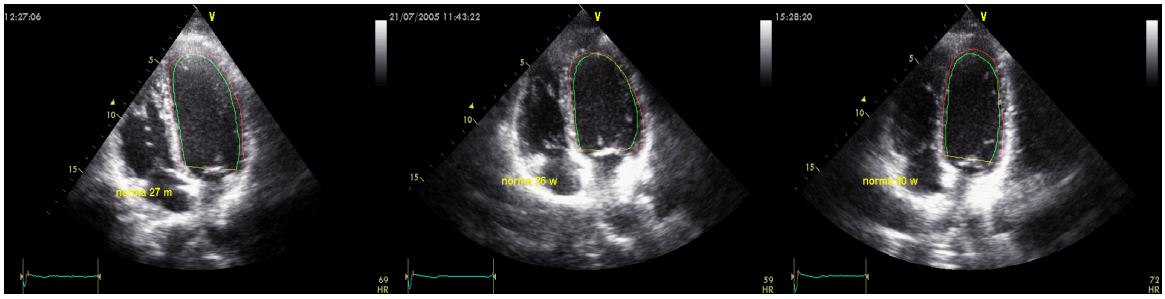
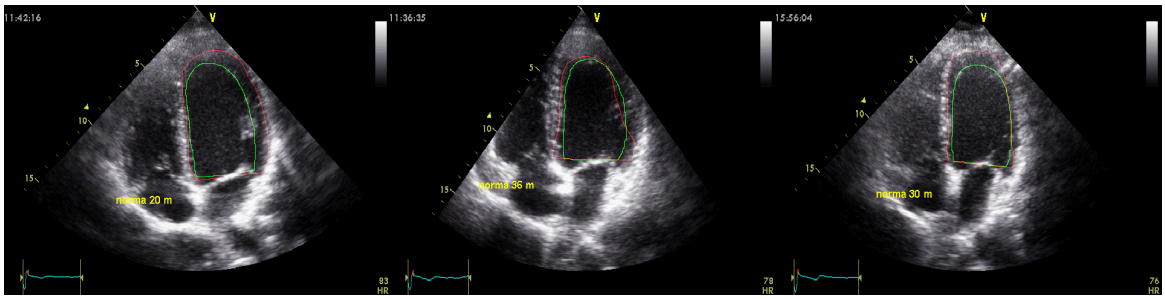
ISSN 2223-3792

Машинное обучение и анализ данных

2015 год **Том 1, номер 11**





Машинное обучение и анализ данных

Journal of Machine Learning and Data Analysis

Журнал «Машинное обучение и анализ данных» публикует новые теоретические и обзорные статьи с результатами научных исследований в области теоретических основ информатики и ее приложений. Цель журнала — развитие теории машинного обучения, интеллектуального анализа данных и методов проведения вычислительных экспериментов. Принимаются статьи на английском и русском языках.

Журнал включен в российский индекс научного цитирования РИНЦ. Информация о цитировании статей находится на сайте Российского индекса научного цитирования, ISSN 2223-3792, номер свидетельства о регистрации ЭЛ № ФС 77-55486.

- Архив журнала http://www.ccas.ru/jmlda/
- Новостной сайт http://jmlda.org/
- Электронная система подачи статей http://jmlda.org/papers/

Тематика журнала:
— классификация, кластеризация, регрессионный анализ;
— алгебраический подход к проблеме синтеза корректных алгоритмов;
— многомерный статистический анализ;
— выбор моделей и сложность;
— предсказательное моделирование;
— статистическая теория обучения;
— методы прогнозирования временных рядов;
— методы обработки и распознавания сигналов;
— методы оптимизации в задачах машинного обучения и анализа данных;
— методы визуализации данных;
— обработка и распознавание речи и изображений;
— анализ и понимание текста;
— информационный поиск;
— прикладные задачи анализа данных.

Вычислительный центр Российской академии наук
Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

Москва, 2015

Содержание

Алгоритм классификации временных рядов акселерометра по комбинированному признаковому описанию*

М. П. Кузнецов, Н. П. Ивкин
mikhail.kuznecov@phystech.edu, ivkinnikita@gmail.com
Московский физико-технический институт, Москва

Рассматривается задача многоклассовой классификации временных рядов. Временные ряды являются объектами сложной структуры, для которых не задано исходное признаковое описание. Исследуются различные методы построения признакового пространства для временного ряда: метод экспертного задания порождающих функций и метод построения признаков на основе гипотезы порождения данных. Рассматривается комбинированное признаковое описание временного ряда. В качестве прикладной задачи рассматривается задача классификации данных акселерометра. Показано, что использование расширенного множества признаков приводит к значительному улучшению качества классификации.

Ключевые слова: *временные ряды; авторегрессионная модель; сингулярный спектр; метод опорных векторов*

Time series classification algorithm using combined feature description*

M. P. Kuznetsov and N. P. Ivkin
Moscow Institute of Physics and Technology, Moscow, Russia

A problem of time series multiclass classification is considered. Time series are regarded as complex-structured objects having no explicit feature description. In general, complex objects classification problem can be divided into two stages: to form feature space and to construct a decision rule on this space. The focus of the paper is on the first stage, namely, to construct the feature space where the points of different classes are separable or close to separable. Having constructed such space, a simple linear or polynomial decision rule is used to discriminate the classes.

Various methods of time series feature space construction are investigated. The first method is the expert definition of basic functions. Namely, for the time series data, mean value, deviation, absolute deviation, and empirical distribution of values are considered. The second method involves data generation hypothesis and uses optimal estimations of generation parameters as the considered features. Furthermore, a combined feature description of a time series is considered. The computations show that using the extended feature space allows to significantly improve the classification quality.

The proposed approach is used for the accelerometer time series classification. The problem is to classify each time series segment to one of six classes-actions: Jogging, Walking, Upstairs, Downstairs, Sitting, and Standing. It is shown that the combined approach achieves very good accuracy comparing with the separate feature construction methods.

Keywords: *time series; autoregressive model; singular spectrum; support vector machine*

*Работа выполнена при финансовой поддержке РФФИ, проект 15-37-50324 мол_нр.

1 Введение

Решается задача классификации объектов сложной структуры, т. е. таких объектов, для которых не сформировано исходное признаковое описание и матрица плана. Объекты подобного типа возникают во многих задачах анализа данных: распознавание объектов на изображении, классификация звуковых сигналов, тематическое моделирование. В данной работе рассматривается задача классификация одномерных и многомерных временных рядов произвольной, возможно, различной длины [1, 2]. В этих терминах временной ряд является объектом сложной структуры: значения временного ряда, соответствующие различным временным отсчетам, не могут рассматриваться в качестве признакового описания объекта.

В качестве прикладной задачи классификации временных рядов рассматривается задача классификация данных акселерометра [3]. Данные представляют собой измерения акселерометра некоторого устройства, например мобильного телефона в кармане человека, и могут использоваться для идентификации действия человека в каждый момент времени. Примером действия служат ходьба, бег, подъем по лестнице и др. Ранее задача распознавания данных акселерометра для определения типа действия человека также была поставлена в форме задачи онлайн сегментации временного ряда [4]. В отличие от задачи сегментации в данной работе исследуются предобработанные данные: размеченные временные ряды, соответствующие различным действиям. Решение задачи классификации в такой постановке позволяет выделить некоторые характерные признаки, которые могут быть использованы в дальнейших работах.

В общем случае задача классификации объектов сложной структуры разделяется на два этапа. На первом этапе формируется признаковое описание объектов, на втором этапе строится решающее правило классификации. Отметим, что эти два этапа, вообще говоря, зависимы друг от друга, однако в простейшем случае они могут рассматриваться отдельно. В данной работе основной упор сделан на построение признакового пространства, наиболее полным образом описывающего выделяемые классы действия; классификаторы на втором этапе ищутся в классе линейных или низкой степени полиномиальных решающих правил. Такая мотивация следует из отделяемости этапа построения признакового пространства от задачи классификации. Построенные признаки, если они описывают целевую переменную достаточно адекватно, предлагается использовать в последующих задачах сегментации и онлайн обучения.

В работе рассматриваются два основных метода построения признакового пространства. Первый метод заключается в экспертном назначении базовых функций. В [5] выделяют такие базовые функции, как среднее временного ряда, стандартное отклонение, среднее расстояние между пиками, распределение значений временного ряда. Воспользуемся этим методом в качестве получения базового признакового пространства и результатов классификации.

Второй метод построения признакового пространства заключается в том, что назначается параметрическая гипотеза порождения объекта сложной структуры. В частности, для временного ряда в качестве такой гипотезы рассматривается модель авторегрессии, дополняемая анализом условной гетероскедастичности [6]. Для каждого объекта — временного ряда — вычисляются оптимальные параметры порождения, в частном случае коэффициенты авторегрессии; эти вычисленные параметры составляют новое признаковое пространство [7, 8]. Таким образом, процедура классификации выполняет разбиение пространства параметров модели на области, принадлежащие различным классам. Помимо авторегрессионной модели исследуется также модель сингулярного спектра временно-

го ряда, где выделяемыми признаками являются собственные числа траекторной матрицы [9].

Структура работы организована следующим образом. В первом разделе поставлена задача классификации объектов сложной структуры и выделения признаков в общем смысле. Во втором разделе рассмотрена задача классификации временных рядов, приводится конкретный вид гипотез порождения данных для выделения признаков на основе оценки оптимальных параметров порождения. В третьем разделе рассматривается задача классификации данных акселерометра, описывается процедура ручной генерации признаков. Раздел вычислительного эксперимента выполняет сравнение предлагаемых методов выделения признаков. Оказывается, что различные методы помогают лучше выделять отдельные классы. Окончательный эксперимент использует полное признаковое описание, заключающее в себе все предлагаемые методы выделения признаков.

2 Классификация объектов сложной структуры

2.1 Постановка задачи классификации

Пусть $s \in \mathcal{S}$ — объект сложной структуры. Рассматривается задача восстановления зависимости

$$y = f(s),$$

где функция f отображает пространство объектов сложной структуры $\mathcal{S}$ в пространство ответов Y, $y \in Y$.

Задана выборка $\mathfrak{D}$ объектов сложной структуры и ответов:

$$\mathfrak{D} = \{(s_i, y_i)\}_{i=1}^{m}.$$

Задана функция потерь $l(f(s_i), y_i)$, выражающая величину ошибки классификации функции f на объекте s_i выборки $\mathfrak{D}$. Требуется найти функцию f, минимизирующую суммарные потери на выборке $\mathfrak{D}$.

2.2 Пространство признаков объекта сложной структуры

Отображение f будем рассматривать в классе суперпозиций $f = g \circ h$ таких, что

$$f(s) = g(h(s), \mathbf{b}),$$

где

$$h(s) : \mathcal{S} \to \Theta \subset \mathbb{R}^n$$

является отображением пространства $\mathcal{S}$ в признаковое пространство $\Theta \subset \mathbb{R}^n$, а $g(\mathbf{h}, \mathbf{b})$ является параметрическим отображением пространства Θ в пространство ответов Y.

Через $\hat{y}_i$ обозначим значение функции f на объекте s_i:

$$\hat{y}_i = f(s_i) = g(\mathbf{h}_i, \mathbf{b}),$$

где $h_{ij} \equiv h_j(s_i)$ является j-й компонентой значения вектор-функции h на объекте s_i. Значение h_{ij} будем называть j-м признаком объекта s_i, или j-й статистикой.

В данной работе рассматриваются следующие способы построения признакового пространства h_{ij}:

– способ ручной генерации признаков. Такой способ использует экспертную информацию о структуре сложного объекта. Например, в рассматриваемой задаче классификации временных рядов признаками могут быть среднее значение временного ряда на отрезке, среднеквадратичное отклонение, максимальное значение и другие статистики;

– способ введения гипотезы порождения объекта сложной структуры. В этом случае статистики h_{ij} являются оценками параметров рассматриваемой гипотезы. Подробнее об этом методе будет рассказано в следующем параграфе.

2.3 Гипотеза порождения объекта сложной структуры

Будем полагать, что объект сложной структуры s_i представляет собой множество реализаций объектов «простой структуры»:

$$s_i = \{x_{i1}, ..., x_{it(i)}\},$$

где x_{it} является реализацией случайной величины $X_i \sim P_{\boldsymbol{\theta}_i}$, $\boldsymbol{\theta}_i \in \Theta \subset \mathbb{R}^n$.

Будем полагать, что задана функция ошибки $S(s_i, \boldsymbol{\theta}, \boldsymbol{\lambda})$, имеющая одинаковый вид для всех объектов сложной структуры s_i. Эта функция ошибки может являться, например, функцией правдоподобия данных X_i и служит для определения оптимальных параметров $\hat{\boldsymbol{\theta}}_i$ для объекта s_i:

$$\hat{\boldsymbol{\theta}}_i = \arg\min_{\boldsymbol{\theta}\in\Theta} S(s_i, \boldsymbol{\theta}, \boldsymbol{\lambda}). \tag{1}$$

Вектор $\boldsymbol{\lambda}$ является вектором внешних для функции ошибки S параметров. Будем называть $\boldsymbol{\lambda}$ вектором *структурных параметров*.

В качестве нового признакового описания h_{ij} объекта s_i будем рассматривать оптимальные значения вектора параметров $\hat{\boldsymbol{\theta}}_i$:

$$\mathbf{h}_i \equiv \hat{\boldsymbol{\theta}}_i(s_i).$$

Таким образом, для задачи классификации осуществляется разбиение пространства параметров Θ порождения объектов сложной структуры s на множества принадлежности меткам классов Y.

Частный вид гипотез порождения данных, структурных параметров и функции ошибки для задачи классификации временных рядов будет приведен в следующих разделах. Например, гипотезой порождения будет служить авторегрессионная модель порождения, структурным параметром — длина авторегрессионной модели, а функцией ошибки — качество прогнозирования временного ряда.

2.4 Определение оптимальных параметров

Согласно вышеизложенному, задача определения оптимальных параметров проводится в два этапа.

1. Построение признакового пространства h_{ij} объекта сложной структуры s_i. Статистики $\mathbf{h}_i$ могут быть построены вручную экспертным путем, либо путем минимизации функции ошибки (1) для гипотезы порождения объекта s_i:

$$\mathbf{h}_i \equiv \hat{\boldsymbol{\theta}}_i = \arg\min_{\boldsymbol{\theta}\in\Theta} S(s_i, \boldsymbol{\theta}, \boldsymbol{\lambda}).$$

2. Определение оптимальных параметров $\hat{\mathbf{b}}$ в задаче классификации объектов $(\mathbf{h}_i, y_i)_{i=1}^m$ выборки $\mathfrak{D}$ в новом признаковом пространстве,

$$\hat{y}_i = g(\mathbf{h}_i, \hat{\mathbf{b}}),$$

путем минимизации функции ошибки:

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} L(g, \mathbf{b}, \boldsymbol{\lambda}, \mathfrak{D}). \tag{2}$$

3 Многоклассовая классификация одномерных временных рядов

В качестве частного случая объектов сложной структуры в данной работе будут рассматриваться одномерные (а далее — и многомерные) временные ряды переменной длины. Объектом s_i в данном случае будет являться последовательность

$$s_i = \{x_1, ..., x_{T(i)}\},$$

где длина временного ряда $T(i)$ является переменной и зависит от индекса i.

Задана выборка $\mathfrak{D} = \{(s_i, y_i)\}_{i=1}^m$, где s_i — временной ряд, y_i — метка класса. Будем строить отображение временных рядов в пространство меток классов Y в виде классификации параметров $\hat{\boldsymbol{\theta}}_i$ модели авторегрессии, определенных для каждого из временных рядов s_i:

$$\hat{y}_i = g(\hat{\boldsymbol{\theta}}_i, \hat{\mathbf{b}}),$$

где оптимальные параметры $\hat{\mathbf{b}}$ минимизируют ошибку классификации (2):

$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} \sum_{i=1}^{m} I\left(g(\hat{\boldsymbol{\theta}}_i, \mathbf{b}) \neq y_i\right). \tag{3}$$

Здесь вид функции g является одним из стандартных методов классификации, в частности в данной работе рассматриваются метод многоклассовой логистической регрессии и метод опорных векторов (SVM — support vector machine).

Далее рассмотрим различные варианты выделения признаков h_{ij}. Будут рассмотрены способы введения гипотезы порождения временного ряда s_i и оценки параметров порождения, $\mathbf{h}_i \equiv \hat{\boldsymbol{\theta}}_i(s_i)$, а также способ ручной генерации признаков на основе экспертно заданных функций.

3.1 Модель авторегрессии для одномерного временного ряда

В качестве гипотезы порождения временного ряда s рассмотрим авторегрессионную модель порядка n. Здесь n является структурным параметром, элементом (в данном случае, единственным) вектора $\boldsymbol{\lambda}$:

$$x_t = \theta_0 + \sum_{j=1}^{n} \theta_i x_{t-j} + \varepsilon_t.$$

В этом случае, оптимальные параметры $\hat{\boldsymbol{\theta}}_i \equiv \mathbf{h}_i$ для объекта s_i определяются минимизацией среднеквадратичной ошибки прогнозирования

$$\hat{\boldsymbol{\theta}}_i = \arg\min_{\boldsymbol{\theta}\in\Theta} S(s_i, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \arg\min_{\boldsymbol{\theta}\in\Theta} \left(\sum_{t=1}^{T(i)} \|x_i - \hat{x}_i\|^2 \right), \tag{4}$$

где

$$\hat{x}_i = \theta_0 + \sum_{j=1}^{n} \theta_i x_{t-j}.$$

3.2 Анализ сингулярного спектра

В качестве альтернативной гипотезы порождения данных рассмотрим модель ряда SSA (Singular Spectrum Analysis) [9]. Поставим в соответствие временному ряду $s = \{x_1, ..., x_T\}$ траекторную матрицу $\mathbf{X}$, т. е. матрицу следующего вида:

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ x_2 & x_3 & \cdots & x_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m-n+1} & x_{m-n+2} & \cdots & x_m \end{pmatrix}.$$

Построим сингулярное разложение матрицы $\mathbf{X}^\intercal\mathbf{X}$:

$$\mathbf{X}^\intercal\mathbf{X} = \mathbf{V}\mathbf{H}\mathbf{V}^\intercal, \quad \mathbf{H} = \operatorname{diag}(h_1, ..., h_n),$$

где $h_1, \ldots, h_n$ — собственные числа матрицы $\mathbf{X}^\intercal\mathbf{X}$, отвечающие за величины различных частот спектра временного ряда s. Для временного ряда s_i будем рассматривать вектор сингулярных чисел $\mathbf{h}_i$ в качестве нового признакового описания и строить классификатор $\hat{y}_i$ в виде

$$\hat{y}_i = g(\mathbf{h}_i, \mathbf{b}),$$

где оптимальные параметры $\hat{\mathbf{b}}$ минимизируют ошибку (3).

Для того чтобы описать метод ручного выделения признаков, необходима постановка прикладной задачи и экспертные знания о структуре объектов. Прикладную задачу классификации данных акселерометра и метод экспертного выделения признаков рассмотрим следующем разделе.

4 Классификация данных акселерометра

В качестве прикладной задачи рассматривается задача классификации активности человека по данным с акселерометра. Данные с акселерометра представляют собой трехмерный временной ряд $\{a_x(t), a_y(t), a_z(t)\}_{t=1}^{\mathrm{T}}$. Каждому временному ряду поставлена в соответствие метка класса y_i, обозначающая один из шести возможных типов действия: Jogging (Бег), Walking (Ходьба), Upstairs (Ходьба вверх), Downstairs (Ходьба вниз), Sitting (Сидение), Standing (Стояние). Требуется построить алгоритм классификации, ставящий в соответствие временному ряду метку класса.

В качестве базового метода построения признакового пространства рассматривается метод, изложенный в работе [5]. Признаки задаются экспертным путем; для трехмерного временного ряда (X, Y, Z-компоненты ускорения) длины 200 (что соответствует 10 с) выбираются следующие 40 признаков:

- $\{3\}$ — среднее ускорение по каждой оси;
- $\{3\}$ — стандартное отклонение ускорения по каждой оси;
- $\{3\}$ — среднее абсолютное отклонение ускорения по каждой оси;
- $\{1\}$ — среднее результирующее ускорение;
- $\{30\}$ — распределение значений временного ряда по каждой оси. Для каждой компоненты X, Y, Z вычисляется наибольшее и наименьшее значения на всем промежутке; область значений компоненты разбивается на 10 равных промежутков; для каждого промежутка вычисляется процент попавших в него значений компоненты временного ряда.

5 Вычислительный эксперимент

Приведем некоторые свойства признаков, получаемых оценкой параметров порождения временных рядов акселерометра. В этом разделе за критерий качества примем процент совпавших объектов внутри всех классов:

$$Q(\mathbf{b}) = \frac{1}{m}\sum_{i=1}^{m} I\left(g(\hat{\boldsymbol{\theta}}_i, \mathbf{b}) = y_i\right). \tag{5}$$

5.1 Иллюстрация свойств моделей порождения

Согласно предыдущему разделу, моделью порождения временного ряда $a_i(t)$ будем считать модель авторегрессии с длиной предыстории n:

$$a_i(t) = \theta_0 + \sum_{j=1}^{n} \theta_i a_i(t-j) + \varepsilon_t. \tag{6}$$

Оптимальные параметры авторегрессионной модели $\hat{\boldsymbol{\theta}}$ определим с помощью минимизации среднеквадратичной ошибки прогнозирования (4). На рис. 1 показаны примеры исходных временных рядов для каждого из шести классов, а также их авторегрессионный прогноз при оптимальных значениях параметров $\hat{\theta}_i$.

Для определения параметра n, оптимальной длины предыстории в модели авторегрессии, построим график зависимости критерия качества (5) от n на тестовой выборке. В качестве модели классификации рассмотрим модель многоклассовой логистической регрессии. График зависимости показан на рис. 2. Синей линией показан график среднего на контроле качества для модели одномерной авторегрессии (6), зеленой линией — для трехмерной, где авторегрессионные параметры вычисляются отдельно для каждой из компонент $\{a_x(t), a_y(t), a_z(t)\}_{t=1}^{\mathrm{T}}$. Видно, что модель трехмерной авторегрессии существенно превосходит одномерную модель.

Для построения финальной модели классификации была выбрана трехмерная модель авторегрессии и длина предыстории для нее $n = 20$.

Иллюстрация собственных чисел для метода SSA показана на рис. 3: синие линии — примеры для класса Jogging; красные — Walking; зеленые — Standing. Для этих трех классов видна характерная разделимость выборки по спектру.

5.2 Результаты классификации

В этом разделе приведем результаты классификации данных акселерометра для различных методов выделения признаков с использованием порядковой логистической регрессии и SVM в качестве классификаторов в новом признаковом пространстве.

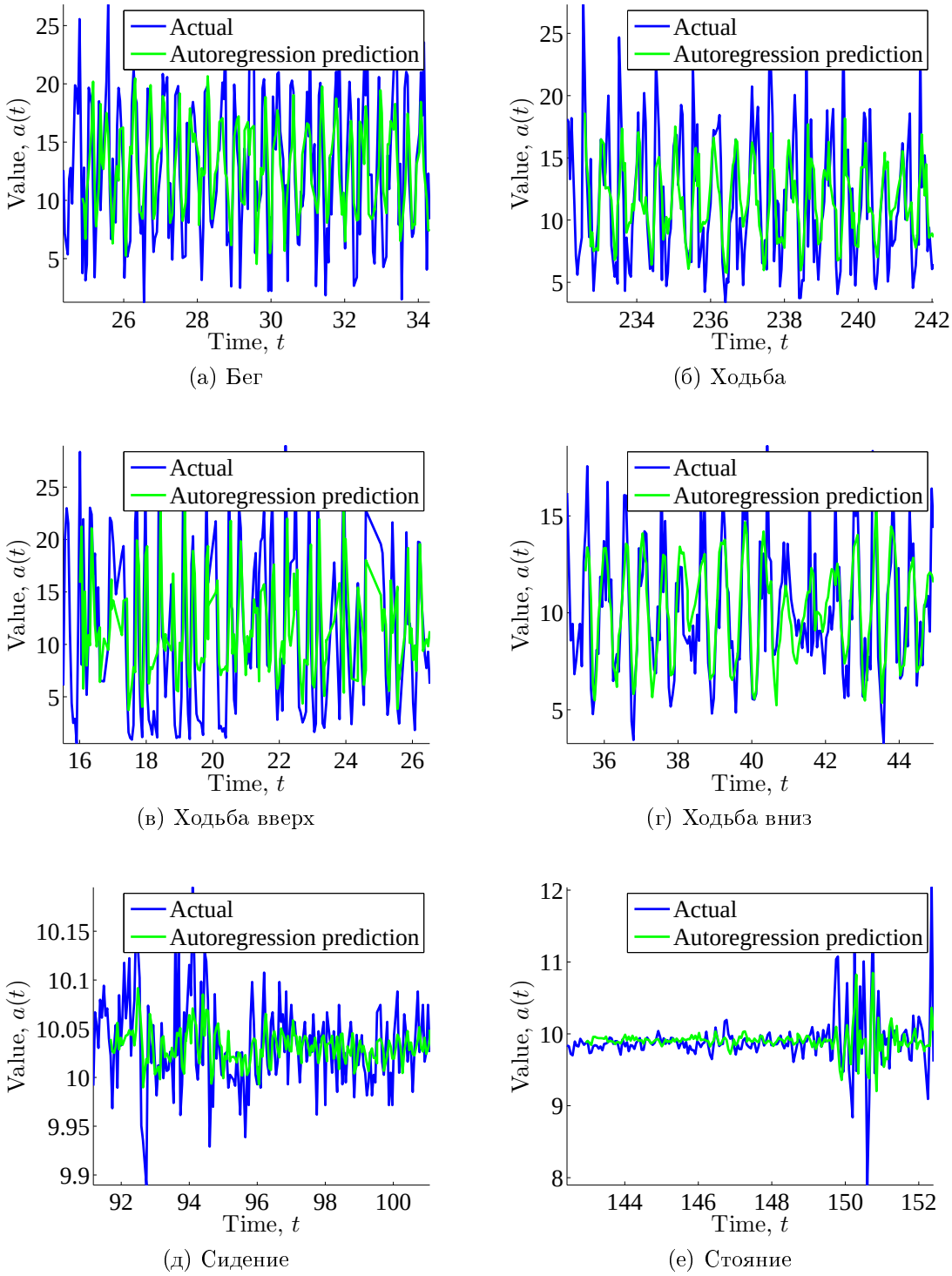
В качестве процедуры разбиения выборки на обучение и контроль был реализован метод разбиения с повторениями. На каждой итерации выборка разбивалась случайно в пропорции 70% обучения / 30% контроль; разбиения происходили независимо 50 раз. Представленное в табл. 1 качество Q является значением средней достоверности (accuracy) многоклассового классификатора на контрольных подвыборках:

$$Q = \frac{\sum_{k=1}^{6} \mathrm{tp}_k}{m},$$

где tp_k — количество правильно выделенных объектов внутри класса $k \in \{1 \ldots 6\}$.

Отметим, что, во-первых, процедура разбиения с повторениями не позволяет оценить дисперсию исследуемой величины достоверности, однако позволяет с достаточной степенью точности провести оценку среднего значения достоверности в силу большого количества генерируемых подвыборок. Во-вторых, исследуемая величина достоверности имеет явный перекос в сторону классов с большим количеством объектов, которыми в данной задаче являются классы Walking и Jogging. Для решения этой проблемы приводятся значения достоверности tp_k/m_k внутри каждого класса по отдельности.

Отметим также, что для итоговой классификации были выбраны два наиболее простых алгоритма классификации (многоклассовая логистическая регрессия и SVM с полиноми-



(а) Бег (б) Ходьба

(в) Ходьба вверх (г) Ходьба вниз

(д) Сидение (е) Стояние

Рис. 1 Исходные данные и оптимальная авторегрессионная модель

альным ядром степени 3), разделяющих объекты в линейном пространстве гиперплоскостями простой формы. Такой выбор связан с желанием авторов продемонстрировать свойства нового признакового пространства, а не решающего правила классификации. Как будет видно из результатов эксперимента, даже такие алгоритмы классификации

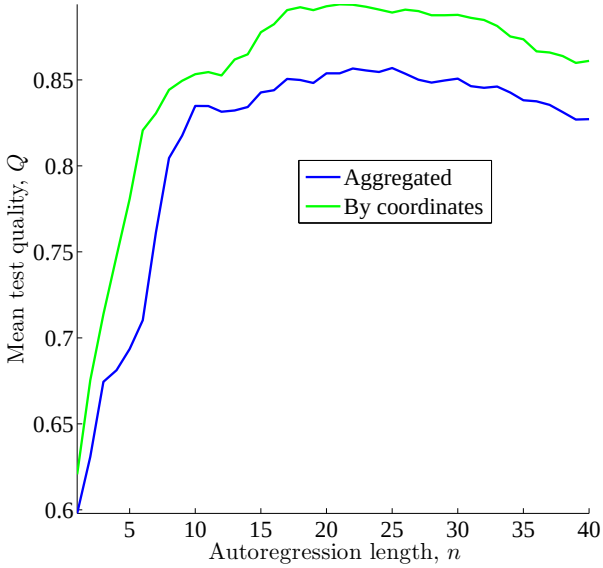


Рис. 2 Зависимость критерия качества от длины предыстории

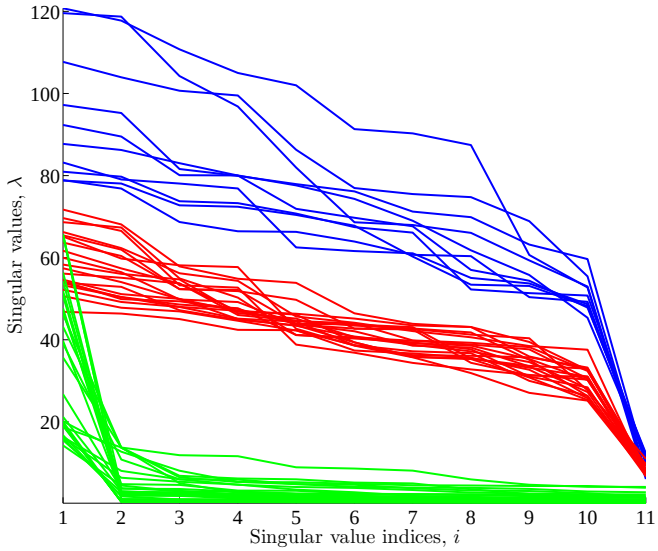


Рис. 3 Иллюстрация собственных чисел для метода SSA: синие линии — примеры для класса Бег; красные — Ходьба; зеленые — Стояние

позволяют достичь хороших результатов, что позволяет говорить о полноте множества выбранных признаков и переходить к решению задачи онлайн сегментации временного ряда по выбранным признакам.

Результаты классификации для модели авторегрессии с лагом $n = 20$ и логистической регрессии показаны на рис. 4, *а*. Синими столбиками отмечено общее количество объектов в классе, зелеными — среднее количество правильно классифицированных на контрольных подвыборках. Общее значение критерия качества составило $Q = 90\%$.

В табл. 1, *а* представлены результаты классификации. По горизонтали отложены актуальные значения классов, про вертикали — предсказанные. Число в таблице равно среднему количеству предсказаний за чужой класс. Последний столбец в таблице показывает долю правильно классифицированных объектов внутри каждого класса.

Таблица 1 Результаты классификации

(a) AR-модель, логистическая регрессия

	Предсказанный класс						
	Бег	Ходьба	Ходьба вверх	Ходьба вниз	Сидение	Стояние	Точность
Бег	469	6	4	5	0	0	0,97
Ходьба	3	604	6	10	1	1	0,97
Ходьба вверх	18	23	112	9	1	1	0,68
Ходьба вниз	10	15	19	86	2	2	0,64
Сидение	3	1	5	1	68	5	0,82
Стояние	2	0	3	2	5	58	0,83

(b) AR-GARCH-модель, SVM

	Предсказанный класс						
	Бег	Ходьба	Ходьба вверх	Ходьба вниз	Сидение	Стояние	Точность
Бег	480	4	3	1	0	0	0,98
Ходьба	2	614	4	6	0	1	0,98
Ходьба вверх	4	5	146	10	1	0	0,88
Ходьба вниз	2	4	11	115	0	1	0,86
Сидение	0	0	1	1	72	7	0,89
Стояние	0	1	1	1	8	57	0,84

(c) Анализ спектра, логистическая регрессия

	Предсказанный класс						
	Бег	Ходьба	Ходьба вверх	Ходьба вниз	Сидение	Стояние	Точность
Бег	475	10	3	5	1	0	0,96
Ходьба	2	595	12	6	1	2	0,96
Ходьба вверх	14	44	85	21	1	0	0,52
Ходьба вниз	1	34	28	68	0	0	0,52
Сидение	0	0	2	1	79	3	0,93
Стояние	0	5	0	2	2	60	0,87

(d) Ручное выделение признаков, SVM

	Предсказанный класс						
	Бег	Ходьба	Ходьба вверх	Ходьба вниз	Сидение	Стояние	Точноесть
Бег	483	4	4	2	0	0	0,98
Ходьба	1	607	10	9	0	0	0,97
Ходьба вверх	6	17	120	18	0	0	0,77
Ходьба вниз	1	10	12	106	0	0	0,82
Сидение	2	6	7	7	57	5	0,72
Стояние	0	9	8	4	2	45	0,71

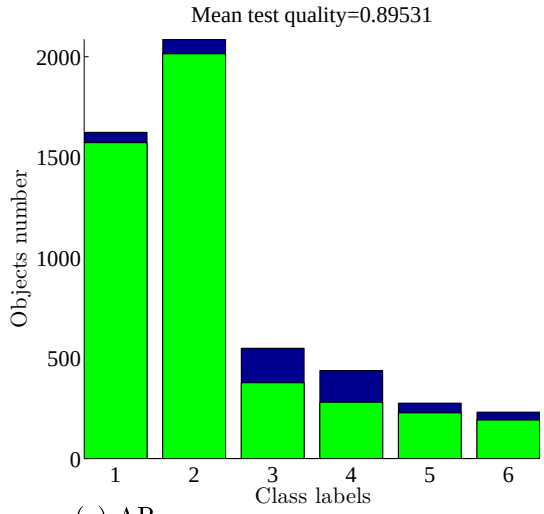
(e) Совместное использование всех признаков, SVM

	Предсказанный класс						
	Бег	Ходьба	Ходьба вверх	Ходьба вниз	Сидение	Стояние	Точность
Бег	490	1	2	1	0	0	0,99
Ходьба	0	622	1	4	0	0	0,99
Ходьба вверх	1	2	154	5	0	0	0,95
Ходьба вниз	0	2	4	124	0	0	0,95
Сидение	0	1	2	0	79	1	0,95
Стояние	0	0	1	1	1	65	0,96

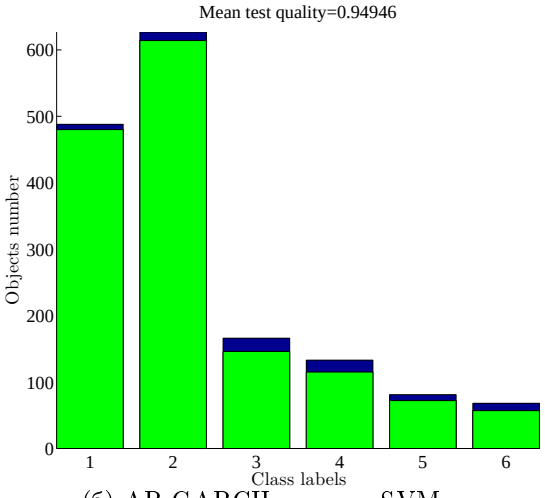
С помощью усложнения прогностической модели до AR-GARCH (с параметрами AR = 20, ARCH = 2, GARCH = 1), а также использования SVM с квадратичным ядром удалось увеличить качество классификации до 95% (см. рис. 4, *б* и табл. 1, *b*). Описание GARCH модели подробно изложено, например, в [6, гл. 14].

В качестве классифицирующей функции $g(\boldsymbol{\lambda}_i, \mathbf{b})$ для спектрального метода выделения признаков рассмотрим многоклассовую логистическую регрессию. Результаты классификации спектрального метода показаны на рис. 4, *в* и в табл. 1, *c*. Несмотря на то что среднее качество на контрольной выборке составило 87%, следует отметить, что данный метод позволил лучше остальных выделить классы Sitting (93%) и Standing (87%).

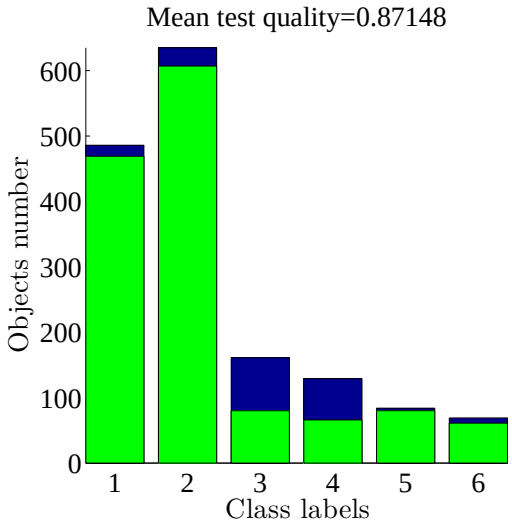
На рис. 4, *г* и в табл. 1, *d* показаны результаты классификации для экспертного метода выделения признаков. Среднее качество на контрольной выборке составило 91%.



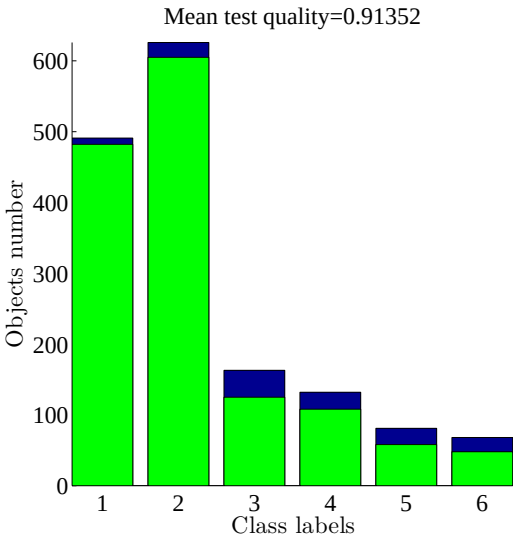
(а) AR-модель, логистическая регрессия



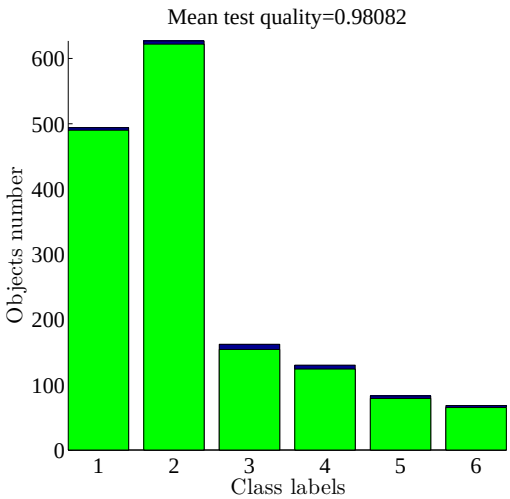
(б) AR-GARCH-модель, SVM



(в) Анализ спектра, логистическая регрессия



(г) Ручное выделение признаков, SVM



(д) Совместное использование всех признаков, SVM

Рис. 4 Результаты классификации

Таблица 2 Итоговые результаты классификации для различных методов выделения признаков

Действие	Точность				
	AR-logistic	AR-Garch-SVM	Manual-SVM	Spectral-logistic	**All feats-SVM**
Бег	0,97	*0,98*	*0,98*	0,96	**0,99**
Ходьба	0,97	*0,98*	0,97	0,96	**0,99**
Ходьба вверх	0,68	*0,88*	0,75	0,52	**0,95**
Ходьба вниз	0,64	*0,86*	0,82	0,52	**0,95**
Сидение	0,82	0,89	0,68	*0,93*	**0,95**
Стояние	0,83	0,84	0,66	*0,87*	**0,96**
Суммарно	0,90	*0,95*	0,91	0,87	**0,98**

Результаты классификации для использования всех перечисленных признаков в совокупности и SVM с полиномиальным ядром степени 3 показаны на рис. 4, *д* и в табл. 1, *е*. Итоговое качество классификации составило 98%.

Итоговые результаты для всех методов представлены в табл. 2. Курсивом показаны наилучшие результаты классификации для отдельных методов выделения признаков. Видно, что наибольшую точность (95%) имеет метод AR-Garch-SVM, однако альтернативные методы выделения признаков лучше справляются с выделением отдельных классов. Результаты классификации в случае использования всех признаков в совокупности иллюстрирует последний столбец.

6 Заключение

Рассмотрена задача многоклассовой классификации временных рядов как объектов сложной структуры. Исследованы различные методы построения признакового описания временных рядов, в частности метод экспертного построения признаков и метод построения признакового описания на основе гипотезы порождения данных. Проведено исследование различных методов построения признаков и решающих правил в задаче классификации данных акселерометра. Результаты показывают, что построенное признаковое пространство достаточным образом описывает зависимую переменную и приводит к высоким результатам классификации.

Авторы выражают благодарность В. В. Стрижову за постановку задачи и внимательное отношение к работе.

Литература

[1] *Geurts P.* Pattern extraction for time series classification // Principles of data mining and knowledge discovery. — Springer, 2001. P. 115–127.

[2] *Wei L., Keogh E.* Semi-supervised time series classification // 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Proceedings. — New York, NY, USA: ACM, 2014. P. 748–753. http://doi.acm.org/10.1145/1150402.1150498.

[3] *Wang W., Liu H., Yu L., Sun F.* Human activity recognition using smart phone embedded sensors: A linear dynamical systems method // Joint Conference (International) on Neural Networks (IJCNN), 2014. P. 1185–1190.

[4] *Ignatov A., Strijov V.* Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer // Multimedia Tools Applications, 2015. P. 1–14. doi: 10.1007/s11042-015-2643-0. http://dx.doi.org/10.1007/s11042-015-2643-0.

[5] *Kwapisz J. R., Weiss G. M., Moore S. A.* Activity recognition using cell phone accelerometers // SIGKDD Explor. Newsl., 2011. Vol. 12. No. 2. P. 74–82. doi: 10.1145/1964897.1964918. `http://doi.acm.org/10.1145/1964897.1964918`.

[6] *Lukashin Y. P.* Adaptive methods for short-term forecasting. — Finansy i Statistika, 2003.

[7] *Mörchen F.* Time series feature extraction for data mining using DWT and DFT. 2003.

[8] *Zhang H., Ho T. B., Lin M. S.* A non-parametric wavelet feature extractor for time series classification // Advances in knowledge discovery and data mining. — Springer, 2004. P. 595–603.

[9] *Hassani H.* Singular spectrum analysis: Methodology and comparison. — Cardiff University and Central Bank of the Islamic Republic of Iran, 2007. 19 p.

References

[1] Geurts, P. 2001. Pattern extraction for time series classification. *Principles of data mining and knowledge discovery*. Springer. 115–127.

[2] Wei, L., and E. Keogh. 2014. Semi-supervised time series classification. *12th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining Proceedings*. New York, NY: ACM. P. 748–753. Available at: `http://doi.acm.org/10.1145/1150402.1150498`.

[3] Wang, W., H. Liu, L. Yu, and F. Sun. 2014. Human activity recognition using smart phone embedded sensors: A linear dynamical systems method. *Joint Conference (International) on Neural Networks (IJCNN)*. 1185–1190.

[4] Ignatov, A., and V. Strijov. 2015. Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. *Multimedia Tools Applications*. 1–14. doi: 10.1007/s11042-015-2643-0. Available at: `http://dx.doi.org/10.1007/s11042-015-2643-0`.

[5] Kwapisz, J. R., G. M. Weiss, and S. A. Moore. 2011. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.* 12(2):74–82. doi: 10.1145/1964897.1964918. Available at: `http://doi.acm.org/10.1145/1964897.1964918`.

[6] Lukashin, Y. P. 2003. *Adaptive methods for short-term forecasting*. Finansy i Statistika.

[7] Mörchen, F. 2003. Time series feature extraction for data mining using DWT and DFT.

[8] Zhang, H., T. B. Ho, and M. S. Lin. 2004. A non-parametric wavelet feature extractor for time series classification. *Advances in knowledge discovery and data mining*. Springer. 595–603.

[9] Hassani, H. 2007. *Singular spectrum analysis: Methodology and comparison*. Cardiff University and Central Bank of the Islamic Republic of Iran.

Восстановление пропущенных значений в разнородных шкалах с большим числом пропусков*

О. Ю. Бахтеев
bakhteev@phystech.edu
Московский физико-технический институт

Рассматривается задача восстановления пропущенных значений в выборках, содержащих значительное число пропусков. Вводится понятие устойчивости восстановления пропуска, а также исследуется возможность применимости подхода для восстановления пропущенных значений. Исследуется случай, когда восстановление производится по k ближайшим соседям. Рассматриваются теоретические аспекты применимости данного подхода для сильно разреженных данных. Рассматривается вариант восстановления пропущенных значений с использованием восстановленных значений в качестве источника для восстановления других элементов.

Ключевые слова: *восстановление пропущенных значений; k ближайших соседей; разнородные шкалы*

Handling missing values in mixed-scale datasets with large amount of missing values*

O. Y. Bakhteev
Moscow Institute of Physics and Technology

Background: The paper investigates the problem of missing values handling in datasets with a large amount of missing values. One of the problems of missing values filling methods is their instability. The order of missing values filling can seriously change the efficiency of the method. The paper considers the case when the dataset has significant amount of features with discrete scales with low cardinality.
Methods: There are different methods of missing values handling. The paper focuses on the filling missing values using the metric properties of the dataset. The paper proposes some definitions and statements in order to formalize the problem of instability. The method using k nearest neighbors is considered. The paper considers a variation of the method that uses already filled missing values as values of nearest neighbors for new fill. Also, some theoretical aspects of this method implementation are considered.
Results: In order to analyze the behavior and efficiency of the considered method, two experiments were conducted. The results were compared with other missing values filling techniques such as filling with decision trees and filling with average value of the scale.
Concluding Remarks: The proposed mathematical framework can be used for further research of missing values filling methods.
Keywords: *imputation; missing values; k nearest neighbours; mixed-scale datasets*

1 Введение

В работе исследуется проблема восстановления значительного количества пропусков в выборке в задачах анализа данных. Основной трудностью, связанной с восстановлением

*Работа выполнена при поддержке РФФИ, грант №14-07-31045.

пропусков, является неустойчивость полученной модели при последовательном восстановлении части пропусков: порядок восстановления пропусков может значительно изменить вид восстановленной выборки. Примером данных с подобным количеством пропущенных значений является выборка историй болезни лошадей, обследованных в ветеринарной клинике [1]. В выборке присутствуют 28 признаков в разнородных шкалах, около 30% выборки заполнено пропущенными значениями. В данной работе алгоритмы восстановления пропусков исследуются на примере социологических данных [2]. В данной выборке содержится 1000 объектов с признаковым описанием в номинальных, линейных и порядковых шкалах.

Ранее был предложен ряд подходов, используемых для обработки пропущенных значений. В работе [3] рассматривается исключение из выборки данных с пропущенными значениями. При значительном количестве пропущенных значений данный метод не позволяет построить адекватную модель выборки. Кроме того в случае если в выборке не существует объекта с полностью восстановленными атрибутами, метод неприменим. В работах [3, 4] рассматривается построение математических моделей на подмножествах атрибутов, соответствующих восстановленным атрибутам объектов. Для каждого объекта выборки находится подмножество его восстановленных атрибутов, и для нее строится математическая модель. Данный подход требует согласования математических моделей, учитывающих разный набор атрибутов каждого объекта, и потому требует больших вычислительных ресурсов. Оба этих метода отбрасывают пропущенные значения в данных. Другим подходом к их обработке является восстановление пропусков [5] по имеющимся данным выборки. Перечислим некоторые его варианты:

- восстановление средними значениями атрибута по всей выборке [6]. Метод является достаточно простым для реализации, однако дает грубые результаты и может ухудшить результаты работы дальнейшей классификации или регрессии [7];
- восстановление с использованием предсказательной модели (Predictive value imputation) [3]. Данный метод предполагает восстановление пропущенного значения на основе некоторой зависимости данных исходной выборки;
- восстановление с использованием распределения значений атрибута [3, 8]. Метод предполагает оценку распределения значений атрибута и дальнейшее восстановление данных с использованием этого распределения. Данный подход можно встретить, например, в алгоритме построения дерева решений C4.5.

В работах [3, 6] проводится обзор основных подходов к восстановлению пропущенных значений. В работе [8] описывается подход к восстановлению, основанный на методах прикладной статистики и теории вероятностей.

В работах [9, 10] рассматривается подход множественных заполнений (Multiple imputation), основанный на методе Монте Карло. При использовании этого подхода восстановление каждого пропуска происходит несколько раз, таким образом генерируются несколько полностью восстановленных выборок. Затем происходит слияние полученных выборок.

В работах [6, 11, 13, 14, 15, 12] рассматривается подход к восстановлению пропущенных значений, основанный на методе k ближайших соседей. Данный подход восстановления пропусков восстанавливает значения как в непрерывных шкалах, так и в дискретных [6]. В работах [14, 13, 11, 6] отражены результаты экспериментов по восстановлению пропущенных значений с применением данного подхода. В работе [15] для оценки погрешности метода k ближайших соседей использовались методы математической статистики,

производились оценки среднеквадратичного отклонения реального значения атрибута от значения, полученного методом k ближайших соседей. В работе [16] рассматривается проблема восстановления пропусков по неполностью восстановленным соседям. Частично эта проблема решается в [12], где предлагается итеративная версия восстановления пропущенных значений. На первой итерации все пропуски восстанавливаются средним значением признака.

В данной работе исследуется задача восстановления пропусков в случае значительного числа признаков, выполненных в дискретных шкалах малой мощности. В работе не вводятся статистические предположения о распределении значений признаков. Подобный класс данных встречается в задачах экспертного оценивания [17, 18].

Для восстановления пропущенных значений рассматривается подход, основанный на восстановлении пропусков по k ближайшим соседям. Вводится функция устойчивости восстановления, учитывающая, насколько восстановление может улучшить дальнейшее восстановление пропусков выборки. Предлагается подход, позволяющий проводить транзитивные восстановления [16], т. е. использование объектов с пропуском в некотором поле для дальнейшего восстановления пропусков в этом же поле для других объектов. Второй вариант алгоритма не использует транзитивное восстановление. Вводится функция ошибки восстановления пропущенных значений, соответствующая сумме расстояний до реальных значений объектов, в метрике пространства объектов. Изучаются границы применимости предлагаемого алгоритма восстановления пропущенных значений.

2 Формальная постановка задачи

В данном разделе вводятся формальная постановка задачи восстановления пропущенных значений и определения, требуемые для формализации задачи восстановления пропущенных значений.

Определение 1. *Шкала $\mathbb{L}$ — алгебраическая структура* [19] *с заданным набором операций и отношений, удовлетворяющая фиксированному набору аксиом.*

Определение 2. *Номинальная шкала $\mathbb{C}$ — шкала с заданным на ней бинарным отношением равенства:*

1. $x = y \vee x \neq y$;
2. $x, y : x = y \Rightarrow y = x$;
3. $x, y, z : x = y \wedge y = z \Rightarrow x = z$,

где x, y, z — объекты, представленные в шкале $\mathbb{C}$: $x, y, z \in \mathbb{C}$.

Определение 3. *Порядковая шкала $\mathbb{O}$ — номинальная шкала с заданным на ней бинарным отношением R, для которого выполнены следующие свойства:*

1. xRx,
2. $xRy \wedge yRx \Rightarrow x = y$;
3. $xRy \wedge yRz \Rightarrow xRz$;

где $x, y, z \in \mathbb{O}$.

Определение 4. *Линейная шкала $\mathbb{W}$ — порядковая шкала с отношением полного порядка и определенными операциями сложения и вычитания.*

Задана выборка $\mathbf{X}$ — множество вектор-строк:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^\top \subset \mathbb{X},$$

$$\mathbf{X} = \begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \mathbf{x}_3\end{bmatrix} \begin{bmatrix}1 & \square\\ \square & 2\\ 4 & 3\end{bmatrix} \to \hat{\mathbf{X}}^1 = \begin{bmatrix}1 & 3\\ \square & 2\\ 4 & 3\end{bmatrix} \to \hat{\mathbf{X}}^2 = \begin{bmatrix}1 & 3\\ 4 & 2\\ 4 & 3\end{bmatrix}$$

Рис. 1 Пример восстановления пропущенных значений

лежащих в пространстве $\mathbb{X}$:

$$\mathbb{X} = (\mathbb{L}_1 \cup \{\square\}) \times \ldots \times (\mathbb{L}_n \cup \{\square\}).$$

где $\mathbb{X}$ — множество возможных значений векторов признаков объектов или пространство объектов с введенной на нем метрикой d; $\mathbb{L}_j$ — линейная, номинальная или порядковая шкала, $\square$ — символ, соответствующий пропущенному значению. В выборке находится ℓ пропущенных значений, $\ell > 0$.

Определение 5. *Пусть $\mathbf{x}_i \in \mathbf{X}$ — объект, имеющий пропуск в j-м признаке. Пусть объекты $\mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k} \in \mathbf{X}$ — объекты с заполненными значениями j-го признака. Операцией восстановления j-го признака объекта $\mathbf{x}_i$ по объектам $\mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k} \in \mathbf{X}$ назовем следующее отображение:*

$$\mathbf{x}_i \leftarrow \{x_{q_1 j}, \ldots, x_{q_k j}\} = [x_{i1}, \ldots, x_{ij-1}, \operatorname{average}([x_{q_1 j}, \ldots, x_{q_k j}]), x_{ij+q}, \ldots, x_{in}],$$

где

$$\operatorname{average}([x_{q_1 j}, \ldots, x_{q_k j}]) = \begin{cases} \operatorname{mean}([x_{q_1 j}, \ldots, x_{q_k j}]), & \text{если шкала } \mathbb{L}_j \text{ — линейная;}\\ \operatorname{median}([x_{q_1 j}, \ldots, x_{q_k j}]), & \text{если шкала } \mathbb{L}_j \text{ — порядковая;}\\ \operatorname{mode}([x_{q_1 j}, \ldots, x_{q_k j}]), & \text{если шкала } \mathbb{L}_j \text{ — нормальная;}\end{cases}$$

k — множество соседей, т. е. объектов по которым восстанавливается признак.

В дальнейшем операцию восстановления $\mathbf{x}_i \leftarrow \{\mathbf{x}_{q_1 j}, \ldots, \mathbf{x}_{q_k j}\}$ будем отождествлять с кортежем вида $\mathbf{t} = (i, j, q_1, \ldots, q_k)$. Также будем обозначать через $\hat{\mathbf{X}}^b$ выборку, полученную из исходной $\mathbf{X}$ последовательным выполнением b операций восстановления. Будем полагать $\hat{\mathbf{X}}^0 = \mathbf{X}$.

Определение 6. *Операцию $\mathbf{t} = (i, j, q_1, \ldots, q_k)$ назовем корректной для выборки $\mathbf{X}$, если $x_{ij} = \square$ и $x_{q_r j} \neq \square$ для $r \in \{1, \ldots, k\}$.*

Определение 7. *Последовательность операций восстановления $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_b)$ назовем корректной, если каждая операция $\mathbf{t}_p \in \{\mathbf{t}_1, \ldots, \mathbf{t}_b\}$ корректна для выборки $\hat{\mathbf{X}}^{p-1}$, где $\hat{\mathbf{X}}^{p-1}$ — выборка, полученная из $\mathbf{X}$ последовательным выполнением операций $\mathbf{t}_1, \ldots, \mathbf{t}_{p-1}$, $\hat{\mathbf{X}}^0 = \mathbf{X}$, $p \in \{1, \ldots, b\}$.*

Пример восстановления значений с помощью последовательности из двух операций $\mathbf{T} = ((1, 2, 3)(2, 1, 3))$ приведен на рис. 1.

Определение 8. *Множество корректных последовательностей операций восстановления длины ℓ обозначим как $\mathbf{C}_\ell$.*

Определение 9. *Обозначим за $\operatorname{filled}_o(\mathbf{x}, \mathbf{X})$ множество индексов заполненных значений объекта $\mathbf{x}$ в выборке $\mathbf{X}$. Обозначим за $\operatorname{filled}_f(j, \mathbf{X})$ множество индексов объектов с заполненным признаком j в выборке $\mathbf{X}$.*

Определение 10. *Пусть* $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_b)$ *— корректная последовательность операций,* $\mathbf{t}_b = \{i, j, q_1, \ldots, q_k\}$, $|\mathbf{T}| \geqslant 0$. *Устойчивостью восстановления* $x_{ij} \leftarrow \mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k}$ *под действием* $\mathbf{T}$ *назовем следующую величину:*

$$u(\mathbf{x}_i \leftarrow x_{q_1 j}, \ldots, x_{q_k j}|\mathbf{T}) = \mathrm{mean}_{r\in\{1,\ldots,k\}} \left(\frac{|\mathrm{filled}_o(\hat{\mathbf{x}}_i^b, \mathbf{X}^b) \cap \mathrm{filled}_o(\hat{\mathbf{x}}_{q_r}^b, \mathbf{X}^b)|}{n} \frac{|\mathrm{filled}_f(j, \mathbf{X}^b)}{m} \right).$$

Определение 11. *Пусть* $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_l)$ *— корректная последовательность длины* ℓ, $\mathbf{t}_b = (i_b, j_b, q_{(b,1)}, \ldots, q_{(b,k)})$, *где индекс* b *пробегает от* 1 *до* ℓ, $b \in \{1, \ldots, \ell\}$.

Устойчивостью последовательности восстановлений $\mathbf{T}$ *для выборки* $\mathbf{X}$ *назовем величину:*

$$U(\mathbf{X}|\mathbf{T}) = u(\mathbf{x}_{i_1} \leftarrow x_{q_{(1,1)}j_1}, \ldots, x_{q_{(1,k)}j_1}) + \sum_{b=2}^{\ell} u(\mathbf{x}_{i_b} \leftarrow x_{q_{(b,1)}j_b}, \ldots, x_{q_{(b,k)}j_b}|(\mathbf{t}_1, \ldots, \mathbf{t}_{b-1})).$$

Требуется найти последовательность восстановлений $\mathbf{T}$, решающую следующую задачу оптимизации:

$$\mathbf{T} = \underset{\mathbf{T}'\in\mathbf{C}_\ell, U(\mathbf{X}|\mathbf{T}')=\max}{\arg\min} \sum_{i=1}^{m}\sum_{j=1}^{n} d(\hat{x}_{ij}, x'_{ij}),$$

где $\hat{\mathbf{x}}_i = [x_{i1}, \ldots, x_{in}]$ — объект, восстановленный под действием последовательности $\mathbf{T}$; $\mathbf{x}'$ — объект с реальными значениями пропусков; d — метрика на пространстве $\mathbb{X}$. Подробно метрики в разнородных шкалах описаны в разд. 5.

Таким образом, исходная задача разбивается на две подзадачи:

1. Нахождение множества корректных последовательностей $\mathbf{T}'$, под действием которых устойчивость U выборки максимальна.
2. Выбор последовательности $\mathbf{T}$, доставляющей минимум сумме расстояний от восстановленных объектов до реальных.

Рассмотрим подробнее операцию восстановления пропущенных значений. Будем восстанавливать пропущенные значения с восстановлением пропусков по k ближайшим соседям. При использовании данного алгоритма пропущенные значения объекта выборки восстанавливаются по множеству k ближайших к нему объектов выборки. Рассмотрим данный алгоритм восстановления пропусков на примере.

2.1 Пример 1

Пусть задана выборка $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} \begin{bmatrix} 1 & 2 & \square & \square & \square \\ 1 & 2 & 6 & 5 & \square \\ 4 & 8 & 0 & \square & 5 \end{bmatrix}.$$

Будем считать, что на пространстве $\mathbb{X}$ и на каждом его подпространстве $\mathbb{X}_\mathcal{J} = \prod_{j\in\mathcal{J}} \mathbb{L}_j$ задана метрика $d_\mathcal{J}$, принимающая значения из отрезка $[0; 1]$.

Восстановим пропущенное значение x_{13} с применением алгоритма k ближайших соседей. Рассмотрим случай $k = 1$.

Для объектов $\mathbf{x}_1$, $\mathbf{x}_2$ и $\mathbf{x}_3$ имеется общее подпространство $\mathbb{X}_\mathcal{J}$, т. е. такое пространство, в котором ни у одного объекта не содержится пропущенных значений. Это подпространство соответствует проекции пространства объектов $\mathbb{X}$ на первые два признака. Определим, какой из объектов $\mathbf{x}_2$, $\mathbf{x}_3$ является ближайшим для объекта $\mathbf{x}_1$:

$$\mathbf{x}' = \underset{\mathbf{x}'\in\{\mathbf{x}_2,\mathbf{x}_3\}}{\arg\min}\ d_\mathcal{J}\big(\mathrm{f}_\mathcal{J}(\mathbf{x}_1), \mathrm{pr}_\mathcal{J}(\mathbf{x}')\big), \tag{1}$$

где $d_{\mathcal{J}}$ — метрика на пространстве $\mathbb{X}_{\mathcal{J}}$, принимающая значения из $[0;1]$; $\mathrm{pr}_{\mathcal{J}}$ — функция, проецирующая объекты на пространство $\mathbb{X}_{\mathcal{J}}$. Пусть согласно метрике $d_{\mathcal{J}}$ $\mathbf{x}' = \mathbf{x}_2$.

Восстановим пропущенный признак x_{13} по ближайшему соседу $\mathbf{x}'$:

$$\hat{\mathbf{x}}_1 = \mathbf{x}_1 \leftarrow \{x_{23}\}.$$

В случае $k > 1$ восстановленное значение усредняется по нескольким ближайшим соседям. Так, в данном примере при $k = 2$ восстановленное значение x_{13} будет равняться среднему соответствующих значений объектов $\mathbf{x}_2, \mathbf{x}_3$.

Рассмотрим теперь случай, когда для всех трех объектов $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ не существует общего пространства $\mathbb{X}_{\mathcal{J}}$.

2.2 Пример 2

Пусть задана выборка $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & \square & \square & \square \\ \square & 2 & 6 & 5 & \square \\ 4 & \square & 0 & \square & 5 \end{bmatrix}.$$

Здесь общим пространством для объектов $\mathbf{x}_1, \mathbf{x}_2$ является подпространство $\mathbb{X}_{\{2\}}$, содержащее только второй признак, для объектов $\mathbf{x}_1, \mathbf{x}_3$ — подпространство $\mathbb{X}_{\{1\}}$, содержащее только первый признак.

В данном случае ближайший сосед будет определяться по различным подпространствам. Если метрики принимают значения из одного множества, будем находить ближайших соседей в различных пространствах, сравнивая полученные значения метрик между собой.

Предлагается использовать для восстановления пропущенных значений транзитивное восстановление, т. е. восстановление с использованием объектов с незаполненными полями в качестве соседей для восстановления этого же поля для других объектов. Поясним данный момент на следующем примере.

2.3 Пример 3

Пусть задана выборка $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} 1 & \square & \square \\ 1 & \square & 3 \\ \square & 2 & 3 \end{bmatrix}.$$

В данном примере для восстановления пропуска x_{12} требуется сперва восстановить пропуск x_{22} по соседу $\mathbf{x}_3$. Разрешение такого транзитивного восстановления усложняет алгоритм, однако при этом позволяет восстановить более широкий класс данных.

3 Нахождение оптимальной последовательности восстановления пропусков

Для построения стратегии восстановления пропусков будем использовать аппроксимацию функции устойчивости, которая не будет зависеть от соседей, по которым восстанавливается объект. Для дальнейшего рассмотрения задачи введем ряд определений.

Определение 12. *Пусть $\mathbf{x}_i \in \mathbf{X}$ — объект, имеющий пропуск в j-м признаке. Абстрактной операцией восстановления j-го признака объекта $\mathbf{x}_i$ назовем множество всех возможных операцией восстановления данного признака по одному соседу:*

$$\mathbf{x}_i \leftharpoondown j = \{(\mathbf{x}_i \leftarrow \{x_{qj}\}), q \in \mathrm{filled}_f(j, \mathbf{X})\}.$$

По аналогии с операцией восстановления абстрактную операцию восстановления $\mathbf{x}_i \leftharpoondown \leftharpoondown j$ будем отождествлять с кортежем (i, j).

Определение 13. *Последовательность абстрактных операций восстановления* $\bar{\mathbf{T}} = (\bar{\mathbf{t}}_1, \bar{\mathbf{t}}_b), \bar{\mathbf{t}} = (i, j)$, *где* $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$, *будем называть корректной последовательностью абстрактных операций восстановления для выборки* $\mathbf{X}$, *если для каждого* $\bar{\mathbf{t}} = (i, j)$: $x_{ij} = \square$ *и каждая пара кортежей* $\bar{\mathbf{t}}_1, \bar{\mathbf{t}}_2 \in \bar{\mathbf{T}}$ *отличается хотя бы по одной координате.*

Определение 14. *Пусть* $\bar{\mathbf{T}} = (\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_b)$ — *корректная последовательность абстрактных операций восстановления,* $|\bar{\mathbf{T}}| \geqslant 0$. *Пусть* $\bar{\mathbf{X}}^b$ — *выборка, полученная из исходной выборки* $\mathbf{X}$ *восстановлением значений* $x_{ij}, (i, j) \in \bar{\mathbf{T}}$ *произвольными значениями соответствующей шкалы,* $\bar{\mathbf{x}}_i^b \in \bar{\mathbf{X}}^b$. *Аппроксимированной устойчивостью пропуска* x_{ij} *назовем следующую величину:*

$$\bar{u}(x_{ij}|\bar{\mathbf{T}}) = \frac{|\text{filled}_o(\bar{\mathbf{x}}_i^b, \bar{\mathbf{X}}^b)|}{n} \frac{|\text{filled}_f(j, \bar{\mathbf{X}}^b)|}{m}.$$

Данная функция, в отличие от функции устойчивости u, не учитывает пересечение множества заполненных признаков восстанавливаемого объекта $\mathbf{x}_i$ и объектов, по чьим значениям восстанавливается пропуск. Таким образом, аппроксимированная устойчивость является верхней оценкой функции u.

Теорема 1. *Для каждого* $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$, *любого множества объектов* $\mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k}$, *имеющих заполненный признак* j *и корректной последовательности операций восстановления* $\mathbf{T}, |\mathbf{T}| = |\bar{\mathbf{T}}|$, *такой, что для любой операции* $\mathbf{t}_p \in \mathbf{T}$ *первые два элемента кортежа* $\mathbf{t}_p$ *равны элементам кортежа* $\bar{\mathbf{t}}_p \in \bar{\mathbf{T}}$, *следует, что*

$$u(\mathbf{x}_i \leftharpoondown \mathbf{x}_{q_1 j}, \ldots, \mathbf{x}_{q_k j}|\mathbf{T}) \leqslant \bar{u}(x_{ij}|\bar{\mathbf{T}}).$$

Доказательство. Доказательство следует из определений устойчивости заполнения и аппроксимированной устойчивости:

$$u(\mathbf{x}_i \leftharpoondown \mathbf{x}_{q_1 j}, \ldots, \mathbf{x}_{q_k j}|\mathbf{T}) = \text{mean}_{r \in \{1,\ldots,k\}} \frac{|\text{filled}_o(\hat{\mathbf{x}}_i^b, \hat{\mathbf{X}}^b) \cap \text{filled}_o(\hat{\mathbf{x}}_{q_r}^b, \hat{\mathbf{X}}^b)|}{n} \frac{|\text{filled}_f(j, \hat{\mathbf{X}}^b)|}{m} \leqslant$$
$$\leqslant \frac{|\text{filled}_o(\bar{\mathbf{x}}_i^b, \bar{\mathbf{X}}^b)|}{n} \frac{|\text{filled}_f(j, \bar{\mathbf{X}}^b)|}{m} = \bar{u}(x_{ij}|\bar{\mathbf{T}}).$$

■

По аналогии с определением устойчивости последовательности операций восстановления системы введем понятие аппроксимированной устойчивости последовательности абстрактных операций восстановления.

Определение 15. *Пусть* $\bar{\mathbf{T}} = (\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_b)$ — *последовательностей корректных абстрактных операций восстановления.* b*-Аппроксимированной устойчивостью последовательности абстрактных операций восстановления для выборки* $\mathbf{X}$ *назовем следующую величину:*

$$\bar{U}^b(\mathbf{X}|\bar{\mathbf{T}}) = \bar{u}(x_{i_1 j_1}) + \sum_{r=1}^{b-1} \bar{u}(x_{i_{r+1} j_{r+1}}|\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_r) + \sum_{(i,j) \notin \bar{\mathbf{T}}, x_{ij} = \square} \bar{u}(x_{ij}|\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_b).$$

Из определения и предыдущей теоремы немедленно вытекает следующее утверждение.

Теорема 2. *Пусть* $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_\ell)$ *— корректная последовательность длины* ℓ; $\bar{\mathbf{T}} = (\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_\ell)$ *— корректная последовательность абстрактных операций восстановления длины* ℓ, *такая что для любой операции* $\mathbf{t}_p \in \mathbf{T}$ *первые два элемента кортежа* $\mathbf{t}_p$ *равны элементам кортежа* $\bar{\mathbf{t}}_p \in \bar{\mathbf{T}}$. *Тогда*

$$U(\mathbf{X}|\mathbf{T}) \leqslant \bar{U}^\ell(\mathbf{X}|\bar{\mathbf{T}}).$$

Вместо исходной задачи максимизации устойчивости системы $U(\mathbf{X}|\mathbf{X})$ будем оптимизировать аппроксимацию $\bar{U}(\mathbf{X}|\bar{\mathbf{T}})$. На каждом шаге итерации алгоритм должен отбирать пропуск x_{ij}, имеющий корректную операцию восстановления $\mathbf{t}$, дающую максимум устойчивости $u(x_{ij} \leftarrow (\mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k}))$. Для учета последующих шагов восстановления будем просматривать аппроксимированную устойчивость выборки на несколько шагов вперед.

Для вычисления аппроксимированной b-устойчивости определим понятие графа зависимостей, соответствующего выборке $\mathbf{X}$.

Определение 16. *Графом зависимости* $\langle \mathbf{V}, \mathbf{E} \rangle$, *соответствующим выборке* $\mathbf{X}$, *назовем совокупность вершин и ребер, где каждая вершина* v_{ij} *соответствует элементу* x_{ij}, *а ребра строятся по следующим правилам:*

- *если в объекте* $\mathbf{x}_i$ *существует два пропущенных значения* x_{ij_1}, x_{ij_2}, *то между вершинами* v_{ij_1}, v_{ij_2} *существует ребро* e_{ij_1,ij_2};
- *если в объектах* $\mathbf{x}_{i_1}, \mathbf{x}_{i_2}$ *пропущено значение* j*-го признака, то между вершинами* v_{i_1}, v_{i_2} *существует ребро* $e_{i_1 j, i_2 j}$.

Приведем пример графа зависимости.

3.1 Пример 4

Пусть задана выборка

$$\mathbf{X} = \begin{bmatrix} 1 & \square & \square \\ 1 & \square & 3 \\ 1 & \square & 3 \\ \square & 2 & 3 \end{bmatrix}.$$

Граф зависимости для данной выборки изображен на рис. 2.

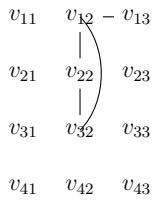


Рис. 2 Граф, соответствующий выборке $\mathbf{X}$

Определение 17. *Объектной степенью вершины* $\deg_O(v)$ *назовем степень вершины* v *с учетом только ребер между вершинами, соответствующими одному объекту.*
Признаковой степенью вершины $\deg_F(v)$ *назовем степень вершины* v *с учетом только ребер между вершинами, соответствующими одному признаку.*

Докажем ряд утверждений для реализации жадной стратегии выбора операции восстановления.

