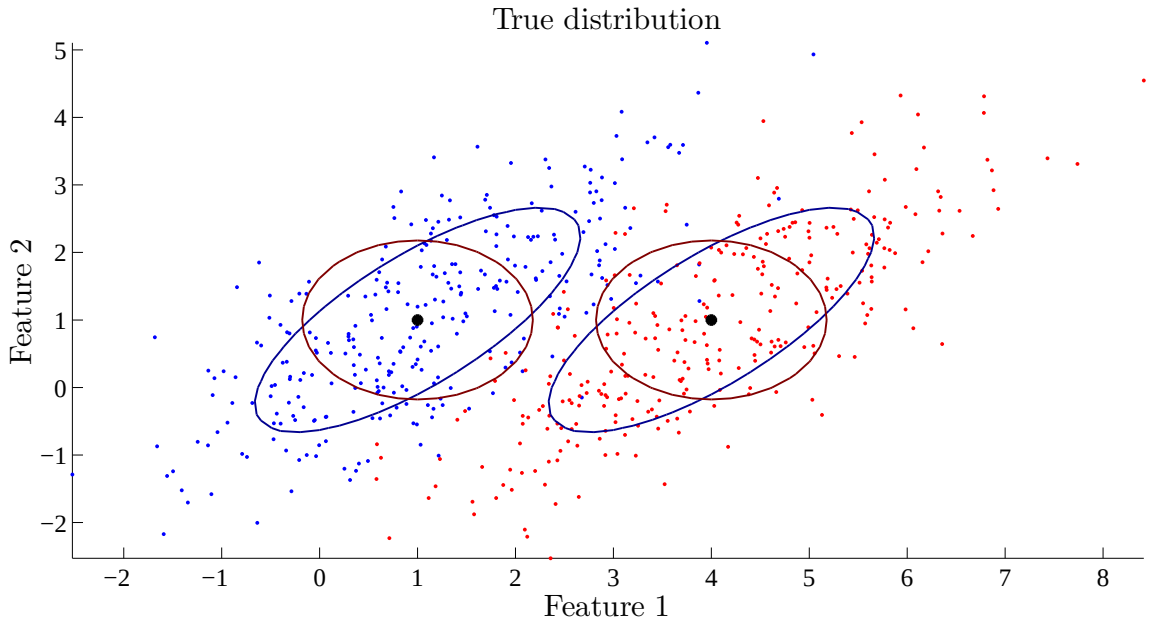
ISSN 2223-3792

Машинное обучение и анализ данных

2016 год **Том 2, номер 1**



Машинное обучение и анализ данных

Журнал «Машинное обучение и анализ данных» публикует новые теоретические и обзорные статьи с результатами научных исследований в области искусственного интеллекта, теоретических основ информатики и 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. Журнал зарегистрирован в системе Crossref, doi `http://dx.doi.org/10.21469/22233792`.

- Новостной сайт http://jmlda.org/
- Электронная система подачи статей http://jmlda.org/papers/
- Правила подготовки статей http://jmlda.org/papers/doc/authors-guide.pdf

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

Вычислительный центр им. А. А. Дородницына ФИЦ ИУ РАН
Московский физико-технический институт
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

Москва, 2016

Journal of Machine Learning and Data Analysis

The journal Machine Learning and Data Analysis publishes original research papers and reviews of the developments in the field of artificial intelligence, theoretical computer science and its applications. The journal aims to promote the theory of machine learning and data mining and methods of conducting computational experiments. Papers are accepted in English and Russian.

The journal is included in the Russian science citation index RSCI. Information about citation to articles can be found at the Russian science citation index website. ISSN 2223-3792. Mass media registration certificate ЭЛ № ФС 77-55486. The Crossref journal doi is http://dx.doi.org/10.21469/22233792.

- Journal news and archive http://jmlda.org/
- Open journal system for papers submission http://jmlda.org/papers/
- Style guide for authors http://jmlda.org/papers/doc/authors-guide.pdf

The scope of the journal:
— classification, clustering, regression analysis;
— multidimensional statistical analysis;
— Bayesian methods for regression and classification;
— model selection and complexity;
— deep learning;
— Statistical Learning Theory;
— time series forecasting techniques;
— methods of signal processing and speech recognition;
— optimization methods for solving machine learning and data mining problems;
— methods of big data analysis;
— data visualization techniques;
— methods of image processing and recognition;
— text analysis, text mining and information retrieval;
— applied data analysis problems.

Dorodnicyn Computing Centre FRC CSC RAS
Moscow Institute of Physics and Technology
Department of Control and Applied Mathematics
Division "Intelligent Systems"

Moscow, 2016

Содержание

Contents

Алгоритм приближенного поиска ближайшего цифрового массива в иерархически структурированном наборе данных*

М. М. Ланге, С. Н. Ганебных, А. М. Ланге
lange_mm@ccas.ru, sng@ccas.ru, lange_am@mail.ru
ФИЦ «Информатика и управление» РАН, г. Москва, ул. Вавилова, 44/2

Предлагается алгоритм быстрого приближенного поиска в заданном наборе многомерных цифровых массивов ближайшего соседа к предъявляемому массиву. Дефект приближенного поиска определяется отношением разности расстояний от предъявляемого массива до реально найденного и до ближайшего соседа к расстоянию до ближайшего соседа. Алгоритм использует пирамидальные представления массивов с многоуровневым разрешением и стратегию иерархического поиска. При большом линейном размере массивов и большой мощности набора данных получена асимптотическая оценка вычислительного выигрыша алгоритма приближенного поиска относительно алгоритма точного поиска. Для набора изображений рукописных цифр из базы данных MNIST построены экспериментальные оценки среднего дефекта поиска, стандартного отклонения дефектов поиска и вычислительной сложности алгоритма при различных значениях параметра стратегии поиска. Используя полученные оценки, построена зависимость среднего дефекта поиска от вычислительной сложности алгоритма.

Ключевые слова: *многомерный массив; набор данных; ближайший сосед; пирамидальное представление; приближенный поиск; дефект поиска; вычислительная сложность*

DOI: 10.21469/22233792.2.1.01

1 Введение

Задача поиска в заданном наборе векторов евклидова пространства представителя, достаточно близкого к предъявляемому вектору, известна как задача приближенного поиска ближайшего соседа [1–5]. В пространстве фиксированной размерности $d \geqslant 1$ построены алгоритмы, реализующие поиск в наборе из n векторов представителя на расстоянии $D \leqslant (1+\varepsilon)D_{\min}$ от предъявляемого вектора, где $D_{\min}$ — расстояние до ближайшего соседа, а $\varepsilon > 0$ — допустимое отклонение. Алгоритмы с логарифмической сложностью используют древовидные структуры данных и при больших значениях n и фиксированных d и ε имеют вычислительную сложность $O(\log n)$ [1, 4]. Для сравнения алгоритм полного перебора, реализующий поиск ближайшего представителя, имеет сложность $\Theta(dn)$, и при больших значениях n доля сложности алгоритма приближенного поиска относительно сложности переборного алгоритма составляет $O(n^{-1}\log n)$.

Как правило, мультипликативный коэффициент в оценках сложности известных приближенных алгоритмов растет экспоненциально с увеличением d и по степенному закону с уменьшением ε. Явная зависимость сложности от указанных параметров дана в оценке $O(d\lceil 1+6d/\varepsilon\rceil^d \log n)$, полученной на решающем BBD-дереве (Balanced Box-Decomposition tree) [4]. Характер зависимости сложности от размерности d и допустимого отклонения ε от ближайшего соседа ограничивает применение такого алгоритма для поиска массивов размерности $d = N^m$ с параметрами $N \geqslant 10$, $m \geqslant 1$ и, в частности, для изображений

*Работа выполнена при частичной финансовой поддержке РФФИ, проекты № 15-01-04671 и № 15-07-07516.

большого размера. На практике известные алгоритмы эффективны в пространстве малой размерности ($d \leqslant 8$) и не обеспечивают быстрого приближенного поиска цифровых массивов типа изображений размера 1024×1024, для которых $d > 10^6$.

В настоящей работе предложен иерархический алгоритм приближенного поиска на множестве многомерных цифровых массивов большого размера ближайшего представителя к предъявляемому массиву. Алгоритм построен в пространстве пирамидальных представлений массивов с многоуровневым разрешением [6]. Такие представления дают описания цифровых массивов в форме деревьев, индекс ветвления которых определяется параметром размерности m [7, 8]. Основой предлагаемого алгоритма является процедура приближенного поиска ближайшего соседа к предъявляемому объекту в многоуровневой сети эталонов, которая разработана для быстрого распознавания образов в пространстве древовидно-структурированных представлений с многоуровневым разрешением [9]. Получена оценка вычислительной сложности алгоритма и проведена его апробация на множестве изображений рукописных цифр [10]. По результатам апробации построены экспериментальные оценки среднего значения и дисперсии величины $(D - D_{\min})/D_{\min}$ (по множеству предъявляемых изображений) при различных значениях параметра алгоритма поиска.

2 Формализация задачи

Рассматривается источник, порождающий множество массивов X^m, в котором любой массив $x^m \in \mathrm{X}^m$ задан m-мерным кубом ($m \geqslant 1$), содержащим N^m элементов из алфавита $A = \{0, 1, \ldots, q-1\}$ ($q \geqslant 2$). Допустимыми считаются массивы, средние значения элементов которых положительны. Предполагается, что $N = 2^L$, где $L \gg 1$ и любой допустимый массив $x^m \in \mathrm{X}^m$ имеет набор описаний

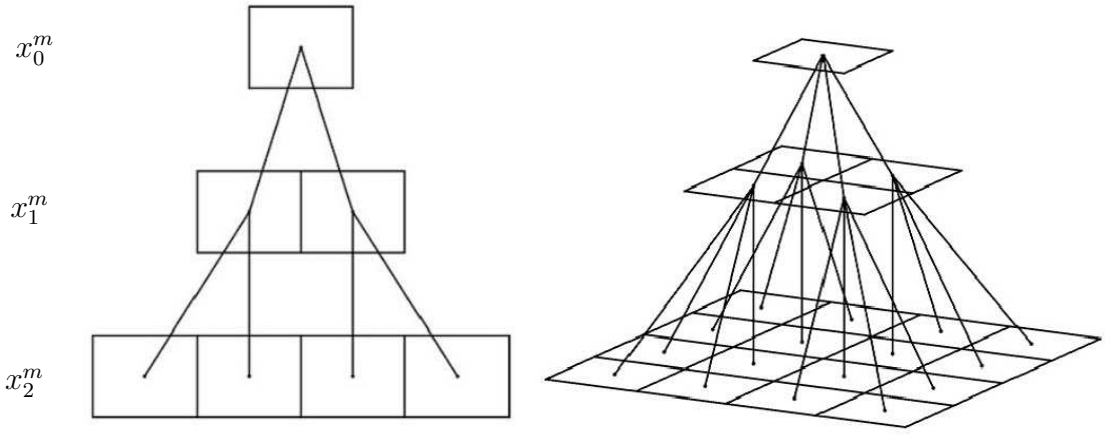
$$\mathbf{x}_L^m = (x_0^m, \ldots, x_l^m, \ldots, x_L^m), \tag{1}$$

образующих 2^m-пирамиду [6] высоты $L = \log_2 N$, в которой описание l-го уровня x_l^m является m-мерным кубом объема 2^{lm}. В случае $m = 1, 2, 3, \ldots$ набор описаний (1) образует соответственно бинарную, квадро- и октопирамиду. Уровни пирамиды строятся рекурсивно: каждый элемент в описании x_l^m вычисляется как среднее значение по 2^m смежным элементам в описании x_{l+1}^m. Основание пирамиды x_L^m совпадает с исходным массивом x^m, вершина x_0^m представлена средним значением элементов массива x^m. Примеры представления одномерного ($m = 1$) и двумерного ($m = 2$) массивов соответственно в форме бинарной пирамиды и квадропирамиды высоты $L = 2$ даны на рис. 1. Бинарная пирамида дает многоуровневое представление последовательности элементов длины N, квадропирамида является многоуровневым представлением изображения размера $N \times N$.

