговой компании, осуществляющей деятельность по бухгалтерскому и налоговому учету. Основным перечнем услуг подобных компаний является: ведение бухгалтерии клиентов, открытие и закрытие ИП и ООО, подготовка деклараций 3-НДФЛ, составление и сдача отчетности в ФНС, ПФР, ФСС и РОССТАТ [3].

Анализ функциональности готовых CRM-решений, которые могут быть использованы для автоматизации процессов таких компаний, показал, что базовым набором является тот, что представлен в табл. 1 [6–12]. Однако только им ПО не ограничивается. Это влияет и на сложность реализации продукта, и на конечную цену, что также является немаловажным фактором при выборе CRM-системы.

В связи с этим, ставится задача разработать и программно реализовать CRM-систему, отвечающую запросам малого бизнеса. Это значит, что приложение должно иметь только ту функциональность, которую планирует использовать компания, отбрасывая излишние для малых предприятий возможности.

Сравнительный анализ CRM-систем

Функциональные возможности	АмоCRM	Битрикс24	1С: CRM	Retail CRM	Мегаплан
Воронка продаж	+	+	+	+	+
База клиентов	+	+	+	+	+
История взаимодействия с клиентом	+	+	+	+	+
Мониторинг эффективности персонала	–	+	+	+	+
Тайм-менеджмент	–	+	+	–	+
Отчёты	+	+	+	+	+
Интеграция с почтой	+	+	+	+	+
Хранилище файлов	+	+	+	+	+
Экспорт/импорт данных	+	+	+	+	+

2. Функциональность CRM-решения

Учитывая, что целью внедрения CRM-системы является привлечение и удержание клиентов посредством установления и улучшения отношений с ними, то функциональность решения для рассматриваемых компаний подразумевает реализацию следующих процессов:

- создание и консолидация документов;
- ведение учета всех задач с предоставлением отчетов;
- формирование плана коммуникаций для телефонной связи;
- сбор статистики о работе сотрудников.

Такой набор с одной стороны будет минимальным, с другой – реализует полноценный сервис CRM, который позволит:

- получить общую для компании стандартизованную базу контактов (клиентов, контрагентов);
- осуществлять контроль качества работы сотрудников;
- получать статистику и аналитику работы со всеми инструментами плана коммуникаций;
- планировать повышение качества работы и разрабатывать стратегию развития бизнеса.

3. Архитектура CRM-приложения

Главная страница приложения должна содержать вкладки, реализующие основные режимы работы: добавить сделку, предстоящие задачи, клиенты, добавить клиента и статистику. За счет этого, пользовательский интерфейс будет с одной стороны максимально простым, но с другой — удобным для работы [13]. Если CRM система будет усложнять процесс взаимодействия с клиентами и увеличивать количество действий, которые потребуются сотрудникам для работы, то такая система останется невостребованной. При этом, каждая из вкладок интуитивно отображает весь бизнес-процесс осуществления сделки.

Работа в приложении по процессу «Добавить сделку» приведена на рис. 1–3. Изначально открывается окно для выбора услуги, а затем создания новой сделки. По каждой сделке формируется пул задач, которые сотруднику предстоит реализовать. Под задачами в статье подразумевается этап выполнения сделки.

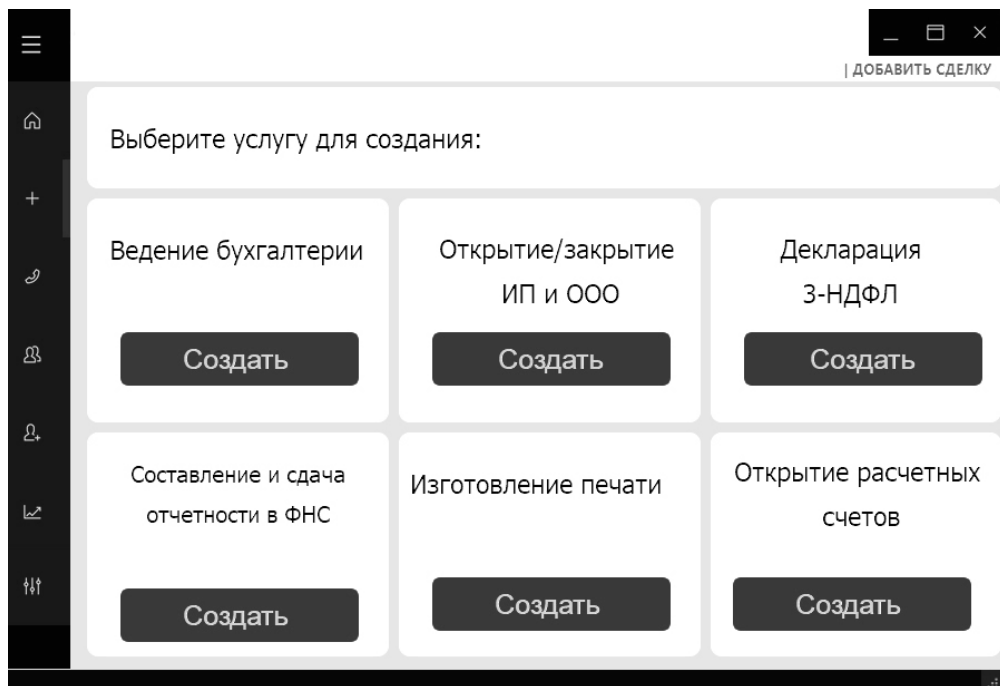


Рис. 1. Страница создания новой услуги

Для сбора статистики о сделках и задачах в частности, предусмотрен сервис, реализуемый по нажатию кнопки «Предстоящие задачи». В открывшемся окне есть возможность выбора вида задач: срочные, текущие или завершённые (рис. 3) с дальнейшей детализацией информации (рис. 4).

Под задачами в данном контексте подразумевается этап выполнения сделки. Так, например, для сделки «Составление и сдача отчетности в ФНС» имеются различные задачи, такие как: позвонить и договориться о встрече, напомнить клиенту о необходимости сдачи входящих документов, сообщить клиенту о некорректности входящих документов и вернуть их на доработку и т. д.

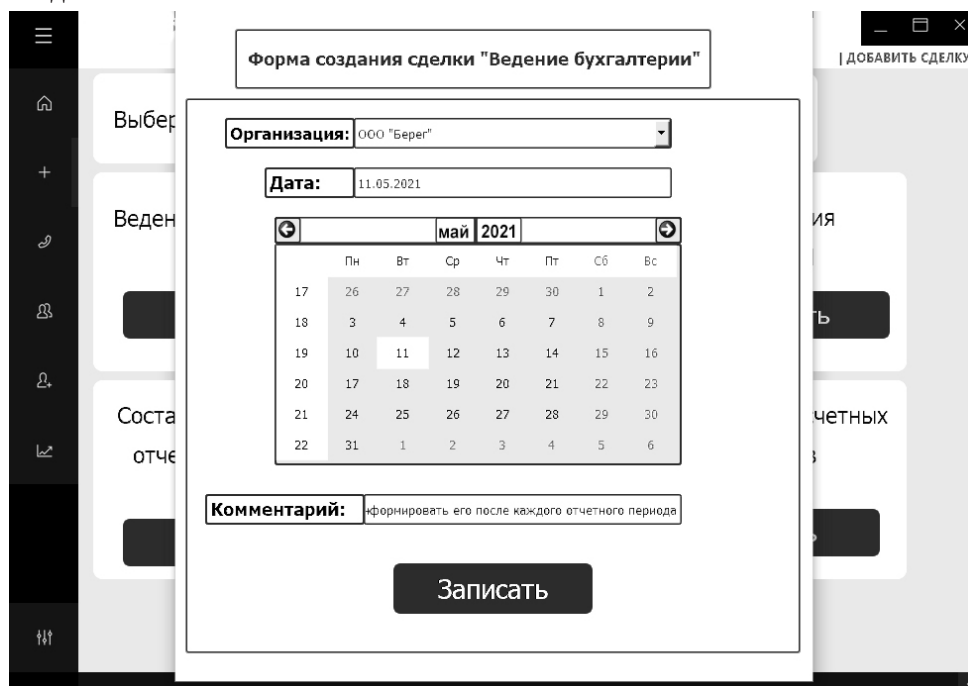


Рис. 2. Страница создания сделки

НОМЕР СДЕЛКИ	ДЕДЛАЙН	ЗАДАЧА	СДЕЛКА	ОРГАНИЗАЦИЯ
1	25.05.2021	Сообщить клиенту о некорректности данных в документах	Составление и сдача отчетности в ФНС	ООО "Первый шаг"
2	02.05.2021	Напомнить клиенту подать документы	Составление и сдача отчетности в ФНС	ООО "Северное сияние"
2	03.05.2021	Сообщить клиенту о выполненной услуге	Составление и сдача отчетности в ФНС	ООО "Северное сияние"
3	03.05.2021	Сообщить клиенту о перечне документов	Открытие ИП/ООО	ИП "Ногтевой шик"

Рис. 3. Страница просмотра статуса задач

Помимо этого, каждая из задач имеет свой дедлайн — крайний срок выполнения задачи. Если дедлайн приближается к текущей дате, то данная задача перемещается во вкладку «Срочные». Таким образом, можно увидеть какому клиенту, когда и по какому вопросу нужно позвонить, для того, чтобы успеть выполнить всю сделку в имеющиеся сроки.

Номер сделки	Дедлайн	Задача	Сделка	Организация
137	12.03.2021	Напомнить о сдаче ...	Составление и сдача ...	ООО "Пчеловод"
254	12.03.2021	Напомнить о сдаче ...	Составление и сдача ...	ООО "Пальто"
124	12.03.2021	Обговорить ...	Ведение бухгалтерии	ИП "Щит"
324	12.03.2021	Сообщить о ...	Открытие ИП	ООО "Кантри"
112	13.03.2021	Сообщить о ...	Составление и сдача ...	ИП "Победа"
107	13.03.2021	Сообщить ...	Открытие ООО	ООО "Звено"
127	13.03.2021	Сообщить о ...	Декларация 3-НДФЛ	ООО "Пальто"
134	13.03.2021	Сообщить о ...	Открытие Расчетных ...	ИП "Калина"
152	14.03.2021	Сообщить о ...	Составление и сдача ...	ИП "Инструмент"
182	14.03.2021	Напомнить о сдаче ...	Составление и сдача ...	ИП "Континент"
143	14.03.2021	Обговорить ...	Составление и сдача ...	ООО "Турист"
172	14.03.2021	Уточнить вопрос по ...	Ведение отчетности	ООО "Стрим"
241	14.03.2021	Сообщить о ...	Открытие ИП	ИП "Стрелков"

Рис. 4. Страница просмотра срочных задач

При такой организации интерфейса ПО пользователь сможет видеть кому, когда и по какому вопросу нужно позвонить клиенту, для того, чтобы успеть провести сделку в заданные сроки.

Заключение

CRM-решения с предложенной в статье архитектурой максимально прозрачны в реализации и не требуют специальной подготовки сотрудников для работы. При этом весь функционал, необходимый для малого бизнеса в них реализован. Архитектура не делает ограничений

на количество и тип реализуемых бизнес-процессов, что позволяет его адаптировать под любую компанию. Более того, CRM система с данной архитектурой позволяет быстро изменять настройки в зависимости от хода исполнения бизнес-процесса. Такая возможность позволит более полно определить и автоматизировать каждый процесс компании. За счет простой реализации ПО не относится, как большинство CRM к «тяжелому» ПО, а значит, не является затратным по системным ресурсам.

Литература

1. Бояршинов А. А. Потребность бизнеса в повышении лояльности клиентов [Электронный ресурс] / А. А. Бояршинов. – Электрон. текст. дан. – Санкт-Петербург : 2016. – URL: <http://www.marketing.spb.ru> (дата обращения 10.02.2022).
2. Богомолова М. А. Архитектура предприятия: учеб. пособие / М. А. Богомолова. – Самара : Изд-во ПГУТИ, 2016. – 155 с.
3. Бухгалтерский учет : учебное пособие / авт.-сост. Н. Н. Илышева, Е. Р. Синянская, О. В. Савостина. – Екатеринбург : Изд-во Урал. ун-та, 2016. – 156 с.
4. Варзунов А. В., Торосян Е. К., Сажнева Л. П. Анализ и управление бизнес-процессами // Учебное пособие. – Санкт-Петербург : Университет ИТМО, 2016 – 112 с.
5. Малис Н. И. Налоговый учет и отчетность : учебник и практикум для СПО / Н. И. Малис, Л. П. Грундел, А. С. Зинягина ; под ред. Н. И. Малис. – 3-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 407 с.
6. Официальный сайт CRM-системы «amoCRM»: демонстрация интерфейса «amoCRM». – URL: https://www.amocrm.ru/support/edit_contact (дата обращения 20.02.2022).
7. Официальный сайт CRM-системы «Битрикс 24» [Электронный ресурс]: демонстрация интерфейса «Битрикс 24». – URL: <https://www.bitrix24.ru/features> (дата обращения 10.01.2021).
8. Официальный сайт CRM-системы «1С:CRM» [Электронный ресурс]: демонстрация интерфейса «1С:CRM». – URL: <http://www.1crm.ru> (дата обращения 20.01.2022).
9. Официальный сайт CRM-системы «RetailCRM» [Электронный ресурс]: демонстрация интерфейса «RetailCRM». – URL: <https://www.retailcrm.ru> (дата обращения 15.01.2021).
10. Официальный сайт CRM-системы «Мегаплан» [Электронный ресурс]: демонстрация интерфейса «Мегаплан». – URL: <https://megaplan.ru> (дата обращения 20.01.2022).
11. Сайт Habr Интерфейсы CRM: главное, чтобы костюмчик сидел. – URL: <https://habr.com/ru/post/236915> (дата обращения 2.02.2022).
12. ТОП-10 лучших CRM систем для бизнеса – рейтинг 2018. – URL: <http://bisnesideya.ru/podborki/top-10-crm-sistem-dlya-biznesa.html> (дата обращения 10.01.2022).
13. Уклова В. В. Информационные системы в управлении предприятием: учебно-методическое пособие / В.В. Уклова; Воронежский государственный университет. – Воронеж : Издательский дом ВГУ, 2018. – 36 с.

Третьякова Дарья Ивановна — студентка 1-го курса магистратуры кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: tasha.tret@mail.ru

Уклова Вера Владимировна (научный руководитель) — канд. физ.-мат. наук, доцент кафедры математических методов исследования операций Воронежского государственного университета. E-mail: Ukhlova@amm.vsu.ru

ОБРАТНЫЙ АЛГОРИТМ ПРОЦЕДУР ДИАГНОСТИРОВАНИЯ ДИСКРЕТНЫХ ОБЪЕКТОВ

Тюлюш С. Т.

Калмыцкий государственный университет имени Б. Б. Городовикова

Введение

Дискретные объекты (ДО) являются элементами различных коммуникационных систем, например, компьютеров, мобильных телефонов, навигационных систем и т. п. Производство таких объектов растет и развивается. Проверки исправности и поиска возможных дефектов является одним из этапов производства и эксплуатации дискретных объектов. Стоимость аппаратуры зависит от быстрого поиска дефектов, то есть диагностирования объектов.

Алгоритм диагностирования (АД) — это совокупность предписаний о проведении диагностирования. Он устанавливает состав и порядок проведения так называемых элементарных проверок [1, 28].

Суть параллельного диагностирования заключается в одновременной (параллельной) подаче теста T в некоторое множество ДО и сравнении выходных сигналов с идентичных выходов с целью вычисления функции неравнозначности (ФН). Момент приобретения ФН значения «1» фиксируется и является исходной информацией для определения места нахождения обнаруженного дефекта [2].

Эффективность алгоритмов параллельного диагностирования множества однотипных дискретных объектов (ДО) такие как:

- параллельная процедура с повторными запусками теста;
- параллельная процедура с продолжением теста;
- параллельная процедура с неполными возвратами теста;
- v -процедура;
- v -процедура с неполными возвратами теста и другие.

доказаны в работах [2–5].

Предлагается новый алгоритм диагностирования дискретных объектов, который имеет выигрыш в суммарных временных затратах чем выше перечисленные алгоритмы.

Новый алгоритм поиска дефектов в дискретных объектах (ДО), назовем обратным алгоритмом. Докажем, что обратные алгоритмы последовательной процедуры, параллельной процедуры эффективнее алгоритмов перечисленных выше, то есть имеет выигрыш в суммарно временных затратах.

1. Сравнение обратного алгоритма с последовательной процедурой с реверсированием теста

Пусть разработан тест обнаруживающий все дефекты заданного множества в ДО, длина теста T , t_0 и t_k — соответственно начальный и конечный векторы теста.

Временная диаграмма последовательной процедуры с реверсированием теста, которая может быть интерпретирована как ориентированный граф, вершины которого соответствуют состояниям ДО в моменты обнаружения дефектов, а на дугах задана числовая функция значений времен поиска обнаруженного дефекта, приведена на рис. 1.

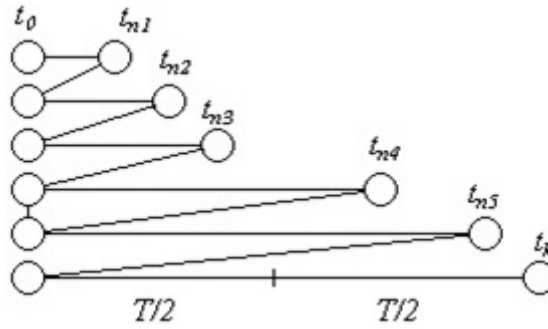


Рис. 1. Временная диаграмма последовательного алгоритма обнаружения пяти дефектов в ДО ($\mu = 5$)

Суть предлагаемого обратного алгоритма заключается в следующем:

- тест подаётся в ДО и после обнаружения первого дефекта, через время t_{n1} , реверсируется в исходное состояние t_0 , рис. 2.

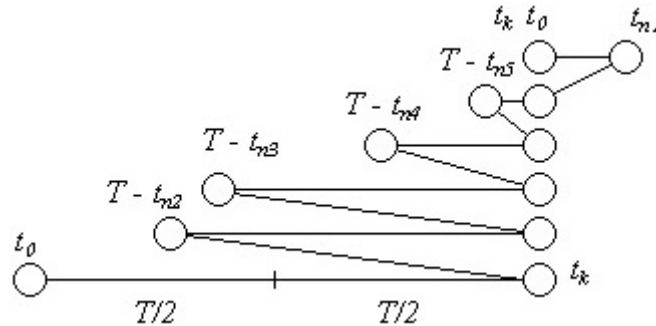


Рис. 2. Временная диаграмма обратного алгоритма обнаружения пяти дефектов в ДО ($\mu = 5$)

- после устранения обнаруженного дефекта (первого) тест не повторяется снова, как на рис. 1, а запускается в обратном порядке, то есть формируются сигналы соответствующие тестовому вектору t_{k-1} , t_{k-2} , ... и т. д., которые (сигналы) и подаются в ДО.

Так как моменты обнаружения дефектов распределены по длине теста по закону показанному на рис. 2, то теперь обнаруживается пятый дефект, время поиска которого при прямой подаче теста t_{n5} , но сейчас на его поиск потребовалось время $T - t_{n5}$.

- после этого тест реверсируется в исходное состояние и обнаруженный дефект устраняется.

- при очередном запуске теста в обратном порядке обнаруживается следующий дефект (четвёртый), время поиска этого дефекта будет $T - t_{n4}$.

Процесс повторяется, в результате чего для рассматриваемого примера обнаруживаются третий и второй дефекты, на поиск которых при обратной подаче теста затрачивается время $T - t_{n3}$ и $T - t_{n2}$.

Последний прогон теста из состояния t_k в t_0 , на котором не обнаруживается ни одного дефекта, свидетельствует об исправности ДО.

Суммарные временные затраты (СВЗ), на поиск всех (пяти) дефектов в ДО по последовательному алгоритму, в соответствии с формулой (1), вычисляются с помощью выражения:

$$T_{\text{послед}} = t_{i_1} + t_{i_2} + t_{i_3} + t_{i_4} + t_{i_5} + T$$

или в общем виде:

$$\dot{O}_{\text{послед}} = \sum_{i=1}^{\mu} t_{i_1} + T, \quad (1)$$

где μ — количество дефектов в ДО.

На основании рис. 2 для алгоритма, реализующего обратный прогон теста, СВЗ для рассматриваемого примера определяются с помощью выражения:

$$T_{\text{обрат.}} = t_{\text{п}_1} + (T - t_{\text{п}_2}) + (T - t_{\text{п}_3}) + (T - t_{\text{п}_4}) + (T - t_{\text{п}_5}) + T$$

или в общем виде:

$$T_{\text{обрат.}} = t_{\text{п}_1} + \sum_{i=2}^{\mu} (T - t_{\text{п}_i}) + T. \quad (2)$$

Пусть для рассматриваемого примера

$$t_{\text{п}_1} = 3, \quad t_{\text{п}_2} = 6, \quad t_{\text{п}_3} = 9, \quad t_{\text{п}_4} = 15, \quad t_{\text{п}_5} = 18, \quad T = 20,$$

тогда на основании выражений (1) и (2) СВЗ для последовательного алгоритма равны:

$$T_{\text{послед.}} = 3 + 6 + 9 + 15 + 18 + 20 = 71 \text{ условная единица.}$$

А для обратного алгоритма:

$$T_{\text{обрат.}} = 3 + (20 - 18) + (20 - 15) + (20 - 9) + (20 - 6) + 20 = 55 \text{ условных единиц.}$$

Соответственно выигрыш во времени от применения обратного алгоритма по сравнению с последовательным:

$$w = 71/55 = 1.29$$

или в общем виде:

$$w = \frac{T_{\text{послед.}}}{T_{\text{обрат.}}} = \frac{\sum_{i=1}^{\mu} t_{\text{п}_i} + T}{t_{\text{п}_1} + \sum_{i=1}^{\mu} (T - t_{\text{п}_i}) + T}. \quad (3)$$

Анализируя рис. 2 можно заметить, что при обратном алгоритме СВЗ тем меньше, чем больше дефектов обнаруживаются во второй половине теста, то есть чем меньше величина $T - t_{\text{п}_i}$.

Следовательно, общее число обнаруживаемых дефектов μ можно разбить на два подмножества:

- μ_1 — количество дефектов, обнаруживаемых в первой половине теста;
- μ_2 — количество дефектов, обнаруживаемых во второй половине теста;

Примем среднее время обнаружения всех дефектов подмножества μ_1 $t_{\text{сред}\mu_1} = T/4$, а подмножества μ_2 $t_{\text{сред}\mu_2} = 3T/4$, тогда формулу (3) можно преобразовать

$$w = \frac{T_{\text{послед.}}}{T_{\text{обрат.}}} = \frac{\left(\mu_1 \frac{T}{4} + \mu_2 \frac{3T}{4}\right) + T}{t_{\text{п}_1} + \mu_1 \left(T - \frac{T}{4}\right) + \mu_2 \left(T - \frac{3T}{4}\right) + T}. \quad (4)$$

На рис. 3 построен график коэффициента соотношения w между временными затратами при последовательном алгоритме и временными затратами при обратном алгоритме в зависимости от величины μ_1/μ_2 , то есть соотношения между количеством дефектов обнаруживаемых в первой половине теста и количеством дефектов обнаруживаемых во второй половине теста, при этом длина теста $T = 100$, количество обнаруживаемых дефектов $\mu = 20$, а $t_{\text{п}_1} = T/4$.

Анализируя график рис. 3 можно сделать следующие выводы:

Предложенный обратный алгоритм обнаружения дефектов в дискретных объектах обеспечивает выигрыш во времени по сравнению с последовательным алгоритмом в тех случаях, когда число дефектов обнаруживаемых во второй половине теста больше числа дефектов обнаруживаемых в первой половине теста, то есть $\mu_2 > \mu_1$.

При $\mu_1 = \mu_2$ алгоритмы с точки зрения временных затрат адекватны.

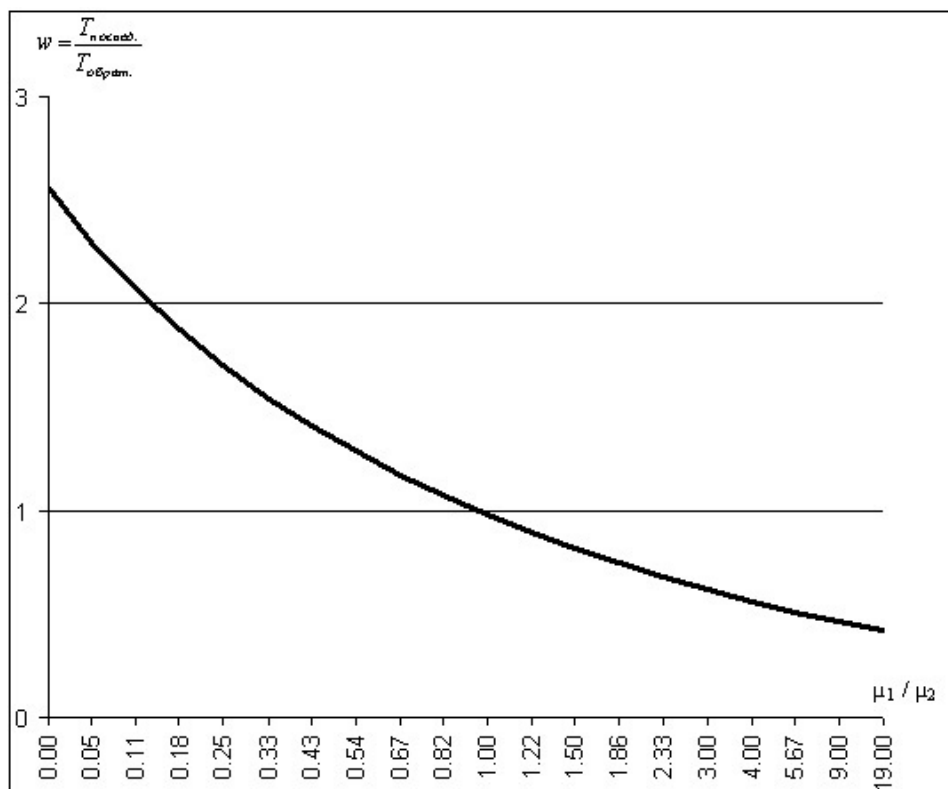


Рис. 3. Изменение коэффициента соотношения w между временными затратами при последовательном алгоритме и временными затратами при обратном алгоритме в зависимости от величины $\mu_1 = \mu_2$

При $\mu_1 > \mu_2$, то есть, если закон распределения моментов обнаружения дефектов по длине используемого теста таков, что большинство из них обнаруживаются в первой половине теста, то применение обратного алгоритма не эффективно.