Теорема 3. *Пусть* $\bar{\mathbf{T}} = (\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_b)$ *— последовательность абстрактных операций восстановления,* $\bar{\mathbf{t}}_r = (i_r, j_r), r \in \{1, \ldots, b\}$. *Аппроксимированная устойчивость пропуска* x_{ij} *при условии последовательности* $\bar{\mathbf{T}}$ *выглядит следующим образом:*

$$\bar{u}(x_{ij}|\mathbf{t}_1, \ldots, \mathbf{t}_b) = \bar{u}(x_{ij}) + \delta(x_{ij}, x_{i_1 j_1}|\varnothing) + \sum_{r=2,\ldots,b} \delta(x_{ij}, x_{i_r j_r}|\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_{r-1}).$$

Здесь

$$\delta(x_{ij}, x_{qr}|\mathbf{I}) = \begin{cases} 0, \text{ если } i \neq q \text{ и } j \neq k; \\ \dfrac{(n - \deg_O(v_{ij}) - 1 + |O(i, \mathbf{I})|)}{mn}, \text{ если } j = r; \\ \dfrac{(m - \deg_F(v_{ij}) - 1 + |F(j, \mathbf{I})|)}{mn}, \text{ если } i = q, \end{cases}$$

где $O(i, \mathbf{I})$ *— множество кортежей из* $\mathbf{I}$*, на первом месте которых стоит* i*;* $F(j, \mathbf{I})$ *— множество кортежей из* $\mathbf{I}$*, на втором месте которых стоит* j*.*

Доказательство.

Пусть для начала $b = 0$. Тогда равенство является тривиальным.

Пусть теперь $b = 1$. Рассмотрим изменение аппроксимированной устойчивости $\Delta = \bar{u}(x_{ij}|\mathbf{t}_1) - \bar{u}(x_{ij})$ при условии кортежа $\bar{\mathbf{t}}_1$. Если $i \neq i_1$ и $j \neq j_1$, то восстановление пропуска $x_{i_1 j_1}$ никак не влияет на аппроксимированную устойчивость x_{ij}. Если $i = i_1$, то filled_o увеличится на единицу и, следовательно, аппроксимированная устойчивость увеличится на $|\text{filled}_f(j, \mathbf{X})|/(mn) = (m - \deg_F(v_{ij}) - 1)/(mn)$ относительно величины $\bar{u}(x_{ij})$. Если $j = j_1$, то filled_f увеличится на единицу и, следовательно, аппроксимированная устойчивость увеличится на $|\text{filled}_o(\mathbf{x}_i, \mathbf{X})|/(mn) = (n - \deg_O(v_{ij}) - 1)/(mn)$ относительно величины $\bar{u}(x_{ij})$, и равенство выполняется.

В случае $b > 1$ доказательство производится аналогично. Рассмотрим изменение аппроксимированной устойчивости для каждого $r \in \{1, \ldots, b\}$: $\Delta = \bar{u}(x_{ij}|\mathbf{t}_1, \ldots, \mathbf{t}_r) - \bar{u}(x_{ij}|\mathbf{t}_1, \ldots, \mathbf{t}_{r-1})$ при условии кортежей $\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_r$. Аналогично случаю $b = 1$ Δ может измениться на $|\text{filled}_f(j, \bar{\mathbf{X}}^{r-1})|/(mn) = \delta(x_{ij}, x_{i_r j_r}|\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_{r-1})$ или $|\text{filled}_o(\mathbf{x}_i, \bar{\mathbf{X}}^{r-1})|/(mn) = \delta(x_{ij}, x_{i_r j_r}|\bar{\mathbf{t}}_1, \ldots, \bar{\mathbf{t}}_{r-1})$, и равенство выполняется. ■

Рассмотрим частный случай 1-аппроксимированной устойчивости выборки.

Теорема 4. *Пусть задан кортеж* $\bar{t} = (i, j)$*, такой что* $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$*,* $x_{ij} = \square$*. Тогда аппроксимированная* 1*-устойчивость выборки при условии* t *будет равняться:*

$$\bar{U}^1(\mathbf{X}|\mathbf{t}) = \bar{U}^0(\mathbf{X}) + \frac{\deg_O(v_{ij})}{n} + \frac{\deg_F(v_{ij})}{m} - \sum_{e_{ij,ij_2} \in \mathbf{E}} \frac{\deg_F(v_{ij_2}) + 1}{mn} - \sum_{e_{ij,i_2 j} \in \mathbf{E}} \frac{\deg_O(v_{i_2 j}) + 1}{mn}.$$

Доказательство. Всего существует $\deg_O(v_{ij})$ пропусков в объекте $\mathbf{x}_i$ и $\deg_F(v_{ij})$ пропусков в объектах в признаке j, не считая пропуск u_{ij}. Суммируя аппроксимированную устойчивость всех пропусков и группируя значения функции δ по пропускам в объекте i и в признаке j, получаем требуемое равенство. ■

4 Формализация рассматриваемого алгоритма

Формализуем полученный алгоритм. Для дальнейшего описания алгоритма введем понятие разрешимого пропуска, т. е. пропуска, который может быть восстановлен алгоритмом. В терминах предложенного алгоритма данное понятие определяется следующим образом.

Вход: Выборка $\mathbf{X}$ с пропущенными значениями; число соседей k; длина аппроксимации b;
Выход: Восстановленная выборка $\hat{\mathbf{X}}$;

1: **пока** множество разрешимых пропусков $\mathbf{R}$ не пусто
2: $x_{ij} = \underset{x_{i_1 j_1}, \bar{\mathbf{t}}_2, \ldots, \bar{\mathbf{t}}_b}{\arg\max} \bar{U}^b(\mathbf{X}|(i_1, j_1), \bar{\mathbf{t}}_2, \ldots, \bar{\mathbf{t}}_b))$;
3: Для пропуска x_{ij} получить соседей $\mathbf{N}$;
4: Упорядочить $\mathbf{N}$ в лексикографическом порядке по устойчивости восстановления x_{ij} и расстоянию до $\mathbf{x}_i$;
5: Получить первые k объектов $\mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k}$ из упорядоченного множества соседей;
6: $\hat{\mathbf{x}}_i = \mathbf{x}_i \leftarrow (x_{q_1 j}, \ldots, x_{q_k j})$;

Рис. 3 Псевдокод предложенного алгоритма восстановления

Определение 18. *Пропуск x_{ij} является разрешимым, если существуют объекты $\mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k}$, каждый из которых имеет непустой признак j, а также хотя бы один общий заполненный признак с $\mathbf{x}_i$, т. е.* $\text{filled}_O(\mathbf{x}_i, \mathbf{X}) \cap \text{filled}_O(\mathbf{x}_{q_r}, \mathbf{X}) \neq \varnothing, r \in \{1, \ldots, k\}$.

Из определения разрешимого пропуска следует, что для каждого разрешимого пропуска x_{ij} существуют объекты $\mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k}$ такие, что устойчивость операции восстановления данного x_{ij} по этим объектам больше нуля:

$$u(x_{ij} \leftarrow (\mathbf{x}_{q_1}, \ldots, \mathbf{x}_{q_k})) > 0\,.$$

Пусть задано число соседей k и длина аппроксимации b. На каждой итерации алгоритма будем отбирать множество разрешимых пропусков $\mathbf{R}$.

Из множества разрешимых пропусков выберем такой пропуск x_{ij}, для которого b-аппроксимированная устойчивость пропуска максимальна, т. е. $x_{ij} = \underset{x_{i_1 j_1}, \bar{\mathbf{t}}_2, \ldots, \bar{\mathbf{t}}_b}{\arg\max} \bar{U}^b(\mathbf{X}|(i_1, j_1), \bar{\mathbf{t}}_2, \ldots, \bar{\mathbf{t}}_b))$.

Для полученного пропуска x_{ij} получим всех соседей, т. е. такие объекты $\mathbf{N}$, что $\mathbf{x}_{qj} \neq \square$ и $\text{filled}_O(\mathbf{x}_q) \cap \text{filled}_O(\mathbf{x}_q) \neq \varnothing$, где $\mathbf{x}_q \in \mathbf{N}$.

Упорядочим объекты из $\mathbf{N}$ в лексиографическом порядке по устойчивости восстановления x_{ij} и расстоянию до $\mathbf{x}_i$:

$$\mathbf{x}_{q_1} \prec \mathbf{x}_{q_2}, \text{ если } \left[\begin{array}{c} u(x_{ij} \leftarrow \mathbf{x}_{q_1}) > u(x_{ij} \leftarrow \mathbf{x}_{q_2}); \\ u(x_{ij} \leftarrow \mathbf{x}_{q_1}) = u(x_{ij} \leftarrow \mathbf{x}_{q_2}) \text{ и } d(\mathbf{x}_i, \mathbf{x}_{q_1}) < d(\mathbf{x}_i, \mathbf{x}_{q_2}). \end{array}\right.$$

Восстановим пропущенное значение x_{ij} по k первым объектам полученного упорядоченного множества.

Псевдокод представленного алгоритма показан на рис. 3. Сложность алгоритма оценивается как $O(\ell(\ell^{b+1} + km^2))$, что намного больше сложности алгоритма без транзитивного восстановления пропусков.

5 Функции расстояния для разнородных шкал

В данном разделе проводится краткий обзор функций расстояния для различных типов шкал — линейной, порядковой, а также смешанной. Предлагается функция расстояния для выборок, описанных в линейных, номинальных и порядковых шкалах с заданным на них полным порядком.

5.1 Функция расстояния для линейных шкал

Рассмотрим обобщенную функцию расстояния для множества объектов с введенной линейной шкалой:

$$r(\mathbf{x}_i, \mathbf{x}_q) = \left(\left(|\mathbf{x}_i - \mathbf{x}_q|^p \right)^{\mathrm{T}} \mathbf{S}^{-1} |\mathbf{x}_i - \mathbf{x}_q|^p \right)^{1/(2p)},$$

где p — некоторое число; $\mathbf{S}$ — симметричная неотрицательно определенная матрица, например единичная матрица $\mathbf{I}$, а возведение вектора в степень понимается как покомпонентное возведение, т. е. $\mathbf{x}^p = (x_1^p, \ldots, x_n^p)$.

В табл. 1 представлены соответствия представленной функции различным именным функциям расстояния при фиксированных параметрах.

Таблица 1 Соответствие функции расстояния именным функциям расстояния

p	$\mathbf{S}$	Название функции	Формула
1	—	Расстояние Махаланобиса	$r(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_q)^{\mathrm{T}} \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_q)}$
—	$\mathbf{I}$	Расстояние Минковского	$r(\mathbf{x}_i, \mathbf{x}_q) = \left(\sum_{k=j}^{n} |x_{ij} - x_{qj}|^q \right)^{1/q}, q = 2p$
1	$\mathbf{I}$	Евклидова Метрика	$r(\mathbf{x}_i, \mathbf{x}_q) = \sqrt{\sum_{j=1}^{n} (x_{ij} - x_{qj})^2}$
0,5	$\mathbf{I}$	Расстояние городских кварталов	$r(\mathbf{x}_i, \mathbf{x}_q) = \sum_{j=1}^{n} |x_{ij} - x_{qj}|$
$+\infty$	$\mathbf{I}$	Расстояние Чебышёва	$r(\mathbf{x}_i, \mathbf{x}_q) = \max\limits_{j=1\ldots n} (x_{ij} - x_{qj})$

5.2 Функция расстояния для порядковых шкал

Введем матричные функции $\mathbf{H}^{q+}$ и $\mathbf{H}^{q-}$ для проекции множества объектов $\mathbf{X}$ на q-й признак, где соответствующая шкала $\mathbb{L}_q$ — порядковая. Каждая компонента вектора $\mathbf{H}_i^{q+}$ определяет отношение порядка q-го признака i-го объекта с остальными объектами выборки:

$$\left(\mathbf{H}_i{}^{j+}\right)_q = \begin{cases} 1, & \text{если } x_{ij} \succ x_{qj}; \\ 0 & \text{иначе;} \end{cases}$$

$$\left(\mathbf{H}_i{}^{j-}\right)_q = \begin{cases} 1, & \text{если } x_{qj} \succ x_{ij}; \\ 0 & \text{иначе.} \end{cases}$$

Так как $||\mathbf{H}_j^{q+}||_2^2 + ||\mathbf{H}_j^{q-}||_2^2 \leqslant m$, введем функцию расстояния pdist:

$$\mathrm{pdist}(x_{iq}, x_{qj}) = \frac{m - \left(\langle \mathbf{H}_i{}^{j+}, \mathbf{H}_q{}^{j+} \rangle + \langle \mathbf{H}_i{}^{j-}, \mathbf{H}_q{}^{j-} \rangle \right)}{m}, \tag{2}$$

где m — множество объектов в выборке. Функция принимает значения из диапазона $[0;1]$.

5.3 Обобщение функции расстояния HEOM

Дополним функцию HEOM [20] для случая объектов, описанных как в номинальных и линейных шкалах, так и в порядковых шкалах с полным порядком:

$$d(\mathbf{x}_i, \mathbf{x}_q) = \frac{1}{\sqrt{n}} \left(\sum_{j=1}^{n} r(x_{ij}, x_{qj})^2 \right)^{1/2}. \tag{3}$$

Здесь

$$r(x_{ij}, x_{qj}) = \begin{cases} \text{overlap}(x_{ij}, x_{qj}), & \text{если } \mathbb{L}_j \text{ — номинальный признак;} \\ \text{pdist}(x_{ij}, x_{qj}), & \text{если } \mathbb{L}_j \text{ — порядковый признак;} \\ \text{diff}(x_{ij}, x_{qj}) & \text{иначе,} \end{cases}$$

где

$$\text{overlap}(x_{ij}, x_{qj}) = \begin{cases} 1, & \text{если } x_{ij} \neq x_{qj}; \\ 0 & \text{иначе;} \end{cases}$$

$$\text{diff}(x_{ij}, x_{qj}) = \frac{|x_{ij} - x_{qj}|}{\max\limits_{\mathbb{L}_j} - \min\limits_{\mathbb{L}_j}},$$

т. е. функция $\text{diff}(x_{ij}, x_{qj})$ определяется как нормированный модуль разницы между значениями j-го признака двух объектов.

Таким образом, мы получили метрику для смешанных шкал. Функция d принимает значения из отрезка $[0; 1]$.

6 Вычислительный эксперимент

Основной целью вычислительного эксперимента является определение границы применимости предложенного метода. С этой целью было проведено два эксперимента. В обоих экспериментах в качестве исходных данных использовалась выборка кредитозаемщиков Германии [2]. В выборке присутствует 1000 объектов и 21 признак в линейных, номинальных и порядковых шкалах. В каждом эксперименте производилась генерация подвыборки мощностью 100 объектов и добавление в нее пропущенных значений, при этом не допускалось такое добавление пропусков, при котором какой-либо объект имел бы пустое описание. Исходный код экспериментов доступен по адресу [21].

В первом эксперименте исследовалось количество неразрешимых пропусков при использовании транзитивного восстановления и без. Результаты данного эксперимента показаны на рис. 4. По оси Y отложен процент неразрешимых пропусков, по оси X — процент добавленных пропусков. Было проведено 40 запусков, результат был усреднен. Как видно из результатов, оба алгоритма могут разрешить все пропуски при достаточно большом проценте пропущенных значений.

Во втором эксперименте исследовалась эффективность рассматриваемого алгоритма восстановления пропусков. В качестве критерия ошибки Q использовалось среднее расстояние от реальных объектов до восстановленных вариантов:

$$Q = \sum_{\mathbf{x}_i \in \mathbf{X}, \exists j : \hat{\mathbf{x}}_{ij} = \square} d(\mathbf{x}_i, \hat{\mathbf{x}}_i) \cdot \frac{1}{R},$$

где $\hat{\mathbf{x}}_i$ — объект, восстановленный методом k ближайших соседей, $k = 1$; R — количество объектов, имеющих пропущенные значения.

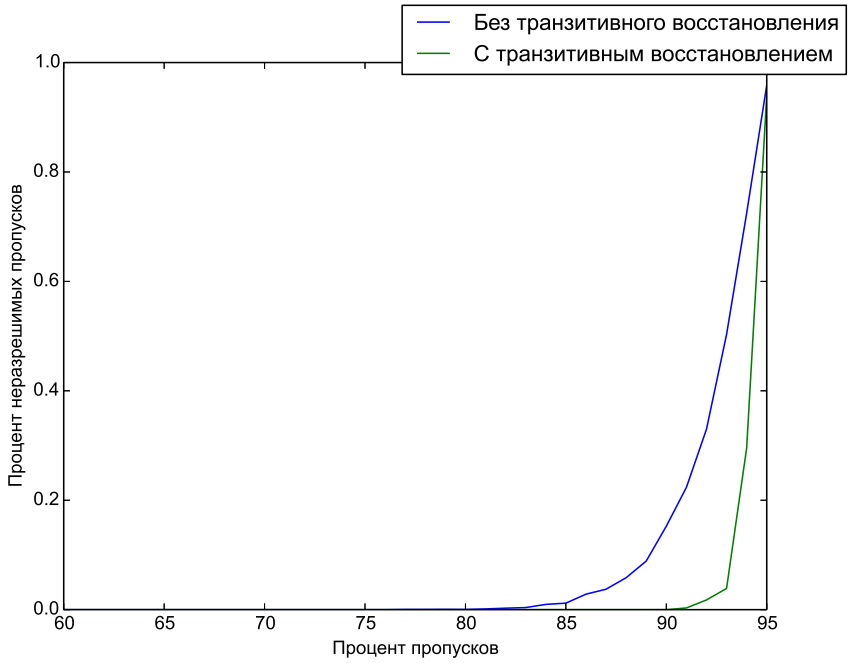


Рис. 4 Результаты первого эксперимента

Результаты данного эксперимента показаны на рис. 5. По оси X отложен процент добавленных пропусков, по оси Y — средняя ошибка восстановления. Было проведено 20 запусков, результат был усреднен. В эксперименте рассматривались алгоритм восстановления пропусков по k ближайшим соседям без транзитивного восстановления, итеративная версия алгоритма, описанная в [12], 0- и 1-аппроксимации, а также восстановление пропущенных значений средним и алгоритм восстановления с использованием дерева решений. В качестве критерия остановки итеративной версии алгоритма использовалось правило:

$$S = [\text{mean}_{\mathbf{x}\in\mathbf{X}}(d(\hat{\mathbf{x}}^u, \hat{\mathbf{x}}^{u+1}) < 0{,}01)]\,,$$

где $\hat{\mathbf{x}}^u$ — объект, восстановленный на итерации u. Как видно из результатов, наилучший результат был показан алгоритмом восстановления пропусков с использованием дерева решений. 0- и 1-аппроксимации показали результат, близкий к исходному алгоритму без транзитивного восстановления, при этом 1-аппроксимация в целом оказалась менее эффективна, чем 0-аппроксимация.

7 Заключение

В работе была рассмотрена проблема восстановления пропущенных значений в разнородных шкалах в случае значительного количества пропусков. Для формализации рассмотренной проблемы было введено понятие устойчивости восстановления пропуска и устойчивости восстановления выборки. Были рассмотрены варианты алгоритма заполнения пропусков по k ближайшим соседям, а также теоретические аспекты их применимости. Для оценки качества рассмотренных алгоритмов был проведен вычислительный эксперимент со сравнением данных алгоритмов с заполнением средними значениями и алгоритмом заполнения по дереву решений. Эксперимент показал, что наилучший результат достигается алгоритмом заполнения с использованием дерева решений.

Автор выражает благодарность д. ф.-м. н. Вадиму Викторовичу Стрижову за постановку задачи и внимание к работе.

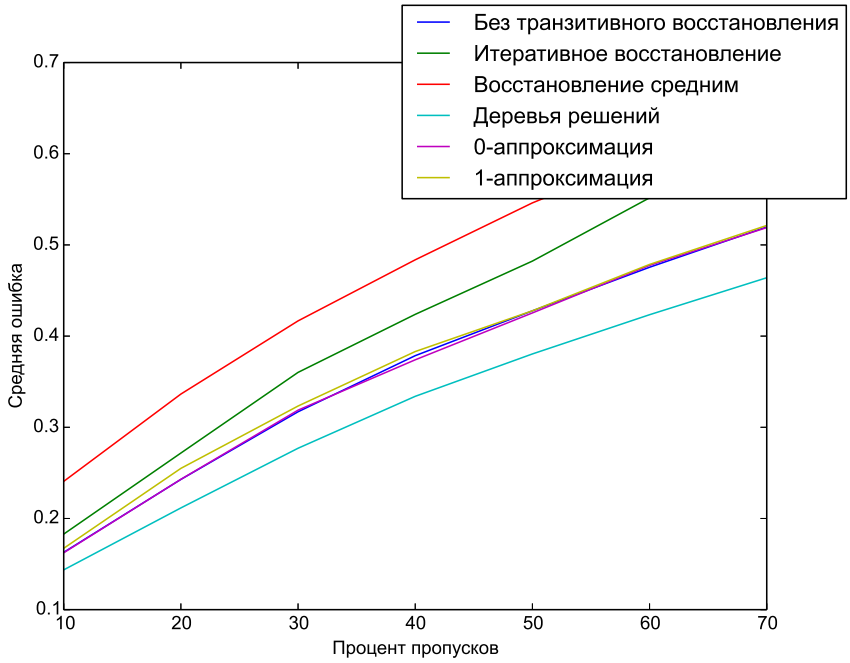


Рис. 5 Результаты второго эксперимента

Литература

[1] Horse Colic Data Set. `https://archive.ics.uci.edu/ml/datasets/Horse+Colic`.

[2] `http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/`.

[3] *Saar-Tsechansky M., Provost F.* Handling missing values when applying classification models // J. Machine Learning Res. Arch., 2007. Vol. 8. P. 1623–1657.

[4] *Sharpe P. K., Solly R. J.* Dealing with missing values in neural network-based diagnostic systems // Neural Comput. Appl., 1995. Vol. 3. No. 2. P. 73–77.

[5] *Saunders J. A., Morrow-Howell N., Spitznagel E., Dori P., Proctor E. K., Pescarino R.* Imputing missing data: A comparison of methods for social work researchers // Social Work Res., 2006. Vol. 30. No. 1. P. 19–31.

[6] *Batista G., Monard M. C.* A study of k-nearest neighbour as an imputation method // 2nd Conference (International) on Hybrid Intelligent Systems Proceedings. Santiago, Chile: IOS Press, 2002. P. 251–260.

[7] *Durrant G. B.* Imputation methods for handling item-nonresponse in the social sciences: A methodological review, 2005. http://eprints.ncrm.ac.uk/86/1/MethodsReviewPaperNCRM-002.pdf.

[8] *Marlin B. M.* Missing data problems in machine learning. — Toronto: University of Toronto, 2008. PhD Thesis. 164 p.

[9] *Shapfer J.L.* Multiple imputation: A primer // Stat. Meth. Medical Res., 1999. Vol. 81. No. 1. P. 3–15.

[10] *Bouhlila D. S., Sellaouti F.* Multiple imputation using chained equations for missing data in TIMSS: A case study // Large-Scale Assessments in Education, 2013. Vol. 1. No. 4.

[11] *Acuna E., Rodrigez C.* The treatment of missing values and its effect on classifier accuracy // Classification, clustering and data mining applications. — Berlin–Heidelberg: Springer-Verlag, 2004. P. 639–648.

[12] *Bras L.G., Menezes J.C.* Improving cluster-based missing value estimation of DNA microarray data // Biomol. Eng., 2007. Vol. 24. No. 2. P. 273–282.

[13] *Eskelson B. N. I., Temesgen H., Lemay V., Barrett T. M.,Crookston N. L., Hudak A. T.* The roles of nearest neighbor methods in imputing missing data in forest inventory and monitoring databases // Scand. J. Forest Res., 2009. Vol. 24. No. 3. P. 235–246.

[14] *Pan L.* k-Nearest neighbor based missing data estimation algorithm in wireless sensor networks // Wireless Sensor Network, 2010. Vol. 2. No. 2. P. 115–122.

[15] *Lim J. K., Fuller W. A., Bell W. R.* Variance estimation for nearest neighbor imputation for US Census long form data // Annal. Appl. Stat., 2011. Vol. 5. No. 2A. P. 824–842.

[16] *Jonsson P., Wohlin C.* An evaluation of k-nearest neighbour imputation using Likert data // 10th Symposium (International) on Software Metrics Proceedings, 2004. P. 108–118.

[17] *Kuznetsov M.P., Strijov V.V.* Methods of expert estimations concordance for integral quality estimation // Expert Syst. Appl., 2015. Vol. 41. No. 4. P. 1988–1996.

[18] *Stenina M. M., Kuznetsov M. P., Strijov V. V.* Ordinal classification using Pareto fronts // Expert Syst. Appl., 2015. Vol. 42. No. 14. P. 5947–5953.

[19] *Cooke D. J., Bez H. E.* Computer mathematics. — 1st ed. — Cambridge University Press, 1984. 408 p.

[20] *Wilson D. R., Martinez T. R.* 1997. Improved heterogeneous distance functions // J. Artif. Intell. Res., Vol. 6. P. 1–34.

[21] `http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Bakhteev2014MissData/source/.`

References

[1] Horse Colic Data Set. Available at: `https://archive.ics.uci.edu/ml/datasets/Horse+Colic` (accessed June 21, 2015).

[2] Available at: `http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/` (accessed June 21, 2015).

[3] Saar-Tsechansky, M., and F. Provost. 2007. Handling missing values when applying classification models. *J. Machine Learning Res. Arch.* 8:1623–1657.

[4] Sharpe, P. K., and R. J. Solly. 1995. Dealing with missing values in neural network-based diagnostic systems. *Neural Comput. Appl.* 3(2):73–77.

[5] Saunders, J. A., N. Morrow-Howell, E. Spitznagel, P. Dori, E. K. Proctor, and R. Pescarino. 2006. Imputing missing data: A comparison of methods for social work researchers. *Social Work Res.* 30(1):19–31.

[6] Batista, G., and M. C. Monard. 2002. A study of k-nearest neighbour as an imputation method. *2nd Conference (International) on Hybrid Intelligent Systems Proceedings.* Santiago, Chile: IOS Press. 251–260.

[7] Durrant, G. B. 2005. Imputation methods for handling item-nonresponse in the social sciences: A methodological review. Available at: http://eprints.ncrm.ac.uk/86/1/MethodsReviewPaperNCRM-002.pdf (accessed October 15, 2015).

[8] Marlin, B. M. 2008. Missing data problems in machine learning. Toronto: University of Toronto. PhD Thesis. 164 p.

[9] Shapfer, J. L. 1999. Multiple imputation: A primer. *Stat. Meth. Medical Res.* 81(1):3–15.

[10] Bouhlila, D. S., and F. Sellaouti. 2013. Multiple imputation using chained equations for missing data in TIMSS: A case study. *Large-Scale Assessments in Education* 1(4).

[11] Acuna, E., and C. Rodrigez. 2004. The treatment of missing values and its effect in the classifier accuracy. *Classification, clustering and data mining applications*. Berlin–Heidelberg: Springer-Verlag. 639–648.

[12] Bras, L. G., and J. C. Menezes. 2007. Improving cluster-based missing value estimation of DNA microarray data. *Biomol. Eng.* 24(2):273–282.

[13] Eskelson, B. N. I., H. Temesgen, V. Lemay, T. M. Barrett, N. L. Crookston, and A. T. Hudak. 2009. The roles of nearest neighbor methods in imputing missing data in forest inventory and monitoring databases. *Scand. J. Forest Res.* 24(3):235–246.

[14] Pan, L. 2010. k-Nearest neighbor based missing data estimation algorithm in wireless sensor networks. *Wireless Sensor Network* 2(2):115–122.

[15] Lim, J. K., W. A. Fuller, and W. R. Bell. 2011. Variance estimation for nearest neighbor imputation for US Census long form data. *Annal. Appl. Stat.* 5(2A):824–842.

[16] Jonsson, P., and C. Wohlin. 2004. An evaluation of k-nearest neighbour imputation using Likert data. *10th Symposium (International) on Software Metrics Proceedings*. 108–118.

[17] Kuznetsov, M. P., and V. V. Strijov. 2015. Methods of expert estimations concordance for integral quality estimation. *Expert Syst. Appl.* 41(4):1988–1996.

[18] Stenina, M. M., M. P. Kuznetsov, and V. V. Strijov. 2015. Ordinal classification using Pareto fronts. *Expert Syst. Appl.* 42(14):5947—5953.

[19] Cooke, D. J., and H. E. Bez. 1984. *Computer mathematics*. 1st ed. Cambridge University Press. 408 p.

[20] Wilson, D. R., and T. R. Martinez. 1997. Improved heterogeneous distance functions. *J. Artif. Intell. Res.* 6:1–34.

[21] Available at: `http://sourceforge.net/p/mlalgorithms/code/HEAD/tree/Group074/Bakhteev2014MissData/source/` (accessed June 21, 2015).

Статистическое распознавание образов на основе посегментного анализа однородности*

А. В. Савченко
avsavchenko@hse.ru
НИУ Высшая школа экономики, Нижний Новгород, Россия

Исследуется проблема малых выборок в задаче статистического распознавания образов на основе методов ближайших соседей, точность которых во многом определяется выбранной мерой близости, при этом их реализация в режиме реального времени может оказаться невозможной уже при наличии тысяч классов. Для преодоления указанных проблем предложен новый подход к разработке классификаторов с посегментным анализом однородности и быстрой последовательной иерархической обработкой на основе вероятностной модели кусочно-однородного объекта. Экспериментальные исследования в задаче распознавания лиц продемонстрировали повышение точности на 1%–10% по сравнению с традиционными методами (SVM, SIFT, LBP, собственные лица). Вычислительная эффективность оказалась в 2–3 раза выше по сравнению с известным методом Pyramid HOG (Hystograms of Oriented Gradients). Показано, что описанная методология посегментного анализа однородности характеризуется высокой точностью и приемлемой производительностью для случая малых выборок и большого числа классов.

Ключевые слова: *статистическое распознавание образов; иерархическая классификация; методы приближенного поиска ближайшего соседа; классификация изображений; гистограммы ориентированных градиентов*

Statistical pattern recognition based on segment homogeneity testing*

A. V. Savchenko
National Research University Higher School of Economics, N. Novgorod, Russian Federation

Background: This paper is focused on a small-sample size problem in statistical recognition of audiovisual objects with the nearest neighbor method. Its accuracy depends on the applied similarity measure. Moreover, the computing efficiency is insufficient if thousand of classes are available.

Methods: The author introduces an approach to design classifiers of audiovisual objects by testing of segment homogeneity based on the probabilistic model of composite object represented by a sequence of independent identically distributed segments. The asymptotic properties of this approach allow to implement sequential hierarchical classification with approximate search of the nearest neighbor to speed up the decision process.

Results: Experimental study in constrained face recognition with HOG features shows that the proposed approach allows to increase accuracy in 1%–10% in comparison with conventional image recognition techniques (k-NN, SVM, SIFT, histogram of local binary patterns, eigenfaces). Moreover, it is 2–3 times faster than the pyramid HOG (Hystograms of Oriented Gradients) hierarchical classifier.

*Исследование выполнено в Национальном исследовательском университете «Высшая школа экономики» за счет средств гранта Российского научного фонда (проект № 14-41-00039)

Concluding Remarks: Described methodology with segment homogeneity testing allows to achieve high accuracy with sufficient performance in case of small-sample-size and medium-sized number of classes.

Keywords: *statistical pattern recognition; hierarchical classification; approximate nearest neighbor methods; image classification; Histograms of Oriented Gradients (HOG)*

1 Введение

Традиционный подход к организации многих систем обработки аудиовизуальной информации состоит в последовательном соединении модулей извлечения признаков, классификации и управления [1, 2]. Попытки построить модели и методы, адекватно описывающие решение человеком таких плохо формализованных задач до настоящего момента не привели к значимым практическим результатам [3]. Поэтому прикладные исследования в области синтеза таких систем в основном сосредоточились на разработке методов принятия решений в модуле классификации, в котором входному объекту X ставится в соответствие один из $C > 1$ заранее точно не определенных классов. Предполагается, что для обучения системы доступна база данных, содержащая $R \geqslant C$ эталонных объектов (прецедентов) $\{X_r\}, r \in \{1, \ldots, R\}$, где класс каждого r-го эталона $c(r) \in \{1, \ldots, C\}$ считается известным [1]. В настоящее время считается, что, несмотря на существенные отличия в процедурах извлечения признаков изображений и речевых сигналов, для решения все более сложных задач промышленные системы классификации следует рассматривать в более широком смысле и учитывать доступную структурную и семантическую информацию [4]. На начальном уровне обработки входной объект X и все эталоны X_r разбиваются на несколько сегментов, каждый из которых описывается множеством одинаково распределенных значений признаков, а в новом модуле структурного распознавания осуществляется посегментный анализ результатов классификации сегментов на более низких уровнях. Такой подход является достаточно универсальным в связи с наличием хорошо изученных алгоритмов сегментации как аудио, так и визуальной информации. К сожалению, применение известных способов реализации такого посегментного анализа, основанных на скрытых Марковских моделях [5], возможно лишь в том случае, если для обучения доступна репрезентативная база данных большого объема (сотни эталонов для каждого класса, $R \gg C$). Такое ограничение оказывается слишком жестким для многих промышленных систем, в которых проявляется проблема малых выборок ($R \approx C$): число эталонов для каждого класса недостаточно для обучения сложного классификатора [6]. Проблема усиливается в характерном именно для задач обработки аудиовизуальной информации случае наличия помех как на этапе формирования базы данных эталонов, так и при наблюдении входного объекта X. В случае малых выборок для решения обычно применяются методы ближайшего соседа, точность которых определяется используемой мерой близости [1]. Существующая теория не дает строгого ответа на вопрос о выборе оптимальной меры близости, поэтому на практике исследователи полагаются на свой опыт, что, разумеется, далеко не всегда приводит к наилучшему результату. Более того, в связи с большой размерностью пространства признаков для описания аудиовизуальной информации, реализация такого подхода, основанного на полном переборе базы данных эталонов, в режиме (мягкого) реального времени оказывается во многих промышленных системах невозможной уже при наличии сотни классов [7]. Кажется, что для ускорения процедуры поиска ближайших соседей можно использовать известные приближенные алгоритмы, например рандомизированные kd-деревья [8] или perm-sort [9]. К сожалению,

такие методы разрабатываются специально для автоматизированных систем поиска объектов по содержанию из сверхбольших баз данных (десятки и сотни тысяч классов), поэтому их применение не приводит к существенному снижению времени классификации, если число классов не превышает несколько тысяч единиц.

Со всех перечисленных точек зрения повышенный интерес представляет описанный в работе [10] подход к решению задачи групповой классификации [11], в котором за счет объединения входной и эталонной выборок повышается точность оценки распределения каждого класса для случая малых размерностей эталонных выборок и/или при наличии в них шума. Применению такого подхода при анализе однородности сегментов входного и эталонного объектов для преодоления указанной проблемы недостаточной точности и вычислительной эффективности современных промышленных систем обработки и классификации аудиовизуальной информации при наличии в базе данных большого числа классов (тысячи альтернатив) и малого количества эталонов для каждого класса ($R \approx C$) и посвящена настоящая статья.

2 Посегментный анализ однородности в задаче классификации составных объектов

Прежде всего, сформулируем следующий целевой критерий (ЦК) оценки эффективности методов классификации:

$$\overline{\alpha}_\eta \to \min; \qquad \text{s.t.}\, t_{\text{avg}} < t_0, \tag{1}$$

согласно которому наилучший классификатор характеризуется минимальной *средней вероятностью ошибки* $\overline{\alpha}_\eta$, где для контроля устойчивости к тестовым данным добавляются помехи η с дисперсией $\sigma_\eta^2 \geqslant 0$. При этом накладывается ограничение на максимально допустимое среднее время принятия решения t_{avg}, которое не может превышать фиксированный порог t_0. Проверка такого условия может производиться как с помощью инструментария теории алгоритмов (О-оценок), так и непосредственным сопоставлением среднего времени классификации для различных методов в рамках натурных испытаний. Таким образом, ЦК (1) соответствует Парето-оптимальному решению задачи многокритериальной оптимизации с одновременной минимизацией средней вероятности ошибки и среднего времени классификации. Выражение (1) фактически устанавливает частичный порядок на множестве всех методов классификации. Из двух классификаторов, которые удовлетворяют требованию к среднему времени отклика $t_{\text{avg}} < t_0$, наилучшим является тот, который характеризуется наименьшей средней вероятностью перепутывания $\overline{\alpha}_\eta$. Если один из двух классификаторов в силу своей вычислительной сложности не может быть реализован в режиме «мягкого» реального времени ($t_{\text{avg}} \geqslant t_0$), то второй классификатор будет являться более предпочтительным, даже если его точность оказывается ниже точности первого метода. Наконец, если два метода характеризуются неприемлемым временем принятия решения, то с точки зрения практической реализации оба таких классификатора считаются неудовлетворительными.

Для решения задачи распознавания входной объект X с помощью известной процедуры сегментации (например, разбиения на фреймы фиксированной размерности или наращивания областей) [5, 12] представляется в виде последовательности из K сегментов $X(k)$, каждому из которых ставится в соответствие совокупность из $n(k)$ векторов значений признаков $\mathbf{x}_j(k), j \in \{1, \ldots, n(k)\}$ фиксированной размерности M. Аналогично эталоны X_r разбиваются на K_r частей и k-й сегмент r-го эталона определяется как последовательность векторов значений признаков $\mathbf{x}_j^{(r)}(k), j \in \{1, \ldots, n_r(k)\}$, размерности M в k-м

сегменте r-го эталона, а $n_r(k)$ — число признаков. В рамках модели кусочно-однородного объекта [13] предполагается, что векторы $\mathbf{x}_j(k)$ и $\mathbf{x}_j^{(r)}(k)$ являются реализациями случайных векторов, сегменты $X(k)$ — простые выборки (независимых одинаково распределенных) векторов $\mathbf{x}_j(k)$, а эталонные сегменты $X_r(k)$ — простые выборки векторов $\mathbf{x}_j^{(r)}(k)$. Чтобы учесть недостаточную точность детектирования границ, все сегменты $X(k)$ рассматриваются в пределах, близких к k номеров k_r сегментов r-го эталона из множества $N_r(k)$ (окрестность «соседей», определяемая спецификой предметной области). Тогда задача сводится к проверке гипотез $W_r, r \in \{1, \ldots, R\}$, о неизвестном законе распределения значений признаков сегментов объекта X.

Рассмотрим случай *дискретных* признаков с конечным множеством из $N > 1$ значений $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Тогда сегменты $X(k)$ и $X_r(k)$ полностью описываются своими гистограммами $\{w_i(k)\}$ и $\{\theta_i^{(r)}(k)\}, i \in \{1, \ldots, N\}$ соответственно. В таком случае можно показать [14], что для применяемой в известной вероятностной нейронной сети (ВНС) [15] непараметрической оценки неизвестных распределений сегментов с помощью ядра Розенблатта–Парзена $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ решение задачи с помощью байесовского критерия минимума среднего риска сводится к следующему критерию:

$$\nu = \arg\min_{r \in \{1,\ldots,R\}} \left(\frac{1}{Kn} \sum_{k=1}^{K} \min_{k_r \in N_r(k)} \sum_{i=1}^{N} w_i(k) \ln \frac{w_{K;i}(k)}{\theta_{K;i}^{(r)}(k_r)} - \ln p_r \right). \tag{2}$$

Здесь $\theta_{K;i}^{(r)}(k_r) = \sum_{j=1}^{N} K_{ij} \theta_j^{(r)}(k_r)$; $w_{K;i}^{(r)}(k) = \sum_{j=1}^{N} K_{ij} w_j^{(r)}(k)$ — свертки с ядром гистограмм r-й эталонной $X_r(k_r)$ и входной выборок $X(k)$; p_r — априорная вероятность появления класса, определяемого r-м эталоном; $n = \sum_{k=1}^{K} n(k)/K$ — средняя длительность сегмента.

К сожалению, как известно [10], при подстановке вместо неизвестных распределений их оценок классификатор (2) становится неоптимальным. Для повышения точности в ЦК (1) автором был предложен новый подход к построению классификаторов на основе посегментного анализа однородности (ПАО) признаков входного и эталонного объектов [13], в котором задача сводится к проверке сложных гипотез об однородности сегментов $X(k)$ и $X_r(k_r)$ с оценкой распределения класса по объединенной выборке $\{X(k), X_r(k_r)\}$. Тогда следующий критерий

$$\nu = \arg\min_{r \in \{1,\ldots,R\}} (\rho_{PNNH}(X, X_r) - \ln p_r) \tag{3}$$

является при $n(k) \to \infty$ асимптотически минимаксным для проверки однородности объектов X и r-го эталона. Здесь

$$\rho_{\text{PNNH}}(X, X_r) = \frac{1}{Kn} \sum_{k=1}^{K} \min_{k_r \in N_r(k)} \sum_{i=1}^{N} \left(w_i(k) \ln \frac{w_{K;i}(k)}{\widetilde{\theta}_{\Sigma;i}^{(r)}(k; k_r)} + \frac{n_r(k_r)}{n(k)} \theta_i^{(r)}(k) \ln \frac{\theta_{K;i}^{(r)}(k)}{\widetilde{\theta}_{\Sigma;i}^{(r)}(k; k_r)} \right) \tag{4}$$

есть выход синтезированной в работе [14] ВНС с ПАО, а $\widetilde{\theta}_{\Sigma;i}^{(r)}(k; k_r) = (n(k) w_{K;i}(k) + n_r(k_r) \theta_{K;i}^{(r)}(k_r))/(n(k) + n_r(k_r))$.

Если ядром $K_i j$ является дискретная дельта-функция $\delta(i - j)$, то $\theta_{K;i}^{(r)}(k_r) = \theta_i^{(r)}(k_r)$, $w_{K;i}(k) = w_i(k)$, тогда выражение (2) будет эквивалентно известному принципу минимума информационного рассогласования Кульбака–Лейблера. А если дополнительно предположить, что размеры всех сегментов одинаковы, то рассогласование (4) будет эквивалентно дивергенции Йенсена–Шеннона и, при использовании известной аппроксимации логарифма $\ln(a/b) \approx (a^2 - b^2)/(2ab)$, расстоянию χ^2. Как видно, применение подхода на основе

посегментного анализа однородности позволяет получить не только новые, но и как частный случай хорошо зарекомендовавшие себя на практике критерии.

3 Способы повышения вычислительной эффективности алгоритма классификации с посегментным анализом однородности

Вычислительная сложность критерия (3) оценивается как $O\left(N\sum_{r=1}^{R}\sum_{k=1}^{K}|N_r(k)|\right)$, где $|N_r(k)|$ — число элементов в множестве $N_r(k)$. Как и в любой реализации метода ближайшего соседа, необходимость полного перебора всех эталонов из базы данных может сделать его физически нереализуемым для больших R. Поэтому в настоящем разделе рассмотрены способы повышения вычислительной эффективности критерия (2) на основе посегментного анализа однородности, если они не удовлетворяют ЦК (1) при малом значении порога t_0. Прежде всего, опишем последовательный способ принятия решений, в котором проводится анализ более детализированного уровня описания аудиовизуальной информации только при получении недостаточно надежных решений на предыдущих, более простых уровнях. На первом уровне такой иерархической системы анализируются наиболее грубые приближения входного и эталонных объектов, например с малым числом однородных сегментов K. Запишем критерий (3) в упрощенной форме [7]:

$$W_\nu : \rho_{\mathrm{PNNH}}(X, X_\nu) < \rho_0\,. \tag{5}$$

Для заданной вероятности p_{FRR} ложного пропуска объекта порог ρ_0 устанавливается равным p_{FRR}-квантилю множества расстояний между разноименными объектами. Если условие (5) не было выполнено ни для одного из эталонов, то для выделения входных объектов, не принадлежащих ни одному из эталонных классов, применяется критерий сравнения минимального рассогласования с фиксированным порогом:

$$\rho_{\mathrm{PNNH}}(X, X_\nu) > \rho_1\,. \tag{6}$$

Здесь порог ρ_1 для заданной вероятности p_{FAR} ложного срабатывания устанавливается равным $(1-p_{\mathrm{FAR}})$-квантилю множества рассогласований между одноименными объектами из одного класса. Если условие (6) не выполняется, для проверки надежности решения (3) воспользуемся оптимальным в байесовском смысле правилом Чоу [16]:

$$P(W_\nu \,|\, X) > p_0\,, \tag{7}$$

где значение порога p_0 можно выбрать, зафиксировав вероятность ошибки ложного срабатывания. В работе [13] показано, что для предложенного метода с ПАО (3), апостериорная вероятность $P(W_\nu|X)$ гипотезы W_ν оценивается как

$$\widehat{P}(W_\nu \,|\, X) = \frac{p_\nu \exp(-nK\rho_{\mathrm{PNNH}}(X, X_\nu))}{\sum_{r=1}^{R} p_r \exp(-nK\rho_{\mathrm{PNNH}}(X, X_r))}\,. \tag{8}$$

Если для заданного уровня иерархии можно получить достаточно надежное решение (7), то процесс поиска на нем и останавливается. В противном случае, описание входного объекта детализируется (что приводит к увеличению числа однородных сегментов K), и процесс принятия решений (5)–(7) повторяется заново до тех пор, пока не будет получено надежное решение (7) или время обработки не превысит порог t_0 (1). Обычно количество шагов ограничивается исследователем исходя из особенностей конкретной задачи ($J = const$). Если на J-м шаге надежное решение в смысле (7) не было найдено, среди потенциальных решений, полученных на всех J уровнях, выбирается решение с максимальной

апостериорной вероятностью (8). Заметим, что каждый уровень предложенной иерархической трехпороговой системы можно рассматривать как добавление условия (7) к последовательной тернарной классификации (three-way decisions) [17]. Каждый критерий (5)–(7) определяет приближенное множество с переменной точностью [18], которое выражает степень неуверенности исследователя в принятом решении (3) и задается вероятностями ошибок первого и второго рода. При этом все критерии служат для решения своих задач [19]: выражение (5) является основой для методов сокращения перебора, критерий (6) применяется для отбраковки объектов, не принадлежащих ни к одному классу, а выражение (7) позволяет выбрать решения, наиболее различающиеся от остальных эталонов.

В отличие от известных иерархических методов, таких как Pyramid HOG (PHOG) [20], предложенный подход не требует обязательной обработки всех уровней иерархии, что приводит к повышению вычислительной эффективности классификации. Пусть $K^{(j)}$ — число сегментов во входном объекте на j-м уровне детализации. Вычислительная сложность предложенного последовательного иерархического трехпорогового метода классификации в худшем случае получения ненадежных решений на всех J уровнях совпадает со сложностью традиционных иерархических алгоритмов и может быть оценена как $O\left(N\sum_{j=1}^{J}\sum_{r=1}^{R}\sum_{k=1}^{K^{(j)}}|N_r(k)|\right)$. Если предположить, что на каждом уровне иерархии доля ненадежных решений определяется некоторой фиксированной константой $0 \leqslant \gamma < 1$, то сложность предложенного подхода оценивается как $O\left(N\sum_{j=1}^{J}\left(\gamma^{j-1}\sum_{r=1}^{R}\sum_{k=1}^{K^{(j)}}|N_r(k)|\right)\right)$. Если размерность окрестности $N_r(k)$ является константой ($|N_r(k)| = \Delta$), а количество сегментов на $(j+1)$-м уровне иерархии в m раз превышает количество сегментов на j-м уровне (как для метода PHOG [20]), то алгоритмическая сложность разработанного метода для худшего случая $O(N\Delta RK^{(1)}(m^J-1)/(m-1))$, в то время как средняя сложность для доли γ недостаточно надежных решений на каждом уровне оценивается как $O(N\Delta RK^{(1)}((\gamma m)^J-1)/(\gamma m-1))$, т. е. оказывается примерно в $\gamma^{-(J-1)}$ раз быстрее. Например, для средней доли недостаточно надежных решений 20% уже для $J=2$ уровней среднее время классификации для предложенного подхода оказывается в $0{,}2^{-1}=5$ раз ниже по сравнению с более традиционной обработкой всех уровней иерархии.

Для повышения вычислительной эффективности посегментного анализа однородности на каждом уровне иерархии для больших баз данных (несколько тысяч классов) на основе условия досрочного останова (5) может применяться предложенный автором метод максимально правдоподобного направленного перебора (МПНП) [21]. Вначале выбирается один из эталонов X_{r_1} и вычисляется рассогласование $\rho_{\text{PNNH}}(X, X_{r_1})$. Далее последовательно многократно повторяется следующая процедура. Пусть на k-м этапе были проверены эталоны $X_{r_1},\ldots,X_{r_k}$. Следующий эталон $X_{r_{k+1}}$ ищется по принципу максимального правдоподобия (совместного распределения расстояний $\rho_{\text{PNNH}}(X, X_{r_1}),\ldots,\rho_{\text{PNNH}}(X, X_{r_k})$ при справедливости W_ν). Для оценки этого правдоподобия воспользуемся тем фактом, что статистика $nK\rho_{PNNH}(X, X_{r_i})$ асимптотически (при $n\to\infty, n_r\to\infty$) распределена нормально (при больших KN) как

$$\mathcal{N}\left(\rho_{\nu,r_i}+\frac{N-1}{n};\left(\frac{\sqrt{4nK\rho_{\nu,r_i}+2K(N-1)}}{nK}\right)^2\right), \tag{9}$$

где $\rho_{\nu,r_i}=\rho_{\text{PNNH}}(X_\nu, X_{r_i})$.

В работе [21] показано, что в таком случае для выбора следующего эталона следует воспользоваться выражением:

$$r_{k+1} = \underset{\mu\in\{1,\ldots,R\}-\{r_1,\ldots,r_k\}}{\arg\min} \left(\sum_{i=1}^{K} \varphi_\mu(r_i) - \ln p_\mu \right), \tag{10}$$

где в предположении о том, что средний размер сегмента значительно превышает число различных значений признаков (ячеек в гистограмме) $n \gg N$,

$$\varphi_\mu(r_i) = \frac{(\rho_{\mathrm{PNNH}}(X, X_{r_i}) - \rho_{\mu,r_i})^2}{\rho_{\mu,r_i}}. \tag{11}$$

Здесь $\varphi_\mu(r_i)$ тем меньше, чем ближе между собой рассогласования $\rho_{\mathrm{PNNH}}(X, X_{r_i})$ и ρ_{μ,r_i} и чем больше рассогласование ρ_{μ,r_i} между эталонами X_μ и X_{r_i}. При наличии априорной информации о частотах появления каждого класса согласно (10) в первую очередь будут проверяться объекты из классов с большей априорной вероятностью. Если для эталона $X_{r_{k+1}}$ выполняется условие останова (5), то алгоритм завершается на этом этапе $K_{\mathrm{ML-DEM}} = k+1$. В противном случае процедура направленного поиска наиболее правдоподобного (с точки зрения его согласованности с текущими значениями рассогласований) эталона (10), (11) повторяется.

По сравнению с реализацией критерия (3) метод МПНП требует дополнительной памяти для хранения $R^2 + 2R$ вещественных чисел. Алгоритмическая сложность метода составляет $O\left(K_{\mathrm{ML-DEM}}\left(R + (2N/R)\sum_{r=1}^{R}\sum_{k=1}^{K}|N_r(k)|\right)\right)$, что примерно в $R/K_{\mathrm{ML-DEM}}$ раз быстрее полного перебора (3). При наличии параллельной среды выполнения (кластер машин, многоядерный процессор и т. п.) метод МПНП может быть реализован в параллельном варианте [22], в котором все эталоны распределяются на T доступных узлов, после чего на каждом узле применяется МПНП. Узел, первым нашедший эталон, удовлетворяющий условию (5), выдает команду остальным узлам о досрочном останове перебора. В итоге такая параллельная реализация метода МПНП требует хранения $2R + 2M\sum_{r=1}^{R} K_r + R^2/T$ чисел, а вычислительная сложность в расчете на одну задачу может быть оценена как $O\left((K_{\mathrm{ML-DEM}}/T)\left(R/T + (2N/R)\sum_{r=1}^{R}\sum_{k=1}^{K}|N_r(k)|\right)\right)$.

4 Методология классификации в условиях малых выборок и большого числа классов

На рис. 1 представлена предлагаемая методология ПАО для систем классификации в условиях малых выборок и большого числа классов. Здесь для поступающего на вход классифицируемого объекта X осуществляется его предварительная обработка с целью достижения инвариантности к наиболее характерным для предметной области изменениям условий наблюдения и последующая сегментация с помощью существующих методов, хорошо зарекомендовавших себя в конкретной прикладной области, таких как разбиение на небольшие фреймы фиксированной размерности, наращивание областей, выделение контуров и др. [12]. Далее для каждого выделенного сегмента выполняется извлечение примитивных признаков, таких как яркость или направление градиента пикселя (для изображений) [23] или оценки спектральной плотности мощности, кепстральные коэффициенты (для речевых сигналов) [5]. Рассматривая последовательность примитивных признаков как простую случайную выборку из одной (для фиксированного сегмента) генеральной совокупности, на основе описанных выше методов определяется мера близости

Рис. 1 Методология посегментного анализа однородности для систем классификации

входного и эталонных объектов (4) с ПАО и выравниванием сегментов в некоторой окрестности $N_r(k)$, выбираемой, исходя из особенности конкретной задачи и метода сегментирования. Решение принимается в пользу класса, соответствующего ближайшему эталону (3). Для повышения вычислительной эффективности классификации по обучающей выборке оценивается порог для досрочного останова (5), после чего в рамках иерархической трехпороговой системы [19] для поиска приближенного решения применяется метод МПНП [21]. Если для полученного решения условие (5) не выполняется, последовательно (6) и (7) проверяется его надежность. Если решение оказывается недостаточно надежным (7), то можно либо задействовать режим переспроса, либо при наличии времени для уточнения решения повысить степень детализации входного объекта X (например, снизить порог для объединения нескольких небольших фреймов в один сегмент, что приведет к возрастанию количества сегментов K и уменьшению средней размерности сегмента n), после чего процедура (5)–(7) повторяется. Если надежное решение не было получено на всех уровнях иерархии, итоговое решение принимается с помощью комитета классификаторов [1]. В условиях малых выборок последний строится на основе принципа максимума апостериорной вероятности (8). Далее принятое решение может использоваться в составе интеллектуальной системы в рамках существующих методов [1, 4] для дальнейшей обработки (например, сохранения результатов классификации в базе данных [1]) и для автоматического или автоматизированного принятия управленческих решений в составе системы поддержки принятия решений.

Для примера, рассмотрим применение методологии (см. рис. 1) в задаче распознавания изображений. Прежде всего во входном изображении происходит детектирование классифицируемого объекта X с высотой U и шириной V и осуществляется его предварительная обработка (нормирование по освещенности, эквализация гистограмм, медианная фильтрация и т. п.). Далее выполняется его сегментация на $K_1 \times K_2$ блоков фиксированного размера (по K_1 строк и K_2 столбцов). В модуле извлечения признаков вычисляются направления градиента из хорошо зарекомендовавшего себя на практике метода HOG [23]. Множество значений ориентации градиента разбивается на N частей и для каждого сегмента (k_1, k_2) вычисляют гистограммы $H(k_1, k_2)$. Каждый выделенный объект классифицируется в модулях выбора эталонов (с помощью метода МПНП) и измерения сходства на основе посегментного анализа однородности (4). Предполагается, что база данных содержит эталоны X_r, которые тоже разбиваются на фрагменты и для каждого блока вычисляют гистограммы $H_r(k_1, k_2)$ направлений градиентов r-го эталона. Согласно модели кусочно-однородного объекта, соседями сегмента (k_1, k_2) считаются участки $(\widetilde{k}_1, \widetilde{k}_2)$, такие что $|\widetilde{k}_1 - k_1| \leqslant \Delta, |\widetilde{k}_2 - k_2| \leqslant \Delta$, где обычно $\Delta = 0$ или 1. Решение задачи ищется в виде, аналогичном (3):

$$\nu = \underset{r \in \{1,\ldots,R\}}{\arg\min} \left(\sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \rho(H(k_1, k_2), H_r(k_1 + \Delta_1, k_2 + \Delta_2)) - \ln p_r \right), \tag{12}$$

где мера близости $\rho(H(k_1, k_2), H_r(k_1 + \Delta_1, k_2 + \Delta_2))$ определяется аналогично (4). Далее согласно иерархической трехпороговой схеме для ближайшего эталона вычисляется апостериорная вероятность (8), которая используется для принятия окончательного решения в модуле обработки результатов и адаптации. В этом же модуле для каждого объекта из текущего кадра находится ближайший в смысле (12) объект из предыдущего кадра. Если такой объект не найден или минимальное расстояние превышает фиксированный порог (6), то в список изображений идентичных объектов добавляется новый объект. Если объект из списка не был найден в течение некоторого достаточно продолжительного времени, он удаляется из списка. Если для первоначального уровня детализации изображения можно получить достаточно надежное решение (7), (8), то процесс поиска на нем и останавливается. В противном случае количество блоков K_1, K_2 увеличивается и процесс принятия решений повторяется заново.

Аналогичный подход может применяться и в задаче распознавания изолированных слов [5]. Отличие состоит в том, что каждый сегмент входного слова следует сопоставлять не со всеми сегментами всех эталонов (3), а только с L эталонными фонемами (или сенонами) из фонетической базы данных. Поэтому здесь не требуется наличия речевых сигналов для каждой команды из словаря. Вместо этого оказывается достаточно сохранить в базе данных соответствующие сегментам фонемы, изолированно произнесенные диктором [24]. Так как количество фонем L невелико, применение методов приближенного ближайшего соседа, таких как МПНП, оказывается здесь нецелесообразным. Однако иерархический анализ сигналов с последовательным уменьшением длительности сегментов может представлять существенный интерес.

5 Результаты экспериментальных исследований

5.1 Посегментный анализ однородности

В настоящем разделе результаты экспериментальных исследований представленной методологии (см. рис. 1) с точки зрения сформулированного ЦК (1) в задаче распозна-

Таблица 1 Оценка вероятности ошибки распознавания лиц (в %)

Метод	$x_\eta = 0$	$x_\eta = 3$	$x_\eta = 5$
SIFT	$20{,}4 \pm 2{,}2$	$22{,}1 \pm 2{,}2$	$25{,}8 \pm 2{,}3$
Eigenfaces	$27{,}7 \pm 1{,}3$	$27{,}9 \pm 1{,}5$	$28{,}5 \pm 1{,}6$
LBPH	$15{,}9 \pm 1{,}1$	$16{,}1 \pm 1{,}2$	$18{,}2 \pm 1{,}3$
SVM	$12{,}8 \pm 1{,}3$	$14 \pm 1{,}3$	$17{,}1 \pm 1{,}6$
Евклида	$11{,}7 \pm 1{,}6$	$13{,}6 \pm 1{,}6$	$16{,}8 \pm 1{,}7$
ВНС (2), $\Delta = 0$	$10{,}8 \pm 1{,}4$	$12{,}1 \pm 1{,}5$	$14{,}3 \pm 1{,}6$
ПАО (4), (5), $\Delta = 0$	$9{,}4 \pm 1{,}2$	$10{,}9 \pm 1{,}2$	$\mathbf{12{,}9} \pm 1{,}3$
Евклида, $\Delta = 1$	$9{,}9 \pm 1{,}5$	$12{,}4 \pm 1{,}5$	$18{,}8 \pm 1{,}9$
ВНС (2), $\Delta = 1$	$8{,}5 \pm 1{,}3$	$10{,}9 \pm 1{,}7$	$15{,}6 \pm 1{,}8$
ПАО (4), (5), $\Delta = 1$	$\mathbf{7{,}7} \pm 1{,}2$	$\mathbf{9{,}4} \pm 1{,}2$	$14{,}5 \pm 1{,}4$

вания лиц [25, 26]. Все эксперименты проводились на современном ноутбуке (процессор Intel Core i7-2630QM, тактовая частота 2 ГГц, объем ОЗУ 6 ГБ). Лица на фотографиях выделялись с помощью библиотеки OpenCV. Далее выполнялась медианная фильтрация с размером окна 3×3 пикселя [12, 14]. Число элементов в гистограмме ориентированных градиентов было выбрано $N = 8$. Для непараметрических оценок распределений в (2), (4) использовалось гауссовское ядро Парзена–Розенблатта с параметром сглаживания (среднеквадратичным отклонением) $\sigma = 0{,}577$. Эти значения показали в описываемых экспериментах наилучшую точность для рассматриваемых баз данных. Общее рассогласование между двумя фотографиями рассчитывалось как сумма рассогласований между соответствующими фрагментами (12) с выравниванием сегментов в окрестности $\Delta = 0$ и 1. Кроме мер близости (2) и (4) использовалось традиционное расстояние Евклида и машина опорных векторов (SVM) [1, 2]. Кроме признаков HOG проводилось исследование классического метода SIFT [27], а также известные методы распознавания лиц, реализованные в библиотеке OpenCV: Eigenfaces [25] и LBPH (Local Binary Patterns Histograms) [26]. В настоящем подразделе предположим, что ограничения на максимальное время классификации отсутсвуют ($t_0 = +\infty$) и все методы сопоставляются только по значениям средней вероятности ошибки $\overline{\alpha}_\eta$ (1).

Далее представим некоторые результаты для классификации 2720 фронтальных изображений 994 различных людей из базы FERET для случая равенства априорных вероятностей $p_r = 1/R$ (полная априорная неопределенность).

Итоговая близость между изображениями лиц оценивается как взвешенная сумма рассогласований (12) между соответствующими частями. К интенсивности каждой точки всех изображений из контрольной выборки добавлялось случайное число в диапазоне $[-x_\eta; x_\eta]$, где $x_\eta \geqslant 0$ определяет уровень шума. В табл. 1 приведены оценки средних значений и стандартных отклонения для вероятности ошибки $\overline{\alpha}_\eta$ в ЦК (1), которые оценивались методом кросс-валидации с помощью 20-кратного случайного выбора фиксированного числа эталонов $R = 1370$. Здесь, во-первых, предложенный подход с ПАО оказывается достаточно устойчивым к небольшим искажениям тестовых данных. Так, при выравнивании сегментов в $\Delta = 1$-окрестности каждого блока для уровня шума $x_\eta \leqslant 3$ точность уменьшается на 1,5%–1,8%, что несколько лучше по сравнению с другими мерами близости. Однако для большого уровня шума ($x_\eta = 5$) выравнивание ($\Delta = 1$) приводит к росту вероятности

ошибки на 1,3%–2%, поэтому в таком случае следует применять наиболее простые алгоритмы классификации без выравнивания сегментов $\Delta = 0$. Во-вторых, предложенный подход (3), (4), (12) с выравниванием сегментов в Δ-окрестности характеризуется наименьшей вероятностью ошибки при наличии слабых помех ($x_\eta \leqslant 3$) по сравнению с традиционным суммированием расстояний между гистограммами ($\Delta = 0$). В-третьих, вероятность ошибки традиционных методов SIFT и SVM превосходит даже правило ближайшего соседа с метрикой Евклида, что может быть объяснено недостаточным для анализа числом эталонов для каждого класса (человека), что оказывает негативное влияние на качество обучения SVM. Однако и для этого случая предложенные модификации ВНС с посегментным анализом однородности (3), (4) характеризуются наименьшей вероятностью ошибки. Гистограммы локальных бинарных шаблонов (LBPH) оказались намного более точными (на 11%–12%) по сравнению с собственными лицами (eigenfaces). Однако вероятность ошибки LBPH на 1,6%–4,2% выше вероятности ошибки для признаков HOG даже для простейшей метрики Евклида и отсутствия выравнивания ($\Delta = 0$). А использование ПАО позволяет повысить точность еще на 2,3%–3,9%. При этом во всех случаях критерий Мак-Немара на уровне значимости 0,05 показал, что предлагаемый ПАО приводит к значимому повышению точности распознавания изображений по сравнению как с алгоритмами распознавания лиц из библиотеки OpenCV (Eigenfaces, гистограммы LBP), так и с традиционными подходами (машина опорных векторов, метод ближайшего соседа).

5.2 Метод максимально правдоподобного перебора

В настоящем подразделе рассмотрена возможность учета важных для практики ограничений на среднее время принятия решений в рамках ЦК (1). На рис. 2 представлено среднее время классификации t_{avg} для полного перебора, рандомизированного k-d дерева [8], зарекомендовавшего себя для небольших R метода perm-sort [9] и описанного выше МПНП [21].

Анализ табл. 1 и рис. 2 позволяет сделать следующие выводы. Во-первых, с точки зрения ЦК (1) применение традиционной метрики Евклида является приемлемым только для предельно жестких ограничений на среднее время классификации $t_0 \leqslant 3$ мс, так как вероятность ошибки здесь превышает аналогичный показатель для предложенного

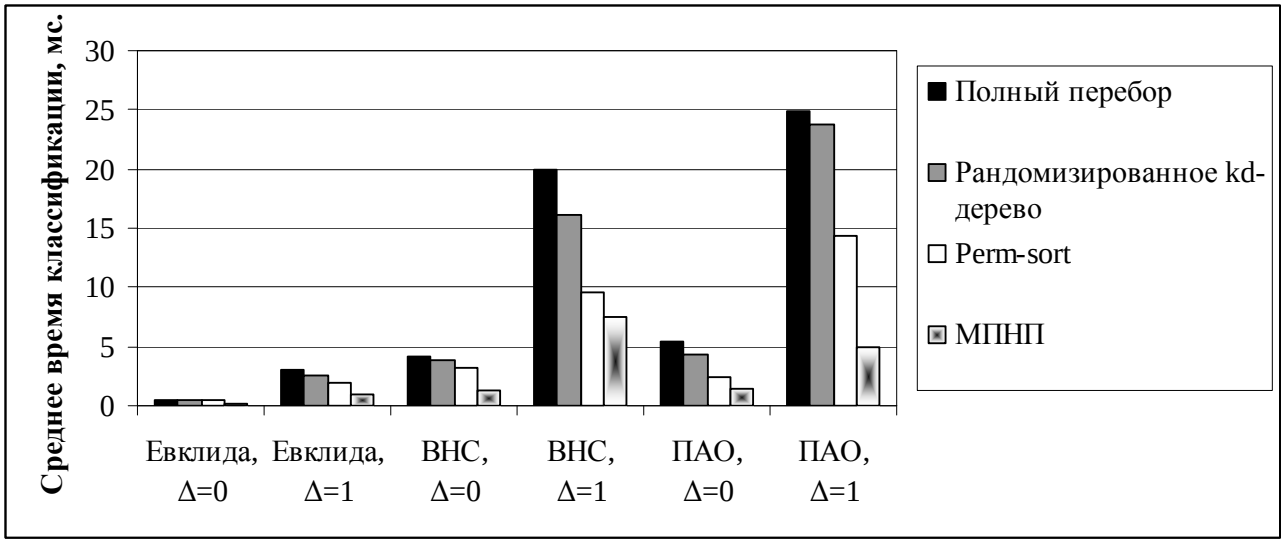


Рис. 2 Среднее время распознавания одного лица

подхода (2), (4) на 2%–4%. При этом для наиболее простого с вычислительной точки зрения случая $\Delta = 0$ применение методов приближенного поиска ближайшего соседа оказывается неоправданным. В то же время с точки зрения ЦК (1) байесовский критерий проверки гипотез о распределении в реализации ВНС (2) во всех случаях оказывается хуже ПАО (3), (4). Во-вторых, реализация предложенного подхода (3), (4) с выравниванием сегментов в Δ-окрестности в рамках полного перебора (12) удовлетворяет ЦК (1) лишь для $t_0 \geqslant 25$ мс. В-третьих, среднее время поиска для рандомизированного k-d дерева практически не отличается от времени распознавания для полного перебора, что можно объяснить небольшим размером обучающей выборки R. В то же время метод perm-sort позволил в 1,3–2,3 раз ускорить классификацию по сравнению с полным перебором. Наконец, основной вывод состоит в том, что предложенный метод МПНП (5), (10), (11) при сопоставимой вероятности ошибки классификации характеризуется наилучшей вычислительной эффективностью: в 2,5–5 раз быстрее по сравнению с полным перебором и в 1,4–3 раза быстрее, чем perm-sort. При этом для наиболее актуального для промышленных систем случая $5 < t_0 < 25$ мс наилучшим с точки зрения ЦК оказывается именно применение метода МПНП для ПАО (3), (4) с выравниванием сегментов ($\Delta = 1$). Здесь вероятность ошибки 8,2% всего на 0,5% выше вероятности ошибки полного перебора (12), а время принятия решений $t_{\text{avg}} = 4{,}9$ мс.

5.3 Иерархическая трехпороговая система

В заключительном эксперименте исследовался иерархический подход (5)–(8). Для повышения быстродействия в пирамиде (иерархии) присутствуют всего два уровня: сетка 5×5 и сетка 10×10. В предложенном подходе второй уровень иерархии (10×10) анализируется только в том случае, если для первого уровня (10×10) оценка апостериорной вероятности (8) не превышает порог $p_0 = 0{,}85$ (7). и 4 приведены оценки средней вероятности ошибки и времени классификации для неиерархического распознавания (сетки «5×5» и «10×10») в сравнении с предложенным подходом. На рис. 3 Последний был

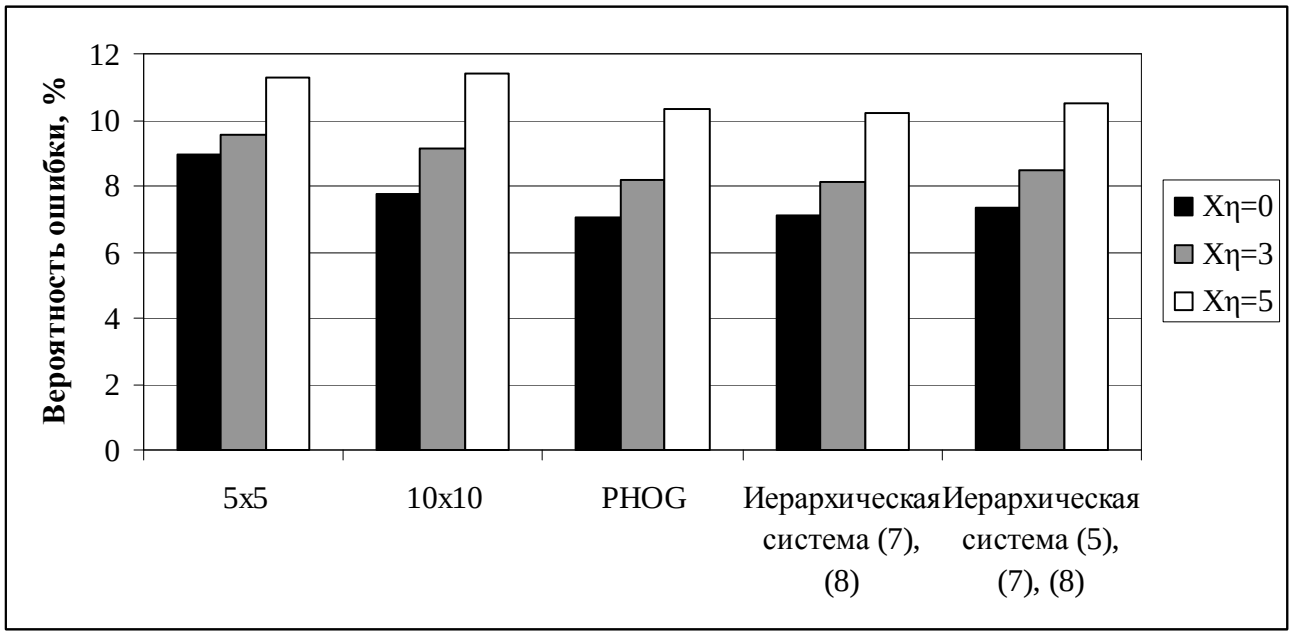


Рис. 3 Оценка вероятности ошибки распознавания для последовательного анализа пирамид HOG

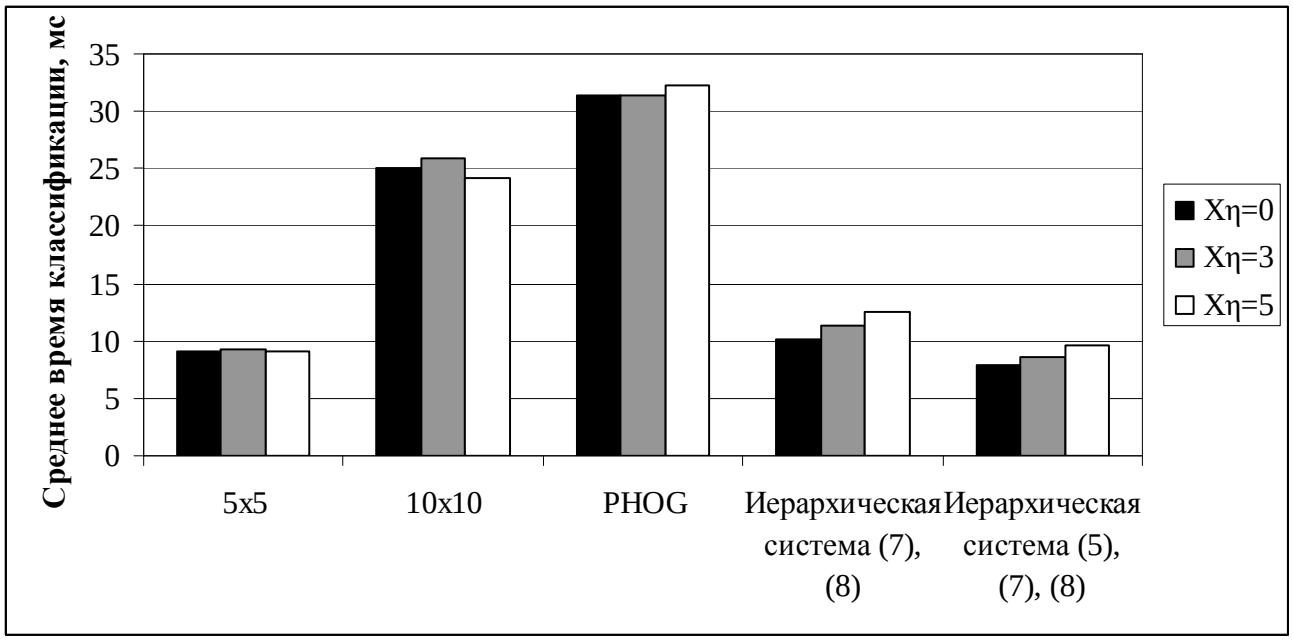


Рис. 4 Среднее время распознавания одного лица для последовательного анализа пирамид HOG

реализован в двух вариантах: с полным анализом каждого уровня и с досрочным остановом перебора (5). Кроме того, использовался традиционный метод PHOG [20], в котором решение принимается по минимуму взвешенной суммы расстояний между гистограммами ориентированных градиентов для каждого уровня иерархии.