Деление значений элементов в описаниях $x_l^m, l = 1, \ldots, L$, из (1) на значение элемента вершины x_0^m (для допустимых массивов среднее значение элементов больше нуля) дает нормализованное представление

$$\mathbf{y}_L^m = (y_1^m, \ldots, y_l^m, \ldots, y_L^m) \tag{2}$$

в виде последовательности L описаний массива x^m с нарастающим разрешением (числом элементов). Нормализация описаний в представлении (2) обеспечивает их слабую зависимость от размера q используемого алфавита A. Элементы каждого описания y_l^m в представлении (2) снабжены векторами индексов $\mathbf{k}_l^m = (k_{l1}, \ldots, k_{lm})$, где каждый индекс является номером элемента по соответствующей координате m-мерного куба с ребром 2^l и принимает одно из целочисленных значений $1, \ldots, 2^l$. Нормализованные представления (2) образуют множество $\mathbf{Y}_L^m : \mathrm{X}^m \to \mathbf{Y}_L^m$.



(*а*) $m = 1$ (*б*) $m = 2$

Рис. 1 Примеры пирамидальных представлений одномерного и двумерного массивов

Для любой пары m-мерных массивов $x^m \in \mathrm{X}^m$ и $\hat{x}^m \in \mathrm{X}^m$, имеющих нормализованные представления $\mathbf{y}_L^m \in \mathbf{Y}_L^m$ и $\hat{\mathbf{y}}_L^m \in \mathbf{Y}_L^m$ вида (2), описания l-го уровня образованы m-мерными кубами $y_l^m = \{z_{\mathbf{k}_l^m}\} \in \mathbf{y}_L^m$ и $\hat{y}_l^m = \{\hat{z}_{\mathbf{k}_l^m}\} \in \hat{\mathbf{y}}_L^m$, в которых элементы с одинаковыми векторами индексов являются соответственными. Используя соответствие элементов в нормализованных описаниях любой пары массивов, определим для пары массивов $x^m \in \mathrm{X}^m$ и $\hat{x}^m \in \mathrm{X}^m$ меру их различия l-го порядка:

$$D_l(x^m, \hat{x}^m) = 2^{-lm} \sum_{\mathbf{k}_l^m} \left| z_{\mathbf{k}_l^m} - \hat{z}_{\mathbf{k}_l^m} \right| = 2^{-lm} \sum_{k_{l1}=1}^{2^l} \cdots \sum_{k_{lm}=1}^{2^l} \left| z_{k_{l1},\ldots,k_{lm}} - \hat{z}_{k_{l1},\ldots,k_{lm}} \right| . \tag{3}$$

Суммирование мер $D_i(x^m, \hat{x}^m)$, $i = 1, \ldots, l$, вида (3) с весовыми коэффициентами $w_i > 0$ дает взвешенную меру l-го порядка:

$$D_l^w(x^m, \hat{x}^m) = \sum_{i=1}^{l} w_i D_i(x^m, \hat{x}^m) . \tag{4}$$

В качестве весовых коэффициентов в (4) выбираются энтропии уровней пирамиды (2):

$$w_i = \log_2 2^{im} = im. \tag{5}$$

Соотношения (3)–(5) порождают последовательность взвешенных мер различия массивов множества X^m:

$$D_l^w(x^m, \hat{x}^m) , \quad l = 1, \ldots, L , \tag{6}$$

которые определены на множестве нормализованных представлений $\mathbf{Y}_L^m$.

Пусть подмножество массивов $\hat{\mathrm{X}}^m \subset \mathrm{X}^m$ мощности $\|\hat{\mathrm{X}}^m\| = n$ образует набор данных, в котором производится точный или приближенный поиск ближайшего соседа для всех массивов из подмножества $\mathrm{X}^m \setminus \hat{\mathrm{X}}^m$. Для любого предъявляемого массива $x^m \in \mathrm{X}^m \setminus \hat{\mathrm{X}}^m$ решение принимается по мере (6) наибольшего порядка L на наборе $\hat{\mathrm{X}}_*^m \subseteq \hat{\mathrm{X}}^m$, который выбирается согласно принятой стратегии поиска и в общем случае зависит от предъявляемого массива. Решающее правило определяется функцией:

$$\hat{x}_*^m = \arg \min_{\hat{x}^m \in \hat{\mathrm{X}}_*^m} D_L^w(x^m, \hat{x}^m) . \tag{7}$$

Стратегия выбора набора $\hat{\mathrm{X}}^m_*$ в (7) определяет решающий алгоритм, который в случае $\hat{\mathrm{X}}^m_* = \hat{\mathrm{X}}^m$ реализует точный поиск ближайшего массива $\hat{x}^m_* \in \hat{\mathrm{X}}^m$ на основе полного перебора, а в случае $\hat{\mathrm{X}}^m_* \subset \hat{\mathrm{X}}^m$ — приближенный поиск на основе направленного (иерархического) перебора, при котором решение $\hat{x}^m_* \in \hat{\mathrm{X}}^m_*$ совпадает или отличается от ближайшего представителя в $\hat{\mathrm{X}}^m$.

Для выбора набора $\hat{\mathrm{X}}^m_*$ предлагается использовать параметрическую стратегию с параметром $n^* = 1, 2, \ldots, n$, значение которого определяет мощность этого набора $\hat{\mathrm{X}}^m_* : \|\hat{\mathrm{X}}^m_*\| = n^* \leqslant n$. Такая стратегия порождает семейство решающих алгоритмов по критерию (7), включающее алгоритмы приближенного поиска с параметром $n^* < n$ и алгоритм точного поиска с параметром $n^* = n$. Качество алгоритма с параметром n^* на множестве предъявляемых массивов $\mathrm{X}^m \setminus \hat{\mathrm{X}}^m$ определяется средним дефектом поиска

$$\varepsilon_{n^*} = \frac{1}{\|\mathrm{X}^m \setminus \hat{\mathrm{X}}^m\|} \sum_{x^m \in \mathrm{X}^m \setminus \hat{\mathrm{X}}^m} \left(\frac{\min\limits_{\hat{x}^m \in \hat{\mathrm{X}}^m_*} D^w_L(x^m, \hat{x}^m)}{\min\limits_{\hat{x}^m \in \hat{\mathrm{X}}^m} D^w_L(x^m, \hat{x}^m)} - 1 \right) \tag{8}$$

и стандартным отклонением

$$\sigma_{n^*} = \left(\frac{1}{\|\mathrm{X}^m \setminus \hat{\mathrm{X}}^m\|} \sum_{x^m \in \mathrm{X}^m \setminus \hat{\mathrm{X}}^m} \left(\frac{\min\limits_{\hat{x}^m \in \hat{\mathrm{X}}^m_*} D^w_L(x^m, \hat{x}^m)}{\min\limits_{\hat{x}^m \in \hat{\mathrm{X}}^m} D^w_L(x^m, \hat{x}^m)} - 1 \right)^2 - \varepsilon^2_{n^*} \right)^{1/2} \tag{9}$$

дефектов относительно среднего значения (8). Очевидно, что $\varepsilon_{n^*} \geqslant 0$ и $\sigma_{n^*} \geqslant 0$, причем нулевые значения этих статистик достигаются в случае $\hat{\mathrm{X}}^m_* = \hat{\mathrm{X}}^m$ при всех $x^m \in \mathrm{X}^M \setminus \hat{\mathrm{X}}^m$. Вычислительная сложность решающего алгоритма с параметром n^* определяется количеством элементарных операций C_{n^*}, требуемых для поиска решения (7). Рассматривается стратегия выбора набора $\hat{\mathrm{X}}^m_*$ в (7), которая обеспечивает невозрастающие значения ε_{n^*} и σ_{n^*} и неубывающие значения C_{n^*} с ростом n^*.

Решаемая задача заключается в получении оценок характеристик ε_{n^*}, σ_{n^*} и C_{n^*} как функций параметра n^*. Строятся асимптотические оценки вычислительной сложности решающих алгоритмов для источника с параметрами $m \geqslant 1$, $N \to \infty$, $n \to \infty$. Демонстрируется область значений параметра n^*, которая с ростом N обеспечивает стремление к нулю доли сложности алгоритмов приближенного поиска относительно сложности алгоритма точного поиска. Для набора полутоновых изображений рукописных цифр с параметрами $N = 32$, $m = 2$, $n = 50\,000$ строятся экспериментальные оценки ε_{n^*}, σ_{n^*} и C_{n^*} как функции переменной n^*/n на отрезке $[1/n, 1]$. Используя для указанного источника изображений оценки функций ε_{n^*} и C_{n^*} и исключая параметр n^* из условия $C_{n^*} = C^*$ ($C^* > 0$ — заданная допустимая сложность), для заданного набора изображений вычисляется функция «дефект–сложность»:

$$\varepsilon(C^*) = \varepsilon_{n^*} : n^* = \arg(C_{n^*} = C^*)\,. \tag{10}$$

3 Структура набора данных и алгоритм поиска решения

Будем считать, что каждый массив $\hat{x}^m \in \hat{\mathrm{X}}^m$ имеет нормализованное пирамидальное представление $\hat{\mathbf{y}}^m_L$ вида (2). Подмножество таких представлений $\hat{\mathbf{Y}}^m_L : \hat{\mathrm{X}}^m \to \hat{\mathbf{Y}}^m_L$ образует многоуровневую сеть представлений данных

$$\hat{\mathbf{Y}}^m_1, \ldots, \hat{\mathbf{Y}}^m_l, \ldots, \hat{\mathbf{Y}}^m_L, \tag{11}$$

в которой $\hat{\mathbf{Y}}_l^m$ — подмножество представлений $\mathbf{y}_l^m = (y_1^m, \ldots, y_l^m)$, заданных l уровнями нормализованной пирамиды (2). Каждое подмножество в последовательности (11) содержит представления всех массивов из набора $\hat{\mathrm{X}}^m$ и, следовательно, имеет мощность n.