2. Обратный алгоритм и параллельная процедура поиска дефектов с реверсированием теста

Приведем пример для параллельной процедуры. Временная диаграмма параллельной процедуры диагностирования трех ДО с разными дефектами $\mu_1 = 4$, $\mu_2 = 2$, $\mu_3 = 6$ приведена рис. 4, длина теста T , t_0 и t_k — соответственно начальный и конечный векторы теста. И пусть

$$t_{n_1}^{(1)} = 10, t_{n_2}^{(1)} = 15, t_{n_3}^{(1)} = 80, t_{n_4}^{(1)} = 95,$$

$$t_{n_1}^{(2)} = 30, t_{n_2}^{(2)} = 150, t_{n_3}^{(2)} = 190,$$

$$t_{n_1}^{(3)} = 8, t_{n_2}^{(3)} = 20, t_{n_3}^{(3)} = 40, t_{n_4}^{(3)} = 100, t_{n_5}^{(3)} = 170, t_{n_6}^{(3)} = 180, T = 200,$$

Суммарные временные затраты параллельной процедуры определяются в виде:

$$T_{\text{пар.}} = [t_{n_1}^{(1)} + t_{n_2}^{(1)} + t_{n_3}^{(1)} + t_{n_4}^{(1)}] + [t_{n_1}^{(2)} + t_{n_2}^{(2)} + t_{n_3}^{(2)}] + [t_{n_1}^{(3)} + t_{n_2}^{(3)} + t_{n_3}^{(3)} + t_{n_4}^{(3)} + t_{n_5}^{(3)} + t_{n_6}^{(3)}] + T$$

или в общем виде:

$$\dot{O}_{\text{пар.}} = \sum_{j=1}^N \sum_{i=1}^{\mu_j} t_{n_i}^j + T, \quad (5)$$

где μ — количество дефектов в ДО.

$T_{\text{пар.}} = [10 + 15 + 80 + 95] + [30 + 150 + 190] + [8 + 20 + 40 + 100 + 170 + 180] + 200 = 1288$ условных единиц.

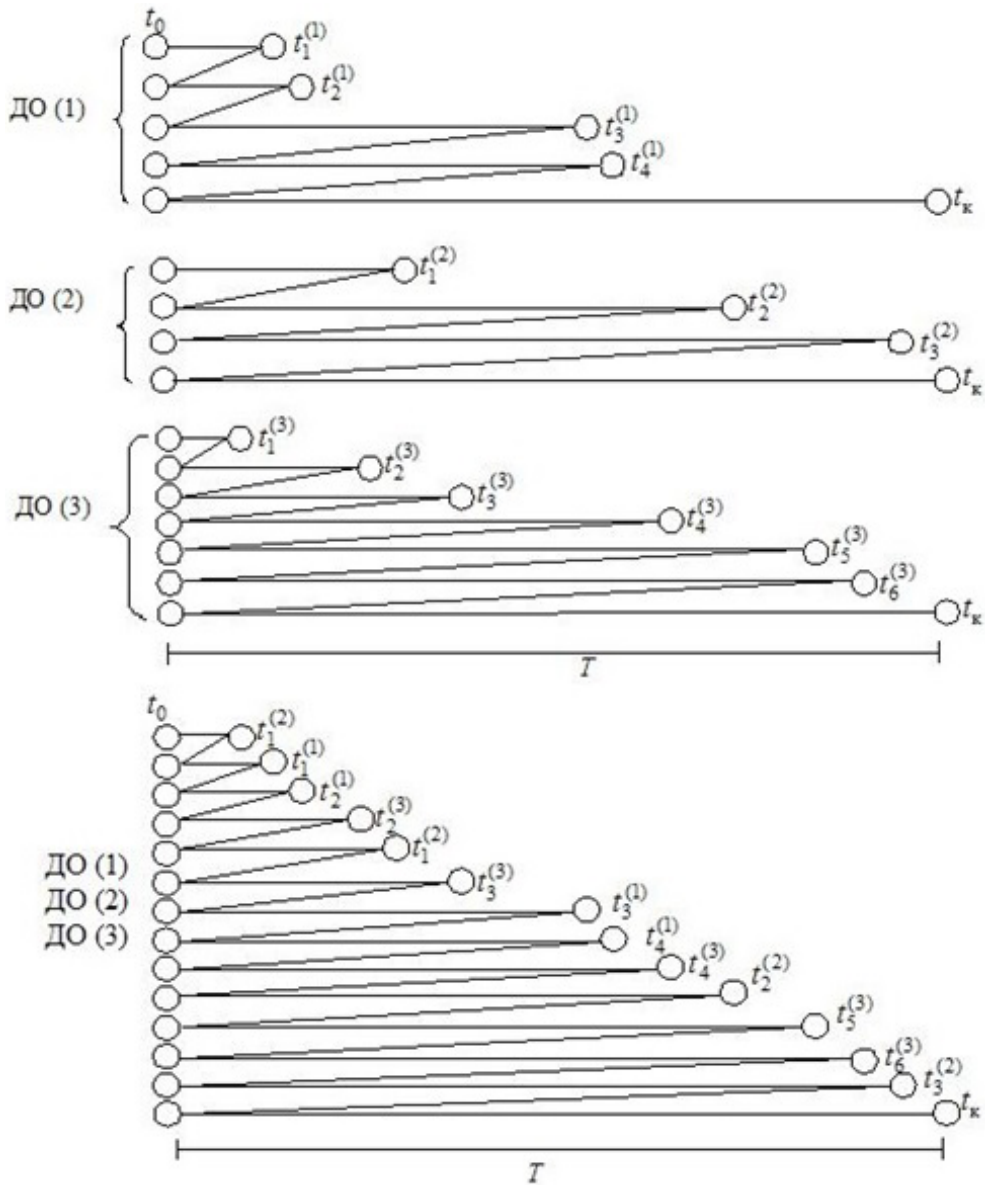


Рис. 4. Временная диаграмма параллельного алгоритма обнаружения дефектов в трех ДО ($\mu_1 = 4$, $\mu_2 = 2$, $\mu_3 = 6$)

Суть обратного алгоритма параллельной процедуры в отличие от параллельной процедуры, опишем на примере:

– тест подаётся одновременно на несколько ДО, в данном случае на три ДО, и после обнаружения первого дефекта в каждом объекте, через время $t_1^{(1)}$, $t_1^{(2)}$, $t_1^{(3)}$, реверсируется в исходное состояние t_0 , рис. 5.

– после устранения обнаруженного дефекта (первого) тест не повторяется снова, как на рис. 4, а запускается в обратном порядке.

На основании рис. 5 для алгоритма, реализующего обратный прогон теста, СВЗ для рассматриваемого примера определяются с помощью выражения:

$$\begin{aligned}
 T_{\text{обрат.}} = & \left[t_{n_1}^{(1)} + (T - t_{n_4}^{(1)}) + (T - t_{n_3}^{(1)}) + (T - t_{n_2}^{(1)}) \right] + \\
 & + \left[t_{n_1}^{(2)} + (T - t_{n_4}^{(2)}) + (T - t_{n_3}^{(2)}) + (T - t_{n_2}^{(2)}) \right] + \\
 & + \left[t_{n_1}^{(3)} + (T - t_{n_6}^{(3)}) + (T - t_{n_5}^{(3)}) + (T - t_{n_4}^{(3)}) + (T - t_{n_3}^{(3)}) + (T - t_{n_2}^{(3)}) \right] + T
 \end{aligned}$$

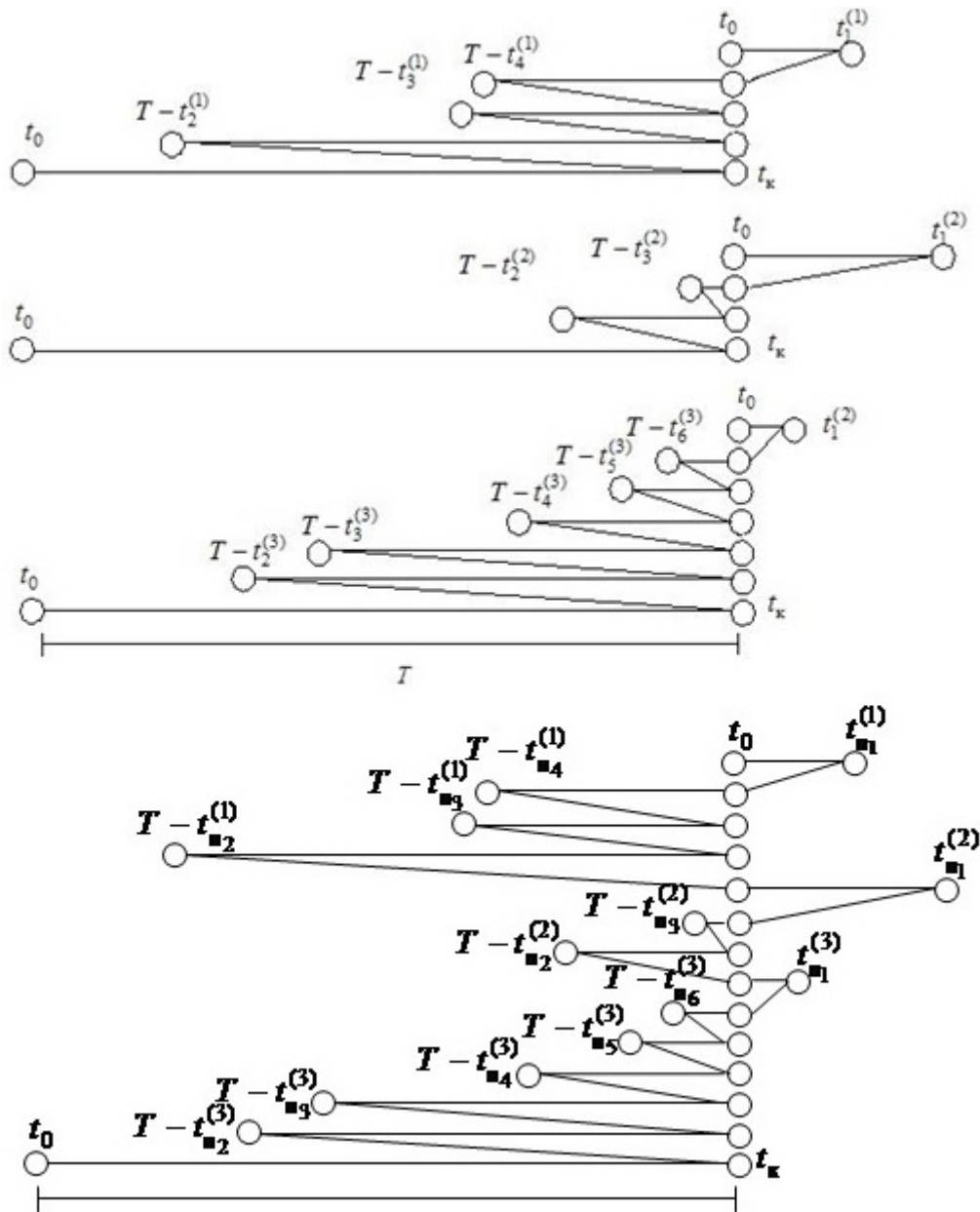


Рис. 5. Временная диаграмма обратного алгоритма обнаружения дефектов в трех ДО ($\mu_1 = 4$, $\mu_2 = 2$, $\mu_3 = 6$)

$T_{\text{обрат.пар}} = [10 + (200 - 95) + (200 - 80) + (200 - 15)] + [30 + (200 - 190) + (200 - 150)] + [8 + (200 - 20) + (200 - 40) + (200 - 100) + (200 - 170) + (200 - 180)] + 200 = 1208$ условных единиц, или в общем виде:

$$T_{\text{обрат.}} = \sum_{j=1}^N t_{\Pi_j}^{(j)} + \sum_{j=1}^N \sum_{i=2}^{\mu_j} (T - t_{\Pi_i}^{(j)}) + T. \quad (6)$$

Соответственно выигрыш во времени от применения обратного алгоритма по сравнению с параллельным:

$$w = 1288/1208 = 1.07$$

или в общем виде:

$$w = \frac{T_{\text{пар.}}}{T_{\text{обр. пар.}}} = \frac{\sum_{j=1}^N \sum_{i=1}^{\mu_j} t_{\pi_i}^j + T}{\sum_{j=1}^N t_{\pi_1}^{(j)} + \sum_{j=1}^N \sum_{i=2}^{\mu_j} (T - t_{\pi_i}^{(j)}) + T}. \quad (7)$$

Анализируя рис. 5 видно, что при обратном алгоритме СВЗ тем меньше, чем больше дефектов обнаруживаются во второй половине теста, то есть чем меньше величина $T - t_{\pi_i}^{(j)}$.

Следовательно, общее число обнаруживаемых дефектов μ можно разбить на два подмножества:

- μ_1 — количество дефектов, обнаруживаемых в первой половине теста;
- μ_2 — количество дефектов, обнаруживаемых во второй половине теста;

Примем среднее время обнаружения всех дефектов подмножества μ_1 $t_{\text{сред}\mu_1} = T/4$, а подмножества μ_2 $t_{\text{сред}\mu_2} = 3T/4$, тогда формулу (7) можно преобразовать

$$w = \frac{T_{\text{пар.}}}{T_{\text{обрат.}}} = \frac{N \left(\mu_1 \frac{T}{4} + \mu_2 \frac{3T}{4} \right) + T}{\sum_{j=1}^N t_{\pi_1}^{(j)} + N \left(\mu_1 \left(T - \frac{T}{4} \right) + \mu_2 \left(T - \frac{3T}{4} \right) \right) + T}. \quad (8)$$

На рис. 6 построен график коэффициента соотношения w между временными затратами параллельной процедуры и временными затратами при обратном алгоритме параллельного алгоритма в зависимости от величины μ_1/μ_2 , то есть соотношения между количеством дефектов обнаруживаемых в первой половине теста и количеством дефектов обнаруживаемых во второй половине теста, при этом длина теста $T = 100$, количество обнаруживаемых дефектов $\mu = 20$, $t_{\pi_1}^{(j)} = T/4$, а $N = 5$ — число ДО.

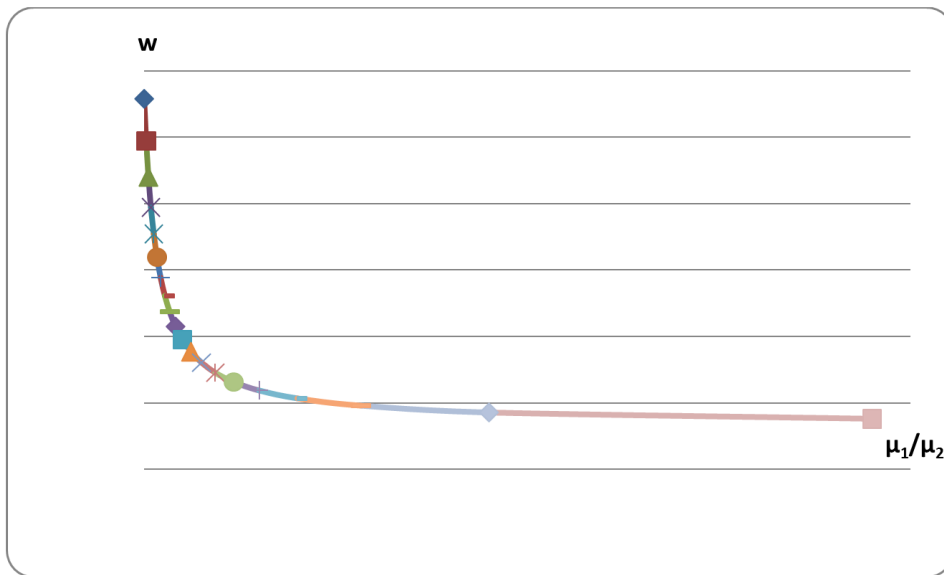


Рис. 6. Изменение коэффициента соотношения w между временными затратами параллельной процедуры диагностирования ДО и временными затратами при обратном алгоритме параллельного алгоритма в зависимости от величины μ_1/μ_2

Анализируя график рис. 6 можно сделать следующие выводы:

Обратный алгоритм параллельной процедуры обнаружения дефектов в дискретных объектах:

- обеспечивает выигрыш во времени по сравнению с параллельной процедурой диагностирования дискретных объектов в тех случаях, когда число дефектов обнаруживаемых во вто-

рой половине теста больше числа дефектов обнаруживаемых в первой половине теста, то есть $\mu_2 > \mu_1$.

- при $\mu_2 = \mu_1$ алгоритмы с точки зрения временных затрат адекватны.
- при $\mu_2 < \mu_1$, то есть, если закон распределения моментов обнаружения дефектов по длине используемого теста таков, что большинство из них обнаруживаются в первой половине теста, то применение обратного алгоритма не эффективно.

Заключение

Предложен и проведен анализ нового алгоритма диагностирования дискретных объектов, названного обратным. А именно обратный алгоритм для последовательной процедуры с реверсом, параллельной процедуры с реверсом. Доказано эффективность использования обратного алгоритма в суммарных временных затратах по сравнению последовательной и по сравнению параллельной процедуры, когда число дефектов обнаруживаемых во второй половине теста больше числа дефектов обнаруживаемых в первой половине теста ($\mu_2 > \mu_1$), то есть если:

$\mu_2 > \mu_1$ — обратный алгоритм имеет выигрыш по сравнению при последовательной процедуре и по сравнению при параллельной.

$\mu_2 = \mu_1$ — обратный алгоритм имеет равный выигрыш по сравнению при последовательной процедуре и по сравнению при параллельной, с точки зрения временных затрат адекватны

$\mu_2 < \mu_1$ — обратный алгоритм не имеет выигрыш по сравнению при последовательной процедуре и по сравнению при параллельной.

Литература

1. *Бородин С. М.* Основы технической диагностики электронных средств : учебное пособие / С. М. Бородин. – Ульяновск : УлГТУ, 2019. – 48 с.
2. *Никифоров С. Н.* Теория параллельного диагностирования. Дискретные объекты. – СПб. ^: СПб. гос. архит.-строит. ун-т, 2009. – 142 с.
3. *Никифоров С. Н.* Параллельное диагностирование объектов методом сравнения с неисправным объектом // Вестник Новгородского госуд. ун-та. – 2005. – № 30. – С. 100–104.
4. *Никифоров С. Н., Тюлюш С. Т.* Эффективность параллельных алгоритмов диагностирования дискретных объектов / С. Н. Никифоров, С. Т. Тюлюш // Вестник гражданских инженеров. – 2011. – № 2 (27). – С. 187–190.
5. *Тюлюш С. Т.* Диагностирование дискретных объектов / С.Т. Тюлюш // Вестник гражданских инженеров. – 2012. – №5 (34). – С. 243–246.

Тюлюш Снежана Танчиковна — старший преподаватель кафедры информатики, информационной безопасности и цифровой экономики КалмГУ,

Горбанева Ольга Ивановна (научный руководитель) — д-р техн. наук, доц., доцент кафедры прикладной математики и программирования ЮФУ.

ИЗ ИСТОРИИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Д. Г. Усков

Воронежский государственный университет

К настоящему времени разработано несколько десятков тысяч языков программирования.

Одними из первых были созданы машинные языки, которые называются языками программирования первого поколения.

Разработка операционных систем, программного обеспечения для экспертных систем и баз данных были бы невозможны при использовании только машинных языков. Следующим шагом по направлению к более современным методам программирования являются языки ассемблера, которые считаются языками второго поколения.

Языки третьего поколения отличаются от языков предшествующих поколений тем, что являются машинно-независимыми.

Русский алгоритмический язык разработал в начале 80-х годов прошлого века для обучения школьников основам информатики и вычислительной техники Андрей Петрович Ершов, академик АН СССР, один из пионеров теоретического и системного программирования, создатель известной во всем мире сибирской школы информатики.

Известные примеры языков третьего поколения: FORTRAN (FORmula TRANslation), разработанный для создания научных и инженерных прикладных программ и COBOL (COmmon Business Oriented Language) — язык программирования для промышленных и правительственных учреждений, созданный военно-морским ведомством США для прикладных программ в бизнесе.

Языки программирования позволяют избежать применения регистров, адресов памяти, машинных циклов в процессе разработки программы и дают возможность сосредоточиться на алгоритме решаемой задачи.

Самым распространенным языком программирования в семидесятых годах прошлого века был язык Бейсик, год рождения которого 1964. Язык был прост и примитивен, содержал всего два десятка команд. Авторы языка Бейсик преподаватели математического факультета Дартмутского колледжа Джон Кемени и Томас Курц. Программно реализовала Бейсик в 1967 году команда студентов колледжа.

Название языка получилось из сокращений: Beginners All-purpose Symbolic Instruction Code — универсальный код символических инструкций для начинающих.

Классик программирования Э. Дейкстра предупреждает: «It is practically impossible to teach good programming to students that have had a prior exposure to BASIC: as potential programmers they are mentally mutilated beyond hope of regeneration.»

Долгое время (до 2010 года) в качестве языка программирования для изучения на первом курсе факультета ПММ ВГУ дисциплины «Информатика и программирование» был язык Паскаль [1, 2].

Паскаль — вычислительный язык программирования высокого уровня создан в 1971 году для обучения программированию как систематической дисциплины. Разработан профессором, директором института информатики Швейцарской высшей политехнической школы в Цюрихе Никлаусом Виртом. Он автор целого ряда книг и учебников по Паскалю, например: «Algorithms + Data Structures = Programs», К. Йенсен, Н. Вирт. Паскаль. Руководство для пользователя и описание языка.

Язык Паскаль получил свое название в честь выдающегося французского ученого (математика и философа). Блез Паскаль (1623–1662 гг.) в 1642 году сконструировал первую счетную машину, которая выполняла всего две операции + и -. Машина была сконструирована в 19-летнем возрасте, чтобы помочь своему отцу, который был сборщиком налогов и выполнял многочисленные и утомительные расчеты.

Д. Ритчи, создатель языка Си, считает «Паскаль — очень элегантный язык. Он по-прежнему жив. Он породил немало последователей и оказал глубокое воздействие на проектирование языков».

До 2010 года на практических занятиях по информатике и программированию использовался задачник-практикум по программированию на языке Паскаль [1]. Он издан в 10 городах России, Украины и Белоруссии: Москва, Санкт-Петербург, Нижний Новгород, Воронеж, Ростов на Дону, Екатеринбург, Самара, Киев, Минск. Эта книга представляет собой не просто задачник с набором примеров и упражнений различной степени сложности. Основные цели этого учебного пособия придать курсу «Информатика и программирование» научно-обоснованный базис, сформировать на его основе определенную культуру проектирования и разработки программ, структурировать соответствующим образом учебный процесс. Авторы, преподаватели кафедры математического обеспечения ЭВМ факультета прикладной математики, информатики и механики ВГУ, большую часть практических заданий посвятили не столько синтаксическим особенностям языка Паскаль, сколько методам программирования, технологиям проектирования алгоритмов и разработки программных систем.

На многих мероприятиях факультета ПММ ВГУ демонстрируется видеофильм о выступлении Никлауса Вирта 5 сентября 2006 года на международной конференции программистов в Нижегородском государственном университете. Фильм сняли студенты ПММ ВГУ Антон Поцюс и Александр Хаустов, которые как победители Всероссийской студенческой олимпиады 2005 года по программированию, присутствовали на выступлении Н. Вирта.

Нельзя не согласиться с высказыванием Н. Вирта:

«Ключ к тайнам компьютеров в гармонии математики, инженерии и программирования».

История языка программирования C++, который на факультете ПММ ВГУ изучается, начиная с первого курса, обширна и продолжительна [3].

Язык программирования C++ разработан на основе языка С Бьёрном Страуструпом (Bjarne Stroustrup) и вышел за пределы его исследовательской группы в начале 80-х годов прошлого века.

На первых этапах разработки (1980 год) язык C++ носил название «Си с классами». В 1983 году Рик Масситти придумал название C++.

Язык программирования C++ является расширением языка С, поэтому программы, написанные на С, могут обрабатываться компилятором языка C++ (но не наоборот).

В 1998 году Международный комитет по стандартизации принял стандарт языка C++ (ISO/IEC 14882 «Standard for the C++ Programming Language»).

Популярность C++ невероятно высока. Существует множество различных платформ и версий C++. Можно выделить наиболее часто используемые:

Microsoft C++ (DOS, Windows),

Borland C++ (DOS, Windows),

Visual C++ (Windows),

GNU CC (Linux).

В связи с этим можно говорить о единстве только ядра C++, в то время, как интерфейс среды программирования и некоторые инструкции различаются для разных платформ и версий. На русский язык переведена книга создателя языка программирования C++ [4].

Первым разработчиком прикладной компьютерной программы для решения дифференциальной краевой задачи второго порядка (1951 г.) был Селим Григорьевич Крейн (совместно с С. А. Авраменко и С. А. Богомолец).

Селим Григорьевич Крейн, математик с мировым именем, заслуженный деятель науки РСФСР успешно работал в 60–70 годах в Воронежском государственном университете заведующим кафедрой уравнений в частных производных, воспитал несколько поколений математиков-профессионалов. Среди его учеников известные профессора и доценты Воронежского университета: Костин В. А., Глушко В. А., Семенов Е. М., Овчинников В. И., Курина Г. А., Зубова С. П., Прозоровская О. И., Шаблицкая Л. Н., Складнев С. А., Гудович И. Н., Гудович Н. Н., Савченко Ю. Б., Руссман И. Б., Ускова О. Ф. [6].

Литература

1. Программирование на языке Паскаль : Задачник / О. Ф. Ускова, М. В. Бакланов, И. Е. Воронина, О. Д. Горбенко, Г. Э. Воцинская, Н. В. Огаркова, В. М. Мельников; Под ред. О. Ф. Усковой. – М. ; СПб. ; Н. Новгород ; Воронеж ; Ростов н/Д. ; Екатеринбург ; Самара ; Киев ; Харьков ; Минск : Питер, 2005 (2002, 2003). – 336 с.

2. Солодова, А. И. «Азбука» для начинающих программистов / А. И. Солодова, О. Ф. Ускова // Современные методы прикладной математики, теории управления и компьютерных технологий (ПМТУКТ-2015) : сборник трудов 8 Международной конференции, Воронеж, 21–26 сентября 2015 г. – Воронеж, 2015. – С. 337–339.

3. Ускова, О. Ф. Информатика и программирование: задачник-практикум по структурному программированию на языке С++ / О. Ф. Ускова, Н. А. Каплиева, О. Д. Горбенко. – Воронеж : Издательский дом ВГУ, 2020. – 279 с.