Здесь точность классификации для иерархического подхода оказывается выше, чем точность каждого уровня пирамиды, что особенно заметно для малых обучающих выборок. Различия в вероятности ошибки PHOG и предложенного подхода (см. рис. 3) оказываются незначимыми. Однако с точки зрения ЦК (1) разработанная иерархическая система имеет несомненное преимущество с точки зрения вычислительной эффективности. Среднее время распознавания для предложенного подхода оказывается в 2,5–3 раза ниже по сравнению с PHOG, так как в большинстве случае уже на первом уровне (сетки 5×5) удается найти достаточно надежное решение (7). А использование досрочного останова перебора (5) позволяет еще на 30% снизить t_{avg} без существенных потерь в точности.

6 Заключение

Таким образом, в настоящей работе подробно описана комплексная методология ПАО (см. рис. 1), позволяющая повысить точность и вычислительную эффективность систем классификации аудиовизуальной информации в условиях малых выборок и большого числа классов. Ее ключевым звеном является подход к классификации аудиовизуальной информации с анализом однородности сегментов и их динамическим выравниванием, который позволил строго обосновать наблюдаемую во многих промышленных системах недостаточную точность классического байесовского классификатора, реализуемого в ВНС, по сравнению с методами ближайшего соседа, обычно объясняемую «наивностью» предположения о независимости признаков. На основе этого подхода разработан иерархический трехпороговый метод классификации, позволивший в 1,5–3,5 раза ускорить процедуру принятия решений по сравнению с традиционными иерархическими методами (такими, как PHOG). Наконец, рассмотрен приближенный поиск ближайшего соседа в рамках МПНП, основанный на асимптотических свойствах критерия (3), (4). Экспериментально

показано, что метод МПНП ускоряет классификацию лиц в 2–5 раз по сравнению с известными алгоритмами (рандомизированные k-d деревья, perm-sort) для баз данных, содержащих тысячи фотографий. С точки зрения развития данной темы следующим шагом, на наш взгляд, может стать подробный анализ особенностей реализации предложенного подхода (см. рис. 1) к задачам классификации сигналов (прежде всего, распознавания речи), а также реализация ПАО для задач регрессии в виде расширения известного метода GRNN (Generalized Regression Neural Network) [28].

Литература

[1] Pattern recognition / Eds. S. Theodoridis, K. Koutroumbas. — Academic Press, 2008. 984 p.

[2] *Haykin S.* Neural networks and learning machines — 3rd ed. — Prentice Hall, 2008. 936 p.

[3] *Журавлев Ю. И., Рудаков К. В., Гуров С. И., Дюкова Е. В., Кутуков Г. П., Матюнин С. Н., Местецкий Л. М.* Состояние и перспективы развития исследований в области обработки и распознавания видеоинформации. Аналитический обзор // Наука и образование, 2005. Т. 1. `http://technomag.bmstu.ru/doc/48995.html`.

[4] *Hammerstrom D. W., Rehfuss S.* Neurocomputing hardware: Present and future // Artificial Intell. Rev., 1993. Vol. 7. No. 5. P. 285–300.

[5] Springer handbook of speech processing / Eds. J. Benesty, M. Sondh, Y. Huang. — Springer, 2008. 1176 p.

[6] *Tan X., Chen S., Zhou G. H, Zhang F.* Face recognition from a single image per person: A survey // Pattern Recogn., 2006. Vol. 39. No. 9. P. 1725–1745.

[7] *Savchenko A. V.* Directed enumeration method in image recognition // Pattern Recogn., 2012. Vol. 45. No. 8. P. 2952–2961.

[8] *Silpa-Anan C., Hartley R.* Optimised KD-trees for fast image descriptor matching // IEEE Conference (International) on Computer Vision and Pattern Recognition (CVPR) Proceedings, 2008. P. 1–8.

[9] *Gonzalez E. C., Figueroa K., Navarro G.* Effective proximity retrieval by ordering permutations // Pattern Anal. Machine Intell., 2008. Vol. 30. No. 9. P. 1647–1658.

[10] *Боровков А. А.,* Математическая статистика: дополнительные главы. — М.: Наука, 1984. 144 с.

[11] *Абусев Р. А., Лумельский Я. П.* Статистические модели классификации многомерных наблюдений // Обозрение прикладной и промышленной математики, 1996. Т. 3. С. 7–30.

[12] *Shapiro L. G, Stockman G. C.* Computer vision. — Prentice Hall, 2001. 608 p.

[13] *Савченко А. В.* Образ как совокупность выборок независимых одинаково распределенных значений признаков в задачах распознавания сложноструктурированных объектов // Заводская лаборатория. Диагностика материалов, 2014. Т. 80. №3. С. 70–80.

[14] *Savchenko A. V.* Probabilistic neural network with homogeneity testing in recognition of discrete patterns set // Neural Networks, 2013. Vol. 46. P. 227–241.

[15] *Specht D.F.* Probabilistic neural networks // Neural Networks, 1990. Vol. 3. P. 109–118.

[16] *Chow C. K.* On optimum recognition error and reject trade–off // IEEE Trans. Inform. Theory, 1970. Vol. 16. P. 41–46.

[17] *Yao Y. Y.* Granular computing and sequential three-way decisions // Conference (International) on Rough Sets and Knowledge Technology (RSKT) Proceedings, LNCS/LNAI, 2013. Vol. 8171. P. 16–27.

[18] *Dou H., Yang X., Fan J., Xu S.* The models of variable precision multigranulation rough sets // Conference (International) on Rough Sets and Knowledge Technology (RSKT) Proceedings, LNCS/LNAI, 2012. Vol. 7414. P. 465–473.

[19] *Савченко А. В.* Трехпороговая система автоматического распознавания изображений // Искусственный интеллект и принятие решений, 2011. № 4. С. 102–109.

[20] *Bosch A., Zisserman B., Munoz X.* Representing shape with a spatial pyramid kernel // ACM Conference (International) on Image and Video Retrieval (CIVR) Proceedings, 2007. P. 401–408.

[21] *Savchenko A. V.* An optimal greedy approximate nearest neighbor method in statistical pattern recognition // Conference (International) on Pattern Recognition and Machine Intelligence (PReMI), LNCS, 2015. Vol. 9124. P. 1–10.

[22] *Savchenko A. V.* Real-time image recognition with the parallel directed enumeration method // Conference (International) on Vision Systems (ICVS) Proceedings, LNCS, 2013. Vol. 7963. P. 123–132.

[23] *Dalal N., Triggs B.* Histograms of oriented gradients for human detection // IEEE Conference (International) on Computer Vision and Pattern Recognition (CVPR) Proceedings, 2005.

[24] *Savchenko A. V.* Phonetic words decoding software in the problem of Russian speech recognition // Automation Remote Control, 2013. Vol. 74. No. 7. P. 1225–1232.

[25] *Turk M. A., Pentland A. P.* Face recognition using eigenfaces // IEEE Conference (International) on Computer Vision and Pattern Recognition (CVPR) Proceedings, 1991. P. 586–591.

[26] *Ahonen T., Hadid A., Pietikainen M.* Face recognition with local binary patterns // European Conference on Computer Vision Proceedings, 2004. P. 469–481.

[27] *Lowe D.* Distinctive image features from scale-invariant keypoints // Int. J. Computer Vision, 2014. Vol. 60. No. 2. P. 91–110.

[28] *Specht D. F.* A general regression neural network // IEEE Trans. Neural Networks, 1991. Vol. 2. No. 6. P. 568–576.

References

[1] Theodoridis, S., and K. Koutroumbas, eds. 2008. *Pattern recognition.* 4th ed. Academic Press. 984 p.

[2] Haykin, S. 2008 *Neural networks and learning machines.* 3rd ed. Prentice Hall. 936 p.

[3] Zhuravlev, Yu. I., K. V. Rudakov, S. I. Gurov, E. V. Dyukova, G. P. Kutukov, S. N. Matunin, and L. M. Mestetsky. 2005. Sostoyanie i perspectivy razvitiya issledovaniy v oblasti obrabotki i raspoznavaniya videoinformatsii. Analiticheskiy obzor [State and development perspectives of research in video processing and recognition. Analytical survey]. *Nauka i obrazovanie* [Science

and Education] 1. Available at: http://technomag.bmstu.ru/doc/48995.html (accessed June 23, 2015). (In Russian.)

[4] Hammerstrom, D. W., and S. Rehfuss. 1993 Neurocomputing hardware: Present and future. *Artificial Intell. Rev.* 7(5):285–300.

[5] Benesty, J., M. Sondh, and Y. Huang, eds. 2008. *Springer handbook of speech processing.* Springer. 1176 p.

[6] Tan, X., S. Chen, G. H. Zhou, and F. Zhang. 2006. Face recognition from a single image per person: A survey. *Pattern Recogn.* 39(9): 1725–1745.

[7] Savchenko, A. V. 2012. Directed enumeration method in image recognition. *Pattern Recogn.* 45(8):2952–2961.

[8] Silpa-Anan, C., and R. Hartley. 2008. Optimised KD-trees for fast image descriptor matching. *IEEE Conference (International) on Computer Vision and Pattern Recognition (CVPR) Proceedings.* 1–8.

[9] Gonzalez, E. C., K. Figueroa, and G. Navarro. 2008. Effective proximity retrieval by ordering permutations *IEEE Trans. Pattern Anal. Machine Intell.* 30(9):1647–1658.

[10] Borovkov, A. A. 1984. *Matematicheskaya statistika: Dopolnitelnye glavy* [Mathematical statistics: Additional chapters]. Moscow: Nauka. 144 p. (In Russian.)

[11] Abusev, R. A., and Ya. P. Lumelskiy. 1996. Statisticheskie modeli classifikatsii mnogomernykh nablyudeniy [Statistical models of classification of multivariate observations]. *Obozrenie prikladnoy i promyshlennoy matematiki* [Review of Applied and Industrial Mathematics] 3:7–30. (In Russian.)

[12] Shapiro, L. G., and G. C. Stockman. 2001. *Computer vision.* Prentice Hall. 608 p.

[13] Savchenko, A. V. 2014. Obraz kak sovokupnost' vyborok nezavisimykh odinakovo raspredelennykh znacheniy priznakov v zadachakh raspoznavaniya sloghnostrukturirovannykh ob'ektov [Pattern as a set of samples of independent identically distributed features in the tasks of recognition of complex objects]. *Zavodskaya laboratoriya. Diagnostika materialov* [Industrial Laboratory. Materials Diagnostics] 80(3):70–80. (In Russian.)

[14] Savchenko, A. V. 2013. Probabilistic neural network with homogeneity testing in recognition of discrete patterns set. *Neural Networks* 46:227–241.

[15] Specht, D. F. 1990. Probabilistic neural networks. *Neural Networks* 3:109–118.

[16] Chow, C. K. 1970. On optimum recognition error and reject trade–off. *IEEE Trans. Inform. Theory* 16:41–46.

[17] Yao, Y. Y. 2013. Granular computing and sequential three-way decisions. *Conference (International) on Rough Sets and Knowledge Technology (RSKT) Proceedings, LNCS/LNAI.* 8171:16–27.

[18] Dou, H. X. Yang, J. Fan, and S. Xu. 2012. The models of variable precision multigranulation rough sets. *Conference (International) on Rough Sets and Knowledge Technology (RSKT) Proceedings, LNCS/LNAI.* 7414:465–473.

[19] Savchenko, A. V. 2011. Trekhporogovaya sistema avtomaticheskogo raspoznavaniya izobrazheniy [Automatic image recognition three-threshold system]. *Iskusstvenny intellect i prinyatie resheniy* [Artificial Intelligence and Decision Making] 4:102–109. (In Russian.)

[20] Bosch, A., B. Zisserman, and X. Munoz. 2007. Representing shape with a spatial pyramid kernel. *ACM Conference (International) on Image and Video Retrieval (CIVR) Proceedings*. 401–408.

[21] Savchenko, A. V. 2015. An optimal greedy approximate nearest neighbor method in statistical pattern recognition. *Conference (International) on Pattern Recognition and Machine Intelligence (PReMI) Proceedings, LNCS*. 9124:1–10

[22] Savchenko, A. V. 2013. Real-time image recognition with the parallel directed enumeration method. *Conference (International) on Vision Systems (ICVS), LNCS*. 7963:123–132.

[23] Dalal, N., and B. Triggs. 2005. Histograms of oriented gradients for human detection. *IEEE Conference (International) Computer Vision and Pattern Recognition (CVPR) Proceedings*. 886–893.

[24] Savchenko, A. V. 2013 Phonetic words decoding software in the problem of Russian speech recognition. *Automation Remote Control* 74(7):1225–1232.

[25] Turk, M. A., and A. P., Pentland. 1991. Face recognition using eigenfaces. *IEEE Conference (International) on Computer Vision and Pattern Recognition (CVPR) Proceedings*. 586–591.

[26] Ahonen, T., A. Hadid, and M. Pietikainen. 2004. Face recognition with local binary patterns. *European Conference on Computer Vision Proceedings*. 469–481.

[27] Lowe, D. 2014. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision* 60(2):91–110.

[28] Specht, D. F. 1991. A general regression neural network. *IEEE Trans. Neural Networks* 2(6):568–576.

Композиции признаков для видеотрекинга при помощи фильтра частиц*

Е. А. Нижибицкий
nizhibitsky@cs.msu.ru
Москва, Факультет ВМК МГУ имени М. В. Ломоносова

Рассмотрены модели правдоподобия, основанные на композиции мер сходства извлекаемых из изображений признаков, которые широко используются для задачи отслеживания объектов на видео при помощи фильтра частиц. Предложены новые способы оптимального многократного извлечения признаков из различных регионов одного и того же изображения. Оптимизация при этом выполняется за счет построения интегральных изображений, впервые примененных в компьютерном зрении для признаков Хаара в алгоритме Виолы–Джонса, для других исследуемых признаков. Экспериментально показана возможность эффективного использования композиций групп признаков при неэффективности использования каждой группы в отдельности. С помощью рассмотренных композиций достигнуто качество трекинга, сравнимое с более сложными по своей структуре методами, основанными на построении ансамблей с помощью бустинга, и превышающее результаты схожей работы с применением метода каскадов.

Ключевые слова: *трекинг; фильтр частиц; интегральное изображение; гистограмма направленных градиентов; признаки Хаара; локальные бинарные шаблоны; композиция признаков*

Feature composition in video tracking using particle filters*

E. A. Nizhibitsky
Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

This work considers the likelihood models based on similarity measures extracted from image features which are widely used in the field of video tracking using particle filters. New computationally optimal methods for multiple feature extraction from several regions of the same image are proposed. The optimization is performed by using integral images, first prominently used in computer vision within Viola–Jones object detection framework for Haar rectangles and for other studied features. It is experimentally demonstrated that feature compositions can be used even in the tasks where each of them is useless by itself. The performance achieved using the proposed compositions is greater than one in the similar study and comparable to the performance of more complicated models based on ensemble boosting.

Keywords: *tracking; particle filter; integral image; histogram of oriented gradients; HOG; Haar features; local binary patterns; LBP; feature composition*

1 Введение

Задача трекинга объектов на видео является частью таких прикладных областей, как построение систем видеонаблюдения, отслеживания дорожного трафика (в частности, наблюдения за определенными транспортными средствами в потоке) [1], создание интерфейсов человек–компьютер [2], программ для передачи и сжатия видео [3] и др.

*Работа выполнена при финансовой поддержке РФФИ, проект № 14-07-00965.

За последние годы было предложено множество успешных подходов по решению данной задачи [4], но многие из них накладывают свои ограничения на обрабатываемые данные — например, статичный фон и фиксированный ракурс [5], знание о типе наблюдаемого объекта [1] или даже наличие множества камер [6]. Одни подходы уделяют мало внимания вычислительной сложности, другие, наоборот, учитывают строгие ограничения по ресурсам, примером чего являются приложения в робототехнике [7].

Многие из них (см. обзор в [4]) опираются на использование фильтра частиц для приближения вероятностного распределения на положения объекта на видео с помощью частиц, или сэмплов, которым отвечают регионы на видео и те или иные дополнительные характеристики [8]. Для каждой частицы при этом необходимо подсчитывать ее вес, пропорциональный схожести данного региона с регионом для отслеживаемого объекта; следовательно, необходимо уметь выделять признаки из них.

Каждая из таких работ при введении упомянутых мер сходства опирается на свой ограниченный набор признаков, тогда как другие признаки не рассматриваются или по причине самостоятельной неэффективности в условиях каких-то возникающих на исследуемых видео сложностей (например, изменяющееся освещение), или из-за вычислительной сложности многократного выделения признаков для каждой частицы. В разных работах при этом одновременно могут (не)использоваться одни и те же признаки при схожей аргументации (см., к примеру, цветовые признаки и LBP (local binary patterns) в [1, 8]).

Целью данной работы является исследование возможности эффективного использования композиций признаков даже там, где каждый из них может быть неэффективен сам по себе, а также получение способов оптимального многократного извлечения этих признаков из различных регионов одного кадра видеоряда. Это позволит в некоторых задачах задействовать простые композиции «слабых» признаков, не прибегая к вычислительно более затратным. В качестве базовой модели используется модель трекинга из [8] без бустинга, которая дополняется моделями правдоподобия на основе изучаемых признаков.

2 Постановка задачи

Рассматривается задача трекинга объекта на фрагменте видео, где в каждом кадре под положением объекта понимается прямоугольный регион, наилучшим образом его описывающий, — для каждого номера кадра t есть истинное значение $X_t = (x, y, w, h)$, при этом X_0 считается заданным, тем самым осуществляется сопровождение заданного своим начальным положением объекта. Целью отслеживания в данной постановке является поиск приближения $\hat{X}_1, \ldots, \hat{X}_T$ для истинных значений $X_1, \ldots, X_T$ ($\hat{X}_0 = X_0$).

2.1 Используемая мера качества

Для того чтобы определять, насколько выделяемый регион похож на реальный регион, соответствующий отслеживаемому объекту, нужно учитывать не только то, насколько близок центр рассчитанного выделения к реальному центру отслеживаемого объекта, но и то, насколько велика разница между реальными и вычисленными размерами объекта. Для экспериментов в данной работе использовалась мера схожести регионов, предложенная в работе [8], определяющая долю пересечения двух регионов в их объединении (рис. 1).

Аналогично упомянутой работе будем считать, что доле перекрытия выше 33% соответствует правильное определение положения объекта, а итоговое качество наблюдения будем считать как процент кадров, на которых это правильное определение произошло:

$$J(\{\hat{X}_t\}_{t=1}^T, \{X_t\}_{t=1}^T) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}[\text{overlap}\left(\hat{X}_t, X_t\right) \geqslant 0{,}33] \,.$$

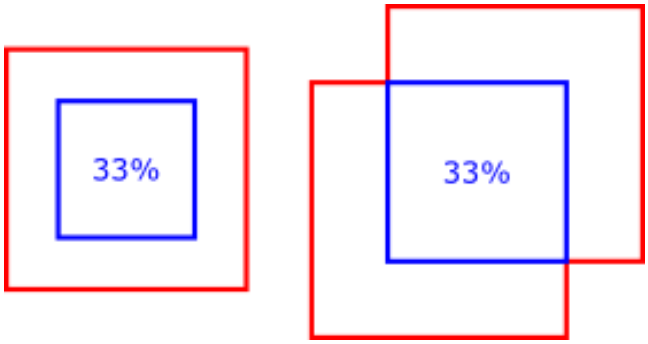


Рис. 1 Пересечение, соответствующее правильному определению положения объекта

3 Фильтр частиц для видеотрекинга

В этом разделе будет рассмотрено применение фильтра частиц для задачи видеотрекинга на основе модели, представленной в [8]. В последующих разделах будет подробнее освещено использование модели правдоподобия алгоритма, которая основывается на комбинировании методов оценивания схожести отдельных регионов изображения с отслеживаемым объектом.

Распределение $p(x_t)$ возможных положений отслеживаемого объекта на i-м кадре видео приближается набором $S_t = \{s_t^i\}_{i=1}^N$ (далее N=1000) взвешенных частиц $s_t^i = (x_t^i, \pi_t^i)$:

$$\hat{p}(x_t) = \sum_{i=1}^{N} \pi_t^i \delta_{x_t^i}(x_t).$$

В работе в качестве состояния частицы рассматривается вектор $(x, y, v_x, v_y, s)^\top$, учитывающий положение и скорость перемещения рамки, содержащей отслеживаемый объект, а также масштаб относительно ее первоначальных размеров при инициализации (считаем, что пропорции объекта не изменяются). При инициализации начальное положение вместе с рамкой (x, y, w, h) передается алгоритму, а веса частиц принимаются равными между собой ($\pi_0^j = 1/N$), начальные скорости (v_x, v_y) получаются из нормального распределения.

Для каждого нового кадра получается новый набор на основе алгоритма фильтра частиц с промежуточным ресэмплингом с учетом важности (веса) каждой частицы (Sampling Importance Resampling Particle Filter):

Алгоритм 1 Фильтр частиц для видеотрекинга (SIR PF)

Вход: S_{t-1}
Выход: S_t
  для $i = 1$ **to** N
    получить $x_t^i \sim p(x_k \,|\, x_{t-1}^i)$ *// модель движения*
    вычислить $\pi_t^i = p(Z_t \,|\, X_t = x_t^i, Z_0, Z_1, \ldots, Z_{t-1})$ *// модель правдоподобия*
  вычислить $w = \Sigma_{i=1}^N \pi_t^i$
  для $i = 1$ **to** N
    вычислить $\pi_t^i = w^{-1}\pi_t^i$ *// нормализация весов*
  для $i = 1$ **to** N
    получить $\hat{x}_t^i \sim \{p(x_k = x_t^j) = \pi_t^j, j = 1, \ldots, N\}$ *// ресэмплинг*
  return $\{(\hat{x}_t^i, N^{-1})\}_{i=1}^N$

Таким образом на каждом шаге на основе имеющихся частиц получаются новые с помощью моделирования их перемещения, затем оценивается правдоподобие нового состояния каждой частицы. В качестве промежуточного этапа перед новым шагом вместо частиц с различными весами получаются частицы с одинаковыми весами, где их состояния будут выборкой нужного размера из дискретного распределения на предыдущих состояниях с вероятностями, пропорциональными их весам.

Модель движения в рассматриваемых экспериментах учитывает имеющиеся скорости частиц для определения новых координат. Сами же скорости вместе с масштабом рамки для каждой частицы получаются на основе зашумления предыдущих значений нормальным распределением:

$$\begin{aligned} v_{x,t} &= v_{x,t-1} + N(0, \sigma_x^2); \\ v_{y,t} &= v_{y,t-1} + N(0, \sigma_y^2); \\ x_t &= x_{t-1} + v_{x,t}; \\ y_t &= y_{t-1} + v_{y,t}; \\ s_t &= s_{t-1} + N(0, \sigma_s^2). \end{aligned}$$

3.1 Модель правдоподобия

Чтобы вычислить вес каждой частицы, нужно оценить правдоподобие наблюдения, отвечающего ей, т. е. оценить схожесть региона, отвечающего частице, с шаблоном — таким же регионом для отслеживаемого объекта. Для этого достаточно извлекать признаки из регионов изображения и сравнивать их между собой. Далее под шаблоном также будут пониматься значения признаков для целевого объекта.

В данной работе рассматривались признаки, зарекомендовавшие себя в задаче отслеживания объектов, моделирующие представления объекта с разной стороны, а значит, подходящие для различных сложностей в исследуемых видео — одни модели используют цветовые признаки, другие же моделируют текстуру, контур или иные характеристики объекта. Для набора рассматриваемых признаков $\{f\}$ определялись меры схожести $\{\rho_f\}$ с шаблонами для реального объекта.

Распределение правдоподобия на основе одной метрики $\rho(\cdot, \cdot)$ можно получить по формуле:

$$p(Z_t|x_t) \propto \exp\left\{-\frac{\rho^2(\hat{h}_f, h_f(x_t))}{\lambda}\right\},$$

где $\hat{h}_f$ — шаблон; $h_f(x_t)$ — признаки для региона изображения, соответствующего состоянию x_t частицы s_t; λ — параметр, подбираемый отдельно для каждой пары признака и соответствующей ему метрики.

Общее правдоподобие наблюдения можно посчитать как произведение правдоподобий по каждому признаку:

$$p(Z_t|x_t) \propto \Pi_f \exp\left\{-\frac{\rho_f^2(\hat{h_f}, h_f(x_t))}{\lambda_f}\right\}.$$

В следующих разделах перейдем к описанию предлагаемых к использованию признаков вместе с оптимизациями для их многократного выделения.

4 Используемые признаки

4.1 Цветовые гистограммы

В качестве первой группы признаков рассматривались простые поканальные гистограммы для каждого из трех каналов RGB-изображения. Для оптимизации в дальнейшем эти значения в каждом канале объединялись в 8 *корзин* из 32 значений интенсивности. Три группы корзин, объединенные между собой и затем нормализованные, образуют вектор-признак из 24 значений. Метрика сходства двух гистограмм при этом определялась на основе Евклидовой метрики:

$$\rho_{\text{hist}}(\hat{h}, h(\mathbf{x_t})) = \sqrt{\sum_{i=1}^{24}(\hat{h}_i - h_i)^2}\,,$$

где $\hat{h}$ — шаблон; $h(x_t)$ — гистограмма для региона x_t, отвечающего частице s_t.

4.2 Признаки Хаара

Признаки Хаара впервые были описаны в [9], где они использовались в алгоритме для распознавания лиц, и на сегодняшний день применяются во многих алгоритмах классификации, так как обладают большей дискриминативной способностью, чем значения пикселей сами по себе. Для получения признаков каждый подрегион разбивается дополнительно на условно светлые и темные области, состоящие из одного или нескольких прямоугольников (рис. 2), для каждой из которых затем вычисляется среднее значение по каналам.

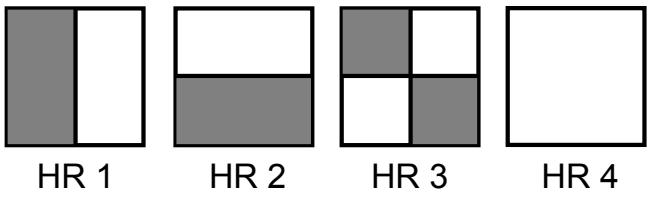


Рис. 2 Четыре признака Хаара для извлечения средних значений по каналам

Чтобы вычислить разницу между признаками для региона, отвечающего частице, и шаблоном, можно также воспользоваться Евклидовой метрикой для векторов из разностей значений в светлых и темных областях:

$$\rho_{\text{hr}}(\hat{c}, c(x_t)) = \sqrt{\sum_{v=1}^{4}((\hat{\text{red}}_v - \text{red}_{(x_t,v)})^2 + (\hat{\text{green}}_v - \text{green}_{(x_t,v)})^2 + (\hat{\text{blue}}_v - \text{blue}_{(x_t,v)})^2)}\,,$$

где $\hat{c}$ — цветовая информация из шаблона; $c(x_t)$ — цветовая информация для частицы с состоянием x_t; v — один из типов признаков, изображенных на рис. 2.

4.3 Гистограммы направленных градиентов

Признаки на основе цветов подходят для многих задач трекинга, даже когда происходят частичные наложения. Тем не менее, они плохо себя показывают в ситуации, когда на фоне присутствуют похожие цвета. Было предложено множество других типов признаков для использования вместе с цветовыми. В [10] показали, что комбинация цветовой модели

вместе с моделью контуров позволяет получить более быстрое и стабильное отслеживание объекта.

Для получения информации о контурах предлагается использовать гистограммы направленных градиентов (Histogram of Oriented Gradients, HOG). По своей природе они устойчиво себя ведут в условиях изменения освещенности и в случае схожести фона и объекта по цветовой модели. Для нахождения границ необходимо перевести RGB-изображение в градации серого, а затем вычислить операторы Собеля K_x и K_y [11]:

$$G_x(x,y) = K_x * I(x,y), G_x(x,y) = K_y * I(x,y) \text{ (под } * \text{ понимается свертка).}$$

Тогда сила (резкость перехода) и ориентация границы вычисляются по формулам:

$$\begin{aligned} S(x,y) &= \sqrt{G_x^2(x,y) + G_y^2(x,y)}; \\ \theta &= \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right). \end{aligned}$$

Чтобы избавиться от шума, можно применить порог T к значению $S(x,y)$ (используем $T = 100$). Затем границы распределяются по K корзинам в соответствии с их направлениями, после чего значения их сил суммируются, и получается нужная нам гистограмма. В оригинальной работе [12] лучше всего показало себя $K = 9$. Схожесть между шаблоном и гистограммой для каждой частицы вычисляется аналогично предыдущим признакам с помощью Евклидовой метрики для векторов из корзин.

4.4 Локальные бинарные шаблоны

Другим примером получения информации о структуре региона изображения является извлечение локальных бинарных шаблонов, которые в некотором смысле характеризуют текстуру изображения в каждой конкретной точке, для чего и были описаны впервые в [13].

Главная идея данного метода состоит в извлечении локальной структуры путем сравнения интенсивности каждого пикселя с его соседями — для каждого соседа получается число, которое будет равно 1, если интенсивность соседа больше рассматриваемого центрального пикселя, и 0 — в противном случае. Если объединить полученные значения по часовой стрелке, то текстуру в окрестности каждого пикселя будет описывать вектор из 0 и 1 вроде 00010011, который и называется локальным бинарным шаблоном. Полученные вектора можно просуммировать по всем пикселям региона и, пронормировав, рассматривать их как гистограмму, определяющую текстурную характеристику всего региона. Схожесть между шаблоном и гистограммой для каждой частицы вычисляется аналогично предыдущим признакам.

5 Оптимизация многократного выделения признаков

Для начала стоит рассмотреть **интегральные изображения**, которые лежат в основе одних из используемых признаков (Хаара), а также будут играть значительную роль в оптимизации подсчета других. Затем будет показано, как можно оптимизировать многократный подсчет всех рассмотренных ранее признаков для различных регионов одного изображения.

5.1 Интегральные изображения

Чтобы получить интегральное изображение I на основе исходного F, для каждого пикселя необходимо вычислить значение по формуле:

$$I(x,y) = F(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1),$$

где $I(x,-1) = I(-1,y) = 0$. А это, очевидно, можно сделать за один проход по результирующему изображению с помощью динамического программирования.

После того как было получено интегральное изображение, для подсчета суммы интенсивностей для прямоугольника с верхним левым углом (x_1, y_1) и нижним правым углом (x_2, y_2) нужно воспользоваться формулой:

$$\sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} F(x,y) = I(x_2,y_2) - I(x_2,y_1-1) - I(x_1-1,y_2) + I(x_1-1,y_1-1)\,.$$

Данное выражение эквивалентно подсчету суммы для региона D изображения F с помощью вычисления $(A+B+C+D)-(A+B)-(A+C)+A$ для изображения I.

5.2 Цветовые гистограммы

Для оптимизации подсчета цветовых гистограмм для регионов на основе значения каждого канала всего текущего кадра видео создается 8 бинарных изображений, в каждом из которых значения будут характеризовать попадание интенсивности в нужную корзину гистограммы. Таким образом, каждое значение интенсивности цветового канала «оставит след» ровно на одном из 8 изображений, которые и будут исходными для получения интегральных изображений. Затем уже на основе этих интегральных изображений можно производить оптимальный расчет гистограмм для любых регионов в кадре — каждый элемент вектора гистограммы получается на основе соответствующих интегральных изображений за несколько простейших операций.

5.3 Признаки Хаара

Для признаков Хаара все гораздо проще — нам требуется создать только три интегральных изображения на основе цветовых каналов изображения из текущего кадра видео, с помощью которых для каждого прямоугольника Хаара каждого региона, отвечающего частице, подсчет нужных характеристик будет производиться за несколько простейших операций. Именно для этого вида признаков интегральные изображения впервые были использованы для оптимизации в [9].

5.4 Гистограммы направленных градиентов

Для эффективного вычисления гистограммы направленных градиентов подобно случаю «обычных» гистограмм для всего изображения можно построить K интегральных изображений на основе K исходных, каждое из которых вместо значений интенсивностей будет содержать либо значение силы границы, если направление совпадает с одним из девяти имеющихся, либо 0 в противном случае — т. о., все значения мощностей границ разойдутся по 9-ти исходным изображениям. Элементы вектора гистограммы региона затем получаются на основе сумм значений интенсивности в регионе для каждого из полученных исходных изображений, которые вычисляются за константное время на основе интегральных изображений.

5.5 Локальные бинарные шаблоны

Так же как и в случае с гистограммами направленных градиентов, значения для локальных бинарных шаблонов можно разделить на 8 отдельных корзин, для каждой из которых построить сначала исходное изображение, а затем интегральное, и с помощью рассмотренных выше приемов для каждого региона получать нужные гистограммы за константное время.

5.6 Теоретические результаты по ускорению

С помощью описанных выше оптимизаций удается избежать затратного многоразового извлечения признаков из регионов для каждой частицы. В табл. 1 приведены оценки вычислительных затрат на выделение N указанных признаков из региона $h \times w$ изображения размера $H \times W$.

Так, к примеру, для регионов, занимающих 10% площади кадра, при оптимизированном подсчете признаков Хаара получается 100-кратная экономия по вычислительным ресурсам при использовании 1000 частиц.

Таблица 1 Теоретические оценки сложности

Группа признаков	Без оптимизации	С оптимизацией
Интегральное изображение	O(Nwh)	O(WH)
Цветовые гистограммы	O(Nwh)	O(WH)
Гистограммы направленных градиентов	O(Nwh)	O(WH)
Локальные бинарные шаблоны	O(Nwh)	O(WH)

6 Вычислительные эксперименты

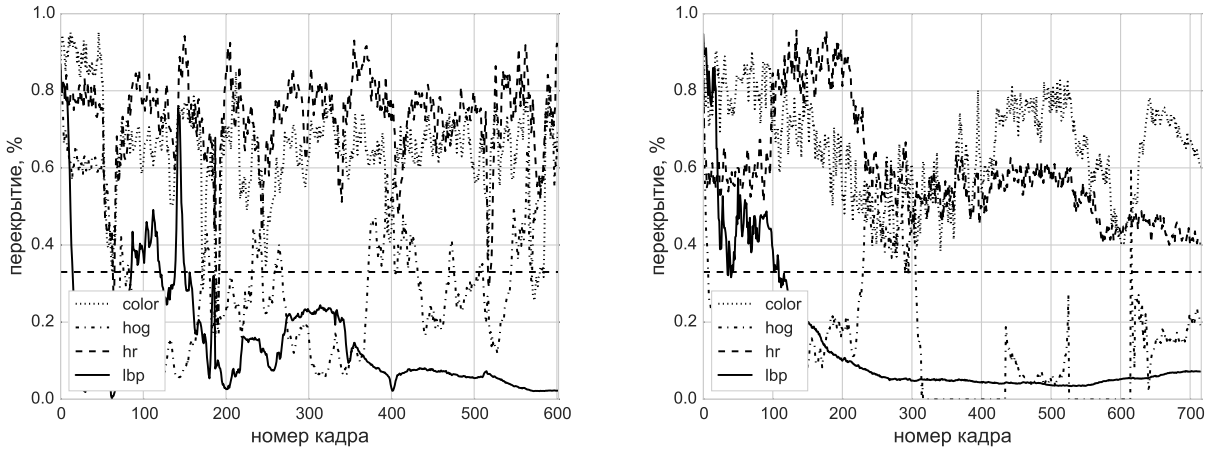
Все рассмотренные алгоритмы с учетом различных признаков были реализованы на языке `C++`. Для высокоуровневой работы с изображениями и видео использовалась библиотека `OpenCV` — в частности, с помощью нее происходила загрузка видео и разбор по кадрам на отдельные изображения, для которых в свою очередь применялись встроенные функции для операторов Собеля (`cv::Sobel`), подсчета интегральных изображений (`cv::integral`) и проверки на попадание в нужный интервал (`cv::inRange`).

6.1 Данные для экспериментов

Для вычислительных экспериментов использовался набор данных BoBoT, содержащий около десятка видео размера 320×240, каждое из которых отвечает тем или иным сложностям, возникающим при трекинге объектов, как-то: сложный неоднородный фон, изменяющееся освещение, перекрытие объектов, сильные перемещения объекта и/или камеры. Он доступен для скачивания на сайте одного из авторов работы [8] (`http://www.iai.uni-bonn.de/~kleind/tracking/`), где также предоставлен исходный код программы на `Java` для оценки качества на основе файлов истинной разметки и разметки, получаемой алгоритмами.

6.2 Результаты экспериментов

В первой части экспериментов были рассмотрены алгоритмы, использующие только одну характерную группу признаков. На рис. 3 изображены примеры графиков с распределением качества по кадрам для алгоритмов, которые для подсчета правдоподобия используют гистограммы цветов (`color`), гистограммы направленных градиентов (`hog`), прямоугольники Хаара (`hr`) или локальные бинарные шаблоны (`lbp`). Горизонтальными линиями отображен порог в 33%, по которому определяется успешность отслеживания объекта в данный момент времени. Из приведенных графиков можно сделать выводы, что самодостаточными для трекинга признаками являются цветовые гистограммы и прямоугольники Хаара. Лишь в некоторых видео значимый результат также показывали гистограммы градиентов. Использование локальных бинарных шаблонов видится осмысленным только в композиции с другими признаками.



(а) Видео A (ball) (б) Видео G (rubick's cube)

Рис. 3 Графики качества для алгоритмов на основе одной группы признаков

При использовании двух групп признаков `color` и `hr` значимые отличия в результатах проявились в анализе двух видео. На видео B с кружкой на очень пестром и разнообразном по текстуре фоне оба алгоритма на основе этих признаков теряют на некоторое время цель, но в разное время — один алгоритм находит похожую на кружку синюю сплошную часть доски, другой видит похожие с шаблонными перепады в цветах на графиках и фотографиях на доске. На видео H признаки Хаара быстро находят схожую по перепадам область и там же и остаются, тогда как цветовые гистограммы находят объект только при фиксированном типе освещения.

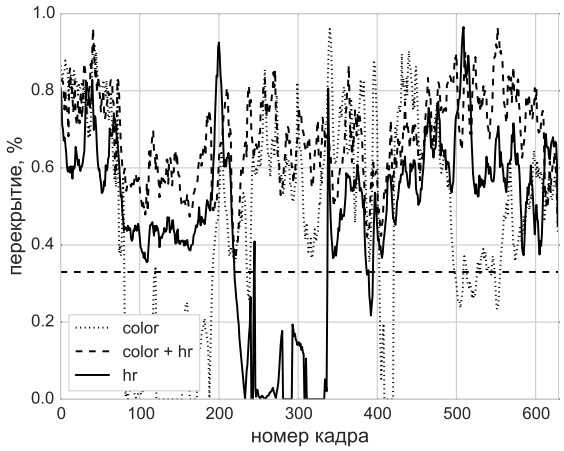
Так как оба типа признаков имеют схожую вычислительную сложность, было решено сравнить их качество по отдельности с композицией, а затем исследовать, как улучшает качество композиции добавление двух оставшихся «несамодостаточных» признаков.

Использование композиции позволяет нам брать лучшее от обеих групп признаков, что подтверждается экспериментами — на видео B при использовании композиции предсказание никогда не уходит от отслеживаемой кружки к схожим частям фона, что позволяет достичь 100%-ного качества. На видео E композиция лучше справляется с перекрытием объекта и не захватывает перекрывающий объект в качестве предполагаемого (рис. 4). Самый сильный же эффект от использования композиции достигается на видео H — наблюдается 100%-ное качество трекинга с постоянным пересечением с реальной областью на уровне 80%, в то время как каждый признак сам по себе совсем не справлялся с трекингом на данном видео (рис. 5).

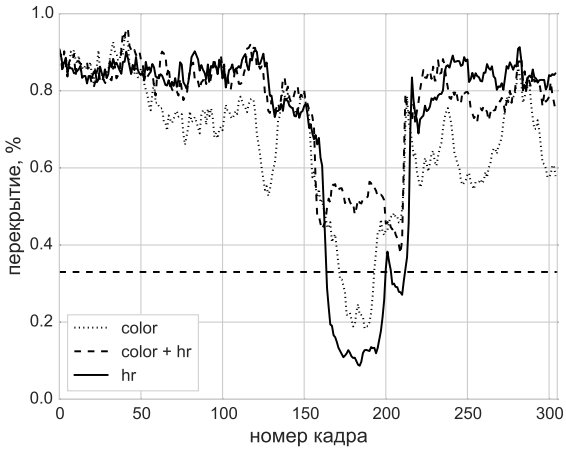
После добавления к изученной композиции «структурных» признаков, учитывающих структуру и текстуру объекта, на видео C стабильно лучше себя показали алгоритмы с добавлением LBP-признаков, что привело к очень высокому качеству трекинга для такой сложной задачи со значительным движением камеры и изменением масштаба объекта. На видео G добавление HOG уменьшило пересечения в конце где-то на четверть, что привело к вылету из минимальной зоны 33%-ного качества (рис. 6).

7 Заключение

В работе изучены способы выделения признаков из изображений для задачи трекинга объектов на видео, а также приведены способы их оптимального многоразового подсчета

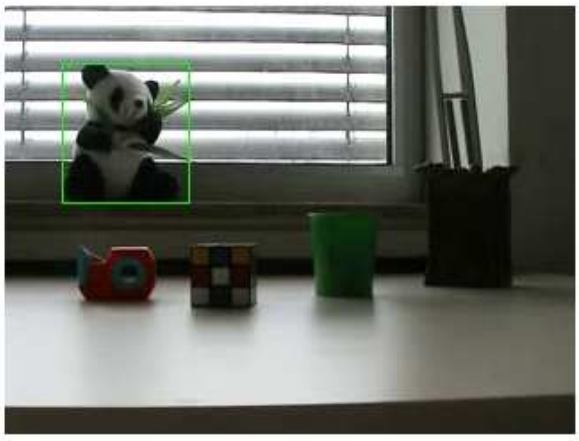


(а) График качества для видео B

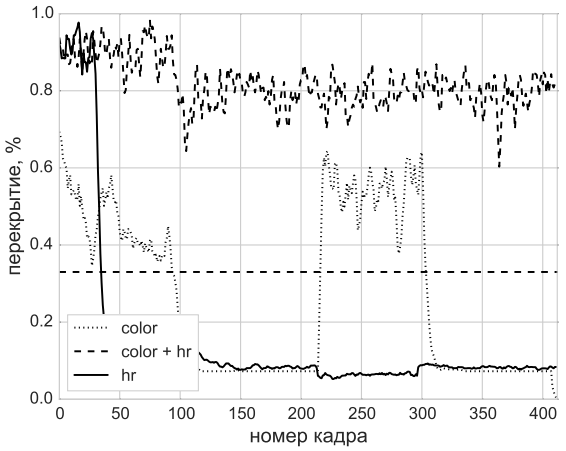


(б) График качества для видео E

Рис. 4 Пример объединения двух признаков в композицию.



(а) Кадр из видео H



(б) График качества для видео H

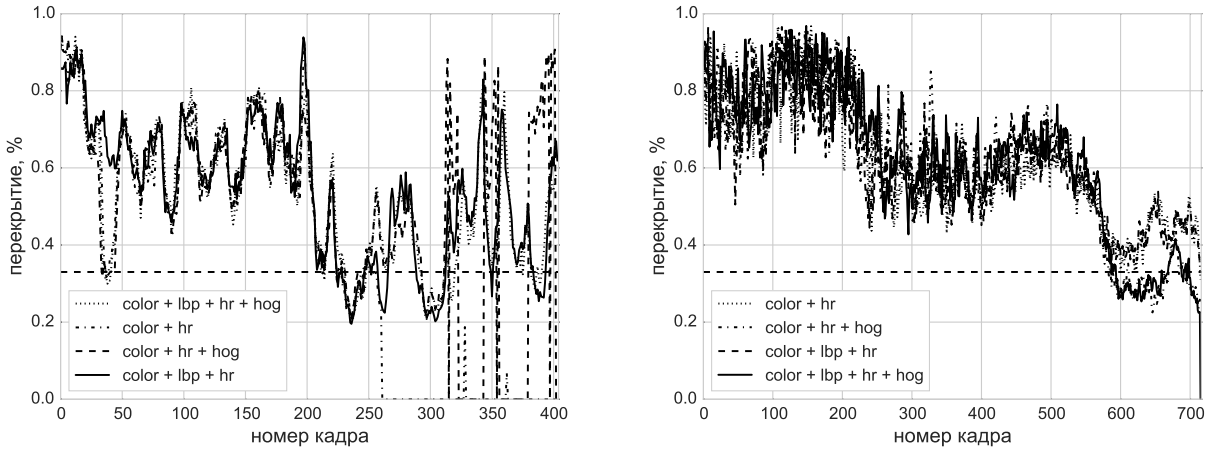
Рис. 5 Наиболее выраженный эффект от объединения двух признаков в композицию.

на основе обобщенного применения интегральных изображений, что позволило построить более богатые алгоритмы с использованием композиций рассмотренных признаков.

Экспериментально показано, что использование композиций признаков позволяет получить идеальные результаты при отслеживании положения сопровождаемого объекта на видео даже там, где алгоритмы на основе каждого из признаков в отдельности с этой задачей справиться не могли (рис. 5).

Из проведенных экспериментов можно также заключить, что наиболее универсальным в рамках рассматриваемых видео оказался композиционный алгоритм, основанный на использовании цветовых интегральных признаков, цветовых гистограмм и локальных бинарных шаблонов. Тем не менее простая комбинация из двух цветовых групп признаков также дает высокие результаты, компенсируя недостатки каждой группы в отдельности.

В табл. 2 приведено сравнение результатов экспериментов для всех рассмотренных алгоритмов на вышеупомянутом наборе данных — полужирным выделены наилучшие



(а) Видео C (juice) (б) Видео G (rubick's cube)

Рис. 6 Графики качества после добавления «структурных» признаков

Таблица 2 Сравнительная таблица качества всех рассмотренных алгоритмов

Группа признаков	A	B	C	D	E	F	G	H	I
`color`	0,966	0,732	0,252	0,947	0,931	0,556	**1,000**	0,441	0,719
`hr`	0,986	0,804	0,475	0,977	0,855	0,439	**1,000**	0,084	0,649
`hog`	0,367	0,184	0,178	0,714	0,200	0,569	0,117	0,415	0,074
`lbp`	0,102	0,081	0,091	0,114	0,347	0,256	0,146	0,038	0,066
`hr + color`	0,993	**1,000**	0,564	**0,991**	**1,000**	0,602	0,997	**1,000**	0,779
`hr + color + lbp`	**0,998**	**1,000**	0,836	**0,991**	**1,000**	**0,613**	**1,000**	**1,000**	**0,782**
`hr + color + hog`	0,996	0,992	0,764	**0,991**	**1,000**	**0,613**	0,853	**1,000**	0,770
`hr + color + lbp + hog`	0,996	0,992	**0,863**	0,990	0,996	**0,613**	0,863	**1,000**	0,772

результаты для каждого видеоряда. Напомним, что под качеством трекинга видео понимается доля кадров с достаточно сильным перекрытием между истинной и предсказанной рамкой, содержащей отслеживаемый объект (качеству 1,0 соответствует исход, при котором объект всегда находится в «поле зрения» модели).

С помощью рассмотренных композиций признаков достигнуто качество трекинга, сравнимое с более продвинутыми методами, основанными на построении сложных ансамблей с помощью бустинга [8], и превышающее результаты схожей работы [1] с использованием метода каскадов.

Литература

[1] *Samuelsson O.* Video tracking algorithm for unmanned aerial vehicle surveillance. Master's Degree Project at KTH Electrical Engineering. Stockholm, 2012.

[2] *Bradski G. R.* Computer vision face tracking for use in a perceptual user interface // Intel Technology J., 1998.

[3] *Vieux W. E., Schwerdt K., Crowley J. L.* Face-tracking and coding for video compression. Lecture notes in computer science ser., 1999.

[4] *Yilmaz A., Javed O., Shah M.* Object tracking: A survey // ACM Comput. Surveys, 2006. Vol. 38. No. 4.

[5] *Li H., Xiong S., Duan P., Kong X.* Multitarget tracking of pedestrians in video sequences based on particle filters // Advanced MultiMedia, 2012. Vol. 2012.

[6] *Xu M., Orwell J., Jones G.* Tracking football players with multiple cameras // IEEE Conference (International) on Image Processing Proceedings. — Los Alamitos, CA, USA: IEEE Computer Society Press, 2004. P. 2909–2912.

[7] *Fox D., Thrun S., Dellaert F., Burgard W.* Particle filters for mobile robot localization // Sequential Monte Carlo methods in practice. — New York, NY, USA: Springer Verlag, 2000.

[8] *Klein D A., Schulz D., Frintrop S., Cremers A. B.* Adaptive real-time video-tracking for arbitrary objects // IEEE Conference (International) on Intelligent Robots and Systems (IROS) Proceedings, 2010. P. 772–777.

[9] *Viola P., Jones M.* Rapid object detection using a boosted cascade of simple features // Conference on Computer Vision and Pattern Recognition Proceedings, 2001. P. 511–518.

[10] *Isard M., Blake A.* Icondensation: Unifying low-level and high-level tracking in a stochastic framework // 5th European Conference on Computer Vision Proceedings, 1998. Vol. I. P. 893–908.

[11] *Sobel I., Feldman G.* A 3×3 isotropic gradient operator for image processing. A talk at the Stanford Artificial Project. — Stanford, CA, USA, 1968.

[12] *Dalal N., Triggs B.* Histograms of oriented gradients for human detection // Conference on Computer Vision and Pattern Recognition Proceedings, 2005. P. 886–893.

[13] *Pietikäinen M., Ojala T., Xu Z.* Performance evaluation of texture measures with classification based on kullback discrimination of distributions // 12th IAPR Conference (International) on Pattern Recognition, 1994. P. 582–585.

References

[1] Samuelsson, O. 2012. Video tracking algorithm for unmanned aerial vehicle surveillance. Master's Degree Project at KTH Electrical Engineering. Stockholm.

[2] Bradski, G. R. 1998. Computer vision face tracking for use in a perceptual user interface. *Intel Technology J.*

[3] Vieux, W. E., K. Schwerdt, and J. L. Crowley. 1999. *Face-tracking and coding for video compression.* Lecture notes in computer science ser.

[4] Yilmaz, A., O. Javed, and M. Shah. 2006. Object tracking: A survey. *ACM Comput. Surveys* 38(4).

[5] Li, H., S. Xiong, P. Duan, and X. Kong. 2012. Multitarget tracking of pedestrians in video sequences based on particle filters. *Advanced MultiMedia* 2012.

[6] Xu, M., J. Orwell, and G. Jones. 2004. Tracking football players with multiple cameras. *IEEE Conference (International) on Image Processing Proceedings.* Los Alamitos, CA: IEEE Computer Society Press. 2909–2912.

[7] Fox, D., S. Thrun, F. Dellaert, and W. Burgard. 2000. Particle filters for mobile robot localization. *Sequential Monte Carlo methods in practice.* New York, NY: Springer Verlag.

[8] Klein, D A., D. Schulz, S. Frintrop, and A. B. Cremers. 2010. Adaptive real-time video-tracking for arbitrary objects. *IEEE Conference (International) on Intelligent Robots and Systems (IROS) Proceedings.* 772–777.

[9] Viola, P., and M. Jones. 2001. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition Conference Proceedings.* 511–518.

[10] Isard, M., and A. Blake. 1998. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. *5th European Conference on Computer Vision Proceedings* I:893–908.

[11] Sobel, I., and G. Feldman. 1968. A 3x3 isotropic gradient operator for image processing. A talk at the Stanford Artificial Project.

[12] Dalal, N., and B. Triggs. 2005. Histograms of oriented gradients for human detection. *Conference on Computer Vision and Pattern Recognition Proceedings.* 886–893.

[13] Pietikäinen, M., T. Ojala, and Z. Xu. 1994. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *12th IAPR Conference (International) on Pattern Recognition Proceedings.* 582–585.

Анализ результатов оконтуривания левого желудочка сердца на эхографических изображениях у здоровых пациентов с помощью автоматического алгоритма*

***В. В. Зюзин*[1], *С. В. Поршнев*[1], *А. О. Бобкова*[1], *А. А. Мухтаров*[1], *В. В. Бобков*[2]**
zvvzuzin@gmail.com
[1]Уральский федеральный университет имени первого Президента России Б. Н. Ельцина, Екатеринбург
[2]Медицинские информационные технологии, Верхняя Пышма

Приведены результаты, подтверждающие работоспособность автоматического оконтуривания левого желудочка (ЛЖ) на эхографическом изображении апикальной четырехкамерной проекции сердца человека. Описан алгоритм автоматического оконтуривания ЛЖ. Продемонстрирована работоспособность алгоритма для пациентов без патологий сердечной мышцы. Исследованы качества оконтуривания. Предложен критерий по определению контуров неправильной формы. Определены направления дальнейших исследований для улучшения качества оконтуривания.

Ключевые слова: *оконтуривание; левый желудочек; детектирование объектов; эхокардиография; УЗИ-изображения*

The analysis of results of the left ventricle contouring using automatic algorithm on ultrasound images*

***V. V. Zyuzin*[1], *S. V. Porshnev*[1], *A. O. Bobkova*[1], *A. A. Mukhtarov*[1], *and V. V. Bobkov*[2]**
[1]Ural Federal University named after First President of Russia B. N. Yeltsin, Ekaterinburg, Russia
[2]Medical Information Technologies, Verhnjaja Pyshma, Russia

The features of automatic contouring of the left ventricle (LV) on echographic sequences are discussed. The automatic algorithm contouring of the LV of the heart on frames, which contain the image of the apical four-chamber projection of the human heart, is proposed. The algorithm is based on the LV area selection using morphological operations; finding the points of attachment of the mitral valve to the heart muscle (the point of the LV base); building of signature LV of the heart in a polar coordinate system centered at the midpoint of the segment connecting points of the LV base; using of the piecewise polynomial approximation of the signature built in a polar coordinate system; calculation of the coordinates of the contour points in the coordinate system of the source frame ultrasound image of a contour by converting the signature from the polar to the Cartesian coordinate system. The quality assessment of automatic contouring is obtained with comparison of expert contours and automatically generated contours. It is shown that using parameters such as precision and recall traditionally used in assessing, the quality of information search, when comparing expert and automatically generated contours, cannot obtain adequate from the physical point of view assessments of the quality of the automated algorithm. The study results in the kinematics of the mass center of the LV region of the heart, which allowed to propose a criterion for the automatic evaluation of

*Работа выполнена при финансовой поддержке ФГБУ «Фонд содействия развитию малых форм предприятий в научно-технической сфере» в рамках госконтракта № 11475р/20975.

the proper construction of an LV contour on separate frames of the video sequence. Identified areas for further research are aimed at improving the quality of contouring.
Keywords: *contouring; left ventricle; detecting objects; echocardiography; ultrasound images*

1 Введение

Полуавтоматический алгоритм оконтуривания ЛЖ на эхографических изображениях, пример которого представлен на рис. 1, описан в [1]. Затем данный алгоритм был дополнен методом автоматического определения точек основания контура ЛЖ [2], а также методом построения сглаженных границ ЛЖ, основанном на аппроксимации сигнатуры контура тремя полиномами 3-го порядка [3]. Анализ результатов полуавтоматического оконтуривания ЛЖ показал, что необходимым условием, обеспечивающим построение адекватного контура ЛЖ на эхографическом изображении, является правильность нахождения точек основания контура — ключевых точек. При выполнении данного условия на ультразвуковых (УЗ) изображениях пациентов, не имеющих серьезных патологий, удается построить границы, а также обесмечить их гладкость. Полученные результаты позволили сделать обоснованный вывод о работоспособности полуавтоматического алгоритма оконтуривания ЛЖ и возможности его модернизации с целью создания полностью автоматического алгоритма оконтуривания.

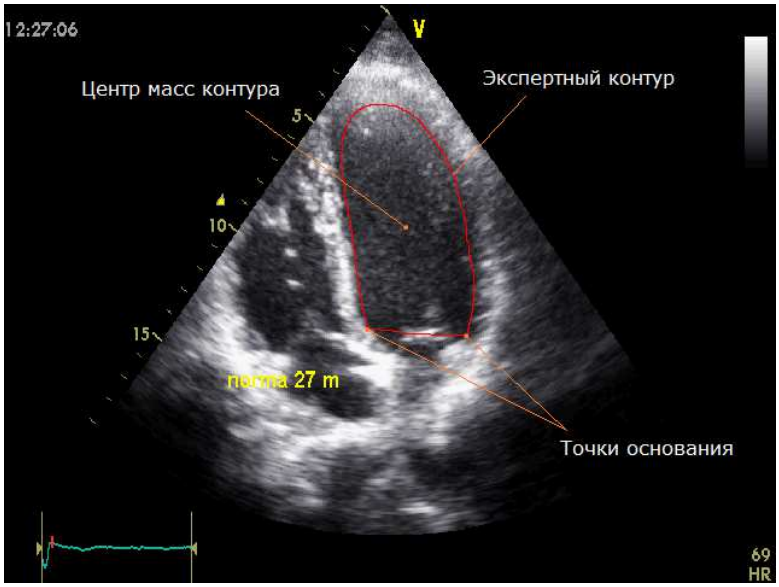


Рис. 1 Исходный УЗ-кадр апикальной четырехкамерной проекции

В статье обсуждаются результаты разработки автоматического алгоритма оконтуривания ЛЖ на УЗ-изображениях.

2 Алгоритмы оконтуривания левого желудочка на ультразвуковых изображениях

Полуавтоматический алгоритм оконтуривания ЛЖ на эхографических изображениях описан в [1]. Данный алгоритм предусматривает предварительную обработку изображения, которая состоит в удалении спекл-шумов и выделении артефактов внутри области ЛЖ методом, описанным в [4], и их удалении.

В данном алгоритме на первом этапе эксперт наносит прямоугольную область, в которой находится область ЛЖ. Соответственно, при обработке видеопоследовательности приходится на каждом кадре указывать данную область. В этой ситуации понятно, что процесс обработки большого количества данных является трудоемким процессом, а потому разработка автоматического алгоритма обработки УЗ-видеопоследовательностей являлется актуальной.

Далее обсуждаемый алгоритм был модифицирован до автоматического определения контура ЛЖ. Модифицированный алгоритм реализуется следующей последовательностью действий:

1. Преобразование исходного кадра в полутоновое изображение путем вычисления взвешенной суммы каждой компоненты пространства RGB:

$$I = 0{,}2989R + 0{,}5870G + 0{,}1140B,$$

   где R, G, B — яркостные компоненты интенсивности красного, зеленого и синего цветов соответственно.
2. Приведение исходного кадра к размеру 480×640 пикселей с использованием кубической интерполяции.
3. Двухпороговая сегментация изображения по параметру интенсивности изображения I, в которой значения порогов сегментации были подобраны эмпирически. Нижний порог оказался равным 18, верхний порог — 90. Пример преобразования изображения при использовании двухпороговой сегментации представлен на рис. 2.
4. Морфологическая обработка изображения, позволяющая разделить соприкасающиеся области и сгладить контуры объектов, в которой в качестве структурирующего элемента используется диск с радиусом 6 пикселей (операция морфологического открытия). Пример изображения, полученного в результате применения операции морфологического открытия, приведен на рис. 3.
5. Преобразование изображения в бинарное представление.
6. Определение центров масс (ЦМ) отдельных областей на изображении (рис. 4).
7. Определение области ЛЖ на бинарном изображении. Условием для выбора искомой области ЛЖ является выбор ЦМ области с наименьшим расстоянием до точки

Рис. 2 Результат сегментации изображения по двум порогам

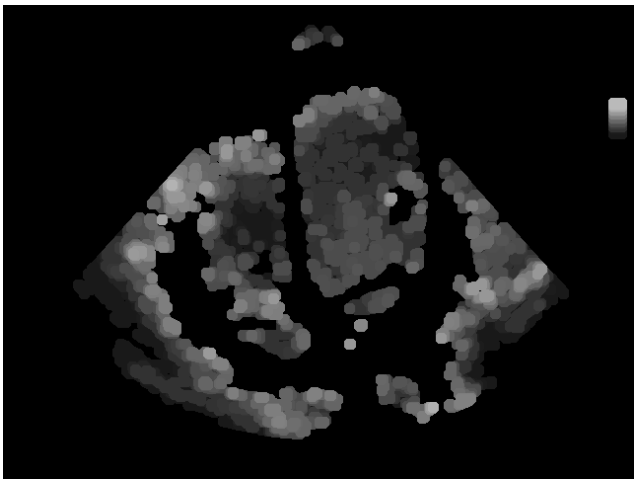


Рис. 3 Результат применения операции морфологического открытия изображения

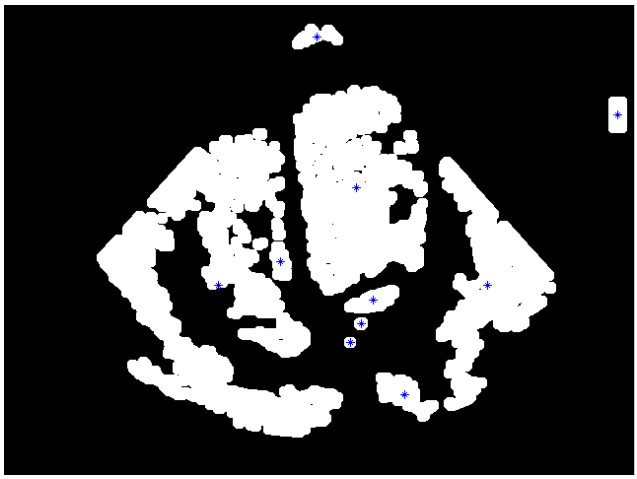


Рис. 4 Результат сегментации изображения по двум порогам

Рис. 5 Результат применения операции морфологического открытия изображения

$(340, 176)$ среднего значения координат ЦМ экспертных контуров, вычисленных по видеопоследовательностям УЗ-изображений сердечной мышцы 18 пациентов, не имеющих патологий. Общее число экспертных контуров равнялось 360.

8. Морфологическая операция закрытия со структурирующим элементом «диск» с радиусом 10 пикселей (рис. 5).

Результатом выполнения вышеизложенного алгоритма является бинарное изображение с областью ЛЖ (см. рис. 5). Из рис. 5 видно, что границы контура, в отличие от экспертных контуров, оказываются негладкими. В этой связи необходимо использование дополнительной процедуры сглаживания границы контура, описанной в [3]. Здесь координаты пикселей бинарного изображения, соответствующие границам полости ЛЖ, преобразуются из декартовых в полярную систему координат, начало которой находится в центре основания ЛЖ. Алгоритм поиска точек основания ЛЖ на эхографическом кадре подробно описан в [2]. Данный алгоритм основан на выделении и классификации ярких областей кадра, соответствующих сердечным тканям, их классификации, целью которой является нахождение областей, соответствующих левой стенке сердца, перегородке, правой стенке, клапану ЛЖ (между перегородкой и правой стенкой), и использовании того медицинского факта, что основанием ЛЖ можно считать отрезок, вершины которого являются точками крепления клапана к стенкам сердца. Зависимость $\rho_i = rho(\Theta_i)$, где i — номер пикселя, полости ЛЖ называют сигнатурой изображения. Пример сигнатур контура, построенного экспертом, контура, построенного по бинарному изображению ЛЖ, и результаты его аппроксимации тремя различными полиномами 3-го порядка представлены на рис. 6. Из рис. 6 видно, что сигнатура контура ЛЖ, построенного автоматическим алгоритмом, оказывается близкой к сигнатуре экспертного контура.

Далее было проведено сравнение результатов автоматического оконтуривания ЛЖ с соответствующими контурами, построенными экспертами, результаты которого изложены в следующем разделе.

3 Исследование качества оконтуривания левого желудочка автоматизированным алгоритмом

В проведенных исследованиях использовались УЗ-изображения 18 пациентов, которые были отнесены специалистами к категории здоровых людей. Для каждого пациента из

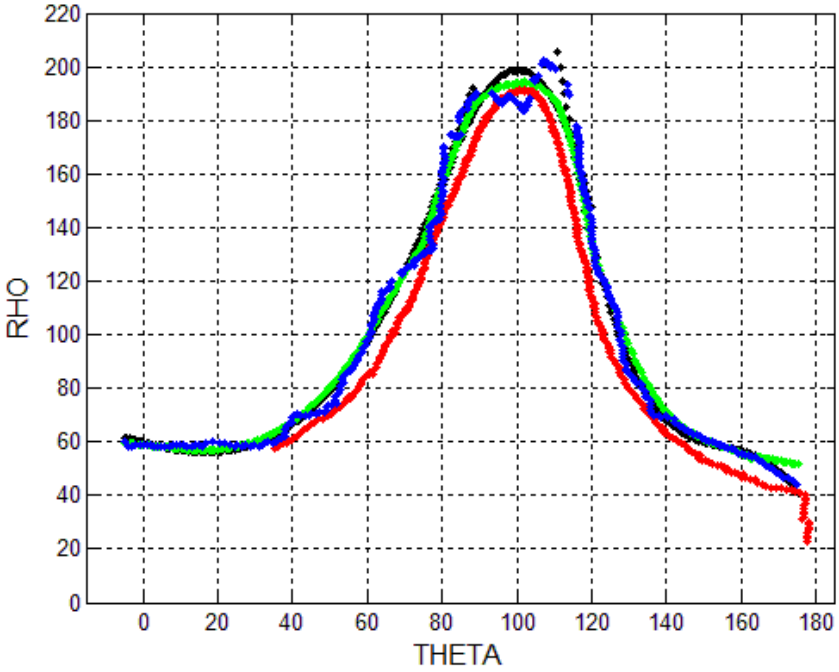
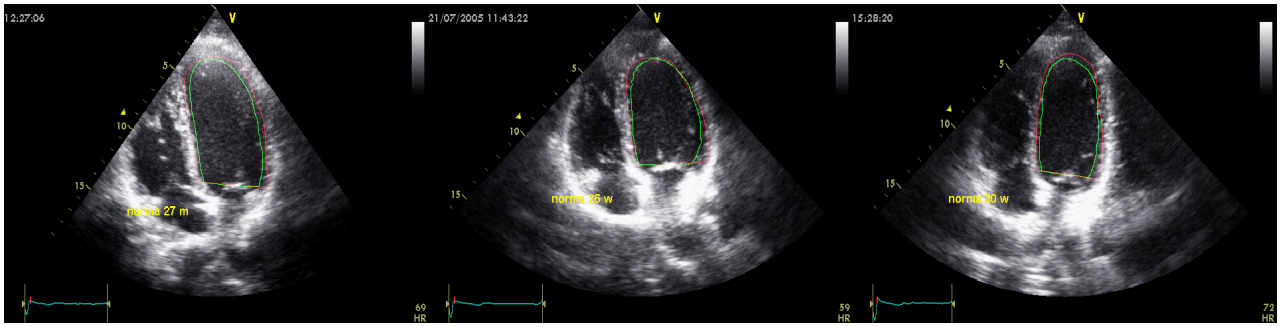


Рис. 6 Результат построения сигнатуры контура: красные точки — сигнатура экспертного контура; синие точки — границы ЛЖ, полученные из бинарного изображения; черные точки — аппроксимирующие кривые границ ЛЖ; зеленые точки — скользящее сглаживание аппроксимирующих кривых



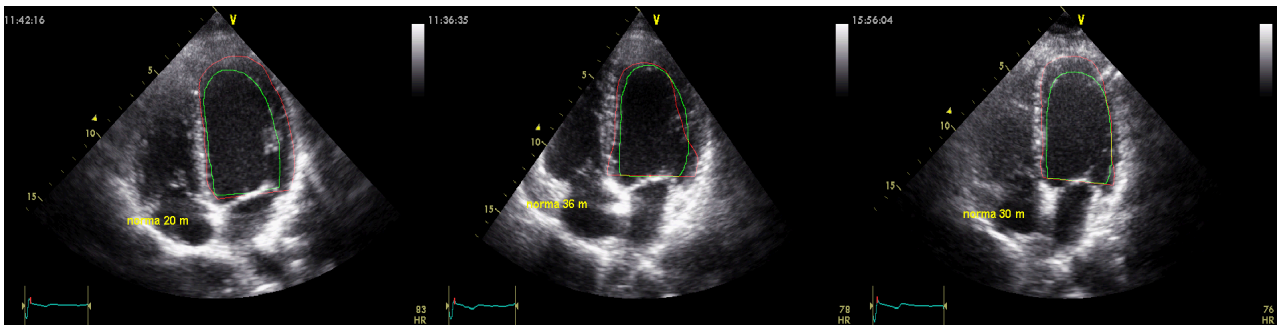
(а) (б) (в)

Рис. 7 Ультразвуковые изображения сердца с удовлетворительно построенными контурами: зеленый — экспертный контур; красный — контур, построенный с помощью алгоритма

всей видеозаписи использовались 20 кадров, которые соответствовали полному сердечному циклу. На каждом кадре в автоматическом режиме строился контур ЛЖ. Общее число кадров составило 360. (Примеры правильных и неправильных контуров, построенных с помощью автоматического алгоритма, представлены на рис. 7 и 8.)

Каждый контур ЛЖ, построенный с помощью автоматического алгоритма, сравнивался с экспертным контуром. Для этого использовались следующие количественные показатели:

– точность (Precision) — отношение области пересечения контуров и области, полученной автоматическим алгоритмом;



(а) (б) (в)

Рис. 8 Ультразвуковые изображения сердца с неудовлетворительно построенными контурами: зеленый — экспертный контур; красный — контур, построенный с помощью алгоритма

– полнота (Recall) — отношение области пересечения контуров и области экспертного контура.