Предлагаемая параметрическая стратегия поиска решения (7) в сети представлений (11) базируется на последовательном сужении зоны поиска на уровнях $l = 1, \ldots, L$ в соответствии с экспоненциальной функцией

$$n_l = \lfloor n 2^{-\alpha m (l-1)} \rfloor, \quad l = 1, \ldots, L\,, \tag{12}$$

с коэффициентом $\alpha = (L-1)^{-1} \log_{2^m}(n/n^*)$, где $n^* = 1, 2, \ldots, n$ — свободный параметр, определяющий мощность набора $\hat{\mathrm{X}}_*^m$ в (7). Значения функции (12) соответствуют количествам массивов, среди которых выполняется поиск на соответствующих уровнях сети (11).

Алгоритм поиска. Для любого предъявляемого массива $x^m \in \mathrm{X}^m$ на последовательных уровнях $l = 1, \ldots, L-1$ сети (11) вычисляются значения взвешенной меры различия $D_l^w(x^m, \hat{x}^m)$ вида (4) по n_l массивам $\hat{x}^m \in \hat{\mathrm{X}}^m$ и среди них отбираются n_{l+1} массивов с наименьшими значениями меры $D_l^w(x^m, \hat{x}^m)$; на уровне $l = L$ среди $n_L = n^*$ массивов отбирается ближайший массив ($n_{L+1} = 1$) с наименьшим значением меры $D_L^w(x^m, \hat{x}^m)$, который дает решение (7).

В случае $1 \leqslant n^* < n$ ($\alpha > 0$) стратегия (12) порождает иерархический алгоритм приближенного поиска ближайшего представителя в наборе данных $\hat{\mathrm{X}}^m$, а в случае $n^* = n$ ($\alpha = 0$) — переборный алгоритм точного поиска. Схемы поиска приближенного и точного решений с помощью указанных алгоритмов даны на рис. 2. В обоих случаях вычисление меры производится рекурсивно с использованием соотношения

$$D_l^w(x^m, \hat{x}^m) = D_{l-1}^w(x^m, \hat{x}^m) + w_l D_l(x^m, \hat{x}^m), \quad l = 1, \ldots, L\,, \tag{13}$$

и начального условия $D_0^w(x^m, \hat{x}^m) = 0$. Мера (13) вычисляется для n_l массивов из набора $\hat{\mathrm{X}}^m$, отбираемых на последовательных уровнях сети (11) согласно (12). В случае приближенного поиска: $n^* \leqslant n_l \leqslant n$, $l = 1, \ldots, L$; в случае точного поиска: $n_l = n$, $l = 1, \ldots, L$.

Вычислительная сложность решающих алгоритмов определяется числом элементарных операций, затрачиваемых на вычисление меры на всех уровнях сети (11), и на сортировку значений меры на последовательных уровнях для отбора ближайших массивов согласно стратегии (12), включая отбор решения на последнем уровне. Элементарной операцией вычисления меры является сравнение пары соответственных элементов в представлениях $\mathbf{y}_l^m$ и $\hat{\mathbf{y}}_l^m$, $l = 1, \ldots, L$, сравниваемых массивов, а элементарной операцией сортировки — сравнение пары значений вычисленной меры на заданном уровне сети (11).

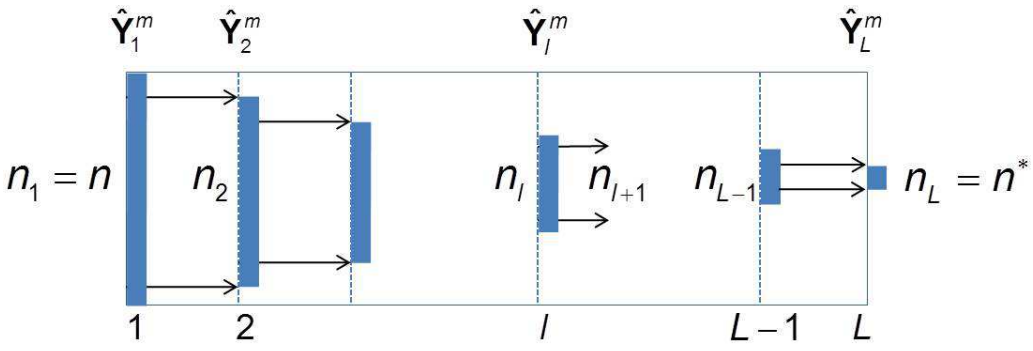


Рис. 2 Схемы поиска приближенного ($n^* < n$) и точного ($n^* = n$) решений

Поскольку число элементов в описаниях l-го уровня равно 2^{ml}, то сложность вычисления меры различия предъявляемого массива с массивами набора данных, отбираемыми согласно стратегии (12), равна

$$C_{n^*}^{\mathrm{msr}} = \sum_{l=1}^{L} n_l 2^{ml} = \sum_{l=1}^{L} \left\lfloor n \left(\frac{n^*}{n} \right)^{(l-1)/(L-1)} \right\rfloor 2^{ml}. \tag{14}$$

Отбор n_{l+1} наименьших значений меры из n_l на уровнях с номерами $l = 1, \ldots, L$ ($n_{L+1} = 1$ соответствует решению) может быть выполнен путем вычисления соответствующей порядковой статистики, что эквивалентно частичной сортировке со сложностью $O(n_l)$ [11]. Поскольку $n_l \leqslant n$ при всех $1 \leqslant l \leqslant L$, то оценка сложности частичной сортировки в решающем алгоритме с параметром $n^* \leqslant n$ имеет вид:

$$C_{n^*}^{\mathrm{srt}} = \begin{cases} O\left(\sum_{l=1}^{L} n_l\right) = O(nL), & n^* < n\,; \\ n-1, & n^* = n\,. \end{cases} \tag{15}$$

В случае $n^* = n$ на уровнях с номерами $l = 1, \ldots, L-1$ отбираются все массивы набора данных, а $(n-1)$ сравнений затрачивается на поиск решения на L-м уровне, что эквивалентно поиску первой порядковой статистики на полном множестве (мощности n) значений меры.

Соотношения (14) и (15) дают асимптотические при $n \to \infty$ оценки вычислительной сложности алгоритма точного поиска с параметром $n^* = n$ ($\alpha = 0$) и алгоритма приближенного поиска с параметром $n^* \leqslant n2^m/N^m$ ($\alpha \geqslant 1$) при $m \geqslant 1$ и $N^m \geqslant \log_q n$. Эти оценки имеют следующий вид:

$$C_{n^* \leqslant n2^m/N^m} = C_{n^* \leqslant n2^m/N^m}^{\mathrm{msr}} + C_{n^* \leqslant n2^m/N^m}^{\mathrm{srt}} \leqslant n2^m L + O(nL) = O(n \log N)\,; \tag{16}$$

$$C_{n^*=n} = C_{n^*=n}^{\mathrm{msr}} + C_{n^*=n}^{\mathrm{srt}} = \frac{2^m}{2^m - 1}(N^m - 1)n + (n-1) = \Omega(nN^m)\,. \tag{17}$$

Из оценок (16) и (17) следует

Утверждение. Доля сложности алгоритма приближенного поиска решения с параметром $n^* \leqslant n2^m/N^m$ относительно сложности алгоритма точного поиска решения с параметром $n^* = n$ удовлетворяет оценке

$$\frac{C_{n^* \leqslant n2^m/N^m}}{C_{n^*=n}} = O\left(\frac{\log N}{N^m}\right)$$

при $m \geqslant 1$, $N^m \geqslant \log_q n$ и $n \to \infty$.

4 Экспериментальные результаты

В данном разделе представлены экспериментальные зависимости показателей качества приближенного поиска ε_{n^*} и σ_{n^*}, определенные соотношениями (8) и (9), и зависимость отношения сложностей $C_{n^*<n}/C_{n^*=n}$ алгоритмов приближенного и точного поиска от величины n/n^*. Указанные зависимости получены для набора полутоновых изображений рукописных цифр из базы данных MNIST [10]. Вычислительный эксперимент выполнен с помощью кода, написанного на языке MATLAB [12]. Параметры изображений: $m = 2$, $N = 32$, $q = 256$; число уровней сети представлений изображений $L = \log_2 N = 5$; мощность

Таблица 1 Оценки качества и сложности поиска

$\log_2(n/n^*)$	$C^{\mathrm{msr}}_{n^*}/C_n$	$C^{\mathrm{srt}}_{n^*}/C_n$	C_{n^*}/C_n	ε_{n^*}	σ_{n^*}
0	0,9993	0,0007	1,0000	0	0
1	0,5325	0,0356	0,5681	0	0
2	0,2885	0,0285	0,3170	0	0
3	0,1597	0,0237	0,1834	0	0
4	0,0908	0,0205	0,1113	2E-6	0,0002
5	0,0535	0,0282	0,0717	3E-6	0,0002
6	0,0329	0,0166	0,0496	4E-6	0.0013
7	0,0214	0,0154	0,0368	8E-6	0,0021
8	0,0146	0,0146	0,0292	0,0004	0,0065
9	0,0107	0,0139	0,0246	0,0007	0,0085

базы $\|\mathrm{X}^m\| = 70\,000$. Цифры на изображениях базы нормированы по размеру и центрированы в поле изображения. В качестве набора данных $\hat{\mathrm{X}}^m$ использован обучающий набор (train set) мощности 60 000; в качестве предъявляемого набора $\mathrm{X}^m \setminus \hat{\mathrm{X}}^m$ — тестовый набор (test set) мощности 10 000. Вычисление характеристик ε_{n^*}, σ_{n^*} и C_{n^*}/C_n выполнено для значений $n/n^* = 2^k$, $k = 0, 1, \ldots, 10$, обеспечивающих коэффициент сужения зоны поиска $\alpha = (\log_{2^m}(n/n^*))/(L-1) = k/(m\log_2(N/2))$ в диапазоне значений $0 \leqslant \alpha \leqslant 5/4$. Следует отметить, что при $k = 0$ ($n^* = n$) алгоритм поиска дает точное решение, а при $k > 0$ ($n^* < n$) — приближенное решение.