4. Страуструп, Б. Язык программирования С++ / Б. Страуструп; Пер. с англ. М. Г. Пиголкина, В. А. Яницкого. – М. : Радио и связь, 1991 (2011). – 348 с.

5. Ускова, О. Ф. Российской информатике 70 лет / О. Ф. Ускова, О. Д. Горбенко, Н. А. Каплиева // Актуальные проблемы прикладной математики, информатики и механики [Электронный ресурс] : сборник трудов Международной научной конференции, Воронеж, 17–18 декабря 2018 г. – Воронеж, 2019. – С. 1416–1418.

Усков Даниил Геннадьевич — магистрант 1-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: uskov.dan@mail.ru

РАЗРАБОТКА АЛГОРИТМА ОБРАБОТКИ ОСЦИЛЛОМЕТРИЧЕСКИХ ДАННЫХ КАРДИОЛОГИЧЕСКИХ БОЛЬНЫХ

Д. Г. Усков

Воронежский государственный университет

Введение

При мониторинге состояния больных с сердечно-сосудистыми заболеваниями важно оперативно получать информацию. Основным методом измерения основного показателя состояния — артериального давления. Для этого используется неинвазивный осциллографический метод. Его основой является нахождения пульсации крупной артерии, в основном плечевой, при ее компрессии или декомпрессии. Для этого артерия сначала сжимается до уровня давления, превышающее типичный уровень систолического давления, после чего давление постепенно и равномерно снижается, что позволяет увидеть те самые пульсирующие колебания. Отмеченные колебания называются осциллограммами. На практике, определение осциллограмм осуществляется используя различные методы аппроксимации. В рассмотренном случае для этого использовался медианный оконный фильтр.

1. Разработка алгоритма обработки осциллометрических данных

Работа посвящена разработке прототипа программно-аппаратного комплекса для автоматической диагностики состояния пациентов с заболеваниями сердечно-сосудистой системы и создать алгоритм обработки полученных осциллометрических данных. Для этого разрабатывается аппаратная и программная компоненты прибора. При этом в данной работе основой программного обеспечения является алгоритм, который будет выделять пульсации давления, соответствующие биению сердца из полученных аппаратных данных.

1.1. Аппаратная часть

Разработанный аппаратный комплекс был создан на основе микроконтроллера Arduino Uno Ваниль и состоит из микроконтроллера Ваниль, датчика давления MPX5050 GP, манжета для тонометра B.Well и ручного нагнетателя с механическим регулятором M1/MX.

Для уменьшения цены и увеличения доступности на территории РФ вместо оригинальных моделей была выбрана плата Ваниль – программируемый контроллер, являющимся полным аналогом Arduino Uno и построенный на основе микроконтроллера ATmega328P-PU в корпусе DIP-28 [2].

В измерительную часть был включен аналоговый дифференциальный датчик давления Freescale MPX5050GP. Это легкий и компактный датчик, предназначенный для работы в сухой среде и обладающий удобным уровнем рабочего давления 0–50 кПа, а также открытым входом, совпадающим по диаметру с используемыми в тонометрах резиновыми шлангами.

В механической части были использованы компоненты, взятые с цифрового датчика давления B.Well Swiss AG. Они включают в себя веерообразную однотрубную манжету, соединенную с механическим нагнетателем посредством двух резиновых шлангов, соединенных трехсторонним переходником. Это позволяет подключить измерительную часть прибора, не меняя базовую заводскую конфигурацию. Так как стандартный нагнетатель не удовлетворял

условию наличия регулятора для контролирования уровня давления, то он был заменен на механический нагнетатель модели М1/МХ.

1.2. Разработка алгоритма предварительной обработки осциллометрических данных

Для обработки полученных с аппаратной части данных был разработан специальный механизм фильтрации, который является оконным медианным фильтром. Он обрабатывает полученные с аппаратной части программы значения в движущемся окне длиной в 41 элемент. Для этого созданный фильтр находит среднее арифметическое значение первых двадцати элементов, идущие до среднего, двадцать первого элемента, и последних двадцати элементов для получения двух нормализованных значений. Затем он находит медиану из двух полученных значений и среднего элемента в и вычитает найденную медиану из среднего элемента [4].

Описанный фильтр срабатывает для каждого полученного с аппаратной части комплекса значений, на каждом шаге вычитая предыдущей результат фильтра из нового среднего элемента.

2. Анализ параметров работы алгоритма

Результаты применения разработанного фильтра представлены на рис. 1 и рис. 2.

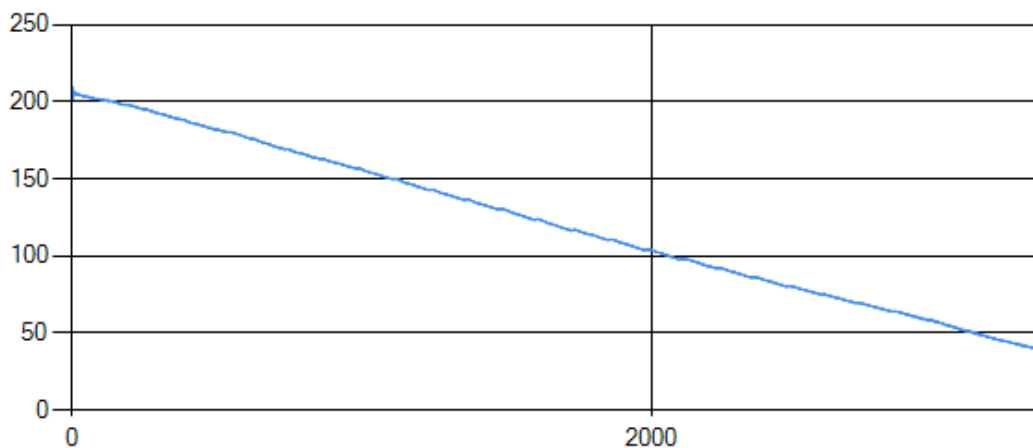


Рис 1. График полученного с аппаратной части давления в манжете

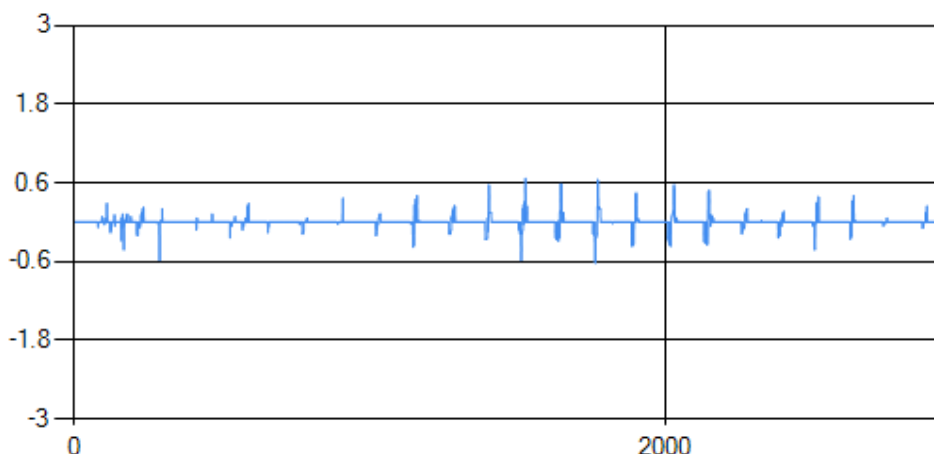


Рис 2. Выделенные через медианный фильтр артериальные пульсации

Результаты показывают (рис. 2) качественное соответствие результатов. Так установлено, что период, в котором выделенные колебания начинаются и заканчиваются соответствует давлению в манжете, достаточному для полного или частичного пережатия артерии.

Заключение

Проведенный анализ показал возможность разработки достаточно простых и доступных устройств для первичной диагностики состояния пациентов с сердечно-сосудистыми заболеваниями. В дальнейшем планируется реализовать дополнить обработку полученных данных для уменьшения ошибок и получения данных по артериальному давлению, а также получению дополнительных показателей состояния сердечно-сосудистой системы.

Литература

1. *Лебедев, В. И.* Функциональный анализ и вычислительная математика / В. И. Лебедев. – Москва : Физматлит, 2000. – 292 с.
2. *Петин, В. Г.* Проекты с использованием микроконтроллера Arduino / В. Петин. – Санкт-Петербург : БВХ-Петербург, 2015. – 401 с.
3. *Абакунов, Н. В.* Методика математического моделирования сердечно-сосудистой системы / Н.В. Абакунов // Математическое моделирование: сборник научных трудов. – 2000. – № 3. – С. 106–117.
4. Andres James, inventor; Siemens Medical Electronics Inc., assignee. Pulse signal extraction apparatus for an automatic blood pressure gauge. United States patent US 5,355,890 1998 October 18.

Усков Даниил Геннадьевич — магистрант 1-го года обучения кафедры математического обеспечения ЭВМ Воронежского государственного университета. E-mail: uskov.dan@mail.ru

Абрамов Геннадий Владимирович (научный руководитель) — д-р техн. наук, проф., заведующий кафедрой математического обеспечения ЭВМ Воронежского государственного университета.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РАЗВЛЕКАТЕЛЬНОЙ КОМПЬЮТЕРНОЙ ИГРЫ ДЛЯ ШКОЛЬНИКОВ

У. В. Чеботарев

Воронежский государственный университет

Введение

Разработка программного обеспечения (ПО) для мобильных устройств актуальна из-за распространённости смартфонов. Почти у каждого человека в наше время есть телефон с сенсорным экраном или планшетный компьютер. Кроме того, рассматриваемая тема разработки видеоигр наибольшим спросом пользуется на мобильных устройствах.

Цель создаваемого программного обеспечения: разработать одно развлекательное приложение, ориентированное на школьников, в течение недели, применить на практике знания, полученные из курсов программирования и ООП на факультете ПММ ВГУ, и получить опыт работы в межплатформенной среде разработки UnityEngine.

1. Общее описание программного обеспечения

Сценарий игры: держать персонажа в пределах видимости экрана, перепрыгивая с одной движущейся вверх платформы на другую так, чтобы не «уехать вверх» и не упасть с платформ вниз, за пределы экрана, попутно собирая «монетки». За каждые собранные 25 «монеток» голова действующего персонажа украшается очередной короной. Само ПО получило название «Go up Horned» (авторский перевод: «поднимайся, Рогатый»).

Движущиеся вверх платформы и «монетки» специально появляются, используя равномерный закон распределения (алгоритм появления и сбора монеток будет описан далее), так, чтобы не было возможность просчитать наиболее выгодную стратегию заранее. Таким образом, успешность в игре зависит только от того, как быстро пользователь анализирует происходящее на экране и как реагирует на разные игровые события, которые начинают усложнять игровой процесс после получения определённого количества «монеток».

2. Выбор среды разработки

Первым шагом в создании ПО был выбор среды разработки, подходящей именно для реализации разрабатываемого приложения. Из нескольких вариантов был выбран Unity [5].

Достоинства межплатформенной среды разработки Unity Gaming Services перечислены ниже:

1. Кроссплатформенность. В современном мире принципиально важно иметь возможность обеспечить работу своего ПО на нескольких аппаратных платформах и операционными системами, а именно Windows, Android и IOS.

2. Unity — единая платформа с гибкими решениями для разработки и развития игр, а также управления ими.

3. Все доходы и авторские права принадлежат самому разработчику, а не среде разработки.

4. Платформа бесплатна и полностью доступна для физического лица и для студии с годовым оборотом до 100.000\$.

5. Множество доступных интернет ресурсов, ориентированных на помощь разработчику.

6. Движок использует для написания скриптов язык программирования C#, который относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++, который известен из курсов программирования на факультете ПММ ВГУ.

3. Общая информация о создаваемом программном обеспечении

Всё приложение состоит из двух сцен. Первая — непосредственно игровое поле, на котором происходят все действия, вторая — главное меню, содержащее пользовательский интерфейс с кнопками: «Начать игру», «Выйти из приложения» и «Посмотреть рекламу». Последняя необходима, чтобы дать возможность пользователям поддержать разработчика, если они сами того пожелают, не пропускаемой рекламы «Go up Horned» не содержит. Переход между сценами осуществляется кнопками: «Выйти в главное меню», «Начать игру» или смертью игрового персонажа.

Пользователь взаимодействует с игрой через своего единственного персонажа, который описан с помощью следующих объектов:

Встроенного класса `Rigidbody 2D` из `UnityEngine`, который позволяет назначить объекту физические свойства. Так игровой персонаж получил массу «`Mass`», и силу тяжести «`Gravity Scale`», чтоб он мог падать, прыгать и взаимодействовать с платформами. Кроме того, я убрал возможность изменять угол наклона относительно осей X и Z во избежание переворота героя и платформ.

Встроенного класса `Box Collider 2D` из `UnityEngine`, который используется для проверки соприкосновений с триггерами.

Четырёх скриптов: два из них отвечают за передвижение персонажа [4], один за сбор «монеток» и последний за поражение.

Из других игровых объектов выделяю лишь следующие три.

Джойстик — инструмент управления персонажем, передвигая его горизонтально, задаётся движение, а вертикально вверх — прыжок.

Платформа — простой физический объект, появляющийся каждый отрезок времени чуть ниже поля видимости игрового поля, используя равномерный закон распределения случайных величин для задания координаты X , движущийся строго вверх с постоянной скоростью. Персонаж может толкать платформу слева или справа, так же, под его же массой, она замедляет свой подъём, кроме того, если взаимодействовать с платформой снизу, под углом от n до $2n$, — она будет игнорировать свою твердость, точнее, персонаж сможет проходить сквозь неё. Кроме того, достигая определённой высоты, платформа будет уничтожаться, сталкиваясь со специальным игровым объектом, похожим по реализации на «Области смерти», описанный ниже.

Области смерти — игровые объекты, которые не имеют физических свойств, но имеют коллайдер, при соприкосновении с которым игрок погибает, и вызывается игровое событие — поражение (скрипт, обнуляющий текущий результат и возвращающий пользователя в главное меню).

4. Описание алгоритма взаимодействия с «монеткой»

За сбор «монеток» отвечают 3 класса: `ScoreScript`, `HighScoreScript` и `CoinCollection`, все они являются наследниками класса `MonoBehaviour` из `UnityEngine`. Основным является `CoinCollection`, он отвечает непосредственно за сбор «монетки» и направляет информацию об этом двум другим классам. Алгоритм сбора «монеток» можно рассмотреть на блок-системе (рис. 1).

`CoinCollection` реализован 12 полями:

Высота и ширина игрового поля, минимальное расстояние появления монетки от игрока, необходимое расстояние от центра игрока до места появления новой короны, счетчик, буле-

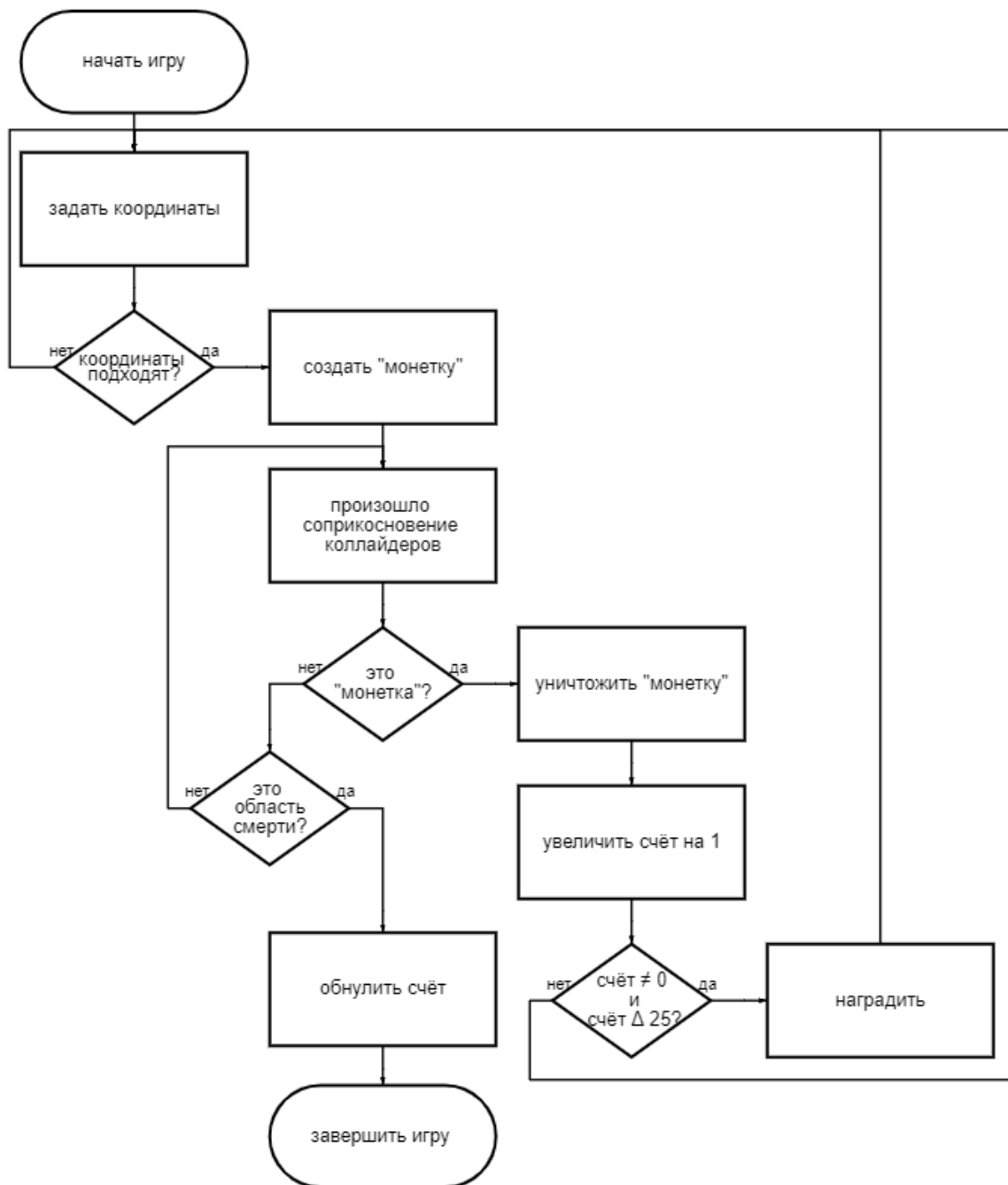


Рис. 1

вая переменная, указывающая, существует ли монетка на поле или нет, и поля для самих классов, используемых игровых объектов, например, сама «монетка».

4 методами:

`void spawnCoin` — метод, проверяющий, находится ли «монетка» на игровом поле, если нет, то задаёт две переменных — координату появления новой «монетки», используя значения заданного размера игрового поля и метод класса `Random.Range` из `UnityEngine`, для генерации случайных значений по равномерному закону распределения, до тех пор пока расстояние от игрока до предполагаемого места появления «монетки» было больше минимального, после чего, методом класса `GameObject` из `UnityEngine`, `Instantiate` создаётся «монетка» с этими координатами.

`void Start` — срабатывает сразу при запуске игровой сцены и вызывает метод `spawnCoin`.

`void OnTriggerEnter2D` — проверяет, соприкоснулись ли коллайдеры игрока с другими коллайдерами. Если персонаж «дотронулся» до монетки, то логическая переменная существования «монетки» на игровом поле принимает значение «ложь», монетка уничтожается методом `Destroy` класса `GameObject` из `UnityEngine`, а статическое поле класса `ScoreScript`, отвечающее за счетчик «монеток» увеличивается на единицу, а если количество собранных становится не равным нулю и нулю равен остаток от деления счета на необходимое количество «монеток» для награждения, то сдвигается позиция появления короны на голове игрока, и там методом `ToReward`, описанном далее, появится корона. Если же игрок коснулся области смерти — счет обнуляется.

`void ToReward` — просчитывает позицию появления короны, относительно положения игрока и, используя метод `Instantiate` создаёт корону на данной позиции, делая её наследником игрового персонажа, чтоб при его перемещении корона передвигалась вместе с ним.

Классы `ScoreScript`, `HighScoreScript`. Оба класса являются счетчиками «монеток» и в текстовом виде показывают счет пользователю. Их разница только в типе хранящихся данных, если `ScoreScript` считает фактическое количество и удаляет его после смерти персонажа, то `HighScoreScript` запоминает данные с наилучшим результатом и хранит их даже после завершения работы всей программы.

Всего для реализации «Go up Horned» потребовалось несколько десятков скриптов, 6 анимаций, множество изображений, использовался `UnityEngine.UI` для работы с пользовательским интерфейсом и других внутренних инструментов `Unity`.

Заключение

Основные игровые механики, были реализованы в течение недели, но жизнь проекта на этом не закончилась, в течение месяца продолжалась доработка разрешения экрана [3], текстур, системы управления, пользовательского интерфейса, вводились особые платформы и возможности. На данный момент ведется работа над включением звуков взаимодействия и фоновой музыки.

Выбор межплатформенной среды разработки `Unity Gaming Services` себя оправдал, внутренние инструменты среды разработки оказались удобными, а основные из них даже интуитивно понятными. Данная платформа сильно упростила задачу с разработкой развлекательного приложения в течение недели, а синтаксическая схожесть, используемого для написания скриптов в `UnityEngine C#`, с изучаемым на курсах программирования и ООП на факультете ПММ ВГУ языком программирования `C++` позволило на практике применить полученные знания [1].

Литература

1. Ускова О. Ф., Каплиева Н. А., Горбенко О. Д. Информатика и программирование. Задачник-практикум на языке программирования `C++`. – Воронеж : Издательский дом ВГУ, 2020. – 379 с.
2. Болотова С. Ю., Махортов С. Д. Программирование для мобильных устройств. Учебное пособие. – Воронеж : Издательско-полиграфический центр ВГУ, 2013. – 103 с.
3. Ориентация экрана на мобильных устройствах – Режим доступа: <https://www.youtube.com/c/EmeraldPowder>. – (Дата обращения: 18.03.2022).
4. 2D контроллер движения персонажа в `Unity` – Режим доступа: <https://www.youtube.com/channel/UChr72IH-33mrD2h09MR3N3Q/videos>. – (Дата обращения: 10.03.2022).

5. Unity User Manual 2021.3 (LTS) – Режим доступа: <https://docs.unity3d.com/Manual/index.html>. – (Дата обращения: 12.03.2022).

Чеботарев Устин Владимирович — студент 2-го курса Воронежского государственного университета. E-mail: ustin_ch@list.ru

Ускова Ольга Федоровна (научный руководитель) – канд. техн. наук, доц., профессор кафедры математического обеспечения ЭВМ Воронежского государственного университета.

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ ЛИНЕЙНОЙ ДИНАМИЧЕСКОЙ СИСТЕМОЙ

Д. Г. Черненко

Воронежский государственный университет

Введение

Работа посвящена компьютерной реализации алгоритма нахождения оптимального управления линейной динамической системой. В принципе, идея такого алгоритма известна давно. Например, она описана в книге Первозванского [3] 1986 г. Но изложение в [3] рассчитано на ручной счет, поэтому некоторые детали не доведены до уровня компьютерного алгоритма. Например, на некотором этапе необходимо решать переопределенную систему линейных уравнений, про которую известно, что она имеет единственное решение, но неизвестно, какие уравнения можно отбросить. Пока уравнений мало, систему можно решать, контролируя процесс глазами. Когда уравнений становится много, все должно выполняться автоматически.

Линейной динамической системой называют [3, 4, 6] объект, который описывается уравнениями

$$\begin{aligned}x'(t) &= Ax(t) + bu(t), \\x(0) &= x_0,\end{aligned}$$

где $A \in \mathbb{K}^{N \times N}$ — квадратная матрица, $b \in \mathbb{K}^N$ — вектор-столбец, x принимает значения в \mathbb{K}^N , а u — скалярная функция. Функцию u называют *управлением*. Ее можно менять (задавать произвольным образом). Задача оптимального управления заключается в том, чтобы сформировать решение x , обладающее необходимыми свойствами. В работе рассматривается случай, когда желаемые свойства решение на математическом уровне заключаются в том, что величина

$$\Theta = \int_0^{+\infty} x^H(t) Q x(t) dt + \int_0^{+\infty} u^H(t) R_0 u(t) dt$$

должна быть минимальна. Смысл первого интеграла заключается в минимизации отклонения x от нуля. Вторым интегралом описывают затраты на реализацию выбранного управления. Конечно, идеальными были бы выход и управление, тождественно равные нулю, но взять такие нельзя из-за наличия ненулевого начального условия.

Из общей теории [3, 5, 9] известно, что оптимальное управление имеет вид

$$u(t) = -K e^{(A-BK)t} x_0,$$

где матрица K находится как решение *матричного уравнения Риккати*

$$A^H X + XA - XBR_0^{-1} B^H X + Q = 0.$$

При этом решение x имеет вид

$$x(t) = e^{(A-BK)t} x_0.$$

Таким образом, с математической точки зрения задача сводится к решению уравнения Риккати. Его решению и посвящена основная часть работы.

1. Понятие о динамической системе

В данной работе поле скаляров обозначается символом \mathbb{K} . Под \mathbb{K} может пониматься как поле действительных чисел \mathbb{R} , так и для поле комплексных чисел \mathbb{C} . Символом $\mathbb{K}^{n \times m}$ мы обо-

значаем множество всех матриц A размера $n \times m$. Символом A^H мы обозначаем сопряженно транспонированную по отношению к матрице A . По умолчанию все векторы записываются в виде столбцов.

Предположим, что $A \in \mathbb{K}^{N \times N}$, а $b \in \mathbb{K}^N$. Рассмотрим задачу

$$\begin{aligned} x'(t) &= Ax(t) + bu(t), \\ x(0) &= x_0, \end{aligned} \quad (1)$$

где x принимает значения в \mathbb{K}^N , u — скалярная функция, а x_0 — некоторое начальное условие. Объект, изменение которого во времени описывается уравнениями (1), называют *непрерывной динамической системой*.

Прикладная интерпретация поставленной задачи (1) заключается в том, что дифференциальное уравнение $x'(t) = Ax(t) + f(t)$ описывает систему, состоящую из множества объектов — отдельные координаты пространства \mathbb{K}^N соответствуют различным параметрам этих объектов. *Управление системой* происходит посредством задания функции u , называемой *входом* или *входным воздействием*. Оно приводит к изменению всего лишь одного из параметров. Ему соответствует вектор b , т. е. управляющий сигнал u изменяет лишь этот параметр.