Визуальный анализ контуров ЛЖ, проведенных экспертами, показал, что из 360 контуров только у 195 контуров форма оказалась правильной. В этом случае средние значения количественных показателей оказались следующими:

$$\text{Precision} = 0{,}946 \pm 0{,}037;$$

$$\text{Recall} = 0{,}816 \pm 0{,}031.$$

В случае неправильной формы показатели следующие:

$$\text{Precision} = 0{,}672 \pm 0{,}050;$$

$$\text{Recall} = 0{,}416 \pm 0{,}121.$$

Однако для использования данных характеристик на практике необходимо наличие экспертных контуров. В этой связи были проведены исследования свойств ЦМ экспертных контуров, обсуждаемых далее в статье.

Таким образом, оказывается, что полнота построения для удовлетворительно построенных контуров была выше 0,8, а точность построения — выше 0,9. Примеры УЗ-изображений апикальной четырехкамерной проекции сердца с экспертным и удовлетворительно построенным автоматическим контуром ЛЖ представлены на рис. 7. Из рис. 7 видно, что границы автоматически построенного контура оказываюися достаточно близкими к экспертному контуру.

Примеры УЗ-изображений апикальной четырехкамерной проекции сердца с экспертным и неудовлетворительно построенным автоматическим контуром ЛЖ представлены на рис. 8.

На рис. 8, *а* форма экспертного контура соответствует автоматически построенному контуру, однако правая граница контура построена неверно, что обусловлено наличием неконтрастных тканей. На рис. 8, *б* форма правой границы построена неверно из-за наличия артефактов внутри области. На рис. 8, *в* вершина контура построена неверно.

Таким образом, полнота и точность не дают точной оценки правильности построения контура ЛЖ. В качестве дополнительного критерия правильности построения контура авторами было предложено использовать особенности движения ЦМ ЛЖ сердца.

4 Критерий оценки правильности построения контура

Сравнительный анализ экспертного контура ЛЖ, построенного экспертом, и контура, построенного в автоматическом режиме, позволяет сделать вывод, что координаты ЦМ данных контуров значимо отличаются друг от друга. Таким образом, можно предположить, что значения координаты ЦМ контура ЛЖ сердца являются информативным параметром, по которому можно идентифицировать контуры правильной и неправильной формы (рис. 9).

Для проверки данной гипотезы было проведено исследование особенностей движения ЦМ экспертных контуров. Координаты ЦМ экспертных контуров ЛЖ представлены на рис. 10. Из рис. 10 видно, что ЦМ находятся внутри некоторого эллипса.

Рис. 9 Экспертный контур и контур, построенный автоматическим алгоритмом, ЦМ контуров

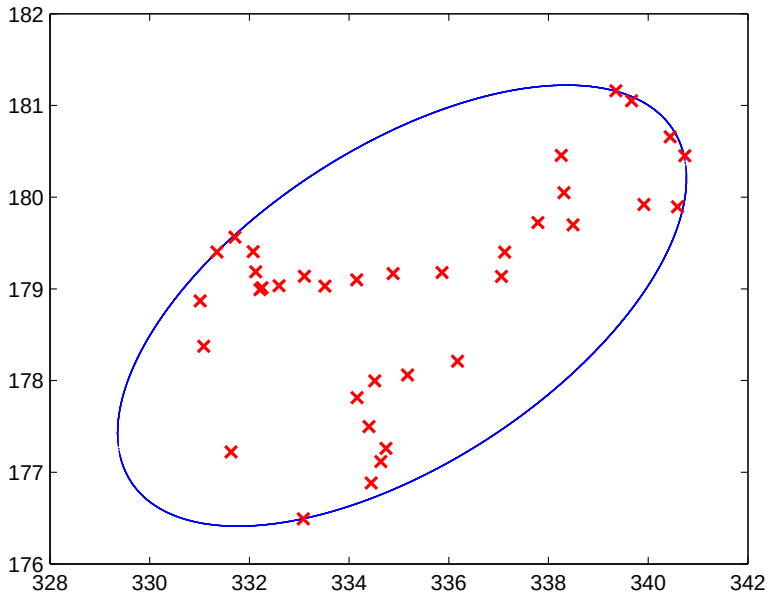


Рис. 10 Центры масс экспертных контуров для одного пациента и эллипс, построенный по точкам ЦМ

Таблица 1 Отношение площади эллипса, охватывающего ЦМ ЛЖ экспертных контуров, к площади экспертного контура ЛЖ в систоле

Пациент	Коэффициент (эксперт)	Коэффициент (автоматический алгоритм)	Пациент	Коэффициент (эксперт)	Коэффициент (автоматический алгоритм)
B	0,0086	0,0424	K	0,0137	0,5185
C	0,0113	0,1180	L	0,0117	0,0237
D	0,0168	0,0110	N	0,0163	0,0349
E	0,0138	0,0117	O	0,0161	0,0189
F	0,0095	0,0136	R	0,0200	0,0235
G	0,0143	0,0674	T	0,0254	0,0205
H	0,0173	0,0533	V	0,0167	0,0981
I	0,0168	0,0987	X	0,0193	0,2188

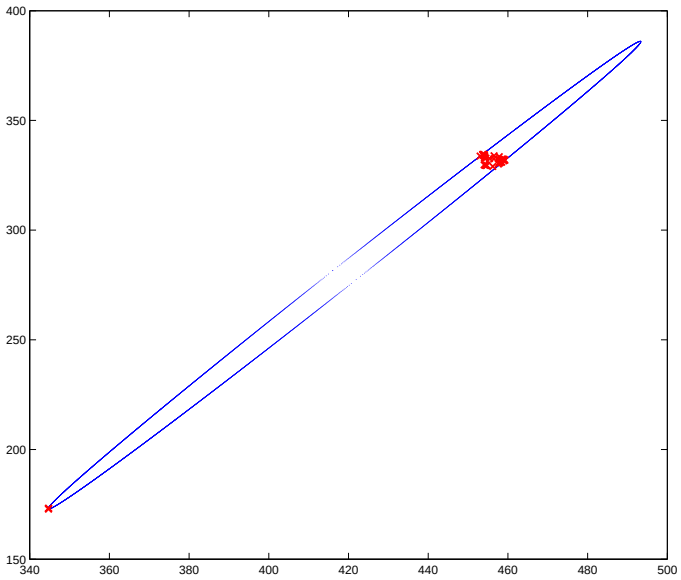


Рис. 11 Расположение ЦМ для пациента с неправильно построенным контуром в видеопоследовательности

Аналогичные результаты были получены для остальных пациентов. Далее для каждого пациента была вычислена площадь эллипсов, которая далее сравнивались с площадью ЛЖ в систоле. Рассчитанные коэффициенты представлены в табл. 1. Результаты аналогичных вычислений для контуров, построенных автоматически, часть из которых имеет неправильную форму, представлены в табл. 1. Из табл. 1 видно, что площадь эллипса, охватывающего ЦМ ЛЖ, не превосходит 3% площади ЛЖ в систоле. Из табл. 1 видно, что максимальное значение отношения площади эллипса, охватывающего ЦМ ЛЖ, к площади экспертного контура в систоле составляет 51%. Таким образом, по положению ЦМ можно идентифицировать правильность построенного контура автоматически.

Примеры расположения ЦМ для случая, когда форма одного из контуров неправильна, представлены на рис. 11. Из рис. 11 видно, что ЦМ для контуров с правильной формой

группируются в некоторой области, тогда как ЦМ контуров с неправильной формой располагаются на удаленном расстоянии от области группировки. Таким образом, задача идентификации контуров неправильной формы сводится к задаче выделения кластеров, состоящих из точек, распределенных по двумерной плоскости. Методы решения задачи кластеризации в данной постановке описаны, например, в [5].

5 Заключение

Описан алгоритм автоматического нахождения контура ЛЖ сердца на эхографических видеопоследовательностях и преиведены результаты, подтверждающие его работоспособность. Обнаружено, что при наличии на кадре низкоконтрастных областей изображения форма контура может оказаться отличной от формы экспертного контура. Неудовлетворительное качество построения контура связано:

- со спецификой проведения правой границы, обусловленной тем, что реальная граница ЛЖ расположена левее контрастной ткани (это связано с наличием неконтрастной ткани — эндокарда, в этой ситуации для проведения левой границы сердца врачи-кардиологи просматривают эндокард в динамике);
- низким контрастом мышечных тканей относительно области ЛЖ;
- наличием более одного артефакта внутри области ЛЖ.

Представляется целесообразным для увеличения качества работы автоматизированного алгоритма дополнить его возможностью учета динамики движения сердечных тканей, что позволит, с точки зрения авторов, более точно строить правую границу контура ЛЖ.

Литература

[1] *Бобкова А. О., Поршнев С. В., Зюзин В. В., Бобков В. В.* Способ полуавтоматического оконтуривания левого желудочка сердца человека на эхографических изображениях // Фундаментальные исследования, 2013. № 8. С. 44–48.

[2] *Бобкова А. О., Поршнев С. В., Зюзин В. В.* Опыт поиска точек основания левого желудочка сердца на эхографических изображениях // Физика и технические приложения волновых процессов. Труды междунар. науч.-технич. конф. Екатеринбург, 2012. С. 361–363.

[3] *Бобкова А. О., Поршнев С. В., Зюзин В. В., Бобков В. В.* The study of features of expert signature for left ventricle on ultrasound images // 9-я Открытая германо-российская конференция «Pattern Recognition and Image Understanding»: Тр. конф. 2014. С. 274–276.

[4] Bobkova A. O., Porshnev S. V., Zyuzin V. V., Bobkov V. V. Analysis of methods for removing noise and artifacts on echocardiographic images // 11-я Междунар. конф. «Распознование образов и анализ изображений – 2013» (РОАИ-11-2013). Самара, 2013. № 2. С. 525–528.

[5] *Загоруйко Н. Г.* Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999. 270 с.

References

[1] Bobkova, A. O., S. V. Porshnev, V. V. Zyuzin, and V. V. Bobkov. 2013. Semiautomatic contouring algorithm of the left ventricle on the echocardiographic images. *Fundamental Research* 8:44–48. (In Russian.)

[2] Bobkova, A. O., S. V. Porshnev, and V. V. Zyuzin. 2012. The way of finding of left ventricle base points on echocardiographic images. *Physics and Technical Applications of Wave Processes: Conference Proceedings*. Yekaterinburg. 361–363.

[3] Bobkova, A. O., S. V. Porshnev, V. V. Zyuzin, and V. V. Bobkov. 2014. The study of features of expert signature for left ventricle on ultrasound images. *9th Open German-Russian Workshop on Pattern Recognition and Image Ubderstanding Proceedings.* Koblenz. 274–276.

[4] Bobkova, A. O., S. V. Porshnev, V. V. Zyuzin, and V. V. Bobkov. 2013. Analysis of methods for removing noise and artifacts on echocardiographic images. *11th Conference (International) on Pattern Recognition and Image Analysis: New Information Technologies Proceedings.* Samara. 2:525–528.

[5] Zagoruiko, N. G. 1999. *Applied methods of data and knowledge analysis.* Novosibirsk: Publishing House of Mathematics Institute. 270 p.

Идентификация имитационных моделей транспортных потоков с помощью разнородных источников прецедентной информации

***Г. Е. Петров*[1], *Ю. В. Чехович*[2]**
greexon@gmail.com, chehovich@forecsys.ru
[1]Московский государственный университет им. М. В. Ломоносова, Москва, Ленинские горы, 1
[2]Вычислительный центр им. А. А. Дородницына РАН, Москва, ул. Вавилова, 42

Рассматривается задача вычисления плотности транспортных потоков с использованием данных из разнородных источников: транспортные детекторы и GPS (Global Positioning System) трекеры. Строится имитационная модель, позволяющая изучить способы и определить границы данных и качества данных, необходимых для восстановления характеристик транспортного потока, и предлагается схема эксперимента возможности комплексирования данных. Ключевую роль играют модели восстановления плотности транспортного потока по его скорости. На основе вычислительных экспериментов получены границы доли транспортных средств и точности определения местоположения для определения параметров транспортного потока.

Ключевые слова: *транспортные потоки; имитационное моделирование; GPS, точность местоположения*

Identification of traffic flow simulation using dissimilar information sources

***G. E. Petrov*[1] *and Y. V. Chehovich*[2]**
[1]M. V. Lomonosov Moscow State University, Moscow, Russia
[2]Dorodnicyn Computing Centre of RAS, Moscow, Russia

In the last few years, automobilization has grown rapidly. One of the most important problems is to collect information about traffic flows. Unfortunately, there do not exist absolutely full and reliable sources of this information. There are some approaches such as using traffic detectors, GPS (Global Positioning System) technology, and photocapture and video recording but they all have disadvantages. For example, traffic detectors determine traffic flow parameters in a limited part of road networks, GPS technology has low spatial accuracy and penetration rate, and photocapture and video recording depend on daylight and weather conditions. In this paper, traffic flow simulation is used to understand the bounds of data quality and data size needed to determine traffic flow parameters such as density and method of conducting the experiment. Single lane encircling highway is considered with installed traffic detectors and running vehicles. Many experiments are carried out with different values of GPS penetration rate and spatial accuracy. Traffic flow density is computed from vehicle speed by using Tanaka's, Greenshield's, and Greenberg's models. According to experimental results, there is a clear bound of 7% of vehicles carrying the traffic monitoring equipment when the quality of determining traffic flow parameters such as density cannot be improved significantly with more penetration rate. In addition, GPS samples spatial accuracy of 100 m leads to 2 percent measurements relative error.

Keywords: *traffic flow; simulation; GPS; spatial accuracy*

1 Введение

1.1 Предисловие

Математическое моделирование транспортных потоков становится все более актуальной проблемой. Это связано с возросшим числом автомобилей и, следовательно, необходимостью оптимизации нагрузки на дорожную сеть. Например, нужно заранее просчитывать последствия изменения транспортной системы при постройке новой развязки или прокладывании автомагистрали, так как недостаточно эффективное решение может ухудшить пропускную способность сети.

Обычно выделяют два вида математических моделей транспортных потоков: макроскопические модели, в которых транспортный поток рассматривается как единое целое и применяются методы гидродинамики, и микроскопические модели, в которых моделируется движение отдельных транспортных средств. Модели первого типа используются для решения глобальных задач транспортной сети, таких как строительство дорог, изменение маршрутов, второго — для решения локальных задач, таких как настройка режима светофора [1].

Значительную сложность при моделировании транспортных потоков представляет сбор информации о транспортной системе. В настоящее время, к сожалению, не существует абсолютно полных и достоверных источников этой информации. Наиболее активно используются следующие источники данных: транспортные детекторы, основанные на различных физических принципах, сбор координат транспортных средств от спутников GPS или ГЛОНАСС (Глобальная навигационная спутниковая система) и передача их на сервер по беспроводным каналам связи, фото- и видеосъемка [1]. У каждого из этих способов есть свои достоинства и недостатки. Например, детекторы измеряют количество и характеристики проезжающих транспортных средств достаточно точно, но только на строго ограниченных участках дорожной сети, в то время как системы, основанные на сборе координат, обладают меньшей точностью, ими оборудована только малая доля автомобилей. Системы фото- и видеофиксации существенно зависимы от условий освещенности и климатических воздействий; кроме того, они требуют периодической очистки внешней оптики от загрязнений. Решить проблему с качеством и полнотой данных может помочь совместное использование данных из разнородных источников.

При экстраполяции данных из разных источников необходимо прежде понять, как ошибка в одном источнике влияет на возможность восстановления характеристик транспортного потока. Для изучения описанной проблемы в рамках настоящей работы строится стенд, позволяющий изучить способы сбора информации и понять границы объема данных и качества необходимых данных.

1.2 Обзор литературы

Рассмотрим некоторые связанные с этой темой работы, в которых для вычисления плотности транспортного потока используются данные GPS.

В статье [2] на примере одного из европейских городов подробно рассматриваются иерархии улиц и их геометрические и топологические свойства. В исследованиях используются данные с GPS-трекеров, установленных на такси, собранные в течение одной недели с периодом 10 с. Замечено, что чем больше скорость автомобиля, тем меньше данных поступает, поэтому применяется корректировка с использованием разницы средней скорости при переезде с одной улицы на другую. В итоге для анализа результатов все координаты автомобильных транспортных средств (АТС), собранные за 24 ч, наносятся на

карту, и становится понятно, какие из улиц наиболее загружены. В результате эксперимента получаются интересные выводы. Например, через 20% улиц проходит 80% трафика и через 1% улиц проходит примерно 20% трафика. Или у улиц длиной более 100 м есть регулярность, иначе наблюдается разное поведение потока.

При постановке эксперимента в статье [3] рассматривались два города: Сан-Франциско и Шанхай. При построении пути движения АТС его координата относится к тому или иному ребру графа дорог. Но так как в координатах присутствуют ошибки, то это не всегда происходит правильно. Для борьбы с ошибками предлагаются следующие способы. Во-первых, если АТС едет из точки A в точку B, то можно предположить, что оно поедет по кратчайшему пути, и посчитать этот путь в графе из вершины A в вершину B по алгоритму Дейкстры. Во-вторых, можно использовать сторонний сервис MapQuest Directions API [4], который по точкам A и B показывает быстрый путь между ними. После этого проводится анализ: связность графа, корреляции между транспортным потоком и различными величинами (например, количество ресторанов). В итоге получается, что для городов работают разные методы анализа, так как они имеют разницу в дизайне и планировании: структура сетки в Сан-Франциско и более беспорядочная структура в Шанхае.

В статье [5] проводится эксперимент по оценке движения транспортного потока в реальном времени с использованием GPS навигатора, встроенного в мобильный телефон. Для того чтобы информация была более анонимной, в устройства ввели координаты виртуальных линий, по пересечении которых записывались данные о позиции и скорости автомобиля. Движение транспортного потока оценивалось с помощью нелинейных моделей течения. Для проверки точности модели использовался транспортный детектор. Также для проверки точности уже после проведения эксперимента в реальном времени мобильные телефоны записывали положение автомобиля каждые 3 с. Качество и точность получаемых данных при использовании виртуальных линий зависит от количества автомобилей, оборудованных GPS навигаторами, которые их пересекают. Для проверки этой гипотезы виртуальные линии поместили прямо в место расположения транспортного детектора. Оказалось, что 3%–4% оборудованных автомобилей уже достаточно для получения более-менее точных результатов. В результате авторы приходят к выводу, что телефоны с GPS модулем могут быть успешно использованы в качестве датчиков движения и для этого нужно не так много оборудованных ими автомобилей.

В статье [6] говорится о методах кластеризации для оценки транспортного потока с использованием географических особенностей дороги. Основная идея состоит в том, чтобы разделить дорогу на различные типы групп с помощью каких-то признаков. Затем для каждой группы по историческим данным вычисляется паттерн потока. Для применения метода достаточно для конкретной области дороги определить группу по ее признакам и затем применить паттерн к этому участку.

1.3 Общие понятия

Введем обозначения, которые в дальнейшем будут активно использованы:

$\rho(t, x)$ — число АТС на единицу длины в момент времени t в окрестности точки трассы с координатой x (плотность транспортного потока);

$v(t, x)$ — скорость АТС в момент времени t в окрестности точки трассы с координатой x;

$Q(\rho)$ — количество АТС, проходящих в единицу времени через заданное сечение (интенсивность транспортного потока).

Зависимость $Q(\rho)$ также часто называют фундаментальной (основной) диаграммой. Обычно данная зависимость выглядит как на рис. 1.

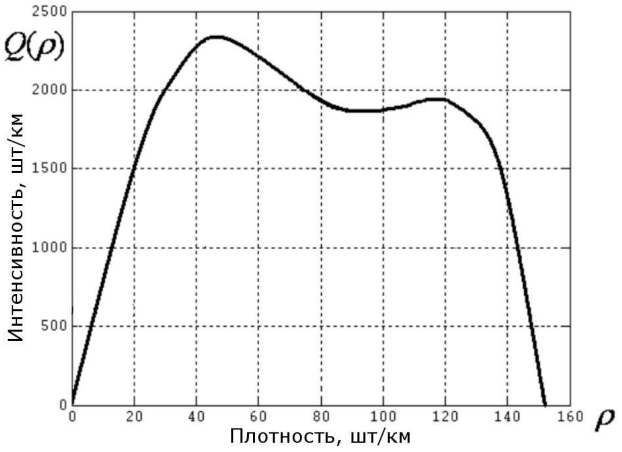


Рис. 1 Фундаментальная диаграмма

Провал интенсивности потока при плотностях $\rho \sim 60$–115 шт./км можно объяснить тем, что в этом случае на интенсивность потока существенно влияют перемещения АТС между полосами при опережении друг друга [7].

2 Постановка задачи

На графе дорог в известных точках установлены детекторы, которые измеряют среднюю скорость транспортного потока. Некоторая часть АТС оборудована GPS-трекерами, неточно определяющими их положение [8]. Под ошибкой GPS-трекера будем понимать следующее: изменение положения автомобиля от правильного равномерно в круге радиуса r (например, в [9] в качестве ошибки рассматривается двумерное нормальное распределение). Необходимо определить границы объема данных и качества данных, влияющих на качество восстановления плотности транспортного потока.

Дополнительно к основной задаче появляются и другие интересные вопросы. Например: каким образом необходимо расположить детекторы для минимизации ошибки определения плотности?

3 Теоретическое описание

3.1 Метод исследования

Рассмотрим два источника данных: транспортные детекторы и GPS-трекеры.

Как уже отмечалось выше, для измерения различных характеристик и определения дорожной ситуации в конкретном ее месте устанавливают транспортные детекторы. Они могут собирать такие данные, как индивидуальная скорость АТС, количество АТС, заполненность (отношение количества времени, когда автомобиль находился под транспортным детектором, к общему времени измерения). Очевидно, недостатком такого подхода является то, что все показатели известны только в определенной точке дороги.

Вторым важным источником данных для определения дорожной ситуации является использование GPS-трекеров, которые широко распространены в последнее время. Недостатками такого подхода являются погрешности измерения приборов (*spatial accuracy*), относительно невысокий процент оборудованных ими АТС (*penetration rate*) и неполнота данных (положение известно лишь в некоторых конкретных точках).

В работе [10] показано, что, используя данные положения сотовых телефонов, можно правильно определить дорогу, по которой движется АТС, для 98,4% всех улиц и для 98,9% всех автомагистралей в Калифорнии, если система определяет положение (*spatial accuracy*) с точностью до 100 м с интервалом обновления в 1 с. А в работе [11] показано, что для достижения хорошего покрытия дороги в случае автомагистрали необходимо как минимум 3% АТС с известными положениями (*penetration rate*) и в случае улицы — как минимум 5%.

Таким образом, можно выделить три причины получения неточных данных:

1) погрешность измерения детектора;
2) погрешность измерения GPS-трекера;
3) погрешность работы самого алгоритма.

Для более точного подбора подходящего алгоритма решения задачи нужно минимизировать ошибки первых двух типов, поэтому реальные данные не подходят для анализа. Этот факт следует из тех соображений, что в случае возникновения большой ошибки вычисления плотности транспортного потока неясно, что является ее источником: ошибочные показания детектора, высокая погрешность GPS-трекера или же ошибки самого алгоритма. В связи с этим возникает необходимость в генерации искусственных данных и в варьировании параметров, которые применялись при генерации, для определения зависимости точности работы алгоритма от входных данных, поэтому для решения задачи построим имитационную модель транспортного потока. Таким образом, в случае использования имитационной модели можно четко определить все интересующие характеристики транспортной системы, так как в любой момент времени можно просто «посмотреть» на нее и провести измерения. В реальной же ситуации такой возможности нет: есть лишь исторические данные, собранные с помощью неточных приборов.

Исследование будем проводить в несколько этапов. Сначала будем сравнивать плотность, вычисленную с помощью данных детектора, с реальной плотностью транспортного потока. После этого сравним плотность, вычисленную по данным GPS-трекера, с плотностью по данным детектора. Также будем исследовать влияние ошибки в измерении положения АТС GPS-трекерами и наличия оборудования на правильность измерения плотности транспортного потока.

3.2 Описание конфигурации

Опишем все необходимые параметры, которые будут в дальнейшем использоваться для эксперимента.

Рассмотрим однополосную кольцевую дорогу с длиной трассы $2\pi R$. Пусть все параметры объектов, участвующих в транспортной системе, пересчитываются с периодом t_{iter} (период дискретизации); t_a — период обновления положения АТС; t_{det} — период обновления собранной детектором статистики по дорожной ситуации. Пусть также на дороге расположены объекты: множество детекторов D; множество АТС A. Изначально все АТС расположены друг за другом на одинаковом расстоянии.

3.3 Измерение качества

Для измерения качества определения плотности транспортного потока выберем интересующий нас участок дороги длиной L. В течение всего времени T симуляции дорожного движения будем измерять плотность на этом участке и строить график ее зависимости от времени. Для определения ошибки измерения плотности потока методом f относительно метода g будем использовать формулу:

$$\frac{\sum_{i=0}^{T/t_{\text{iter}}} |f(it_{\text{iter}}) - g(it_{\text{iter}})| t_{\text{iter}}}{\sum_{i=0}^{T/t_{\text{iter}}} g(it_{\text{iter}}) t_{\text{iter}}},$$

т. е. это отношение невязки плотности.

3.4 Вычисление плотности

В зависимости от исходных данных будем вычислять плотность одним из следующих способов.

1. Измерение истинного значения плотности.
   В каждый момент времени определяем точное значение количества автомобилей n, находящихся на интересующем нас участке дороги, и вычисляем значение плотности по формуле: $\rho = n/L$.
2. Измерение плотности на основе данных детектора.
   С периодом t_{det} считаем суммарную скорость V_{sum} и количество автомобилей n, проехавших под детектором. Тогда из определения плотности следует:

$$\rho = \frac{n}{(V_{\text{sum}}/n) t_{\text{det}}} = \frac{n^2}{V_{\text{sum}} t_{\text{det}}}.$$

3. Измерение плотности на основе данных GPS-трекера.
   Можно выделить два основных способа подсчета плотности транспортного потока на основе данных GPS-трекера: (а) отслеживание количества попавших в интересующую нас область дороги АТС для определения плотности; (б) вычисление средней скорости движения АТС в выделенной нами области и применение различных функционалов для определения плотности.
   Минусами первого подхода, очевидно, является тот факт, что далеко не все АТС оборудованы GPS-трекерами, поэтому вычисленное значение плотности будет пропорционально зависеть от их доли. Другая проблема заключается в том, что нам не известна эта доля оборудованных GPS-трекерами АТС, иначе можно было бы просто разделить значение плотности на процент АТС с GPS-трекерами и получить реальное значение плотности транспортного потока (при условии равномерного распространения оборудования).
   Минусами второго подхода (как и первого) является возможность неравномерного распределения оборудования среди АТС и погрешность измерения. Все остальное зависит уже от метода вычисления плотности.

Здесь параллельно возникает следующая интересная задача: каким образом расположение светофоров влияет на точность определения плотности транспортного потока? Насколько важно отношение времени того, когда горит зеленый свет, ко времени работы красного света? Где лучше всего следует расположить детектор: до светофора, после или прямо в одной точке вместе с ним?

3.5 Пересчет физических параметров участников дорожного движения

Отметим некоторые особенности пересчета физических параметров объектов (светофоры, АТС, детекторы), которые участвуют в эксперименте.

С пересчетом характеристик светофора и сбором статистики детектором все относительно просто, поэтому не будем заострять внимание на этих объектах. Наибольший интерес представляют АТС, для которых необходимо заново обновлять положение, скорость

и ускорение на каждой итерации симуляции со временем дискретизации t_a. Все эти параметры пересчитываются исходя из физических законов:

$$V_i = V_{i-1} + a_{i-1}t_a\,;$$
$$s_i = V_{i-1}t_a + a_{i-1}\frac{t_a^2}{2}\,,$$

где a_{i-1} и V_{i-1} — ускорение и скорость АТС на предыдущем шаге соответственно; V_i и s_i — скорость и пройденное расстояние АТС на текущем шаге соответственно.

Особенность описанной выше конфигурации в том, что дорога является окружностью, поэтому положение АТС пересчитывается следующим образом:

$$\beta_i = \frac{s_i}{R}\,;$$
$$x_i = C_x + R\cos\left(\alpha_{i-1} + \beta_i\right);$$
$$y_i = C_y + R\sin\left(\alpha_{i-1} + \beta_i\right),$$

где C_x и C_y — абсцисса и ордината центра окружности; α_{i-1} — угол положения АТС в предыдущий момент времени; β_i — текущий угол смещения автомобиля за время t_a.

Самый изменчивый параметр АТС — ускорение a, через него уже выражаются все остальные параметры. Будем изменять ускорение по следующим правилам:

(1) пока скорость АТС не достигла некоторого максимума $V_{\max}$, ускорение будет иметь значение a_1, иначе считаем ускорение равным 0;
(2) если впереди АТС на расстоянии L_v возникает какое-то препятствие в виде, например, светофора и/или другого АТС, то ускорение пересчитывается согласно модели «следования за лидером».

Все параметры подбираются опытным путем.

3.6 Модель следования за лидером

Для имитации движения АТС будем использовать модель следования за лидером. Обозначим через s_n координату центра n-го АТС в момент времени $t \geqslant 0$. Принцип простейшей модели следования за лидером в следующем: ускорение n-го АТС прямо пропорционально разности скоростей $(n+1)$-го АТС с коэффициентом пропорциональности, обратно пропорциональным расстоянию до впереди идущего АТС, т. е.:

$$a_n(t+\tau) = \alpha\frac{v_{n+1}(t) - v_n(t)}{s_{n+1}(t) - s_n(t)}\,, \quad \alpha > 0\,,$$

где α — коэффициент чувствительности, характеризующий скорость реакции водителя; τ — время, характеризующее реакцию водителей.

Данная модель была предложена в 1959 г. сотрудниками компании Дженерал Моторс.

3.7 Измерение плотности на основе данных GPS-трекера

Для измерения плотности на основе GPS-трекеров будем использовать следующие модели, подбирая оптимальные параметры по сетке. Все эти модели устанавливают зависимость плотности транспортного потока ρ от его скорости v.

1. Модель Танака [12]:

$$\rho(v) = \frac{1}{d(v)}\,. \tag{1}$$

Здесь $d(v)$ — среднее (безопасное) расстояние между АТС:

$$d(v) = L + c_1 v + c_2 v^2, \tag{2}$$

где L — средняя длина АТС; c_1 — время, характеризующее реакцию водителя; c_2 — коэффициент пропорциональности тормозному пути. При нормальных условиях (сухой асфальт): $L = 4{,}5$; $c_1 = 0{,}504$; $c_2 = 0{,}0285$.

2. Модель Гриндшилдса [13]:

$$\rho = \rho_{\max}\left(1 - \frac{v}{v_{\max}}\right) + c\,, \tag{3}$$

где $\rho_{\max}$ — максимальная плотность потока (при отсутствии движения); $v_{\max}$ — максимальная скорость движения АТС (при пустой дороге: на расстоянии L_v нет препятствий).

3. Модель Гринберга [13]:

$$\rho = \rho_{\max} e^{-v/c}\,, \tag{4}$$

где c — неотрицательная константа с размерностью скорости.

4. Модель Гриндшилдса–Гринберга [13]:

$$\rho = \rho_{\max}\left(1 - \frac{v}{v_{\max}}\right)^{2/(n+1)}, \quad n \leqslant 0\,,$$

где $\rho_{\max}$ — максимальная плотность потока (при отсутствии движения); $v_{\max}$ — максимальная скорость движения АТС (при пустой дороге); n — параметр.

5. Модель Гриндшилдса–Гринберга (другой вид формулы):

$$\rho = \rho_{\max}\left(1 - (n+1)\frac{v}{c}\rho_{\max}^{-(n+1)/2}\right)^{2/(n+1)}, \quad n \neq -1\,,\ n \neq 1\,,$$

где $\rho_{\max}$ — максимальная плотность потока (при отсутствии движения); $v_{\max}$ — максимальная скорость движения АТС (при пустой дороге); c — неотрицательная константа с размерностью скорости; n — параметр.

4 Вычислительные эксперименты

4.1 Значения параметров

Для расчетов будем использовать параметры, указанные в табл. 1. Начальная конфигурация указана на рис. 2.

4.2 Эксперименты

Так как в начале движения транспортный поток еще не пришел в состояние равновесия, то измерение плотности будет производиться не с начала запуска системы, а с какого-то момента (этот порог подбирается визуально по графику вычисления плотности транспортного потока). Далее параметры модели будут выбираться по сетке таким образом, чтобы они доставляли минимум невязке плотности. Чтобы уменьшить случайность в экспериментах, будем проводить их несколько раз (100 раз) и усреднять полученные результаты. Дополнительно введем распределение на начальные условия:

$$v_{\max} \sim U\ [60\ \text{км/ч},\ 80\ \text{км/ч}]\,.$$

Таблица 1 Значения параметров для проведения экспериментов

Параметр	Значение
Ускорение АТС	$a_1 = 0{,}5$ м/с^2
Дальность обзора водителя	$L_v = 100{,}0$ м
Параметр модели следования за лидером	$\alpha = 4{,}5$
Длина трассы L_0 (радиус кольца R)	12 566 м (2000 м)
Максимальная скорость $v_{\max}$	$\sim U$ [60 км/ч, 80 км/ч]
Максимальная плотность $\rho_{\max}$	115 шт./км
Количество итераций пересчета параметров в эксперименте	600 000
Период дискретизации t_{iter}	0,01 с
Период обновления положения АТС t_a	180,0 с
Период обновления статистики детектора t_{det}	180,0 с
Длина участка дороги для измерения плотности	1000,0 м
Количество АТС N	120 шт.

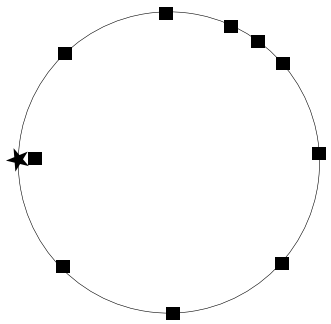


Рис. 2 Начальная конфигурация: направление движения по часовой стрелке, прямоугольники — положения детекторов, звездочка — начальное положение АТС

Ниже приведены результаты ошибки вычисления плотности транспортного потока по данным GPS-трекера относительно плотности, вычисленной по данным детектора, для различных моделей. Также найдена граница процента оборудования GPS-трекерами АТС, при которой происходит существенное уменьшение этой ошибки (примерно 7%).

4.3 Модель Танака (модель № 1)

Модель описывается формулой (1).

Оптимальные значения параметров L, c_1 и c_2 в (2), подобранных по сетке, указаны в табл. 2. Результаты вычисления плотности транспортного потока по средней скорости с использованием модели Танака показаны в табл. 3 и 4 и на рис. 3.

Таблица 2 Оптимальные значения параметров для модели Танака

Параметр	Оптимальное значение	Сетка
L	11,4 м	$1 \cdot 5{,}7, 2 \cdot 5{,}7, \ldots, 10 \cdot 5{,}7$, где 5,7 м — длина АТС
c_1	0,9 м/с	0,1, 0,2, ..., 1,0
c_2	0,3 м/с^2	0,1, 0,2, ..., 1,0

Таблица 3 Относительная ошибка вычисления плотности потока для модели Танака

АТС с GPS, %	Радиус ошибки, м				
	0	5	10	15	20
10	0,053	0,054	0,055	0,055	0,057
20	0,053	0,053	0,054	0,055	0,055
30	0,053	0,053	0,054	0,054	0,055
40	0,053	0,053	0,054	0,054	0,054
50	0,053	0,053	0,054	0,054	0,053
60	0,053	0,053	0,053	0,053	0,054
70	0,053	0,053	0,053	0,053	0,054
80	0,053	0,053	0,053	0,053	0,054
90	0,053	0,053	0,053	0,054	0,054
100	0,053	0,053	0,053	0,054	0,054

Таблица 4 Относительная ошибка вычисления плотности потока для модели Танака

АТС с GPS, %	Радиус ошибки, м									
	1	2	3	4	5	6	7	8	9	10
1	0,772	0,772	0,772	0,772	0,772	0,772	0,773	0,773	0,773	0,773
2	0,692	0,693	0,692	0,693	0,693	0,693	0,693	0,693	0,693	0,693
3	0,532	0,532	0,532	0,533	0,533	0,533	0,532	0,533	0,533	0,532
4	0,333	0,333	0,333	0,333	0,333	0,333	0,334	0,333	0,334	0,333
5	0,133	0,133	0,133	0,134	0,134	0,134	0,134	0,135	0,134	0,136
6	0,133	0,133	0,134	0,134	0,133	0,133	0,134	0,135	0,133	0,134
7	0,053	0,053	0,053	0,054	0,054	0,054	0,055	0,055	0,055	0,056
8	0,053	0,054	0,053	0,053	0,054	0,054	0,054	0,054	0,055	0,055
9	0,053	0,053	0,053	0,054	0,053	0,054	0,054	0,054	0,054	0,055
10	0,053	0,053	0,053	0,054	0,053	0,053	0,054	0,054	0,054	0,054

4.4 Модель Гриндшилдса (модель № 2)

Модель описывается формулой (3).

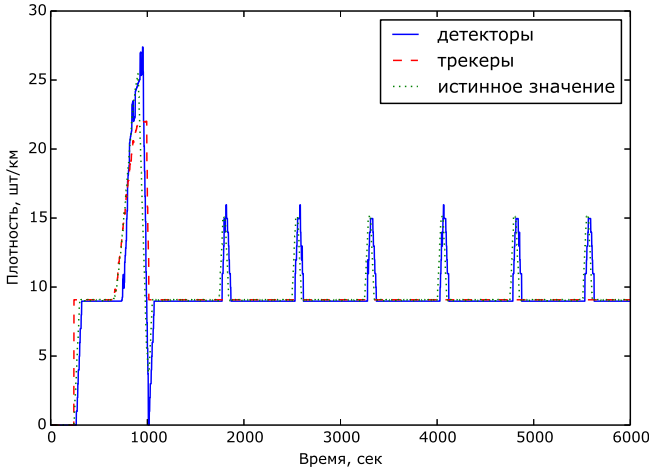
Параметры и результаты для этой модели можно увидеть в табл. 5–7 и на рис. 4.

Таблица 5 Оптимальные значения параметров для модели Гриндшилдса

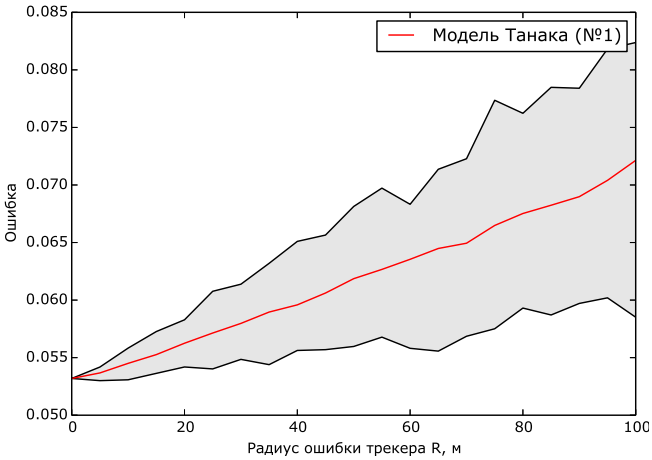
Параметр	Оптимальное значение	Сетка
c	10 шт./км	1, 2, ..., 10

4.5 Модель Гринберга (модель № 3)

Модель описывается формулой (4).



(а) Плотность транспортного потока при доле АТС 100% и радиусе ошибки $R = 0$ м



(б) Относительная ошибка плотности транспортного потока (среднее значение, максимальное и минимальное)

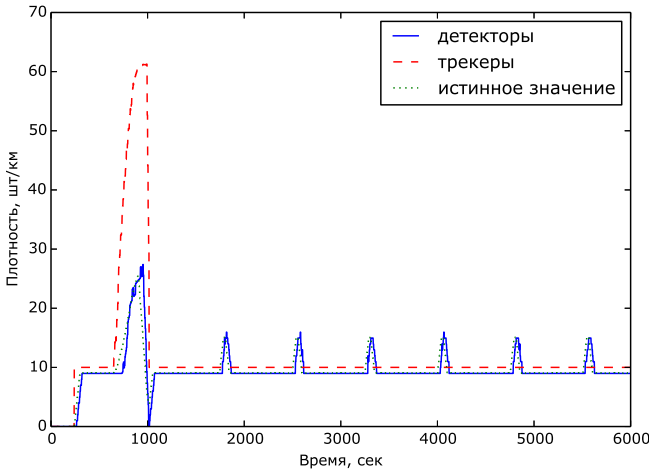
Рис. 3 Модель Танака (модель № 1)

Таблица 6 Относительная ошибка вычисления плотности потока для модели Гриндшилдса

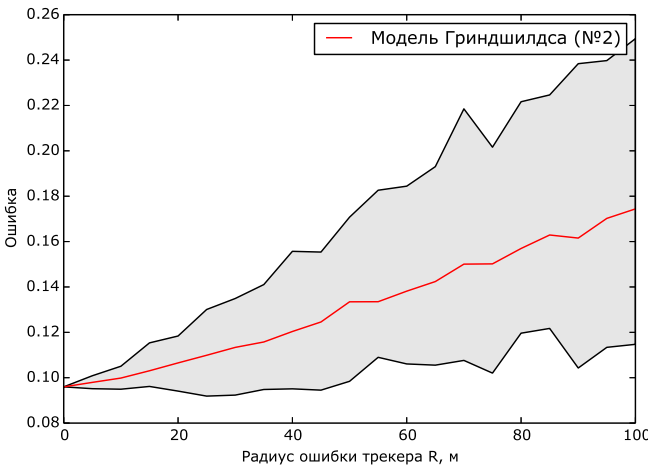
АТС с GPS, %	Радиус ошибки, м				
	0	5	10	15	20
10	0,087	0,088	0,094	0,093	0,098
20	0,087	0,088	0,091	0,091	0,091
30	0,087	0,087	0,089	0,090	0,089
40	0,087	0,088	0,088	0,089	0,094
50	0,087	0,087	0,088	0,089	0,088
60	0,087	0,088	0,089	0,089	0,087
70	0,087	0,088	0,087	0,089	0,088
80	0,087	0,087	0,087	0,088	0,090
90	0,087	0,087	0,087	0,089	0,090
100	0,087	0,087	0,088	0,090	0,090

Таблица 7 Относительная ошибка вычисления плотности потока для модели Гриндшилдса

АТС с GPS, %	Радиус ошибки, м									
	1	2	3	4	5	6	7	8	9	10
1	0,761	0,761	0,761	0,763	0,761	0,763	0,762	0,757	0,759	0,758
2	0,689	0,689	0,689	0,689	0,690	0,689	0,690	0,687	0,688	0,688
3	0,527	0,527	0,527	0,526	0,526	0,527	0,526	0,530	0,526	0,525
4	0,347	0,347	0,348	0,347	0,348	0,348	0,353	0,349	0,351	0,353
5	0,168	0,168	0,170	0,170	0,169	0,172	0,173	0,175	0,171	0,172
6	0,168	0,169	0,170	0,171	0,169	0,169	0,171	0,174	0,171	0,170
7	0,096	0,097	0,097	0,097	0,097	0,098	0,103	0,101	0,101	0,103
8	0.096	0,097	0,096	0,098	0,098	0,098	0,098	0,100	0,100	0,099
9	0,096	0,096	0,097	0,099	0,098	0,100	0,099	0,099	0,099	0,106
10	0,096	0,097	0,097	0,097	0,097	0,097	0,100	0,101	0,099	0,097



(а) Плотность транспортного потока при доле АТС 100% и радиусе ошибки $R = 0$ м



(б) Относительная ошибка плотности транспортного потока (среднее значение, максимальное и минимальное)

Рис. 4 Модель Гриндшилдса (модель № 2)

Параметры и результаты для этой модели можно увидеть в табл. 8–10 и на рис. 5.

Таблица 8 Оптимальные значения параметров для модели Гринберга

Параметр	Оптимальное значение	Сетка
c	7 м/с	1, 2, . . . , 10

Таблица 9 Относительная ошибка вычисления плотности потока для модели Гринберга

АТС с GPS, %	Радиус ошибки, м				
	0	5	10	15	20
10	0,112	0,112	0,111	0,112	0,111
20	0,112	0,112	0,113	0,110	0,111
30	0,112	0,112	0,112	0,112	0,110
40	0,112	0,112	0,112	0,112	0,113
50	0,112	0,112	0,112	0,112	0,112
60	0,112	0,112	0,112	0,112	0,111
70	0,112	0,112	0,112	0,112	0,111
80	0,112	0,112	0,111	0,112	0,112
90	0,112	0,112	0,112	0,112	0,112
100	0,112	0,112	0,112	0,113	0,112

Таблица 10 Относительная ошибка вычисления плотности потока для модели Гринберга

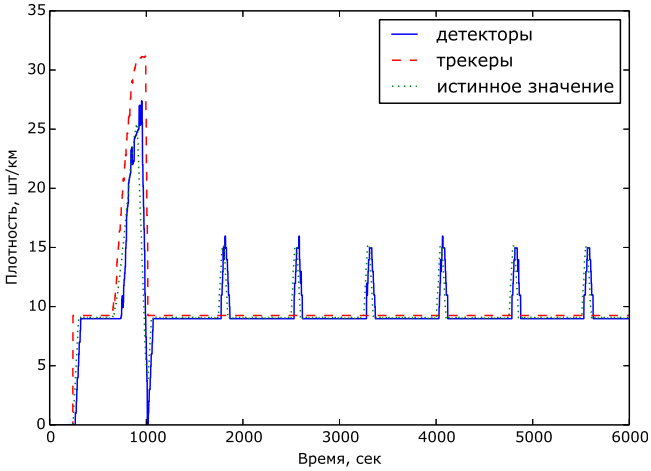
АТС с GPS, %	Радиус ошибки, м									
	1	2	3	4	5	6	7	8	9	10
1	0,758	0,759	0,758	0,759	0,759	0,759	0,758	0,757	0,759	0,757
2	0,692	0,692	0,691	0,692	0,692	0,692	0,692	0,690	0,691	0,691
3	0,530	0,531	0,531	0,530	0,530	0,530	0,530	0,531	0,531	0,530
4	0,364	0,364	0,364	0,363	0,363	0,363	0,364	0,363	0,364	0,366
5	0,197	0,197	0,197	0,198	0,196	0,198	0,198	0,199	0,198	0,194
6	0,197	0,197	0,197	0,197	0,197	0,197	0,197	0,197	0,198	0,197
7	0,130	0,130	0,130	0,129	0,130	0,130	0,132	0,130	0,130	0,130
8	0,130	0,131	0,130	0,131	0,131	0,130	0,130	0,130	0,129	0,129
9	0,130	0,130	0,131	0,131	0,130	0,131	0,130	0,129	0,131	0,132
10	0,130	0,131	0,130	0,130	0,131	0,131	0,131	0,131	0,130	0,129

4.6 Сравнение моделей

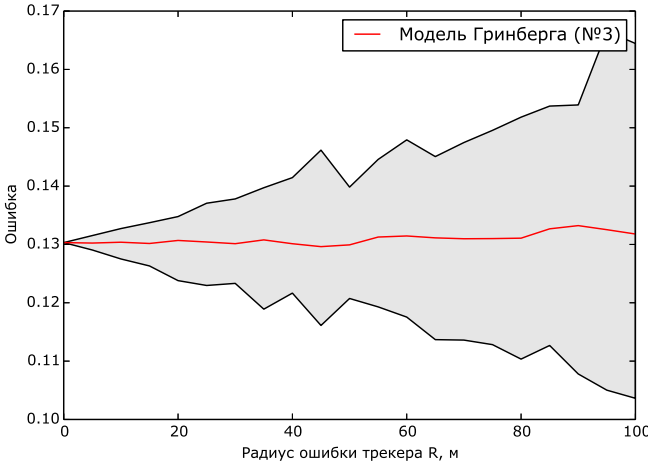
Рассмотрим более подробно, как изменяется ошибка измерения плотности потока при фиксированном значении радиуса ошибки R, а также при фиксированном значении P доли АТС с трекерами (рис. 6).

4.7 Результаты

По приведенным выше результатам экспериментальных расчетов можно сделать следующие выводы. Модель Танака (модель № 1) лучше моделей Гриндшилдса и Гринберга (модели № 2 и № 3) вычисляет плотность транспортного потока: у нее меньше как среднее значение относительной ошибки, так и дисперсия. Также существует ярко выраженная граница в 7% доли оборудованных приборами для определения местоположения АТС, необходимых для расчета плотности транспортного потока. При меньшем количестве АТС относительная ошибка возрастает экспоненциально. При точности определения положения АТС до 100 м относительная ошибка вычисления плотности АТС уменьшается на 2%.

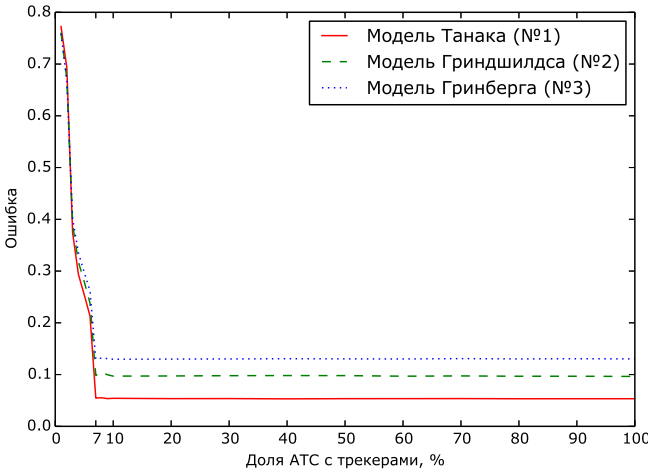


(а) Плотность транспортного потока при доле АТС 100% и радиусе ошибки $R = 0$ м

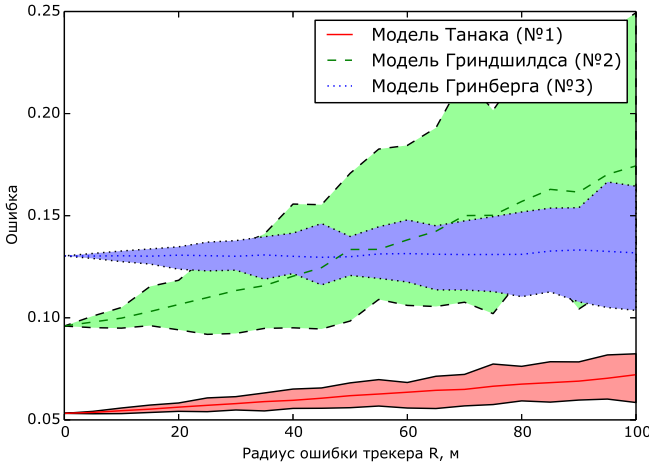


(б) Относительная ошибка плотности транспортного потока (среднее значение, максимальное и минимальное)

Рис. 5 Модель Гринберга (модель № 3)



(а) При фиксированном уровне ошибки детектора $R = 10$



(б) При фиксированной доле АТС с оборудованием $P = 7\%$

Рис. 6 Относительная ошибка вычисления плотности потока

В работе [11] показано, что для достижения хорошего покрытия дороги необходимо 3%–5% (в зависимости от топологии дорожной сети) АТС с оборудованием для измерения положения. Проблема заключается в том, насколько часто и как точно считывать показания приборов. В представленном же в данной работе эксперименте учитывались все эти факторы одновременно и параметры были максимально приближены к реальным значениям. Таким образом, можно утверждать, что существует модель (модель Танака), которая при доле АТС $\geqslant 7\%$, оборудованных приборами для измерения положения с точностью до 40 м, позволяет определить плотность транспортного потока по отношению к детектору с ошибкой $\leqslant 7\%$.

5 Заключение

В данной работе предложена схема эксперимента возможности комплексирования данных транспортных детекторов и GPS-трекеров для определения плотности транспортного потока. Реализована имитационная модель, с помощью которой проведено исследование методов определения плотности транспортного потока и экспериментально рассчитаны оптимальные параметры. Получены предельные экспертные оценки границы доли АТС и границы точности определения местоположения, необходимых для расчета параметров транспортного потока.

Литература

[1] *Воронцов К. В., Чехович Ю. В.* Интеллектуальный анализ данных в задачах моделирования транспортных потоков // Введение в математическое моделирование транспортных потоков / Под общ. ред. А. В. Гасникова. — М.: МЦНМО, 2013. С. 226–249.

[2] *Jiang B.* Street hierarchies: A minority of streets account for a majority of traffic flow // Int. J. Geogr. Inf. Sci., 2009. Vol. 23. Iss. 8. P. 1033–1148.

[3] *Leung I. X. Y., Chan S.-Y., Hui P., Lio P.* Intra-city urban network and traffic flow analysis from GPS mobility trace. arXiv:1105.5839, 2011.

[4] MapQuest Directions API. `http://developer.mapquest.com/web/products/open`.

[5] *Amin S., Andrews S.* Mobile century. Using GPS mobile phones as traffic sensors: A field experiment // 15th World Congress on Intelligent Transportation Systems. New York, NY, USA, 2008.

[6] *Caceresa N., Romerob L. M., Benitezc F. G.* Estimating traffic flow profiles according to a relative attractiveness factor // Energy Efficient Transportation Networks, 2012. Vol. 54. No. 4. P. 1115–1124.

[7] *Гасников А. В.* Введение в математическое моделирование транспортных потоков. — М.: МФТИ, 2010. 361 с.

[8] *Hoh B., Gruteser M., Xiong H., Alrabady A.* Preserving privacy in GPS traces via uncertainty-aware path cloaking // ACM CCS, 2007.

[9] *Jerde C. L., Visscher D. R.* GPS measurement error influences on movement model parameterization // Ecological Appl., 2005. Vol. 15. No. 3. P. 806–810.

[10] *Cayford R., Johnson T.* Operational parameters affecting use of anonymous cell phone tracking for generating traffic information // 82th TRB Annual Meeting, 2003.

[11] *Dai X., Ferman M., Roesser R.* A simulation evaluation of a real-time traffic information system using probe vehicles // IEEE Intelligent Transportation Systems Proceedings, 2003. P. 475–480.

[12] *Gartner N. H., Messer C. J.* Traffic flow theory: A state-of-the-art report. — Washington, DC, USA: Transportation Research Board, 2002. 385 p.

[13] *Иносэ Х., Хамада Т.* Управление дорожным движением. — М.: Транспорт, 1983. 248 с.

References

[1] Vorontsov, K. V., and J. V. Chekhovich. 2013. Intelligent data mining for traffic flows. *Introduction into mathematical traffic flow modeling.* Ed. A. V. Gasnikov. Moscow: MCCME. 226–249.

[2] Jiang, B. 2009. Street hierarchies: A minority of streets account for a majority of traffic flow. *Int. J. Geogr. Inf. Sci.* 3(8):1033–1148.

[3] Leung, I. X. Y., S.-Y. Chan, P. Hui, and P. Lio. 2011. Intra-city urban network and traffic flow analysis from GPS mobility trace. arXiv:1105.5839.

[4] MapQuest Directions API. Available at: `http://developer.mapquest.com/web/products/open` (accessed June 1, 2015).

[5] Amin, S., and S. Andrews. 2008. Mobile century. Using GPS mobile phones as traffic sensors: A field experiment. *15th World Congress on Intelligent Transportation Systems*. New York, NY.

[6] Caceresa, N., L. M. Romerob, and F. G. Benitezc. 2012. Estimating traffic flow profiles according to a relative attractiveness factor. *Energy Efficient Transportation Networks* 54(4):1115–1124.

[7] Gasnikov, A. V. 2010. *Introduction into mathematical traffic flow modeling*. Moscow: MIPT, 2010. 361 p.

[8] Hoh, B., M. Gruteser, H. Xiong, and A. Alrabady. 2007. Preserving privacy in GPS traces via uncertainty-aware path cloaking. *ACM CCS*.

[9] Jerde, C. L., and D. R. Visscher. 2005. GPS measurement error influences on movement model parameterization. *Ecological Appl.* 15(3):806–810.

[10] Cayford, R., and T. Johnson. 2003. Operational parameters affecting use of anonymous cell phone tracking for generating traffic information. *82th TRB Annual Meeting*.

[11] Dai X., M. Ferman, and R. Roesser. 2003. A simulation evaluation of a real-time traffic information system using probe vehicles. *IEEE Intelligent Transportation Systems Proceedings*. 475–480.

[12] Gartner N. H., Messer C. J. 2002. *Traffic flow theory: A state-of-the-art report*. Washington, DC: Transportation Research Board. 385 p.

[13] Inose, H., and T. Hamada. 1983. *Traffic management*. Moscow: Transport. 248 p.

Методы повышения эффективности логических корректоров*

Е. В. Дюкова, Ю. И. Журавлев, П. А. Прокофьев
edjukova@mail.ru, zhuravlev@ccas.ru, p_prok@mail.ru
Вычислительный центр им. А. А. Дородницына РАН, Москва, ул. Вавилова, 42

Рассматривается алгебро-логический подход к корректному распознаванию по прецедентам для задач с целочисленными признаками. Исследуются вопросы повышения распознающей способности и скорости обучения логических корректоров — процедур распознавания, основанных на голосовании по семействам корректных наборов элементарных классификаторов. Вводится понятие корректного набора элементарных классификаторов общего вида, и на этой основе строится модель логического корректора, в которой голосующие семейства наборов элементарных классификаторов формируются итеративно. Рассматривается более широкий, чем в ранее построенных моделях, класс корректирующих функций. Качество работы построенной модели логического корректора тестируется на прикладных задачах.

Ключевые слова: *корректное распознавание по прецедентам; логические процедуры распознавания; алгебро-логический подход; логические корректоры; корректный набор элементарных классификаторов; локальный базис; бустинг*

Methods to improve the effectiveness of logical correctors*

E. V. Djukova, Yu. I. Zhuravlev, and P. A. Prokofjev
Dorodnicyn Computing Centre of RAS, Moscow

Background: One of the key concepts used to build the correct recognition procedures is the concept of elementary classifier. Elementary classifier is an elementary conjunction defined on integer attributive descriptions of objects. Elementary classifier is *correct* if it highlights only the objects of the same class. Classical correct logical recognition procedures are based upon the construction of correct elementary classifier families. There are challenges that cannot find a sufficient number of correct informative elementary classifiers. One way to solve the problem is to build the recognition procedures based on the construction of the families of the correct sets of elementary classifiers (*logical correctors*). The elementary classifiers of the sets of these families are not necessarily correct.
Methods: Some new results concerning the improvement of recognition quality and learning rate of logical correctors are presented. The model of the logical corrector based on a more general concept of the correct set of elementary classifiers is built.
Results: New design allows more succinctly describe the patterns in the classes of objects. New logical correctors have a higher quality of recognition in almost all test problems. Learning rate of the logical correctors increases due to the preselection of high-informative elementary classifiers (local basis).
Concluding Remarks: The proposed methods allow to apply logical correctors for the large-size problems and well-known logical classifiers. Further refinement of the proposed models can be produced by introducing the partial orders on the sets of feature values.

*Работа частично поддержана грантами РФФИ № 13-01-00787-а, № 14-07-00819-а и грантом президента РФ НШ-4908.2014.1.

Keywords: *correct classifier; logic classifier; algebraic-logical approach; logical correctors; correct set of elementary classifiers; local basis; boosting*

1 Введение

Рассматривается задача распознавания по прецедентам с множеством объектов M, представимым в виде объединения непересекающихся подмножеств $K_1, \ldots, K_l$, называемых классами. Задано обучающее множество объектов $T = \{S_1, \ldots, S_m\}$ из M. Каждый объект $S_i \in T$ описан набором значений признаков $x_1, \ldots, x_n$ (числовых характеристик объекта S_i), и известен номер класса $y_i \in \{1, \ldots, l\}$, которому принадлежит S_i. Объекты из T называются *прецедентами* или *обучающими объектами*. Требуется построить алгоритм $A_T : M \to \{0, 1, \ldots, l\}$, ставящий в соответствие произвольному объекту из M, представленному описанием в системе признаков $\{x_1, \ldots, x_n\}$, либо номер класса, которому он принадлежит, либо 0 в случае отказа от распознавания. Алгоритм A_T называется *алгоритмом (процедурой) распознавания*.

Алгоритм распознавания называется *корректным*, если он не ошибается на обучающих объектах. Качество работы алгоритма распознавания на объектах, не являющихся прецедентами, характеризует его *обобщающую способность*. Представляет интерес синтез корректных алгоритмов распознавания с хорошей обобщающей способностью.

Пусть x_j — признак из $\{x_1, \ldots, x_n\}$, S — объект из M и $H = (x_{j_1}, \ldots, x_{j_r})$ — набор признаков. Обозначим через $x_j(S)$ значение признака x_j на объекте S и через $H(S)$ вектор значений признаков $(x_{j_1}(S), \ldots, x_{j_r}(S))$.

В случае, когда множество допустимых значений каждого признака конечно и состоит из целых чисел, задача корректного распознавания успешно решается в рамках *логического* подхода [1–5]. Одним из базовых понятий этого подхода является понятие элементарного классификатора [3].

Пусть $H = (x_{j_1}, \ldots, x_{j_r})$ — набор различных признаков и $\sigma = (\sigma_1, \ldots, \sigma_r)$ — набор, в котором σ_q — допустимое значение признака x_{j_q}, $q \in \{1, \ldots, r\}$. Пара (H, σ) называется *элементарным классификатором* (эл.кл.). Число r называется *рангом* эл.кл. (H, σ). Говорят, что эл.кл. (H, σ) является фрагментом описания объекта S (выделяет объект S), если $H(S) = \sigma$. Элементарный классификатор (H, σ) называется *корректным* для класса $K, K \in \{K_1, \ldots, K_l\}$, если не существует двух выделяемых эл.кл. (H, σ) прецедентов S_i и S_t таких, что $S_i \in K$, $S_t \notin K$, т.е. множество прецедентов, выделяемых корректным эл.кл. (H, σ), является подмножеством либо $T \cap K$, либо $T \setminus K$.

В классических логических процедурах распознавания на этапе обучения для каждого класса K формируется семейство корректных для K эл.кл. При распознавании объекта осуществляется голосование по эл.кл. построенных семейств. Корректность процедуры распознавания обеспечивается за счет корректности каждого эл.кл., участвующего в голосовании. Естественно, что качество работы распознающей процедуры напрямую связано с информативностью использующихся корректных эл.кл. Этап формирования семейств из информативных корректных эл.кл. является наиболее трудоемким в плане вычислительной сложности. Хорошие результаты дает предварительный анализ обучающей выборки, нацеленный на выделение «типичных» обучающих объектов [3].

Довольно часто встречаются задачи, когда почти все корректные эл.кл. имеют большой ранг. Такие задачи являются сложными для рассматриваемых алгоритмов. Несмотря на то что каждый голосующий эл.кл. корректен для некоторого класса K, он плохо характеризует класс K в целом (не является информативным для K). Возникает эффект

переобучения, связанный с тем, что вместо выявления скрытых закономерностей класса фактически происходит «копирование» прецедентов этого класса по отдельности. Зачастую описанная ситуация возникает в связи с большой значностью признаков (под значностью признака понимается число его различных значений, встречающихся в обучающей выборке).

Одним из способов решения указанной проблемы является корректная перекодировка признаков [5]. Другой способ заключается в построении корректной процедуры распознавания на базе произвольных эл.кл., необязательно корректных, что, как правило, осуществляется методами *алгебро-логического* подхода, объединяющего идеи логического и *алгебраического* подходов.

Алгебраический подход применяется, когда требуется скорректировать работу нескольких различных алгоритмов, каждый из которых безошибочно классифицирует лишь часть обучающих объектов. Цель коррекции — сделать так, чтобы ошибки одних алгоритмов были скомпенсированы другими и качество результирующего алгоритма оказалось лучше, чем каждого из базовых алгоритмов в отдельности (см., например, [7, 6]).

Об алгебро-логическом подходе говорят, когда каждый базовый алгоритм распознавания однозначно определяется некоторым эл.кл. и корректирующие функции являются булевыми функциями. Идея алгебро-логического синтеза корректных логических процедур распознавания предложена в [8]. В указанной работе введено понятие корректного набора эл.кл. Подход развит в работах [9–12], в которых рассмотрены вопросы практического применения различных моделей логических корректоров — корректных процедур распознавания, основанных на голосовании по корректным наборам эл.кл.

Определим понятие корректного набора эл.кл. Пусть имеется упорядоченный набор эл.кл. $U = ((H_1, \sigma_1), \ldots, (H_d, \sigma_d))$. Набор U ставит в соответствие объекту S из M бинарный вектор $U(S) = ([H_1(S) = \sigma_1], \ldots, [H_d(S) = \sigma_d])$, который называется *откликом* набора эл.кл. U на объекте S (здесь и далее через $[p]$ обозначается предикат, принимающий значение 1 в случае, когда выражение p истинно, и 0 — в противном случае). Набор эл.кл. U называется *корректным* для класса K, если для любых двух обучающих объектов S_i и S_t таких, что $S_i \in K$ и $S_t \notin K$, отклики $U(S_i)$ и $U(S_t)$ различны. Булевая функция $F(t_1, \ldots, t_d)$ такая, что для любых двух обучающих объектов $S_i \in K$ и $S_t \notin K$ выполняется $F(U(S_i)) \neq F(U(S_t))$, называется *корректирующей* для U.

На этапе обучения логического корректора для каждого класса K формируется семейство корректных для K наборов эл.кл. При распознавании объекта осуществляется голосование по построенным семействам. Корректность процедуры распознавания обеспечивается за счет корректности каждого набора эл.кл., участвующего в голосовании.

Наиболее существенным и трудоемким является этап построения семейства корректных наборов эл.кл., в котором каждый набор обладает высокой распознающей способностью. Для эффективного осуществления этого этапа применяются генетические алгоритмы, а также итерационные и стохастические методы предобработки обучающей информации с целью формирования так называемых *локальных базисов классов*. Под локальным базисом класса понимается специальный корректный набор эл.кл., который в дальнейшем используется для построения искомого семейства корректных наборов эл.кл. Наилучшее качество показывают процедуры голосования по наборам эл.кл. с монотонной корректирующей функцией. Подробный обзор результатов, полученных ранее в рассматриваемой области, приведен в следующем разделе.

В настоящей работе введено более общее понятие корректного набора эл.кл. и на его основе построена новая модель логического корректора. Для удобства описания модели

выбран язык предикатов. На этапе обучения строятся семейства предикатов, каждый из которых порождается некоторым корректным набором эл.кл. и зависит от свойств корректирующей функции. На конструкцию предиката влияет характер монотонности корректирующей функции по ее отдельным переменным, что является важным отличием от ранее построенных логических корректоров. Семейства голосующих предикатов формируются итеративно по принципу бустинга. Приведены результаты тестирования построенного логического корректора на прикладных задачах.

2 Обзор предыдущих результатов

Классическими логическими распознающими процедурами принято считать тестовый алгоритм (голосование по тестам) [1] и голосование по представительным наборам [2].

Тестом называется набор признаков H такой, что для любых двух прецедентов S_i и S_t, принадлежащих разным классам, векторы значений признаков $H(S_i)$ и $H(S_t)$ различны. На этапе обучения тестового алгоритма формируется семейство тестов $\mathcal{H}$. Распознавание объекта S осуществляется путем голосования по построенным тестам. В простейшей модификации тестового алгоритма для каждого класса K вычисляются оценки принадлежности объекта S классу K, имеющие вид:

$$\Gamma(S,K) = \frac{1}{|\mathcal{H}|}\sum_{H\in\mathcal{H}}\frac{1}{|T\cap K|}\sum_{S_i\in T\cap K}[H(S_i)=H(S)]\,.$$

Объект S относится к тому классу K, для которого оценка $\Gamma(S,K)$ имеет наибольшее значение. Если таких классов несколько, то алгоритм отказывается от распознавания.

Представительным набором класса K называется корректный для K эл.кл., являющийся признаковым подописанием хотя бы одного прецедента из K.

На этапе обучения процедуры голосования по представительным наборам для каждого класса K, $K \in \{K_1,\ldots,K_l\}$, строится семейство C_K представительных наборов, которое является некоторым подмножеством всех представительных наборов класса K. Распознавание объекта S осуществляется путем взвешенного голосования по построенным представительным наборам. Для каждого класса K вычисляются оценки принадлежности объекта S классу K, имеющие вид:

$$\Gamma(S,K) = \sum_{(H,\sigma)\in C_K}\alpha_{(H,\sigma)}[H(S)=\sigma]\,.$$

Вес $\alpha_{(H,\sigma)}$ положителен и, как правило, пропорционален числу прецедентов из K, выделяемых представительным набором (H,σ). Ясно, что корректность процедуры распознавания обеспечивается за счет корректности каждого представительного набора, участвующего в голосовании.

Заметим, что на практике хорошо себя зарекомендовало голосование по представительным наборам небольшого ранга. При этом, как правило, строятся семейства из так называемых тупиковых представительных наборов. Представительный набор (H,σ), $H = (x_{j_1},\ldots,x_{j_r})$, $\sigma=(\sigma_1,\ldots,\sigma_r)$, называется *тупиковым*, если для любого $q\in\{1,\ldots,r\}$ эл.кл. (H',σ'), где $H'=(x_{j_1},\ldots,x_{j_{q-1}},x_{j_{q+1}},\ldots,x_{j_r})$ и $\sigma'=(\sigma_1,\ldots,\sigma_{q-1},\sigma_{q+1},\ldots,\sigma_r)$, не является корректным. Ясно, что чем больше прецедентов выделяет представительный набор класса K, тем лучше он характеризует класс K в целом. Поэтому процедуры голосования по тупиковым представительным наборам или по представительным наборам небольшого ранга, как правило, обладают лучшей обобщающей способностью.

Как было сказано во введении, в рамках алгебро-логического подхода был построен ряд моделей логических корректоров [9–12].

Простейший логический корректор построен в [9]. На этапе обучения логического корректора для каждого класса K строится семейство W_K корректных для K наборов эл.кл. Распознавание объекта S осуществляется путем голосования по наборам эл.кл. построенных семейств. Для каждого класса K вычисляется оценка принадлежности объекта S классу K, имеющая вид:

$$\Gamma(S,K) = \frac{1}{|W_K|} \sum_{U \in W_K} \frac{1}{|T \cap K|} \sum_{S_i \in T \cap K} [U(S_i) = U(S)] .$$

Далее используется стандартное решающее правило голосования. Корректность распознающего алгоритма обеспечивается за счет корректности каждого набора эл.кл., участвующего в голосовании. Практика показывает, что качество распознавания может быть улучшено за счет построения семейств из тупиковых корректных наборов эл.кл. Корректный для K набор эл.кл. U называется *тупиковым*, если любое его собственное подмножество не является корректным для K набором эл.кл.

Заметим, что вид оценки принадлежности распознаваемого объекта S классу K, вычисляемой по семейству корректных наборов эл.кл., аналогичен виду оценки, вычисляемой в тестовом алгоритме, т. е. понятие корректного набора эл.кл. близко к понятию теста.

Фактически коррекция эл.кл. осуществляется за счет того, что при распознавании объекта S отклик $U(S)$ каждого корректного набора эл.кл. U из семейства W_K сравнивается с откликами $U(S_i)$, $S_i \in T \cap K$. Из корректности набора эл.кл. U следует, что предикат $[U(S_i) = U(S)]$ обращается в 1 только в случае, когда $S \notin T \setminus K$.

В [8] предложено два способа сравнения откликов. Первый способ основан на отношении «равно» и используется в описанной выше процедуре голосования по корректным наборам эл.кл.: распознаваемый объект S близок к прецеденту S_i по набору эл.кл. U, если $U(S_i) = U(S)$. Второй — на отношении «меньше или равно» и предполагает, что распознаваемый объект S близок к прецеденту S_i по набору эл.кл. U, если каждая координата отклика $U(S_i)$ не превосходит соответствующую координату отклика $U(S)$.

Способ сравнения откликов влияет на свойства корректирующей булевой функции. Отношение «равно», вообще говоря, не накладывает никаких ограничений на ее вид. В случае же использования отношения «меньше или равно» корректирующая функция должна быть монотонной булевой функцией. Корректный набор эл.кл. с монотонной корректирующей функцией называется *монотонным*.