Численная оценка вычислительной сложности решающего алгоритма с параметром $n^* < n$ получена с использованием процедуры быстрой сортировки вставками [11], которая на наборе из n элементов имеет среднюю вычислительную сложность $n \log n$. С учетом затрат на вычисление меры и затрат на сортировку значений меры на последовательных уровнях сети представлений данных (11) оценка сложности решающего алгоритма с параметром n^* определяется суммой

$$C_{n^*} = C^{\mathrm{msr}}_{n^*} + C^{\mathrm{srt}}_{n^*}\,, \tag{18}$$

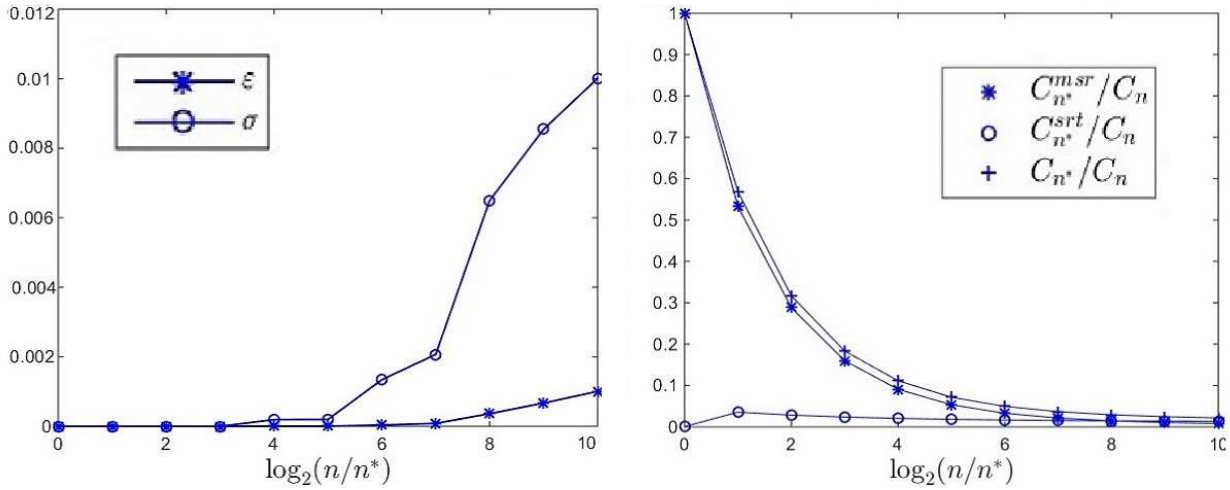
где

$$C^{\mathrm{msr}}_{n^*} = \sum_{l=1}^{L} n_l 2^{ml}\,;\;\; C^{\mathrm{srt}}_{n^*} = (n-1)[n^* = n] + \left((n^*-1) + \sum_{l=1}^{L-1} n_l \log n_l\right)[n^* < n]\,; \tag{19}$$

$[f]$ — индикатор f. В случае $n^* = n$ формулы (18) и (19) дают сложность поиска точного решения, а в случае $n^* < n$ — сложность поиска приближенного решения. Формулы (18) и (19) использованы для вычисления отношения C_{n^*}/C_n при значениях $n/n^* = 2^k$, $k = 0, 1, \ldots, 10$. Экспериментальные оценки характеристик качества поиска ε_{n^*} и σ_{n^*} и численные оценки долей сложности $C^{\mathrm{msr}}_{n^*}/C_n$, $C^{\mathrm{srt}}_{n^*}/C_n$, C_{n^*}/C_n представлены в таблице 1.

Построенные по данным таблицы графики зависимостей ε_{n^*} и σ_{n^*} от $\log_2(n/n^*)$ представлены на рис. 3, *а*, а графики зависимостей $C^{\mathrm{msr}}_{n^*}/C_n$, $C^{\mathrm{srt}}_{n^*}/C_n$ и C_{n^*}/C_n от $\log_2(n/n^*)$ — на рис. 3, *б*. Экспериментальная оценка функции «дефект–сложность» вида (10) представлена графиком зависимости ε_{n^*} от C_{n^*}/C_n на рис. 4.

Графики на рис. 3 демонстрируют вычислительный выигрыш алгоритма приближенного поиска по сравнению с алгоритмом точного поиска от 34,25 до 46,30 раз при сохранении высоких показателей качества приближенного поиска: $0{,}0004 \leqslant \varepsilon_{n^*} \leqslant 0{,}0010$



(*а*) (*б*)

Рис. 3 Характеристики качества (*а*) и сложности (*б*) поиска

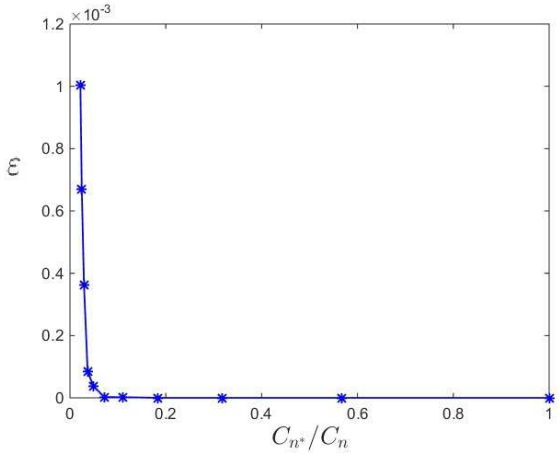


Рис. 4 Функция «дефект–сложность»

и $0{,}0065 \leqslant \sigma_{n^*} \leqslant 0{,}0100$ в диапазоне значений $8 \leqslant \log_2(n/n^*) \leqslant 10$. Показано достижение нулевого дефекта (точного решения) на иерархическом алгоритме, обеспечивающем вычислительный выигрыш в 5,45 раз по сравнению с алгоритмом перебора ($n^* = n/8$). В рамках предложенной стратегии сужения зоны поиска наименьшее значение $C_{n^*=1}/C_n$ (наибольший вычислительный выигрыш) и соответственно наибольший дефект $\varepsilon_{n^*=1}$ дает иерархический алгоритм приближенного поиска с параметром $n^* = 1$.

5 Заключение

Для массивов, заданных m-мерными кубами из N^m элементов дискретного алфавита, предложен иерархический алгоритм приближенного поиска ближайшего соседа к предъявляемому массиву среди множества массивов, образующих набор данных. Разработанный алгоритм ориентирован на ускорение поиска массивов большого размера с параметрами $m \geqslant 1$, $N = 2^L$ при $L \gg 1$, включая изображения с высоким уровнем разрешения. Алгоритм использует пирамидальные представления массивов с многоуровневым разрешением и параметрическую стратегию сужения зоны поиска на последовательных уров-

нях представления набора данных. Показано, что при фиксированной размерности $m \geqslant 1$, большом линейном размере массивов N и большой мощности набора данных n доля вычислительной сложности иерархического алгоритма приближенного поиска относительно сложности переборного алгоритма точного поиска составляет $O(\log N/N^m)$. Экспериментальная апробация разработанного иерархического алгоритма проведена на наборе изображений рукописных цифр из базы данных MNIST. По результатам эксперимента средний дефект иерархического алгоритма приближенного поиска ближайшего соседа оценивается величиной порядка 0,1% при 40-кратном вычислительном выигрыше по сравнению с переборным алгоритмом точного поиска. Дополнительное уменьшение вычислительной сложности приближенного поиска ближайшего соседа может быть достигнуто на объединении алгоритмов, использующих многоуровневое представление массивов и структуру набора данных в форме решающего дерева.

Литература

[1] *Friedman J. H., Bentley J. L., Finkel R. A.* An algorithm for finding best matches in logarithmic expected time // ACM Trans. Math. Softw., 1977. Vol. 3. No. 3. P. 209–226.

[2] *Cleary J. G.* Analysis of an algorithm for finding nearest neighbors in Euclidean space // ACM Trans. Math. Softw., 1979. Vol. 5. No. 2. P. 183–192.

[3] *Soleymani M. R., Morgera S. D.* An efficient nearest neighbor search method // IEEE Trans. Comm., 1987. Vol. 35. No. 6. P. 677–679.

[4] *Arya S., Mount D. M., Netanyahu N. S., Silverman R., Wu A. Y.* An optimal algorithm for approximate nearest neighbor searching in fixed dimensions / J. ACM, 1988. Vol. 45. No. 6. P. 891–923.

[5] *Andoni A., Indyk P.* Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions // Comm. ACM, 2008. Vol. 51. No. 1. P. 117–122.

[6] *Rosenfeld A.* Quadtrees and pyramids for pattern recognition and image analysis // 5th Conference (International) on Pattern Recognition Proceedings. Miami Beach, FL, USA, 1980. P. 802–811.

[7] *Jackins C. L., Tanimoto S. L.* Quadtrees, octtrees, and K-trees: A generalized approach to recursive decomposition of Euclidean space // IEEE Trans. PAMI, 1983. Vol. 5. No. 5. P. 533–539.

[8] *Samet H.* The quadtree and related hierarchical data structures // Computing Survey, 1984. Vol. 16. No. 2. P. 187–260.

[9] *Lange M. M., Stepanov D. Yu.* Recognition of objects given by collections of multichannel images // Pattern Recogn. Image Anal., 2014. Vol. 24. No. 3. P. 431–442.

[10] MNIST database of handwritten digits. `http://www.machinelearning.ru/wiki/index.php?title=MNIST_database_of_handwritten_digits`.

[11] *Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.* Introduction to algorithms. — 3rd ed. — MIT Press, 2009. 1292 p.

[12] Algorithm of approximate search for the nearest neighbour. `https://sourceforge.net/projects/edivis/files/`.

Поступила в редакцию 21.12.2015

Algorithm of approximate search for the nearest digital array in a hierarchical data set*

M. M. Lange, S. N. Ganebnykh, and A. M. Lange
lange_mm@ccas.ru, sng@ccas.ru, lange_am@mail.ru
Federal Research Center "Computer Science and Control" of RAS, 44/2 Vavilova st., Moscow, Russia

An algorithm of approximate fast search in a given set of multidimensional digital arrays for the nearest neighbor of a submitted array is suggested. A search error is defined by a ratio of a difference of distances from a submitted array to the really found array and to the nearest neighbor relative to the distance to the nearest neighbor. The proposed algorithm uses pyramid-based multiresolution representations of the arrays and a hierarchical search strategy. For a large linear size of the arrays and a large cardinality of the data set, an asymptotic computational gain of the approximate search algorithm with respect to the exact search algorithm is estimated. Given data set of grayscale handwritten digit images taken from the MNIST database, a mean search error, a standard deviation of the search errors, and a computational complexity of the algorithm as the appropriate functions of the search parameter are experimentally estimated. Using these estimates, a dependence of the mean search error on the computational complexity is calculated.

Keywords: *multidimensional array; data set; nearest neighbor; pyramid-based representation; approximate nearest search; search error; computational complexity*

DOI: 10.21469/22233792.2.1.01

References

[1] Friedman, J. H., J. L. Bentley, and R. A. Finkel. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.* 3(3):209–226.

[2] Cleary, J. G. 1979. Analysis of an algorithm for finding nearest neighbors in Euclidean space. *ACM Trans. Math. Softw.* 5(2):183–192.

[3] Soleymani, M. R., and S. D. Morgera. 1987. An efficient nearest neighbor search method. *IEEE Trans. Comm.* 35(6):677–679.

[4] Arya, S., D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. 1988. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM* 45(6):891–923.

[5] Andoni A., and P. Indyk. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Comm. ACM* 51(1):117–122.

[6] Rosenfeld, A. 1980. Quadtrees and pyramids for pattern recognition and image analysis. *5th Conference (International) on Pattern Recognition Proceedings*. Miami Beach, FL. 802–811.