Если вместо одного входа имеется два, то в задаче (1) свободный член $bu(t)$ необходимо заменить на сумму двух: $b_1u_1(t) + b_2u_2(t)$. Аналогично в случае большого числа входов.

2. Связь функционала качества и уравнения Риккати

(Алгебраическим или матричным) уравнением второго порядка называют матричное уравнение вида

$$BX + XA + XRX + Q = 0. \quad (2)$$

Предполагается, что в уравнении (2) $A \in \mathbb{K}^{m \times m}$, $B \in \mathbb{K}^{n \times n}$, $R \in \mathbb{K}^{m \times n}$, $Q \in \mathbb{K}^{n \times m}$ — известные матрицы, а $X \in \mathbb{K}^{n \times m}$ — неизвестная матрица.

Уравнением Риккати называют матричное уравнение вида

$$A^H X + XA + XRX + Q = 0, \quad (3)$$

в котором все матрицы имеют размер $N \times N$. Также предполагается, что заданные матрицы R и Q являются эрмитовыми¹. Уравнение Риккати является частным случаем уравнения (2) и возникает в следующем контексте.

Рассмотрим динамическую систему, которая преобразует входной сигнал u (называемый управлением) в выходной сигнал x (называемый фазовым состоянием) по правилу

$$\begin{aligned} x'(t) &= Ax(t) + Bu(t), \quad t \geq 0, \\ x(0) &= x_0, \end{aligned} \quad (4)$$

где u и x — вектор-функции, зачастую разного размера ($u(t) \in \mathbb{K}^q$, $x(t) \in \mathbb{K}^N$), матрицы A и B имеют соответствующие размеры.

Систему (4) или короче пару $(A \ B)$ называют *стабилизируемой*, если существует матрица K , такая что матрица $A - BK$ устойчива². Матрица K имеет смысл коэффициента усиления в обратной связи

$$u(t) = -Kx(t).$$

Будем считать, что некоторые (энергетические, материальные) затраты на исполнение управления u задаются функционалом качества

¹Матрицу $A \in \mathbb{K}^{n \times n}$ называют *эрмитовой*, если $A^H = A$

²Матрицу $A \in \mathbb{K}^{n \times n}$ называют *устойчивой*, если ее спектр содержится в открытой левой полуплоскости $\{\lambda \in \mathbb{C} : \operatorname{Re} \lambda < 0\}$

$$\Theta = \Theta(x, u) = \int_0^{+\infty} x^H(t) Q x(t) dt + \int_0^{+\infty} u^H(t) R_0 u(t) dt. \quad (5)$$

Здесь положительно определенные матрицы R_0 и Q описывают трату различных ресурсов, используемых при осуществлении управления объектом. Например, с их помощью можно отразить в модели эффективность использования топлива для работы двигателей самолета при нужной мощности.

Требуется найти такое управление u , для которого значение функционала Θ (при заданном начальном значении x_0) минимально. Именно в этом нам поможет уравнение Риккати.

Теорема 1 ([3, с. 287], [1, теорема 9.35]). *Предположим, что пара (A, B) стабилизируема, а матрица R положительно определена. Пусть уравнение Риккати*

$$A^H X + XA - XBR_0^{-1}B^H X + Q = 0$$

имеет положительно определенное решение X (очевидно, матрице R из (3) в рассматриваемом случае соответствует $R = -BR_0^{-1}B^H$). Тогда для любых начальных значений x_0 минимизирующая функционал качества (5) стабилизация обеспечивается обратной связью

$$u(t) = -Kx(t), \quad (6)$$

где K — коэффициент усиления

$$K = R_0^{-1}B^H X. \quad (7)$$

(Подчеркнем, что для нахождения коэффициента K по формуле (7) нам заранее известны все параметры, кроме матрицы X .)

При этом величина минимальных затрат на осуществления управления находится по формуле

$$\Theta_{\min} = x_0^H X x_0,$$

а управление u и фазовое состояние x меняются по следующим правилам соответственно

$$x(t) = e^{(A-BK)t} x_0, \quad u(t) = -K e^{(A-BK)t} x_0.$$

(Появление в этих формулах матрицы $A-BK = A-BR_0^{-1}B^H X = A+RX$ объясняется подстановкой (6) в уравнение (4).)

Выполнение условия стабилизируемости, используемого в теореме 1, может быть проверено с помощью приводимой ниже теоремы 2.

Теорема 2. *Для того чтобы пара (A, B) была стабилизируема, достаточно, чтобы ранг блочной матрицы $(B \ AB \ \dots \ A^{N-1}B)$ был равен N .*

Сопоставим уравнению (3) вспомогательную блочную матрицу

$$H = \begin{pmatrix} A & R \\ -Q & -A^H \end{pmatrix}. \quad (8)$$

Нам понадобится еще одно условие, связанное со стабилизируемостью. Не вдаваясь в обсуждение, скажем, что решение X уравнения Риккати (3) называют стабилизирующим, если матрица $A+RX$ устойчива.

Теорема 3 ([9, с. 327]). *Допустим, что матрица R неотрицательно или неположительно определена. В таком случае для существования у уравнения Риккати (3) стабилизирующего решения X необходимо и достаточно, чтобы пара (A, R) была стабилизируема, а у матрицы (8) не было точек спектра на мнимой оси. Если при этом матрица Q положительно определена, то стабилизирующее решение X также положительно определено.*

Отметим, что условия теоремы 3 для случайных матриц выполняется с вероятностью 1.

3. Алгоритм решения уравнения Риккати

Опишем алгоритм нахождения положительно определенного стабилизирующего решения уравнения Риккати (3). Будем полагать, что условия теоремы 3 выполнены.

Предположим, что X — положительно определенное стабилизирующее решение уравнения (3). Умножением проверяется, что выполнение уравнения (3) равносильно равенству

$$\begin{pmatrix} I & 0 \\ -X & I \end{pmatrix} \begin{pmatrix} A+RX & -R \\ 0 & -(A+RX)^H \end{pmatrix} \begin{pmatrix} I & 0 \\ -X & I \end{pmatrix}^{-1} = \begin{pmatrix} A & R \\ -Q & -A^H \end{pmatrix}. \quad (9)$$

Поскольку матрица, содержащая блоки $A+RX$ и $-(A+RX)^H$, — треугольная, её спектр равен объединению спектров диагональных блоков. Поскольку решение X стабилизирующее, спектры $\sigma(A+RX)$ и $\sigma(-(A+RX)^H)$ лежат в левой и правой полуплоскостях соответственно. Следовательно³, функция sgn к ней применима.

Применим к левой и правой частям (9) функцию sgn . Одновременно введем сокращенное обозначение

$$\operatorname{sgn} \begin{pmatrix} A+RX & -R \\ 0 & -(A+RX)^H \end{pmatrix} = \begin{pmatrix} -I & Z \\ 0 & I \end{pmatrix},$$

где Z — неизвестная матрица (зависящая от X). Поэтому, применив функцию sgn к (9), получим

$$\begin{pmatrix} I & 0 \\ -X & I \end{pmatrix} \begin{pmatrix} -I & Z \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ -X & I \end{pmatrix}^{-1} = \operatorname{sgn} \begin{pmatrix} A & R \\ -Q & -A^H \end{pmatrix}.$$

Обозначим стоящую справа матрицу через W (она эффективно вычисляется):

$$W = \operatorname{sgn} \begin{pmatrix} A & R \\ -Q & -A^H \end{pmatrix}.$$

Перепишем последнее равенство в виде

$$\begin{pmatrix} I & 0 \\ -X & I \end{pmatrix} \begin{pmatrix} -I & Z \\ 0 & I \end{pmatrix} \begin{pmatrix} I & 0 \\ -X & I \end{pmatrix}^{-1} = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}.$$

Затем перепишем его в виде

$$\begin{pmatrix} I & 0 \\ -X & I \end{pmatrix} \begin{pmatrix} -I & Z \\ 0 & I \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix} \begin{pmatrix} I & 0 \\ -X & I \end{pmatrix}.$$

Выполнив умножение, получим

$$\begin{pmatrix} -I & Z \\ X & I - XZ \end{pmatrix} = \begin{pmatrix} W_{11} - W_{12}X & W_{12} \\ W_{21} - W_{22}X & W_{22} \end{pmatrix},$$

что эквивалентно системе уравнений

$$\begin{cases} W_{11} - W_{12}X = -I, \\ W_{21} - W_{22}X = X, \\ W_{12} = Z, \\ W_{22} = I - XZ. \end{cases}$$

Чтобы исключить из этих уравнений Z , выразим Z из третьего уравнения и подставим его в четвертое. Одновременно применим к четвертому уравнению операцию комплексного сопряжения (при этом учтем то, что X положительно определена и поэтому является эрмитовой), чтобы сделать все уравнения линейными по X . В результате придем к системе уравнений

³ sgn — аналитическая функция, принимающая значение $+1$ в правой комплексной полуплоскости и значение -1 в левой

$$\begin{cases} W_{12}X & = I + W_{11}, \\ (I + W_{22})X & = W_{21}, \\ W_{12}^H X & = I - W_{22}^H. \end{cases} \quad (10)$$

В этой системе количество неизвестных меньше количества уравнений. О том, что решение существует и единственно мы знаем. Тем не менее, система является переопределенной. Выбор той или иной линейно независимой подсистемы уравнений может приводить к разной обусловленности и, как следствие, к разной величине ошибок округления. Чтобы обойти эту проблему, мы решаем систему (10) методом псевдообращения, который учитывает все уравнения и минимизирует ошибки округления. Выбор данного метода для решения системы (10) можно считать небольшим научным продвижением в исследовании рассматриваемой задачи.

Далее остается заметить, что после того, как мы найдем решение X , мы можем найти коэффициент усиления (7) и определить управление (6), минимизирующее функционал качества (5).

На основе описанного алгоритма была написана компьютерная программа на языке Wolfram Mathematica. Она тестировалась на случайных исходных данных. На рис. 1 приведен один из результатов работы этой программы.

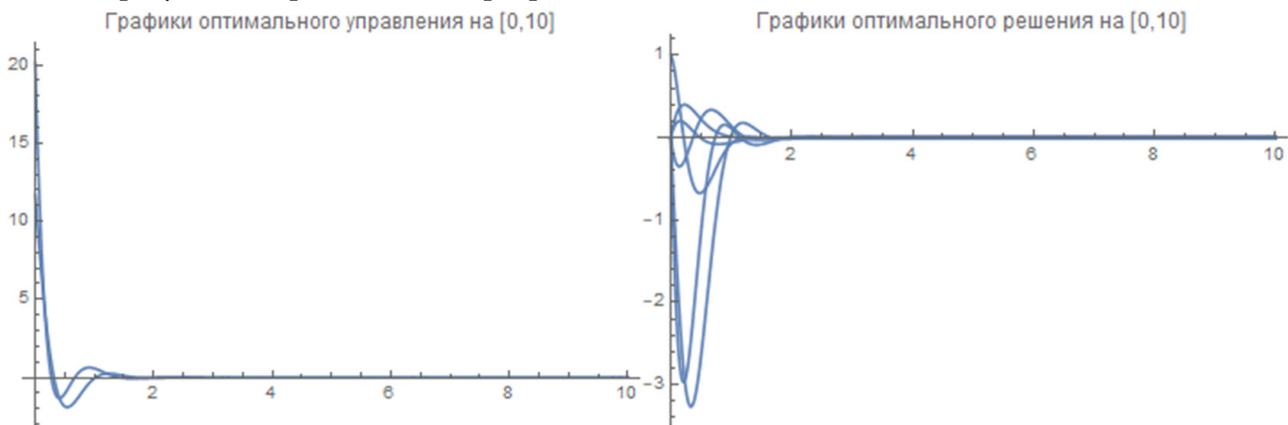


Рис. 1. Графики оптимального управления и оптимального решения

Можно заметить, что найденное решение действительно стабилизирует систему. Хорошо видно, что спустя некоторое время значения каждого из графиков решений становятся практически нулевыми. Из графиков можно сделать выводы о том, как много времени требуется для того, чтобы стабилизировать систему, какие конкретные подсистемы были изменены и каким образом. Также можно наблюдать, как происходит управление и с какой интенсивностью оно воздействует на систему.

4. Проверка решения на оптимальность

Чтобы убедиться в том, что получаемое с помощью алгоритма и компьютерной программы решение действительно является оптимальным, проводилась специальная проверка. Суть этой проверки заключается в сравнении полученного оптимального управления с другими вариантами управления, отличающимися другим выбором X . Если в ходе проверки будет обнаружено X , при котором затраты на осуществление соответствующего ему управления будут меньше, чем затраты Θ на осуществление управления, соответствующего оригинальному решению, то новое решение окажется оптимальнее и выгоднее, чем найденное после применения алгоритма, и, следовательно, найденное решение не является оптимальным.

Итак, для проверки необходимо получить некоторое решение, отличное от оригинального, причем из соображений надежности проверки желательно выбрать несколько таких решений.

Разумеется, все решения, стабилизирующие систему, перебрать и сравнить с исходным не представляется возможным ввиду их огромного количества.

Предлагается построить несколько отличных от оригинального решений, отличающихся от него на малую величину обратной связи X . Если все такие решения будут менее эффективными, то это можно считать подтверждением того, что исходное решение является оптимальным.

На рис. 2 приведен один из результатов проверки оптимальности полученного управления с помощью построения трех дополнительных вариантов управления, немного отличающихся от первоначального.

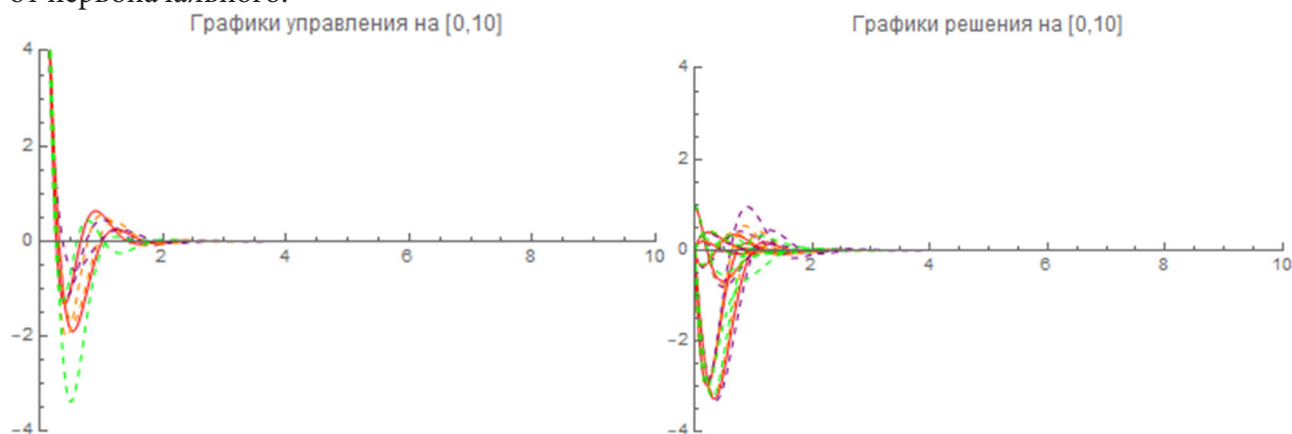


Рис. 2. Графики возмущенного управления и возмущенного решения

Литература

1. Курбатов, В. Г. Вычислительные методы спектральной теории : учебное пособие / В. Г. Курбатов, И. В. Курбатова. – Воронеж : Издательский дом ВГУ, 2019. – 323 с.
2. Курбатов, В. Г. Пакет «Математика» в прикладных научных исследованиях : учебное пособие / В. Г. Курбатов, В. Е. Чернов. – Воронеж : Издательский дом ВГУ, 2016. – 240 с.
3. Первозванский, А. А. Курс теории автоматического управления / А. А. Первозванский. – М. : Наука, 1986. – 615 с.
4. Antoulas, A. C. Approximation of large-scale dynamical systems / A. C. Antoulas. – Philadelphia, PA : Society for Industrial and Applied Mathematics (SIAM), 2005. – Vol. 6 of Advances in Design and Control. – xxvi+479 p.
5. Gohberg, I. Indefinite linear algebra and applications / I. Gohberg, P. Lancaster, L. Rodman. – Basel : Birkhäuser Verlag, 2005. – xii+357 p.
6. Polderman, J. W. Introduction to mathematical systems theory. A behavioral approach / J. W. Polderman, J. C. Willems. – New York : Springer-Verlag, 1998. – Vol. 26 of Texts in Applied Mathematics. – xxx+424 p.
7. Wolfram, S. The Mathematica book / S. Wolfram. – Fifth Edition. – New York : Wolfram Media, 2003. – 1488 p.
8. Wolfram, S. An elementary introduction to the Wolfram Language / S. Wolfram. – Second edition. – Champaign, IL : Wolfram Media, 2017. – xv+339 p.
9. Zhou, K. Robust and optimal control / K. Zhou, J. C. Doyle, K. Glover. – New Jersey : Prentice Hall, 1996. – Vol. 40. – 586 p.

Черненко Дмитрий Геннадьевич — студент 4-го курса кафедры системного анализа и управления Воронежского государственного университета.
E-mail: dmitrij.chernenko2015@yandex.ru

Курбатов Виталий Геннадьевич (научный руководитель) — д-р физ.-мат. наук, проф., профессор кафедры системного анализа и управления Воронежского государственного университета. E-mail: kv51@inbox.ru

ТРЕХМЕРНАЯ РЕКОНСТРУКЦИЯ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

Д. Е. Черняев, В. А. Еномян, Д. А. Морозов,
А. В. Шатковская, И. Ю. Новоскольцев, В. Г. Петрова

Воронежский государственный университет

Введение

Обработка изображений является неотъемлемой частью множества задач, связанных с медициной, в частности, с анализом медицинских изображений. Качество изображения влияет на визуализацию той области на изображении, которая является основой для постановки медицинского диагноза или планирования для предстоящей операции [2].

Одним из наиболее широко используемых методов диагностики является магнитно-резонансная томография (МРТ). МРТ представляет собой визуализацию некоторой области в трехмерном пространстве. Квалифицированный специалист должен уметь выявлять аномалии строения тела без хирургического вмешательства путем просмотра отдельных изображений. Реконструкция модели данной области с возможностью визуальных преобразований делает анализ более эффективным. Воссозданные из реальных томограмм 3D модели могут быть также использованы для лечения радио- и адронной терапией. Использование таких моделей позволяет получить более достоверные результаты с учетом особенностей пациента, поэтому должно помочь разработать более четкие планы лечения. Кроме того, технологии дополненной реальности позволяют связать предоперационные данные с текущим состоянием организма, либо использовать их во время операций.

В данной статье рассмотрены обработка изображений формата DICOM, а также различные методики реконструкции 3D моделей из изображений данного формата.

1. Медицинские изображения формата DICOM

Результаты томографии хранятся в файле формата DICOM. Формат DICOM (Digital Imaging and Communications in Medicine) является универсальным форматом для цифровых изображений в медицине, который поддерживаем различной медицинской аппаратурой для клинических диагностических мероприятий и симуляций.

Метаданные пациентов и другая медицинская информация хранится в DICOM файлах как объекты с присвоенными характеристиками [1]. Процесс получения DICOM файла представлен на рис. 1.



Рис. 1. Процесс получения DICOM

Объекты и параметры DICOM файла должны быть определены в соответствии с форматом информационных объектов (IOD). Так, IOD пациента обычно описывает его ФИО, номер медицинского полиса (или его ID), пол, возраст и т. д.

Последовательность данных DICOM состоит из отдельных элементов, в которых передаются значения атрибутов IOD (имя пациента, возраст, название учреждения, тип изображения и т. п.). Каждый элемент данных имеет следующие поля:

• *Тег* — уникальный идентификатор элемента, состоящий из пары 16-битных слов, определяющих номер группы и номер элемента. Пользователь может вводить свои собственные теги, передавая их для согласования соответствующему сервису.

• *Поле типа данных (VR)* — 2-х символьная строка, содержащая аббревиатуру типа данных. Наряду с классическими типами (целыми, вещественными, строковыми и текстовыми), вводятся специфические типы для времени, возраста, имени, уникальных идентификаторов и т. д. Определенный тип элементов может содержать в своем поле данных другие элементы. Поле типа является необязательным, и при его отсутствии тип данных определяется по тегу.

• *Поле длины* — в зависимости от типа это 16- или 32-битное беззнаковое слово, содержащее число байт в поле данных;

• *Поле данных* — передаваемые атрибуты IOD.

Для декодирования DICOM файла и извлечения из него изображения и метаданных существует различное программное обеспечение (ПО), так называемые DICOM Viewers. Многие из них помогают получить 2D-изображение и такого результата будет достаточно для того, чтобы сформировать диагноз. Более существенным методом визуализации DICOM файла является реконструкция 3D модели на основе комбинации 2D-изображений, которые соответствуют разным снимкам (сечениям) МРТ исследования. В результате реконструкции важно получить не только внешний вид реконструируемого объекта, но и связать его геометрию с физическими характеристиками, такими как плотность слоев и т. д.

Расстояние между последовательными участками МРТ-изображения обычно больше размера двумерного пикселя на плоскости. Объемная модель будет состоять из вокселей, которые являются трехмерным аналогом 2D-пикселей. Таким образом, геометрические параметры вокселей объемной модели могут отличаться во всех трех плоскостях. Элемент данных состоит из вокселей, которые соотносятся с размером пикселей, принадлежащих плоскости и высоте, соответствующую расстоянию между сечениями изображения.

В данной статье рассмотрены методики 3D-реконструкции DICOM изображения, а также предложен один из вариантов предобработки 2D-изображения для повышения качества реконструкции.

2. Предобработка изображения

Изображения формата DICOM в большинстве случаев требуют некоторой предобработки, прежде чем использовать их в различных методах. Чтобы способствовать предобработке, можно «улучшить» изображение. Для этого улучшения предлагается методика выравнивания гистограммы, которая используется для улучшения контрастности изображения. На рис. 2 представлено DICOM изображение и значения интенсивности без выравнивания гистограммы.

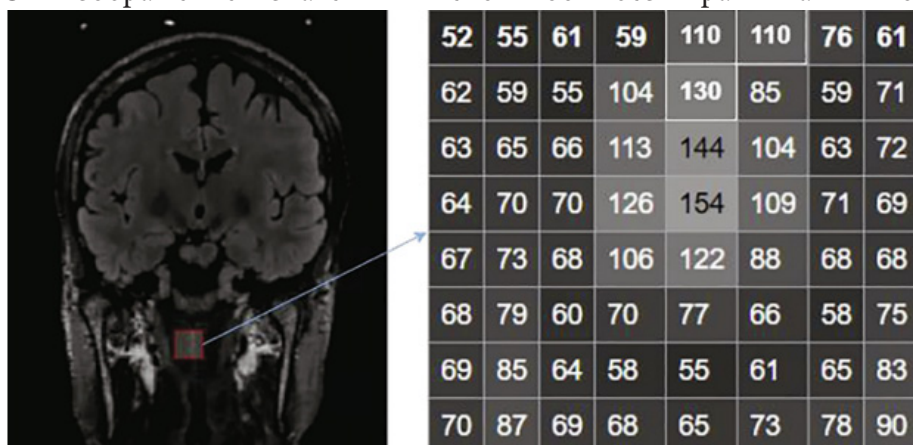


Рис. 2. DICOM изображение и значения интенсивности без выравнивания гистограммы

Выравнивание гистограммы[10] использует кумулятивную функцию распределения интенсивности пикселей. Эта функция отражает вероятность появления пикселей с уровнем интенсивности i изображения $x(p_x(i))$:

$$p_x(i) = p(x=i) = \frac{n_i}{n}, 0 \leq i \leq L, \quad (1)$$

где L — максимальное значение уровня интенсивности пикселя.

Так как МРТ-изображение формата DICOM отражает черно-белое изображение с интенсивностью (v) в пределах от 0 до 255 для каждого вокселя, то $L = 256$. Результаты, полученные из (1), позволяют получить значение кумулятивной функции распределения для каждого из значений интенсивности пикселей:

$$cdf_x(v) = \sum_{j=0}^v p_x(x=j). \quad (2)$$

Таким образом, получаем выравнивание гистограммы:

$$h(v) = \left(\frac{cdf(v) - cdf_{\min}}{(M \times N) - cdf_{\min}} \right) \times (L - 1). \quad (3)$$

Значение, полученное в результате выполнения выравнивания гистограммы, заменяет изначальное значение интенсивности и таким образом получается улучшенное изображение, с более высокой контрастностью нежели оригинал. Разницу между изображениями можно увидеть, рассмотрев рис. 3 и рис. 4.

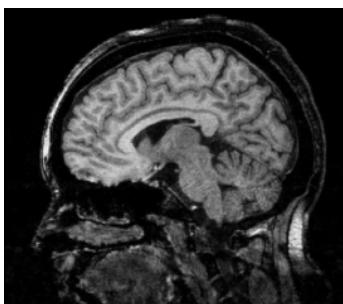


Рис. 3. Изначальное изображение

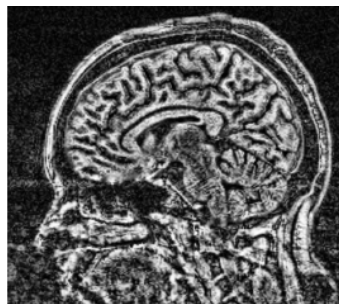


Рис. 4. Результат выравнивания гистограммы

Далее, будут рассмотрены методики реконструкции изображения, которые обобщены в три группы: мульти-планарный рендеринг, объемный рендеринг и рендеринг поверхности.

3. Мульти-планарный рендеринг

Мульти-планарный рендеринг (МПР) — это метод, который используется для визуализации значений интенсивности серого цвета для объемного изображения. По факту, МПР — это двумерное представление реконструкции DICOM изображения в другой плоскости. Аксиальные срезы группируются в сагитальную и коронарную плоскости, что позволяет увидеть всю модель со всех сторон.[3]

Главным достоинством мульти-планарной методики является то, что она не зависит от того, с какого ракурса была проведена томография, что делает возможным визуализировать данные, которые были измерены в различных срезах одного двумерного изображения. Другим немало важным плюсом является вычислительная скорость данного метода. Недостатком же будет то, что визуализированные данные, как и оригинальное изображение, представлены лишь в двух плоскостях[4]. Пример МПР представлен на рис. 5.