В [9] помимо описанного выше логического корректора построена модель, в которой используются только корректные наборы эл.кл. с монотонной корректирующей функцией (корректор МОН). Счет на прикладных задачах показал, что корректор МОН превосходит по качеству корректор, основанный на голосовании по корректным наборам эл.кл. с произвольной корректирующей функцией.

В [11] показано, что вычисление оценки $\Gamma(S,K)$ принадлежности объекта S классу K по корректным наборам эл.кл. можно осуществлять не только на основании сравнения откликов объекта S с откликами прецедентов из K, но также сравнивая их с откликами прецедентов не из K. На этом принципе построен корректор АМОН, в котором в качестве корректирующей функции использовалась монотонная булева функция. В случае двух классов корректоры МОН и АМОН эквивалентны. Если же в задаче более двух классов, то в ряде случаев корректор АМОН опережает корректор МОН.

Построение семейств корректных наборов эл.кл. с хорошей распознающей способностью является сложной дискретной задачей [8]. Каждый корректный для K набор эл.кл. однозначно соответствует покрытию булевой матрицы L_K, специальным образом построенной по обучающей выборке. Каждому столбцу матрицы L_K соответствует один из эл.кл. Каждая строка L_K образована одной из пар прецедентов $S_i \in T \cap K$ и $S_t \in T \setminus K$. Элемент, находящийся на пересечении строки (S_i, S_t) и столбца (H, σ), равен 1, только если эл.кл. (H, σ) позволяет различить объекты S_i и S_t.

Перечислять все покрытия L_K и выбирать среди них наилучшие очень трудоемко. В [9] для построения семейства W_K используется генетический алгоритм. Кроме этого, временные затраты удается существенно сократить за счет использования только одноранговых эл.кл.

В [10] построены две модели логических корректоров, в которых голосование ведется по корректным наборам эл.кл. произвольного ранга. Для снижения временных затрат при построении голосующих семейств добавлена процедура формирования локальных базисов классов. Под локальным базисом класса K понимается корректный для K набор $\mathcal{U}_K$, состоящий из информативных эл.кл. Семейство W_K формируется из корректных наборов эл.кл., каждый из которых является подмножеством локального базиса $\mathcal{U}_K$.

Вообще говоря, идея применения локального базиса в алгебраическом подходе не нова и впервые встречается в работе К.В. Воронцова [13]. Однако применение локального базиса в логических корректорах из [10] имеет свои особенности, вследствие чего потребовалось разработать специальные алгоритмы формирования локального базиса, лучшим из которых оказался алгоритм, основанный на методе бустинга [14]. Отметим, что один из простейших способов построения локального базиса является его случайный выбор. Этот метод был успешно реализована в [12] при построении стохастического логического корректора МОНС, который опережает по качеству распознавания корректор МОН, в основном благодаря снятию ограничения на ранг эл.кл.

Метод бустинга в [10] используется не только для построения локального базиса, но и для итеративного формирования семейств голосующих наборов эл.кл. Вообще говоря, метод бустинга является универсальным методом построения алгоритмов взвешенного голосования по базовым распознающим алгоритмам произвольного типа. При обучении логического корректора на каждой итерации ищется корректный набор эл.кл. такой, что его добавление в семейство наилучшим образом компенсирует ошибки ранее построенных наборов. Пополнение семейств останавливается при достижении требуемого качества или после выполнения заданного числа итераций. Каждый набор эл.кл. получает «оптимальный» вес, и при распознавании объекта осуществляется взвешенное голосование по построенным наборам. Одним из достоинств бустинга является то, что с его помощью удается построить семейства из «непохожих» наборов эл.кл.

3 Логический корректор общего вида

3.1 Основные понятия и обозначения

Пусть $K \in \{K_1, \ldots, K_l\}$. Введем обозначения $\overline{K} = M \setminus K$, $\mathbb{K}^+ = \{K_1, \ldots, K_l\}$, $\mathbb{K}^- = \{\overline{K}_1, \ldots, \overline{K}_l\}$ и $\mathbb{K}^\pm = \mathbb{K}^+ \cup \mathbb{K}^-$.

Из соображения удобства перейдем на язык предикатов. Рассмотрим произвольный предикат $B : M \to \{0, 1\}$, заданный на множестве объектов M. Будем говорить, что B *корректен для* $K \in \mathbb{K}^\pm$, если множество прецедентов, на которых предикат B равен 1, является подмножеством либо $T \cap K$, либо $T \setminus K$. Корректный для K предикат B будем

называть *представительным для* $K \in \mathbb{K}^{\pm}$, если существует прецедент $S_i \in T \cap K$ такой, что $B(S_i) = 1$.

Понятия корректного эл.кл., представительного набора, теста и корректного набора эл.кл. могут быть переформулированы на языке предикатов.

Элементарный классификатор (H, σ) корректен для K (является представительным набором класса K) тогда и только тогда, когда предикат $B(S) = [H(S) = \sigma]$ является корректным (представительным) для K.

Набор признаков H является тестом тогда и только тогда, когда для любого $K \in \mathbb{K}^{+}$ и любого прецедента $S_i \in T \cap K$ предикат $B_i(S) = [H(S_i) = H(S)]$ корректен для K.

Пусть $U = ((H_1, \sigma_1), \ldots, (H_d, \sigma_d))$ — набор эл.кл. и $F(t_1, \ldots, t_d)$ — булева функция от d переменных. Обозначим через $F(U)$ предикат, задаваемый композицией $F(U(S)) = F([H_1(S) = \sigma_1], \ldots, [H_d(S) = \sigma_d]),\ S \in M$.

Набор эл.кл. U корректен для класса $K \in \mathbb{K}^{+}$ тогда и только тогда, когда существует булева функция F такая, что предикаты $F(U)$ и $1 - F(U)$ являются представительными соответственно для K и $\overline{K}$.

Ослабим условия, которым удовлетворяет корректный набор эл.кл. Набор эл.кл. U будем называть *полукорректным* для $K \in \mathbb{K}^{\pm}$, если существует булева функция F такая, что предикат $F(U)$ является представительным для K. Функция F называется *корректирующей* для набора U относительно класса K. В общем случае корректирующая функция F определена неоднозначно, поскольку ее значения заданы лишь в точках $U(S_i)$, $S_i \in T \backslash K$, в которых $F(U(S_i)) = 0$.

Ясно, что каждый корректный для K набор эл.кл. является полукорректным как для K, так и для $\overline{K}$. Набор эл.кл., состоящий из одного представительного набора класса K, является полукорректным для K.

3.2 Информативность предиката

Пусть объект $S_i \in T$ имеет неотрицательный вес w_i. Обозначим $\boldsymbol{w} = (w_1, \ldots, w_m)$. Пусть B — предикат на множестве объектов M и $K \in \mathbb{K}^{\pm}$. Введем зависящие от взвешенной выборки $(T, \boldsymbol{w})$ функционалы

$$P(B, K) = \sum_{S_i \in K} w_i B(S_i)\,; \quad N(B, K) = \sum_{S_i \in \overline{K}} w_i B(S_i)\,.$$

Потребуем, чтобы веса объектов из T удовлетворяли дополнительному условию нормировки, $w_1 + \cdots + w_m = 1$. Тогда $\boldsymbol{w}$ можно интерпретировать как распределение вероятностей объектов из T, и значение $P(B, K)$ будет равно вероятности того, что случайно выбранный из T объект принадлежит K и выделяется предикатом B. Очевидно, $N(B, K) = P(B, \overline{K})$, т. е. $N(B, K)$ — вероятность того, что случайно выбранный из T объект не принадлежит K и выделяется предикатом B.

Вероятность того, что предикат B выделяет случайно выбранный из T объект S_i только в случае, когда S_i лежит в K, равна

$$\sum_{S_i \in K} w_i B(S_i) + \sum_{S_i \notin K} w_i (1 - B(S_i)) = P(B, K) - N(B, K) + \sum_{S_i \notin K} w_i\,.$$

Вероятность указанного события является естественной характеристикой качества предиката B относительно K. Чем она больше, тем лучше предикат B подходит для описания K. Разность $P(B, K) - N(B, K)$ будем называть *информативностью* предиката B

для K и обозначать через $I(B,K)$. Очевидно, если предикат B представителен для K, то $N(B,K)=0$ и $I(B,K)=P(B,K)$.

3.3 Корректные предикаты специального вида

Рассмотрим множество бинарных логических операций $\mathcal{O}=\{o(x,y):\{0,1\}^2\to\{0,1\}\}$, которое состоит из 16 элементов (отношений). Пусть $O=(o_1,\ldots,o_d)$ — набор операций из $\mathcal{O}$ и $\alpha=(\alpha_1,\ldots,\alpha_d)$, $\beta=(\beta_1,\ldots,\beta_d)$ — бинарные векторы. Введем обозначение:

$$O(\alpha,\beta)=\bigwedge_{j=1}^{d}o_j(\alpha_j,\beta_j)\,.$$

Пусть G — набор объектов из M, U — набор эл.кл., O — набор отношений из $\mathcal{O}$ и длины наборов U и O совпадают. Построим предикат

$$B_{(U,O,G)}(S)=\bigvee_{S'\in G}O(U(S'),U(S)).$$

Выявим условия, при которых предикат $B_{(U,O,G)}(S)$ корректен для $K\in\mathbb{K}^{\pm}$.

Пусть G_1 и G_2 — множества объектов из M, U — набор эл.кл., O — набор операций из $\mathcal{O}$ и длины наборов U и O совпадают. Будем говорить, что набор эл.кл. U *отделяет* объекты из G_1 от объектов из G_2 с помощью набора бинарных логических операций O, если не существует двух объектов $S'\in G_1$ и $S''\in G_2$, для которых выполняется равенство $O(U(S'),U(S''))=1$.

В частности, когда набор эл.кл. U отделяет прецеденты из $T\cap K$ от прецедентов из $T\setminus K$ с помощью набора O, состоящего из одинаковых бинарных логических операций, совпадающих с отношением «равно» («меньше или равно»), U является (монотонным) корректным для класса K.

Утверждение 1. *Пусть $K\in\mathbb{K}^{\pm}$, G — набор объектов из M и набор эл.кл. U отделяет объекты из G от прецедентов из $\overline{K}$ с помощью набора операций O.*

Тогда предикат $B_{(U,O,G)}(S)$ корректен для K, и набор эл.кл. U является полукорректным для K с корректирующей функцией

$$F_{(U,O,G)}(t_1,\ldots,t_d)=\bigvee_{S'\in G}O(U(S'),(t_1,\ldots,t_d))\,.$$

Доказательство. Справедливость утверждения 1 очевидно вытекает из конструкции предиката $B_{(U,O,G)}(S)$. ■

Далее рассматривается ряд полезных свойств корректных предикатов вида $B_{(U,O,G)}(S)$. Пусть B_1 и B_2 — предикаты, корректные для K. Будем говорить, что B_1 *эквивалентен* B_2, если для любого прецедента S_i из K выполняется $B_1(S_i)=B_2(S_i)$. В случае, когда предикаты $B_{(U,O,G)}(S)$ и $B_{(U',O',G')}(S)$ эквивалентны, есть смысл отдавать предпочтение тому предикату, который имеет более простую конструкцию. Докажем несколько свойств, связанных с отношением эквивалентности предикатов.

Утверждение 2. *Пусть $U=((H_1,\sigma_1),\ldots,(H_d,\sigma_d))$ — набор эл.кл., $O=(o_1,\ldots,o_d)$ — набор отношений из $\mathcal{O}$, G — набор объектов из M и $B_{(U,O,G)}(S)$ — корректный для K предикат. Тогда выполняются следующие свойства.*

1. *Если для некоторого объекта $S^*\in G$ найдется $j\in\{1,\ldots,d\}$ такой, что $o_j(B_{(H_j,\sigma_j)}(S^*),0)=o_j(B_{(H_j,\sigma_j)}(S^*),1)=0$, то предикаты $B_{(U,O,G)}(S)$ и $B_{(U,O,G\setminus\{S^*\})}(S)$ эквивалентны.*

2. *Если наборы O' и U' получаются путем удаления соответственно из O операции o_j и из U эл.кл. (H_j, σ_j) таких, что $\forall S \in G$, $o_j(B_{(H_j,\sigma_j)}(S), 0) = o_j(B_{(H_j,\sigma_j)}(S), 1) = 1$, то предикаты $B_{(U,O,G)}(S)$ и $B_{(U',O',G)}(S)$ эквивалентны.*

Доказательство. Первое свойство следует из того, что $\forall S \in M$, $O(U(S^*), U(S)) = 0$, т. е. слагаемое, соответствующее объекту S^*, можно не включать в дизъюнкцию, задающую предикат $B_{(U,O,G)}(S)$. Второе свойство вытекает из очевидного тождества $O'(U(S), U(S')) = O(U(S), U(S'))$, $\forall S \in G, \forall S' \in M$. ■

Утверждение 2 фактически дает два правила «упрощения» предикатов вида $B_{(U,O,G)}(S)$. Из наборов U, O и G можно удалять элементы, не влияющие на результат применения предиката к прецедентам.

Возможен другой путь упрощения предиката $B_{(U,O,G)}(S)$. Рассмотрим множество операций $\mathcal{O}^* = \{[x \leqslant y], [x \geqslant y], [x \vee y], [\neg x \vee \neg y]\}$. Каждая операция из $\mathcal{O}^*$ принимает нулевое значение всего лишь на одной паре значений аргументов. Например, $[x \leqslant y] = 0$, только если $x = 1$ и $y = 0$. Легко убедиться, что любую операцию o из $\mathcal{O}$ можно представить в виде $o(x, y) = o_1(x, y) \wedge \cdots \wedge o_u(x, y)$, $\{o_1, \ldots, o_u\} \subseteq \mathcal{O}^*$.

Утверждение 3. *Пусть U — набор эл.кл., O — набор операций из $\mathcal{O}$, G — набор объектов из M и $B_{(U,O,G)}(S)$ — корректный для K предикат.*

Тогда существуют набор отношений O' из $\mathcal{O}^*$ и набор эл.кл. U' такие, что предикаты $B_{(U,O,G)}(S)$ и $B_{(U',O',G)}(S)$ эквивалентны.

Доказательство. Построим требуемые наборы O' и U' соответственно из наборов O и U по следующему правилу. Каждую операцию o_j в O, не принадлежащую $\mathcal{O}^*$, заменим на набор операций $\{o'_1, \ldots, o'_u\} \subseteq \mathcal{O}^*$ таких, что $o_j(x, y) = o'_1(x, y) \wedge \cdots \wedge o'_u(x, y)$, и каждой операции $o'_v, v \in \{1, \ldots, u\}$ сопоставим эл.кл. (H'_v, σ'_v), совпадающий с эл.кл. (H_j, σ_j). Полученные в результате замен наборы O' и U' и будут определять предикат $B_{(U',O',G)}(S)$, эквивалентный предикату $B_{(U,O,G)}(S)$. ■

Утверждение 3 позволяет при построении предикатов вида $B_{(U,O,G)}(S)$ не использовать отношения из $\mathcal{O} \setminus \mathcal{O}^*$.

Утверждение 4. *Пусть $U = ((H_1, \sigma_1), \ldots, (H_d, \sigma_d))$ — набор эл.кл., $O = (o_1, \ldots, o_d)$ — набор отношений из $\mathcal{O}$, G — набор объектов из M и $B_{(U,O,G)}(S)$ — корректный для K предикат. Тогда выполняются следующие свойства.*

1. *Для любого подмножества $G' \subset G$ предикат $B_{(U,O,G')}(S)$ корректен для K.*
2. *Для любого эл.кл. (H', σ') и любого отношения o' из $\mathcal{O}^*$ предикат $B_{(U',O',G)}(S)$, $U' = ((H_1, \sigma_1), \ldots, (H_d, \sigma_d), (H', \sigma'))$, $O' = (o_1, \ldots, o_d, o')$, корректен для K.*

Доказательство. Доказательство основывается на том, что предикат $B_{(U,O,G)}(S)$ выделяет все объекты, выделяемые и предикатом $B_{(U,O,G')}(S)$, и предикатом $B_{(U',O',G)}(S)$. ■

Пусть $U = ((H_1, \sigma_1), \ldots, (H_d, \sigma_d))$ — набор эл.кл., $O = (o_1, \ldots, o_d)$ — набор отношений из $\mathcal{O}$, G и G^* — наборы объектов из M такие, что $G \subseteq G^*$. Корректный для K предикат $B_{(U,O,G)}(S)$ будем называть *тупиковым относительно* G^*, если выполняются два условия:

1) для любого объекта $S^* \in G^* \setminus G$ предикат $B_{(U,O,G\cup\{S^*\})}(S)$ не является корректным для K;
2) для любого $j \in \{1, \ldots, d\}$ предикат $B_{(U',O',G)}(S)$, порожденный наборами $U' = ((H_1, \sigma_1), \ldots, (H_{j-1}, \sigma_{j-1}), (H_{j+1}, \sigma_{j+1}), \ldots (H_d, \sigma_d))$, $O' = (o_1, \ldots, o_{j-1}, o_{j+1}, \ldots, o_d)$, не является корректным для K.

Тупиковый корректный для K относительно $T \cap K$ предикат будем просто называть тупиковым. Пусть $G_1 \subseteq G_2 \subseteq M$. Обозначим через $\mathcal{P}_K(G_1, G_2)$ множество всех корректных для K предикатов вида $B_{(U,O,G)}(S)$, для которых $G_1 \subseteq G \subseteq G_2$. Будем оценивать информативность предикатов из $\mathcal{P}_K(G_1, G_2)$, полагая, что контрольная выборка совпадает с основной. Нетрудно видеть, что при $T^* = T$ функция информативности $I(B_{(U,O,G)}, K)$ совпадает с функцией $P(B_{(U,O,G)}, K)$, которая на множестве $\mathcal{P}_K(G_1, G_2)$ достигает локального максимума в каждом тупиковом относительно G_2 предикате. Обозначим через $\mathcal{P}^*_K(G_1, G_2)$ множество тупиковых относительно G_2 предикатов из $\mathcal{P}_K(G_1, G_2)$.

Пусть предикат $B_{(U,O,G)}(S)$ корректен для K. Исследуем свойства корректирующей функции $F_{(U,O,G)}$ полукорректного набора эл.кл. U. В частности, выясним условия, при которых $F_{(U,O,G)}$ является монотонной или поляризуемой булевой функцией (булева функция $F(t_1, \ldots, t_d)$ называется *поляризуемой*, если для некоторого бинарного вектора $(\alpha_1, \ldots, \alpha_d)$ функция $F(t_1 \oplus \alpha_1, \ldots, t_d \oplus \alpha_d)$ монотонна). Разделим множество отношений $\mathcal{O}^*$ на два подмножества $\mathcal{O}^*_0 = \{[x \geqslant y], [\neg x \vee \neg y]\}$ и $\mathcal{O}^*_1 = \{[x \leqslant y], [x \vee y]\}$.

Утверждение 5. *Пусть $B_{(U,O,G)}(S)$ — корректный для K предикат. Тогда имеют место следующие два критерия.*

1. *Корректирующая функция $F_{(O,U,G)}$ является монотонной тогда и только тогда, когда предикат $B_{(U,O,G)}(S)$ эквивалентен предикату $B_{(U',O',G)}(S)$ такому, что каждое отношение из набора O' принадлежит $\mathcal{O}^*_1$.*
2. *Корректирующая функция $F_{(O,U,G)}$ является поляризуемой тогда и только тогда, когда предикат $B_{(U,O,G)}(S)$ эквивалентен предикату $B_{(U',O',G)}(S)$, $U' = ((H'_1, \sigma'_1), \ldots, (H'_u, \sigma'_u))$, $O' = (o'_1, \ldots, o'_u)$, такому, что каждое отношение из набора O' принадлежит $\mathcal{O}^*$, и в наборе U' не существует двух одинаковых эл.кл. (H'_i, σ'_i) и (H'_t, σ'_t), для которых $o'_i \in \mathcal{O}^*_0$ и $o'_t \in \mathcal{O}^*_1$.*

Доказательство. Заметим, что $[x \leqslant y] = [\neg x \vee y]$ и $[x \geqslant y] = [x \vee \neg y]$.

Докажем первый критерий. Использование отношений из $\mathcal{O}^*_1$ гарантирует, что в дизъюнктивную нормальную форму корректирующей функция $F_{(U',O',G)}$ не будут входить переменные с отрицанием, что эквивалентно монотонности $F_{(U',O',G)}$.

Второй критерий справедлив, поскольку переменные корректирующей функции $F_{(U',O',G)}$, соответствующие одинаковым эл.кл. набора U', либо входят в $F_{(U',O',G)}$ только с отрицанием (им соответствуют операции из $\mathcal{O}^*_0$), либо входят только без отрицания (им соответствуют операции из $\mathcal{O}^*_1$). ■

Проиллюстрируем на примере с модельной задачей распознавания преимущества предикатов вида $B_{(U,O,G)}$.

Пример 1. Рассмотрим задачу распознавания с двумя классами и двумя признаками, изображенную на рис. 1. Построим следующие наборы эл.кл.:

1. Набор $U_1 = ([x_2 = 0],\ [x_1 = 0],\ [x_1 = 1],\ [x_1 = 2],\ [x_1 = 3],\ [x_1 = 4])$ принадлежит семейству монотонных корректным для класса K_1 наборов эл.кл.
2. Набор $U_2 = ([x_1 = 5],\ [x_1 = 0],\ [x_2 = 0])$ принадлежит семейству корректных для класса K_1 наборов эл.кл. и не является монотонным.
3. Набор $U_3 = ([x_1 = 5],\ [x_2 = 0],\ [x_1 = 0])$ отделяет прецеденты из K_1 от прецедентов из K_2 с помощью набора отношений $O = ([x \geqslant y], [x \leqslant y], [x \leqslant y])$.

Отметим, что наборы U_1 и U_2 являются наименее «громоздкими» представителями своих семейств. Выпишем предикаты, которые порождаются наборами U_1, U_2, U_3 и преце-

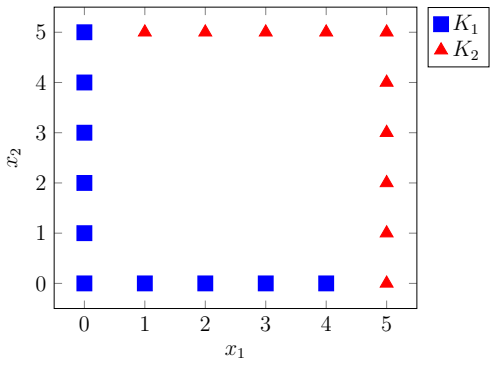


Рис. 1 Модельная задача распознавания с двумя классами и двумя признаками из примера 1

дентами класса K_1. Оценим информативность этих предикатов по обучающей выборке T, полагая, что вес каждого объекта равен $1/20$.

1. Набор U_1 и прецеденты из K_1 порождают следующие предикаты: $[x_1 = 0]$, $[x_1 = 0 \wedge x_2 = 0]$, $[x_1 = 1 \wedge x_2 = 0]$, $[x_1 = 2 \wedge x_2 = 0]$, $[x_1 = 3 \wedge x_2 = 0]$, $[x_1 = 4 \wedge x_2 = 0]$. Первый предикат имеет информативность 0,3, информативности остальных равны 0,05.
2. Набор U_2 и прецеденты из K_1 порождают предикаты $[x_1 \neq 5 \wedge x_1 \neq 0 \wedge x_2 = 0]$, $[x_1 = 0 \wedge x_2 \neq 0]$ и $[x_1 = 0 \wedge x_2 = 0]$, информативности которых соответственно равны 0,2, 0,25 и 0,05.
3. Набор U_3 и прецеденты из K_1 порождают предикаты $[x_1 \neq 5 \wedge x_2 = 0]$ и $[x_1 = 0]$, информативности которых соответственно равны 0,25 и 0,3. Отметим, что приведены все предикаты $B_{(U_3,O,\{S_i\})}, S_i \in K_1$.

Видно, что набор U_3 порождает более лаконичные предикаты с высокой информативностью. Это становится возможным благодаря тому, что с помощью расширенного набора отношений удается одним предикатом проверить как наличие, так и отсутствие некоторого признакового подописания у распознаваемого объекта.

3.4 Голосование по представительным предикатам

Опишем логический корректор, основанный на голосовании по предикатам вида $B_{(U,O,G)}(S)$.

На *этапе обучения* для каждого класса $K \in \mathbb{K}^+$ строятся два семейства Z_K и $Z_{\overline{K}}$ предикатов на множестве объектов M. Каждый предикат семейства $Z_K, K \in \mathbb{K}^{\pm}$, является представительным для K. Для каждого предиката $B \in Z_K$ задается положительный вес α_B.

Семейства предикатов Z_K, $K \in \mathbb{K}^{\pm}$, формируются итеративно. При инициализации берутся $Z_K := \varnothing$, $K \in \mathbb{K}^{\pm}$. На итерации $t \geqslant 1$ по некоторому правилу выбираются $K \in \mathbb{K}^{\pm}$ и подмножества прецедентов $G_1 \subseteq G_2 \subseteq T \cap K$. Далее осуществляется поиск одного или нескольких предикатов из $\mathcal{P}^*_K(G_1, G_2)$ с высокой информативностью. Каждый найденный предикат B получает вес α_B, вычисляемый по определенному правилу, и добавляется в семейство Z_K. Если не выполнен критерий останова, то происходит переход к следующей итерации.

Этап обучения имеет несколько параметров:

1) правило выбора $K \in \mathbb{K}^{\pm}$ и подмножеств прецедентов $G_1 \subseteq G_2 \subseteq T \cap K$ на каждой итерации;
2) алгоритм поиска корректных предикатов с высокой информативностью;

3) правило вычисления весов предикатов;
4) критерий останова обучения.

При распознавании осуществляется взвешенное голосование по предикатам, построенным на этапе обучения. Возможны два режима распознавания: базовый и аддитивный.

1. В *базовом режиме* для распознаваемого объекта S вычисляются оценки $\Gamma(S,K)$ принадлежности объекта S классу $K \in \mathbb{K}^+$, имеющие вид:

$$\Gamma(S,K) = \sum_{B \in Z_K} \alpha_B B(S) - \sum_{B \in Z_{\overline{K}}} \alpha_B B(S)\,.$$

2. В *аддитивном режиме* для распознаваемого объекта S и каждого предиката $B_{(U,O,G)} \in \in Z_K$, $K \in \mathbb{K}^{\pm}$, вычисляется оценка

$$\gamma(S, B_{(U,O,G)}) = \frac{1}{|G|} \sum_{S_i \in G} O(U(S_i), U(S))\,.$$

Затем для каждого класса $K \in \mathbb{K}^+$ вычисляется оценка $\Gamma(S,K)$ принадлежности объекта S классу K, имеющая вид:

$$\Gamma(S,K) = \sum_{B \in Z_K} \alpha_B \gamma(S,B) - \sum_{B \in Z_{\overline{K}}} \alpha_B \gamma(S,B)\,.$$

Описанный распознающий алгоритм будем называть *логическим корректором общего вида*. Для обеспечения его корректности достаточно, чтобы было справедливо

Утверждение 6. *Пусть A — логический корректор общего вида и $\{Z_K,\ K \in \mathbb{K}^{\pm}\}$ — семейства предикатов, по которым осуществляется голосование при распознавании объектов. Алгоритм A корректен, если для любого класса $K \in \mathbb{K}^+$ и любого прецедента $S_i \in K$ выполняется одно из двух условий:*

1) в семействе Z_K найдется предикат, выделяющий S_i;
2) для каждого $\overline{K'} \neq \overline{K}$, $\overline{K'} \in \mathbb{K}^-$, в семействе $Z_{\overline{K'}}$ найдется предикат, выделяющий S_i.

Доказательство. Пусть P — семейство предикатов на множестве объектов M и S — объект из M. Введем обозначение $b(P,S) = \{B \in P : B(S) = 1\}$.

Зафиксируем класс $K \in \mathbb{K}^+$ и объект $S_i \in K$.

1) Если $b(P_K, S_i) \neq \varnothing$, то $\Gamma(S_i, K) > 0$. Поскольку $\forall K' \in \mathbb{K}^+ \setminus \{K\}$, $\Gamma(S_i, K') \leqslant 0$, отступ $\Delta(S_i, K) > 0$, и объект S_i распознается алгоритмом A правильно.

2) Если $\forall \overline{K'} \in \mathbb{K}^- \setminus \{\overline{K}\}$, $b(Z_{\overline{K'}}, S_i) \neq \varnothing$, то $\forall K'' \in \mathbb{K}^+ \setminus \{K\}$, $\Gamma(S_i, K'') < 0$. Поскольку $\Gamma(S_i, K) \geqslant 0$, отступ $\Delta(S_i, K) > 0$, и объект S_i распознается алгоритмом A правильно. ■

3.5 Построение корректных предикатов

В работе [8] построение тупиковых корректных наборов эл.кл. сводится к поиску неприводимых покрытий булевой матрицы, построенной специальным образом по обучающей выборке. В данном подразделе выполняется аналогичное сведение построения корректного для K предиката вида $B_{(U,O,G)}$ к поиску покрытия булевой матрицы. Отдельно рассматривается вопрос поиска в семействах $\mathcal{P}_K(G_1,G_2)$ и $\mathcal{P}^*_K(G_1,G_2)$ предикатов с наибольшей информативностью.

Пусть $L = \|a_{ij}\|$ — булева матрица размера $m \times n$. Говорят, что столбец с номером j покрывает строку с номером i булевой матрицы L, если $a_{ij} = 1$. Обозначим через $R_0(L,J)$

набор строк матрицы L, непокрытых ни одним столбцом из J. *Покрытием* булевой матрицы L называется набор столбцов J такой, что каждую строку матрицы L покрывает хотя бы один столбец из J, т. е. $R_0(L,J)=\varnothing$. Обозначим через $\mathcal{C}(L)$ набор покрытий булевой матрицы L. Покрытие J матрицы L называется *неприводимым*, если любое его собственное подмножество не является покрытием матрицы L. Обозначим через $\mathcal{P}(L)$ набор неприводимых покрытий булевой матрицы L.

Пусть $K\in\mathbb{K}^{\pm}$, $G_1\subseteq G_2\subseteq T\cap K$. Покажем, что каждый тупиковый корректный для K предикат из $\mathcal{P}_K(G_1,G_2)$ однозначно соответствует неприводимому покрытию булевой матрицы, построенной по прецедентной информации и зависящей от G_1 и G_2.

Обозначим множество всех эл.кл. через $\mathcal{U}^*$. Построим булеву матрицу $L_{T\setminus K}(G_1,G_2)$ по следующему правилу. Каждой строке матрицы $L_{T\setminus K}(G_1,G_2)$ сопоставим пару обучающих объектов (S_i,S_t) таких, что $S_i\in G_2$ и $S_t\in T\setminus K$. Столбцы матрицы $L_K(G_1,G_2)$ будут иметь один из двух типов. Каждому столбцу первого типа сопоставим тройку (H,σ,o), где (H,σ) — эл.кл. из $\mathcal{U}^*$ и o — отношение из $\mathcal{O}^*$. Каждому столбцу второго типа — прецедент S_j из $G_2\setminus G_1$. Элемент матрицы $L_{T\setminus K}(G_1,G_2)$, расположенный на пересечении строки (S_i,S_t) и столбца первого типа (H,σ,o), равен $1-o(B_{(H,\sigma)}(S_i),B_{(H,\sigma)}(S_t))$. Элемент матрицы $L_{T\setminus K}(G_1,G_2)$, расположенный на пересечении строки (S_i,S_t) и столбца второго типа S_j, равен $[i=j]$. Матрицу, построенную по указанному правилу, принято называть *матрицей сравнения*.

Утверждение 7. *Пусть $K\in\mathbb{K}^{\pm}$, $G_1\subseteq G\subseteq G_2\subseteq T\cap K$, $U=((H_1,\sigma_1),\ldots,(H_d,\sigma_d))$ — набор эл.кл. и $O=(o_1,\ldots,o_d)$ — набор отношений из $\mathcal{O}^*$.*

Предикат $B_{(U,O,G)}$ является (тупиковым относительно G_2) корректным для K тогда и только тогда, когда набор столбцов матрицы $L_{T\setminus K}(G_1,G_2)$, соответствующих прецедентам из $G_2\setminus G$ и тройкам $(H_1,\sigma_1,o_1),\ldots,(H_d,\sigma_d,o_d)$, является (неприводимым) покрытием матрицы $L_{T\setminus K}(G_1,G_2)$.

Доказательство. Обозначим $J=(G_2\setminus G)\cup\{(H_1,\sigma_1,o_1),\ldots,(H_d,\sigma_d,o_d)\}$. Корректность предиката $B_{(U,O,G)}$ для K по определению означает, что выполняется $B_{(U,O,G)}(S_t)=0$, $\forall S_t\in T\setminus K$. Поскольку верны тождества

$$B_{(U,O,G)}(S)=\bigvee_{S_i\in G}O(U(S_i),U(S))=\bigvee_{S_j\in G_2}[S_j\notin G_2\setminus G]\,O(U(S_j),U(S))\,,$$

корректность предиката $B_{(U,O,G)}$ эквивалентна условию

$$\bigvee_{S_j\in G_2}\bigvee_{S_t\in T\cap K}[S_j\notin G_2\setminus G]\,O(U(S_j),U(S_t))=0\,. \tag{1}$$

Отрицая левую и правую часть равенства (1), получаем условие

$$\bigwedge_{S_j\in G_2}\bigwedge_{S_t\in T\cap K}[S_j\in G_2\setminus G]\vee\neg o_1(B_{(H_1,\sigma_1)}(S_j),B_{(H_1,\sigma_1)}(S_t))\vee\cdots\vee\neg o_d(B_{(H_d,\sigma_d)}(S_j),B_{(H_d,\sigma_d)}(S_t))=$$
$$=1,$$

которое равносильно тому, что набор столбцов J покрывает матрицу $L_{T\setminus K}(G_1,G_2)$.

Из определения тупикового корректного предиката легко выводится, что корректный для K предикат $B_{(U,O,G)}$ является тупиковым относительно G_2 тогда и только тогда, когда при удалении любого столбца из J получается набор, не являющийся покрытием $L_{T\setminus K}(G_1,G_2)$, т. е. J — неприводимое покрытие матрицы $L_{T\setminus K}(G_1,G_2)$. ■

Пусть $K \in \mathbb{K}^{\pm}$, $G_1 \subseteq G_2 \subseteq T \cap K$. Рассмотрим задачу построения предиката $B_{(U,O,G)}$ из $\mathcal{P}_K(G_1, G_2)$, обладающего максимальной информативностью.

В случае, когда логический корректор используется в базовом режиме распознавания, информативность корректного для K предиката $B_{(U,O,G)}$ будем оценивать значением $I(B_{(U,O,G)}, K)$. Ставится оптимизационная

Задача 1.

$$I(B_{(U,O,G)}) \underset{B_{(U,O,G)} \in \mathcal{P}_K(G_1,G_2)}{\to} \max .$$

В аддитивном режиме более адекватную оценку информативности предиката $B_{(U,O,G)}$ дает функционал

$$\hat{I}(B_{(U,O,G)}, K) = \hat{P}(B_{(U,O,G)}, K) - \hat{N}(B_{(U,O,G)}, K) ,$$

где

$$\hat{P}(B_{(U,O,G)}, K) = \sum_{S \in G} P(B_{(U,O,\{S\})}, K) , \quad \hat{N}(B_{(U,O,G)}, K) = \sum_{S \in G} N(B_{(U,O,\{S\})}, K) .$$

Задача 2.

$$\hat{I}(B_{(U,O,G)}) \underset{B_{(U,O,G)} \in \mathcal{P}_K(G_1,G_2)}{\to} \max .$$

Задачи 1 и 2 могут быть рассмотрены в варианте, когда в качестве области поиска предикатов вместо $\mathcal{P}_K(G_1, G_2)$ берется его подмножество $\mathcal{P}^*_K(G_1, G_2)$, состоящее из тупиковых относительно G_2 предикатов.

Сформулируем две дискретные оптимизационные задачи, являющиеся специальными разновидностями задачи о поиске покрытий булевой матрицы.

Задача 3 (поиск набора столбцов, покрывающего оптимальную комбинацию матриц). Пусть даны булевы матрицы $L_1, \ldots, L_d$ и их веса $\alpha_1, \ldots, \alpha_d$. Каждая матрица имеет n столбцов. Вес α_i матрицы L_i либо является рациональным числом, либо равен $+\infty$. Требуется найти набор столбцов $J \subseteq \{1, \ldots, n\}$ такой, что сумма весов матриц, непокрытых набором J, минимальна, т. е.

$$\sum_{i=1}^{d} \alpha_i [J \not\in \mathcal{C}(L_i)] \underset{J \subseteq \{1,\ldots,n\}}{\to} \min .$$

Очевидно, не теряя общности, можно считать, что веса всех матриц отличны от нуля, число матриц с весом $+\infty$ не превосходит единицы и ни одна из матриц не содержит нулевой строки. Случай, когда все веса положительны, тривиален, так как одним из решений всегда будет набор, состоящий из всех столбцов $\{1, \ldots, n\}$. Возможен вариант постановки задачи, когда решение должно являться неприводимым покрытием матрицы с весом $+\infty$.

Задача 4 (поиск набора столбцов, покрывающего оптимальную комбинацию строк). Пусть дана булева матрица L размера $m \times n$. Для каждой строки $i \in \{1, \ldots, m\}$ задан вес β_i, который либо является рациональным числом, либо равен $+\infty$. Требуется найти набор столбцов $J \subseteq \{1, \ldots, n\}$ такой, что сумма весов строк, непокрытых набором J, минимальна, т. е.

$$\sum_{i=1}^{m} \beta_i [i \not\in R_0(L, J)] \underset{J \subseteq \{1,\ldots,n\}}{\to} \min .$$

Снова, не теряя общности, можно считать, что веса всех строк отличны от нуля, и в L нет нулевой строки. В случае, когда веса всех строк положительны, описанная задача тривиальна, поскольку набор столбцов $\{1,\ldots,n\}$ является решением. Возможен вариант постановки задачи, когда решение должно являться неприводимым покрытием подматрицы, составленной из строк с весом $+\infty$.

Покажем, что задачи 1 и 2 сводятся соответсвенно к задачам 3 и 4. Для каждого объекта $S_i^* \in T^*$ построим матрицу сравнения $L_{\{S_i^*\}}(G_1, G_2)$. Справедливо

Утверждение 8. *Пусть $K \in \mathbb{K}^{\pm}$, $U = ((H_1,\sigma_1),\ldots,(H_d,\sigma_d))$ — набор эл.кл., $O = (o_1,\ldots,o_d)$ — набор отношений из $\mathcal{O}^*$, $G_1 \subseteq G \subseteq G_2 \subseteq T \cap K$ и $J = (G_2 \setminus G) \cup \cup\{(H_1,\sigma_1,o_1),\ldots,(H_d,\sigma_d,o_d)\}$ — покрытие матрицы $L_{T\setminus K}(G_1,G_2)$. Тогда*

$$\begin{aligned} P(B_{(U,O,G)},K) &= \sum_{S_i^* \in K} w_i[J \notin \mathcal{C}(L_{\{S_i^*\}}(G_1,G_2))]\,; \\ N(B_{(U,O,G)},K) &= \sum_{S_i^* \notin K} w_i[J \notin \mathcal{C}(L_{\{S_i^*\}}(G_1,G_2))]\,. \end{aligned}$$

Доказательство. Первого равенства следует из простой цепочки тождеств:

$$\begin{aligned} P(B_{(U,O,G)},K) &= \sum_{S_i^* \in K} w_i B_{(U,O,G)}(S_i^*) = \sum_{S_i^* \in K} w_i \bigvee_{S_j \in G} O(U(S_j),U(S_i^*)) = \\ &= \sum_{S_i^* \in K} w_i \bigvee_{S_j \in G_2} [S_j \notin G_2 \setminus G]\, O(U(S_j),U(S_i^*)) = \\ &= \sum_{S_i^* \in K} w_i \Big(1 - \bigwedge_{S_j \in G_2} [S_j \in G_2 \setminus G] \vee \neg O(U(S_j),U(S_i^*))\Big) = \\ &= \sum_{S_i^* \in K} w_i [J \notin \mathcal{C}(L_{\{S_i^*\}}(G_1,G_2))]\,. \end{aligned}$$

Равенство для $N(B_{(U,O,G)},K)$ доказывается аналогично. ■

Построим матрицу сравнения $L_{(T\setminus K)\cup T^*}(G_1,G_2)$. Аналогично утверждению 8 доказывается

Утверждение 9. *Пусть $K \in \mathbb{K}^{\pm}$, $G_1 \subseteq G \subseteq G_2 \subseteq T \cap K$, $U = ((H_1,\sigma_1),\ldots,(H_d,\sigma_d))$ — набор эл.кл., $O = (o_1,\ldots,o_d)$ — набор отношений из $\mathcal{O}^*$ и $J = (G_2 \setminus G) \cup \cup\{(H_1,\sigma_1,o_1),\ldots,(H_d,\sigma_d,o_d)\}$ — покрытие матрицы $L_{T\setminus K}(G_1,G_2)$. Найдем множество строк матрицы $L_{(T\setminus K)\cup T^*}(G_1,G_2)$, непокрытых набором столбцов J. Обозначим $R_0 = R_0(L_{(T\setminus K)\cup T^*}(G_1,G_2),J)$. Тогда*

$$\begin{aligned} \hat{P}(B_{(U,O,G)},K) &= \sum_{(S,S_i^*) \notin R_0} w_i[S_i^* \in K]\,; \\ \hat{N}(B_{(U,O,G)},K) &= \sum_{(S,S_i^*) \notin R_0} w_i[S_i^* \notin K]\,. \end{aligned}$$

Каждой матрице $L_{\{S_i^*\}}(G_1,G_2)$, $S_i^* \in T^*$, и каждой строке (S,S_i^*), $S \in G_2$, $S_i^* \in T^*$, матрицы $L_{(T\setminus K)\cup T^*}(G_1,G_2)$ припишем вес

$$\begin{cases} -w_i, & S_i^* \in K\,; \\ w_i, & S_i^* \notin K\,, \end{cases}$$

а матрице $L_{T\setminus K}(G_1, G_2)$ и каждой строке (S, S_i), $S \in G_2$, $S_i \in T \cap K$, матрицы $L_{(T\setminus K)\cup T^*}(G_1, G_2)$ — вес, равный $+\infty$.

Из утверждений 7 и 8 следует, что набор столбцов, покрывающий оптимальную комбинацию взвешенных матриц $L_{T\setminus K}(G_1, G_2), L_{\{S_1^*\}}(G_1, G_2), \ldots, L_{\{S_p^*\}}(G_1, G_2)$, является решением задачи 1.

Аналогично из утверждений 7 и 9 следует, что набор столбцов, покрывающий оптимальную комбинацию взвешенных строк матрицы $L_{(T\setminus K)\cup T^*}(G_1, G_2)$, является решением задачи 2.

Аналоги задачи 3 авторам неизвестны. Очевидно, что если матрицы с числовыми весами однострочны, то задачи 3 и 4 эквивалентны. Задача 4 обобщает ряд известных задач, однако ее исследования в приведенной постановке не проводились.

Задача 5 (Red-Blue Set Cover Problem (RBSC)). Простой вариант задачи RBSC формулируется следующим образом [16, 15]. Входом являются множество «красных» элементов R, множество «синих» элементов B и набор $\mathcal{D}$ подмножеств множества $R \cup B$. Говорят, что элемент $e \in R \cup B$ покрыт набором $\mathcal{D}' \subseteq \mathcal{D}$, если e принадлежит хотя бы одному множеству из $\mathcal{D}'$. Обозначим через $\mathcal{C}(\mathcal{D}')$ множество элементов, покрытых набором $\mathcal{D}'$. Требуется найти подмножество $\mathcal{D}'$ множества $\mathcal{D}$, которое покрывает все синие элементы и как можно меньше красных элементов, т. е.

$$|R \cap \mathcal{C}(\mathcal{D})| \underset{\mathcal{D}' \subseteq \mathcal{D}: B \subseteq \mathcal{C}(\mathcal{D}')}{\to} \min.$$

В [16] также рассматривается «взвешенный» вариант RBSC. Во взвешенном варианте RBSC каждому красному элементу присваивается положительный вес и требуется минимизировать сумму весов покрытых красных элементов.

В случае, когда часть строк матрицы L имеет вес $+\infty$, а остальные строки имеют отрицательные веса, задача 4 эквивалентна RBSC, в которой синими элементами являются строки матрицы L, имеющие вес $+\infty$, красными — остальные строки L и вес каждого красного элемента равен весу соответствующей строки, взятому с противоположным знаком.

Задача 6 (Positive–Negative Partial Set Cover Problem (±PSC)). Входом, аналогично RBSC, являются множество «красных» (отрицательных) элементов R, множество «синих» (положительных) элементов B и набор $\mathcal{D}$ подмножеств множества $R \cup B$. Требуется найти подмножество $\mathcal{D}'$ множества $\mathcal{D}$, которое покрывает как можно больше синих элементов и как можно меньше красных, т. е.

$$|R \cap \mathcal{C}(\mathcal{D})| - |B \cap \mathcal{C}(\mathcal{D})| \underset{\mathcal{D}' \subseteq \mathcal{D}: B \subseteq \mathcal{C}(\mathcal{D}')}{\to} \min.$$

Задача ±PSC изучается в [17].

В случае, когда каждая строка матрицы L имеет вес ± 1, задача 4 эквивалентна ±PSC, в которой красными элементами являются строки матрицы L, имеющие вес 1, синими — остальные строки L.

В настоящей работе для решения задач 3 и 4 использовался метод ветвей и границ на базе алгоритмов дуализации из [18]. Сложность такого варианта решения не исследовалась. Однако очевидно, что она существенно зависит от размеров входных матриц. Например, даже для сравнительно небольших прикладных задач матрица сравнения

$L_{T\setminus K}(G_1, G_2)$ может иметь достаточно большой размер, поскольку у нее $|T \setminus K||G_2|$ строк и $|\mathcal{U}^*||\mathcal{O}^*|+|G_2\setminus G_1|$ столбцов. Далее рассматриваются методики, позволяющие строить логические корректоры без использовании всех строк и столбцов матриц сравнения, работая только с их подматрицами.

4 Повышение эффективности логических корректоров

4.1 Построение семейств предикатов методом бустинга

В данном подразделе предлагается и исследуется бустинг-алгоритм построения логического корректора. Применяя бустинг для обучения логического корректора, можно одновременно решить две проблемы: сократить временные затраты и повысить качество распознавания. Временные затраты снижаются благодаря тому, что при поиске предикатов, добавляемых в семейство Z_K, вместо всей матрицы сравнения $L_{T\setminus K}(G_1, G_2)$ используется лишь часть ее строк. Качество распознавания логического корректора улучшается за счет настройки весов предикатов, а также построения семейств предикатов с высоким уровнем диверсификации. Под диверсификацией семейства Z_K подразумевается различность входящих в него предикатов. Чем разнообразнее наборы прецедентов, выделяемые различными предикатами семейства, тем лучше распознающая способность алгоритма в целом.

Обозначим через A_t логический корректор, голосующий по предикатам, построенным за t, $t \geqslant 0$, итераций. Пусть $S_i \in T$, y_i — номер класса, которому принадлежит S_i, и $K \in \mathbb{K}^+$. Обозначим через $\Gamma_t(S_i, K)$ оценку за отнесение объекта S_i к классу K, вычисляемую по семействам предикатов Z_K и $Z_{\overline{K}}$ логического корректора A_t. Далее понадобится обозначение $M_t(S_i, K) = \Gamma_t(S_i, K_{y_i}) - \Gamma_t(S_i, K)$.

Для числа ошибок и отказов алгоритма A_t на обучении справедливо неравенство

$$Q(A_t) = \sum_{i=1}^{m}[A_t(S_i) \neq y_i] \leqslant \sum_{i=1}^{m} \sum_{K'\neq K_{y_i}, K'\in\mathbb{K}^+} [M_t(S_i, K') \leqslant 0]\,,$$

которое в случае двух классов ($\mathbb{K}^+ = \{K_1, K_2\}$) обращается в равенство.

Построим логический корректор A_{t+1}, не меняя предикаты и их веса, найденные на итерациях $1, \ldots, t$. На итерации $t+1$ по некоторому правилу выберем класс $K \in \mathbb{K}^{\pm}$ и сформируем предикат B. Добавим B в семейство Z_K с весом $\alpha_B > 0$. Семейства $Z_{K'}$, $K' \neq K$, $K' \in \mathbb{K}^{\pm}$, оставим без изменений.

Предикат B и его вес α_B целесообразно выбирать так, чтобы суммарные потери $Q(A_{t+1})$ были минимальными. Однако решать оптимизационную задачу $Q(A_{t+1}) \to \min$ неудобно. В методе бустинга предлагается вместо нее решать задачу

$$\hat{Q}(A_{t+1}) = \sum_{i=1}^{m} \sum_{K'\neq K_{y_i}, K'\in\mathbb{K}^+} d(M_{t+1}(S_i, K')) \to \min\,,$$

где $d(x)$ — монотонно убывающая, дифференцируемая на $\mathbb{R}$ функция, ограничивающая сверху функцию $f(x) = [x \leqslant 0]$. В этом есть смысл, поскольку верно неравенство $Q(A_{t+1}) \leqslant \hat{Q}(A_{t+1})$.

Рассмотрим случай, когда логический корректор используется в базовом режиме. Для аддитивного режима применимы все приводимые ниже рассуждения с незначительными изменениями.

Введем вспомогательные обозначения:

$$\begin{aligned} D &= \{(S_i, K') \in T \times \mathbb{K}^+ \colon S_i \notin K'\}; \\ D(K) &= \{(S_i, K') \in D \colon K' = K \text{ или } S_i \in K\}, \quad K \in \mathbb{K}^+; \\ D(K) &= D(\overline{K}), \quad K \in \mathbb{K}^-; \\ z_i(K) &= 2[S_i \in K] - 1. \end{aligned}$$

Нетрудно показать, что если на итерации $t+1$ в семейство Z_K добавляется предикат B с весом α_B, то выполняется равенство:

$$\hat{Q}(A_{t+1}) = \sum_{(S_i,K')\in D\setminus D(K)} d(M_t(S_i, K')) + \sum_{(S_i,K')\in D(K)} d(M_t(S_i, K') + \alpha_B z_i(K) B(S_i)).$$

По теореме Лагранжа для функции $g(\alpha_B) = d(M_t(S_i, K') + \alpha_B z_i(K) B(S_i))$ при $\alpha_B > 0$ имеет место представление:

$$g(\alpha_B) = g(0) + \alpha_B g'(\xi) = d(M_t(S_i, K')) + \alpha_B z_i(K) B(S_i) d'(M_t(S_i, K') + \xi z_i(K) B(S_i)),$$

где $\xi \in (0, \alpha_B)$. Воспользуемся аппроксимацией

$$g(\alpha_B) \approx d(M_t(S_i, K')) + +\alpha_B z_i(K) B(S_i) d'(M_t(S_i, K'))$$

и вместо $\hat{Q}(A_{t+1})$ будем минимизировать

$$\sum_{(S_i,K')\in D\setminus D(K)} d(M_t(S_i, K')) + \sum_{(S_i,K')\in D(K)} d(M_t(S_i, K')) + \alpha_B z_i(K) B(S_i) d'(M_t(S_i, K')) =$$
$$= \hat{Q}(A_t) - \alpha_B \sum_{(S_i,K')\in D(K)} z_i(K) B(S_i)(-d'(M_t(S_i, K'))). \quad (2)$$

Зафиксируем вес предиката α_B и сопоставим каждому прецеденту S_i вес

$$\tilde{w}_t(S_i, K) = \sum_{(S_i,K')\in D(K)} (-d'(M_t(S_i, K'))).$$

Пусть $G \subseteq T$. Введем обозначение $\tilde{W}_t(G, K) = \sum_{S_i \in G} \tilde{w}_t(S_i, K)$. Нормируем веса объектов:

$$w_t(S_i, K) = \frac{\tilde{w}_t(S_i, K)}{\tilde{W}_t(T, K)}.$$

Для сумм нормированных весов будем использовать аналогичное обозначение: $W_t(G, K) = \sum_{S_i \in G} w_t(S_i, K)$.

Чтобы подчеркнуть, что качество предиката B оценивается на итерации t, обозначим:

$$\begin{aligned} P_t(B, K) &= \sum_{S_i \in T \cap K} w_t(S_i, K) B(S_i); \\ N_t(B, K) &= \sum_{S_i \in T \setminus K} w_t(S_i, K) B(S_i); \\ I_t(B, K) &= P_t(B, K) - N_t(B, K). \end{aligned}$$

Поскольку верно равенство

$$\sum_{(S_i,K')\in D(K)} z_i(K)B(S_i)(-d'(M_t(S_i,K'))) = \tilde{W}_t(T,K)I_t(B,K),$$

значение (2) минимально при максимальном значении информативности $I_t(B,K)$. Найдем и зафиксируем предикат B с максимальной информативностью, а затем определим значение веса α_B, доставляющее минимум $\hat{Q}(A_{t+1})$.

Наиболее простые выкладки получаются при $d(x) = e^{-x}$. Модель бустинг-алгоритмов с такой функцией потерь носит название AdaBoost. Рассмотрим эту модель более подробно.

В результате несложных преобразований получаем:

$$\hat{Q}(A_{t+1}) = \hat{Q}(A_t) + \tilde{W}_t(T,K)\left((e^{-\alpha_B}-1)P_t(B,K) + (e^{\alpha_B}-1)N_t(B,K)\right). \tag{3}$$

При условии, что $P_t(B,K) > N_t(B,K) > 0$, минимальное значение $\hat{Q}(A_{t+1})$ по α_B достигается в точке

$$\alpha_B = \frac{1}{2}\ln\frac{P_t(B,K)}{N_t(B,K)}.$$

Однако если предикат B корректен для K, то $N_t(B,K) = 0$. Чтобы избежать появление неопределенностей, будем вычислять вес предиката α_B по другой формуле, для ввода которой потребуются следующие обозначения:

$$\begin{aligned} w_t^*(G,K) &= \frac{1}{2}\min_{S_i\in G} w_t(S_i,K); \\ N_t^*(B,K) &= \max\{N_t(B,K), w_t^*(T,K)\}. \end{aligned}$$

Если выполнятся неравенство $P_t(B,K) > N_t^*(B,K)$, то вес

$$\alpha_B = \frac{1}{2}\ln\frac{P_t(B,K)}{N_t^*(B,K)} \tag{4}$$

определен и положителен.

Введем вспомогательное обозначение:

$$J_t(B,K) = \frac{\tilde{W}_t(T,K)}{\hat{Q}(A_t)}\left(\sqrt{P_t(B,K)} - \sqrt{N_t^*(B,K)}\right)^2.$$

Утверждение 10. *Пусть после $t_0 \geqslant 0$ итераций построен логический корректор A_{t_0} и после $t > t_0$ итераций построен логический корректор A_t.*

Если при построении A_t на каждой итерации i, $t_0 < i \leqslant t$, в некоторое семейство Z_K добавлялся предикат B с весом α_B, найденным по формуле (4)*, и при этом всякий раз выполнялось неравенство*

$$J_{i-1}(B,K) > \frac{\ln\hat{Q}(A_{t_0})}{t-t_0},$$

то распознающий алгоритм A_t корректен.

Доказательство. Подставив (4) в (3), можно убедиться, что верно неравенство

$$\hat{Q}(A_i) \leqslant \hat{Q}(A_{i-1})(1 - J_{i-1}(B,K))\,. \tag{5}$$

Обозначим $\delta = \ln(\hat{Q}(A_{t_0}))/(t - t_0)$. Из (5) получаем цепочку неравенств:

$$Q(A_t) \leqslant \hat{Q}(A_t) < \hat{Q}(A_{t_0})(1-\delta)^{t-t_0} \leqslant \hat{Q}(A_{t_0})e^{-\delta(t-t_0)} = 1.$$

Значение $Q(A_t)$ должно быть целым числом; следовательно, $Q(A_t) = 0$. Доказательство закончено. ■

Следствие 1. *Пусть после $t > 0$ итераций построен логический корректор A_t. Если при построении A_t на каждой итерации i, $1 \leqslant i \leqslant t$, в некоторое семейство Z_K добавлялся предикат B с весом α_B, найденным по формуле* (4)*, и при этом всякий раз выполнялось неравенство $J_{i-1}(B,K) > (\ln m)/t$, то распознающий алгоритм A_t корректен.*

Утверждение 10 и его следствие позволяют заменить требование корректности всех используемых при голосовании предикатов другим условием, как правило, более мягким. Далее будет показано, что это зачастую сокращает временные затраты обучения логического корректора за счет использования при поиске предиката подматрицы матрицы сравнения, составленной из относительно небольшой части ее строк.

Пусть $G \subseteq T$, $K \in \mathbb{K}^{\pm}$, $G^+ \subseteq T \cap K$ и $G^- \subseteq T \setminus K$. Введем обозначения:

$$\begin{aligned} W_t^*(G,K) &= \max\{W_t(G,K), w_t^*(G,K)\}\,; \\ \delta_t(G^+,G^-,K) &= \frac{\tilde{W}_t(T,K)}{\hat{Q}(A_t)}\left(\sqrt{W_t(G^+,K)} - \sqrt{W_t^*(T\setminus K\setminus G^-,K)}\right)^2. \end{aligned}$$

Утверждение 11. *Пусть $K \in \mathbb{K}^{\pm}$, набор эл.кл. U отделяет набор обучающих объектов $G \subseteq T \cap K$ от набора обучающих объектов $G^- \subseteq T \setminus K$ с помощью набора отношений O и предикат $B = B_{(U,O,G)}$. Тогда верно неравенство $J_t(B,K) \geqslant \delta_t(G,G^-,K)$.*

Доказательство. Из условия утверждения и конструкции предиката $B_{(U,O,G)}$ следуют оценки $P_t(B_{(U,O,G)},K) \geqslant W_t(G,K)$ и $N_t^*(B_{(U,O,G)},K) \leqslant W_t^*(T\setminus K\setminus G^-,K)$, которых достаточно для завершения доказательства. ■

Несложно убедиться, что набор эл.кл. $U = ((H_1,\sigma_1),\ldots,(H_d,\sigma_d))$ отделяет обучающие объекты из $G \subseteq T\cap K$ от обучающих объектов из $G^- \subseteq T\setminus K$ с помощью набора отношений $O = (o_1,\ldots,o_d)$ тогда и только тогда, когда набор $((H_1,\sigma_1,o_1),\ldots,(H_d,\sigma_d,o_d))$ покрывает подматрицу $L_{G^-}(G_1^+,G_2^+)$, $G_1^+ \subseteq G \subseteq G_2^+$, матрицы сравнения $L_{T\setminus K}(G_1^+,G_2^+)$. Подматрица $L_{G^-}(G_1^+,G_2^+)$ имеет $|G_2^+||G^-|$ строк.

Процедура `BuildPrettyGoodPredicate`, представленная на схеме алгоритма 1, использует «жадную» стратегию поиска таких $K \in \mathbb{K}^{\pm}$, $G_1^+ \subseteq G_2^+ \subseteq T \cap K$, $G^- \subseteq T \setminus K$, для которых $\delta_t(G_1^+,G^-,K) > \delta$, значение $|G_2^+||G^-|$ близко к наименьшему. Затем по матрице сравнения $L_{G^-}(G_1^+,G_2^+)$ ищется предикат $B = B_{(U,O,G)}$ с наибольшей информативностью $I_t(B,K)$ такой, что $G_1^+ \subseteq G \subseteq G_2^+$ и набор эл.кл. U отделяет прецеденты из G от прецедентов из G^- с помощью набора отношений O.

Рассмотрим следующие величины, зависящие от обучающей выборки:

$$\delta^* = \frac{(\sqrt{m}-1)^2}{lm}\,; \quad t^* = \frac{\ln m}{\delta^*}\,.$$

Алгоритм 1 Построение предиката с достаточно большим значением $J_t(B,K)$

ПРОЦЕДУРА `BuildPrettyGoodPredicate`($\mathbb{K}^{\pm}$, T, t, δ, r)
Параметры: $\mathbb{K}^{\pm}$ — классы и их дополнения; T — обучающая выборка; t — число выполненных итераций; $\delta > 0$ — пороговый параметр; r — рекомендуемая мощность G_2^+;
Выход: B — предикат, добавляемый в семейство Z_K, $K \in \mathbb{K}^{\pm}$, такой, что $J_t(B,K) > \delta$;

инициализировать $\mathbb{K}_t(\delta) := \{K \in \mathbb{K}^{\pm} \colon \delta_t(T \cap K, T \setminus K, K) > \delta\}$;
если $\mathbb{K}_t(\delta) = \varnothing$ **то** $\mathbb{K}_t(\delta) := \mathbb{K}^{\pm}$;
выбрать случайный K из распределения вероятностей $W_t(T \cap K, K), K \in \mathbb{K}_t(\delta)$;
упорядочить объекты $T \cap K$ и $T \setminus K$ по убыванию весов $w_t(S_i, K)$;
найти числа r_1 и r_2 такие, что
1) набор G_1^+, состоящий из первых r_1 объектов упорядоченного $T \cap K$ и набор G^-, состоящий из первых r_2 объектов упорядоченного $T \setminus K$, удовлетворяют условию $\delta_t(G_1^+, G^-, K) > \delta$,
2) значение $r_1 r_2$ минимально и
3) $r_1 \leqslant r$;
если найти r_1 и r_2, удовлетворяющие пункту 3), не удалось, **то**
    найти числа r_1 и r_2 такие, что выполнен пункт 1) и значения r_1 и r_2 минимальны;
    в качестве G_1^+ и G_2^+ взять первые r_1 объектов упорядоченного $T \cap K$;
иначе
    в качестве G_1^+ взять первые r_1 объектов упорядоченного $T \cap K$;
    в качестве G_2^+ взять первые r объектов упорядоченного $T \cap K$;
в качестве G^- взять первые r_2 объектов упорядоченного $T \setminus K$;
по матрице сравнения $L_{G^-}(G_1^+, G_2^+)$ найти предикат $B = B_{(U,O,G)}$ с наибольшей информативностью $I_t(B,K)$;

Алгоритм 2 Построение логического корректора методом бустинга

Вход: T — обучающая выборка;
$t_{\max}$ — максимальное число итераций;
r — рекомендуемая мощность G_2^+;
Выход: $A_{t_{\max}}$ — логический корректор;

инициализировать семейств предикатов $Z_K := \varnothing$, $\forall K \in \mathbb{K}^{\pm}$;
для $t = 1, \ldots, t_{\max}$
    вычислить веса объектов $\tilde{w}_{t-1}(S_i, K') := \exp(-M_{t-1}(S_i, K'))$, $(S_i, K') \in D$;
    найти $K \in \mathbb{K}^{\pm}$ и предикат B для добавления в Z_K вызовом процедуры `BuildPrettyGoodPredicate`($\mathbb{K}^{\pm}$, T, $t-1$, $\ln m / t_{\max}$, r);
    вычислить вес α_B по формуле 4;
    добавить B в Z_K;

Утверждение 12. *Если $\delta < \delta^*$, то процедура* `BuildPrettyGoodPredicate` *выбирает $K \in \mathbb{K}^{\pm}$ и строит предикат B такие, что $J_t(B,K) > \delta$.*

Доказательство. Заметим, что

$$\delta_t(T \cap K, T \setminus K, K) = \frac{\tilde{W}_t(T,K)}{\hat{Q}(A_t)} \left(\sqrt{W_t(T \cap K, K)} - \sqrt{w_t^*(T \setminus K, K)} \right)^2 .$$

Непосредственной проверкой можно установить, что $\tilde{W}_t(T,K_1)+\cdots+\tilde{W}_t(T,K_l)=2\hat{Q}(A_t)$ и $W_t(T\cap K,K)+W_t(T\cap\overline{K},\overline{K})=1$. Поэтому всегда найдется $K\in\mathbb{K}^{\pm}$, для которого верны неравенства:

$$\frac{\tilde{W}_t(T,K)}{\hat{Q}(A_t)}\geqslant\frac{2}{l};\quad W_t(T\cap K,K)\geqslant\frac{1}{2};\quad w_t^*(T\setminus K,K)\leqslant\frac{1}{2m},$$

из которых выводится неравенство $\delta_t(T\cap K,T\setminus K,K)\geqslant\delta^*$. Это означает, что если выполнены условия доказываемого утверждения, то процедура `BuildPrettyGoodPredicate` выбирает K из $\mathbb{K}^{\pm}$ такой, что $\delta_t(T\cap K,T\setminus K,K)>\delta$ и строит предикат B, для которого по утверждению 11 выполняется неравенство $J_t(B,K)>\delta$. Утверждение доказано. ■

Алгоритм 2 демонстрирует, как можно использовать бустинг для построения логического корректора общего вида из предикатов, необязательно являющихся корректными. Однако выход алгоритма 2 не всегда является корректной распознающей процедурой. Сформулируем достаточное условие корректности.

Теорема 1. *Если число итераций $t_{\max}>t^*$, то алгоритм 2 строит корректную процедуру распознавания.*

Доказательство. Из утверждения 12 следует, что для предиката B, строящегося на шаге 4 алгоритма 2, верно неравенство $J_{t-1}(B,K)>\ln m/t_{\max}$, $t\in\{1,\ldots,t_{\max}\}$. Таким образом, справедливы предпосылки следствия из утверждения 10, из которого заключаем, что распознающая процедура $A_{t_{\max}}$ корректна. Теорема доказана. ■

4.2 Локальные базисы классов

В данном подразделе рассматривается вопрос сокращения временных затрат поиска предикатов с высокой информативностью за счет отбрасывания части столбцов матрицы сравнения.

Пусть $K\in\mathbb{K}^{\pm}$. Обозначим матрицу сравнения $L_{T\setminus K}(T\cap K,T\cap K)$ через L_K. Каждый столбец матрицы сравнения L_K соответствует тройке (H,σ,o), где (H,σ) — эл.кл. и o — отношение из $\mathcal{O}^*$. Множество всех таких троек обозначим через $\mathcal{V}^*$. Мощность $\mathcal{V}^*$ даже в задаче с небольшим числом признаков может оказаться существенной. Большое число столбцов матрицы сравнения затрудняет поиск корректных предикатов. Предлагается использовать не всю матрицу сравнения, а лишь подматрицу, состоящую из части ее столбцов.

Набор $\mathcal{V}_K=\{(H_1,\sigma_1,o_1),\ldots,(H_d,\sigma_d,o_d)\}$ троек из $\mathcal{V}^*$ будем называть *локальным базисом класса* K, если набор эл.кл. $((H_1,\sigma_1),\ldots,(H_d,\sigma_d))$ отделяет прецеденты из $T\cap K$ от прецедентов из $T\setminus K$ с помощью набора отношений $(o_1,\ldots,o_d)$.

Ясно, что $\mathcal{V}_K$ является локальным базисом класса K тогда и только тогда, когда подматрица, составленная из столбцов $\mathcal{V}_K$ матрицы сравнения L_K, не имеет нулевых строк, т. е. для этой подматрицы существует покрытие.

Пусть $G_1\subseteq G_2\subseteq T\cap K$. Нетрудно заметить, что если $\mathcal{V}_K$ является локальным базисом класса K, то подматрица, составленная из столбцов $\mathcal{V}\cup(G_2\setminus G_1)$ матрицы сравнения $L_{T\setminus K}(G_1,G_2)$, не имеет нулевых строк. Таким образом, «в рамках» локального базиса класса K всегда можно найти корректный для K предикат $B_{(U,O,G)}$ такой, что $G_1\subseteq G\subseteq\subseteq G_2$.

Набор $\mathcal{V}\subseteq\mathcal{V}^*$, являющийся локальным базисом для каждого из классов $K_1,\ldots,K_l$, будем называть *локальным базисом задачи*. Например, набор $\mathcal{V}_1$, состоящий из троек $(H,\sigma,o)\in\mathcal{V}^*$ таких, что эл.кл. (H,σ) имеет ранг 1, является локальным базисом задачи.

Опишем универсальный метод построения локального базиса класса, состоящего из эл.кл. произвольного ранга.

Пусть $K \in \mathbb{K}^{\pm}$. Рассмотрим задачу распознавания с двумя классами K и $\overline{K}$. Построим семейство эл.кл. C_K и каждому эл.кл. $(H, \sigma) \in C_K$ присвоим ненулевой вес $\alpha_{(H,\sigma)}$. Рассмотрим распознающую процедуру

$$A_T^K(S) = \operatorname{sign}\Big(\sum_{(H,\sigma)\in C_K} \alpha_{(H,\sigma)}[H(S) = \sigma]\Big), \tag{6}$$

где $\operatorname{sign}(x)$ — функция «знак», возвращающая 1 при $x < 0$, -1 при $x < 0$ и 0 при $x = 0$. Будем считать процедуру A_T^K корректной в случае, когда $A_T^K(S_i) = 1$, $\forall S_i \in T \cap K$, и $A_T^K(S_i) = -1$, $\forall S_i \in T \setminus K$. Построим по взвешенному семейству C_K набор $\mathcal{V}_K$ такой, что каждому эл.кл. (H, σ) из C_K однозначно соответствует тройка $(H, \sigma, o) \in \mathcal{V}_K$, в которой $o = [x \leqslant y]$ при $\alpha_{(H,\sigma)} > 0$ и $o = [x \geqslant y]$ при $\alpha_{(H,\sigma)} < 0$. Очевидно, что справедливо

Утверждение 13. *Если распознающая процедура A_T^K корректна, то набор $\mathcal{V}_K$, построенный по взвешенному семейству эл.кл. C_K, является локальным базисом класса K, причем упорядоченный набор, составленный из эл.кл. семейства C_K, является корректным для K и имеет поляризуемую корректирующую функцию.*

Существует ряд методов построения корректных распознающих процедур вида (6), например бустинг или построение деревьев решений. В [10] лучшим алгоритмом построения локального базиса оказался бустинг-алгоритм BOOSTLO. В настоящей работе используется два метода: голосование по представительным наборам и бустинг-алгоритм, аналогичный BOOSTLO.

Практика показывает, что для прикладной задачи с большой значностью признаков редко удается построить небольшой локальный базис. Заметим, что при использовании бустинга для формирования семейств голосующих предикатов на каждой итерации ищется набор эл.кл. U, отделяющий некоторое подмножество прецедентов G^+ от подмножества прецедентов G^-. При этом совсем необязательно осуществлять поиск набора U в локальном базисе задачи. Целесообразно на каждой итерации формировать локальный базис, ориентированный на отделение G^+ от G^- и учитывающий текущие веса остальных прецедентов.

Были реализованы и протестированы 4 модификации логического корректора общего вида (каждая из модификаций для формирования семейств голосующих предикатов использует бустинг):

1. ОЛК1 — логический корректор, в котором предикаты строятся в рамках локального базиса задачи, состоящего из троек (H, σ, o) таких, что ранг эл.кл. (H, σ) равен 1 и $o \in \{[x \leqslant y], [x \geqslant y]\}$;
2. ОЛК2 — логический корректор, в котором предикаты строятся в рамках локального базиса задачи, построенного бустинг-алгоритмом;
3. ОЛК3 — логический корректор, в котором предикаты строятся в рамках локального базиса, формируемого на каждой итерации голосованием по представительным наборам;
4. ОЛК4 — логический корректор, в котором предикаты строятся в рамках локального базиса, формируемого на каждой итерации бустинг-алгоритмом.

5 Эксперименты

Новые логические корректоры были протестированы на прикладных задачах из репозитория UCI. В табл. 1 даны характеристики задач. В столбцах l, m, n и z приведены соответственно число классов, число строк, число столбцов и число всех представительных наборов ранга 1, характеризующее значность признаков.

Задачи по трудоемкости можно разбить на 3 группы. Задачи с номерами 1–11 имеют средний объем обучения, и поэтому для тестирования на этих задачах применяется методика 10-кратного скользящего контроля. В задачах 12 и 13 много объектов, поэтому для сокращения времени счета выборка делится только 1 раз на обучающую и тестовую. В задачах 14 и 15, наоборот, очень мало объектов, поэтому используется методика скользящего контроля по одному объекту (leave-one-out).