[7] Jackins, C. L., and S. L. Tanimoto. 1983. Quadtrees, octtrees, and K-trees: A generalized approach to recursive decomposition of Euclidean space. *IEEE Trans. PAMI* 5(5):533–539.

[8] Samet, H. 1984. The quadtree and related hierarchical data structures. *Computing Survey* 16(2):187–260.

[9] Lange, M. M., and D. Yu. Stepanov. 2014. Recognition of objects given by collections of multichannel images. *Pattern Recogn. Image Anal.* 24(3):431–442.

[10] MNIST database of handwritten digits. Available at: http://www.machinelearning.ru/wiki/index.php?title=MNIST_database_of_handwritten_digits (accessed February 8, 2016).

*The work was partially supported by the Russian Foundation for Basic Research (grants 15-01-04671 and 15-07-07516).

[11] Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2009. *Introduction to algorithms*. 3rd ed. MIT Press. 1292 p.

[12] Algorithm of approximate search for the nearest neighbour. Available at: `https://sourceforge.net/projects/edivis/files/` (accessed February 8, 2016).

Received December 21, 2015

Метрическое обучение и снижение размерности пространства в задачах кластеризации*

Р. В. Исаченко, А. М. Катруца
isa-ro@yandex.ru, amkatrutsa@yandex.ru
Московский физико-технический институт, г. Долгопрудный, Институтский пер., 9

Работа посвящена использованию методов метрического обучения в задачах кластеризации. Применение метрического обучения позволяет модифицировать расстояния между объектами, сближая объекты из одного кластера и отдаляя объекты из разных кластеров. В данной работе расстояние измеряется при помощи метрики Махаланобиса. Процедура метрического обучения состоит в определении оптимальной матрицы ковариаций множества объектов. Кластеризация осуществляется алгоритмом k-средних и алгоритмом адаптивного метрического обучения, понижающим размерность признакового пространства. Для сравнения этих методов произведен вычислительный эксперимент на синтетических и реальных данных, сделан вывод об эффективности рассматриваемых методов.

Ключевые слова: *кластеризация; k-средних; алгоритм адаптивного метрического обучения; EM-алгоритм*

DOI: 10.21469/22233792.2.1.02

1 Введение

Методы метрического обучения [1, 2] применяются при идентификации лиц [3], классификации текстовых документов [4], распознавании рукописных цифр [5]. В данной работе решается задача метрического обучения при кластеризации объектов на заданное количество кластеров [6]. Требуется выявить наборы объектов, похожих между собой, причем степень сходства определяется расстоянием между объектами. Целью метрического обучения является выбор оптимального способа измерения расстояния между объектами.

Для решения данной задачи в работе [7] используются деревья принятия решений [8], основанные на логических схемах. Но проблема построения оптимального дерева является NP-полной, и практическое применение данного метода основано на эвристических алгоритмах [9]. Другой подход к кластеризации используется в статье [10], где предложен эффективный способ отбора признаков, а в качестве алгоритма кластеризации выбран EM (expectation-maximization) алгоритм. В работе [11] для кластеризации объектов применяется метрическое обучение. Эта работа используется в настоящем исследовании в качестве базовой.

Предлагаемый алгоритм адаптивного метрического обучения объединяет задачи кластеризации и метрического обучения в одну задачу максимизации функционала качества. Ключевой идеей алгоритма адаптивного метрического обучения является понижение размерности пространства объектов таким образом, чтобы расстояния между кластерами были максимальны. Для решения оптимизационной задачи используется EM-подход. На каждой итерации алгоритм находит ортогональное преобразование, понижающее размерность признакового пространства, и кластеризует объекты в новом пространстве.

В данной работе алгоритм адаптивного метрического обучения применяется к синтетическим и реальным данным. Цель эксперимента — показать работоспособность предложенного подхода и провести его сравнение с базовым алгоритмом. В качестве базового для

*Проект поддержан грантом РФФИ № 16-37-00485.

сравнения алгоритма выбран алгоритм k-средних [12]. Данный алгоритм минимизирует суммарное квадратичное отклонение точек кластеров от центров этих кластеров. Получена оценка качества работы построенного алгоритма. Проведен сравнительный анализ результатов, полученных с помощью метрического обучения и без него.

2 Постановка задачи метрического обучения

Пусть $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{T \times N}$ — множество объектов. Объект $\mathbf{x}_i = [x_i^1, \ldots, x_i^T]^\top$ задан в виде вектора в пространстве признаков. Требуется выявить кластерную структуру данных и разбить множество объектов $\mathbf{X}$ на множество непересекающихся кластеров, т. е. построить отображение

$$a : \mathbf{X} \to \{1, \ldots, K\}.$$

Обозначим $y_i = a(\mathbf{x}_i)$, $y_i \in \{1, \ldots, K\}$, — метка кластера объекта $\mathbf{x}_i$. Необходимо выбрать метки кластеров $\{y_i\}_{i=1}^N$ таким образом, чтобы расстояния между кластерами были максимальными. Центр $\boldsymbol{\mu}$ множества объектов $\mathbf{X}$ и центры кластеров $\{\boldsymbol{\mu}_k\}_{k=1}^K$ вычисляются по формулам:

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \,; \quad \boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} [y_i = y_k] \mathbf{x}_i}{\sum_{i=1}^{N} [y_i = y_k]} \,. \tag{1}$$

Введем на множестве объектов $\mathbf{X}$ расстояние Махаланобиса

$$\rho(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}^{-1} (\mathbf{x}_i - \mathbf{x}_j)} \,, \tag{2}$$

где $\mathbf{A}$ — это матрица ковариаций множества $\mathbf{X}$

$$\mathbf{A} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top. \tag{3}$$

Определение 1. *Функционалом качества кластеризации Q назовем межкластерное расстояние:*

$$Q(\{\boldsymbol{\mu}_k\}_{k=1}^K) = \sum_{k=1}^{K} N_k \rho^2(\boldsymbol{\mu}_k, \boldsymbol{\mu}) \,,$$

где $N_k = \sum_{i=1}^N [y_i = y_k]$ — число объектов в кластере k.

Поставим задачу кластеризации как задачу максимизации функционала

$$Q\big(\{\boldsymbol{\mu}_k\}_{k=1}^K\big) \to \max_{\boldsymbol{\mu}_k \in \mathbb{R}^T} . \tag{4}$$

Для улучшения качества решения этой задачи предлагается применить метод метрического обучения к ковариационной матрице $\mathbf{A}$. Найдем такую матрицу $\mathbf{A}$, для которой функционал качества принимает максимальное значение:

$$\mathbf{A}^* = \arg\max_{\mathbf{A} \in \mathbb{R}^{T \times T}} Q\big(\{\boldsymbol{\mu}_k^*\}_{k=1}^K\big) \,, \tag{5}$$

где $\{\boldsymbol{\mu}_k^*\}_{k=1}^K$ — решение задачи кластеризации (4).

3 Алгоритм адаптивного метрического обучения

Для решения поставленных оптимизационных задач (4), (5) используется алгоритм адаптивного метрического обучения. Предлагается понизить размерность пространства

объектов $\mathbf{X}$ с помощью линейного ортогонального преобразования $\mathbf{G} \in \mathbb{R}^{T\times L}$, $\mathbf{G}^\top\mathbf{G} = \mathbf{I}$, где новая размерность $L < T$

$$\mathbf{X} \ni \mathbf{x}_i \mapsto \hat{\mathbf{x}}_i = \mathbf{G}^\top\mathbf{x}_i \in \mathbb{R}^L, \quad i = 1, \ldots, N.$$

Центр $\hat{\boldsymbol{\mu}}$ множества объектов $\{\hat{\mathbf{x}}_i\}_{i=1}^N$ вычисляется по формуле (1). Расстояния между объектами вычисляются по формуле (2), где в качестве матрицы $\hat{\mathbf{A}}$ используется матрица ковариаций (3) множества объектов $\{\hat{\mathbf{x}}_i\}_{i=1}^N$

$$\hat{\mathbf{A}} = \frac{1}{N}\sum_{i=1}^N(\hat{\mathbf{x}}_i - \hat{\boldsymbol{\mu}})(\hat{\mathbf{x}}_i - \hat{\boldsymbol{\mu}})^\top = \frac{1}{N}\sum_{i=1}^N \mathbf{G}^\top(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top\mathbf{G} = \mathbf{G}^\top\mathbf{A}\mathbf{G}.$$

Определение 2. *Индикаторной матрицей назовем матрицу* $\mathbf{F} = \{\delta_{ik}\} \in \mathbb{R}^{N\times K}$, *где*

$$\delta_{ik} = \begin{cases} 1, & \text{если } a(\mathbf{x}_i) = y_k; \\ 0, & \text{если } a(\mathbf{x}_i) \neq y_k. \end{cases}$$

Определение 3. *Взвешенной индикаторной матрицей назовем матрицу* $\mathbf{L} = \mathbf{F}(\mathbf{F}^\top\mathbf{F})^{-1/2} = \{l_{ik}\} \in \mathbb{R}^{N\times K}$, *элементы которой равны:*

$$l_{ik} = \begin{cases} \dfrac{1}{\sqrt{N_k}}, & \text{если } a(\mathbf{x}_i) = y_k; \\ 0, & \text{если } a(\mathbf{x}_i) \neq y_k. \end{cases}$$

Утверждение 1. С использованием данных обозначений задача кластеризации (4) и задача метрического обучения (5) сводятся к общей задаче максимизации функционала качества [13]

$$Q = \frac{1}{N}\operatorname{tr}(\mathbf{L}^\top\mathbf{X}^\top\mathbf{G}\hat{\mathbf{A}}^{-1}\mathbf{G}^\top\mathbf{X}\mathbf{L}) = \frac{1}{N}\operatorname{tr}(\mathbf{L}^\top\mathbf{X}^\top\mathbf{G}(\mathbf{G}^\top\mathbf{A}\mathbf{G})^{-1}\mathbf{G}^\top\mathbf{X}\mathbf{L}) \to \max_{\mathbf{G},\mathbf{L}}. \tag{6}$$

4 Решение задачи метрического обучения

Для решения задачи (6) алгоритм адаптивного метрического обучения использует EM-подход. На каждом шаге итеративно вычисляются локальные оптимальные значения матриц $\mathbf{G}$ и $\mathbf{L}$. На E-шаге необходимо найти матрицу $\mathbf{L}$, которая является решением оптимизационной задачи (6) при фиксированной матрице $\mathbf{G}$. В качестве начального приближения получим взвешенную индикаторную матрицу $\mathbf{L}$ с помощью алгоритма кластеризации k-средних с евклидовой метрикой. На M-шаге производится нахождение оптимального значения матрицы $\mathbf{G}$ при фиксированной матрице $\mathbf{L}$. Алгоритм завершается при стабилизации функционала Q на последовательности итераций.