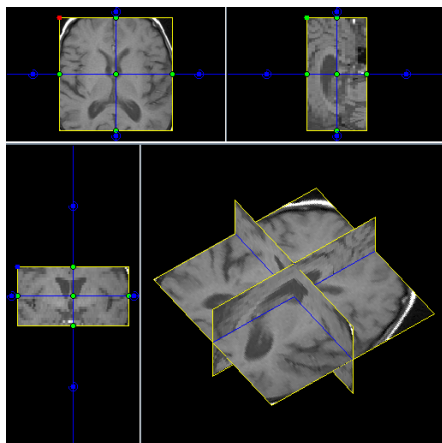


Рис. 5. Пример МПР на срезе человеческого мозга

4. Объемный рендеринг

Объемный рендеринг выполняется методом Ray Casting (RC). При RC, также известном как обратное отображение, луч запускается из каждого пикселя в плоскости обзора, а информация из всех вокселей собирается в объемные данные, пересекающие текущий луч или пиксель. Основная цель RC — позволить наилучшим образом использовать трехмерные данные и не пытаться наложить на них какую-либо геометрическую структуру. Это решает одно из наиболее важных ограничений текстурных методов, а именно то, как они (методы) отображают проекцию тонкой оболочки в пространстве.

Само моделирование предполагает маркировку как самих текстур на всех слоях модели, так и границ между текстурами [6]. Объемный рендеринг включает в себя шаг создания двумерного изображения из проекции трехмерного набора данных. Основной целью данного метода является превращение трехмерного набора данных в 2D изображения, где будут отчетливо видны необходимые области. В отличие от рендеринга поверхностей, объемный рендеринг не использует жесткий набор базовых элементов для визуализации объекта, однако определяет модель, рассматривая всю область изображения как полупрозрачную (с низким уровнем интенсивности) объемную модель и задавая значения плотности (интенсивности) для каждого вокселя в изображении. Воксели не отображаются непосредственно в процессе рендеринга, вместо этого значения интенсивности из вокселей используются передаточной функцией, чтобы маркировать набор данных. Передаточная функция преобразует каждое значение в трехмерном наборе данных в цвет и интенсивность пикселей, отображаемых на экране. Пример объемного рендеринга участка мозга человека представлен на рис. 6.

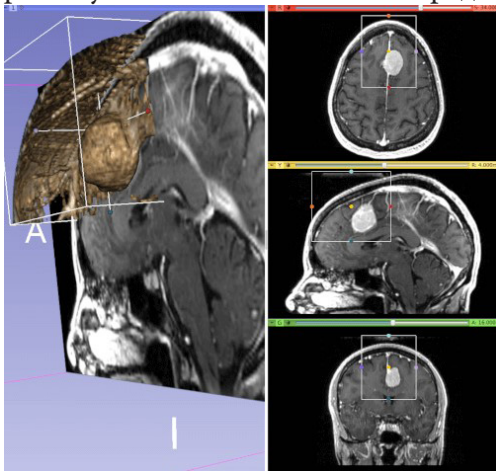


Рис. 6. Пример объемного рендеринга участка мозга человека

5. Рендеринг поверхности

Рендеринг поверхности — это способ визуализировать объект с помощью данных изображения в виде жесткого набора определенных элементов, таких как воксели. Эти наборы представляют либо границы реконструируемой модели, либо саму модель и формируют треугольную сетку вокселей, которая является полигональной сеткой трехмерного скалярного поля или же изоповерхностью. Это поверхность скалярного поля $u = u(x, y, z)$, у которой множество точек пространства принимает только значение (изозначение) c , т. е. $u(x, y, z) = c$. Одним из популярных методов построения изоповерхности из трехмерного набора данных является алгоритм «марширующих кубов». Этот алгоритм создаёт треугольную сетку (сетку полигонов) путём формирования (вычисления) изоповерхности из набора дискретных данных. Алгоритм марширующего куба можно разделить на две основных части. В первой части необходимо выяснить, как определить сечения поверхности, которая разрезает отдельно взятый куб [9]. Всего 256 возможные конфигурации углов, если классифицируется каждый угол как находящийся ниже или выше изозначения. В этих 256 возможных случаях два являются тривиальными. Они являются случаями, когда все точки поверхности находятся либо внутри, либо снаружи куба и которые не соответствуют изоповерхности. Для остальных конфигураций, необходимо решить, пересекает ли изоповерхность каждое из ребер куба, и используются ли точки этого пересечения для построения одного или нескольких треугольников изоповерхности. Когда значение скалярного поля в вершине больше или равно изозначения, то вершина будет являться внутренней, а когда значение вершины меньше изозначения, вершина будет внешней. Различные вариации пересечения в методе марширующих кубов представлены на рис. 7.

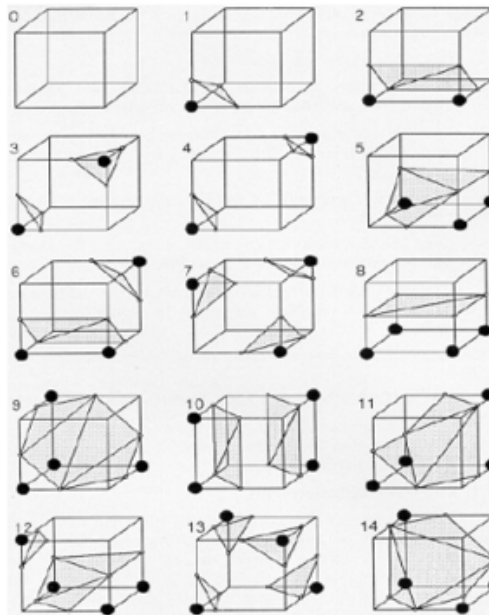


Рис. 7. Различные вариации пересечения в методе марширующих кубов

Последним шагом в марширующих кубах является вычисление единичной нормали для каждой вершины треугольника сетки. Полученные значения используются для создания затенения (по методу Гуро) на полученном трёхмерном изображении в алгоритмах рендеринга. Интерполируем нормали вдоль рёбер каждого куба, чтобы найти нормали сгенерированных вершин, для корректного отображения модели при использовании освещения. [7].

Чтобы посчитать значение нормали поверхности, необходимо определить градиентный вектор G . Градиент в вершине куба (i, j, k) находится по формуле:

$$G_x(i, j, k) = (D(i+1, j, k) - D(i-1, j, k)) / \Delta x$$

$$G_y(i, j, k) = (D(i, j+1, k) - D(i, j-1, k)) / \Delta y,$$

$$G_z(i, j, k) = (D(i, j, k+1) - D(i, j, k-1)) / \Delta z$$

где $D(i, j, k)$ — значение интенсивности в пикселе (i, j) сечения k ; Δx , Δy , Δz — длины ребер куба.

Пример получения реконструкции методом марширующих кубов представлен на рис. 8.

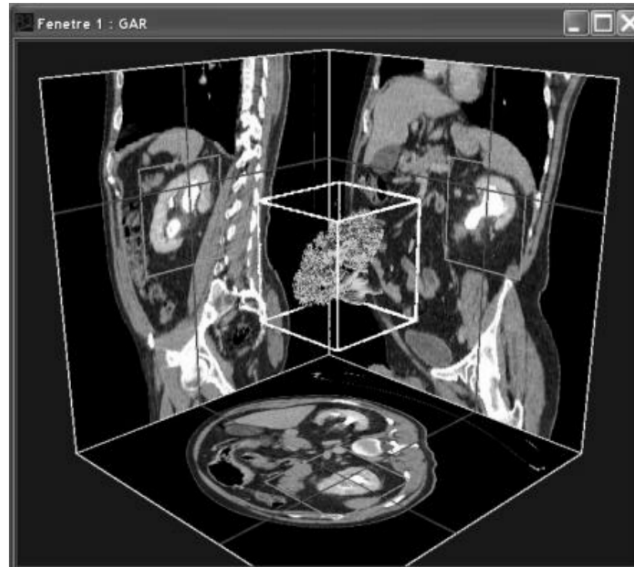


Рис. 8. Пример получения реконструкции «марширующими кубами»

6. Вычислительный эксперимент

Был проведен сравнительный анализ описанных методик 3D реконструкции DICOM изображений. Средствами языка C++ и C#, библиотеками OpenCv, VTK, а также средой Microsoft Visual Studio 2019 были программно реализованы рассмотренные методы. На основании созданного ПО, авторами был проведен эксперимент. Входными данными эксперимента являются 15 МРТ-исследований в формате DICOM. Сделанные выводы, приведенные в табл. 1.

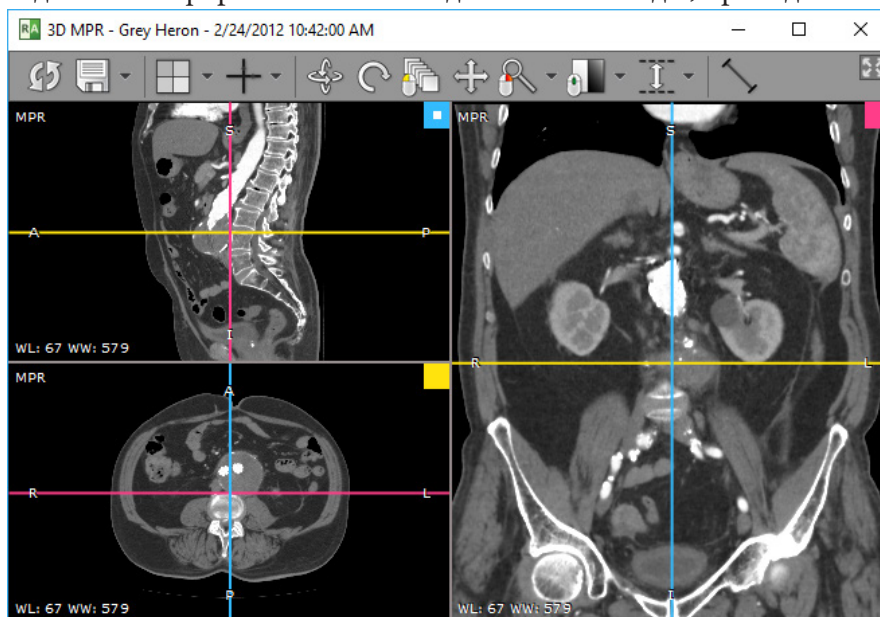


Рис. 9. Пример работы созданного ПО

Сравнительный анализ методов

Наименование метода	Положительные стороны	Отрицательные стороны	Качество выполнения
Мульти-планарный рендеринг	Возможность визуализировать данные, которые были получены из разных срезов DICOM изображения	Визуализация только в 2D	Качество увеличивается с увеличением количества итераций
Марширующие кубы	Прост в реализации	Генерация неоднозначных поверхностей и огромное кол-во треугольных элементов	Быстрое вычисление и качественное визуальное отображение результатов
Рендеринг поверхности	Рендеринг в реальном времени	Много погрешностей, требуются дальнейшие вычисления	Высокое качество визуализации и отсутствие задержки в процессе

В таблице представлены положительные и отрицательные стороны предложенных методов трёхмерной реконструкции. Основываясь на результатах, явно можно выявить лидера в предложенных методах — алгоритм «марширующих кубов» из-за его простоты в программной реализации, низкой вычислительной сложности и качестве

Заключение

В данной статье были рассмотрены некоторые методы трехмерной реконструкции медицинских изображений в формате DICOM. Данное исследование помогает получить представление о доступных методах рендеринга. Также, в ходе эксперимента был выявлен наиболее эффективный метод трехмерной реконструкции на данном этапе развития технологий в лице алгоритма «марширующих кубов».

Работа была выполнена под руководством доцента, канд. физ.-мат. наук, доцента кафедры вычислительной математики и прикладных информационных технологий ВГУ, Медведева Сергея Николаевича и профессора, д-ра техн. наук, заведующего кафедрой вычислительной математики и прикладных информационных технологий ВГУ, Леденевой Татьяны Михайловны.

Литература

1. The DICOM Standard. – Режим доступа: <http://dicom.nema.org/standard.html>. – (Дата обращения: 25.05.2017).
2. *Schardt, D.* Heavy-ion tumor therapy: Physical and radiobiological benefits / D. Schardt, T. Elsasser // *Rev. Mod. Phys.* – 2010. – 82. – P. 383–425.
3. *Fuchs, H.* Optimal Surface Reconstruction from Planar Contours / H. Fuch, Z. M. Kedem, S. P. Uselton // *Communications on ACM.* – 1977. – V. 20, No 10.
4. *Jones, M.* Glossary / M. Jones, A. Leu, R. Satherley, S. Treavett // London: Springer. – 2000. – P. 395–406.

5. *Grau, V.* Improved watershed transform for medical image segmentation using prior information / V. Grau, A.U.J. Mewes, M. Alcaniz, R. Kikinis, S.K. Warfield // Transactions on Medical Imaging. – 2004. – Vol. 23(4) – P. 447–458.

6. *Robert, A.* Volume Rendering / A. Robert, Drebin, L. Carpenter, P. Hanrahan // SIGGRAPH '88. Atlanta. – 1988.

7. *Levoy, M.* Display of Surfaces from Volume Data / M. Levoy // IEEE Computer Graphics and Applications. – 1988. – 8 n. 3.

8. *Lee, B.* A Fast Accurate 3d Surface Reconstruction Method Of Medical Image Based On Modularization / B. Lee, L.-F. Tian, C. Ping, H.-Q. Mo, Z.-Y. Mao // Proceedings of the Fourth International Conference on Machine Learning and Cybernetics. – 2005.

9. *William, E. L.* Marching Cubes: A High Resolution 3d Surface Construction Algorithm / E. L. William, H. E. Cline // Computer Graphics. – 1987. – V. 21, No 4.

10. *Гонсалес Р., Вудс Р.* Цифровая обработка изображений. – М. :Техносфера, 2005. – 1072 с.

Черняев Дмитрий Евгеньевич — студент 1-го курса магистратуры кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: dcherhtc@gmail.com.

Енокян Виктория Арамовна — студентка 1-го курса магистратуры кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: vika-yenokyan@mail.ru

Морозов Дмитрий Андреевич — студент 1-го курса магистратуры кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: white_bear1902@mail.ru

Шатковская Анна Владимировна — студентка 1-го курса магистратуры кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: shatkovskaya_a_v@mail.ru

Новоскольцев Илья Юрьевич — студент 1-го курса магистратуры кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: skreb.rus@gmail.com

Петрова Виктория Геннадьевна — студентка 1-го курса магистратуры кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: petrovavika19971997@gmail.com

ПРОБЛЕМЫ ПОДГОТОВКИ ДАННЫХ ДЛЯ СЕРВИСОВ ВУЗАЛИЗАЦИИ ПЛАТФОРМ BI

Д. В. Шерстюк, В. В. Ухлова

Воронежский государственный университет

Введение

Визуализация результатов исследования в настоящее время является мощнейшим инструментом в принятии решений. Она позволяет упростить понимание информации за счет отображения данных в различных разрезах. Сервисы визуализации на рынке ПО представлены платформы класса Business Intelligence. Их функциональность позволяет импортировать данные из разных источников, агрегировать и рассматривать по различным схемам и моделям данных. Помимо этого, они представляют собой широкий выбор визуального представления данных: от простых таблиц до совмещенных графиков. Однако для ряда задач аналитики не все встроенные функции подходят, а некоторые из них не реализованы вовсе. В статье рассматриваются проблемы подготовки данных, решение которых не возможно средствами автоматизации ПО Business Intelligence (BI) и требует доработки ПО.

1. Обзор существующих решений

Традиционными для аналитиков являются платформы класса Business Intelligence: Power BI (Microsoft Corporation), Tableau (Tableau Software), BI Qlik Sense (Qlik Technologies), Zoomdata, Sisense, Information Builders. Их принцип действия заключается в том, что: к системе подключают источники данных, формируется единое хранилище, производится обработка исходной информации, данные отображаются в отчетах. В качестве источников таких платформ могут выступать облачные системы (Oracle Cloud, Google BigQuery, Microsoft Azure и их аналоги), файловые системы (Excel, XML, PDF и иные табличные файлы), реляционные базы данных (SQL Server, MySQL, Oracle).

В состав BI-решения входят:

- инструменты интеграции и очистки данных;
- аналитическое хранилище данных;
- средства Data Mining;
- инструменты визуализации данных.

Инструменты интеграции и очистки данных (ETL) извлекают информацию из внешних систем-источников, трансформируют её, очищают и загружают в единое хранилище.

Аналитическое хранилище данных представляет собой информационную базу, которая «умеет» структурировать и анализировать данные.

Средства Data Mining — это инструменты, которые обрабатывают данные и анализируют их по различным срезам. Они выявляют зависимости и тренды. При этом могут использоваться самые разные методы обработки информации: от статистики и прогнозирования до семантического анализа.

Инструменты визуализации данных представляют функционал для формирования отчетов пользователя. В зависимости от задач отчеты могут строиться по утверждённому формату или быть аналитическими. При построении аналитических отчетов пользователи самостоятельно устанавливают перечень отображаемых показателей, сортируют данные и выстраивают фильтры.

2. Проблемы работы с данными на примере платформы Power BI

2.1. Заявленный функционал ПО платформы BI

Платформа Power BI (Microsoft) может быть запущена как онлайн-сервис, так и Desktop приложение. Ее возможности позволяют подключить более 70-ти источников данных: txt, xlsx, csv файлы, SQL БД, данные получаемые через прямое подключение через API.

ПО имеет встроенные средства для обработки и очистки данных, приведения их к единому стандарту и объединения всех разрозненных таблиц, изучения созданной модели данных, моделируя новые таблицы, фильтры, вычисления, формулы и дополнительные метрики, создания графических отчетов с интерактивными визуализациями графиков, гистограмм, диаграмм, таблиц, карточек, матриц, срезов, карт и прочих визуализаций.

Согласно названиям инструментов, сценарий работы с данными на платформе представим в виде пяти этапов (рис. 1):

- 1) «Получить данные»;
- 2) «Редактор запросов Power Query»;
- 3) «Связи»;
- 4) «Данные»;
- 5) «Отчет».



Рис.1 Сценарий работы с данными на платформе Power BI

«Получение данных». Пользователю необходимо выбрать источник и активировать сервис загрузки данных. Для возможности обработки данных предусмотрен сервис «Редактор запросов». Сервис позволяет подготовить данные для формирования отчета. Сервис позволяет настраивать шаги обработки в зависимости от потребностей аналитика. Платформой BI реализованы возможности удаления столбцов, строк, переименования столбцов, объединения и разъединения значений ячеек данных, переформатирование данных, фильтрация и сортировка данных.

Режим «связи». Позволяет установить связи для таблиц данных.

Режим «данные». Необходим для создания дополнительных данных (меры, столбцы, таблицы) из уже имеющихся. Все расчеты новых значений производятся в строке формул на основе языка DAX.

Режим «отчеты». Именно в этом режиме в Power BI создаются отчеты с интерактивными визуализациями. Сервисы доступные пользователю — «Визуализации» и «Фильтры». В визуализа-

циях сосредоточены все графические инструменты, при помощи которых можно создавать интерактивные отчеты. В панели фильтры располагаются те фильтры, которые требуются на уровне всего отчета или страницы отчета, или конкретной отдельно взятой плитки визуализации.

2.2. Постановка задачи визуализации

Рассмотрим задачу, заключающуюся в мониторинге дорожно-транспортных происшествий (ДТП). В качестве отчета должна выступать карта, отображающая информацию в режиме реального времени о месте ДТП и его основных характеристиках (вид ДТП, время фиксации) (рис. 2).

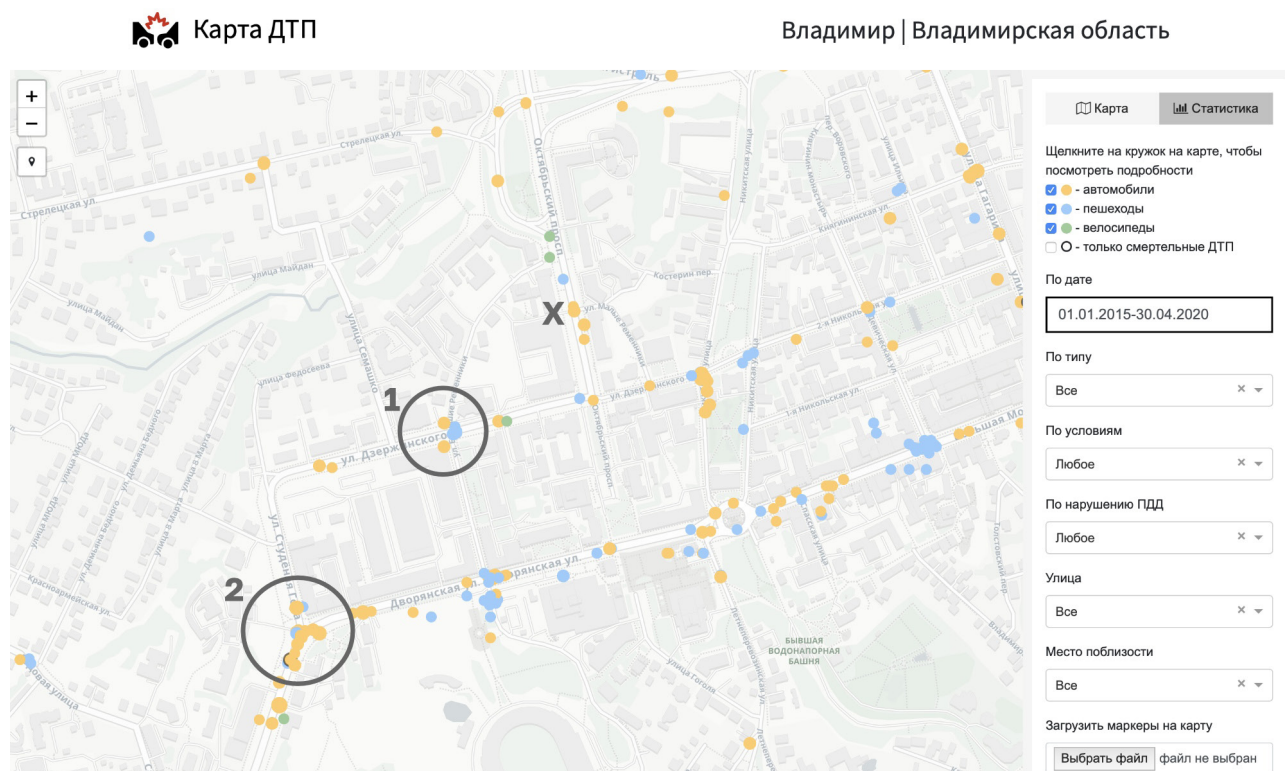


Рис. 2. Карта, отображающая информацию о ДТП

Исходными данными задачи является информация, приходящая из системы Единой дежурной диспетчерской службы (ЕДДС) и базы данных ГИБДД региона (субъекта). Выходными данными является массив координат точек локаций ДТП. Соответственно, решение задачи интеграции заключается в получении данных из внешних ИС, преобразовании этих данных в формат системы ВІ и отображение на карте региона.

Уточним задачу. Данные на платформу ВІ будут выгружаться в «сыром» виде из нескольких источников. Если речь идет о статичных данных, то они выгружаются из системы-источника в исходном формате. В случае потоковых данных возможны сценарии выгрузки через АРІ как «сырых» данных, так и предобработанных (только необходимых данных и в «нужном» виде). Далее данные необходимо сохранить в БД. В случае простой выгрузки («сырые» данные) их следует очистить и преобразовать в соответствии с требованиями конечного вида визуализации.

2.2. Проблемы подготовки данных

При такой постановке задачи первым, с чем сталкивается пользователь, является объединение данных из разных хранилищ.

Проблемы с данными:

- 1) наличие дублей; получение одних и тех же данных из нескольких систем;
- 2) пустые значения по некоторым атрибутам данных;
- 3) противоречивые данные, в том числе и при наличии дублей по ключевым атрибутам;
- 4) несогласованность данных, отсутствие единого формата;
- 5) расхождения в числе записей;
- 6) недостоверные исходные данные или ошибки в данных на самом источнике данных.

В сервисе Power Query реализованы возможности удаления столбцов, строк, переименования столбцов, объединения и разъединения значений ячеек данных, переформатирование данных, фильтрация и сортировка данных.

Не все случаи, рассмотренные выше, решаются с помощью этих инструментов. В случае применения как решение проблемы настройки API следует полностью проработать логику загрузчика.

Если выгрузка данных из источника повторяется через определенные временные интервалы, то проблемой также может стать настройка расписания выгрузки. В этом случае настройка API позволяет нивелировать проблему указанием временных интервалов, но не способна полностью решить проблему согласования выгрузки.

И еще одной проблемой платформы является ограничение с загрузкой и обработкой данных более 2 ГБ (значение зависит от реализации и настройки ПО).

Заключение

Современные задачи аналитики ставят своей целью не отображение данных, получение новых знаний от существующих данных. Это в свою очередь требует, во-первых, отображения (исследования) большого количества данных, во-вторых, изменения подхода к отображению данных. При этом скорость создания и совершенствования сервисов визуализации не поспевает за потребностями аналитиков. В связи с этим еще существенно большей ряд проблем работы с данными требует доработки существующего ПО. Выявление этих проблем и подробное описание существенно позволит их решить, хотя бы за счет разработки средств API, а в дальнейшем и разработки новых сервисов для платформ BI заключении приводятся основные результаты исследования.

Литература

1. Уклова, В. В. Особенности визуализации данных в Ситуационных Центрах субъектов Российской Федерации / В. В. Уклова, Ю. А. Максименко // Межвузовская научная конференция молодых ученых и студентов «Математика, информационные технологии, приложения» (Воронеж, 26 апреля 2021 г.) – Воронеж : Издательско-полиграфический центр «Научная книга», 2021. – С. 120–124.

2. Уклова, В. В. Практические аспекты интеграции информационных систем с целью решения задач аналитики / В. В. Уклова, Ю. А. Максименко // Сборник трудов Международной конференции «Актуальные проблемы прикладной математики, информатики и механики» (Воронеж, 7–9 декабря 2020 г.) – Воронеж : Издательство «Научно-исследовательские публикации», 2021. – С. 407–411.

3. Deckler, G. Learn Power BI: A beginner's guide to developing interactive business intelligence solutions using Microsoft Power BI / G. Deckler – 1st Edition – Birmingham: Packt Publishing Ltd, 2019. – С. 64-100

4. Powell, B. Mastering Microsoft Power BI: Expert techniques for effective data analytics and business intelligence / B. Powell – 1st Edition – Birmingham: Packt Publishing Ltd, 2018. – С. 43–153.