В тестировании помимо логических корректоров ОЛК1–ОЛК4 участвовали следующие алгоритмы распознавания:

1) Т — голосование по тестам (для каждого класса строится не более 200 тестов);
2) ПН — голосование по представительным наборам (для каждого объекта строится не более 5 представительных наборов);
3) МОН — голосование по монотонным корректным наборам эл.кл. (для каждого класса строится не более 200 наборов и эл.кл. имеют ранг 1);
4) Т* — голосование по тестам (голосующие семейства формируются бустингом);
5) ПН* — голосование по представительным наборам (голосующие семейства формируются бустингом);
6) МОН* — голосование по монотонным корректным наборам эл.кл. (голосующие семейства формируются бустингом и эл.кл. в наборах имеют ранг 1).

В табл. 2 приведены результаты счета. Показателем качества является средняя доля ошибок на тестовых выборках. Прочерки соответствуют случаям, когда алгоритм не справился с задачей за 1 ч. Время счета представлено в табл. 3.

Таблица 1 Задачи

№	Название	l	m	n	z
1.	audiology	24	226	69	161
2.	balance scale	3	625	4	20
3.	breast cancer	2	699	9	90
4.	car	4	1728	6	21
5.	dermatology	4	366	34	192
6.	house votes	2	435	16	48
7.	kr vs kp	2	3196	36	73
8.	monks-2	2	601	6	17
9.	nursery	5	12960	8	27
10.	soybean large	19	307	35	132
11.	tic tac toe	2	958	9	27
12.	optdigits	10	5620	64	914
13.	letter recognition	26	20000	16	256
14.	lenses	3	24	4	9
15.	soybean small	4	47	35	72

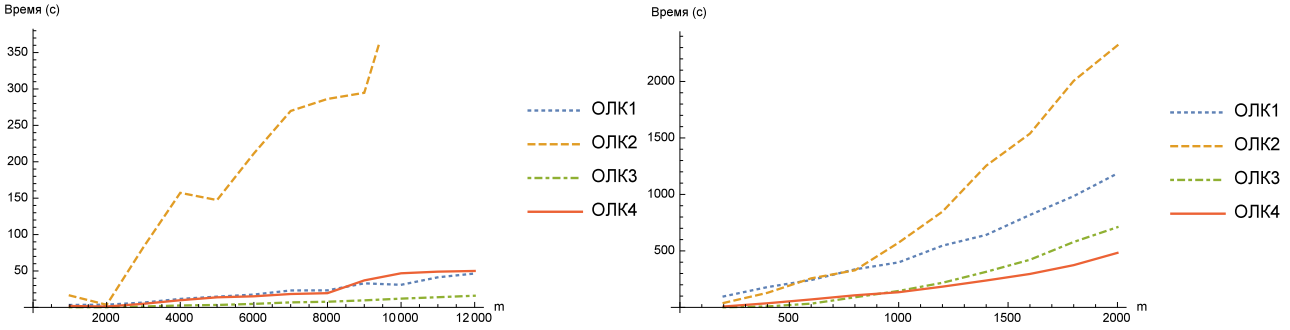
Таблица 2 Средняя частота ошибок на тестовой выборке

№	Задача	Классические			Бустинг			Логические корректоры общего вида			
		T	ПН	МОН	T*	ПН*	МОН*	ОЛК1	ОЛК2	ОЛК3	ОЛК4
1.	audiology	0,14	0,07	0,09	0,03	0,03	0,03	0,03	0,03	**0,02**	0,03
2.	b. scale	0,92	0,27	0,46	0,25	0,2	0,19	**0,18**	0,23	0,23	0,25
3.	b. cancer	0,21	0,05	0,24	0,046	**0,044**	0,057	0,061	0,059	0,065	0,059
4.	car	0,97	0,09	0,27	0,061	0,032	0,033	0,013	0,027	0,022	**0,011**
5.	dermat.	0,84	0,47	0,79	0,41	0,4	0,4	**0,39**	0,42	0,44	0,43
6.	h. votes	0,34	0,06	0,15	0,07	**0,05**	0,07	**0,05**	0,06	0,07	0,08
7.	kr-vs-kp	0,63	0,017	0,101	0,008	0,004	**0,003**	0,008	0,007	0,004	**0,003**
8.	monks-2	0,96	0,52	0,96	0,37	0,55	0,42	**0,04**	0,44	0,56	0,36
9.	nursery	0,66	0,015	0,36	0,027	0,003	0,005	0,002	—	**0,0019**	0,004
10.	soybean l.	0,19	0,094	0,131	0,075	**0,064**	0,072	0,078	0,106	0,083	0,075
11.	tic-tac-toe	0,97	0,005	0,52	0,011	0,002	0,005	0,028	0,002	**0,001**	0,007
12.	letter r.	0,52	0,21	0,63	0,21	**0,16**	0,25	—	—	0,23	0,25
13.	optdigits	0,77	0,19	0,55	0,25	0,23	0,17	0,15	—	0,27	**0,14**
14.	lenses	1	**0,21**	0,46	0,42	0,25	0,29	0,33	0,29	0,38	0,25
15.	soybean s.	0,02	**0**	**0**	**0**	**0**	**0**	**0**	0,02	0,04	**0**

Таблица 3 Время счета в секундах

№	Задача	Классические			Бустинг			Логические корректоры общего вида			
		T	ПН	МОН	T*	ПН*	МОН*	ОЛК1	ОЛК2	ОЛК3	ОЛК4
1.	audiology	1,9	**1,4**	4,6	22,7	8,9	42,2	224,1	420,2	4,1	82,4
2.	b. scale	0,6	**0,4**	2,4	0,8	1,1	2,8	132,5	1251,1	3,9	243,7
3.	b. cancer	1,2	**0,2**	7,8	0,7	0,5	5,2	108,1	110,1	1,3	51,3
4.	car	3,1	**1,3**	10,1	2,3	3,1	7,1	78,5	713,7	7,9	34,9
5.	dermat.	**2,8**	15,4	13,2	40,9	66,5	118,9	272,4	689,6	98,9	345,1
6.	h. votes	2,4	**1,1**	7,6	4,2	8,6	12,1	37,1	87,3	7,9	167,3
7.	kr-vs-kp	36,3	**10,2**	94,4	58,9	79,8	87,5	226,1	192,1	84,8	173,6
8.	monks-2	**0,5**	0,6	1,2	0,9	1,9	2,1	15,4	500,6	5,8	104,1
9.	nursery	163,9	**20,9**	595,5	87,9	89,3	224,4	452,9	—	157,9	589,1
10.	soybean l.	2,5	**2,4**	7,7	28,3	21,2	79,9	329,3	868,6	15,9	249,2
11.	tic-tac-toe	3,2	**0,6**	6,5	4,5	1,9	10,2	45,6	9,3	2,2	16,2
12.	letter r.	58,2	**47,1**	790,7	92,3	233,1	550,5	—	—	363,1	1191,1
13.	optdigits	**25,8**	636,2	277,7	249,6	1570,5	840,6	3117,2	—	2160,6	1110,8
14.	lenses	**0,01**	**0,01**	0,03	**0,01**	0,03	0,06	0,09	4,7	0,03	2,2
15.	soybean s.	0,4	**0,06**	1,1	0,8	0,1	1,5	4,8	0,3	0,1	0,4

На 14 задачах лидируют алгоритмы, в которых применяется бустинг для формирования голосующих семейств. На 11 задачах лидируют новые модели. Лучшими среди новых являются ОЛК3 и ОЛК4, в которых локальный базис формируется на каждой итерации, причем эти логические корректоры демонстрируют хорошие результаты на задачах с большой значностью признаков и имеют сравнительно небольшое время счета почти на всех задачах.



(а) nursery (б) optdigits

Рис. 2 Зависимость времени обучения логических корректоров от размера выборки

Для выявления наилучшей стратегии построения локального базиса с точки зрения времени счета проведена следующая серия экспериментов. Выбраны две задачи с достаточно большим числом объектов: nursery и optdigits. Задача nursery отличается от задачи optdigits тем, что имеет сравнительно небольшую значность и существенно неравномерное распределение объектов по классам. Для задачи nursery было сформировано 60 случайных подвыборок по 5 подвыборок каждого из размеров $1000, 2000, \ldots, 12000$. Для задачи optdigits было сформировано 50 подвыборок по 5 подвыборок каждого из размеров $200, 400, \ldots, 2000$. На рис. 2, *а* и 2, *б* изображены графики зависимости усредненного времени счета логических корректоров ОЛК1–ОЛК4 от размера подвыборки.

Очевидно, самой неудачной является модификация ОЛК2. Время работы ОЛК2 быстро увеличивается с ростом объема обучения, напрямую связанного с мощностью строящегося логическим корректором локального базиса задачи.

На задаче nursery ОЛК3 является наилучшим. Строящиеся логическим корректором ОЛК3 локальные базисы имеют небольшую мощность, поскольку в задачах с малой значностью признаков, как правило, представительные наборы имеют высокую информативность.

На задаче optdigits ОЛК4 обгоняет ОЛК3, начиная с размера подвыборки 800. Бустинг-алгоритм, использующийся в ОЛК4 для формирования локального базиса, не требует корректности эл.кл. Большая значность признаков в задаче optdigits приводит к тому, что в локальный базис, формируемый ОЛК3, попадает много малоинформативных представительных наборов, что плохо сказывается на времени счета.

6 Заключение

В работе введены понятия корректного и представительного предиката. С помощью этих понятий сформулированы классические определения теста, представительного набора, корректного эл.кл., а также определение корректного набора эл.кл.

Предложен способ конструирования корректного предиката по корректному набору эл.кл., учитывающий характер монотонности корректирующей функции по каждой ее переменной. С каждым эл.кл. набора связывается определенное отношение, использующе-еся при сравнении прецедентов с распознаваемыми объектами. Приведены условия, при которых набор эл.кл. имеет поляризуемую или монотонную корректирующую функцию. Приведен пример модельной задачи, на котором явно демонстрируются преимущества предикатов введенной конструкции.

Построена общая модель голосования по представительным предикатам, в терминах которой могут быть описаны процедуры голосования по корректным эл.кл. и по корректным наборам эл.кл.

Поиск корректных предикатов с максимальной информативностью сведен к дискретным задачам, являющимся специальными случаями известной задачи о покрытии булевой матрицы. Сложность решения этих задач существенно зависит от размеров входной матрицы. Входной матрицей при обучении логических корректоров является специальная матрица сравнения, строящаяся по прецедентной информации. Предложено использовать не всю матрицу сравнения, а лишь ее небольшую подматрицу. Набор столбцов этой подматрицы формируется путем построения локального базиса. Набор строк меняется итеративно в зависимости от весов объектов, вычисляемых бустинг-алгоритмом. Предикаты, строящиеся по подматрице, вообще говоря, не являются корректными. Однако получена теоретическая оценка числа итераций бустинг-алгоритма, достаточного для формирования семейств предикатов, голосование по которым является корректной процедурой распознавания.

Эксперименты показали, что логические корректоры общего вида на большинстве тестовых задач ошибаются реже ранее построенных моделей. Преимущество особенно заметно на задачах с большой значностью.

На время счета влияет общая стратегия и алгоритм формирования локального базиса. В случае, когда локальный базис строится для всей задачи и не меняется на последующих итерациях, его мощность, а следовательно и время обучения, оказываются достаточно большими. Если же локальный базис перестраивать на каждой итерации, настраиваясь на объекты, которые вызывают у ранее построенных предикатов наибольшие затруднения, то мощность локального базиса, как правило, не велика.

Одним из дальнейших направлений исследования видится обобщение предложенных моделей на случай, когда на множестве значений признаков заданы определенные отношения порядка. Практический интерес представляют цепи, решетки, полурешетки.

Литература

[1] *Дмитриев А. И., Журавлев Ю. И., Кренделев Ф. П.* Об одном принципе классификации и прогноза геологических объектов и явлений // Известия Сиб. отд. АН СССР, Геология и геофизика, 1968. Т. 5. С. 50–64.

[2] *Баскакова Л. В., Журавлев Ю. И.* Модель распознающих алгоритмов с представительными наборами и системами опорных множеств // ЖВМ и МФ, 1981. Т. 21. № 5. С. 1264–1275.

[3] *Дюкова Е. В., Песков Н. В.* Поиск информативных фрагментов описаний объектов в дискретных процедурах распознавания // ЖВМ и МФ, 2002. Т. 42. № 5. С. 741–753.

[4] *Журавлев Ю. И* Об алгоритмах распознавания с представительными наборами (о логических алгоритмах) // ЖВМ и МФ, 2002. Т. 42. № 9. С. 1425–1435.

[5] *Дюкова Е. В., Журавлев Ю. И., Песков Н. В., Сахаров А. А.* Обработка вещественнозначной информации логическими процедурами распознавания // Искуственный интеллект, НАН Украины, 2004. № 2. С. 80–85.

[6] *Журавлев Ю. И.* Корректные алгебры над множеством некорректных (эвристических) алгоритмов. Ч. I // Кибернетика, 1977. Т. 13. №. 4. С. 5–17.

[7] *Воронцов К. В.* Оптимизационные методы линейной и монотонной коррекции в алгебраическом подходе к проблеме распознавания // ЖВМ и МФ, 2000. Т. 40. № 1. С. 166–176.

[8] *Дюкова Е. В., Журавлев Ю. И., Рудаков К. В.* Об алгебро-логическом синтезе корректных процедур распознавания на базе элементарных алгоритмов // ЖВМ и МФ, 1996. Т. 36. № 8. С. 215–223.

[9] *Dyukova E. V., Zhuravlev Yu. I., Sotnezov M. R.* Construction of an ensemble of logical correctors on the basis of elementary classifiers // Pattern Recogn. Image Anal., 2011. Vol. 21. No. 4. P. 599–605.

[10] *Dyukova E. V., Prokofjev P. A.* Models of recognition procedures with logical correctors // Pattern Recogn. Image Anal., 2013. Vol. 23. No. 2. P. 235–244.

[11] *Дюкова Е. В., Любимцева М. М., Прокофьев П. А.* Об алгебро-логической коррекции в задачах распознавания по прецедентам // Машинное обучение и анализ данных, 2013. Т. 1. № 6. С. 705–713.

[12] *Любимцева М. М.* Логические корректоры в задачах распознавания // Сб. тезисов лучших дипломных работ факультета ВМК МГУ 2014 года. — М: МАКС ПРЕСС, 2014. С. 47–49.

[13] *Воронцов К. В.* О проблемно-ориентированной оптимизации базисов задач распознавания // ЖВМ и МФ, 1998. Т. 38. № 5. С. 870–880.

[14] *Schapire R. E., Singer Y.* Improved boosting using confidence-rated predictions // Machine Learning, 1999. Vol. 37. No. 3. P. 297–336.

[15] *Carr R. D., Doddi S., Konjevod G., Marathe M. V.* On the red-blue set cover problem // 11th ACM-SIAM Symposium on Discrete Algorithms Proceedings, 2000. P. 345–353.

[16] *Peleg D.* Approximation algorithms for the label-cover max and red-blue set cover problems // J. Discrete Algorithms, 2007. Vol. 5. No. 1. P. 55–64.

[17] *Miettinen P.* On the positive–negative partial set cover problem // Inform. Proc. Lett., 2008. Vol. 108. No. 4. P. 219–221.

[18] *Дюкова Е. В., Прокофьев П. А.* Построение и исследование новых асимптотически оптимальных алгоритмов дуализации // Машинное обучение и анализ данных, 2014. Т. 1. № 8. С. 1048–1067.

References

[1] Dmitriev, A. N., Yu. I. Zhuravlev, and F. P. Krendelev. 1966. Mathematical principles of classification of patterns and scenes. *Discrete Anal. (Inst. Mat. Sib. Otd. Akad. Nauk SSSR, Novosibirsk)* 7:3–11.

[2] Baskalova, L. V., and Yu. I. Zhuravlev. 1981. A model of recognition algorithms with representative samples and systems of supporting sets. *Comput. Math. Math. Phys.* 21(5):189–199.

[3] Dyukova, E. V., and N. V. Peskov. 2002. Search for informative fragments in descriptions of objects in discrete recognition procedures. *Comput. Math. Math. Phys.* 42(5):711–723.

[4] Zhuravlev, Yu. I. 2002. Recognition algorithms with representative sets (logic algorithms). *Comput. Math. Math. Phys.* 42(9):1372–1382.

[5] Djukova, E. V., Yu. I. Zhuravlev, N. V. Peskov, and A. A. Saharov. 2004. Processing a real-valued information with logical recognition procedures. *Artificial Intelligence* 2:80–85. (In Russian.)

[6] Zhuravlev, Yu. I. 1977. Correct algebras over sets of incorrect (heuristic) algorithms. I. *Cybernetics* 13(4):489–497.

[7] Vorontsov, K. V. 1998. Optimization methods for linear and monotone correction in the algebraic approach to the recognition problem // *Comput. Math. Math. Phys.* 40(1):159–168.

[8] Dyukova, E. V., Yu. I. Zhuravlev, and K. V. Rudakov. 1996. Algebraic-logic synthesis of correct recognition procedures based on elementary algorithms. *Comput. Math. Math. Phys.* 36(8):1161–1167.

[9] Dyukova, E. V., Yu. I. Zhuravlev, and M. R. Sotnezov. 2011. Construction of an ensemble of logical correctors on the basis of elementary classifiers. *Pattern Recogn. Image Anal.* 21(4):599–605.

[10] Dyukova, E. V., and P. A. Prokofjev. 2013. Models of recognition procedures with logical correctors. *Pattern Recogn. Image Anal.* 23(2):235–244.

[11] Djukova, E. V., M. M. Lyubimtseva, and P. A. Prokofjev. 2013. Algebraic-logical correction in recognition problems. *J. Mach. Learn. Data Anal.* 1(6):705–713.

[12] Lyubimtseva, M. M. 2014. Logical correctors in pattern recognition. *Abstracts of the best theses of the Faculty CMC MSU 2014*. 47–49. (In Russian.)

[13] Vorontsov, K. V. 1998. The task-oriented optimization of bases in recognition problems. *Comput. Math. Math. Phys.* 38(5):838–847.

[14] Schapire, R. E., and Y. Singer. 1999. Improved boosting using confidence-rated predictions. *Machine Learning* 37(3):297–336.

[15] Carr, R. D., S. Doddi, G. Konjevod, and M. V. Marathe. 2000. On the red-blue set cover problem. *11th ACM-SIAM Symposium on Discrete Algorithms Proceedings*. 345–353.

[16] Peleg, D. 2007. Approximation algorithms for the label-cover max and red-blue set cover problems. *J. Discrete Algorithms* 5(1):55–64.

[17] Miettinen, P. On the positive–negative partial set cover problem. 2008. *Inform. Proc. Lett.* 108(4):219–221.

[18] Djukova, E. V., and P. A. Prokofjev. 2014. Construction and investigation of new asymptotically optimal algorithms for dualization. *J. Mach. Learn. Data Anal.* 1(8):1048–1067.

Обзор средств визуализации тематических моделей коллекций текстовых документов*

Р. М. Айсина
rose.aysina@gmail.com
Московский государственный университет им. М. В. Ломоносова, Российская Федерация,
Москва 119991, Ленинские горы, д. 1

Тематическое моделирование является важным инструментом статистического анализа текстовых коллекций. Наглядное представление тематической модели позволяет лучше изучить кластерную структуру коллекции и оценить качество тематической модели. Средства визуализации являются неотъемлемой частью графических пользовательских интерфейсов, облегчающих тематический поиск и навигацию по коллекции. В обзоре описываются средства визуализации на основе веб-интерфейсов для иерархических, динамических и мультимодальных тематических моделей. Приводятся примеры визуализации графов и сетей, предлагается систематизация средств визуализации тематических моделей по их функциональным возможностям.

Ключевые слова: *анализ текстов; тематическое моделирование; кластеризация; научная визуализация; иерархия; граф; сеть*

Survey of visualization tools for topic models of text corpora*

R. M. Aysina
Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow 119991, Russian Federation

Topic modeling is an important tool for statistical analysis of text collections. A visual representation of a topic model enables researchers to study cluster structure of the collection and estimate quality of the topic model. Visualization tools are especially important for graphical user interfaces as they facilitate search and navigation across documents of the collection. In this survey, web-based visualization tools for topic models, including hierarchical, temporal and multimodal models, are described. Examples of graph and network visualization are presented and visualization tools are categorized according to their functionality.

Keywords: *text mining; topic modeling; clustering; scientific visualization; hierarchy; graph; network*

1 Введение

Тематическое моделирование (*topic modeling*) — одно из направлений статистического анализа текстов, активно развивающееся с конца 1990-х гг. [1]. Тематическая модель коллекции текстовых документов определяет, какие термины (ключевые слова или словосочетания) образуют каждую тему и какие темы образуют тематику каждого документа. Тематические модели применяются для выявления трендов в научных публикациях и новостных потоках, для классификации и категоризации документов, изображений и видео, в информационном поиске, в рекомендательных системах и в других приложениях.

*Работа выполнена при поддержке Российского научного фонда (проект №15-18-00091).

Разработаны сотни специализированных моделей, учитывающих различные особенности текстов естественного языка и различные виды дополнительной информации [2]. Многомодальные модели учитывают метаданные документов и позволяют определять тематику не только самих документов, но и связанных с ними *объектов* различных *модальностей* — авторов, пользователей, тэгов, источников, категорий, классов, именованных сущностей, изображений и т. д. Динамические модели учитывают время публикации документов и позволяют отслеживать изменения тематики документов и других объектов во времени. Иерархические модели строят иерархическую тематическую структуру, рекурсивно разделяя темы на подтемы. Сетевые модели учитывают взаимосвязи между документами посредством гиперссылок, цитирования, авторства или комментирования. Тематическое моделирование все чаще применяется для выявления тематических сообществ в социальных сетях.

При таком богатстве моделей и приложений возникает потребность в средствах визуализации. Чем больше объем коллекции, тем острее стоит проблема наглядного представления как исходных данных, так и результатов тематического моделирования. Визуализация обычно преследует несколько целей одновременно. Как минимум, пользователям предоставляется возможность тематического поиска и тематической навигации по коллекции.

Тематический поиск — это возможность по документу, слову, объекту или теме найти документы, слова, объекты той же или схожей тематики. В тематическом поиске, в отличие от более привычного полнотекстового поиска, запросом может быть не только короткая текстовая строка, но и документ произвольной длины. Система тематического поиска определяет тематику документа и формирует результаты поиска либо в виде ранжированного списка, либо в виде структурированного графического представления.

Тематическая навигация — это возможность легкого (по одному клику) перехода пользователя от любого визуального элемента, представляющего документ, тему, объект, термин и др., к тематически связанным с ним элементам, в частности переход от документа к списку (или иному визуальному представлению) его тем, от темы — к списку релевантных ей документов, объектов, терминов и т. д.

Простейшие средства визуализации непосредственно отображают результаты тематического моделирования — распределения тем для каждого документа и распределения терминов для каждой темы. Они могут отображаться в виде ранжированных списков либо с помощью графических средств.

Более функционально богатые средства визуализации предоставляют различные способы отображения кластерной тематической структуры коллекции. Для этого могут использоваться графы, диаграммы, матрицы, сети. Цели визуализации понимаются разными исследовательскими группами и разработчиками по-разному. Это приводит к большому разнообразию идей визуализации: от выбора отображаемых структурных особенностей тематической модели до выбора элементов графического дизайна.

Некоторые средства визуализации нацелены на упрощение экспертной оценки качества (интерпретируемости) тематической модели и даже позволяют вмешиваться в процесс построения модели, изменяя, удаляя или добавляя темы в документах или термины в темах.

В данном обзоре описаны основные идеи визуализации тематических моделей на основе веб-интерфейсов. Рассмотрены визуализаторы для плоских моделей (*Topic Model Visualization Engine, Termite System и TopicNets*), для иерархических моделей (*Hiérarchie, iVisClustering*), для динамических моделей (*TextFlow*), для иерархических динамических моделей (*HierarchicalTopics* и *RoseRiver*). В заключении приводится систематизация средств визуализации по их функциональным возможностям.

2 Вероятностное тематическое моделирование

Пусть D — множество (*коллекция*) текстовых документов, W — множество (*словарь*) всех употребляемых в них терминов (слов или словосочетаний). Каждый документ $d \in D$ представляет собой последовательность n_d терминов $w_1, \ldots, w_{n_d}$ из словаря W.

Вероятностная тематическая модель (ВТМ) описывает каждый документ d дискретным распределением $p(t \mid d)$ на множестве тем T, каждую тему $t \in T$ — дискретным распределением $p(w \mid t)$ на множестве терминов W. Можно также говорить о совместной *«мягкой» кластеризации* множества документов и множества слов по множеству кластеров-тем. «Мягкая» кластеризация означает, что каждый документ или термин нежестко приписывается какой-то одной теме, а распределяется по нескольким темам.

Множество тем T чаще всего задается как конечное множество заданной мощности. Модель сама определяет, какие слова с какими вероятностями войдут в каждую тему. Это создает *проблему интерпретируемости тем*. Тематическая модель не гарантирует, что каждая тема будет иметь содержательную интерпретацию с точки зрения людей-экспертов, понимающих тематику данной коллекции. Тема считается интерпретируемой, если по ранжированному списку наиболее релевантных слов данной темы эксперт в состоянии понять, о чем эта тема, и дать ей название [3]. При визуализации тематических моделей, построенных автоматически без участия экспертов, возникает проблема автоматического именования тем. В простейшем случае она решается путем конкатенации нескольких наиболее репрезентативных слов темы [4]. Другие подходы к автоматическому именованию тем можно найти в [5, 6, 7].

Для построения распределений тем в документах и слов в темах используются различные модели и методы. Самыми простыми являются модели *вероятностного латентного семантического анализа* PLSA (Probabilistic Latent Semantic Analysis) [8] и *латентного размещения Дирихле* LDA (Latent Dirichlet Allocation) [9]. Подавляющее большинство тематических моделей, разработанных за последние 15 лет, являются их модификациями [2]. Несмотря на различные усложнения, в большинстве моделей на выходе формируются те же структуры данных — распределения терминов в темах и тем в документах. Поэтому базовые средства визуализации этих распределений могут быть применены к большинству моделей.

На выходе более сложных моделей могут формироваться *тематические профили* $p(t \mid x)$ объектов x различных модальностей — авторов, моментов времени, категорий и т. д. Иногда их пересчитывают по формуле Байеса в распределения $p(x \mid t) = p(t \mid x)p(x)/p(t)$, чтобы показывать наиболее релевантные объекты в темах t. Распределения $p(x)$ и $p(t)$ легко оцениваются в процессе построения модели. Для отображения таких данных и их взаимосвязей разрабатываются специализированные средства визуализации.

3 Система TMVE

Система *Topic Model Visualization Engine* (TMVE) [4] — это навигатор по коллекции, имеющий два основных типа страниц: страница темы и страница документа. Есть также обзорные страницы, отображающие общую структуру коллекции. Они являются стартовыми, с них начинается работа с навигатором (рис. 1).

Страница темы разделена на три колонки. Слева находится список слов w, упорядоченных по убыванию вероятности $p(w \mid t)$ в данной теме t. По этой последовательности пользователь обычно может быстро понять, о чем тема. Названия тем формируются автоматически как тройки наиболее представительных слов. В центре находится список документов, упорядоченных по убыванию вероятности $p(d \mid t)$. Справа располагается список

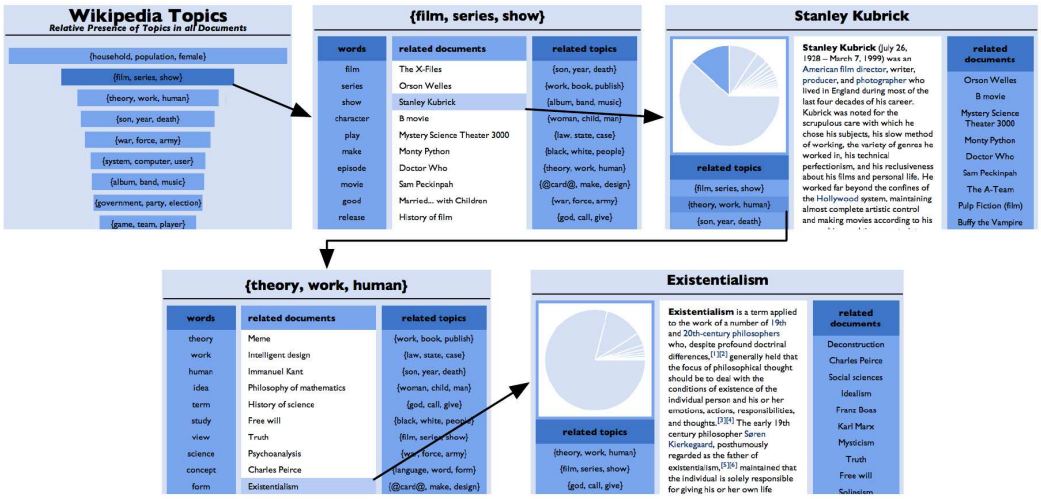


Рис. 1 Система TMVE. Навигация по Википедии (с разрешения авторов [4])

Рис. 2 Система TMVE. Страница документа (слева) и страница темы (справа) (с разрешения авторов [4])

схожих тем, имеющих наиболее близкие распределения слов. Для оценивания сходства тем s и t используется функция расстояния

$$R(s,t) = \sum_{w \in W} [p(w \mid s) > 0]\,[p(w \mid t) > 0]\,\bigl|\log p(w \mid t) - \log p(w \mid s)\bigr|.$$

Эта функция подходит для модели LDA [9], в которой вероятности $p(w \mid t)$, как правило, отличны от нуля. Однако для сравнения тем в разреженных моделях лучше использовать расстояние Хеллингера или косинусное расстояние, не содержащие логарифмов.

Страница документа. В центре отображается текст документа, слева от него — темы, к которым относится документ, и секторная диаграмма, показывающая вероятности $p(t \mid d)$ тем в данном документе. Слева находится список документов, имеющих ту же тематику, что и данный документ (рис. 2).

Обзорные страницы являются «точками входа» для навигации по коллекции. На них представлены темы, упорядоченные согласно вероятностям $p(t)$, причем размер поля пропорционален вероятности (рис. 3).

Рассмотрим взаимодействие пользователя с системой на примере визуализации статей Википедии. Вначале пользователь видит набор тем, составляющих коллекцию. Выбрав тему {*film, series, show*}, пользователь попадает на страницу документов этой темы. Затем, выбрав документ *«Stanley Kubrick»*, можно просмотреть саму статью и темы, к кото-

Рис. 3 Система TMVE. Обзорная страница (с разрешения авторов [4])

рым она относится. При выборе схожей темы {*theory, work, human*} пользователь переходит на страницы документов уже этой темы, где он, например, может прочитать статью «*Existentialism*».

Достоинства: удобный интерфейс для навигации по коллекции; имеются списки схожих документов и схожих тем; возможность просмотра любого документа; автоматическое именование тем; открытый исходный код; возможность адаптации кода для конкретной задачи (возможно создание пользовательских режимов, изменение вида входных данных, изменение алгоритма построения тематической модели). Разработчики прилагают три демонстрационные версии на различных коллекциях (включая Википедию) с исходным кодом, на основе которого можно создавать свои браузеры.

Недостатки: названия тем из трех слов не всегда адекватны; нет визуализации других модальностей, кроме слов; нет возможности изменить модель.

Ссылки:
`https://code.google.com/p/tmve` — сайт проекта;
`http://www.princeton.edu/~achaney/tmve/wiki100k/browse/topic-presence.html` — пример визуализации.

4 Система Termite

Система Termite [10] позволяет визуализировать матрицу терминов тем $p(w \mid t)$ и сравнивать темы друг с другом. Значения в матрице отображаются в виде кругов, радиусы которых пропорциональны вероятностям терминов в темах $p(w \mid t)$. Круги могут накладываться друг на друга. Пользователь может перейти к теме, нажав на круг или на название темы в матрице. При этом раскрываются два дополнительных представления темы. Первое — вероятности слов темы относительно всей коллекции, второе — документы, принадлежащие этой теме (рис. 4).

Termite может фильтровать термины, чтобы показать наиболее вероятные или значимые (*salient*) термины (рис. 5). Пользователь задает число отображаемых терминов от 10 до 250. Список самых вероятных слов содержит общие слова (*based, paper, approach*), в то время как список значимых слов, получаемый с помощью так называемой меры значимости (*saliency measure*), содержит слова, которые характерны для данной темы (*tree, context, task*).

Для определения меры значимости термина w вычисляется условная вероятность $p(t \mid w)$ и вероятность $p(t)$. Отличительность (*distinctiveness*) термина w определяется дивергенцией Кульбака–Лейблера между $p(t \mid w)$ и $p(t)$:

$$\text{distinctiveness}(w) = \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)}.$$

Значимость (*saliency*) термина w определяется следующей формулой:

$$\text{saliency}(w) = p(w) \cdot \text{distinctiveness}(w).$$

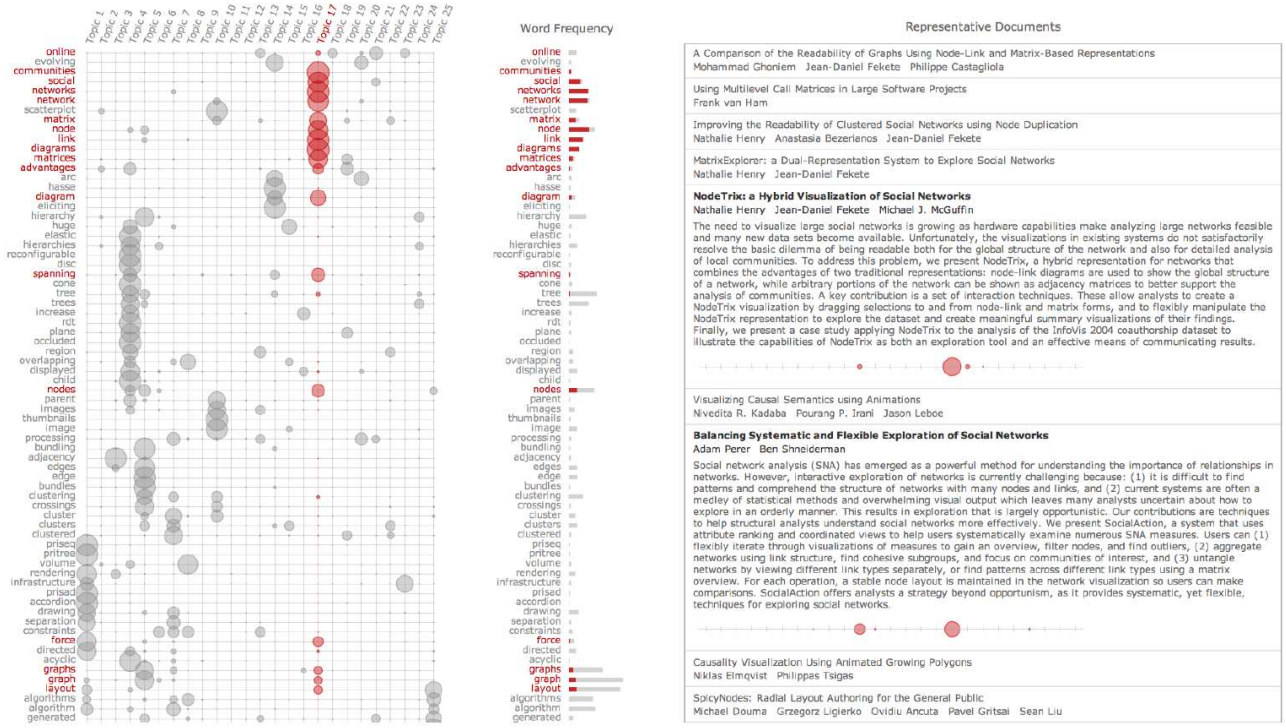


Рис. 4 Система Termite. Когда тема выбрана в матрице (слева), система отображает распределение терминов в теме и во всей коллекции (в середине) и показывает документы, наиболее соответствующие выбранной теме (справа) (с разрешения авторов [10])

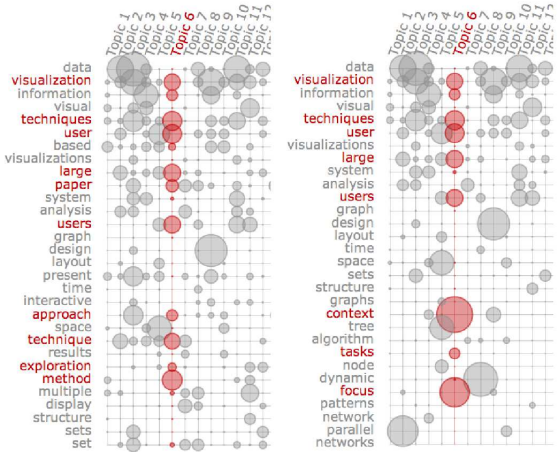


Рис. 5 Система Termite. Первые 30 частых (слева) и существенных (справа) терминов (с разрешения авторов [10])

Если генерировать более разреженную матрицу, то оценка значимости терминов обеспечивает более быструю дифференциацию тем и выявление потенциальных «ненужных» тем, в которых мало значимых терминов.

Termite также предлагает три опции для ранжирования терминов: по алфавиту, по частоте и по совместной встречаемости между парами соседних слов. Для ранжирования по совместной встречаемости оценивается вероятность того, что два слова неслучайно часто появляются вместе. Например, «*social network*» — более вероятная фраза, чем «*network social*» (рис. 6). Такое ранжирование улучшает интерпретируемость тем. Например, в те-

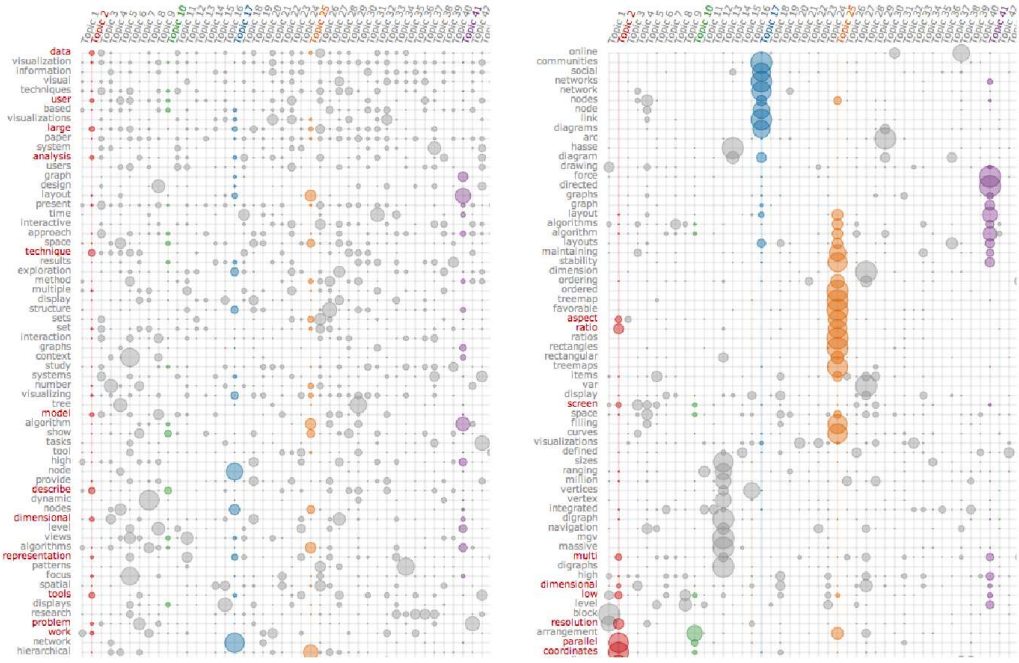


Рис. 6 Система Termite. Термины могут ранжироваться по частоте (слева) либо по совместной встречаемости (справа) (с разрешения авторов [10])

ме 6 существенными словами являются *focus* и *context* (см. рис. 6, слева), в то время как самыми частыми словами являются общие слова *technique* и *method*. В теме 2 характерными являются словосочетания *aspect ratio* и *parallel coordinates* (рис. 6, справа).

Достоинства: несколько способов оценить интерпретируемость тем; различные виды представлений для тем; удобный интерфейс; открытый исходный код, адаптируемый под конкретную задачу.

Недостатки: нет визуализации распределения тем в документах; нет возможности внесения исправлений в модель при обнаружении плохой интерпретируемости; нет автоматического именования тем; нет возможности использовать термины-словосочетания.

Ссылки:
`https://github.com/uwdata/termite-visualizations.git` — сайт проекта;
`http://vis.stanford.edu/topic-diagnostics` — пример визуализации.

5 Система TopicNets

Система TopicNets [11] предназначена для визуализации и интерактивного анализа больших коллекций документов через веб-интерфейс. TopicNets представляет документы и темы вершинами графа. Главным достоинством TopicNets является поддержка интерактивного тематического моделирования. Модель перестраивается в режиме реального времени, прямо во время визуализации, для лучшего представления подмножеств тем и документов. Для этого используется распределенная реализация алгоритма свернутой вариационной байесовской аппроксимации CVB0 (*Collapsed Variational Bayes*) [12]. Документы распределяются на несколько процессоров, и на каждом из них выполняются шаги CVB0, которые потом синхронизируются между собой.

Создание графа «документы-темы». Первым шагом создания графа является тематическое моделирование документов d, в результате которого вычисляются распределения тем в документах $p(t \mid d)$. Если $p(t \mid d)$ превышает установленный пользователем порог, то

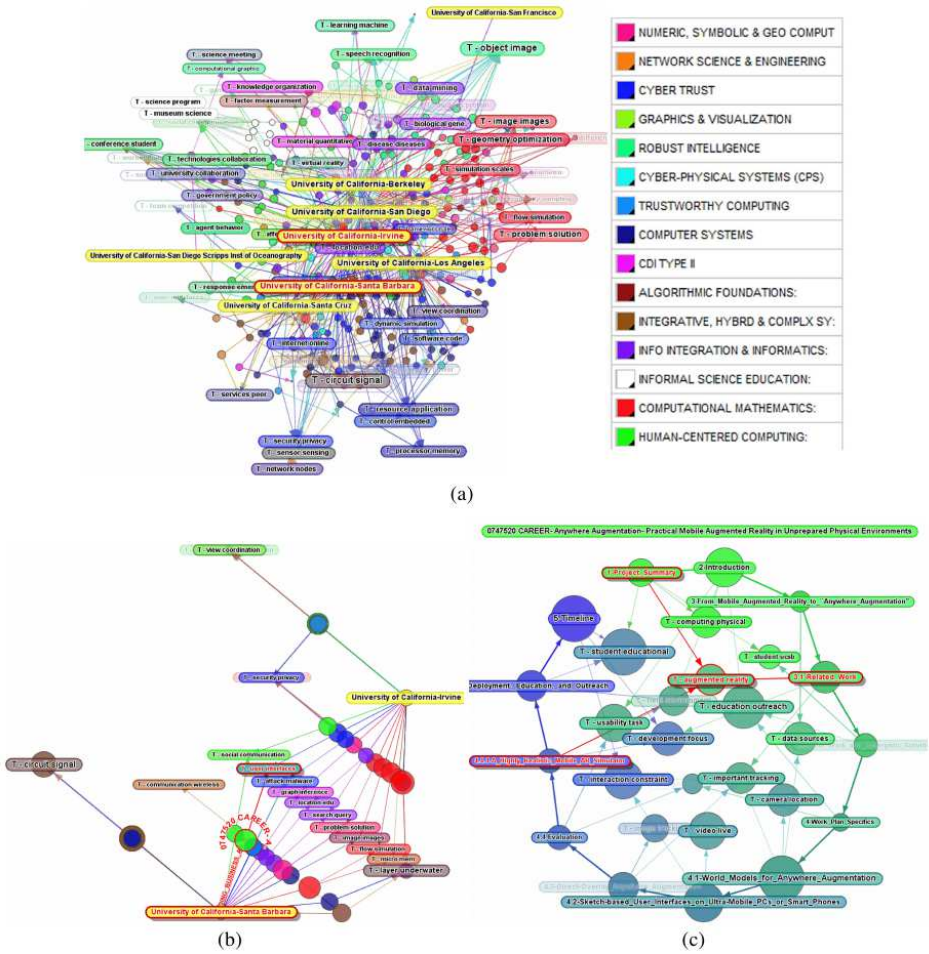


Рис. 7 Система TopicNets. Коллекция грантов NSF. Выбраны вершины, связанные с университетами. Вершины-темы обозначены буквой «T» и раскрашены согласно связанным с ними документам: (a) гранты по тематике *Computer Science*, полученные в университетах Калифорнии; (b) то же самое после того, как пользователь выбрал фрагмент графа и один отдельный грант; (c) визуализация одного документа. Вершины, обозначающие секции документа, находятся по периметру, вершины тем — внутри фигуры. На рисунке выделена одна тема и все связанные с ней вершины (с разрешения авторов [11])

в графе документ d и тема t соединяются ребром, толщина которого пропорциональна $p(t \mid d)$. Далее вершины-темы именуются первыми n наиболее вероятными словами темы (число n также устанавливается пользователем), размер вершины-темы пропорционален вероятности $p(t)$ появления темы в коллекции (рис. 7). Размер вершины-документа пропорционален длине документа.

Раскраска вершин и ребер графа. Цвет вершин-документов определяется набором цветов, который задается пользователем или формируется на основе метаданных коллекции (при их наличии). Цвета можно определить для каждого автора (автоматически или вручную), тогда вершины документов будут наследовать цвета своих авторов. При наличии нескольких модальностей, например авторов, времени создания, организаций и т. д., цвета смешиваются. Цвета интерполируются между последовательными вершинами на временной шкале (рис. 8b). Интерполяция происходит в RGB пространстве, что не дает идеального результата для каждой цветовой пары, однако пользователь может выбрать, какие два цвета смешивать, чтобы корректно отобразить временную шкалу.

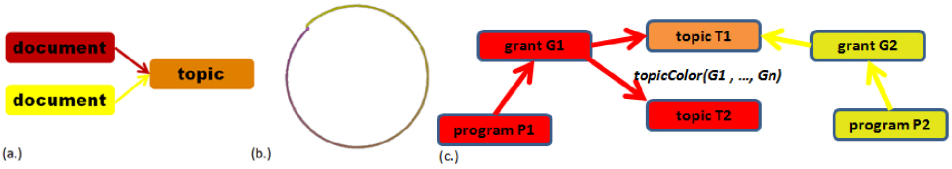


Рис. 8 Система TopicNets: (a) пример смешения цветов для раскраски вершин-тем; (b) пример интерполирования цвета для изображения временной шкалы; (c) пользовательское задание цветов для раскраски вершин-тем (с разрешения авторов [11])

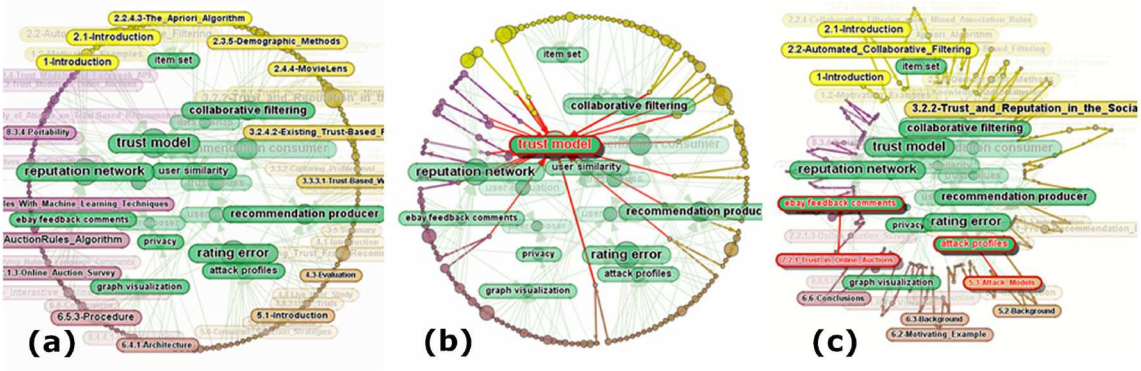


Рис. 9 Система TopicNets: (a) линейная структура одного документа; (b) распределение выделенной темы по документу; (c) размер и расположение относительно центра окружности показывают близость каждой секции и темы к главной теме документа (с разрешения авторов [11])

Ребра графа наследуют цвет вершины-документа, из которой они исходят. Цвет вершины-темы формируется путем смешения цветов ребер, входящих в эту вершину (см. рис. 8). Пользователь также имеет возможность задать свой набор цветов для тем (рис. 9).

Расположение схожих тем. Для определения сходства тем в TopicNets используется симметризованная дивергенция Кульбака–Лейблера между каждой парой распределений слов по темам $p(w \mid t)$. Результирующая матрица несходства тем является входными данными для алгоритма многомерного шкалирования (*multidimensional scaling*), который определяет позицию для каждой вершины-темы. Эти вершины затем фиксируются в соответствующей позиции, и применяется стандартный силовой алгоритм [13] для расстановки вершин-документов в пространстве соответствующей вершины-темы (при этом в качестве расстояния между темой и документом используется вероятность появления данной темы в данном документе).

Ранжирование документов. Если документы ранжированы по дате публикации, TopicNets расставляет их вершины по окружности (рис. 10). Это имеет определенные преимущества перед более привычным отображением временной шкалы в виде прямой линии. Вершина-тема может соединяться с большим числом документов, расположенных далеко друг от друга на прямой. Если тем много, изображение становится запутанным. Окружность сделана слегка винтообразной, чтобы вершины первого и последнего документа не встретились в одной точке.

Кроме сохранения хронологического порядка алгоритм старается расположить схожие по тематике вершины близко друг к другу. В результате темы, которые появлялись в конкретный период времени, оказываются ближе к окружности (в секторе, соответствующем этому периоду), в то время как темы, которые появлялись постоянно, располагаются ближе к центру. Если тема появляется в документах, расположенных диаметрально

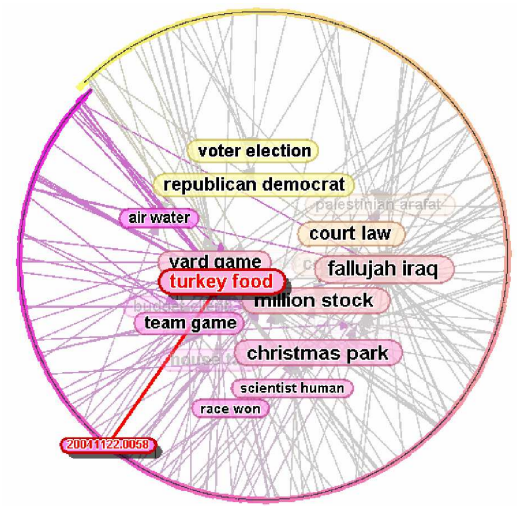


Рис. 10 Система TopicNets. Пример графа в TopicNets, показывающего темы новостей NY Times за ноябрь 2004 (с разрешения авторов [11])

на окружности, то она также будет находиться в центре, что может ввести пользователя в заблуждение. Однако если эта тема появлялась редко, то размер ее вершины будет маленьким. Таким образом, вершина-тема большого размера и близкая к центру может считаться наиболее релевантной для всей коллекции.

Фильтрация графа. В большинстве случаев пользователя интересует не весь граф тематической модели, а только его фрагмент. TopicNets позволяет задавать нужный фрагмент с помощью поискового запроса по названиям вершин или по наиболее вероятным словам тем. Затем можно выбрать нужные из найденных вершин и визуализировать только связанные с ними вершины, скрыв (по желанию) остальной граф. При этом система плавно трансформирует старый граф в новый. Щелчком мыши пользователь может перейти в другую часть графа или вернуться к предыдущей визуализации. Пользователь может выбрать нужные вершины в графе и скрыть другие вершины, не связанные с данными.

TopicNets позволяет добавлять различные типы метаданных и визуализировать их на исходном графе. Например, если в коллекции имеется информация об авторах документов, то возможно свертывание вершин-документов в новые вершины авторов, при этом строится новый граф, соединяющий авторов и темы, и все принципы, описанные выше, сохраняются и для этого графа.

Визуализация одного документа. Все перечисленные способы визуализации могут быть применены не только к ранжированному множеству документов, но и их содержимому одного отдельного документа.

Веб-архитектура. Многие приложения для визуализации тематических моделей, способные сгенерировать интерактивный граф с высокой степенью масштабируемости, устанавливаются как отдельное приложение или плагин для веб-браузера, что может быть ресурсоемким для клиентской машины. TopicNets построен на архитектуре WiGis, разработанной авторами TopicNets для визуализации графов на основе AJAX[1]. Это позволяет запускать TopicNets из веб-браузера и масштабировать граф, состоящий из сотен тысяч

[1] AJAX (*Asynchronous Javascript and XML*) — технология построения интерактивных пользовательских интерфейсов для веб-приложений, поддерживающая фоновый обмен данными между браузером и веб-сервером и постепенную загрузку веб-страниц.

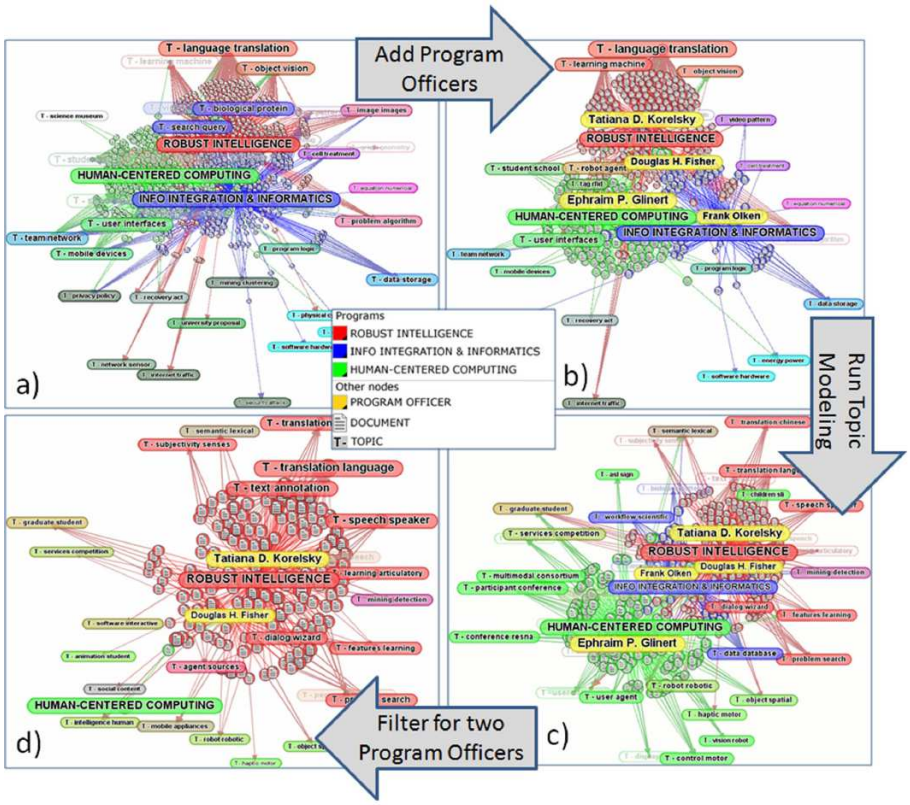


Рис. 11 Система TopicNets. Коллекция грантов NSF (с разрешения авторов [11])

документов. Все вычисления и формирование новых изображений для графа происходят на удаленном сервере.

Рассмотрим пример работы пользователя с коллекцией грантов NSF[2] (рис. 11). Сначала пользователь выбирает три темы, соответствующие программам фонда (рис. 11a), затем добавляет авторов — руководителей грантов (рис. 11b) и запускает повторное моделирование для подграфа, связанного только с авторами *Fisher, Korelsky, Glinert, Olken*. Затем он удаляет темы, не связанные с выбранным подграфом (*processor memory*), и добавляет новые (*language natural*) (рис. 11c). Далее он выбирает авторов *Fisher* и *Korelsky*, так как их вершины близко расположены друг к другу. На рис. 11d становится видно, что эти руководители занимаются программой *Robust Intelligence*, при этом тематика работ *Fisher* также пересекается с *Human-Centered Computing*, так как его вершина соединена с вершинами зеленого цвета.

Достоинства: богатый дружественный интерфейс; интеракивность; возможность учета модальностей; единые способы визуализации коллекции и отдельного документа; возможность масштабирования визуализации и уточнения тематической модели в режиме реального времени; запуск непосредственно из веб-браузера.

Недостатки: при большом числе тем и документов круговая визуализация временной шкалы становится неадекватной; сложность установки; необходимость устанавливать сторонние приложения; сложно адаптируемый код.

Ссылки:
`https://code.google.com/p/topicnets` — страница проекта TopicNets;

[2]Коллекция доступна на `http://www.nsf.gov/awardsearch/`.

`https://code.google.com/p/wigis` — страница проекта WiGis;
`http://youtu.be/-Sgq-msjd-Y` — пример визуализации TopicNets.

6 Система iVisClustering

Система iVisClustering [14] позволяет полностью контролировать процесс кластеризации коллекции документов. Пользователь может создавать и удалять кластеры, разделять кластеры на более мелкие, производить повторную кластеризацию.

Система имеет несколько модулей для визуализации кластерной структуры коллекции.

Модуль *Cluster Relation View* представляет результаты кластеризации в виде графа, вершины которого соответствуют документам (рис. 12A). Для плоского размещения вершин графа используется силовой алгоритм многомерного шкалирования. Каждая вершина-документ изображается цветным кругом, документы одинакового цвета принадлежат одному кластеру. Длина ребра между двумя вершинами пропорциональна оценке сходства документов. Пользователь может задать пороговое значение сходства для отображения ребер: ребра, длина которых меньше этого значения, не будут отображаться.

Ключевые слова кластера изображаются в «общей вершине», которая представляет собой прямоугольник с цветной границей. Если при большом числе кластеров этот вид оказался перегруженным, то доступно еще одно представление — *Cluster Summary View*, где «общие вершины» отображаются в виде таблицы (рис. 12C).

Модуль *Parallel Coordinates View*. В этом представлении каждая вертикальная ось соответствует теме, а каждая линия обозначает документ (рис. 12D). Цвет линии соответствует цвету кластера, которому принадлежит документ. По вертикальной оси откладываются значения $p(t \mid d)$ для каждого документа d. Если линия документа имеет несколько

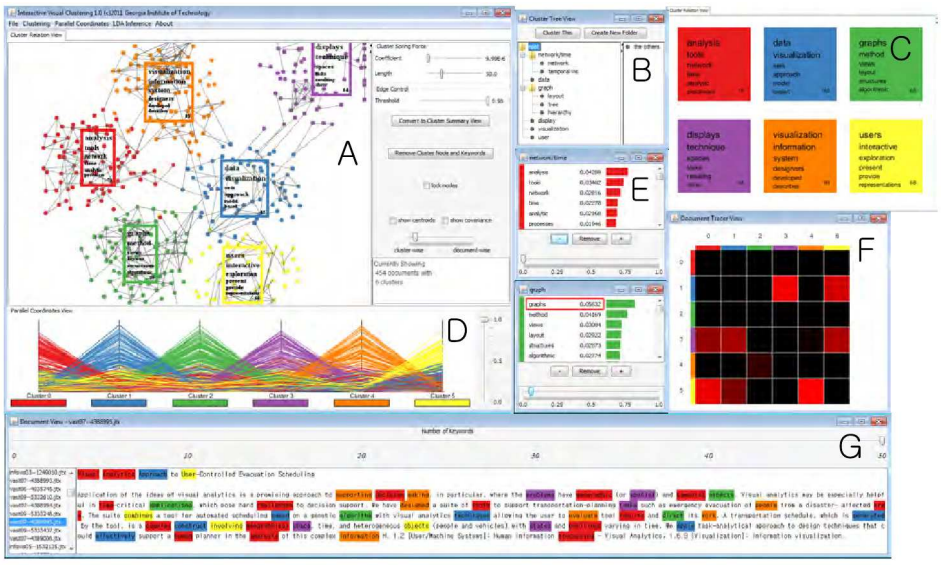


Рис. 12 Система iVisClustering: (A) *Cluster Relation View* — представление кластеров в виде графа; (B) *Cluster Tree View* — представление иерархии в виде дерева; (C) *Cluster Summary View* — представление кластеров без документов и связей между ними; (D) *Parallel Coordinates View* — отображает темы, из которых состоят документы; (E) *Term-Weight View* — визуализация распределений $p(w \mid t)$ для каждой темы; (F) *Document Tracer View* — тепловая карта, показывающая, какие документы перешли из одного кластера в другой; (G) *Document View* позволяет просматривать отдельные документы (с разрешения авторов [14])

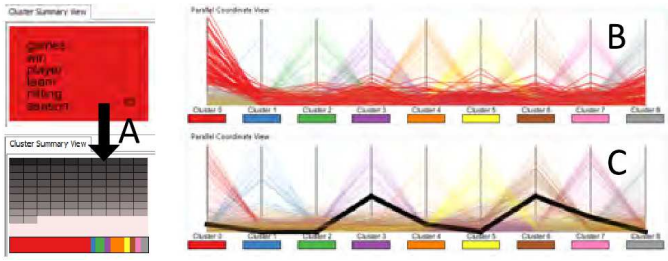


Рис. 13 *X-ray режим:* (A) изображение таблицы документов, под таблицей находится цветовой спектр, показывающий, из каких тем состоит документ; (B) *Parallel Coordinates View* для кластера; (C) *Parallel Coordinates View* для одного документа, он изображен толстой черной линией (с разрешения авторов [14])

«пиков», то это значит, что он принадлежит нескольким темам. Пользователь может задать пороговое значение вероятности $p(t \mid d)$, ниже которого линии не отображаются.

Режим отображения *X-ray* активируется при наведении курсора на «общую вершину» кластера (рис. 13A), при этом документы кластера выделяются в *Parallel Coordinates View* (рис. 13B). Режим *X-ray* представляет документы кластера в виде таблицы, каждая ячейка которой соответствует документу. Чем выше вероятность $p(t \mid d)$, тем темнее ячейка. Если курсор наведен на ячейку, то соответствующий документ отображается в *Parallel Coordinates View* в виде толстой черной линии (рис. 13C). В режиме *X-ray* под таблицей документов находится цветной спектр: каждый цвет спектра соответствует кластеру, так что принадлежность выбранного документа тому или иному кластеру может быть определена визуально по доле цветной области в спектре.

Модуль *Term-Weight View* отображает распределение $p(w \mid t)$ всех терминов в теме (см. рис. 12E). Пользователь может изменять значения $p(w \mid t)$, после чего новая тематическая модель будет построена с учетом этих изменений. Например, если для термина w указанное значение увеличилось, то система начнет чаще относить документы, содержащие w, к теме t. Документы, которые в результате переместились из одного кластера в другой, отображаются в *Document Tracer View*.

Модуль *Document Tracer View* отображает тепловую карту, показывающую перемещения документов между кластерами в процессе взаимодействия пользователя с системой (см. рис. 12F). Тепловая карта имеет размер $T \times T$, где T — количество кластеров. Каждый ее элемент (i, j) отображает, как много документов перешли из кластера i в кластер j. При нажатии на элемент карты открываются соответствующие этому элементу документы в *Document View*.

Модуль *Document View* показывает отдельный документ (см. рис. 12G). При просмотре документа термины окрашиваются в цвета своих тем. Это представление доступно из любого модуля визуализации, кроме *Term-Weight View*.

Модуль *Cluster Tree View*. Одной из целей создания iVisClustering была возможность улучшения тематической модели. В этом представлении пользователь в процессе взаимодействия с системой может изменять иерархическую структуру модели. Для этого используются пять операций: разъединение и соединение кластеров, перемещение и удаление кластера, повторная кластеризация. Результаты этих действий немедленно отображаются в дереве иерархии (см. рис. 12B). Документы удаленного кластера сохраняются, так как они могут понадобиться при отыскании новых тем в коллекции, и их можно включить обратно в коллекцию в любой момент.

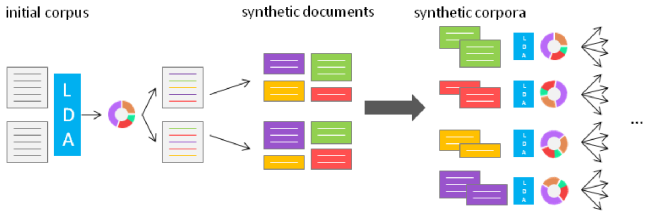


Рис. 14 Система Hiérarchie. Схема алгоритма HLDA (с разрешения авторов [16])

Повторная кластеризация используется при изменении числа кластеров. После нее с помощью *венгерского алгоритма* [15] находятся наилучшие парные соответствия между исходными и новыми кластерами. Цвета новых кластеров изменяются так, чтобы схожие кластеры до и после изменения имели одинаковый цвет.

Достоинства: интерактивное взаимодействие с пользователем с целью улучшения визуального представления и улучшения тематической модели; наличие нескольких различных представлений для отображения коллекции; возможность автоматического и ручного именования тем.

Недостатки: нет автоматического построения иерархической структуры (пользователь должен сам разбивать темы на подтемы); нет возможности тематического поиска.

Пример визуализации:
`ftp://temp:temp@mimi.cc.gt.atl.ga.us/resource/2012_eurovis_ivisclustering.mp4`

7 Система Hiérarchie

Система Hiérarchie [16] — это веб-приложение для построения иерархических тематических моделей и их визуализации в виде дерева.

Для построения тематической модели в Hiérarchie используется иерархический алгоритм HLDA (*Hierarchical Latent Dirichlet Allocation*), который рекурсивно разделяет темы на подтемы. Используя распределения $p(t \mid d)$, HLDA разделяет каждый документ d на *искусственные поддокументы* по каждой теме t. Поддокументы содержат только те слова, которые относятся к данной теме. Таким образом, для каждой темы t генерируется новая *искусственная подколлекция*, и для нее строится тематическая модель следующего уровня, разделяющая данную тему на подтемы (рис. 14). Процесс продолжается до тех пор, пока подколлекции не станут слишком маленькими для моделирования. Пользователь может задавать критерии остановки рекурсивного процесса.

В качестве основной визуализации в Hiérarchie используется круговая диаграмма с исходящими из нее лучами (*sunburst chart*). Она позволяет отображать как маленькие, так и большие иерархии без интерактивной прокрутки, приспосабливаясь к размеру экрана без искажения структуры иерархии.

Рисунок 15 показывает верхний уровень для коллекции твитов и новостей, касающихся пропавшего самолета Малазийских авиалиний. Каждый уровень иерархии изображается в виде кольца, разбитого на дуги, изображающие темы. Чтобы избежать загромождения диаграммы, темы не подписываются, вместо этого их названия изображаются сверху при наведении на них курсора. Таким образом, Hiérarchie придерживается принципа «сначала общий вид, затем масштабирование и фильтрация, детали по запросу». При наведении мыши на тему в середине схемы отображаются слова темы, что экономит пространство экрана и требует минимальных перемещений взгляда.

Когда пользователь выбирает тему, диаграмма перестраивается для отображения только выбранной темы и ее подтем (см. рис. 15). Сверху строится путь от корня всей иерар-

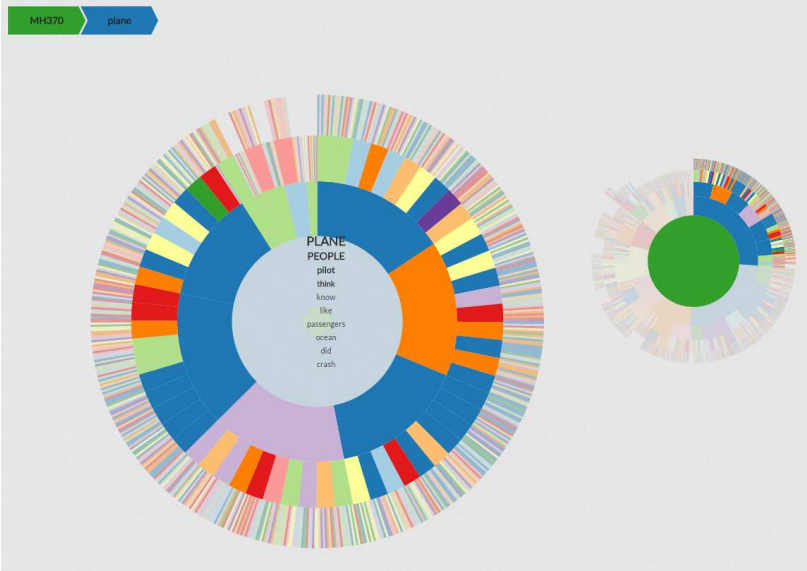


Рис. 15 Система Hiérarchie. Визуализация иерархии с помощью круговой диаграммы (с разрешения авторов [16])

хии до корня текущего уровня. Справа всегда находится «якорь», который отображает всю иерархию с выбранными темами и подтемами. Таким образом, пользователь всегда видит, на каком уровне иерархии он находится.

Рассмотрим взаимодействие пользователя с системой на примере визуализации твитов и новостей о пропавшем малазийском боинге MH-370 (см. рис. 15). Целью является анализ теорий, согласно которым пропал самолет. Была построена модель из 10 тем для каждого уровня. При выборе темы «*plane, people, pilot, think, know*», которая освещает различные теории, пользователь переходит на следующий уровень иерархии. Пользуясь навигацией по уровням, он находит наиболее обсуждаемые теории: самолет приземлился, самолет разбился, самолет был захвачен террористами, пилот разбил самолет в попытке суицида. Если выбрать тему о крушении, то на этом уровне также есть подтемы: самолет разбился в океане, на суше, из-за технических неполадок.

Достоинства: хорошая интерпретируемость, интуитивно понятный интерфейс; высокая детализация тем и подтем; нет ограничения для числа уровней иерархии; удобная навигация по иерархии.

Недостатки: невозможность задать число тем для каждого уровня; число терминов темы также фиксировано и не может быть изменено пользователем; просмотр документов пока невозможен; медленная реакция системы на действия пользователя.

Ссылки:
`https://github.com/mlvl/Hierarchie` — сайт системы *Hiérarchie*;
`http://mlvl.github.io/Hierarchie` — пример визуализации.

8 Система TextFlow

Система TextFlow [17] предназначена для визуализации *динамических* (temporal) тематических моделей таких коллекций, в которых каждый документ имеет отметку времени создания или публикации. TextFlow позволяет находить и анализировать переломные события в темах — их появление и исчезновение, разделение и слияние.

Система состоит из трех основных компонент (рис. 16, слева). *Препроцессор* извлекает основной текст документов. *Тематический анализатор* строит динамическую тематиче-

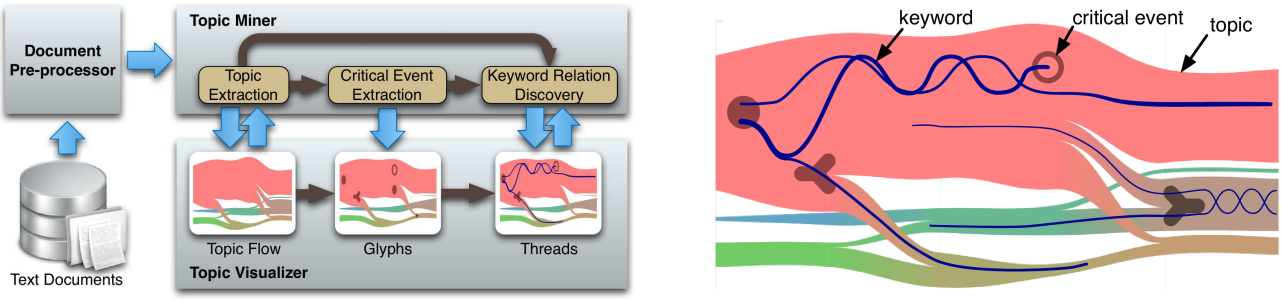


Рис. 16 Система TextFlow. Архитектура системы (слева) и пример визуализации (справа): 4 потока тем, 4 переломных изменения и 5 нитей ключевых слов (с разрешения авторов [17])

скую модель, выявляет моменты слияния и разделения тем, а также переломные события и взаимосвязи между терминами тем. *Тематический визуализатор* отображает результаты в трех компонентах визуализации.