4.1 Алгоритм k-средних

В данной работе базовым алгоритмом для сравнения является алгоритм k-средних. Первым шагом алгоритм выбирает из множества $\mathbf{X}$ случайным образом K объектов $\{\boldsymbol{\mu}_k\}_{k=1}^K$ — начальные центры кластеров. Для каждого объекта $\mathbf{x}_i$ вычисляется расстояние (2) до каждого центра кластера $\boldsymbol{\mu}_k$ с единичной матрицей трансформаций. Объект $\mathbf{x}_i$ относится к кластеру, расстояние до которого оказалось наименьшим. Далее производится

вычисление новых центров кластеров по формуле (1). Алгоритм завершается, если значения центров кластеров прекращают меняться.

4.2 Оптимизация матрицы G с фиксированной матрицей L

Для любых двух квадратных матриц $\mathbf{A}$ и $\mathbf{B}$ справедливо $\mathrm{trace}(\mathbf{AB}) = \mathrm{trace}(\mathbf{BA})$. Данное свойство позволяет переформулировать задачу (6) следующим образом:

$$Q = \frac{1}{N}\operatorname{tr}(\mathbf{L}^\top\mathbf{X}^\top\mathbf{G}(\mathbf{G}^\top\mathbf{A}\mathbf{G})^{-1}\mathbf{G}^\top\mathbf{X}\mathbf{L}) = \frac{1}{N}\operatorname{tr}\big((\mathbf{G}^\top\mathbf{A}\mathbf{G})^{-1}\mathbf{G}^\top\mathbf{X}\mathbf{L}\mathbf{L}^\top\mathbf{X}^\top\mathbf{G}\big).$$

Утверждение 2. Обозначим $\mathbf{B} = \mathbf{X}\mathbf{L}\mathbf{L}^\top\mathbf{X}^\top$. Обозначим через $\mathbf{G} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$ матрицу, состоящую из K собственных векторов матрицы $\mathbf{A}^{-1}\mathbf{B}$, отвечающих наибольшим собственным значениям. Тогда решением (6) является ортогональная матрица, полученная QR-разложением матрицы $\mathbf{G}$.

Функционал качества Q зависит только от матрицы $\mathbf{G}$. Обозначим

$$s(\mathbf{G}) = \operatorname{tr}\big((\mathbf{G}^\top\mathbf{A}\mathbf{G})^{-1}\mathbf{G}^\top\mathbf{B}\mathbf{G}\big).$$

На данном шаге задача (6) принимает вид:

$$\mathbf{G}^* = \arg\max_{\mathbf{G}\in\mathbb{R}^{T\times L}} s(\mathbf{G})\,; \tag{7}$$

$$\mathbf{G}^\top\mathbf{G} = \mathbf{I}\,. \tag{8}$$

Ранг произведения матриц не превосходит рангов сомножителей, поэтому ранг матрицы $\mathbf{B}$ не превосходит K. Решением (7) является матрица $\mathbf{G} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$, состоящая из K собственных векторов матрицы $\mathbf{A}^{-1}\mathbf{B}$, отвечающих наибольшим собственным значениям. Таким образом, размерность нового пространства объектов будет равна количеству кластеров K.

В общем случае матрица $\mathbf{G}$ не является ортогональной. Заметим, что для любой невырожденной матрицы $\mathbf{G}$ верно $s(\mathbf{G}) = s(\mathbf{G}\mathbf{M})$. Для учета условия ортогональности (8) найдем QR-разложение матрицы $\mathbf{G}$. Тогда ортогональная матрица $\mathbf{Q}$ является оптимальным значением $\mathbf{G}^*$.

4.3 Оптимизация матрицы L с фиксированной матрицей G

Утверждение 3. Обозначим $\hat{\mathbf{K}} = (1/N)\mathbf{X}^\top\mathbf{G}\hat{\mathbf{A}}^{-1}\mathbf{G}^\top\mathbf{X}$. Тогда задача (6) эквивалентна задаче кластеризации k-средних с заданным ядром $\hat{\mathbf{K}}$ [14].

При фиксированной матрице $\mathbf{G}$ задача (6) принимает вид:

$$\operatorname{tr}(\mathbf{L}^\top\hat{\mathbf{K}}\mathbf{L}) \to \max_{\mathbf{L}\in\mathbb{R}^{N\times K}}.$$

Матрица $\hat{\mathbf{K}}$ является симметричной и неотрицательно определенной, тем самым может быть выбрана в качестве ядра.

5 Вычислительный эксперимент

В целях проверки работоспособности предложенного подхода проведен вычислительный эксперимент на модельных данных. Сгенерирована выборка объектов, принадлежащих одному из двух классов, в двумерном пространстве. Каждый объект принадлежит

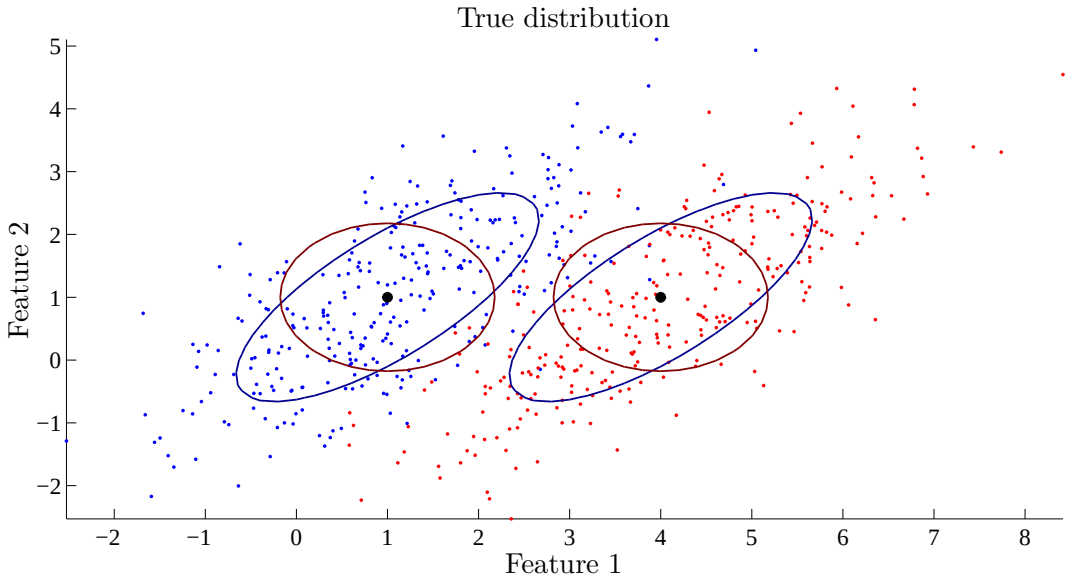


Рис. 1 Истинное распределение двумерных модельных данных

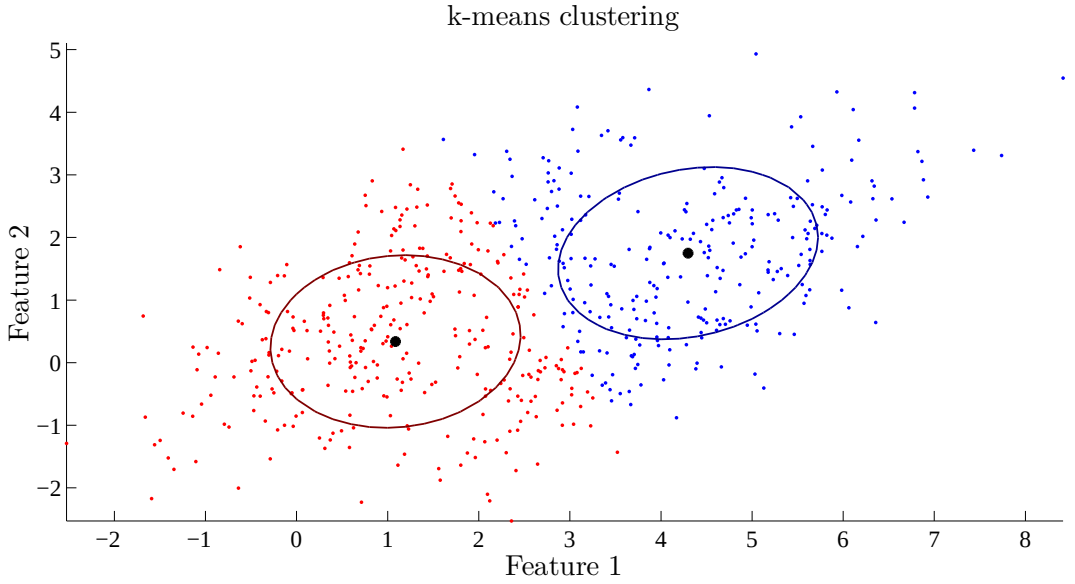


Рис. 2 Результат кластеризации алгоритмом k-средних

многомерному нормальному распределению. На рис. 1 показано истинное распределение объектов, черным цветом выделены истинные центры классов и линии уровня функции распределения.

Применим к данной выборке базовый алгоритм k-средних. Результат кластеризации показан на рис. 2, где черным цветом выделены найденные центры классов и линии уровня функции распределения, построенной по выборочной ковариационной матрице.

Взяв за начальное приближение результаты работы алгоритма k-средних, проведем кластеризацию с помощью алгоритма адаптивного метрического обучения. Результаты работы алгоритма продемонстрированы на рис. 3.

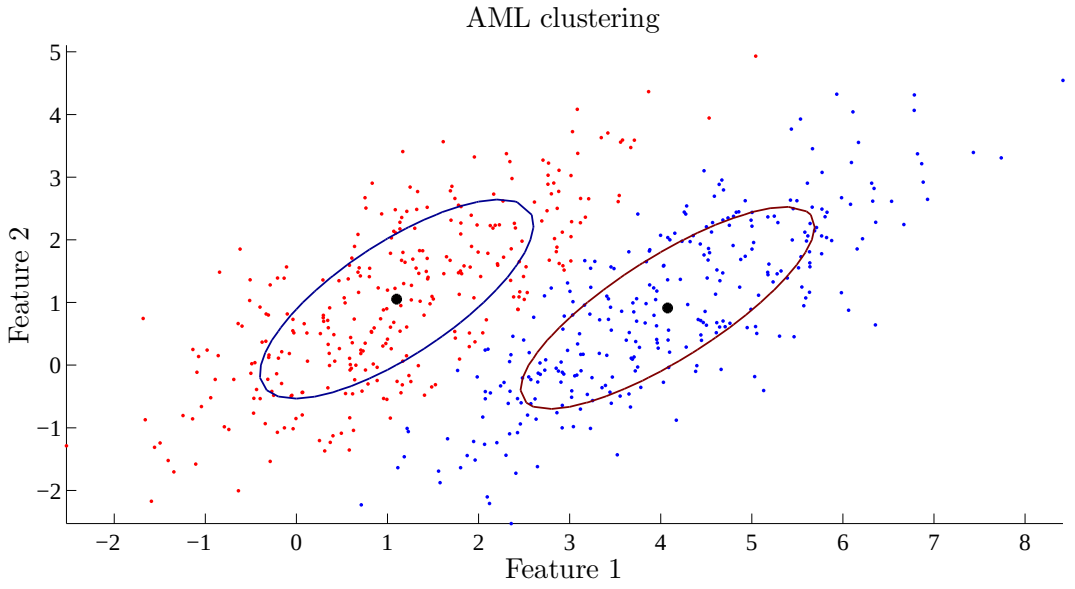


Рис. 3 Результат кластеризации алгоритмом адаптивного метрического обучения

Таблица 1 Результаты кластеризации

Выборка	Качество кластеризации	
	k-средних	AML
Letter Recognition	0,356	**0,428**
Optical Recognition of Handwritten Digits	0,758	**0,790**
Seeds	0,833	**0,881**
Image Segmentation	0,545	**0,737**
Breast Cancer Wisconsin	**0,960**	0,956

На рисунках заметно улучшение результатов кластеризации. Измеренная точность кластеризации алгоритма k-средних составила 0,76, алгоритма адаптивного метрического обучения — 0,94, что говорит о работоспособности данного подхода.