5. *Sherman, R. Business Intelligence Guidebook: From Data Integration to Analytics / R. Sherman. – 1st Edition – Burlington : Morgan Kaufmann, 2014. –510 p.*

Шерстюк Дмитрий Витальевич — студент 4-го курса бакалавриата кафедры математических методов исследования операций Воронежского государственного университета.
E-mail: sherstyuk2000.ru@gmail.com

Углова Вера Владимировна (научный руководитель) — канд. физ.-мат. наук, доцент кафедры математических методов исследования операций Воронежского государственного университета. E-mail: Ukhlova@amm.vsu.ru

ТЕПЛОВАЯ КОНВЕКЦИЯ В КВАДРАТНОЙ ОБЛАСТИ ТЕПЛОВЫДЕЛЯЮЩЕЙ ЖИДКОСТИ

В. А. Шишкалов

*Военный учебно-научный центр Военно-воздушных сил
«Военно-воздушная академия имени профессора Н. Е. Жуковского и Ю. А. Гагарина»*

Введение

В настоящее время в авиации, в основном, используются углеводородные топлива [1]. Развитие авиационной техники идет по пути повышения лётно-технических данных, что связано с переходом на альтернативные виды топлива, среди которых наиболее перспективным является жидкий водород и жидкий кислород [2]. Применение криогенного топлива сопряжено с решением ряда проблем, главная из которых обусловлена необходимостью его хранения при криогенных температурах [3].

В связи с этим рассматриваются тепловые процессы, возникающие в ньютоновской жидкости при хранении в специальных резервуарах, имитируемых квадратной каверной.

1. Решение модельной задачи

1.1. Постановка задачи

Рассматривается квадратная область (рис. 1) со стороной h м, содержащая теплопроводную среду с однородной мощностью тепловыделения q , Вт/м³, и известными теплофизическими характеристиками — плотностью ρ , кг/м³; теплопроводностью λ , Вт/(м·К); теплоемкостью C_p , Дж/(кг·К) одна сторона которой поддерживается при температуре t_1 , К, а остальные — при температуре t_0 , К.

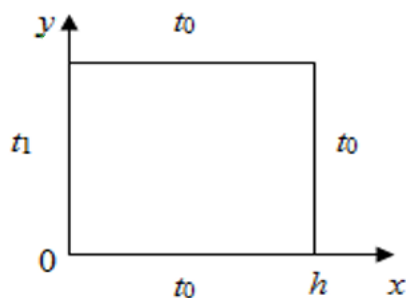


Рис. 1. Моделирование теплопроводной среды

1.2. Математическая модель

Математическая формулировка задачи в этом случае будет:

$$\rho C_p \frac{\partial t}{\partial \tau} = \lambda \left(\frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} \right) + q; \quad (1)$$

$$t(0; y) = t_1; \quad (2)$$

$$t(h; y) = t(x; 0) = t(x; h) = t_0, \quad (3)$$

где τ — текущее время, с; t — локальная температура, К.

Пусть для определенности $t_1 > t_0$, тогда система (1)–(3) в безразмерном виде такова:

$$\frac{\partial t}{\partial \theta} = \frac{1}{\text{Pr}} \left(\frac{\partial^2 T}{\partial X^2} + \frac{\partial^2 T}{\partial Y^2} \right) + \frac{1}{\text{Pr}}; \quad (4)$$

$$T(0; Y) = T_1; \quad (5)$$

$$T(1; Y) = T(X; 0) = T(X; 1) = 0, \quad (6)$$

где $X = \frac{x}{h}$; $Y = \frac{y}{h}$; $\theta = \frac{\tau v}{h^2}$; $T_1 = \frac{(t_1 - t_0)\lambda}{qh^2}$; $\text{Pr} = \frac{v}{a}$; v, a — кинематическая вязкость и теплопроводность среды, $\text{м}^2/\text{с}$.

2. Решение

Будем рассматривать стационарный случай, тогда система (4)–(6) трансформируется в систему

$$\frac{\partial^2 T}{\partial X^2} + \frac{\partial^2 T}{\partial Y^2} = -1; \quad (7)$$

$$T(0; Y) = T_1; \quad (8)$$

$$T(1; Y) = T(X; 0) = T(X; 1) = 0. \quad (9)$$

Применим к (7)–(9) конечное интегральное синус-преобразование [4] по переменной Y :

$$\frac{d^2 T_Y}{dX^2} - \mu^2 T_Y = \frac{1}{\mu} (\cos \mu - 1); \quad (10)$$

$$T_Y(0) = -\frac{T_1}{\mu} (\cos \mu - 1); \quad (11)$$

$$T_Y(1) = 0, \quad (12)$$

где T_Y — изображение T , μ — корни характеристического уравнения $\sin \mu = 0$. Решение уравнения (10) с граничными условиями (11) и (12) имеет вид:

$$T_Y = \frac{\cos \mu - 1}{\mu \cdot \pi} \left[\frac{\text{sh}(\mu X) - \left(T_1 - \frac{1}{\mu^2} \right) \text{sh}[\mu(1 - X)]}{\text{sh} \mu} - \frac{1}{\mu^2} \right]. \quad (13)$$

Используя формулу обращения интегрального синус-преобразования [4], получим

$$T = \frac{2}{\pi} \sum_{m=1}^{\infty} \left\{ \frac{\cos \mu_m - 1}{\mu_m} \left[\frac{\text{sh}(\mu_m X) - \left(T_1 - \frac{1}{\mu_m^2} \right) \text{sh}[\mu_m(1 - X)]}{\text{sh} \mu_m} - \frac{1}{\mu_m^2} \right] \sin(\mu_m Y) \right\}, \quad (14)$$

где $\mu_m = \pi m$, $m = 1, \infty$.

3. Полученные результаты

Для выражения (14) получены графические зависимости температуры нагрева приземного слоя атмосферы T от координат X и Y (рис. 2).

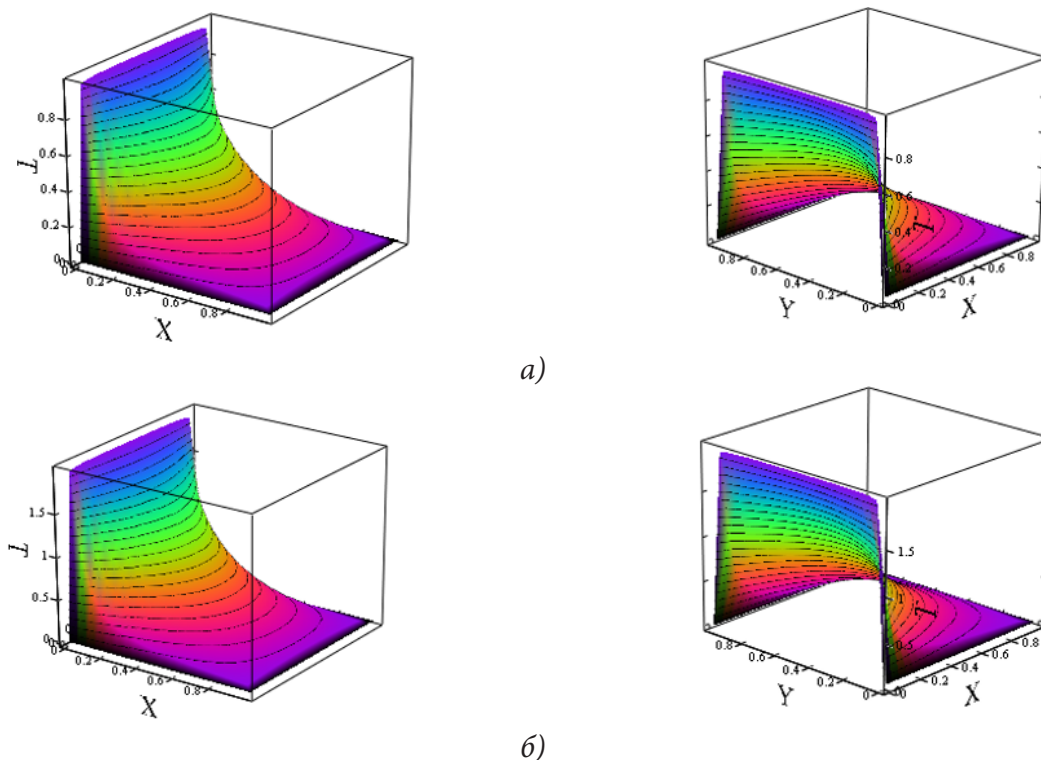


Рис. 2. Температурные поля при различных T_1 : а – 1; б – 2, соответствующих решению (14)

Заключение

Исходя из вида рис. 2, можно сделать вывод, что увеличение T_1 приводит к пропорционально-эквидистантному увеличению значений температур в области решения. Кроме того, несмотря на разрывы 1-го рода по температуре на границе, решения указанных задач находятся в классе непрерывных и дифференцируемых функций. Анализ рядов, проведенный численно, позволяет сделать вывод о быстрой их сходимости. Например, представленные температурные поля на рис. 2, получены при $n = 25$, что вполне по точности удовлетворяет инженерной практике. Таким образом, неоднородность температурного поля определяется неоднородностью граничных условий, которую необходимо учитывать в задачах переноса теплоты в атмосфере в кондуктивном режиме.

Литература

1. Рыбин Н. П. Авиационные смазочные материалы и спецжидкости. – М. : МИИГА, 1985
2. Яновский Л. С., Дубовкин Н. Ф., Галимов Ф. М. и др. Горюче-смазочные материалы для авиационных двигателей. – Казань : Казанский государственный технический университет им. А. Н. Туполева, 2002.
3. Архаров А. М., Кунис И. Д. Криогенные заправочные системы стартовых ракетно-космических комплексов. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2006. – 252 с.
4. Корн Г. Справочник по математике для научных работников и инженеров / Г. Корн, Т. Корн. – М. : Наука. – 1973. – 832 с.

Шишкалов Владислав Александрович – студент 1-го курса ВУНЦ ВВС ВВА, ф-т авиационное оборудование, vsum@rambler.ru

Сумин Виктор Александрович (научный руководитель) — канд. физ.-мат. наук, доц., доцент кафедры МиКМ (ВГУ), кафедры общепрофессиональных дисциплин (ВУНЦ ВВС ВВА).

РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ ЛИЦА, ВОЗРАСТА И ГЕНДЕРА В РЕАЛЬНОМ ВРЕМЕНИ

А. А. Щеглеватых, Е. М. Аристова

Воронежский государственный университет

Введение

В истории человечества проблема безопасности всегда занимала особое место. На протяжении веков формировались принципы противостояния различным вызовам безопасности населения от внешних угроз. Сегодня спектр вызовов безопасности: это проблемы терроризма. С проблемой терроризма тесно связана проблема распространения региональных конфликтов, т. к. они тесно связаны с первыми. Прибавились сегодня и нетрадиционные вызовы, которые, наравне с традиционными, угрожают стабильности и безопасности жизнедеятельности людей одной страны и даже в какой-то степени всего мира. В первую очередь, это касается острой проблемы экологии, риски инфекционных заболеваний, а также угроза несчастного случая.

Чтобы пропустить человека на какой-то объект, часто прибегают к применению электронных карточек с данными о человеке на них, но эти карточки часто теряются, забываются. Поэтому возникает вопрос, что же может послужить личным идентификатором человека? Ответ прост — наше лицо. Лицо нельзя забыть или потерять. Так и появилась идея создания пропускной системы с использованием распознавания лиц.

1. Используемый стек технологий

Для реализации приложения был выбран язык JavaScript. Данный язык программирования сочетает в себе все самое лучшее от функциональных языков, например, автоматическое приведение типов (язык программирования C++), функции, как объекты первого класса (язык программирования Scala), замыкания (язык программирования C#), анонимные функции (язык программирования Python) и др. Всё это делает язык более гибким и удобным в использовании [1]. Язык с каждым годом все динамичнее развивается и даже предоставляет возможность создания приложений с машинным обучением.

Сейчас насчитывается большое количество библиотек машинного обучения для языка JavaScript — Brain.js, Synaptic, Neataptic, Tensorflow.js и др.

В данной работе использована библиотека Tensorflow.js из-за подробно описанной документации, а также краткого и доступного API [2].

2. Нейронная сеть

Технологии машинного зрения и распознавания лиц развивались очень активно с середины прошлого века. Но только сейчас стали по-настоящему хорошо работать. В качестве причин этого можно выделить:

- появились действительно мощные компьютеры, способные справиться с задачей;
- появились базы данных с фотографиями (спасибо социальным сетям);
- произошел прорыв в области нейросетей.

Все эти события позволили создать практически идеальные алгоритмы распознавания лиц.

Нейронная сеть, в простейшем случае, представляет собой математическую модель, состоящую из нескольких слоёв элементов, выполняющих параллельные вычисления. Изначально такая архитектура была создана по аналогии с мельчайшими вычислительными элементами человеческого мозга — нейронами. С усложнением задач для нейронных сетей, росла и сложность ее архитектуры. Есть разница между задачами по классификации изображений, обнаружению объектов на изображении и сегментацией изображений.

Что касается задачи распознавания лиц при помощи нейронной сети, то данный процесс состоит из следующих пунктов:

1. Непосредственное обнаружение лица — данная задача может решаться как при помощи нейронной сети, так и алгоритмом Виолы-Джонса, который как раз и используется в данной статье.

2. Расстановка антропометрических точек — процесс расстановки точек, по которым впоследствии и будут вычисляться индивидуальные характеристики человека, такие как форма носа, подбородка, контуры лица, разрез глаз, расстояние между ними и др.

3. Определение вектора лица — определение нейронной сетью вектора, который будет уникально характеризовать каждое лицо;

4. Идентификация — сопоставление результатов с определенными лицами/гендеров и т. д.

Среди популярных архитектур для построения нейронных сетей можно выделить R-CNN, YOLO, SSD, MobileSSD и др.

Для реализации приложения по распознаванию лиц, гендера и возраста в реальном времени была выбрана архитектура MobileSSD. Она представляет собой сочетание двух архитектур MobileNetV2 и SSD. Остановимся более подробно на основных составляющих архитектуры MobileSSD [3]:

1. *MobileNetV2*: архитектура представляет собой сверточную нейронную сеть, состоящую из одного обычного свёрточного слоя с 3×3 свёрткой вначале и тринадцати блоков с фильтрами для обработки входного изображения. Число фильтров постепенно увеличивается, а размерность тензора (ребро data-flow графа, представляющее собой многомерный массив данных, используемых в вычислениях в задаче распознавания лиц) уменьшается. В данной архитектуре отсутствуют слои максимальной подвыборки (max pooling-слои). Вместо них осуществляется свёртка с параметром свёрточного слоя 2 (stride). За изменение количества каналов в каждом слое отвечает гиперпараметр «множитель ширины», а за пространственные размеры входных тензоров — «множитель глубины». Настраивая эти параметры, можно варьировать скорость работы и размер потребляемой памяти.

2. *SSD*: данная архитектура может различить объект на изображении за один прогон с помощью заданной сетки окон (default box) на пирамиде изображений. Пример пирамиды изображений представлен на рис. 1.

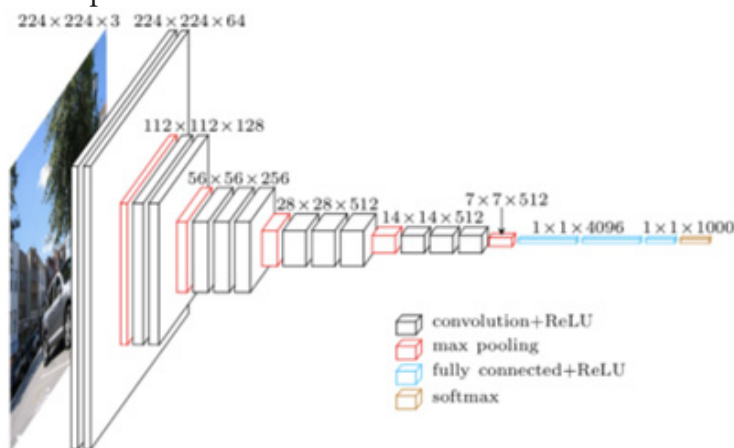


Рис. 1. Пример пирамиды изображений

Пирамида изображений закодирована в сверточных тензорах при последовательных операциях свертки и пулинга (при операции max-pooling пространственная размерность убывает). Исходя из этого, определяются как большие, так и маленькие объекты за один прогон сети.

Таким образом, первая архитектура обеспечивает точность распознавания, а вторая — скорость [3–6].

3. Архитектура приложения

Приложение для определения лица, гендера и возраста в реальном времени состоит из двух частей: серверной части, которая содержит объявление, настройку и обучение нейронной сети, а также клиентской части для демонстрации.

Работу приложения можно представить в виде схемы (рис. 2).



Рис. 2. Структура приложения

На вход программа получает изображение с камеры пользователя. Так как программа работает с динамичным изображением, то полученные входные данные каждый раз разбиваются на 30 кадров, после этого каждый кадр обрабатывается нейронной сетью и выводится усредненный ответ. Для получения изображения с веб-камеры пользователя, использовалось встроенное API браузера.

Пример задания элемента для отображения видео с веб-камеры в браузере:

```
<div style=»position: relative» class=»margin»>
  <video onloadedmetadata=»onPlay(this)» id=»inputVideo» autoplay muted playsinline</video>
  <canvas id=»overlay» />
</div>
```

Работа с полученными данными реализуется через обработчик события onPlay(). После этого запускается сначала алгоритм поиска лица, а потом определения возраста и гендера.

```
const result = await faceapi.detectSingleFace(videoEl, options)
  .withAgeAndGender()
```

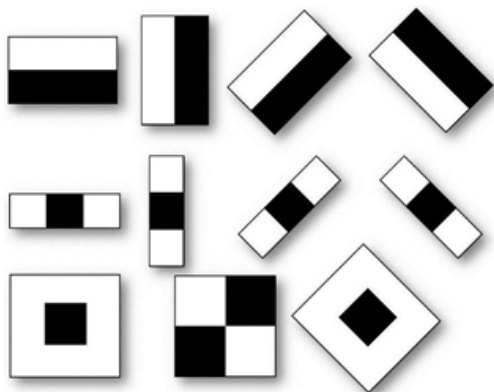


Рис. 3. Примитивы Хаара

Для последующей работы алгоритму требуется определить прямоугольник, содержащий лицо. Для решения этой задачи использовался алгоритм Виолы-Джонса. Алгоритм «сканирует» изображение и накладывает на него примитивы Хаара (рис. 3).

Для корректной работы алгоритма Виолы-Джонса яркость исходного изображения усредняется. Но даже в таком случае область глаз человека будет темнее, чем область щек, лба или переносицы. Сначала накладывается первый примитив, если лицо не может быть обнаружено, то накладывается следующий примитив, если лицо

на данном этапе может быть обнаружено, то накладываются последующие примитивы и так далее, пока лицо не будет найдено [7].

Пример кода для обнаружения прямоугольника с лицом:

```
const results = indices
.map(idx => {
  const [top, bottom] = [
    Math.max(0, boxesData[idx][0]),
    Math.min(1.0, boxesData[idx][2])
  ].map(val => val * padY)
  const [left, right] = [
    Math.max(0, boxesData[idx][1]),
    Math.min(1.0, boxesData[idx][3])
  ].map(val => val * padX)
  return new FaceDetection(
    scoresData[idx],
    new Rect(
      left,
      top,
      right - left,
      bottom - top
    ),
    {
      height: netInput.getInputHeight(0),
      width: netInput.getInputWidth(0)
    }
  )
})
```

После получения рабочей области с лицом нейронная сеть расставляет 68 биометрических точек, которые однозначно определяют лицо человека, как вектор, содержащий Евклидовы расстояния для каждой пары точек.

Для получения результата сравниваются все вектора исходного изображения и изображения пользователя. В качестве ответа берется возраст и гендер, евклидово расстояние до которого минимально.

Пример создания нейронной сети данной архитектуры:

```
export function createSsdMobilenetv1(weights: Float32Array) {
  const net = new SsdMobilenetv1()
  net.extractWeights(weights)
  return net
}

export function createFaceDetectionNet(weights: Float32Array) {
  return createSsdMobilenetv1(weights)
}
```

Код функции для запуска нейронной сети по распознаванию гендера и возраста:

```
public runNet(input: NetInput | tf.Tensor4D): NetOutput {

  const { params } = this

  if (!params) {
    throw new Error(`${this._name} - load model before inference`)
  }

  return tf.tidy(() => {
```

```

const bottleneckFeatures = input instanceof NetInput
  ? this.faceFeatureExtractor.forwardInput(input)
  : input

const pooled = tf.avgPool(bottleneckFeatures, [7, 7], [2, 2], 'valid').
as2D(bottleneckFeatures.shape[0], -1)
const age = fullyConnectedLayer(pooled, params.fc.age).as1D()
const gender = fullyConnectedLayer(pooled, params.fc.gender)
return { age, gender }
})
}

```

Для обучения нейронной сети, использовался набор данных с сайта: <http://shuoyang1213.me/WIDERFACE/> [8]. Данный сайт предоставляет изображения в виде закодированного вектора. Обученная модель хранится в приложении.

4. Демонстрация работы

Для демонстрации работы алгоритма определения лица, возраста и гендера в режиме реального времени реализовано одностороннее веб-приложение.

Пример интерфейса:

Webcam Age and Gender Recognition Webcam Age and Gender Recognition

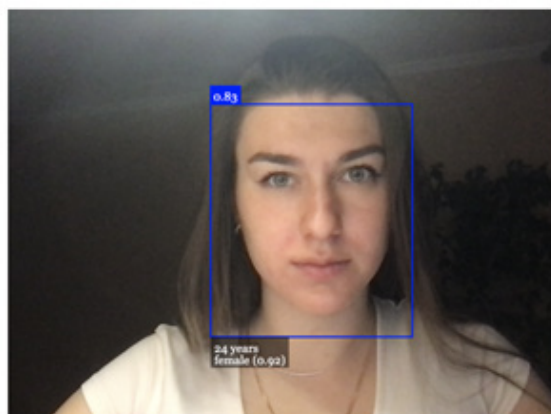


Рис. 4. Пример работы с рамкой для определения лица

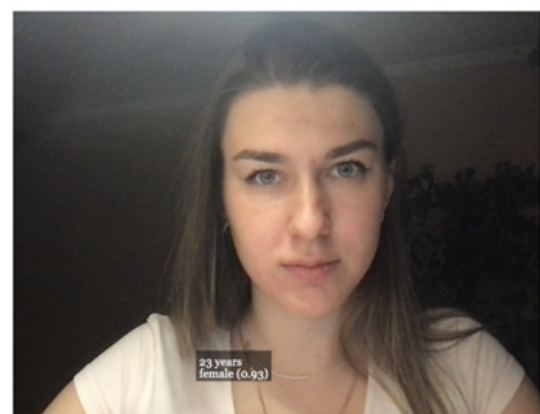


Рис. 5. Пример работы без рамки для определения лица

Для определения качества распознавания лица на изображении использовалась метрика $F1$ — гармоническое среднее между точностью и полнотой, показывающее классификацию ошибок метрики. Формула для расчета метрики $F1$:

$$F1 = 2 \frac{(Precision \times Recall)}{(Precision + Recall)},$$

где $Precision$ — точность алгоритма $\left(Precision = \frac{TP}{TP + FP} \right)$;

Recall — полнота алгоритма $\left(Recall = \frac{TP}{TP + FN} \right)$;

TP, FP, FN — истинно-положительное значение, ложно-положительное значение и ложно-отрицательное значение, соответственно, рассчитанное на основе таблицы контингентности (табл. 1) [9].

Таблица 1

Таблица контингентности

Категория <i>i</i>		Экспертная оценка	
		Положительная	Отрицательная
Оценка системы	Положительная	TP	FP
	Отрицательная	FN	TN

Заключение

В данной статье были рассмотрены основные нюансы построения приложения для распознавания лиц, возраста и гендера по видео в реальном времени. Представлена архитектура построенной нейронной сети. Продемонстрирован пример работы программы.

Данное приложение может быть полезно во многих сферах жизнедеятельности человека. Планируется доработка данного приложения и использование его, как отдельной библиотеки, для усовершенствования веб-приложения по автоматизации деятельности спортивной организации.

Литература

1. Щеглеватых А. А. Разработка веб-приложения для автоматизации деятельности спортивной организации: выпускная квалификационная работа бакалавра: 02.03.02 / А. А. Щеглеватых. – Воронеж : ВГУ, 2021. – 82 с.
2. TensorFlow. – URL: <https://www.tensorflow.org> (дата обращения: 05.04.2021).
3. MobileNet: меньше, быстрее, точнее. – URL: <https://habr.com/ru/post/352804/> (дата обращения: 07.04.2021).
4. Миллион домашних фотографий: лица, лица, лица. – URL: <https://habr.com/ru/post/538360/> (дата обращения: 02.04.2022).
5. Хайкин С. Нейронные сети. – 2 издание. – Москва: Вильямс, 2006. – 1104 с.
6. Нейронные сети. Основы. – URL: http://synset.com/ai/ru/nn/NeuralNet_01_Intro.html (дата обращения: 07.04.2021).
7. Валков В. Руководство хакера по нейронным сетям в JavaScript. – 2 издание. – Москва : Вильямс, 2021. – 536 с.
8. WIDER FACE: A Face Detection Banchmark . – URL: <http://shuoyang1213.me/WIDERFACE/> (дата обращения: 02.04.2022).
9. Оценка классификатора (точность, полнота, F-мера). – URL: <http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html> (дата обращения: 10.04.2022).

Щеглеватых Алина Александровна — магистрант 1-го года обучения кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: alinka030499@mail.ru

Аристова Екатерина Михайловна (научный руководитель) — канд. физ.-мат. наук, доц., доцент кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: pmim@yandex.ru

ВЫБОР СУБД ДЛЯ РЕАЛИЗАЦИИ СИСТЕМЫ ОНЛАЙН ТАРИФИКАЦИИ ТЕЛЕКОММУНИКАЦИОННЫХ УСЛУГ

Г. А. Ювченко

Воронежский государственный университет

Введение

Одной из задач проектирования решений программного обеспечения в телеком бизнесе является выбор системы управления базами данных (СУБД), которая будет использована для реализации программного обеспечения (ПО). Сложность состоит в том, чтобы выбрать систему, наиболее удовлетворяющую требованиям определенного функционала.

Одним из специфичных по требованиям является функционал организации системы онлайн тарификации (Online Charging System, OCS). Он представляет собой сервис ERP системы телеком операторов, который используется для учета предоставления услуг мобильной связи и обеспечивает тарификацию услуг связи в момент их предоставления.

Важность выбора и дальнейшей настройки системы управления базами данных для реализации функционала OCS определяет в возможность достижения функциональных и нефункциональных требований системы. Более того, значение корректного анализа и выбора СУБД заключается в том, что использование современных, соответствующих предъявляемым требованиям, средств программной реализации обеспечивает не только фактическое функционирование бизнеса, но и осуществление его наиболее оптимальным (технически, экономически, организационно) способом. Это и определяет актуальность работы.