Изменение тем как поток. В TextFlow изменения тем с течением времени изображаются в виде графа потоков (рис. 16, справа). Время откладывается вдоль горизонтальной оси. Изменяющаяся высота потока вдоль вертикальной оси пропорциональна количеству документов, которые относятся к данной теме в данный момент времени. Потоки могут разделяться и сливаться. При разделении (слиянии) главной веткой считается та, тематика которой наиболее близка к тематике потока до разделения (после слияния). При соединении потоков цвет получившейся ветви получается смешением цветов соответствующих потоков. Пропорции смешивания определяются высотами исходных веток. Аналогичный механизм применяется при разделении потоков.

Переломные изменения как глифы. Для отображения появления, исчезновения, соединения и разделения потоков были выбраны четыре глифа (рис. 17, слева). Они накладываются поверх изображения потоков в моменты событий, обнаруженных тематическим анализатором. Чем больше глиф, тем важнее обозначаемое им событие.

Корреляции ключевых слов как нити. Для изображения взаимосвязи определенного термина (ключевого слова) темы с другими ее терминами используется визуальный примитив, называемый *нитью* (см. рис. 16). Моменты времени, когда появляется ключевое слово и когда оно исчезает, соединяются кривой линией с волновым эффектом (см. рис. 17, справа). Если несколько нитей взаимодействуют друг с другом, то для изображения совместного появления этих ключевых слов используется волновой пучок в том интервале времени, когда ключевые слова часто совместно появлялись в документах. Амплитуда пучка пропорциональна частоте появления всех ключевых слов темы: чем она больше, тем чаще появляются соответствующие слова в данный момент времени.

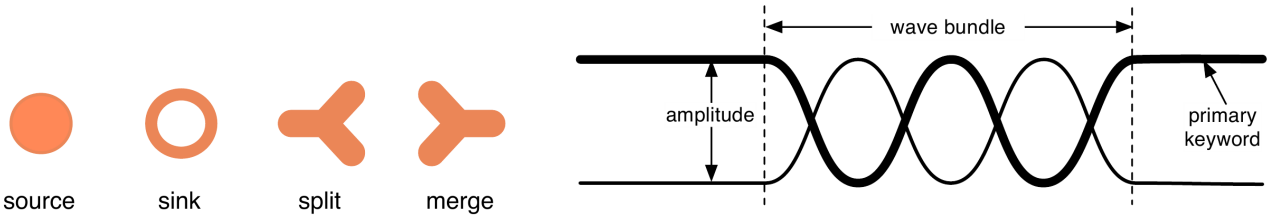


Рис. 17 Система TextFlow. Глифы (слева) отображают переломные моменты: исток, сток, разделение, слияние. Визуальные атрибуты (справа) отображают взаимодействия нитей: амплитуда и волновой пучок (с разрешения авторов [17])

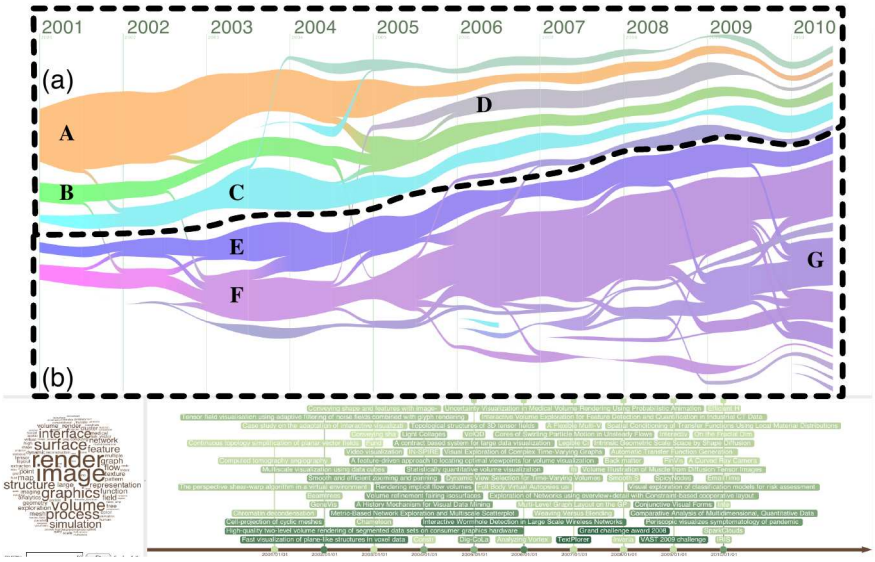


Рис. 18 Система TextFlow на примере визуализации коллекции статей научных конференций Vis и InfoVis. Основной графический интерфейс TextFlow и дополнительные компоненты: облако тегов (слева внизу) и временная шкала (справа внизу) (с разрешения авторов [17])

Дополнительные средства: облако тегов и временная шкала. Описанные выше три компонента отображают только динамику изменений тем, но скрывают детальную текстовую информацию. Поэтому в TextFlow были добавлены два дополнительных компонента (рис. 18, внизу): облако тегов и временная шкала, которые при выделении темы отображают наиболее значимые термины и предложения этой темы.

В системе TextFlow есть две основные функции пользовательского интерфейса: наведение курсора на элемент и его выбор. При наведении пользователь видит агрегированную информацию, что облегчает ему выбор элемента. Например, при наведении на тему будут отображаться ее наиболее существенные термины и ветки, исходящие из нее. При выборе пользователь видит более детальную информацию об элементе (например, облако тегов для темы). Последнее нужно для того, чтобы система могла рекомендовать пользователю шаги дальнейшей навигации по коллекции (например, схожие по составу темы или слова, которые часто появляются вместе с выбранным ключевым словом).

Рассмотрим взаимодействие пользователя с системой на примере визуализации коллекции статей научных конференций IEEE Visualization (Vis) и IEEE Information Visualization (InfoVis) с 2001 по 2010 гг. На рис. 18 показан результат визуализации, где сразу видно несколько особенностей. Темы A–D, относящиеся к конференции Vis, сливались и разделялись до 2006 г., но затем развивались независимо друг от друга. Темы конференции InfoVis E–G, наоборот, активно взаимодействовали друг с другом на всем интервале времени. При этом тема F «исследование/аналитика» является «главным» потоком, порождающим большое количество ответвлений. При детальном анализе темы F (рис. 19) отчетливо видно, что в середине потока есть исток (a). При анализе относящихся к нему документов выяснилось, что большинство из статей относится к теме «аналитика». Это не случайное совпадение, так как в 2006 г. впервые был проведен симпозиум IEEE VAST. Таким образом, при детальном или общем анализе визуализации пользователь может найти интересные особенности в коллекции.

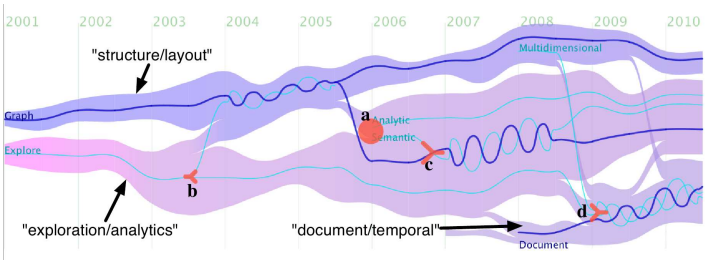


Рис. 19 Система TextFlow на примере визуализации коллекции статей научных конференций Vis и InfoVis. Детализация темы F конференции InfoVis (с разрешения авторов [17])

Достоинства: богатый и удобный интерфейс для исследования тем во времени; визуализация переломных изменений и ключевых слов.

Недостатки: непонятно, как выбирать главное ключевое слово для отображения нитей; во временной шкале отображаются только отрывки документов, причем не всех; неудобно сравнивать темы в интерфейсе; работает на коллекциям малого объема.

Пример визуализации:
`http://cgcad.thss.tsinghua.edu.cn/shixia/publications/textflow/video.avi`

9 Система HierarchicalTopics

Система HierarchicalTopics [18] совмещает в себе построение и интерактивную визуализацию иерархических тематических моделей в их динамическом развитии (рис. 20).

Система реализует четыре стадии обработки исходных данных (рис. 21): (A) накопление данных; (B) предварительная обработка, параллельное тематическое моделирование и суммаризация текстовых документов; (C) построение иерархического дерева путем слияния тематических кластеров; (D) визуализация. Первые две стадии реализуются в режиме оффлайн, последние две — в режиме интерактивного взаимодействия пользователя с системой.

Визуализация HierarchicalTopics состоит из двух синхронизированных представлений: отображение иерархии *(Hierarchical Topic view)* и отображение потоков тем *(Hierarchical ThemeRiver)*.

Hierarchical Topic view. В этом представлении пользователь может активно взаимодействовать с системой (рис. 22). Помимо стандартного масштабирования и прокрутки, он

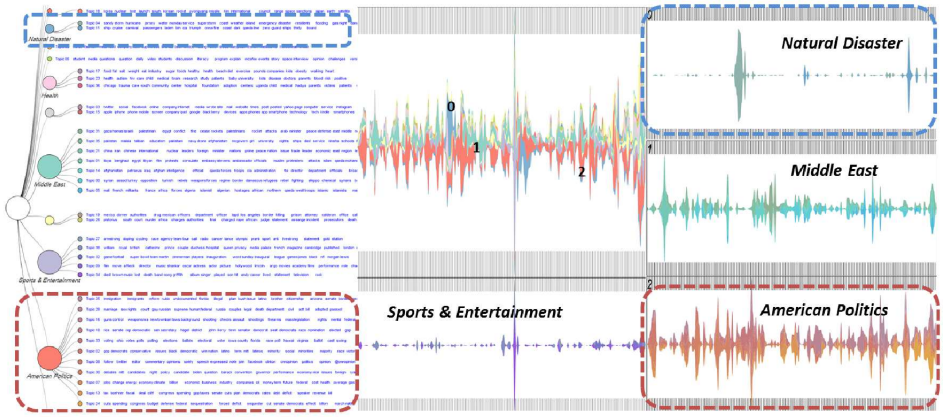


Рис. 20 Система HierarchicalTopics. Основные элементы пользовательского интерфейса (с разрешения авторов [18])

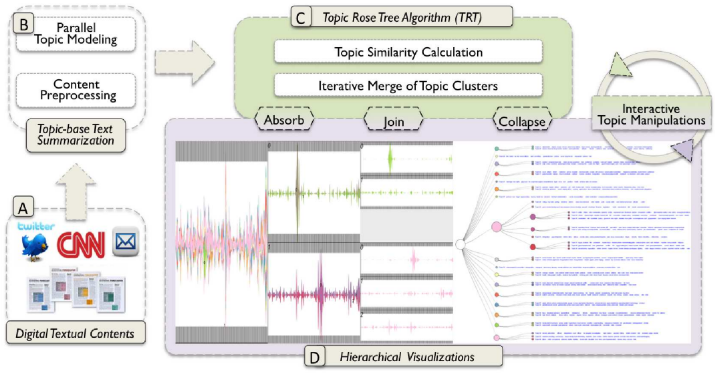


Рис. 21 Система HierarchicalTopics. Архитектура (с разрешения авторов [18])

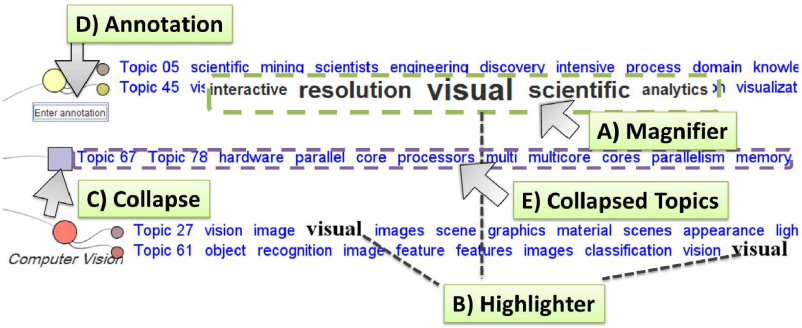


Рис. 22 Система *HierarchicalTopics*. Основные действия пользователь может совершать в Hierarchical Topic view: (A) лупа; (B) выделитель; (C) сворачивание вершин (при этом форма вершины становится прямоугольником); (D) аннотация, которую пользователь может добавить к любой вершине (с разрешения авторов [18])

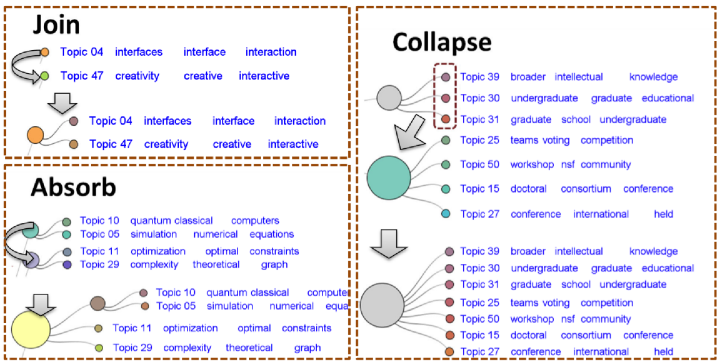


Рис. 23 Система HierarchicalTopics. Три операции, с помощью которых пользователь может изменить иерархию тем: соединение (join), поглощение (absorb) и свертывание (collapse) (с разрешения авторов [18])

может использовать лупу (для увеличения размера шрифта слов темы) и выделитель (для выделения слов темы). Также пользователь может изменять структуру иерархии: группировать несколько вершин дерева в одну, поглощать вершины и сворачивать их (рис. 23). Имеется возможность подписать каждую вершину группы тем.

Hierarchical ThemeRiver. Для отображения динамических изменений в группах тем используется модуль визуализации *ThemeRiver* [19]. Работа пользователя начинается с глав-

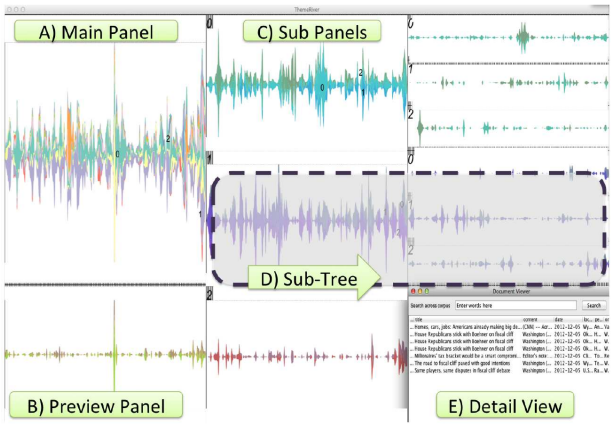


Рис. 24 Система HierarchicalTopics. Основной интерфейс *Hierarchical ThemeRiver* (с разрешения авторов [18])

ной панели, где отображаются изменения самых верхних тем иерархии (рис. 24A). Высота каждой ленты (т. е. части графика, соответствующей конкретной теме) вычисляется как сумма высот листьев соответствующей вершины. При наведении курсора на ленту на панели предварительного просмотра отображаются изменения, произошедшие в дочерних узлах (рис. 24B). Для сравнения нескольких групп тем в HierarchicalTopics используется гибкая структура панелей. Чтобы сравнить различные группы, пользователь может выбрать нужную ему ленту на графике, для которой построится подпанель (рис. 24C), показывающая следующий уровень иерархии для выбранной темы.

Цветовая схема. Для плавного перехода между панелями используются 12 специально подобранных когерентных цветов, которые присваиваются лентам на графике. Ленты дочерних узлов окрашиваются в тот же оттенок, но другой яркости и насыщенности.

Детализация текстовых документов возможна после выделения темы. В представлении *Hierarchical ThemeRiver* пользователь может включить режим «временно́й поддержки» и просматривать множества документов, которые были опубликованы в какой-либо конкретный период времени (рис. 24E).

Достоинства: соединение в одном приложении визуализации как иерархической, так и динамической модели; интуитивно понятный интерфейс; возможность интерактивного просмотра и изменения иерархии тем; возможность детализации текстов документов за любой период времени.

Недостатки: определение высоты ленты на временном графике через сумму высот листьев не является полезной характеристикой для исследования коллекции; при большом количестве тем график перестает быть визуально понятным.

Ссылки:

`http://youtu.be/Vi1FP5kAbOU` — пример визуализации.

10 Система RoseRiver

Система RoseRiver [20] разработана на основе системы *TextFlow* и предназначена для анализа динамики изменений в иерархических тематических моделях. Она позволяет сливать и разделять темы, выбирать уровень детализации иерархии путем задания разреза дерева и прослеживать изменения выбранного множества тем во времени.

Сначала по коллекции документов строится последовательность так называемых эволюционных деревьев тем, которые представляют иерархию тем в коллекции в различ-

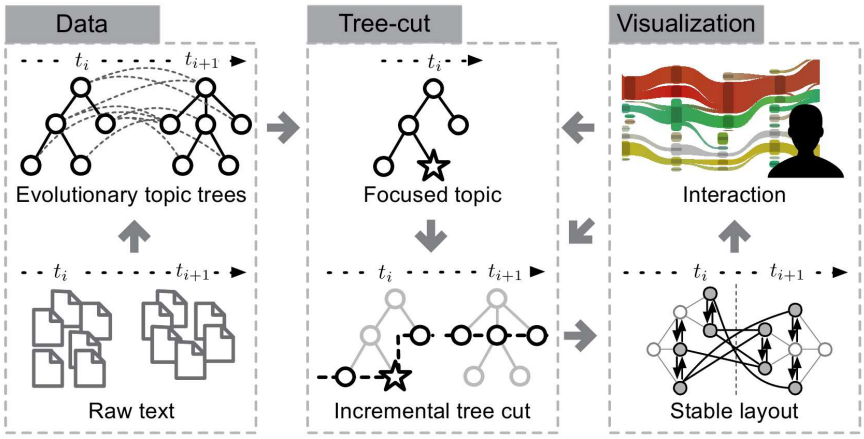


Рис. 25 Система RoseRiver состоит из трех компонентов: модуль обработки данных и тематического моделирования, модуль построения разрезов и модуль визуализации (с разрешения авторов [20])

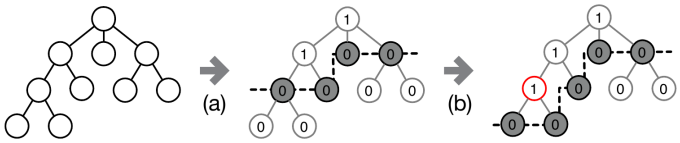


Рис. 26 Система RoseRiver. Создание разреза и его модификация: (a) выделены вершины разреза; (b) тема перестала быть фокусом, сгенерирован новый разрез (с разрешения авторов [20])

ные моменты времени (рис. 25). Кроме последовательности деревьев на этом шаге также формируется множество пар схожих вершин-тем в деревьях, соответствующих смежным моментам времени. Система учитывает интересы пользователя, накапливая информацию о том, за изменением каких тем он следит. Для этого применяется техника построения деревьев с учетом степени интереса (*degree-of-interest, DOI*) [21].

В основе визуализации RoseRiver лежит *инкрементный алгоритм построения разреза в эволюционном дереве (incremental evolutionary tree cut algorithm). Разрезом дерева (tree cut)* называется такое множество вершин, что любой путь из корня дерева к его листу содержит только одну вершину из разреза (рис. 26). Идея алгоритма заключается в том, чтобы разрезы деревьев в последовательные моменты времени состояли из схожих вершин. Для этого пользователь фиксирует момент времени и выбирает одну или более основных вершин, называемых фокусными темами. На основе фокусных тем строится разрез дерева, называемый ключевым. По ключевому разрезу строится производное множество разрезов деревьев в смежные моменты времени, проходящих через вершины-темы, схожие с темами ключевого разреза. Пользователь может исправить автоматически найденные разрезы в интерактивном режиме, чтобы улучшить интерпретируемость тем.

Визуализация потока тем в RoseRiver полностью основана на TextFlow (см. выше), поэтому далее будут описаны только дополнительные элементы, появившиеся в RoseRiver.

Вершины разреза. Каждая вершина из разреза представляется прямоугольником со скругленными углами, при этом уровни иерархии представляются небольшими сдвигами вправо вдоль оси времени (рис. 27).

Документы. Цветная полоска между двумя или более потоками тем, разделенными вершиной из разреза, обозначает число пар документов, относящихся к этим темам

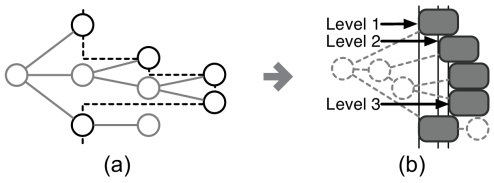


Рис. 27 Система RoseRiver. Вершины разреза (a) и их представление с небольшими сдвигами уровней вдоль оси времени (b) (с разрешения авторов [20])

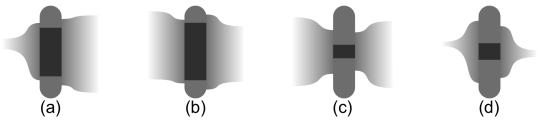


Рис. 28 Система RoseRiver. Четыре примера интерпретации потоков и полоски между ними: (a) возникновение темы; (b) тема мало изменяется; (c) тема сильно изменяется; (d) тема быстро возникает и исчезает (с разрешения авторов [20])

(рис. 28). Темная часть этой полоски обозначает документы, которые принадлежат обеим темам слева и справа от полоски (т. е. двум соседним деревьям). Высота темной части пропорциональна доле таких документов.

Цветовая схема. Фокусные темы имеют полностью насыщенный уникальный цвет (рис. 29, 30), в то время как производные темы (например, полученные в результате слияния) отличаются оттенками, т. е. сохраняется цветовая стратегия *TextFlow*. При этом цвет постепенно сводится к серому, если тема перестает быть похожей на темы-фокусы и уже не удовлетворяет интересам пользователя.

Детали. При наведении курсора на вершину-тему из разреза полоска внутри нее расширяется, показывая характерные для темы слова внутри расширенной области (рис. 31). Чтобы сохранить информацию о глубине такой вершины, расширяется только средняя часть полоски, концы при этом остаются неизменными.

Взаимодействие с системой. Пользователь может изменять потоки тем, сливая или разделяя темы. Для этого при выборе полоски вершины-темы она автоматически распадается на «подполоски», которые обозначают темы, содержащиеся в потоке слева или справа. При выборе нескольких таких «подполосок» можно слить воедино или разъединить несколько потоков тем (рис. 32).

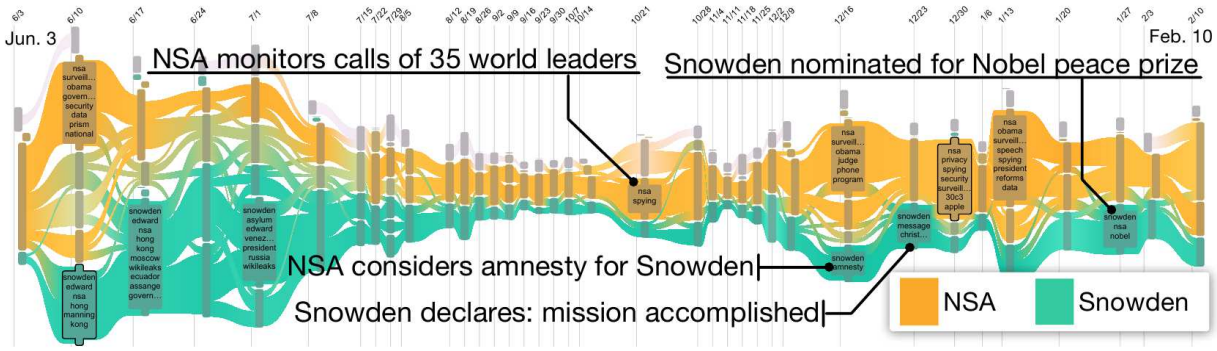


Рис. 29 Система RoseRiver. Коллекция новостей и твитов о PRISM с июля 2013 г. по февраль 2014 г. (с разрешения авторов [20])

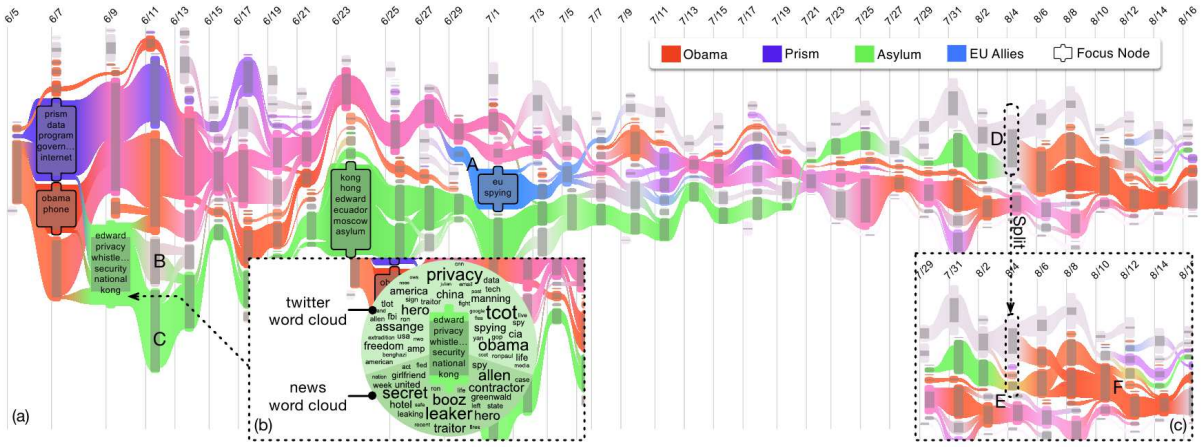


Рис. 30 Система RoseRiver. (a) Новости о PRISM с июня по август 2013 г.; (b) сравнение ключевых слов из Твиттера и новостей, длина дуг в облаке соответствует числу сообщений и статей; (c) после разделения двух потоков сгенерировано новое представление (с разрешения авторов [20])

Рис. 31 Система RoseRiver. Расширение полоски темы с сохранением ее положения (с разрешения авторов [20])

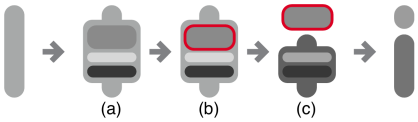


Рис. 32 Система RoseRiver. Пример разделения потоков тем (с разрешения авторов [20])

Также пользователь может изменять фокусные темы в процессе работы. Это возможно с помощью простого выделения интересующей темы или с помощью поиска, если пользователь не видит нужную ему тему на экране.

Рассмотрим работу пользователя с системой на примере визуализации коллекции новостей и твитов о PRISM (см. рис. 29). Для более детального рассмотрения были выбраны тема «*NSA*» (оранжевая) и тема «*Snowden*» (голубая). Как видно из рис. 29, голубая тема была намного популярнее в течение первого месяца, чем оранжевая тема, но с течением времени она стремительно теряла популярность, в то время как оранжевая тема почти не изменялась. Из этого можно сделать вывод, что оранжевая тема является перманентной, а голубая — событийной. То же самое можно сказать и о синей теме «*EU Allies*» (см. рис. 30), которая появилась на короткое время и исчезла. Для выявления причин такого изменения темы пользователь может разделить ее на подтемы и проанализировать их. Например, на рис. 30D видно, что причиной угасания зеленой темы стало появление желтой темы.

Достоинства: комбинированная визуализация иерархической и динамической модели при сохранении всех преимуществ TextFlow; возможность интерактивного взаимодействия с системой для изменения отображаемой иерархии; облако тегов отображается не внизу интерфейса, а прямо рядом с контур-вершиной.

Недостатки: сложный интерфейс для неподготовленного пользователя; непонятно, как выбирать фокусные темы, когда пользователь заранее не знает, о чем коллекция; возможны сбои при построении эволюционных деревьев и контуров в них; нет навигации по конкретным деревьям в конкретный момент времени.

Ссылки:
`http://research.microsoft.com/en-us/um/people/weiweicu/RoseRiver` — сайт проекта;
`http://cgcad.thss.tsinghua.edu.cn/shixia/publications/RoseRiver/video.mp4` — пример визуализации.

11 Краткий обзор других систем визуализации

11.1 Система MetaToMATo

Система MetaToMATo (Metadata and Topic Model Analysis Toolkit) [22] предназначена для визуализации не только тематической модели, но и метаданных коллекции (рис. 33). MetaToMATo позволяет пользователю переименовать темы, удалять темы, добавлять различные типы метаданных — авторов, метку времени, географическую метку, метку источника и т. д. Также с помощью MetaToMATo можно осуществлять тематический поиск по метаданным.

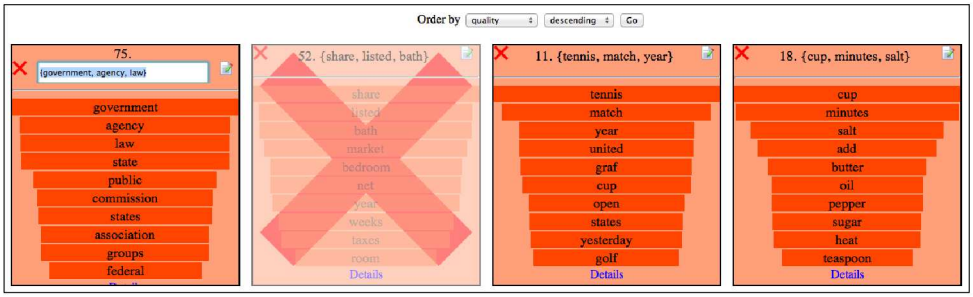


Рис. 33 Система MetaToMATo. Страницы темы (с разрешения авторов [22])

Достоинства: удобный и функциональный интерфейс для отображения метаданных и контроля степени их влияния на отображаемые документы при поиске.

Недостатки: невозможность отображения схожих тем; невозможность улучшить модель (изменить слова темы, слить или разделить темы); список терминов темы состоит из самых вероятных слов, а не из наиболее характерных слов темы.

11.2 Система LDAvis

Система LDAvis показывает, насколько темы весомы в коллекции, насколько они близки друг к другу и из каких значимых терминов они состоят [23]. Для этого в LDAvis имеются два представления (рис. 34). Первое отображает темы в виде вершин, размер которых пропорционален $p(t)$ и расстояние между которыми пропорционально мере сходства тем. Второе показывает распределения $p(w \mid t)$ и $p(w)$ для релевантных терминов темы. Релевантность термина w относительно темы t определяется как

$$\text{relevance}(w, t \mid \lambda) = \lambda \log p(t \mid w) + (1 - \lambda) \log \frac{p(t \mid w)}{p(w)},$$

где параметр λ $(0 \leqslant \lambda \leqslant 1)$ регулируется пользователем (авторы рекомендуют $\lambda = 0{,}6$). Также LDAvis группирует схожие темы в кластеры и обозначает каждый кластер уникальным цветом (максимальное число кластеров равно 10).

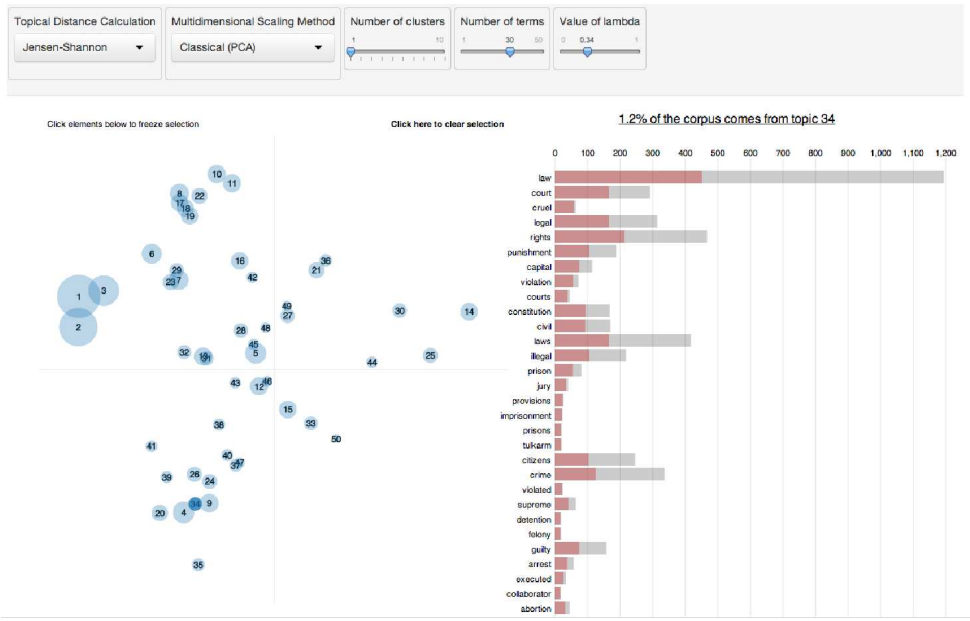


Рис. 34 Система LDAvis (с разрешения авторов [23])

Достоинства: удобный интерактивный интерфейс для исследования распределений слов по темам и по коллекции, где степень релевантности терминов контролируется пользователем.

Недостатки: невозможность просмотра документов и распределений тем по документам, ограниченное количество кластеров и отображаемых терминов, невозможность задать количество тем, отсутствие именования тем.

Сайт: `https://github.com/cpsievert/LDAvis`.

Пример: `http://cpsievert.github.io/LDAvis/reviews/vis`.

11.3 Система TIARA

Система TIARA (Text Insight via Automated Responsive Analytics) предназначена для визуализации динамических моделей. Она представляет темы как потоки в потоковом графе [24]. На каждом потоке в каждый момент времени изображается облако слов темы, популярных в данный момент (рис. 35). Высота потока пропорциональна количеству

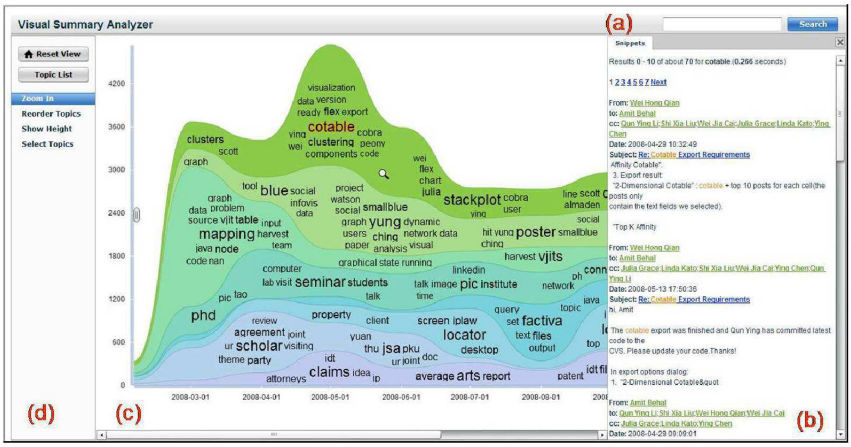


Рис. 35 Система TIARA (с разрешения авторов [24])

документов, относящихся к теме. В TIARA пользователь может улучшать тематическую модель, соединяя/разъединяя потоки, а также осуществлять тематический поиск по коллекции. Продолжением этой системы стал TextFlow.

11.4 Система SolarMap

Система SolarMap предназначена для визуализации динамических моделей (рис. 36) [25]. В SolarMap документ разбивается на аспекты (*facets*). Аспектами могут быть даты или структурные части документов. Например, если коллекция — это статьи с описанием различных заболеваний, то аспектами могут быть разделы «симптомы», «лечение», «профилактика». Кроме того, из текста извлекаются именованные сущности (*named entity*). Аспекты изображаются в виде тонких колец, внутри них группируются сущности в виде облака, вокруг которых изображаются ключевые слова в виде толстого кольца, разбитого на части, относящиеся к группам сущностей. В качестве аспектов можно задавать даты и, меняя тонкие кольца, следить за развитием тем в коллекции.

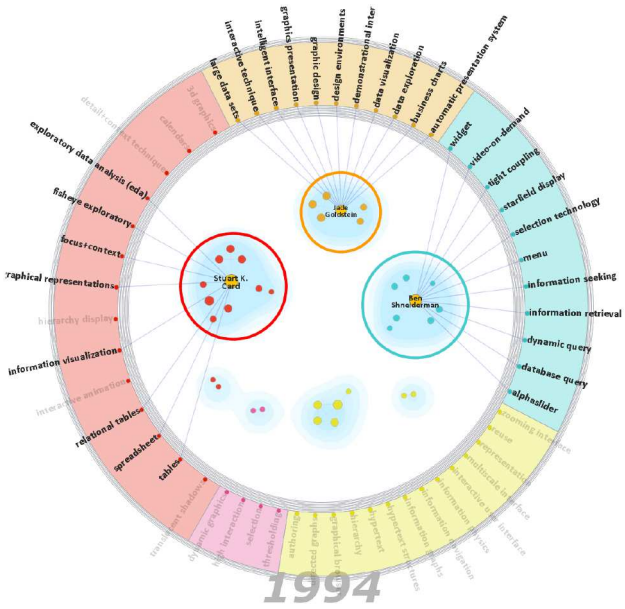


Рис. 36 Система SolarMap (с разрешения авторов [25])

Достоинства: визуализация различных метаданных, включая метки времени; возможность отслеживания конкретной группы сущностей с течением времени.

Недостатки: отсутствие просмотра документов; отсутствие именования тем; невозможность улучшения модели.

Пример: `http://nancao.org/projects/solarmap.tml`.

11.5 Система TextWheel

Система TextWheel предназначена для визуализации новостных потоков [26]. Она состоит из трех компонентов: «конвейера» документов, «колеса» ключевых слов и временной шкалы (рис. 37). Временная шкала отображает значимость документов с течением времени. U-образный конвейер содержит глифы, обозначающие документы, которые соединяется с колесами ключевых слов, при этом слова соединяются с документами через два узла, обозначающих положительное и отрицательное отношение.

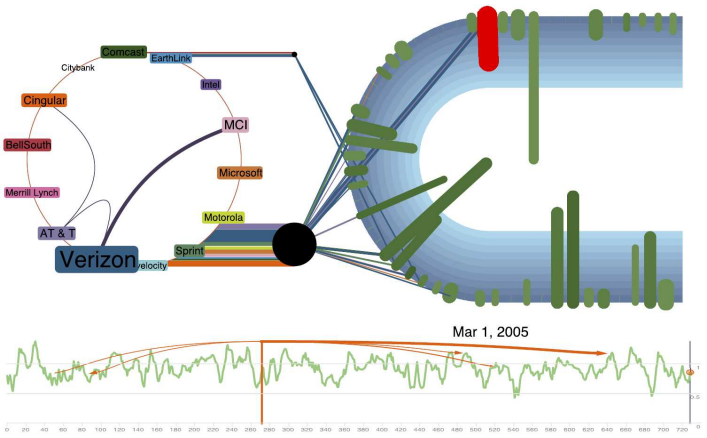


Рис. 37 Система TextWheel (с разрешения авторов [26])

Достоинства: пользователь может контролировать скорость конвейера, соединять глифы, детализировать визуализацию и осуществлять повторную кластеризацию.

Недостатки: нет отображения тем; нет поиска документов; отсутствуют средства визуализации всей коллекции.

11.6 Система ThemeDelta

Система ThemeDelta предназначена для визуализации динамических моделей [27]. ThemeDelta извлекает из коллекции так называемые *тренды* (ключевые слова) и группирует их в темы в каждый момент времени (рис. 38). Тренды изображаются в виде волнистых линий, цвет которых зависит от темы, толщина — от значимости слова в теме. Моменты времени изображаются в виде вертикальных осей, на которых расположены группы трендов (темы). Пользователь может осуществлять поиск трендов, сортировать и фильтровать их, детализировать визуализацию нескольких трендов и трендов, тесно связанных с ними.

Достоинства: возможность отследить изменения темы с течением времени.

Недостатки: избыточная детальность; отсутствие визуализации распределений слов по темам; отсутствие именования тем; невозможность улучшения тематической модели.

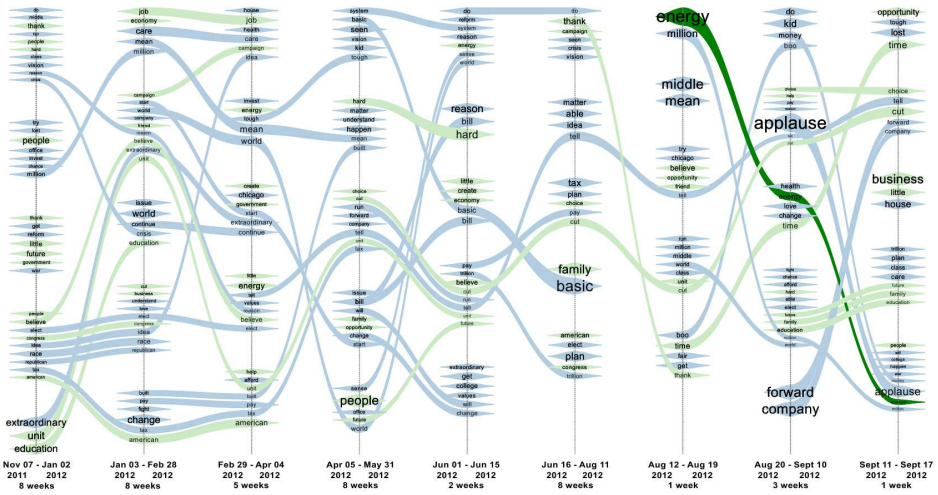


Рис. 38 Система ThemeDelta (с разрешения авторов [27])

11.7 Система D-VITA

Система D-VITA предназначена для интерактивной визуализации динамических моделей [28]. Пользовательский интерфейс системы состоит из следующих компонентов (рис. 39): потоковый граф для отображения тем (толщина потока пропорциональна релевантности темы); потоковый граф для отображения слов темы; круговая диаграмма для отображения пропорций тем в документе; модуль просмотра документов с возможностью поиска схожих по тематике документов.

Достоинства: удобный интерактивный интерфейс для навигации по коллекции; возможность осуществлять тематические запросы.

Недостатки: отсутствие именования тем; отсутствие отображения схожих тем; невозможность улучшения тематической модели, в частности отсутствие настройки степени сглаживания тем во времени.

Сайт: `https://github.com/rwth-acis/D-VITA`.

Пример: `http://monet.informatik.rwth-aachen.de/DVita`.

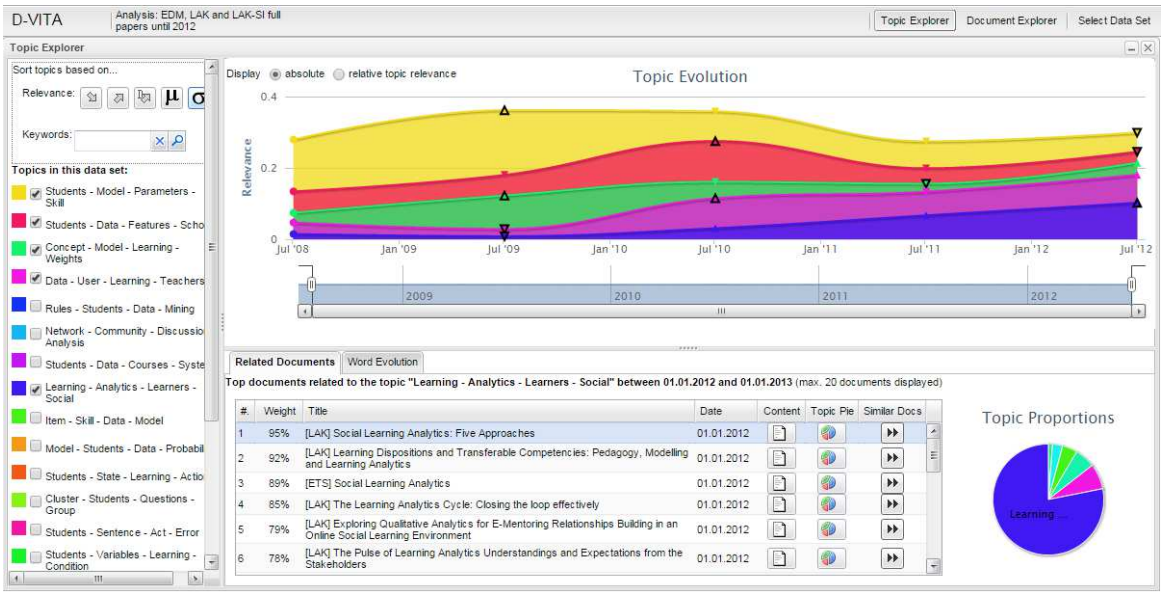


Рис. 39 Система D-VITA (с разрешения авторов [28])

11.8 Система TopicPanorama

Система TopicPanorama предназначена для визуализации иерархических моделей [29]. Пользовательский интерфейс состоит из трех компонентов: основной панели для визуализации, информационной панели и панели управления. TopicPanorama представляет иерархическую структуру тем в виде дерева (рис. 40), при этом корреляции между темами отображаются с помощью областей разной цветовой насыщенности. Информационная панель показывает дополнительную информацию о темах и о документах, относящихся к теме. Панель управления позволяет пользователю задавать параметры визуализации, осуществлять поиск и изменять иерархическую структуру модели.

Достоинства: возможность отображения иерархической структуры в двух представлениях — в виде графа и в виде столбчатой диаграммы, окружающей граф.

Недостатки: отсутствие отображения распределений тем по документам; невозможность изменить иерархическую структуру и вручную именовать темы.

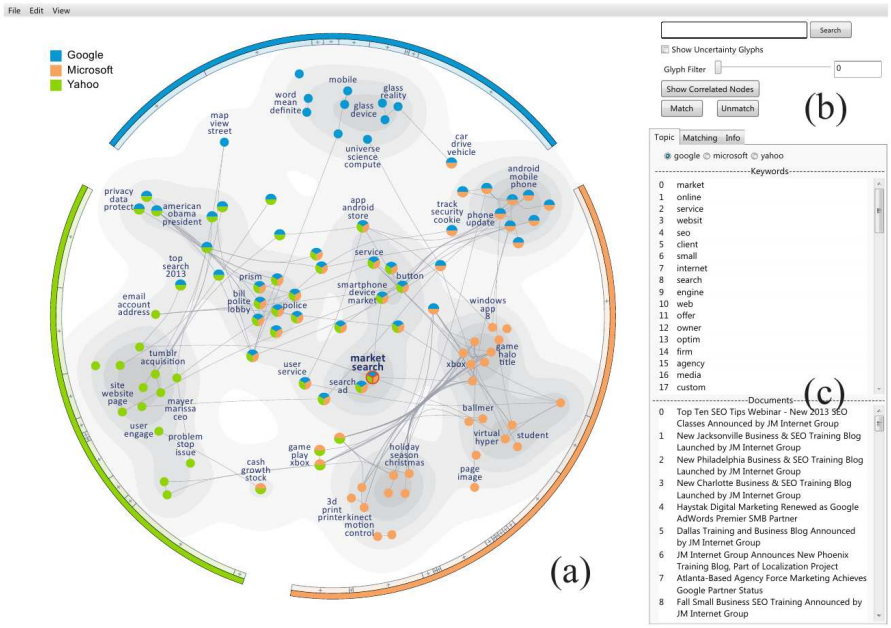


Рис. 40 Система TopicPanorama (с разрешения авторов [29])

Сайт: `http://cgcad.thss.tsinghua.edu.cn/shixia/publications/TopicPanorama/index.htm`.
Пример: `http://cgcad.thss.tsinghua.edu.cn/shixia/publications/TopicPanorama/video_eng.mp4`.

12 Графические библиотеки

Для визуализации тематических моделей в веб-интерфейсах используются также графические библиотеки общего назначения, наиболее известные — Gephi и D3.js.

12.1 Система Gephi

Система Gephi — это платформа с открытым кодом для анализа и визуализации больших сетей. Ее применение в тематическом моделировании основано на том, что разреженные матрицы распределений терминов в темах и тем в документах можно рассматривать как матрицы смежности для двудольных графов с вершинами двух типов — соответственно терминов и тем либо тем и документов.

Например, в [30] анализировалась коллекция текстов с обсуждениями медицинских препаратов для поддержания сна с сайта `www.patientslikeme.com`. Тематическая модель была построена с помощью пакета MALLET[3]. Исходные документы и полученные темы визуализировались с помощью системы Gephi. Документы отображаются точками, темы — своими названиями. Для автоматического размещения вершин графа в Gephi используются методы многомерного шкалирования (рис. 41).

На рис. 42 показаны два варианта интерактивной визуализации двудольного графа, в которых темы отображаются более крупными вершинами, термины — более мелкими.

Сайт: `http://gephi.github.io`.
Пример: `http://dig-eh.org/topic-modeling-and-gephi-a-work-in-progress`.

[3]MALLET (***MA****chine* ***L****earning for* ***L****anguag****E*** *Toolkit*) — пакет под Java для обработки естественного языка, классификации документов, тематического моделирования и других приложений машинного обучения к анализу текстов [31].

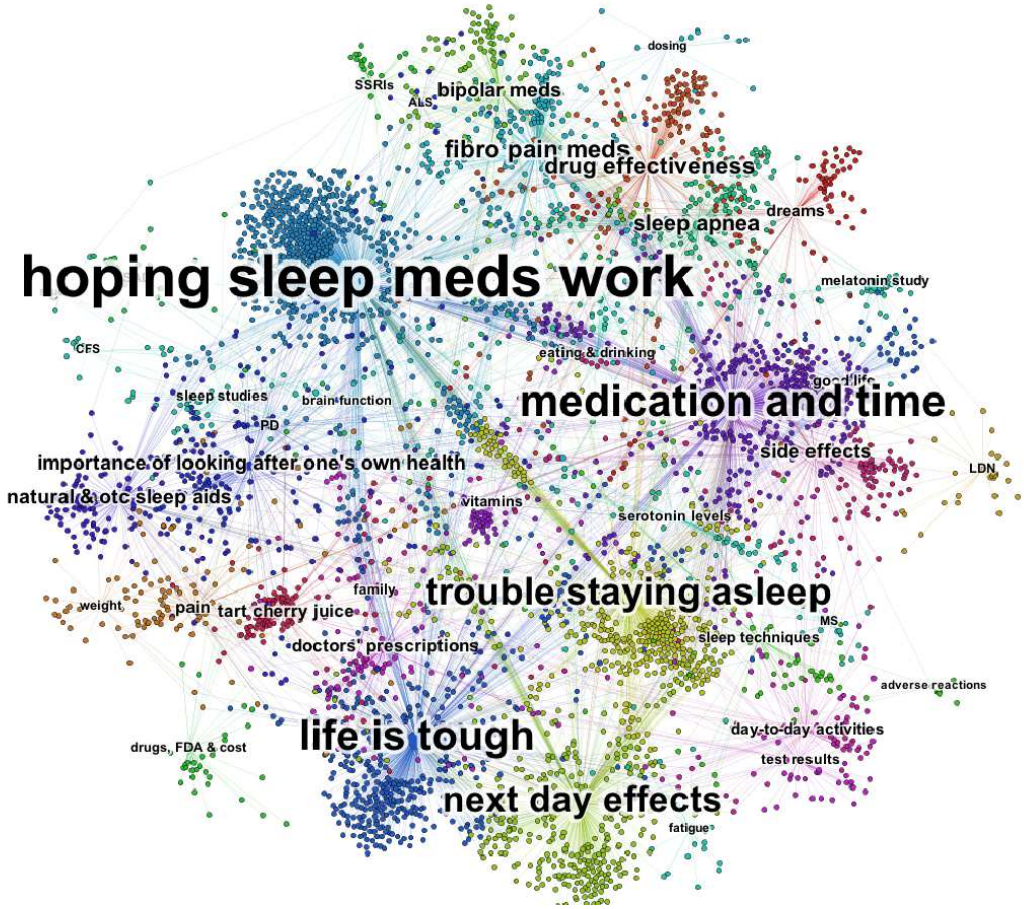


Рис. 41 Визуализация тематической модели LDA, построенной MALLET, с помощью Gephi (с разрешения авторов [30])

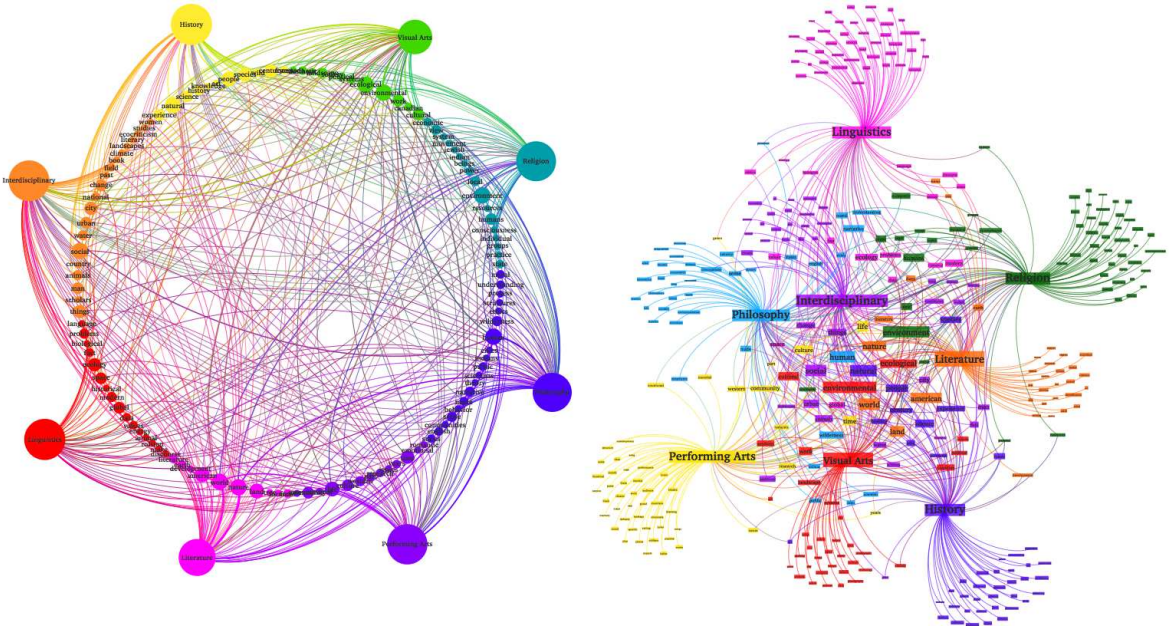


Рис. 42 Визуализация с помощью Gephi

12.2 Библиотека D3.js

Библиотека D3.js (Data-Driven Documents) — это JavaScript-библиотека с открытым кодом для создания и контроля динамических и интерактивных графических элементов, которые могут отображаться в веб-браузере.

D3.js предоставляет мощный интерфейс для построения интерактивных визуализаций на основе больших данных и имеет широкое применение в различных областях, в том числе в тематическом моделировании. Например, рассмотренные выше системы LDAvis и Hiérarchie используют D3.js для визуализации тематических моделей.

Сайт: `http://d3js.org`.

Пример: `https://github.com/mbostock/d3/wiki/Gallery#visual-index`.

13 Заключение

Важной тенденцией современных средств визуализации тематических моделей является использование веб-интерфейсов. Исследователи и разработчики стремятся возложить на мощные сервера всю подготовительную и вычислительную работу, связанную с накоплением коллекции, ее предварительной обработкой, построением модели и поисковых

Таблица 1 Средства визуализации тематических моделей

Требования	TMVE	Termite	TopicNets	Hierarchie	iVisClustering	TextFlow	HierarchicalTopics	RoseRiver	MetaToMATO	LDAvis	TIARA	SolarMap	TextWheel	ThemeDelta	D-VITA	TopicPanorama
Список тем	+	+	−	−	+	−	+	−	+	+	+	−	−	−	+	+
Отображение схожих тем	+	−	+	−	+	+	−	+	+	+	+	+	−	+	−	+
Именование тем:																
• автоматическое	+	+	+	+	+	+	+	+	+	−	−	−	−	−	+	+
• ручное	−	−	−	−	+	−	−	−	+	−	−	−	−	−	−	−
Отображение документов	+	+	+	−	+	+	+	+	+	−	+	−	+	?	+	+
• схожих документов	+	+	+	−	+	−	−	−	+	−	+	−	−	−	+	+
Распределения:																
• $p(t \mid d)$	+	−	+	−	+	?	−	?	+	−	?	−	−	−	+	−
• $p(w \mid t)$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
• $p(t)$	+	−	+	+	−	+	−	+	+	+	+	−	−	−	+	−
Модальности:																
• время	−	−	+	−	−	+	+	+	+	−	+	+	+	+	+	−
• иерархии	−	−	+	+	+	−	+	+	−	−	−	+	−	−	−	+
• авторы	−	−	+	−	−	−	−	−	+	−	−	+	+	−	−	−
• другие	−	−	+	−	−	−	−	−	+	−	−	+	+	−	−	−
Детализация	−	−	+	+	+	+	+	+	−	−	+	?	+	+	+	+
Поисковые запросы	+	−	+	−	−	−	−	+	+	−	+	−	−	+	+	+
Изменение модели:																
• повторное построение	−	−	−	−	+	+	−	+	−	−	+	−	+	−	−	+
• слияние или разделение тем	−	−	−	−	+	+	−	+	−	−	+	−	−	−	−	−
• удаление тем	−	−	−	−	+	+	−	+	+	−	−	−	−	−	−	−
• изменение веса терминов	−	−	−	−	+	−	−	−	−	−	−	−	−	−	−	−
Открытый код	+	+	−	+	−	−	−	−	−	+	−	−	−	−	+	+

индексов, генерацией визуальных представлений. Пользователям таких систем остается управлять параметрами визуального представления и средств навигации. Некоторые системы предполагают также обратную связь с пользователем, возможность внесения экспертных оценок или исправлений с целью улучшения модели.

В настоящее время не сложилось единого понимания, каким должен быть идеальный пользовательский интерфейс для тематического поиска и навигации по большим коллекциям текстовых документов. Отсюда большое разнообразие идей и систем для визуализации. Многие из них являются исследовательскими и далеки от стадии коммерческого использования. Некоторые из них имеют открытый исходный код.

Не претендуя на полноту, подытожим данный обзор табл. 1, в которой сделана попытка систематизации систем визуализации тематических моделей в разрезе основных функциональных требований.

Автор выражает признательность К. В. Воронцову за постановку задачи и внимание к работе.

Литература

[1] *Blei D.* 2012. Probabilistic topic models // Commun. ACM, 2012. Vol. 55. No. 4. P. 77–84.

[2] *Daud A., Li J., Zhou L., Muhammad F.* Knowledge discovery through directed probabilistic topic models: A survey // Frontiers of Computer Science in China, 2010. Vol. 4. No. 2. P. 280–301.

[3] *Chang J., Gerrish S., Wang C., Boyd-graber J. L., Blei D. M.* Reading tea leaves: How humans interpret topic models // Advances in neural information processing systems / Eds. Y. Bengio, D. Schuurmans, J. D. Lafferty, *et al.* — MIT Press, 2009. Vol. 22. P. 288–296.

[4] *Chaney A., Blei D.* Visualizing topic models // Frontiers of Computer Science in China, 2012. Vol. 55. No. 4. P. 77–84.

[5] *Mei Q., Shen X., Zhai C.* Automatic labeling of multinomial topic models // 13th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining. — New York, NY, USA: ACM, 2007. P. 490–499.

[6] *Lau J. H., Grieser K., Newman D., Baldwin T.* Automatic labelling of topic models // 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. — Stroudsburg, PA, USA: ACL, 2011. P. 1536–1545.

[7] *Cano A. E., He Y., Xu R.* Automatic labelling of topic models learned from Twitter by summarisation // 52nd Annual Meeting of the Association for Computational Linguistics. — Baltimore, MD, USA: ACL, 2014. P. 618–624.

[8] *Hofmann T.* Probabilistic latent semantic indexing // 22nd Annual ACM SIGIR Conference (International) on Research and Development in Information Retrieval Proceedings. — ACM, 1999. P. 50–57.

[9] *Blei D., Ng Y., Jordan I.* Latent Dirichlet allocation // J. Mach. Learn. Research, 2003. Vol. 3. P. 993–1022.

[10] *Chuang J., Manning C., Heer J.* Termite: Visualization techniques for assessing textual topic models // Working Conference (International) on Advanced Visual Interfaces Proceedings. — ACM, 2013. P. 74–77.

[11] *Gretarsson B., O'Donovan J., Bostandjiev S., Hollerer T., Asuncion A., Newman D., Smyth P.* TopicNets: Visual analysis of large text corpora with topic modeling // ACM Trans. Intelligent Syst. Technol. (TIST), 2012. Vol. 3. No. 2. P. 23.

[12] *Asuncion A., Welling M., Smyth P., Teh Y.* On smoothing and inference for topic models // 25th Conference on Uncertainty in Artificial Intelligence Proceedings. — AUAI Press, 2009. P. 27–34.

[13] *Eades P.* A heuristic for graph drawing // Congressus Numerantium, 2010. Vol. 42. P. 146–160.

[14] *Lee H., Kihm J., Choo J., Stasko J., Park H.* iVisClustering: An interactive visual document clustering via topic modeling // Comput. Graph. Forum, 2012. Vol. 31. No. 3. P. 1155–1164.

[15] *Kuhn W.* The Hungarian method for the assignment problem // Nav. Res. Logist. Q., 1995. Vol. 2. P. 83–97.

[16] *Smith A., Hawes T., Myers M.* Hiérarchie: Interactive visualization for hierarchical topic models // Workshop on Interactive Language Learning, Visualization, and Interfaces Proceedings. — Baltimore, MD, USA: Association for Computational Linguistics, 2014. P. 71–78.

[17] *Cui W., Liu S., Tan L., Shi C., Song Y., Gao Z, Tong X., Qu H.* TextFlow: Towards better understanding of evolving topics in text // IEEE Trans. Vis. Comput. Gr., 2011. Vol. 17. No. 12. P. 2412–2421.

[18] *Dou W., Yu L., Wang X., Ma Z., Ribarsky W.* HierarchicalTopics: Visually exploring large text collections using topic hierarchies // IEEE Trans. Vis. Comput. Gr., 2013. Vol. 19. No. 12. P. 2002–2011.

[19] *Havre S., Hetzler B., Nowell L.* ThemeRiver: Visualizing thematic changes in large document collections // IEEE Trans. Vis. Comput. Gr., 2002. Vol. 17. No. 12. P. 9–20.

[20] *Cui W., Liu S., Wu Z., Wei H.* How hierarchical topics evolve in large text corpora // IEEE Trans. Vis. Comput. Gr., 2014. Vol. 20. No. 12. P. 2281–2290.

[21] *Card S., Nation D.* Degree-of-interest trees: A component of an attention-reactive user interface // Working Conference on Advanced Visual Interfaces Proceedings. — ACM, 2002. P. 231–245.

[22] *Snyder J., Knowles R., Dredze M., Gormley M., Wolfe T.* Topic models and metadata for visualizing text corpora // Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2013. Vol. 10. P. 5.

[23] *Sievert C., Shirley K.* LDAvis: A method for visualizing and interpreting topics // Workshop on Interactive Language Learning, Visualization, and Interfaces, 2014.

[24] *Wei F., Liu S., Song Y., Pan S., Zhou M., Qian W., Shi L., Tan L., Zhang Q.* TIARA: A visual exploratory text analytic system // 16th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining. — ACM, 2010. P. 153–162.

[25] *Cao N., Gotz D., Sun J., Lin Y., Qu H.* SolarMap: Multifaceted visual analytics for topic exploration // IEEE 11th Conference (International) on Data Mining (ICDM), 2011. P. 101–110.

[26] *Cui W., Qu H., Zhou H., Zhang W., Wolfe T., Skiena S.* Watch the story unfold with TextWheel: Visualization of large-scale news streams // ACM Trans. Intelligent Syst. Technol. (TIST), 2012. Vol. 3. No. 2. P. 20.

[27] *Gad S., Javed W., Ghani S., Elmqvist N., Ewing T., Hampton N., Ramakrishnan N.* ThemeDelta: Dynamic segmentations over temporal topic models // IEEE Trans. Vis. Comput. Gr., 2015. Vol. 21. No. 5. P. 672–685.

[28] *Günnemann N., Jarke M.* D-VITA: A visual interactive text analysis system using dynamic topic mining // BTW Workshops, 2013. P. 237–246.

[29] *Liu S., Wang X., Chen J., Zhu J., Guo B.* TopicPanorama: A full picture of relevant topics // IEEE Simposium on Visual Analytics Science and Technology (VAST) Proceedings, 2014. P. 183–192.

[30] *Chen A., Eichler G.* Topic modeling and network visualization to explore patient experiences // Visual Analytics in Healthcare Workshop, 2013.

[31] *McCallum A. K.* MALLET: A machine learning for language toolkit, 2002. `http://mallet.cs.umass.edu`.

References

[1] Blei, D. 2012. Probabilistic topic models. *Commun. ACM* 55(4):77–84.

[2] Daud, A., J. Li, L. Zhou, and F. Muhammad. 2010. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of Computer Science in China* 4(2):280–301.

[3] Chang, J., S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei. 2009. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*. Eds. Y. Bengio, D. Schuurmans, J. D. Lafferty, *et al.* MIT Press. 22:288–296.

[4] Chaney, A., and D. Blei. 2012. Visualizing topic models. *Frontiers of Computer Science in China* 55(4): 77–84.

[5] Mei, Q., X. Shen, and C. Zhai. 2007. Automatic labeling of multinomial topic models. *13th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining*. New York, NY: ACM. 490–499.

[6] Lau, J. H., K. Grieser, D. Newman, and T. Baldwin. 2011. Automatic labelling of topic models. *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Stroudsburg, PA: ACL. 1536–1545.

[7] Cano, A. E., Y. He, and R. Xu. 2014. Automatic labelling of topic models learned from Twitter by summarisation. *52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, MD: ACL. 618–624.

[8] Hofmann, T. 1999. Probabilistic latent semantic indexing. *22nd Annual ACM SIGIR Conference (International) on Research and Development in Information Retrieval Proceedings*. ACM. 50–57.

[9] Blei, D., Y. Ng, and I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Research* 3:993–1022.

[10] Chuang, J., C. Manning, and J. Heer. 2012. Termite: Visualization techniques for assessing textual topic models. *Working (International) Conference on Advanced Visual Interfaces Proceedings*. ACM. 74–77.

[11] Gretarsson, B., J. O'Donovan, S. Bostandjiev, T. Hollerer, A. Asuncion, D. Newman, and P. Smyth. 2012. TopicNets: Visual analysis of large text corpora with topic modeling. *ACM Trans. Intelligent Syst. Technol. (TIST)* 3(2):23.

[12] Asuncion, A., M. Welling, P. Smyth, and Y. Teh. 2009. On smoothing and inference for topic models. *25th Conference on Uncertainty in Artificial Intelligence Proceedings*. AUAI Press. 27–34.

[13] Eades, P. 2010. A heuristic for graph drawing. *Congressus Numerantium* 42:146–160.

[14] Lee, H., J. Kihm, J. Choo, J. Stasko, and H. Park. 2012. iVisClustering: An interactive visual document clustering via topic modeling. *Comput. Graph. Forum* 31(3):1155–1164.

[15] Kuhn, W. 1995. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* 2:83–97.

[16] Smith, A., T. Hawes, and M. Myers. 2014. Hiérarchie: Interactive visualization for hierarchical topic models. *Workshop on Interactive Language Learning, Visualization, and Interfaces Proceedings*. Baltimore, MD: Association for Computational Linguistics. 71–78.

[17] Cui, W., S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, X. Tong, and H. Qu. 2011. TextFlow: Towards better understanding of evolving topics in text. *IEEE Trans. Vis. Comput. Gr.* 17(12):2412–2421.

[18] Dou, W., L. Yu, X. Wang, Z. Ma, and W. Ribarsky. 2013. HierarchicalTopics: Visually exploring large text collections using topic hierarchies. *IEEE Trans. Vis. Comput. Gr.* 19(12):2002–2011.

[19] Havre, S., B. Hetzler, and L. Nowell. 2002. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Trans. Vis. Comput. Gr.* 17(12):9–20.

[20] Cui, W., S. Liu, Z. Wu, and H. Wei. 2014. How hierarchical topics evolve in large text corpora. *IEEE Trans. Vis. Comput. Gr.* 20(12):2281–2290.

[21] Card, S., and D. Nation. 2002. Degree-of-interest trees: A component of an attention-reactive user interface. *Working Conference on Advanced Visual Interfaces Proceedings*. ACM. 231–245.

[22] Snyder, J., R. Knowles, M. Dredze, M. Gormley, and T. Wolfe. 2013. Topic models and metadata for visualizing text corpora. *Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 10:5.

[23] Sievert, C., and K. Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. *Workshop on Interactive Language Learning, Visualization, and Interfaces*.

[24] Wei, F., S. Liu, Y. Song, S. Pan, M. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. 2010. TIARA: A visual exploratory text analytic system. *16th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining*. ACM. 153–162.

[25] Cao, N., D. Gotz, J. Sun, Y. Lin, and H. Qu. 2011. SolarMap: Multifaceted visual analytics for topic exploration. *IEEE 11th Conference (International) on Data Mining (ICDM)*. 101–110.

[26] Cui, W., H. Qu, H. Zhou, W. Zhang, T. Wolfe, and S. Skiena. 2012. Watch the story unfold with TextWheel: Visualization of large-scale news streams. *ACM Trans. Intelligent Syst. Technol. (TIST)* 3(2):20.

[27] Gad, S., W. Javed, S. Ghani, N. Elmqvist, T. Ewing, N. Hampton, and N. Ramakrishnan. 2015. ThemeDelta: Dynamic segmentations over temporal topic models. *IEEE Trans. Vis. Comput. Gr.* 21(5):672–685.

[28] Günnemann, N., and M. Jarke. 2013. D-VITA: A visual interactive text analysis system using dynamic topic mining. *BTW Workshops*. 237–246.

[29] Liu, S., X. Wang, J. Chen, J. Zhu, and B. Guo. 2014. TopicPanorama: A full picture of relevant topics. *IEEE Simposium on Visual Analytics Science and Technology (VAST) Proceedings*. 183–192.

[30] Chen, A., and G. Eichler. 2013. Topic modeling and network visualization to explore patient experiences. *Visual Analytics in Healthcare Workshop*.

[31] McCallum, A.K. 2002. MALLET: A machine learning for language toolkit. Available at: `http://mallet.cs.umass.edu` (accessed November 3, 2015).

О метрических свойствах медианы Кемени*

С. Д. Двоенко, Д. О. Пшеничный
sergedv@yandex.ru; denispshenichny@yandex.ru
Тульский государственный университет, г. Тула

Рассмотрена новая задача построения медианы Кемени с метрическими свойствами. При согласовании экспертных мнений требуется получить ранжирование, наименее отличающееся от остальных и имеющее смысл группового мнения. Медиана Кемени является эквивалентом среднего в шкалах (квази)порядков и свободна от противоречий, связанных с выявлением групповых мнений по правилу большинства (парадокс Эрроу). Известный локально-оптимальный алгоритм построения медианы Кемени основан на вычислении матрицы штрафов. Считая, что ранжирования, представленные парными расстояниями, погружены в евклидово метрическое пространство, можно определить средний элемент как центр такого множества. Такой центральный элемент также является ранжированием и должен иметь такой же смысл, как и медиана Кемени. Разработана процедура формирования скорректированной матрицы штрафов для построения метрической медианы Кемени, совпадающей со средним элементом данного множества.

Ключевые слова: *парные сравнения; метрика; правило большинства; медиана Кемени; ранжирование; парадокс Эрроу*

On metric characteristics of the Kemeny's median*

S. D. Dvoenko and D. O. Pshenichny
Tula State University, Tula

Background: For aggregating of expert's opinions, it is necessary to find the final ranking, which is the least different from others and represents the group opinion. The Kemeny's median appears to be a good idea of the average for scales of (quasi-)orderings and is free of some contradictions concerning the building of a group opinion based on the majority rules (Arrow's paradox). The well-known locally optimal algorithm to find the Kemeny's median depends on pairwise distances between rankings and calculates the so-called loss matrix.
Methods: It is assumed that the rankings represented by pairwise distances between them are immersed as a set in some Euclidean metric space. According to it, one can define the average element as the center of this set. Such central element is a ranking, too, and needs to be similar to the Kemeny's median. To be the Kemeny's median the mathematically correct center of the set of rankings, it needs to be like the center by its distances to other elements. The procedure is developed to build the modified loss matrix and to find the metric Kemeny's median, which coincides with the average element of the given set.
Results: In general, such center element differs by its distances from the corresponding distances of the Kemeny's median to other set elements. The authors find the metric Kemeny's median which coincides with the average element of the given set. Such ranking coincides with the classic Kemeny's median and proves its metric property, or differs from it, if the metric violations in the set configuration appear to be significant.
Concluding Remarks: The metric Kemeny's median is the correct center of the set of rankings and can be used for the correct version of k-means algorithm and others for ordering scales.