Таблица 1 показывает результаты вычислительного эксперимента на реальных данных. Алгоритм был применен к 5 выборкам, взятых из репозитория UCI [15–19]. Оценкой качества кластеризации служит число правильно кластеризованных объектов. При кластеризации объектов на более чем два класса возникает проблема соотнесения истинных классов с полученными кластерами. Данная проблема была формализована в виде задачи о назначениях и решена с помощью венгерского алгоритма. Вычислительный эксперимент на реальных данных показал увеличение точности кластеризации при использовании метрического обучения.

6 Заключение

В данной работе предложен новый способ снижения размерности задачи кластеризации объектов на заданное число кластеров. Сравнивались результаты кластеризации базового алгоритма k-средних и алгоритма адаптивного метрического обучения. Проведен вычислительный эксперимент на синтетических данных. Он показал наглядную интерпретацию алгоритма адаптивного метрического обучения и улучшение качества кластериза-

ции. Вычислительный эксперимент на реальных данных показал эффективность данного подхода в реальных задачах.

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

Литература

[1] Yang L., Jin R. Distance metric learning: A comprehensive survey. Michigan State University, 2006. Vol. 2. 51 p.

[2] Bellet A., Habrard A., Sebban M. A survey on metric learning for feature vectors and structured data. ArXiv:1306.6709, 2013.

[3] Guillaumin M., Verbeek J., Schmid C. Is that you? Metric learning approaches for face identification // IEEE 12th Conference (International) on Computer Vision Proceedings, 2009. P. 498–505.

[4] Yang L., Jin R., Sukthankar R., Liu Y. An efficient algorithm for local distance metric learning // AAAI, 2006. Vol. 2. P. 543–548.

[5] Globerson A., Roweis S. T. Metric learning by collapsing classes // Advances in neural information processing systems / Eds. Y. Weiss, B. Schölkopf, J. Platt. — Cambridge, MA, USA: MIT Press, 2005. Vol. 18. P. 451–458.

[6] Xing E. P., Ng A. Y., Jordan M. I., Russell S. Distance metric learning with application to clustering with side-information // Advances in Neural Information Processing Systems, 2003. Vol. 15. P. 505–512.

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

[8] Friedl M. A., Brodley C. E. Decision tree classification of land cover from remotely sensed data // Remote Sensing of Environment, 1997. Vol. 61. No. 3. P. 399–409.

[9] Hyafil L., Rivest R. L. Constructing optimal binary decision trees is np-complete // Inform. Proc. Lett., 1976. Vol. 5. No. 1. P. 15–17.

[10] Dy J. G., Brodley C. E. Feature selection for unsupervised learning // J. Machine Learning Res., 2004. Vol. 5. P. 845–889.

[11] Ye J., Zhao Z., Liu H. Adaptive distance metric learning for clustering // IEEE Conference on Computer Vision and Pattern Recognition, 2007. P. 1–7.

[12] MacQueen J., et al. Some methods for classification and analysis of multivariate observations // 5th Berkeley Symposium on Mathematical Statistics and Probability Proceedings, 1967. Vol. 1. P. 281–297.

[13] Ding C., He X., Simon H. D. On the equivalence of nonnegative matrix factorization and spectral clustering // SIAM Data Mining Conference Proceedings, 2005. P. 606–610.

[14] Shawe-Taylor J., Cristianini N. Kernel methods for pattern analysis. — Cambridge University Press, 2004. 478 p.

[15] Letter recognition dataset. `http://archive.ics.uci.edu/ml/machine-learning-databases/letter-recognition/`.

[16] Optical recognition of handwritten digits dataset. `http://archive.ics.uci.edu/ml/machine-learning-databases/optdigits/`.

[17] Seeds dataset. `http://archive.ics.uci.edu/ml/machine-learning-databases/00236/`.

[18] Image segmentation dataset dataset. `http://archive.ics.uci.edu/ml/machine-learning-databases/image/`.

[19] Breast cancer wisconsin dataset. http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/.

Поступила в редакцию 24.02.2016

Metric learning and dimensionality reduction in clustering*

R. V. Isachenko and A. M. Katrutsa
isa-ro@yandex.ru, amkatrutsa@yandex.ru
Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow, Russia

This paper investigates incorporation of metric learning approach in clustering problem. Distance metric is a key issue in many machine learning algorithms, especially in unsupervised learning where distance between objects is the only known information. The metric learning procedure modifies distances between objects to make objects from the same cluster closer and from the different clusters more distant. In this paper, Mahalanobis distance is used as a distance between objects. The goal of the paper is to learn Mahalanobis metric by optimizing the covariance matrix of objects according to their cluster labels. In this case, metric learning procedure is formulated as optimization problem. For clustering, k-means were used as baseline algorithm and Adaptive Metric Learning (AML) algorithm. To solve the problem, AML algorithm uses iterative EM (expectation-maximization) procedure to find the optimum. To compare these algorithms, the computational experiment was carried out in MatLab on synthetic data and real data from UCI repository and conclusions about performance of these algorithms have been made.

Keywords: *clustering; k-means; adaptive distance metric learning; EM-algorithm*

DOI: 10.21469/22233792.2.1.02

References

[1] Yang, L., and R. Jin. 2006. Distance metric learning: A comprehensive survey. Michigan State University. Vol. 2. 51 p.

[2] Bellet, A., A. Habrard, and M. Sebban. 2013. A survey on metric learning for feature vectors and structured data. arXiv:1306.6709.

[3] Guillaumin, M., J. Verbeek, and C. Schmid. 2009. Is that you? Metric learning approaches for face identification. *IEEE 12th Conference (International) on Computer Vision Proceedings*. 498–505.

[4] Yang, L., R. Jin, R. Sukthankar, and Y. Liu. 2006. An efficient algorithm for local distance metric learning. *AAAI* 2:543–548.

[5] Globerson, A., and S. T. Roweis. 2005. Metric learning by collapsing classes. *Advances in neural information processing systems*. Eds. Y. Weiss, B. Schölkopf, and J. Platt. Cambridge, MA, USA: MIT Press. 18:451–458.

[6] Xing, E. P., A. Y. Ng, M. I. Jordan, and S. Russell. 2003. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems* 15:505–512.

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

*The work was supported by the Russian Foundation for Basic Research (grant No. 16-37-00485).

[8] Friedl, M. A., and C. E. Brodley. 1997. Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment* 61(3):399–409.

[9] Hyafil, L., and R. L. Rivest. 1976. Constructing optimal binary decision trees is np-complete. *Inform. Proc. Lett.* 5(1):15–17.

[10] Dy, J. G., and C. E. Brodley. 2004. Feature selection for unsupervised learning. *J. Machine Learning Res.* 5:845–889.

[11] Ye, J., Z. Zhao, and H. Liu. 2007. Adaptive distance metric learning for clustering. *IEEE Conference on Computer Vision and Pattern Recognition*. 1–7.

[12] MacQueen, J., *et al.* 1967. Some methods for classification and analysis of multivariate observations. *5th Berkeley Symposium on Mathematical Statistics and Probability Proceedings*. 1(14):281–297.

[13] Ding, C., X. He, and H. D. Simon. 2005. On the equivalence of nonnegative matrix factorization and spectral clustering. *SIAM Data Mining Conference Proceedings*. 606–610.

[14] Shawe-Taylor, J., and N. Cristianini. 2004. *Kernel methods for pattern analysis*. Cambridge University Press. 478 p.

[15] Letter recognition dataset. Available at: `http://archive.ics.uci.edu/ml/machine-learning-databases/letter-recognition/` (accessed March 28, 2016).

[16] Optical recognition of handwritten digits dataset. Available at: `http://archive.ics.uci.edu/ml/machine-learning-databases/optdigits/` (accessed March 28, 2016).

[17] Seeds dataset. Available at: `http://archive.ics.uci.edu/ml/machine-learning-databases/00236/` (accessed March 28, 2016).

[18] Image segmentation dataset dataset. Available at: `http://archive.ics.uci.edu/ml/machine-learning-databases/image/` (accessed March 28, 2016).

[19] Breast cancer wisconsin dataset. Available at: `http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/` (accessed March 28, 2016).

Received February 24, 2016

Оценка эффекта множественного тестирования в методе оптимальных достоверных разбиений*

***О. В. Сенько*[1], *А. М. Морозов*[2], *А. В. Кузнецова*[3], *Л. Л. Клименко*[4]**

senkoov@mail.ru, alxmopo3ov@gmail.com, azfor@narod.ru, klimenkoll@mail.ru

[1]ФИЦ «Информатика и управление» РАН, г. Москва, ул. Вавилова, 44/2

[2]МГУ им. М. В. Ломоносова, г. Москва, Ленинские горы, 1

[3]Институт биохимической физики им. Н. М. Эмануэля, г. Москва, ул. Косыгина, 4

[4]Институт химической физики им. Н. Н. Семёнова, г. Москва, ул. Косыгина, 4

Разработка методов поиска статистически достоверных эмпирических закономерностей является одной из приоритетных задач интеллектуального анализа данных. Одной из возможных технологий поиска таких закономерностей является метод оптимальных достоверных разбиений (ОДР), который использует для статистической верификации перестановочный тест. В условиях высокой размерности данных оценка достоверности двумерных закономерностей существенно осложняется проблемой множественного тестирования. Использование стандартного метода коррекции Бонферрони требует фиксации чрезвычайно жестких и практически редко достижимых порогов при отборе достоверных закономерностей при размерности данных выше 100. Серия Монте-Карло экспериментов была проведена для оценки истинной достоверности закономерностей, выявленных при решении биомедицинской задачи изучения связи уровня фактора роста сосудов (VEGF — vascular endothelial growth factor) с широким набором биологических показателей. Набор закономерностей, найденных в исходной выборке, сравнивался с наборами закономерностей, найденных в 50 случайных выборках, полученных из исходной путем случайных перестановок значений целевой переменной. Эксперименты показали, что доля двумерных закономерностей, для которых исходная статистическая значимость, рассчитанная с помощью нескорректированного теста не хуже фиксированного уровня α, оказывается в 10–30 раз ниже величины αN_p, где N_p — число просмотренных пар объясняющих переменных. В статье также обсуждаются подходы, направленные на смягчение условий достоверности для закономерностей.