1. Постановка задачи

Основная задача системы OCS заключается в онлайн тарификации. Данный процесс реализуется через механизм авторизации попытки использования услуги. В момент, когда абонент инициирует звонок/ отправку смс/ передачу данных, система управления сетью (OSS) направляет запрос в OCS, чтобы разрешить или запретить дальнейшее предоставление услуги. При этом, помимо данных о запрашиваемой операции система должна получить дополнительные данные, например, о местоположении абонента, состоянии его счета, наличии ограничений и блокировок. Результат авторизации услуги преобразуется в команду оборудованию на его совершение, а результат оказания услуги в CDR, которая направляется в биллинговую систему и запускает функционал тарификации и учета (схема представлена на рис. 1).

Системы онлайн тарификации являются высоконагруженными и высокопроизводительными системами, с соответствующими требованиями к СУБД обеспечивающими их функционирование. Во-первых, задача авторизации абонентов должна выполняться в режиме онлайн и до начала предоставления телеком услуги. С целью обеспечения высокого уровня качества и сервиса услуг телеком операторов стандартом ИМТ-2020 прописаны основные требования к производительности ПО. В частности, очень строго регламентируется время отклика. Для стандарта 4G оно должно находиться в диапазоне 50–100 мс, для сетей стандарта 5G составляет 5 мс [1].

Значимость требования обусловлена тем, что системы онлайн тарификации являются системами массового использования и даже краткосрочный сбой ведет к большому числу отказов абонентов.

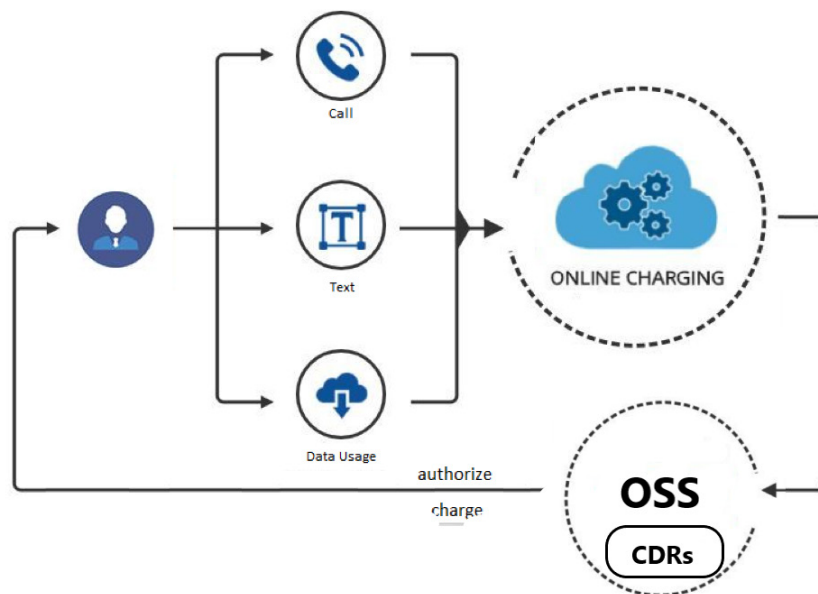


Рис. 1. Принцип работы системы OCS

Вышеуказанные функциональные требования формируют и специфические требования к программному обеспечению, базам данных (БД), реализующим функционал онлайн тарификации.

Исходя из вышесказанного, ставится задача выбора СУБД, позволяющей предоставить регламентируемый стандартом ИМТ-2020 уровень качества и сервиса пользователей при оказании услуг телеком оператора.

2. Определение требований к СУБД для реализации сервиса онлайн тарификация

Для формирования перечня требований к СУБД рассмотрим процесс онлайн тарификации с позиции обработки данных. Для этого разделим анализируемую систему на части с однотипными паттернами обработки данных и требованиями к БД. Для системы онлайн тарификации такими частями будут системы, представленные на рис. 2:

- 1) хранения пользовательских данных и информации о тарифах (Customer profile and plans);
- 2) хранения истории балансов (Balances);
- 3) тарификации использования услуг связи (Usage Charge).

Опишем требования, предъявляемые к каждой из этих частей. Данные о пользователях и тарифах устроены иерархическим образом. Ключевыми требованиями к их использованию является консистентность данных, валидация зависимостей. Определим его как требование 1, требование к производительности ПО из п. 1. Ускорение базы и приведение ее к требованиям, предъявляемым соседними частями системами, для эксплуатации возможно за счет использования методов денормализации и кэширования данных.

Хранение истории балансов — функционал, обязательный к исполнению в соответствии с требованиями регуляторов телекоммуникационной сферы. Содержит в себе все данные об использовании услуг связи. БД для него должна обеспечивать быструю и масштабную запись данных, при редком их поиске и чтении – требование 2.

Система тарификации — ядро функционала OCS, предъявляющее наиболее значимые требования к системе управления БД для своей реализации. Они заключаются в жестких требованиях по отказоустойчивости, задаваемых как величина времени в течение которого функционал системы будет не доступен, безопасности и консистентности данных.

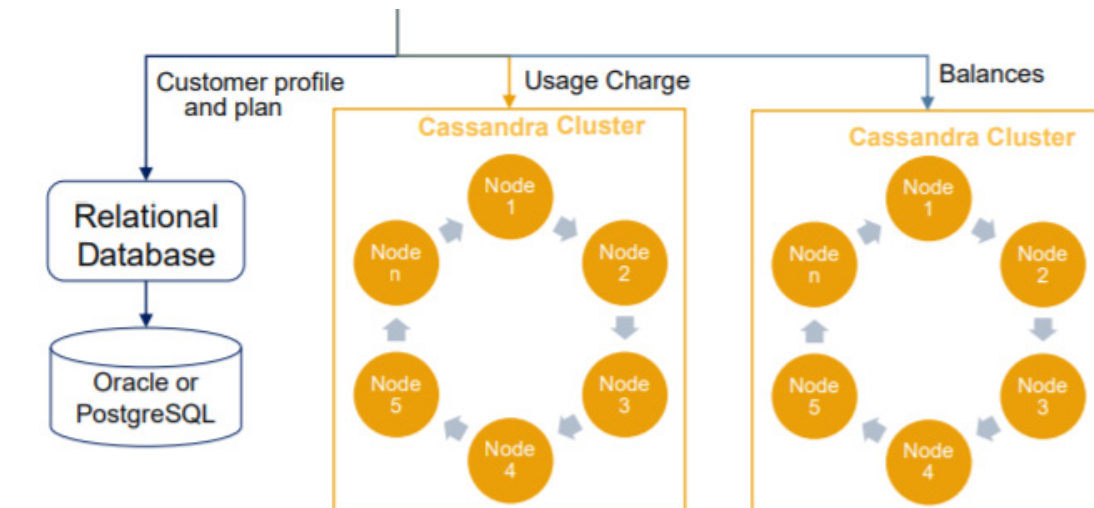


Рис. 2. Функциональные части системы онлайн тарификации

Перечень требований к ПО СУБД будет следующим:

- 1) обеспечение консистентности данных, валидация зависимостей;
- 2) обеспечение быстрой и масштабной записи данных, при редком их поиске и чтении;
- 3) высокая доступность, безопасность данных.

Добавим к этому перечню еще три дополнительных требования, определяющих специфику работы самого телеком оператора.

1. Необходимость поддержки функционала горизонтального масштабирования. Важность его применения обусловлена ростом телекоммуникационного рынка (рост абонентской базы, рост объема предоставляемых услуг и рост объема тарифицируемых на принципах «онлайн» услуг).

2. Возможность гибкой настройки системы. Она несет помимо функциональной и экономическую выгоду. Объем используемых ресурсов всегда соразмерен актуальной нагрузке без переплаты за неиспользуемые мощности.

3. Поддержка использования географически разнесенных (geo-redundant multisite) датацентров (связано с географической протяженностью зоны предоставления услуг связи).

3. Выбор СУБД для сервиса онлайн-тарификации

Метод организации СУБД в телеком бизнесе, зависит от особенностей использования и покрываемого функционала операторов связи. Согласно классификации Дж. Вейда [2], основными БД являются реляционные и нереляционные (NoSQL), которые также можно разделить по принципам организации хранения данных на реляционные, «ключ-значение», документные, графовые и колоночные.

Крупные производители имеют в своем арсенале несколько типов СУБД, как в виде отдельных продуктов, так и в виде внутренней реализации. Например, у Oracle кроме классической реляционной СУБД есть отдельный продукт Oracle NoSQL Database, который может использоваться и как документная, и как колоночная, и как «ключ-значение». Отдельное решение от того же Oracle, Autonomous Data Warehouse — это уже специализированное решение для хранилищ данных. Еще один отдельный продукт от Oracle — Oracle Graph Server для работы с графами. Apache Software Foundation также поддерживает различные типы СУБД: Apache Derby как реляционная СУБД, Cassandra как колоночная нереляционная СУБД.

Реляционные СУБД чаще всего используются для построения решений OLTP (Online Transaction Processing). В таких решениях СУБД работает с небольшими по размерам тран-

закциями, но идущими большим потоком, и при этом от системы требуется минимальное время отклика, а так же возможность, при определенных условиях, отменить любые изменения выполняемых в рамках транзакции. Для системы, в рамках которой требуется хранить значительное количество сущностей (таблиц), с различными типами связей между ними (один-к-одному, один-к-многим, многие-ко-многим), подходят реляционные СУБД. Один из основных признаков БД организуемых реляционным способом является высокая нормализация данных.

СУБД типа ключ-значение — один из самых простых типов СУБД. Это таблица с уникальным ключом и связанным с ним значением, тип значений не регламентируется. В связи с тем, они очень быстро работают и имеют единый уникальный ключ, чаще всего используются для кэширования. В некоторых реализациях данных СУБД есть возможность работать полностью в памяти (in-memory), задавать «время жизни» записи. Такие решения подходят если СУБД будет использоваться для кэширования данных или для брокеров сообщений. Так же, такая СУБД хорошо подходит для БД, где нужно хранить достаточно простые структуры и иметь к ним очень быстрый доступ. Для телеком операторов на базе СУБД типа «ключ-значение» организуются системы выявления мошенничества, сервисы интернет-рекламы, системы мониторинга состояния оборудования с предиктивными свойствами, сервисы кибер-безопасности.

Документные или документно-ориентированные СУБД — это одна из наиболее популярных разновидностей NoSQL СУБД, где основной единицей логической модели данных является документ в виде структурированного текста с определенным синтаксисом [2]. Эти СУБД топологически похожи с реляционными, но существенная разница между ними заключается в том, что БД документов только сохраняют состояние, но не поведение. СУБД данного типа подходят для хранения объектов в одной сущности, но с разной структурой. Если нужно хранить структуры, включая объекты, списки и словари, особенно в формате близкому к JSON. Область применения документных СУБД в телеком бизнесе ограничивается системами управления и мониторинга, хранилищ состояний оборудования, базами данных отдельных микро-сервисов.

Графовые СУБД — специфичный тип, предназначены для работы с графами, с их узлами, свойствами, и произвольными отношениями между узлами [2]. Данные подходящие для управления такой СУБД обычно имеют структуру подобную социальной сети, или имеют признаки системы оценок и рекомендаций. В телеком бизнесе такой функционал используется в системах геопозиционирования и системах предиктивного анализа поведения абонентской базы.

Колоночные СУБД топологически похожи на реляционные. Они так же состоят из строк, которые имеют атрибуты, а строки группируются в таблицах. Основные различия не в логических моделях, а на уровне физического хранения данных. В реляционных СУБД данные хранятся «построчно», для считывания значения определенной колонки, придется прочитать практически всю строку, как минимум от первой до нужной колонки. В колоночной СУБД данные хранятся «поколоночно». На практике это дает кардинальный выигрыш в быстродействии. Основные преимущества колоночных СУБД — эффективное выполнения сложных аналитических запросов на больших объемах, и легкое, практически мгновенное, изменение структуры таблиц с данными, плюс существенная компрессия и сжатие, которое позволяет значительно экономить место. Колоночные СУБД подходят для случаев выборки со сложными аналитическими вычислениями или при наличии жестких требований по обработке больших объемов данных. Косвенный признак таких данных - количество строк, из которых делаются выборки, превышает сотни миллионов. В телеком бизнесе они используются для систем онлайн-тарификации и систем хранения истории тарификации абонентов.

Для удовлетворения требования 1 оптимальным вариантом организации, конфигурирования и наполнения соответствующей базы данных являются реляционные БД (Oracle, PostgreSQL, в зависимости от лицензионной политики).

Для удовлетворения требования 2 подходит нереляционная БД Apache Cassandra. Она основана на модели хранения данных на базе семейства столбцов и оптимальным (по соотношению быстродействие\ресурсоемкость\отказоустойчивость) способом реализует данный функционал. Впрочем, учитывая невысокие функциональные требования, задача может быть решена и с помощью целого ряда других аналогичных БД.

Для соблюдения требований по отказоустойчивости (требование 3) единственной СУБД, готовой к промышленному использованию и предоставляющей функционал leaderless является СУБД Cassandra [3].

Поддержка функционала горизонтального масштабирования реализована в БД Cassandra. Она позволяет производить динамическое (т.е. без влияния на работоспособность существующей системы) добавление нод в кластер с помощью встроенного функционала системы. На текущий момент это единственная в мире система, имеющая практическую реализацию динамического конфигурирования кластера размером в несколько тысяч нод (кейсы Netflix и Facebook).

Поддержка использования географически разнесенных (geo-redundant multisite) датацентров может быть реализована шардированием данных и хранением части данных в режиме Active и части в режиме Standby (с зеркальным отображением на БД развернутых в других датацентрах). Однако, для систем обеспечения телекоммуникационных услуг такой формат неприемлим в силу высокой неопределенности обращения абонента к одному из датацентров (географическая мобильность абонентов). Также, его использование нецелесообразно из-за принятия новых стандартов мобильной связи 5G, по которым время обработки запроса абонента должно составлять от 5 до 20 мс (для разных типов услуг), в то время как только время отклика между географически разделенными датацентрами составляет в среднем 20 мс. Соответственно, это требования ни одной из СУБД однозначно удовлетвориться не может.

Как итог, наиболее приемлемой в контексте рассматриваемой задачи, является СУБД Cassandra Apache. Ее особенности:

- 1) нереляционная, распределенная БД, использующая модель хранения данных на базе семейства столбцов;
- 2) Leaderless функционал с отсутствием активной лидирующей ноды;
- 3) Active-active распределенная архитектура с применением асинхронной репликации, поддерживающий «безконфликтный» формат данных CRDT.

Заключение

Полученное решение основывается на рассмотрении технических аспектов процесса предоставления сервиса онлайн-тарификации. Это позволяет выявить скрытые задержки в работе сервиса, скорректировать его работу на более детальном уровне и максимально совершенствовать логику. Более того, оптимизация процесса позволяет получить телеком оператору и дополнительную экономическую выгоду за счет повышения эффективности использования СУБД и СХД.

Сформулированные требования к СУБД имеют место в банковской сфере (при предоставлении сервисов работы с банковскими картами), в системах мобильных приложений массового обслуживания (сервисы заказа такси или доставки). Соответственно, оно может быть рекомендовано для внедрения в работу организаций этих сфер.

Необходимо отметить, что БД Cassandra как нереляционная система, обладает и рядом недостатков, а точнее, отсутствием некоторого функционала: невозможно осуществление транзакций, и, как следствие, сложных операций с данными и отсутствие механизма блокировки. В зависимости от их значимости для телеком оператора, функционал может быть реализован

на стороне приложений, использующих базу данных. Однако это не меняет итогового решения по выбору СУБД в качестве базового решения задачи.

Литература

1. Minimum requirements related to technical performance for radio interface(s) : IMT-2020. – Изд. 2017
2. Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. – Москва : Вильямс, 2008. – 1328 с.
3. Клеппман, М. Высоконагруженные приложения. Программирование, масштабирование, поддержка / М. Клеппман. – Санкт Петербург : Питер, 2018. – 640 с.
4. Шапиро М. Бесконфликтные реплицированные типы данных / М. Шапиро, Н. Прегуиса, К. Бакеро // Lecture Notes in Computer Science. – 2011. – С. 386–402.

Ювченко Григорий Александрович — магистрант 2-го года обучения кафедры математических методов исследования операций Воронежского государственного университета. E-mail: gramlin@mail.ru

Ухлова Вера Владимировна (научный руководитель) — канд. физ.-мат. наук, доцент кафедры математических методов исследования операций Воронежского государственного университета. E-mail: Ukhlova@amm.vsu.ru

ПРОГНОЗИРОВАНИЕ С ПОМОЩЬЮ ПОСТРОЕНИЯ РЕГРЕССИОННЫХ МОДЕЛЕЙ

Е. В. Яньшина, Е. И. Самсонова, Е. М. Аристова

Воронежский государственный университет

Введение

Прогнозирование является процессом предопределения будущего на основании исходных параметров (опыта, выявленных закономерностей, тенденций, связей, возможных перспектив и т. п.). Оно используется в самых различных областях жизнедеятельности человека: экономике, социологии, демографии, политологии, метеорологии, генетике и др. Экономика является одной из наиболее распространённых разделов. В эконометрике применяется регрессионный анализ, который помогает найти зависимость результата от одного или нескольких факторов, то есть выявить взаимосвязи между неким событием с причиной его происхождения на основе статистических данных.

В данной статье будет рассмотрена задача, в которой прогнозируется стоимость акций компании Газпром в зависимости от цен на нефть. Для этого будет использоваться модель парной регрессии.

1. Парная регрессия

Простая регрессия — это модель, в которой среднее значение зависимой переменной y является функцией одной независимой переменной x .

Парная регрессия в неявном виде — это уравнение вида:

$$\hat{y} = f(x). \quad (1)$$

В явном виде: $\hat{y} = a + bx$, где a и b — параметры уравнения регрессии, ε — погрешность метода, которой можно пренебречь.

Парная регрессия применяется, если имеется доминирующий фактор, обуславливающий большую долю изменения изучаемой объясняемой переменной, который и используется в качестве объясняющей переменной.

Используя уравнение парной регрессии (1), соотношение между значениями переменными y и x (модель связи) можно записать как:

$$\hat{y} = f(x) + \varepsilon,$$

где первое слагаемое $f(x)$ можно интерпретировать как ту часть значения y , которая объяснена уравнением регрессии (1), а второе слагаемое ε как необъясненную часть значения y (или возмущение). Соотношение между этими частями характеризует качество уравнения регрессии, его способность представлять зависимость между переменными y и x [2].

При построении уравнения регрессии ε рассматривается, как ошибка модели, представляющая собой случайную величину, удовлетворяющую определенным предположениям. Наличие составляющей ε обусловлено такими причинами, как наличие дополнительных факторов, оказывающих влияние на переменную y , неверный вид функциональной зависимости $f(x)$ — ошибки спецификации, ошибки измерения, выборочный характер исходных данных.

Построение уравнения регрессии сводится к оценке её параметров. Для оценки параметров регрессий, линейных по параметрам, используют метод наименьших квадратов.

Рассмотрим следующие формулы, используемые для нахождения модели парной регрессии [1].

Если связь между признаками линейная, то система уравнений для нахождения параметров уравнения регрессии примет вид:

$$\begin{cases} na + b \sum_{i=1}^n x_i = \sum_{i=1}^n y_i, \\ a \sum_{i=1}^n x_i + b \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i. \end{cases}$$

Из неё можем выразить параметры уравнения:

$$a = \bar{y} - b\bar{x} \quad (2)$$

$$b = \frac{\overline{xy} - \bar{x} * \bar{y}}{x^2 - \bar{x}^2}. \quad (3)$$

Тесноту связи изучаемых явлений оценивает линейный коэффициент парной корреляции:

$$r_{xy} = b \frac{\sigma_x}{\sigma_y} = \frac{\overline{xy} - \bar{x} * \bar{y}}{\sigma_x * \sigma_y}, -1 \leq r_{xy} \leq 1.$$

Здесь $\sigma_x^2 = \overline{x^2} - \bar{x}^2$, $\sigma_y^2 = \overline{y^2} - \bar{y}^2$.

Оценку качества модели даёт коэффициент детерминации: $D = r^2$.

Оценка значимости уравнения регрессии в целом даётся с помощью F-критерия Фишера.

Выдвигается гипотеза H_0 о статистической незначимости уравнения регрессии и показателя тесноты связи. Для этого выполняется сравнение фактического $F_{\text{факт}}$ и критического (табличного) $F_{\text{табл}}$ значений F-критерия Фишера. $F_{\text{факт}}$ определяется из соотношений значений факторной и остаточной дисперсий, рассчитанных на одну степень свободы:

$$F_{\text{факт}} = \frac{r_{xy}^2}{1 - r_{xy}^2} (n - 2),$$

где n — это максимально возможное значение критерия под влиянием случайных факторов при данных степенях свободы.

Уровень значимости — вероятность отвергнуть правильную гипотезу при условии, что она верна. Обычно принимается равным 0.05 или 0.01.

Если $F_{\text{табл}} < F_{\text{факт}}$, то признаётся статистическая значимость уравнения, в целом. Иначе признаётся статистическая незначимость, ненадежность уравнения регрессии.

Оценка статистической значимости коэффициентов регрессии и корреляции производится с помощью *t-статистики Стьюдента*.

Выдвигается гипотеза H_0 о случайной природе показателей, то есть о статистически значимом отличии показателей от нуля. Оценка значимости коэффициентов регрессии и корреляции с помощью *t-статистики Стьюдента* приводится путем сопоставления их значений с величиной случайной ошибки:

$$t_a = \frac{a}{m_a},$$

$$t_b = \frac{b}{m_b},$$

$$t_r = \frac{r}{m_r}.$$

Случайные ошибки параметров линейной регрессии и коэффициента корреляции:

$$m_a = S_{\text{оцн}} \frac{\sqrt{\sum x^2}}{n\sigma_x},$$

$$m_b = \frac{S_{ocm}}{\sqrt{n}\sigma_x},$$

$$m_r = \sqrt{\frac{1-r_{xy}^2}{n-2}},$$

$$S_{ocm}^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n-2}.$$

Фактическое значение t-критерия Стьюдента сравнивается с табличным.

Если $t_{табл} < t_{факт}$, то нулевая гипотеза H_0 отклоняется, и параметры корреляции a , b , r неслучайно отличаются от нуля и сформировались под влиянием систематически действующего фактора x .

Если $t_{табл} > t_{факт}$, то нулевая гипотеза H_0 принимается, и признается случайная природа формирования a , b , r .

Точечный прогноз (y_p) — прогноз, рассчитанный путем подстановки в уравнение регрессии $\hat{y} = a + bx$ ожидаемого значения фактора x_p .

Для этого вычисляется стандартная ошибка y_p :

$$y_p - m_{\hat{y}_p} \leq y_p \leq y_p + m_{\hat{y}_p}.$$

При этом величина стандартной ошибки:

$$m_{\hat{y}_p} = S_{ocm} \sqrt{1 + \frac{1}{n} + \frac{(x_p - \bar{x})^2}{n\sigma_x^2}}. \quad (4)$$

2. Вычислительный эксперимент

В качестве примера рассмотрим акции российской транснациональной энергетической компании Газпром. Независимая переменная — стоимость фьючерсов на нефть (фьючерс — это ценная бумага в виде контракта на куплю-продажу определенного количества базового актива). Успех и прибыль компании зависит от цен на нефть. Построим регрессионную модель для определения стоимости акций компании Газпром в первую неделю января 2022 года.

Возьмем еженедельные данные за 2021 год — 52 наблюдения [3].

Таблица 1

Исходные данные

Дата	Фактическая цена	Фьючерсы на нефть (\$)	Дата	Фактическая цена	Фьючерсы на нефть (\$)
04.01.2021	225,49	56,16	05.07.2021	295,22	75,59
11.01.2021	226,50	54,82	12.07.2021	279,65	73,06
18.01.2021	215,49	55,25	19.07.2021	279,51	74,22
25.01.2021	212,83	55,88	26.07.2021	287,35	76,32
01.02.2021	222,01	59,51	02.08.2021	282,27	70,35
08.02.2021	226,71	62,65	09.08.2021	292,70	70,12
15.02.2021	226,00	62,71	16.08.2021	292,57	65,13
22.02.2021	217,56	66,03	23.08.2021	299,66	72,69
01.03.2021	227,00	69,64	30.08.2021	317,94	72,57
08.03.2021	233,80	69,23	06.09.2021	326,43	72,90

15.03.2021	225,40	64,45	13.09.2021	334,50	75,30
22.03.2021	227,30	64,43	20.09.2021	344,29	78,07
29.03.2021	228,40	64,83	27.09.2021	363,25	79,25
05.04.2021	222,15	63,07	04.10.2021	366,17	82,53
12.04.2021	235,10	66,74	11.10.2021	367,76	84,76
19.04.2021	232,62	66,01	18.10.2021	358,12	85,86
26.04.2021	231,38	67,29	25.10.2021	350,18	84,41
03.05.2021	241,70	68,28	01.11.2021	350,60	82,57
10.05.2021	249,55	68,54	08.11.2021	333,60	82,05
17.05.2021	258,00	66,63	15.11.2021	337,62	78,51
24.05.2021	262,00	69,07	22.11.2021	326,00	72,36
31.05.2021	274,30	71,68	29.11.2021	345,99	69,66
07.06.2021	277,05	72,60	06.12.2021	332,00	75,38
14.06.2021	270,06	73,27	13.12.2021	327,30	73,24
21.06.2021	277,19	76,17	20.12.2021	338,79	75,81
28.06.2021	285,65	76,11	27.12.2021	342,39	79,30

Ход решения

С помощью программы Excel произведем вычисления. Для начала получим значения $x * y$, x^2 , y^2 .