*Работа выполнена при финансовой поддержке РФФИ, проект № 15-07-02228.

Keywords: *pairwise comparisons; metrics; majority rule; Kemeny's median; ranking; Arrow's paradox*

1 Введение. Проблема согласования ранжирований

Задача выбора естественным образом заключается в выборе некоторого элемента множества альтернатив, который обладает какими-то наилучшими (с точки зрения авторов) характеристиками.

Хорошо известно, что одно из свойств такой задачи заключается в том, что выбор часто довольно трудно рационально обосновать. Выбор на практике часто основан на интуиции и опыте эксперта. Поэтому и говорят об индивидуальном выборе и предпочтениях индивидуума, т. е. эксперта. Если эксперту удалось выбрать наилучшую альтернативу, то, как правило, ему удается снова выбрать следующую наилучшую из оставшихся и т. д. В итоге альтернативы оказываются упорядоченными по предпочтениям данного эксперта и образуют ранжирование.

Пусть $A = \{a_1, \dots a_N\}$ — неупорядоченное множество альтернатив. Если считать, что альтернативы проиндексированы уже после их упорядочения по предпочтениям, то A — упорядоченное множество, представляющее строгое $P = a_1 \succ a_2 \succ \cdots \succ a_N$ или в общем случае нестрогое $P = a_1 \succcurlyeq a_2 \succcurlyeq \cdots \succcurlyeq a_N$ ранжирование. В последнем случае оказывается, что эксперт может не различать некоторые альтернативы (он не столь категоричен в своих предпочтениях и ставит их на одно место).

Таким образом, получение ранжирования означает, что для всех пар $(a_i,\ a_j) \in A \times A$ всегда можно указать, какая альтернатива лучше (не хуже) другой. Заметим, что обратное в общем случае неверно. Восстановление ранжирования на основе парных сравнений является отдельной задачей, которая здесь не рассматривается. Таким образом, ранжирование P также может быть представлено матрицей отношений $M_P(N, N)$ с элементами:

$$m_{ij} = \begin{cases} 1,\ a_i \succ a_j\,; \\ 0,\ a_i \sim a_j\,; \\ -1,\ a_i \prec a_j\,. \end{cases}$$

Для двух ранжирований P_u и P_v, представленных своими матрицами отношений M_{P_u} и M_{P_v}, можно вычислить расстояние:

$$d(P_u, P_v) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} |m_{ij}^u - m_{ij}^v| \tag{1}$$

при условии, что элементы из A перечислены одинаково (необязательно в соответствии с одним из этих ранжирований). Известно [1, 2], что эта величина представляет собой метрику на бинарных отношениях линейного (квази)порядка, к которым относятся ранжирования.

Пусть имеется n различных индивидуальных предпочтений (ранжирований). Нужно построить групповое отношение P, согласованное в некотором смысле с отношениями $P_1, \dots, P_n$. Способы построения группового отношения называют принципами согласования. Существуют разные принципы согласования. Если на способ согласования не наложены ограничения, то он, вообще говоря, может быть любым.

Наиболее распространенным принципом является правило большинства [3]. Например, если $P_1, \dots, P_n$ — ранжирования, то для пары альтернатив, где $a_i \succ a_j$, число

$n(i,j) = \sum\limits_{u=1}^{n} (m=1)_{ij}^{u}$ означает количество таких ранжирований, т. е. количество экспертов, имеющих такое предпочтение. Один из принципов согласования по правилу большинства (мажоритарный) имеет вид: $a_i \succ a_j$ в групповом отношении P, если $n(i,j) \geqslant n(j,i)$. Если ранжирования строгие, то это эквивалентно условию $n(i,j) \geqslant n/2$. Известно, что в общем случае мажоритарное отношение может быть нетранзитивным, даже если все индивидуальные предпочтения транзитивны.

Медиана P^* — это такое ранжирование, расстояние от которого до остальных $P_1, \ldots, P_n$ является минимальным:

$$P^* = \arg\min_{P} \sum_{u=1}^{n} d(P, P_u)\,. \tag{2}$$

Оказывается, если мажоритарное отношение транзитивно (или приведено к такому виду специальными процедурами), то оно является медианой и, в частности, медианой Кемени [1].

2 Построение медианы с метрическими свойствами

Рассмотрим ранжирования $P_1, \ldots, P_n$ как некоторое неупорядоченное множество элементов, погруженных некоторым способом в метрическое пространство и представленных матрицей $D(n,n)$ парных расстояний между собой, например, согласно (1). Если нет метрических нарушений в конфигурации элементов множества [5, 4], то центральный элемент P_0 можно представить своими расстояниями до остальных элементов согласно методу Торгерсона:

$$d^2(P_0, P_i) = d_{0i}^2 = \frac{1}{n}\sum_{p=1}^{n} d_{ip}^2 - \frac{1}{2n^2}\sum_{p=1}^{n}\sum_{q=1}^{n} d_{pq}^2,\ \ i = 1, \ldots, n\,. \tag{3}$$

Заметим, что метрические нарушения могут сделать невозможным вычисление расстояний (3), представляющих центральный элемент P_0 множества, так как второе слагаемое, представляющее дисперсию множества, может превысить по величине первое слагаемое [5, 4]. В то же время отсутствие второго слагаемого позволит вычислить такие расстояния всегда. Это будут расстояния, представляющие элемент P_{00}, вынесенный за пределы выпуклой оболочки данного множества.

Тогда, согласно известному свойству среднего арифметического, центральный элемент P_0 также оказывается наименее удаленным от всех остальных элементов данного множества и должен формально удовлетворять (2), являясь ранжированием, т. е. медианой.

Проблема заключается в том, что уже построенная медиана P^* представлена как ранжированием, так и своими расстояниями до остальных ранжирований из матрицы $D(n,n)$, добавляя к ней дополнительные строку и столбец. В то же время центральный (средний) элемент P_0 как ранжирование не существует, а представлен только своими расстояниями (3) до остальных элементов множества.

Совпадают ли эти ранжирования $P^* = P_0$? Если это так, то возникает возможность строить математически корректные версии известных алгоритмов распознавания и кластер-анализа для объектов, представленных измерениями в менее мощных (качественных) шкалах, на основе их парных сравнений [6].

В целом, такой подход известен [2, 7, 8]. В рамках такого подхода развивается аксиоматика метрик, введенных на бинарных отношениях, представляющих эквивалентности,

(квази)порядки и т. д. В данном же случае сразу предполагается наличие евклидовой метрики для элементов, погруженных в метрическое пространство.

Известно, что строгие и нестрогие ранжирования формально являются измерениями в шкалах строгого и нестрогого порядка. Корректным преобразованием, переводящим шкалы данного типа друг в друга, является монотонное преобразование, которое, перераспределяя положение альтернатив на числовой оси, не меняет их упорядоченного расположения относительно друг друга.

Достаточно очевидно, что в общем случае элемент P_0 и ранжирование P^* представлены каждый своими расстояниями до остальных элементов $P_1, \ldots, P_n$. Необходимо, используя подходящее монотонное преобразование, показать эквивалентность ранжирований P_0 и P^*. Это можно сделать, показав, что эти расстояния могут быть одинаковыми.

Применим для этого локально-оптимальный алгоритм построения медианы Кемени [1], предложенный в [2].

Известно, что алгоритм построения медианы Кемени основан на вычислении матрицы штрафов $Q(N, N)$, где N — число альтернатив. Пусть некоторое произвольное ранжирование P и ранжирования экспертов $P_1, \ldots, P_n$ представлены матрицами отношений M_P и $M_{P_1}, \ldots, M_{P_n}$. Суммарное расстояние от P до всех остальных ранжирований определяется как

$$\sum_{u=1}^{n} d(P, P_u) = \frac{1}{2} \sum_{u=1}^{n} \sum_{i=1}^{N} \sum_{j=1}^{N} |m_{ij} - m_{ij}^u| = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{u=1}^{n} |m_{ij} - m_{ij}^u| = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{u=1}^{n} d_{ij}(P, P_u)\,,$$

где при условии $m_{ij} = 1$ частичные «расстояния» определяются как

$$d_{ij}(P, P_u) = \begin{cases} 0, & m_{ij}^u = 1\,; \\ 1, & m_{ij}^u = 0\,; \\ 2, & m_{ij}^u = -1\,. \end{cases} \tag{4}$$

Элемент матрицы штрафов q_{ij} определяет суммарный штраф на несовпадение предпочтения $a_i \succ a_j$ в неизвестном ранжировании P по сравнению с соответствующим предпочтением в каждом из ранжирований $P_1, \ldots P_n$:

$$q_{ij} = \sum_{u=1}^{n} d_{ij}(P, P_u)\,.$$

Алгоритм построения медианы Кемени находит такое упорядочение альтернатив, что сумма элементов матрицы Q над ее диагональю минимальна.

3 Формирование матрицы штрафов

Представляется достаточно очевидным, что частичные «расстояния» (4) достаточно условны. Поэтому также развивается направление, связанное с введением так называемых метризованных бинарных отношений [2], когда элемент $m_{ij} = w_{ij}$ матрицы отношений имеет значения w_{ij}, отличающиеся от $-1, 0, 1$.

Как было сказано выше, для доказательства метричности медианы Кемени необходимо показать, что $P_0 = P^*$, представив P_0 и P^* своими одинаковыми расстояниями до остальных элементов множества. В данном случае появляется возможность применить такое монотонное преобразовани шкалы ранжирования для эксперта u, что матрица отношения M_{P_u}, представляющая его ранжирование P_u, естественным образом оказывается

метризованной, так как метрическая медиана (как среднее по множеству) не нарушает метричности конфигурации элементов, представляющих ранжирования экспертов.

Рассмотрим ранжирования P_u, P_0 и P^*, где $\delta = d(P_0, P_u) - d(P^*, P_u) \neq 0$. Пусть $\delta > 0$. Для компенсации такой разницы в расстояниях необходимо равномерно распределить значение $\delta > 0$ по ненулевым элементам матрицы отношений M_{P_u}, сформировав новую матрицу отношений с элементами:

$$m_{ij}^u = \begin{cases} +1 + \dfrac{2\delta}{k}\,, & a_i \succ a_j\,; \\ 0\,, & a_i \sim a_j\,; \\ -1 - \dfrac{2\delta}{k}\,, & a_i \prec a_j\,, \end{cases}$$

где $k = N^2 - N - N_0$ — общее число ненулевых элементов без главной диагонали, N_0 — число нулевых недиагональных элементов $m_{ij}^u = 0$. Очевидно, что такая матрица отношений не изменяет ранжирования эксперта P_u.

В этом случае при $m_{ij} = 1$ предположение $a_i \succ a_j$ в неизвестном ранжировании P штрафуется экспертом с формированием частичных расстояний:

$$d_{ij}(P, P_u) = \begin{cases} \dfrac{2\delta}{k}\,, & m_{ij}^u = +1 + \dfrac{2\delta}{k}\,; \\ 1\,, & m_{ij}^u = \ \ 0\,; \\ 2 + \dfrac{2\delta}{k}\,, & m_{ij}^u = -1 - \dfrac{2\delta}{k}\,. \end{cases}$$

Пусть $\delta < 0$. Для компенсации такой разницы в расстояниях также необходимо равномерно распределить значение $\delta < 0$ по ненулевым элементам матрицы отношений M_{P_u}. Кроме того, множество изменяемых элементов $k' = k - \Delta k$ дополнительно уменьшается за счет Δk совпадающих элементов $m_{ij}^u = m_{ij}^*$ в матрицах отношений, представляющих ранжирование эксперта P_u и медиану P^*. Действительно, в этом случае $d_{ij}(P^*, P_u) = 0$ и уменьшить его невозможно.

Добавление любой отрицательной величины к значению m_{ij}^u в этом случае будет означать изменение ранжирования эксперта, при котором расстояние между ранжированиями P_u и P^* только возрастет.

Если величина $-1 < 2\delta/k' < 0$, то можно, не изменяя ранжирования P_u, сформировать новую матрицу отношений M_{P_u} с элементами:

$$m_{ij}^u = \begin{cases} +1 - \left|\dfrac{2\delta}{k'}\right|\,, & a_i \succ a_j\,; \\ 0\,, & a_i \sim a_j\,; \\ -1 + \left|\dfrac{2\delta}{k'}\right|\,, & a_i \prec a_j\,; \\ m_{ij}^*\,, & m_{ij}^u = m_{ij}^*\,, \end{cases}$$

где при $m_{ij} = 1$ предположение $a_i \succ a_j$ в неизвестном ранжировании P штрафуется экспертом с формированием частичных расстояний:

$$d_{ij}(P, P_u) = \begin{cases} 0, & m_{ij}^u = 1\,; \\ 1, & m_{ij}^u = 0\,; \\ 2, & m_{ij}^u = -1\,; \\ \left|\dfrac{2\delta}{k'}\right|, & m_{ij}^u = +1 - \left|\dfrac{2\delta}{k'}\right|\,; \\ 2 - \left|\dfrac{2\delta}{k'}\right|, & m_{ij}^u = -1 + \left|\dfrac{2\delta}{k'}\right|\,. \end{cases}$$

Если величина $2\delta/k' \leqslant -1$, то корректировка элементов матрицы отношений M_{P_u} неизбежно изменит знак некоторых элементов m^u_{ij}, что означает изменение ранжирования P_u. Так как изменять экспертное ранжирование мы не имеем права, то следует изменить ранжирование, соответствующее медиане P^*. Фактически это означает, что ранжирования P_0 и P^* различны. Поэтому, изменяя P^*, найдем другое ранжирование, лучше соответствующее P_0. В этом случае при $-2 \leqslant 2\delta/k' \leqslant -1$ формируется новая матрица отношений M_{P^*} с элементами:

$$m^*_{ij} = \begin{cases} +1 - \left|\dfrac{2\delta}{k'}\right|, & a_i \succ a_j\,; \\ 0, & a_i \sim a_j\,; \\ -1 + \left|\dfrac{2\delta}{k'}\right|, & a_i \prec a_j\,; \\ m^u_{ij}, & m^u_{ij} = m^*_{ij}\,, \end{cases}$$

где при $m_{ij} = 1$ предположение $a_i \succ a_j$ в неизвестном ранжировании P штрафуется ранжированием P^* с формированием частичных расстояний:

$$d_{ij}(P, P^*) = \begin{cases} 0, & m^*_{ij} = 1\,; \\ 1, & m^*_{ij} = 0\,; \\ 2, & m^*_{ij} = -1\,; \\ \left|\dfrac{2\delta}{k'}\right|, & m^*_{ij} = +1 - \left|\dfrac{2\delta}{k'}\right|\,; \\ 2 - \left|\dfrac{2\delta}{k'}\right|, & m^*_{ij} = -1 + \left|\dfrac{2\delta}{k'}\right|\,. \end{cases}$$

Если величина $2\delta/k' < -2$, то распределим по ранее указанным k' элементам только величину -2. Тогда отрицательный остаток $(2\delta/k' + 2)k' = 2\delta + 2k' < 0$ следует распределить по k_0 нулевым элементам $m^*_{ij} = 0$ матрицы отношений для ранжирования P^*. В итоге формируется новая матрица отношений M_{P^*} с элементами:

$$m^*_{ij} = \begin{cases} +1 - 2\,, & a_i \succ a_j\,; \\ -\left|\dfrac{2\delta}{k_0} + \dfrac{2k'}{k_0}\right|, & a_i \sim a_j\,; \\ -1 + 2\,, & a_i \prec a_j\,; \\ m^u_{ij}\,, & m^u_{ij} = m^*_{ij}\,, \end{cases}$$

где при $m_{ij} = 1$ предположение $a_i \succ a_j$ в неизвестном ранжировании P штрафуется ранжированием P^* с формированием частичных расстояний:

$$d_{ij}(P, P^*) = \begin{cases} 0, & m^*_{ij} = 1\,; \\ 1, & m^*_{ij} = 0\,; \\ 2, & m^*_{ij} = -1\,; \\ 1 + \left|\dfrac{2\delta}{k_0} + \dfrac{2k'}{k_0}\right|, & m^*_{ij} = -\left|\dfrac{2\delta}{k_0} + \dfrac{2k'}{k_0}\right|\,. \end{cases}$$

Легко увидеть, что после такой корректировки матрицы отношений M_{P^*} новое ранжирование P^*, вообще говоря, уже не является медианой, так как в нем произошли перестановки (инверсии) на некоторых парах альтернатив.

При вычислении матрицы штрафов Q учитываются только измененные матрицы отношений для ранжирований $P_1, \ldots, P_n$ и измененные матрицы для ранжирования P^* в тех случаях, когда оно корректировалось вместо соответствующего ранжирования эксперта.

4 Эксперименты

4.1 Исходные данные о ранжировании проектов

В качестве экспериментальных данных были рассмотрены материалы исследования 2000 г. по оценке и выбору 14 проектов, обеспечивающих достижение стратегических целей ОАО «Газпром» [9]. В табл. 1 приведен перечень инвестиционных проектов, в осуществлении которых предполагалось участие «Газпрома». Указанный перечень был сформирован по материалам открытой и зарубежной печати.

Таблица 1 Перечень инвестиционных проектов

№	Проекты	Содержание проектов
1	Южный парс	Освоение двух новых нефтегазоносных участков на месторождении «Южный Парс» (Иран)
2	Голубой поток	Строительство морского участка газопровода (двух ниток протяженностью 380 км) и компрессорной станции «Береговая»
3	Ямал–Европа	Завершение строительства трансконтинентального трубопровода для транспортировки сибирского газа в Европу
4	Псковская ГРЭС	Приобретение в счет долгов «Газпрому» Псковской ГРЭС и завершение строительства 3-го энергоблока
5	Метан Кузбасс	Создание компанией «Метан Кузбасса» и администрацией Кемеровской области опытно-промышленного производства по добыче метана из угольных пластов
6	Приразломное	Освоение совместно с партнерами арктического нефтяного месторождения «Приразломное»
7	Трансбалканский трубопровод	Строительство газокомпрессорной станции на Трансбалканском газопроводе. Расширение мощностей транспортировки газа на Трансбалканском газопроводе
8	Газопровод Петрозаводск–Кондопога	Строительство в Карелии 68-километрового трубопровода для обеспечения газом Кондопожского целлюлозно-бумажного комбината и жителей севера республики
9	Экология	Совместный проект сокращения эмиссии углекислого газа на участке «Ужгородского коридора» (Волготрансгаз)
10	Дегазификация энергетики	Снижение расхода газа на нужды электроэнергетики
11	Штокмановское	Освоение Штокмановского газоконденсатного месторождения, разведка и освоение добычи газа и газового конденсата Штокмановского газоконденсатного месторождения
12	Газификация автотранспорта	Газификация автотранспорта в Новосибирской области, отказ от бензина, использование пропан-бутановой смеси метана
13	Космическая связь	Создание группировки спутниковой связи для обеспечения «Газпрома» коммерческой технологической связью
14	АСУ корпоративными финансами	Создание для ОАО «Газпром» автоматизированной системы управления корпоративными финансами

Таблица 2 Стандартизированные ранги инвестиционных проектов

№	Проекты	1	2	3	4	5	6	7	8	МК
1	Южный парс	5	4	14	8	11	12	4,5	5,5	9
2	Голубой поток	1	1	3,5	4	12,5	14	9,5	10	4
3	Ямал-Европа	3	3	8,5	3	14	3	11	7	10
4	Псковская ГРЭС	9	7	10	8	9	4,5	4,5	2	2
5	Метан Кузбасс	8	9,5	5,5	8	2,5	7,5	4,5	11	5
6	Приразломное	6	6	8,5	8	9	9	12,5	13	13
7	Трансбалканский трубопровод	10	5	12,5	13,5	6	4,5	4,5	4	7
8	Газопровод Петрозаводск–Кондопога	11	12,5	11	1	2,5	2	4,5	8,5	3
9	Экология	13	12,5	5,5	12	2,5	2	4,5	3	6
10	Дегазификации энергетики	12	12,5	1	8	12,5	6	14	1	14
11	Штокмановское	2	2	2	2	6	7,5	12,5	14	1
12	Газификация автотранспорта	4	8	12,5	8	9	10,5	4,5	5,5	8
13	Космическая связь	7	9,5	7	8	6	10,5	4,5	12	11
14	АСУ корпоративными финансами	14	12,5	3,5	13,5	2,5	2	9,5	8,5	12

Проекты ранжировались по 8 критериям двух видов (выгоды и негативных эффектов):
1) финансово-экономическая выгода;
2) выгода от изменения конъюнктуры рынка;
3) производственно-технологическая выгода;
4) социально-политическая выгода;
5) политические негативные последствия;
6) негативные последствия рыночной конкуренции;
7) социальные негативные последствия;
8) негативные финансово-экономические последствия.

По характеристикам 1–4 один проект является предпочтительнее другого, если соответствующее значение критерия имеет значение больше, чем у другого проекта. По характеристикам 5–8 проект тем предпочтительнее, чем меньшее значение имеет соответствующий критерий. Наиболее предпочтительная альтернатива получает ранг 1, следующая за ней — 2 и т. д. Возможны случаи, когда две и более альтернатив имеют одинаковые значения критериев, т. е. одинаковый ранг. Нам удобно рассматривать критерии как экспертов, а ранжирования — как результат экспертизы. В табл. 2 представлены 8 экспертных ранжирований 14 проектов, где ранжирования представлены стандартизированными рангами, так как некоторые проекты имели одинаковый ранг.

Алгоритм построения медианы Кемени для матрицы штрафов:

$$\begin{pmatrix}
0 & 8 & 8 & 10 & 8 & 7 & 9 & 9 & 9 & 7 & 12 & 11 & 8 & 6 \\
8 & 0 & 6 & 8 & 6 & 4 & 8 & 10 & 8 & 7 & 8 & 8 & 6 & 8 \\
8 & 10 & 0 & 8 & 8 & 5 & 8 & 8 & 10 & 8 & 12 & 8 & 8 & 8 \\
6 & 8 & 8 & 0 & 8 & 8 & 6 & 7 & 7 & 5 & 10 & 5 & 8 & 6 \\
8 & 10 & 8 & 8 & 0 & 5 & 7 & 8 & 7 & 7 & 9 & 8 & 5 & 7 \\
9 & 12 & 11 & 8 & 11 & 0 & 10 & 10 & 10 & 7 & 13 & 8 & 9 & 10 \\
7 & 8 & 8 & 10 & 9 & 6 & 0 & 9 & 11 & 6 & 9 & 6 & 8 & 7 \\
7 & 6 & 8 & 9 & 8 & 6 & 7 & 0 & 8 & 5 & 6 & 7 & 7 & 6 \\
7 & 8 & 6 & 9 & 9 & 6 & 5 & 8 & 0 & 9 & 8 & 7 & 7 & 5 \\
9 & 9 & 8 & 11 & 9 & 9 & 10 & 11 & 7 & 0 & 10 & 9 & 9 & 7 \\
4 & 8 & 4 & 6 & 7 & 3 & 7 & 10 & 8 & 6 & 0 & 4 & 5 & 8 \\
5 & 8 & 8 & 11 & 8 & 8 & 10 & 9 & 9 & 7 & 12 & 0 & 7 & 6 \\
8 & 10 & 8 & 8 & 11 & 7 & 8 & 9 & 9 & 7 & 11 & 9 & 0 & 8 \\
10 & 8 & 8 & 10 & 9 & 6 & 9 & 10 & 11 & 9 & 8 & 10 & 8 & 0
\end{pmatrix}$$

дает ранжирование P^* (МК), которое также представлено в табл. 2.

4.2 Построение ранжирования, представленного средним объектом

Рассмотрим теперь матрицу парных расстояний между ранжированиями:

$$\begin{pmatrix} 0 & 27 & 98 & 50 & 131 & 156 & 102 & 124 \\ 27 & 0 & 105 & 65 & 128 & 137 & 101 & 113 \\ 98 & 105 & 0 & 88 & 83 & 82 & 128 & 106 \\ 50 & 65 & 88 & 0 & 109 & 116 & 96 & 112 \\ 131 & 128 & 83 & 109 & 0 & 37 & 61 & 101 \\ 156 & 137 & 82 & 116 & 37 & 0 & 80 & 74 \\ 102 & 101 & 128 & 96 & 61 & 80 & 0 & 74 \\ 124 & 113 & 106 & 112 & 101 & 74 & 74 & 0 \end{pmatrix}$$

и проверим метричность конфигурации множества ранжирований как множества элементов, погруженных в метрическое пространство.

Скорректируем метрические нарушения согласно технологии, разработанной в [5, 4]. Определим элемент, вынесенный за пределы выпуклой оболочки данного множества, и представим его своими расстояниями до остальных элементов:

$$(100{,}07 \quad 96{,}243 \quad 93{,}278 \quad 87{,}725 \quad 91{,}928 \quad 97{,}807 \quad 87{,}893 \quad 95{,}576).$$

Построим матрицу нормированных скалярных произведений относительного данного элемента как начала координат и определим ее собственные числа:

$$(-0{,}25079 \quad -0{,}11443 \quad 0{,}11728 \quad 0{,}23633 \quad 0{,}57857 \quad 1{,}0547 \quad 2{,}3446 \quad 4{,}0338).$$

Так как среди них есть отрицательные, то скорректируем эту матрицу скалярных произведений, получив только положительные собственные числа:

$$(0{,}01098 \quad 0{,}04016 \quad 0{,}11631 \quad 0{,}19909 \quad 0{,}47382 \quad 1{,}03831 \quad 2{,}18110 \quad 3{,}94022).$$

Восстановим скорректированную матрицу расстояний:

$$\begin{pmatrix} 0 & 41,281 & 98 & 50 & 125,72 & 143,32 & 102 & 124 \\ 41,281 & 0 & 103,35 & 64,011 & 127,52 & 137,96 & 99,346 & 113,92 \\ 98 & 103,35 & 0 & 88 & 95,464 & 97,307 & 128 & 106 \\ 50 & 64,011 & 88 & 0 & 108,47 & 120,33 & 96 & 112 \\ 125,72 & 127,52 & 95,464 & 108,47 & 0 & 53,025 & 81,431 & 90,68 \\ 143,32 & 137,96 & 97,307 & 120,33 & 53,025 & 0 & 94,451 & 73,325 \\ 102 & 99,346 & 128 & 96 & 81,431 & 94,451 & 0 & 74 \\ 124 & 113,92 & 106 & 112 & 90,68 & 73,325 & 74 & 0 \end{pmatrix}.$$

Так как медиана Кемени построена (МК, см. табл. 2), то определим расстояния от нее до остальных ранжирований по их матрицам отношений. Для скорректированной матрицы расстояний вычислим средний элемент P_0 (СР) и представим его своими расстояниями до остальных элементов множества. Также вычислим разницу δ в расстояниях от среднего элемента и медианы Кемени до остальных элементов множества (табл. 3).

Как было сказано выше, нужно добиться, чтобы не было разницы в расстояниях от медианы Кемени и от среднего элемента до остальных элементов множества. Так как расстояние между ранжированиями определяется по их матрицам отношений, то необходимо распределить соответствующую δ равномерно по элементам матрицы отношений эксперта.

Равномерная корректировка матрицы отношений для ранжирования каждого эксперта выполняется соответствующим удвоенным средним значением $2\delta/k$ (см. табл. 3), так как

Таблица 3 Корректировка матриц отношений для ранжирований экспертов

MK	CP	Разница, δ	Число корректируемых элементов, k	Среднее, $2\delta/k$
76	70,625	−5,375	76	−0,14144
73	69,458	−3,542	66	−0,10732
92	69,102	−22,898	88	−0,52041
66	57,353	−8,647	44	−0,39302
71	65,747	−5,253	58	−0,18113
84	74,666	−9,334	78	−0,23934
68	62,120	−5,880	38	−0,3095
88	66,313	−21,687	86	−0,50435

для вычисления расстояния между двумя ранжированиями каждое различие между парой элементов в матрице отношений учитывается дважды.

Как уже было отмечено, это можно сделать, если такое распределение не изменяет знаков элементов матрицы отношений для соответствующего ранжирования. В этом случае мы не меняем ранжирования эксперта, так как, вообще говоря, мы и не имеем права этого делать. Поэтому для равномерной корректировки выбираются только ненулевые элементы матрицы отношений, так как нуль означает, что два соответствующих проекта занимают одно и то же место в ранжировании эксперта. Такая корректировка естественным образом применяется для компенсации положительной разницы в расстоянии между ранжированиями, так как оно увеличивается.

При распределении отрицательной разницы (см. табл. 3) число корректируемых элементов матрицы отношений дополнительно уменьшается, так как корректируются ненулевые элементы и дополнительно только те, которые отличаются от соответствующих элементов в матрице отношений для медианы. Очевидно, что только в этом случае различие можно уменьшить, учитывая отрицательную разницу. Если при этом знаки элементов матрицы отношений для ранжирования эксперта не изменяются, то его ранжирование не изменяется. Если изменяются, то нельзя можем корректировать матрицу отношений для ранжирования эксперта, так как это означает изменение его ранжирования. Поэтому будем корректировать соответствующие элементы матрицы отношений для медианы. Действительно, это ранжирование все равно должно измениться. В этом случае, если величина $-2 \leqslant 2\delta/k < 0$, то эта разница распределяется по ненулевым элементам и отличающимся от соответствующих элементов матрицы отношений для ранжирования эксперта. Если величина $2\delta/k < -2$, то распределим по таким элементам матрицы отношений для медианы только величину -2.

Естественно, что вся отрицательная разница еще не будет скомпенсирована, так как останется еще распределить остаток отрицательной разницы $(2\delta/k + 2)k$ по k_0 элементам, где k_0 — это число нулевых элементов в матрице отношений. Можно показать, что при $k_0 = 0$ величина $2\delta/k$ распределится полностью, а при $k_0 \neq 0$ остаток распределится полностью.

При формировании матрицы штрафов для алгоритма построения медианы Кемени будем, как и раньше, штрафовать наше предположение о предпочтительности одного проекта перед другим по всем возможным парам, когда соответствующий элемент неизвестной матрицы отношений равен 1. Если при этом корректировалась матрица отношений для ранжирования эксперта, то его ранжирование штрафует наши предположения. Если кор-

Таблица 4 Корректировка матриц отношений для ранжирований экспертов

МК	ВП	Разница, δ	Число корректируемых элементов, k	Среднее, $2\delta/k$
76	97,426	21,426	182	0,23545
73	96,583	23,583	168	0,28075
92	96,327	4,327	174	0,049739
66	88,28	22,28	138	0,32289
71	93,95	22,95	156	0,29423
84	100,39	16,393	170	0,19286
68	91,448	23,448	122	0,38439
88	94,346	6,346	178	0,071308

ректировалась матрица отношений для медианы, то только это ранжирование штрафует наши предположения.

В итоге, алгоритм построения медианы Кемени для измененной матрицы штрафов

$$\begin{pmatrix}
0 & 9{,}23 & 6{,}84 & 10{,}25 & 8{,}75 & 6{,}06 & 9{,}64 & 9{,}75 & 9{,}64 & 5{,}74 & 12{,}81 & 11{,}11 & 7{,}06 & 5{,}06 \\
6{,}77 & 0 & 4{,}86 & 9{,}162 & 5{,}27 & 3{,}58 & 6{,}77 & 10{,}77 & 6{,}77 & 5{,}74 & 9{,}06 & 6{,}77 & 5{,}27 & 7{,}08 \\
9{,}16 & 11{,}14 & 0 & 9{,}16 & 9{,}15 & 4{,}58 & 9{,}16 & 9{,}27 & 10{,}64 & 6{,}55 & 12{,}81 & 9{,}16 & 6{,}75 & 6{,}75 \\
5{,}75 & 6{,}84 & 6{,}84 & 0 & 7{,}16 & 7{,}23 & 5{,}71 & 6{,}19 & 6{,}06 & 3{,}98 & 11{,}05 & 4{,}86 & 7{,}16 & 5{,}06 \\
7{,}25 & 10{,}73 & 6{,}85 & 8{,}84 & 0 & 4{,}75 & 6{,}15 & 8{,}77 & 6{,}26 & 5{,}74 & 9{,}995 & 7{,}25 & 4{,}86 & 5{,}74 \\
9{,}94 & 12{,}42 & 11{,}42 & 8{,}77 & 11{,}25 & 0 & 11{,}06 & 10{,}77 & 10{,}64 & 5{,}74 & 13{,}50 & 8{,}87 & 9{,}49 & 10{,}64 \\
6{,}36 & 9{,}23 & 6{,}84 & 10{,}29 & 9{,}85 & 4{,}95 & 0 & 9{,}75 & 11{,}25 & 4{,}58 & 10{,}05 & 5{,}47 & 6{,}95 & 6{,}06 \\
6{,}25 & 5{,}23 & 6{,}73 & 9{,}81 & 7{,}23 & 5{,}23 & 6{,}25 & 0 & 6{,}98 & 3{,}98 & 7{,}63 & 6{,}25 & 6{,}23 & 5{,}48 \\
6{,}36 & 9{,}23 & 5{,}36 & 9{,}94 & 9{,}74 & 5{,}36 & 4{,}75 & 9{,}02 & 0 & 7{,}44 & 9{,}23 & 6{,}36 & 6{,}36 & 4{,}48 \\
10{,}26 & 10{,}26 & 9{,}45 & 12{,}03 & 10{,}26 & 10{,}26 & 11{,}42 & 12{,}03 & 8{,}56 & 0 & 11{,}26 & 10{,}26 & 10{,}26 & 8{,}56 \\
3{,}19 & 6{,}94 & 3{,}19 & 4{,}95 & 6{,}01 & 2{,}50 & 5{,}95 & 8{,}37 & 6{,}77 & 4{,}74 & 0 & 3{,}19 & 4{,}19 & 6{,}77 \\
4{,}89 & 9{,}23 & 6{,}84 & 11{,}14 & 8{,}75 & 7{,}13 & 10{,}53 & 9{,}75 & 9{,}64 & 5{,}74 & 12{,}81 & 0 & 6{,}30 & 5{,}06 \\
8{,}94 & 10{,}73 & 9{,}25 & 8{,}84 & 11{,}14 & 6{,}51 & 9{,}05 & 9{,}77 & 9{,}64 & 5{,}74 & 11{,}81 & 9{,}70 & 0 & 6{,}55 \\
10{,}94 & 8{,}92 & v9{,}25 & 10{,}94 & 10{,}26 & 5{,}36 & 9{,}94 & 10{,}52 & 11{,}52 & 7{,}44 & 9{,}23 & 10{,}94 & 9{,}45 & 0
\end{pmatrix}$$

дает ту же медиану Кемени (см. табл. 2), у которой ее расстояния до остальных ранжирований не отличаются от расстояний среднего элемента до остальных элементов множества из табл. 3.

Таким образом, мы показали, что среднее множества элементов, представляющих ранжирования, также представленное ранжированием, является медианой Кемени.

4.3 Построение ранжирования, представленного произвольным объектом

Снова построим медиану Кемени для ранжирований экспертов и определим ее расстояния до остальных ранжирований. Снова скорректируем исходную матрицу расстояний между ранжированиями, устранив метрические нарушения в конфигурации элементов множества.

Рассмотрим теперь элемент P_{00} (ВП), вынесенный за пределы выпуклой оболочки множества элементов, представляющих ранжирования экспертов. Представим его своими расстояниями до остальных элементов множества (табл. 4). Найдем соответствующее

Таблица 5 Построенные ранжирования

№	ВП	СР	МК
1	11	9	9
2	5	4	4
3	10	10	10
4	2	2	2
5	6	5	5
6	14	13	13
7	7	7	7
8	3	3	3
9	4	6	6
10	12	14	14
11	1	1	1
12	8	8	8
13	9	11	11
14	13	12	12

ему ранжирование, также применив алгоритм построения медианы Кемени к соответствующим образом измененной матрице штрафов.

Алгоритм построения медианы Кемени для измененной матрицы штрафов:

$$\begin{pmatrix}
0 & 6{,}17 & 7{,}95 & 8{,}88 & 6{,}88 & 6{,}56 & 7{,}55 & 7{,}55 & 7{,}55 & 6{,}12 & 10{,}17 & 9{,}95 & 7{,}95 & 5{,}24 \\
8{,}05 & 0 & 5{,}34 & 6{,}17 & 5{,}91 & 3{,}14 & 8{,}05 & 8{,}17 & 8{,}05 & 6{,}09 & 6{,}17 & 8{,}05 & 5{,}91 & 7{,}72 \\
6{,}17 & 8{,}17 & 0 & 6{,}17 & 6{,}17 & 4{,}19 & 6{,}17 & 6{,}17 & 8{,}17 & 7{,}38 & 10{,}17 & 6{,}17 & 8{,}01 & 8{,}01 \\
5{,}91 & 7{,}95 & 7{,}95 & 0 & 8{,}03 & 7{,}92 & 5{,}90 & 7{,}17 & 6{,}63 & 3{,}73 & 8{,}17 & 4{,}64 & 8{,}03 & 5{,}24 \\
8{,}05 & 8{,}17 & 7{,}99 & 6{,}88 & 0 & 4{,}52 & 6{,}64 & 6{,}85 & 6{,}43 & 6{,}12 & 7{,}36 & 8{,}05 & 4{,}63 & 6{,}09 \\
7{,}49 & 10{,}17 & 9{,}22 & 6{,}79 & 9{,}49 & 0 & 8{,}17 & 8{,}17 & 8{,}17 & 6{,}12 & 11{,}55 & 6{,}79 & 7{,}49 & 8{,}17 \\
7{,}23 & 6{,}17 & 7{,}95 & 8{,}75 & 7{,}55 & 5{,}38 & 0 & 7{,}55 & 9{,}55 & 5{,}06 & 7{,}462 & 5{,}72 & 8{,}06 & 6{,}56 \\
6{,}73 & 5{,}30 & 7{,}44 & 7{,}55 & 7{,}99 & 5{,}30 & 6{,}73 & 0 & 7{,}56 & 3{,}69 & 4{,}17 & 6{,}73 & 6{,}68 & 5{,}11 \\
7{,}23 & 6{,}17 & 5{,}85 & 7{,}55 & 7{,}90 & 5{,}85 & 4{,}59 & 7{,}32 & 0 & 8{,}81 & 6{,}17 & 7{,}23 & 7{,}23 & 4{,}04 \\
7{,}49 & 7{,}46 & 6{,}17 & 9{,}49 & 7{,}49 & 7{,}49 & 8{,}17 & 9{,}45 & 5{,}45 & 0 & 8{,}17 & 7{,}49 & 7{,}49 & 5{,}45 \\
3{,}08 & 8{,}11 & 3{,}08 & 5{,}47 & 6{,}86 & 1{,}70 & 6{,}76 & 10{,}7 & 8{,}05 & 4{,}80 & 0 & 3{,}08 & 4{,}37 & 8{,}05 \\
4{,}51 & 6{,}17 & 7{,}95 & 10{,}17 & 6{,}88 & 7{,}83 & 8{,}60 & 7{,}55 & 7{,}55 & 6{,}12 & 10{,}17 & 0 & 6{,}76 & 5{,}24 \\
6{,}88 & 8{,}17 & 6{,}17 & 6{,}88 & 10{,}16 & 6{,}91 & 6{,}85 & 7{,}55 & 7{,}55 & 6{,}12 & 9{,}46 & 8{,}07 & 0 & 7{,}38 \\
8{,}17 & 6{,}60 & 6{,}17 & 8{,}17 & 7{,}46 & 5{,}85 & 7{,}49 & 9{,}01 & 9{,}94 & 8{,}81 & 6{,}17 & 8{,}17 & 6{,}17 & 0
\end{pmatrix}$$

дает новое ранжирование (ВП), показанное в табл. 5. Легко увидеть, что данное ранжирование отличается от медианы Кемени, расстояние между данным ранжированием и медианой Кемени по матрицам их отношений равно 14.

5 Заключение

В экспериментах было показано, что одному и тому же ранжированию могут соответствовать разные векторы расстояний до остальных элементов данного множества (ранжирований), именно поэтому авторам удалось доказать метричность медианы Кемени на представленных данных. Очевидно, что такое свойство обеспечивается монотонным преобразованием ранговой шкалы, не нарушающим порядок элементов множества.

Также было показано, что вектор расстояний элемента, сильно отличающегося от среднего (центрального) элемента множества, формирует другое ранжирование, не совпадающее с ранжированием для среднего объекта.

В общем случае можно показать, что, и наоборот, разным ранжированиям могут соответствовать одинаковые векторы расстояний до других элементов множества (ранжирований). В частности, используя алгоритм построения медианы Кемени, можно в каждом случае построить ранжирование, имеющее такие же расстояния до остальных, как и индивидуальное ранжирование эксперта. В этом случае ранжирование, которое является медианой относительно исходной матрицы штрафов, неизбежно подвергнется немонотонному преобразованию, которое изменит упорядочение элементов в данном ранжировании.

Литература

[1] *Кемени Дж., Снелл Дж.* Кибернетическое моделирование. — М.: Сов. Радио, 1972. 192 с.

[2] *Литвак Б. Г.* Экспертная информация: методы получения и анализа. — М.: Радио и связь, 1982. 184 с.

[3] *Миркин Б. Г.* Проблема группового выбора. — М.: Наука, 1974. 256 с.

[4] *Двоенко С. Д., Пшеничный Д. О.* О метрической коррекции матриц парных сравнений // Машинное обучение и анализ данных, 2013. Т. 1. № 5. С. 606–620. http://jmlda.org/papers/doc/2013/no5/.

[5] *Двоенко С. Д., Пшеничный Д. О.* Оптимальная коррекция метрических нарушений в матрицах парных сравнений // Машинное обучение и анализ данных, 2014. Т. 1. № 7. С. 885–890. http://jmlda.org/papers/doc/2014/no7/.

[6] *Двоенко С. Д.* Кластеризация множества, описанного парными расстояниями и близостями между его элементами // Сибирский журнал индустриальной математики, 2009. Т. 12. № 1(37). С. 61–73.

[7] *Миркин Б. Г.* Анализ качественных признаков. — М.: Статистика, 1976. 166 с.

[8] *Миркин Б. Г.* Анализ качественных признаков и структур. — М.: Статистика, 1980. 319 с.

[9] http://www.bastion.ru/files/materials/analit/Ekonom/Komplex_ocenka_invest_Gazprom.doc.

References

[1] Kemeny, J., and J. Snell. 1963. *Mathematical models in the social sciences.* New York, NY: Blaisdell. 145 p.

[2] Litvak, B. G., 1982. *Expert information: Methods of acquisition and analysis.* Moscow: Radio i svyaz'. 184 p. (In Russian.)

[3] Mirkin, B. G. 1974. *The problem of group choice.* Moscow: Nauka. 256 p. (In Russian.)

[4] Dvoenko, S. D., and D. O. Pshenichny., 2013. On metric correction of matrices of pairwise comparisons. *JMLDA* 1(5):606–620. (In Russian.) Available at: http://jmlda.org/papers/doc/2013/no5/ (accessed November 3, 2015).

[5] Dvoenko, S. D., and D. O. Pshenichny. 2014. Optimal correction of metrical violations in matrices of pairwise comparisons. *JMLDA* 1(7):885–890. (In Russian.) Available at: http://jmlda.org/papers/doc/2014/no7/ (accessed November 3, 2015).

[6] Dvoenko, S. D., 2009. Clustering and separating of a set of members in terms of mutual distances and similarities. *Trans. Machine Learning Data Mining* 2(2):80–99.

[7] Mirkin, B. G. 1976. Analysis of qualitative attributes. Moscow: Statistics. 166 p. (In Russian.)

[8] Mirkin, B. G. 1980. Analysis of qualitative attributes and structures. Moscow: Statistics. 319 p. (In Russian.)

[9] http://www.bastion.ru/files/materials/analit/Ekonom/Komplex_ocenka_invest_Gazprom.doc.

Цензурирование ошибочно классифицированных объектов выборки*

И. А. Борисова[1,2,3]***, О. А. Кутненко***[1,2,3]
biamia@mail.ru
[1]Институт математики им. С. Л. Соболева СО РАН, Новосибирск; [2]Новосибирский государственный университет, Новосибирск; [3]Конструкторско-технологический институт вычислительной техники СО РАН, Новосибирск

Рассматривается задача цензурирования выборок, изначально содержащих значительное число неверно классифицированных объектов. Предложен алгоритм цензурирования, ориентированый только на локальные характеристики объектов выборки. Для оценки вероятности принадлежности объекта к одному из двух образов используется тернарная относительная мера — функция конкурентного сходства (function of rival simularity — FRiS-функция). В фиксированном признаковом пространстве цензурирование состоит в последовательном удалении объектов, максимально ухудшающих качество описания выборки (или оценку разделимости классов). Результаты тестирования алгоритма на широком спектре модельных задач позволили сделать вывод, что объекты, удаленные до точки перегиба функции, описывающей разделимость классов, как правило, являются выбросами, искажающими структуру данных.

Ключевые слова: *анализ данных; функция конкурентного сходства; компактность образов; разделимость классов; распознавание образов; цензурирование объектов*

Outliers detection in datasets with misclassified objects*

I. A. Borisova[1,2,3]***, and O. A. Kutnenko***[1,2,3]
[1]Sobolev Institute of Mathematics, SB RAS, Novosibirsk; [2]Novosibirsk State University, Novosibirsk; [3]Design Technological Institute of Digital Techniques, SB RAS, Novosibirsk

Background: The problem of outliers detection is one of the important problems in Data Mining. Here, outliers are considered as initially misclassified objects of the dataset. Such objects in small datasets can seriously interrupt the process of classification. This paper describes an algorithm of censoring such data, focusing only on the local characteristics of objects in the dataset.

Methods: Censoring procedure in a fixed feature space consists of sequential removals of objects, which deteriorate the quality of dataset description (a value of classes' separability) in the strongest way. This value depends on the number of objects in the dataset and similarity of objects with their class in competition with the rival class. To evaluate the similarity of the object z with class A in competition with class B, the ternary relative measure called the function of rival similarity (FRiS-function) is used.

Results: The proposed algorithm was tested on a wide range of model problems. Accuracy of k nearest neighbors classification before and after outliers elimination from the datasets was in use to estimate efficiency of the censoring algorithm. In the most tasks, it is appeared to be improvement in classification accuracy after censoring. Analysis of objects which were recognized as outliers showed up to 96% sensitivity and 99% specificity.

*Работа выполнена при финансовой поддержке РФФИ, проект № 14-01-00039.

Concluding Remarks: According to the obtained results, it is possible to conclude that the objects, which were deleted before the inflection point of the classes separability function, usually distort the structure of the data. Therefore, their exclusion from the analyzed dataset increases the reliability of recognition.
Keywords: *Data Mining; function of rival similarity; compactness; class separability; outliers detection; classification*

1 Введение

Одним из следствий бурного развития технологий в последние десятилетия явилось лавинообразное накопление информации, получаемой, обрабатываемой и сохраняемой с их помощью. Этот же факт приводит к тому, что в собранных таким образом данных фигурирует большое количество нерелевантной информации, шумов, ошибок, от которых эти данные необходимо очищать, прежде чем приступать к решению тех или иных задач Data Mining.

Задача цензурирования шумовых объектов, равно как и задача фильтрации нерелевантных признаков, давно и довольно успешно решаются в области анализа данных. Хороших результатов при решении задачи фильтрации нерелевантных признаков или ее аналога — задачи выбора информативных признаков (feature selection) — удалось достичь благодаря использованию функции конкурентного сходства (FRiS-функции) [1]. Это объясняется тем, что FRiS-функция оказалась достаточно простым и надежным способом оценивать вероятность ошибки распознавания каждого отдельного объекта. Переход от бинарного индикатора «есть ошибка – нет ошибки» к количественной величине «вероятность ошибки» позволил более точно оценивать компактность выборок в том или ином пространстве признаков и, как следствие, более качественно выбирать наиболее информативное подпространство признаков и отфильтровывать шумы.

Данное свойство FRiS-функции — оценивать вероятность ошибочного распознавания каждого конкретного объекта — также может оказаться полезным для фильтрации шумовых объектов при решении задачи цензурирования выбросов (outliers detection). Существуют различные интерпретации этой задачи. В одних источниках выбросами называют объекты, порожденные механизмами, отличными от механизмов порождения остальной выборки [2], в других — объекты, резко повышающие сложность модели, в третьих — ошибки измерения и ввода данных, которые оказываются далеко от всех типичных объектов выборки [3]. Соответственно, и подходы к решению задачи поиска выбросов различаются. Наиболее распространенным является статистический подход, при котором выбросы отыскиваются с помощью статистических тестов в предположении об известном законе распределения анализируемой выборки [4]. Но, как правило, распределения выборок в реальных прикладных задачах не вписываются в рамки стандартных моделей. В связи с этим все большую популярность набирает непараметрический подход, опирающийся на метрические характеристики выборки, при котором выбросы отыскиваются среди объектов или кластеров, удаленных от основной массы объектов [5]. Все эти подходы объединены установкой, что, во-первых, количество выбросов обычно незначительно в сравнении с объемом выборки, а во-вторых, выбросы приводят к переобучению алгоритмов распознавания и тем самым увеличивают вероятность ошибки.

В [6] была рассмотрена задача цензурирования периферийных объектов выборки, неоправданно повышающих сложность структуры данных в условиях, когда эта сложность оценивается величиной FRiS-компактности. Для этих целей сначала формировался

набор типичных объектов — столпов, отражающих структуру выборки, и затем с его помощью отфильтровывались выбросы.

В данной статье под задачей цензурирования понимается задача исключения из обучающей выборки ошибочно классифицированных объектов, которые оказались в ней на этапе сбора и сохранения данных. Если таких объектов достаточно много, а объем выборки невелик, то восстановление ее структуры оказывается серьезно затруднено. Для решения подобных задач предлагается осуществлять цензурирование на основе анализа локального окружения объектов. Данный подход опирается на гипотезу локальной компактности [7]. Оценивать количественные характеристики локальной компактности объектов в той или иной части выборки также предполагается с помощью функции конкурентного сходства.

2 FRiS-компактность и качество описания выборки

Гипотеза компактности является одной из важных концепций в анализе данных, а ее выполнение для объектов обучающей выборки — необходимым требованием для успешного решения задачи распознавания. Для получения количественной оценки компактности образов в фиксированном признаковом пространстве предлагается использовать FRiS-функцию, с помощью которой формализуется представление о компактности как о «высоком» сходстве объектов одного образа друг с другом и «низком» сходстве с объектами других образов.

Идея конкурентного сходства возникла в связи с желанием учитывать конкурентную ситуацию — контекст при оценке похожести объекта на другой объект или принадлежности объекта к образу. Данная концепция хорошо согласуется с человеческими особенностями оценки похожести объектов. Два объекта с несовпадающими свойствами могут считаться «сходными» или «не сходными», «близкими» или «далекими» в зависимости от свойств других объектов. Хорошо известная бытовая фраза «все познается в сравнении» на самом деле отражает фундаментальный закон познания. Адекватная мера сходства должна определять величину сходства, зависящую от особенностей конкурентного окружения объекта z. В распознавании образов сходство также является категорией не абсолютной, а относительной. При распознавании принадлежности объекта z к одному из двух образов A или B важно знать не только расстояние $r(z, A)$ до образа A, но и расстояние $r(z, B)$ до конкурирующего образа B.

Для вычисления конкурентного сходства объекта z с объектом a в конкуренции с объектом b предлагается использовать следующую величину:

$$F(z, a|b) = \frac{r(z, b) - r(z, a)}{r(z, b) + r(z, a)}.$$

По мере передвижения объекта z от объекта a к объекту b можно говорить вначале о большом сходстве объекта z с объектом a, об умеренном их сходстве, затем о наступлении одинакового сходства, равного 0, как с объектом a, так и с b. При дальнейшем продвижении z к b возникает умеренное, а затем и большое отличие z от a. Совпадение объекта z с объектом b означает максимальное отличие z от a, что соответствует сходству z с a, равному -1.

Сходство F между объектами не зависит от положения начала координат, поворота координатных осей и одновременного умножения их значений на одну и ту же величину. Но независимые изменения масштабов разных координат меняют вклад, вносимый отдельными характеристиками в оценку и расстояния, и сходства.

Конкурентное сходство объектов с образами будем определять по тому же принципу, что и конкурентное сходство объектов с объектами:

$$F(z, A|B) = \frac{r(z, B) - r(z, A)}{r(z, B) + r(z, A)}, \tag{1}$$

при этом расстояние от объекта z до образов A и B может вычисляться по-разному. В качестве него может использоваться и расстояние $r(z, a)$ до ближайшего объекта a образа A, и среднее расстояние до всех объектов образа, и среднее расстояние до k ближайших объектов образа, и расстояние до центра тяжести образа, и т. д. В дальнейшем в качестве расстояния от объекта до образа по умолчанию будет использоваться расстояние до ближайшего объекта этого образа.

Для произвольного объекта $z \in A$ мера конкурентного сходства этого объекта со своим образом в конкуренции с образом B показывает, насколько этот объект похож на представителей своего образа и не похож на представителей образа B, поэтому при решении задачи распознавания FRiS-функция может интерпретироваться как оценка вероятности принадлежности объекта z к образу A. Усредняя значения FRiS-функции из (1) по всем объектам образов A и B, можно оценить важную характеристику решаемой задачи распознавания — компактность образов, аналогами которой у других авторов [8] выступают такие понятия, как отделимость классов, сложность выборки, качество выборки и т. д.:

$$F_{AB} = \frac{\sum_{a \in A} F(a, A|B) + \sum_{b \in B} F(b, B|A)}{|A| + |B|}. \tag{2}$$

Варьируя способ вычисления расстояния от объекта до образа, с помощью (2) можно моделировать различные варианты компактности.

В [9] был описан один из таких вариантов получения количественной оценки компактности, который затем с успехом использовался при решении задачи выбора информативного набора признаков. Его особенность заключается в том, что вместо всех объектов выборки для вычисления компактности по формуле (2) используются только типичные представители образов — столпы.

Построение множества столпов, сохраняющего основные закономерности задачи, необходимые для хорошего распознавания как объектов исходной выборки, так и новых объектов, является одним из способов проявить особенности данных, перейти к их сжатому описанию, оценить сложность выборки. Чем сложнее структура образов, чем сильнее они пересекаются, тем больше столпов потребуется для описания таких данных. Для построения сжатого описания данных в виде системы столпов используется алгоритм FRiS-Stolp [10], который работает при любом соотношении количества объектов к количеству признаков и при произвольном виде распределения образов. Набор столпов считается достаточным для описания выборки, если сходство F всех объектов обучающей выборки с ближайшими своими столпами в конкуренции с ближайшими объектами других образов превышает пороговое значение F^*, например $F^* = 0$.

Чтобы вычислить величину FRiS-компактности образа A по множеству столпов S_A и S_B образов A и B, соответственно, используется следующая формула:

$$C_{A|B} = \frac{\sum_{a \in A} F(a, S_A|S_B) - |S_A|}{|S_A||A|}. \tag{3}$$

Здесь S_A и S_B — достаточный для описания выборки набор столпов.

Аналогично вычисляется величина $C_{B|A}$ FRiS-компактности образа B в конкуренции с A. Итоговая величина компактности образов A и B вычисляется как геометрическое усреднение величин $C_{A|B}$ и $C_{B|A}$:

$$C_{AB} = \sqrt{C_{A|B}C_{B|A}}\,. \tag{4}$$

Отметим, что количество столпов образа зависит от структуры распределения объектов и величины порога F^*: с ростом F^* увеличивается как число столпов, так и точность описания распределения, но растет и сложность его описания, т. е. множитель $1/S_A$ в (3) является штрафом за структурную сложность образа.

Безошибочное распознавание всех объектов обучающей выборки является неким гарантом того, что построенная система столпов сохраняет основные свойства выборки. Однако, как правило, требование безошибочного распознавания при решении задачи классификации приводит к чрезмерному усложнению решающих правил и, как следствие, к переобучению.

Для решения этой проблемы вводится понятие качества описания выборки набором столпов. Эта величина, с одной стороны, показывает, насколько рассматриваемый набор столпов отражает основные закономерные связи между описывающими характеристиками и целевым признаком, которые можно наблюдать на всей выборке, а с другой — регулирует количество столпов.

Чтобы оценить качество описания выборки набором столпов S_A и S_B образов A и B, используется следующая формула:

$$H(S_A, S_B) = \frac{\sum_{a\in A} F(a, S_A|B) + \sum_{b\in B} F(b, S_B|A)}{|S_A \cup S_B||A \cup B|}\,. \tag{5}$$

При изменении набора столпов меняется качество описания H обучающей выборки и ошибка распознавания E независимой тестовой выборки.

В [11] было экспериментально показано, что если наращивать число столпов, выбирая на роль каждого следующего столпа объект, обеспечивающий максимальный рост величины H в формуле (5), то между H и E имеется закономерная связь, используя которую можно найти количество столпов, не приводящее к переобучению. Примеры кривых H и E для различных модельных задач приводятся на рис. 1.

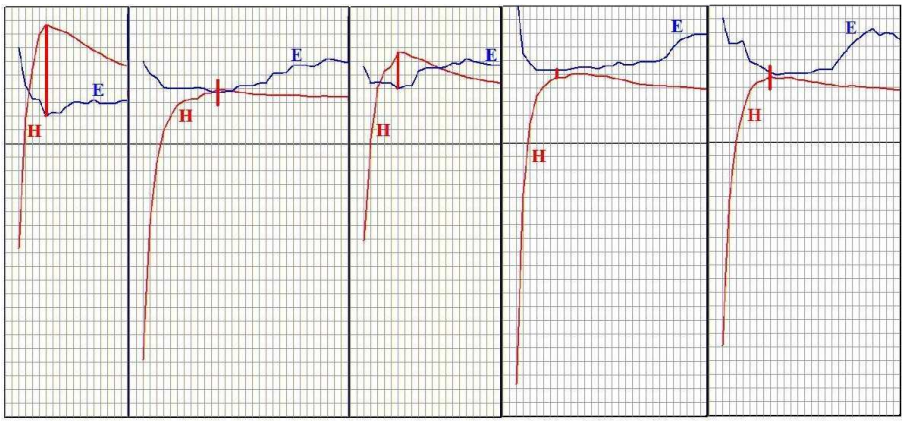


Рис. 1 Графики качества описания обучающей выборки (H) и графики ошибки распознавания (E) в зависимости от числа выбранных эталонов

3 Цензурирование выборки с опорой на столпы

При построении столпов наряду с объектами, хорошо отражающими структуру образов, принимают участие и шумящие объекты, и даже мелкие кластеры таких объектов, влияние которых было бы целесообразно исключить. Для их цензурирования можно применять описанный в [6] алгоритм, использующий в качестве критерия, управляющего процессом повышения компактности обучающей выборки, меру FRiS-компактности образов, вычисляемую по формуле (4), и включающий как составную часть алгоритм FRiS-Stolp.

Для регулирования доли цензурируемых объектов в рассмотрение вводится величина штрафа за исключение объектов из обучающей выборки $(M^*/M)^\gamma$, где M — размерность выборки; M^* — число объектов обучающей выборки, оставшихся после очередного этапа сокращения выборки; $\gamma \geqslant 0$ — параметр, регулирующий вклада штрафа в общую оценку разделимости. С учетом этого Q_{AB} — качество описания выборки после цензурирования образов на каждом шаге сокращения выборки — оценивается следующим образом:

$$Q_{AB} = \left(\frac{M^*}{M}\right)^\gamma C_{AB}\,. \tag{6}$$

Сначала алгоритмом FRiS-Stolp строится достаточный набор столпов, стоящих в центрах своих кластеров, и по формулам (3), (4), (6) вычисляется качество описания выборки Q_{AB}. Затем выбирается кластер, исключение которого обеспечивает максимальное значение Q_{AB} на оставшихся объектах. При этом могут исключаться только кластеры, содержащие не больше m^* объектов. Процесс построения столпов и исключения кластеров повторяется, пока либо доля удаленных объектов не превысит заданный порог, либо не будет кластеров, содержащих не больше m^* объектов.

По списку найденных оценок качества выборки выбирается вариант, соответствующий максимуму величины Q_{AB}. Набор столпов, который был зафиксирован при этом, служит основой решающего правила, используемого для распознавания контрольной выборки.

Алгоритм тестировался на модельных задачах распознавания двух образов. Эксперименты показали, что повышение компактности обучающей выборки более чем в 99% случаев приводит к повышению качества распознавания. Очищенная выборка описывается более простым решающим правилом, что повышает надежность распознавания контрольных объектов. Трудоемкость алгоритма зависит от исходной компактности образов — чем она выше, тем меньше времени требуется для выбора наилучшего варианта цензурирования.

4 Цензурирование выборки без построения столпов

В описанной выше задаче процедура цензурирования использовалась для упрощения структуры данных путем исключения из выборки непредставительных, периферийных объектов либо выбросов, попадающих в выборку с очень малой вероятностью. В этих условиях процедура построения столпов для заданной выборки оказывается устойчивой, а полученный набор столпов корректно описывает структурные особенности выборки.

Однако если рассматривать специфический случай задачи цензурирования — задачу классификации в условиях, когда обучающая выборка имеет малый объем и изначально содержит большое количество неверно классифицированных объектов, ситуация меняется. Алгоритм FRiS-Stolp, как и большинство алгоритмов классификации, строящих сложные решающие правила, не сможет корректно работать. Для цензурирования ошибочно классифицированных объектов в этом случае будет рассматриваться не требующий предварительного построения столпов алгоритм, который ориентируется только на локальные

характеристики объектов выборки. Отметим, что под цензурированием в данной задаче понимается исключение из обучающей выборки неверно классифицированных объектов, снижающих ожидаемую надежность распознавания новых объектов.

Для оценки меры разделимости (простоты) выборки G_{AB} в данной задаче будем использовать локальную компактность выборки, вычисленную по формуле (2) при условии, что в качестве расстояния от объекта до образа используется среднее расстояние до k ближайших объектов этого образа. Кроме того, так как с увеличением количества исключенных объектов повышается вероятность переобучения алгоритма, введем штраф M^*/M, регулирующий количество исключенных объектов. В результате локальная разделимость обучающей выборки будет вычисляться по следующей формуле:

$$G_{AB} = \frac{M^*}{M} F_{AB}\,.$$

Сам алгоритм цензурирования при этом будет следующим:

1. Вычисляется разделимость для всей выборки.

2. Отыскивается объект, удаление которого из выборки максимально повышает ее разделимость. Этот объект признается выбросом и исключается из выборки.

3. Процедура повторяется до момента, когда исключение любого объекта из обучающей выборки только ухудшает ее разделимость. Другими словами, процесс цензурирования продолжается до достижения точки перегиба функции разделимости.

5 Тестирование алгоритма. Результаты экспериментов

Для изучения возможности цензурирования ошибочно классифицированных объектов на основе анализа изменения локальной разделимости выборки был сгенерирован ряд модельных задач распознавания образов с распределениями разной степени сложности. Целевая характеристика для заданной доли объектов, обозначаемой в дальнейшем α, изменялась, тем самым в анализируемые выборки вводилась шумовая компонента, состоящая из неверно классифицированных объектов.

В качестве оценки эффективности предложенного алгоритма использовалась разница в величинах ожидаемой ошибки распознавания до и после цензурирования обучающей выборки. В качестве решающего правила во всех экспериментах использовалось правило k ближайших соседей, $k = 3$.

Для оценки того, как изменяется ошибка распознавания без цензурирования и с цензурированием в зависимости от доли ошибочно классифицированных объектов в анализируемой выборке, был проведен следующий эксперимент. Генерировалась серия из 100 выборок, с одними и теми же распределениями и задаваемой параметром α долей шумов. По каждой выборке методом Cross Validation оценивалась ожидаемая ошибка распознавания. Объем обучающей выборки составлял 100 объектов.

Результаты эксперимента приводятся на рис. 2. Здесь штриховой линией изображена зависимость ошибки распознавания от доли шумов в выборке без цензурирования, а сплошной линией — та же зависимость, но для отцензурированных выборок.

Для более глубокого изучения свойств предложенного алгоритма проводилось сравнение результатов распознавания тестовой выборки до и после цензурирования на серии из 10 задач, отличающихся сложностью и структурой, каждая из которых решалась 100 раз на разных обучающих выборках, т. е. общее количество экспериментов при различных численных реализациях данных было равно 1000. Уровень шума при этом

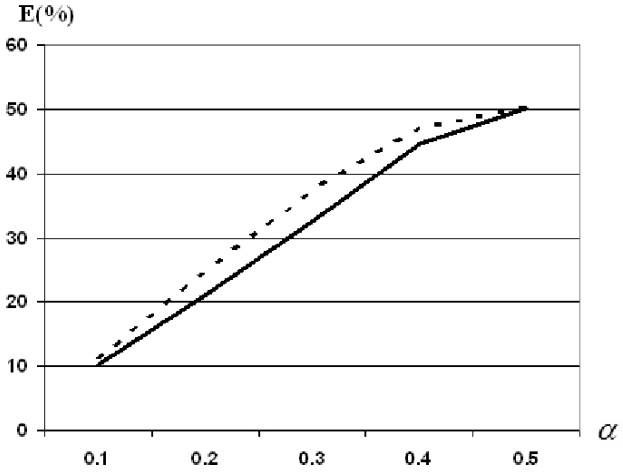


Рис. 2 Зависимость ошибки распознавания от α — доли неверно классифицированных объектов выборки до и после цензурирования

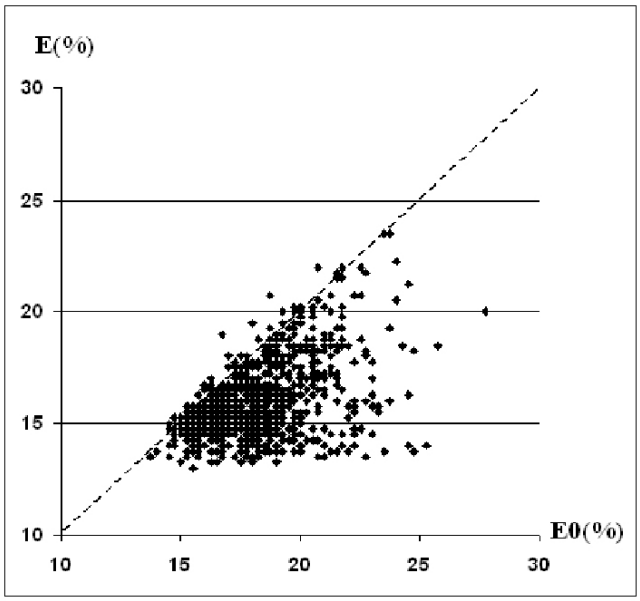


Рис. 3 Связь ошибки распознавания до ($E0$) и после цензурирования (E)

составлял 15%, объем обучающих выборок был 100 объектов. Результаты этого эксперимента приведены на рис. 3. Каждой точке соответствуют результаты распознавания по одной выборке, координата точки по оси абсцисс — это ошибка без цензурирования, по оси ординат — ошибка с цензурированием. Штриховая диагональ задает порог, при котором ошибка без цензурирования равна ошибке с цензурированием. Для экспериментов, результаты которых оказались ниже этого порога, надежность распознавания после цензурирования улучшилась.

Параллельно в этой серии задач отслеживалось количество шумовых объектов, которые реально удается отфильтровать в процессе цензуририрования. Оказалось, что в среднем чувствительность по отношению к шумам составила 96%, специфичность — 99%. Эти результаты позволяют сделать вывод о применимости предложенного подхода к решению задачи фильтрации неверно классифицированных объектов.

6 Заключение

В данной работе исследовалась возможность цензурирования ошибочно классифицированных объектов обучающей выборки для случая, когда доля таких объектов достаточно велика, а объем выборки ограничен. В этом случае цензурирование осуществляется путем снижения сложности выборки. Сложность при этом оценивается величиной локальной разделимости классов, которая вычисляется с помощью функции конкурентного сходства. Проведенные эксперименты на широком спектре модельных задач подтверждают работоспособность предложенного алгоритма цензурирования.

Литература

[1] *Zagoruiko N. G., Borisova I. A., Dyubanov V. V., Kutnenko O. A.* A quantitative measure of compactness and similarity in a competitive space // J. Appl. Ind. Math., 2011. Vol. 5. No. 1. P. 144–154.

[2] *Hawkins D.* Identification of outliers. — London, U.K.: Chapman and Hall, 1980.

[3] *Aggarwal C. C.* Outlier analysis. — Springer, 2013.

[4] *Barnett V., Lewis T.* Outliers in statistical data. — New York, NY, USA: John Wiley, 1994.

[5] *Knorr E., Ng R.* Algorithms for mining distance-based outliers in large datasets // 24th Conference (International) on Very Large Data Bases (VLDB) Proceedings, 1998. P. 392—403.

[6] *Загоруйко Н. Г., Кутненко О. А.* Цензурирование обучающей выборки // Вестник Томского государственного университета. Управление, вычислительная техника и информатика, 2013. № 1(22). С. 66–73.

[7] *Аркадьев А. Г., Браверман Э. М.* Обучение машины распознаванию образов. — М.: Наука, 1964.

[8] *Субботин С. А.* Комплекс характеристик и критериев сравнения обучающих выборок для решения задач диагностики и распознавания образов // Математичні машини і системи, 2010. № 1. С. 25–39.

[9] *Zagoruiko N. G., Borisova I. A., Dyubanov V. V., Kutnenko O. A.* A construction of a compressed description of data using a function of rival similarity // J. Appl. Ind. Math., 2013. Vol. 7. No. 2. P. 275–286.

[10] *Zagoruiko N. G., Borisova I. A., Dyubanov V. V., Kutnenko O. A.* Methods of recognition based on the function of rival similarity // Pattern Recognition Image Anal., 2008. Vol. 18. No. 1. P. 1–6.

[11] *Загоруйко Н. Г., Кутненко О. А., Зырянов А. О., Леванов Д. А. Обучение распознаванию без переобучения* // Машинное обучение и анализ данных, 2014. Т. 1. № 7. С. 891–901.

References

[1] Zagoruiko, N. G., I. A. Borisova, V. V. Dyubanov, and O. A. Kutnenko. 2011. A quantitative measure of compactness and similarity in a competitive space. *J. Appl. Ind. Math.* 5(1):144–154.

[2] Hawkins, D. 1980. *Identification of outliers*. London, U.K.: Chapman and Hall.

[3] Aggarwal, C. C. 2013. *Outlier analysis*. Springer.

[4] Barnett, V., and T. Lewis. 1994. *Outliers in statistical data*. New York, NY: John Wiley.

[5] Knorr, E., and R. Ng. 1998. Algorithms for mining distance-based outliers in large datasets. *24th Conference (International) on Very Large Data Bases (VLDB) Proceedings*. 392—403.

[6] Zagoruiko, N. G., and O. A. Kutnenko. 2013. Censoring of a train dataset. *Vectnik Tomskogo gosydarstvennogo yniversiteta. Upravlenie, vychislitel'naya tekhnika i informatika* 1(22):66–73.

[7] Arkad'ev, A. G., and E. M. Braverman. 1964. *Machine learning to pattern recognition*. Moscow: Nauka.

[8] Subbotin, S. A. 2010. Complex characterization and comparison criteria of training samples for diagnostics and pattern recognition. *Matematichni mashini i sistemi* 1:25–39.

[9] Zagoruiko, N. G., I. A. Borisova, V. V. Dyubanov, and O. A. Kutnenko. 2013. A construction of a compressed description of data using a function of rival similarity. *J. Appl. Ind. Math.* 7(2):275–286.

[10] Zagoruiko, N. G., I. A. Borisova, V. V. Dyubanov, and O. A. Kutnenko. 2008. Methods of recognition based on the function of rival similarity. *Pattern Recognition Image Anal.* 18(1):1–6.

[11] Zagoruiko, N. G., O. A. Kutnenko, A. O. Ziranov, and D. A. Levanov. 2014. Learning to recognition without overfittinig. *Mashinnoe obychenie i analiz dannykh* 1(7):891–901. (In Russian.)