Таблица 2

Полученные данные

$x * y$	x^2	y^2	$x * y$	x^2	y^2
12663,52	3153,946	50845,7401	22315,68	5713,8481	87154,8484
12416,73	3005,2324	51302,25	20431,23	5337,7636	78204,1225
11905,82	3052,5625	46435,9401	20745,23	5508,6084	78125,8401
11892,94	3122,5744	45296,6089	21930,55	5824,7424	82570,0225
13211,82	3541,4401	49288,4401	19857,69	4949,1225	79676,3529
14203,38	3925,0225	51397,4241	20524,12	4916,8144	85673,29
14172,46	3932,5441	51076	19055,08	4241,9169	85597,2049
14365,49	4359,9609	47332,3536	21782,29	5283,8361	89796,1156
15808,28	4849,7296	51529	23072,91	5266,4049	101085,844
16185,97	4792,7929	54662,44	23796,75	5314,41	106556,545
14527,03	4153,8025	50805,16	25187,85	5670,09	111890,25
14644,94	4151,2249	51665,29	26878,72	6094,9249	118535,604
14807,17	4202,9289	52166,56	28787,56	6280,5625	131950,563
14011,00	3977,8249	49350,6225	30220,01	6811,2009	134080,469
15690,57	4454,2276	55272,01	31171,34	7184,2576	135247,418
15355,25	4357,3201	54112,0644	30748,18	7371,9396	128249,934
15569,56	4527,9441	53536,7044	29558,69	7125,0481	122626,032

16503,28	4662,1584	58418,89	28949,04	6817,8049	122920,36
17104,16	4697,7316	62275,2025	27371,88	6732,2025	111288,96
17190,54	4439,5569	66564	26506,55	6163,8201	113987,264
18096,34	4770,6649	68644	23589,36	5235,9696	106276
19661,82	5138,0224	75240,49	24101,66	4852,5156	119709,08
20113,83	5270,76	76756,7025	25026,16	5682,1444	110224
19787,30	5368,4929	72932,4036	23971,45	5364,0976	107125,29
21113,56	5801,8689	76834,2961	25683,67	5747,1561	114778,664
21740,82	5792,7321	81595,9225	27151,53	6288,49	117230,912

Через \bar{y} , \bar{x} , \overline{xy} , $\overline{x^2}$, $\overline{y^2}$ обозначим средние значения этих величин. Исходя из полученных данных в табл. 2, можем найти значения этих параметров.

$$\bar{y} = 14703,1, \quad \bar{x} = 3693,06, \quad \overline{xy} = 1061158,77, \quad \overline{x^2} = 265282,758, \quad \overline{y^2} = 4285897,501.$$

Теперь вычисляем коэффициенты a , b для нашей модели по формулам (2)–(3):

$$a = -118,22, \quad b = 5,6459.$$

Следовательно, модель задачи примет следующий вид: $\hat{y} = -118,22 + 5,6459x$.

Посчитаем значения \hat{y} при известных x :

Таблица 3

Модель парной регрессии

x	\hat{y}	x	\hat{y}	x	\hat{y}	x	\hat{y}
56,16	198,8515	63,07	237,8648	75,59	308,5516	82,53	347,7342
54,82	191,286	66,74	258,5853	73,06	294,2674	84,76	360,3246
55,25	193,7137	66,01	254,4637	74,22	300,8167	85,86	366,5351
55,88	197,2707	67,29	261,6905	76,32	312,6731	84,41	358,3485
59,51	217,7653	68,28	267,28	70,35	278,967	82,57	347,96
62,65	235,4935	68,54	268,7479	70,12	277,6684	82,05	345,0241
62,71	235,8322	66,63	257,9642	65,13	249,4953	78,51	325,0376
66,03	254,5767	69,07	271,7402	72,69	292,1784	72,36	290,3153
69,64	274,9584	71,68	286,4761	72,57	291,5009	69,66	275,0713
69,23	272,6436	72,60	291,6703	72,90	293,3641	75,38	307,3659
64,45	245,6561	73,27	295,4531	75,30	306,9142	73,24	295,2837
64,43	245,5432	76,17	311,8262	78,07	322,5534	75,81	309,7937
64,83	247,8016	76,11	311,4874	79,25	329,2156	79,30	329,4979

Для того чтобы проверить качество полученной модели, найдем коэффициенты детерминации и среднюю ошибку аппроксимации.

Вычислим параметры $\sigma_x = 7,6$, $\sigma_y = 49,72$, а затем коэффициенты корреляции и детерминации: $r_{xy} = 0,862470664$, $D = 0,7439$.

Теперь найдем среднюю ошибку аппроксимации по формуле

$$\bar{A} = \frac{1}{n} \sum \left| \frac{y_i - \hat{y}}{y_i} \right| * 100 \%,$$

получим $A = 7,89 \%$.

Как можно увидеть, коэффициент детерминации близок к 1, и средняя ошибка аппроксимации находится в промежутке 7–8 %, это означает, что данная модель является корректной.

Воспользуемся критерием Фишера и выдвинем гипотезу о статистической незначимости уравнения регрессии и показателя тесноты. Получим $F_{\text{факт}}$: $F_{\text{факт}} = 145,2024288$. При этом $F_{\text{табл}} = 4,03$ при исходных данных $\alpha = 0,05$, $k = 52$. Так как $F_{\text{табл}} < F_{\text{факт}}$, то уравнение статистически значимо.

Теперь, с помощью t-статистики Стьюдента, проведем оценку значимости коэффициентов регрессии и корреляции. Выдвинем гипотезу о случайной природе показателей, то есть о статистически незначимом отличии показателей от нуля. Рассчитаем случайные ошибки параметров линейной регрессии и коэффициента корреляции:

$$m_a = 0,6436, m_b = 0,065, m_r = 0,071574, S_{\text{ост}}^2 = 12,6661.$$

Найдем показатели $t_{\text{факт}}$:

$$t_a = -183,6981, t_b = 86,89, t_r = 12,05.$$

При этом $t_{\text{табл}} = 2,009$, при исходных данных $\alpha = 0,05$, $k_1 = 10$, $k_2 = 52$.

В результате по полученным данным мы можем сделать вывод, что для каждого параметра $t_{\text{табл}} < t_{\text{факт}}$, таким образом, выдвинутая гипотеза отклоняется, наши параметры неслучайно отличаются от 0 и сформировались под влиянием систематически действующего фактора x .

Построим прогноз стоимости акций компании Газпром на первую неделю января (03.01.2022), в зависимости от стоимости фьючерса на нефть x_p по данным финансового портала Финам [3]:

$$x_p = 81,95, \text{ при этом получим, что } y_p = 344,4596.$$

Теперь проведем точечный прогноз, для этого вычислим стандартную ошибку y_p по формуле (4):

$$m_{\hat{y}} = 3,6625.$$

$$\text{Видим, что } y_p - m_{\hat{y}} \leq y_p \leq y_p + m_{\hat{y}} \quad (340,797 < 344,4596 < 348,1221).$$

Прогноз является корректным.

Заключение

В ходе данной работы, мы построили приемлемую и статистически значимую регрессионную модель. Чтобы убедиться в этом, сравним фактическую стоимость акций компании Газпром на 3 января 2022 года с полученным прогнозом. Исходя из данных финансового портала Финам [3], фактическая стоимость составила 346,13 рублей, что удовлетворяет полученному результату, учитывая стандартную ошибку. Таким образом, прогноз является корректным.

Литература

1. Аристова, Е. М. Модели методы принятия решений: учебное пособие / Е. М. Аристова. – Воронеж : Издательский дом ВГУ, 2020. – 64 с.
2. Удинцова, Н. М. Эконометрика. Часть 1. Парная регрессия и корреляция в эконометрических исследованиях: учебное пособие / Н. М. Удинцова, Н. А. Коптева. – зерноград : Азово-Черноморский инженерный институт ФГБОУ ВО Донской ГАУ, 2015. – 61 с.
3. Финам.ру – финансовый портал: новости фондового рынка ценных бумаг, информация и аналитика. – URL: <https://www.finam.ru/> (дата обращения: 15.04.2022).

Яньшина Ева Владимировна — студентка 4-го курса кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: yansh.ev@mail.com

Самсонова Екатерина Ивановна — студентка 4-го курса кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: katty.samsonovaa@yandex.ru

Аристова Екатерина Михайловна (научный руководитель) — канд. физ.-мат. наук, доц., доцент кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: pmim@yandex.ru

КОМПЬЮТЕРНОЕ ЗРЕНИЕ И МАТЕМАТИЧЕСКАЯ ОПТИМИЗАЦИЯ В ЗАДАЧАХ ИДЕНТИФИКАЦИИ ПРОСТРАНСТВЕННОГО ПОЛОЖЕНИЯ ОБЪЕКТА

П. О. Ярощук

Воронежский государственный университет

Введение

Компьютерное зрение часто требует продвинутого программного и аппаратного обеспечения. Также желательно наличие набора данных большого размера, создание которого занимает многие человеко-часы. При этом в случае с нейронными сетями результат обучения не всегда определен. Все это ограничивает область распространения компьютерного зрения, например, в некоммерческих научных исследованиях.

В данной работе представлена попытка создания метода, который не требовал бы значительных аппаратных ресурсов, времени на создание набора данных, а также достаточно точно определял положения жестких объектов по данным всего одной камеры. Для реализации использован комбинированный подход, включающий обучение по компьютерной симуляции, инструменты двумерного компьютерного зрения и математическую оптимизацию.

1. Постановка задачи и общее описание метода

Задача заключается в определении положения объекта в трехмерном пространстве по следующим данным: изображению объекта, полученному с камеры, типу объекта (его трехмерной модели) и положению ключевых ориентиров (визуально заметных особенностей, которые перемещаются вместе с объектом).

Для определения положения жесткого объекта по двумерному изображению необходимо применить алгоритм ригидной (жесткой) регистрации объекта с применением нелинейной оптимизации. Общий алгоритм следующий:

1. Нейронная сеть предварительно обучается распознавать объекты и его ключевые точки.
2. Сеть определяет вид объекта и отмечает координаты его ключевых точек на изображении. Подгружается модель объекта, повторяющая геометрию оригинала.
3. Геометрия объекта выравнивается так, чтобы все ключевые точки модели совпадали.

При реализации используются такие средства и технологии, как QTCreator, C++, Eigen, YOLOv4, Darknet Framework.

2. Подробное описание метода

Обучение с помощью компьютерной симуляции. Нейронную сеть, корректно выделяющую ориентиры на объекте, необходимо предварительно обучить. Обучение — трудный и малопредсказуемый процесс, который можно упростить, беря во внимание специфику задачи. В нашем случае сеть должна научиться определять ключевые точки объекта при любом его повороте. Чтобы упростить создание набора данных, достаточно отрисовать сцену с необходимыми объектами, отметив полигоны ключевых точек. Кроме того, единожды отметив эти точки, программа будет знать об их положении при любом повороте и переносе.

Отметив ключевые точки всего на 10–15 моделях, можно сразу же создать набор данных из 600–900 изображений, состоящий из различных пространственных положений объекта с от-

меченными на них видимыми ориентирами, что и является необходимым и достаточным для дальнейшего обучения нейросети. Кроме того, располагая современными инструментами фотограмметрии, можно также упростить создание моделей, используя набор фотографий требуемых объектов и программу Reality Capture, которая и преобразует их в трехмерные модели.

Выделение ключевых ориентиров. Используя нейросеть архитектуры YOLOv4 на платформе Darknet Framework, можно обнаруживать на изображении множественные объекты и ориентиры за один проход. Это происходит благодаря особой методике детектирования, которая ставит в соответствие объекту X якорь с рамкой, при этом якоря могут быть одновременно расставлены на все пиксели. Подобный метод позволяет одновременно отметить и сам объект, и его ключевые точки, которые в этом случае будут определяться положением якоря соответствующей рамки. Например, на рис. 1 объект отмечен зеленой рамкой (класс автомобиль), ключевые точки отмечены красными рамками (при этом могут иметь разные классы — фары, двери, шины, радиатор).



Рис. 1. Схематическое изображение объекта и его ключевых точек

Используя полученные якоря ключевых точек, мы можем использовать их для сопоставления точкам трехмерной модели, однако сначала отмеченные точки нужно преобразовать по базису в мировые координаты. Для этого мы используем обратные матрицы вида, проекции и камеры. Матрица камеры представляет собой умножение матрицы смещения и поворота камеры относительно мировых координат (рис. 2). Задание матрицы проекции представлено на рис. 3. Умножение на матрицу вида, по сути, представляет собой умножение координат на горизонтальное и вертикальное разрешение экрана.

```
QMatrix4x4 Camera::cameraMatrix()  
{  
    QMatrix4x4 output;  
    output.translate(-m_origin);  
    output *= QMatrix4x4(m_rotation.normalized().conjugated().toRotationMatrix());  
    output.translate(-m_target);  
    return output;  
}
```

Рис. 2. Задание матрицы камеры

```

QMatrix4x4 Camera::perspectiveMatrix(const double vFovRad, const double aspectRatio, const double fPlane, const double nPlane)
{
    const double near = nPlane;
    const double far = fPlane;
    const double right = tan(atan(tan(vFovRad) * aspectRatio) / 2);
    const double left = -right;
    const double top = tan(vFovRad / 2);
    const double bottom = -top;

    return QMatrix4x4(2 / (right - left), 0, (right + left) / (right - left), 0,
                     0, 2 / (top - bottom), (top + bottom) / (top - bottom), 0,
                     0, 0, -(far + near) / (far - near), -2 * near * far / (far - near),
                     0, 0, -1, 0);
}

```

Рис. 3. Задание матрицы проекции

Произведя обратные операции, мы найдем координаты (x_w, y_w) , соответствующие положению точек в пространстве трехмерного объекта, видимых на изображении, но без учета глубины (z_w). С этими точками и осуществляется дальнейшая работа.

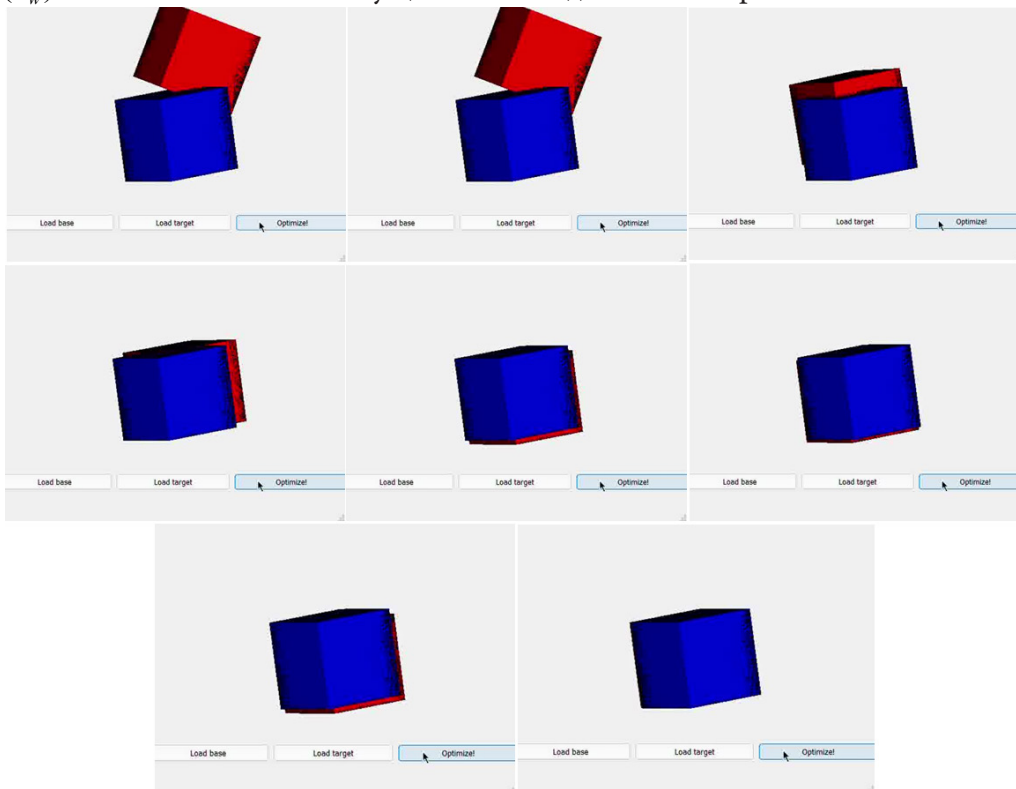


Рис. 4. Этапы оптимизации по ключевым точкам на примере кубов

Математическая оптимизация. Выравнивание автомобиля — набор функций трех переменных $F : R^3 \rightarrow R^3$, принимающая координаты исходных точек модели, возвращающая новые координаты и имеющая в общем случае следующий вид:

$$F(x, y, z) = translation(\Delta x, \Delta y, \Delta z) \cdot rotation(\alpha, \beta, \gamma) \cdot (x, y, z),$$

где *rotation* имеет 3 определяемых параметра — углы Эйлера (крен, тангаж и рысканье), в зависимости от которых точки поворачиваются относительно заданной точки, а *translation* — вектор переноса модели. Эти параметры и вычисляются в ходе математической оптимизации.

В качестве целевой функции используется набор невязок вида

$$(x_{base}^k, y_{base}^k, z_{base}^k) - (x_{target}, y_{target}, z_{target}) \rightarrow \min,$$

где $(x_{base}^k, y_{base}^k, z_{base}^k) = F(x_{base}^{k-1}, y_{base}^{k-1}, z_{base}^{k-1})$.

Данный набор позволяет однозначно выровнять объект в каждой точке пространства, однако мы не располагаем координатой z . В связи с этим невязки принимают вид

$$(x_{base}^k, y_{base}^k) - (x_{target}, y_{target}) \rightarrow \min.$$

Тем не менее расчеты (x_{base}^k, y_{base}^k) все равно производятся по трем координатам, просто z отбрасывается. Так как количество невязок меньше необходимого числа переменных, задача имеет несколько решений, однако на практике показывает хорошие результаты и сходимость (см. рис. 4).

Заключение

Предложенный в работе метод позволит упростить задачу выравнивания положения объекта на основе ключевых характеристик, при этом обладая реализацией быстрого симулятивного обучения и не требуя большой вычислительной мощности (желательно наличие графического ускорителя). При этом исходные данные представляют собой изображение единственной камеры (не стереопары) без дополнительных датчиков глубины. Все это позволяет применить алгоритм в задачах компьютерного зрения, например, в задачи автоматического пилотирования транспорта.

Работа выполнена под руководством канд. физ.-мат. наук, доцента кафедры вычислительной математики и прикладных информационных технологий ВГУ Королькова Олега Геннадьевича.

Литература

1. QT Documentation. – Режим доступа: <https://doc.qt.io>.
2. *Bochkovskiy A.* YOLOv4: Optimal Speed and Accuracy of object detection / A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao // arxiv.org. – 2020.
3. *Yinda Zh.* Physically-Based Rendering for Indoor Scene Understanding Using Convolutional Neural Networks / Zh. Yinda, S. Shuran, Y. Ersin, S. Manolis, L. Joon-Young, J. Hailin, T. Funkhouser // arxiv.org. – 2017.
4. *Коннов И. В.* Нелинейная оптимизация и вариационные неравенства / И. В. Коннов. – Казань : Казанский ун-т, 2013. – 508 с.

Ярошук Павел Олегович — магистрант 2-го года обучения кафедры вычислительной математики и прикладных информационных технологий Воронежского государственного университета. E-mail: pay.pray.spray@gmail.com

Содержание

<i>Блиняева А. Ю.</i> Определение диапазона углов для четырехзвенного манипулятора.....	3
<i>Боль Я. А.</i> Применение классов типов, алгебраических структур и концепций теории категорий в реализации полиморфных методов на языке программирования Scala	7
<i>Бондаренко Ю. В., Будаева С. Ю.</i> Прогнозирование складских запасов для товаров с прерывистым спросом модифицированным бутстреп-методом	13
<i>Бухольцев И. М.</i> Автоматизация ребалансировки портфеля ценных бумаг как способ защиты от факторов, влияющих на фондовый рынок.....	20
<i>Вернер Е. С.</i> Алгоритм работы искусственной иммунной системы	23
<i>Воронова О. С.</i> Использование пиктографиков Чернова в качестве инструмента анализа многомерных данных.....	27
<i>Дорофеева С. Р.</i> Качественное исследование математических моделей динамики развития малого предприятия.....	33
<i>Дроботенко А. Д.</i> Статистический анализ динамики демографических популяций	40
<i>Дыбов Е. И., Бондаренко Ю. В.</i> Математические методы поддержки управления финансами объединения экономических агентов при реализации совместных проектов	46
<i>Еникеева Л. И.</i> Количество арифметических операций, необходимых для обращения треугольной матрицы	53
<i>Жидких А. В.</i> Разработка программного обеспечения, реализующего методы обработки изображений для анализа фаций крови	58
<i>Карякин Н. Н., Щепина И. Н.</i> Метод стохастической границы при оценке эффективности производственного потенциала	63
<i>Каширских О. А.</i> Применение алгоритмов нейронных сетей для анализа эмоциональной окраски текста	67
<i>Копылов М. В., Татаренков Е. А., Дюбин Р. С., Голиков М. В.</i> Математическая модель движения сырья в межвитковом пространстве шнекового оборудования.....	71
<i>Коток И. Д.</i> Модификации приложения по проведению интеллектуальных викторин	74
<i>Крыжановский В. Д.</i> Эффективность применения нейронной сети в задаче классификации кровоподтеков	78
<i>Лихачев М. Ю., Замятин И. В.</i> Анализ аудиоданных. Распознавание мелодии при помощи ИНС-модели обработки данных	83
<i>Ломакин М. А.</i> Построение интеграционной модели автоматизированного тестирования с использованием нейронных сетей.....	90
<i>Медведева Д. Ю., Щепина И. Н.</i> Моделирование диффузии инноваций на примере межстранового анализа распространения Интернета	93
<i>Минаков В. И., Золотарёв С. В.</i> Модель BERT для обработки естественного языка	100
<i>Муратова Н. В.</i> Рекомендательные системы: определение, задачи, архитектура	105
<i>Наумов М. С., Бондаренко Ю. В.</i> Математическая и программная поддержка формирования оптимального комплекта программного обеспечения учебного процесса вуза в условиях импортозамещения	110

<i>Овчинникова А. С., Абрамов Г. В. Мобильный бэк-офис «Z-office»</i>	<i>116</i>
<i>Орешин К. В., Шевцов А. А., Сердюкова Н. А. Численно-аналитическое решение дифференциальных уравнений тепловой обработки семян масличных культур высокотемпературным теплоносителем</i>	<i>122</i>
<i>Палкина С. А. Постановка задачи моделирования процесса управления функционированием городских систем водоснабжения</i>	<i>12</i>
<i>Панков И. А., Юнда Д. Р., Шуба А. В. Распределение поляризации в сферической сегнетоэлектрической наночастице</i>	<i>131</i>
<i>Перепелов Я. Д. Модель потока идеальной жидкости в окрестности цилиндра с отверстиями</i>	<i>137</i>
<i>Петрова Д. А., Каширина И. Л. Проведение интеллектуального анализа данных для увеличения точности модели машинного обучения</i>	<i>142</i>
<i>Петухов И. И. Сравнительное исследование алгоритмов муравьиной колонии и пчелиного роя для построения оптимального кольцевого маршрута.</i>	<i>151</i>
<i>Попов А. И. Разработка системы хранилища документов с использованием компьютерного зрения</i>	<i>155</i>
<i>Пукась М. Ю., Золотарев С. В. Блокчейн</i>	<i>159</i>
<i>Святоха Ж. А. Обзор методов моделирования распространения заболеваний</i>	<i>163</i>
<i>Сотников А. Д. Постановка оптимизационной задачи формирования ассортимента склада</i>	<i>171</i>
<i>Стратиенко С. В. Анализ существующих подходов к разработке приложений дополненной реальности на примере помощника автолюбителю</i>	<i>175</i>
<i>Строгонов А. И. Модернизация LDA-алгоритма в задачах информационного поиска</i>	<i>183</i>
<i>Строгонов В. Г. Краткий обзор протоколов HTTP/1.1 и HTTP/2 и сравнение их производительности</i>	<i>188</i>
<i>Струкова Е. А. Моделирование транспортных систем в среде AnyLogic средствами Библиотеки дорожного движения</i>	<i>192</i>
<i>Суркова П. И., Белоусова Е. П. Дискретная модель задачи управления запасами</i>	<i>198</i>
<i>Сухоручкина А. И., Аристова Е. М. Построение рейтинга объектов</i>	<i>203</i>
<i>Сырых А.С. Основы фотограмметрии</i>	<i>212</i>
<i>Ткачев О. А., Чернопятова С. А., Чернецкий З. П., Науменко И. Н. Датчик оптической плотности</i>	<i>217</i>
<i>Третьякова Д. И. Архитектура и функциональность CRM-решений для малого бизнеса</i>	<i>220</i>
<i>Тюлюш С. Т. Обратный алгоритм процедур диагностирования дискретных объектов</i>	<i>225</i>
<i>Усков Д. Г. Из истории языков программирования</i>	<i>233</i>
<i>Усков Д. Г. Разработка алгоритма обработки осциллометрических данных кардиологических больных</i>	<i>236</i>
<i>Чеботарев У. В. Разработка программного обеспечения развлекательной компьютерной игры для школьников</i>	<i>239</i>

<i>Черненко Д. Г.</i> Оптимальное управление линейной динамической системой	244
<i>Черняев Д. Е., Енокян В. А., Морозов Д. А., Шатковская А.В., Новоскольцев И. Ю., Петрова В. Г.</i> Трехмерная реконструкция медицинских изображений	250
<i>Шерстюк Д. В., Уклова В. В.</i> Проблемы подготовки данных для сервисов визуализации платформ VI.....	258
<i>Шишкалов В. А.</i> Тепловая конвекция в квадратной области тепловыделяющей жидкости...	263
<i>Щеглеватых А. А., Аристова Е. М.</i> Реализация приложения для распознавания лица, возраста и гендера в реальном времени	266
<i>Ювченко Г. А.</i> Выбор СУБД для реализации системы онлайн тарификации телекоммуникационных услуг.....	272
<i>Яньшина Е. В., Самсонова Е. И., Аристова Е. М.</i> Прогнозирование с помощью построения регрессионных моделей.....	278
<i>Ярошук П. О.</i> Компьютерное зрение и математическая оптимизация в задачах идентификации пространственного положения объекта.....	285

Научное издание

Математика,
информационные технологии,
приложения

*Сборник трудов
Межвузовской научной конференции
молодых ученых и студентов*

Воронеж,
27 апреля 2022 г.

Подписано в печать 31.05.2022. Формат 60 × 84 / 8.
Усл. печ. л. 33,95. Тираж 100 экз. Заказ № ***.

Отпечатано с готового оригинал-макета

ООО Издательско-полиграфический центр
«Научная книга»
394030, г. Воронеж, ул. Никитинская, д. 38, оф. 308
Тел. +7 (473) 200-81-02, 200-81-04
<http://www.n-kniga.ru>; e-mail: zakaz@n-kniga.ru

Отпечатано в типографии ООО ИПЦ «Научная книга».
394026, г. Воронеж, Московский пр-т, 116
Тел. +7 (473) 220-57-15, 238-02-38
<http://www.n-kniga.ru>; e-mail: typ@n-kniga.ru