Ключевые слова: *закономерности; перестановочный тест; множественное тестирование*

DOI: 10.21469/22233792.2.1.03

1 Введение

Создание новых биофизических и биохимических методов исследования живых организмов привело к значительному росту числа показателей, заносимых в биомедицинские базы данных. Изучение взаимосвязи данных показателей потенциально может привести к обнаружению новых эмпирических закономерностей, важных для понимания функционирования биологических систем, а также при решении разнообразных задач диагностики и прогнозирования. Однако эффективность поиска действительно достоверных общих для всей генеральной совокупности закономерностей снижается из-за известной проблемы множественного тестирования, состоящей в случайном возникновении в задачах высокой размерности таких конфигураций данных, которые ошибочно верифицируются стандартными нескорректированными статистическими тестами как достоверные закономерности.

*Работа выполнена при финансовой поддержке РФФИ, проект № 14-07-00819.

При этом вероятность хотя бы одного ошибочного объявления конфигурации данных закономерностью может значительно превышать уровень значимости, рассчитанный с помощью стандартного теста. Для того чтобы обеспечить исключение из последующего анализа ложных закономерностей, необходимо использование дополнительных более жестких критериев отбора. Наиболее известными методами модификации статистических критериев являются известная поправка Бонферрони, фактически предложенная в работе [1]. В последующие годы был разработан ряд дополнительных уточняющих критериев, включая критерии Бонферрони–Холма [2], Шидака [3], Хохберга [4]. Настоящая работа посвящена оценке величины эффекта множественного тестирования при поиске двумерных закономерностей с помощью метода ОДР [5–7].

Метод ОДР направлен на поиск одномерных или двумерных закономерностей, описывающих зависимость целевой величины Y от переменных, обозначаемых обычно буквой X, которые далее называются X-переменными. Достоверные двумерные закономерности в методе ОДР ищутся через построение оптимальных разбиений совместных областей допустимых значений для всевозможных пар X-переменных. Верификация закономерностей, характеризуемых оптимальными разбиениями, производится с помощью специального варианта перестановочного теста. Отметим, что технология верификации, основанная на перестановочных тестах, не требует априорных предположений о типе распределений, легко реализуется при произвольном виде статистик и, несмотря на высокую трудоемкость, получает все большее распространение [8–11]. Перестановочным тестом проверяются нулевые гипотезы о независимости Y от X-переменных. При этом общее число таких гипотез равно числу всевозможных пар X-переменных, которое оказывается чрезмерно большим для многих биомедицинских задач. Вследствие этого использование приведенных выше способов коррекции, основанных на оценивании сверху вероятности отклонения хотя бы одной из нулевых гипотез, приводит к неоправданно жестким критериям отбора, серьезно затрудняющим применение двумерного ОДР анализа уже при нескольких десятках X-переменных. Настоящее исследование основано на прямом подсчете встречаемости двумерных закономерностей с различными нескорректированными уровнями значимости в общем наборе двумерных закономерностей, полученных с помощью метода ОДР.

2 Метод оптимальных достоверных разбиений

Метод ОДР представляет собой метод анализа данных, позволяющий описать зависимость целевой переменной Y от некоторой переменной X, или от пары переменных X_1 и X_2 по выборке $\tilde{S}_t$ вида $\{(y_1, x_1), \ldots, (y_m, x_m)\}$ или $\{(y_1, x_{11}, x_{12}), \ldots, (y_m, x_{m1}, x_{m2})\}$, где y_j — значение целевой переменной Y на объекте s_j, а x_j, x_{j1} и x_{j2} — значения на объекте s_j переменных X, X_1 и X_2 соответственно, $j = 1, \ldots, m$. В основе метода лежит попытка построить такое разбиение интервала допустимых значений переменной X или совместной области допустимых значений переменных X_1 и X_2, чтобы объекты выборки $\tilde{S}_t$, принадлежащие разным элементам разбиения, по возможности сильнее отличались по уровням значений переменной Y.

Поиск оптимальных разбиений. Разбиения ищутся внутри нескольких семейств различного уровня сложности, включая

1) семейство I, состоящее из одномерных разбиений с одной граничной точкой;
2) семейство II, состоящее из одномерных разбиений с двумя граничными точками;
3) семейство III двумерных разбиений с двумя границами, параллельными координатным осям.

На рис. 1–3 приведены примеры разбиений из каждого из трех упомянутых семейств.

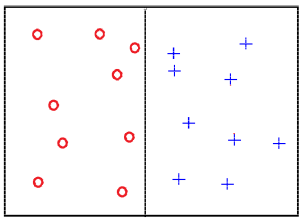


Рис. 1 Семейство I

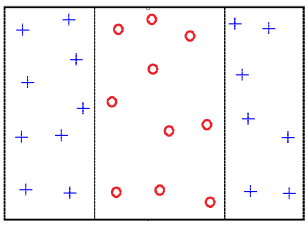


Рис. 2 Семейство II

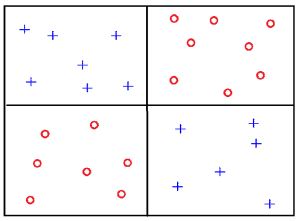


Рис. 3 Семейство III

Произвольное разбиение R из семейства I, которое далее будем обозначать $\widetilde{R}_I$, состоит из двух элементов (квадрантов) — q_1 и q_2. Произвольное разбиение R из семейства III, которое далее будем обозначать $\widetilde{R}_{III}$, состоит из четырех элементов (квадрантов) — q_1, q_2, q_3 и q_4. Пусть $\bar{Y}_0 = (1/m)\sum_{j=1}^{m} y_j$ — среднее значение целевой переменной по всей выборке $\widetilde{S}_t$; m_i — число объектов $\widetilde{S}_t$, для которых значения переменных X_1 и X_2 принадлежат квадранту q_i; $\bar{Y}_i$ — среднее значение целевой переменной по объектам $\widetilde{S}_t$, для которых значения переменных X_1 и X_2 принадлежат квадранту q_i; $D(Y)$ — дисперсия целевой переменной Y по всей обучающей выборке $\widetilde{S}_t$.

Оптимальным внутри семейства $\widetilde{R}_1$ считается разбиение, для которого достигает максимума значение функционала

$$Q^1(\widetilde{S}_t, R) = \frac{1}{D(Y)} \sum_{i=1}^{2} (\bar{Y}_0 - \bar{Y}_i)^2 m_i \,.$$

Оптимальным внутри семейства $\widetilde{R}_3$ считается разбиение, для которого достигает максимума значение функционала

$$Q^3(\widetilde{S}_t, R) = \frac{1}{D(Y)} \sum_{i=1}^{4} (\bar{Y}_0 - \bar{Y}_i)^2 m_i \,.$$

Разбиение, на котором достигается максимум функционалов Q^1 или Q^3, будет обозначаться R_o.

Верификация закономерностей. Верификация закономерностей из семейства $\widetilde{R}_1$ основана на попытке опровержения простой нулевой гипотезы о том, что целевая переменная Y не зависит от переменной X. Для этих целей используется вариант перестановочного теста, состоящий в многократном повторении поиска оптимального разбиения на множестве выборок $\{\widetilde{S}_t^f | f \in \widetilde{f}\}$, полученных из исходной выборки путем случайных перестановок значений целевой переменной Y относительно фиксированных значений переменной X. Через $\widetilde{f}$ обозначено получаемое с помощью генератора случайных чисел множество случайных перестановок чисел из набора $\{1, \ldots, m\}$. В качестве p-значения используется вероятность превышения величины $Q^1(\widetilde{S}_t, R_o)$ при условии соблюдения нулевой гипотезы. Данная вероятность оценивается как доля выборок из $\{\widetilde{S}_t^f | f \in \widetilde{f}\}$, для которых выполняется неравенство

$$Q^1(\widetilde{S}_t^r, R_o^f) \geqslant Q^1(\widetilde{S}_t, R_o).$$

Через R_o^f обозначено оптимальное разбиение, построенное по случайной выборке $\widetilde{S}_t^f$. Описанный вариант перестановочного теста, исследованный в работе [5], далее будем называть первым вариантом.

Первый вариант перестановочного теста не может быть использован для верификации закономерностей, задаваемых разбиениями из семейства III, поскольку его применение приводит к появлению на выходе большого числа так называемых частично ложных закономерностей. Под частично ложной понимается такая двумерная закономерность, для которой достоверность наличия связи между Y и двумя X-переменными на самом деле обеспечивается только одной переменной из пары X_1 и X_2. Включение же второй переменной является по сути случайным.

Второй вариант перестановочного теста основан на попытке опровержения нулевой гипотезы о достаточности одних только одномерных моделей для описания существующей связи. Подобный подход может трактоваться как вариант известного методологического принципа бритвы Оккама. На практике изучается возможность опровержения нулевых гипотез о достаточности одномерных разбиений, ближайших к верифицируемому двумерному разбиению. В качестве ближайших выступают одномерные разбиения, имеющие границы, совпадающие с соответствующими границами верифицируемого двумерного разбиения [7]. Нулевая гипотеза о достаточности одномерного разбиения R считается эквивалентной гипотезе о независимости Y от X-переменных внутри двух квадрантов R. Второй вариант перестановочного теста основан на проверке таких нулевых гипотез.

Опишем данный вариант подробно. Предположим, что оптимальное двумерное разбиение R_o^2 задается границей b_1 для переменной X_1 и границей b_2 для переменной X_2. Пусть R_1 и R_2 — одномерные разбиения, задаваемые границами b_1 и b_2 соответственно. Обозначим через $\widetilde{f}_R$ множество случайных перестановок чисел из набора $\{1,\ldots,m\}$ с запрещенным обменом номерами объектов из $\widetilde{S}_t$ с X-описаниями, принадлежащим разным квадрантам простого одномерного разбиения R. Сгенерируем с помощью датчика случайных чисел множества выборок $\widetilde{\mathbf{S}}_1 = \{\widetilde{S}_t^f | f \in \widetilde{f}_{R_1}\}$ и $\widetilde{\mathbf{S}}_2 = \{\widetilde{S}_t^f | f \in \widetilde{f}_{R_2}\}$. Второй вариант перестановочного теста вычисляет для R^2 два параметра:

1) p_1 — оценку вероятности достижения (или превышения) величины $Q^3(\widetilde{S}_t, R_o)$ при условии соблюдения нулевой гипотезы о независимости Y от X-переменных внутри двух квадрантов R_1;
2) p_2 — оценку вероятности достижения (или превышения) величины $Q^3(\widetilde{S}_t, R_o)$ при условии соблюдения нулевой гипотезы о независимости Y от X-переменных внутри двух квадрантов R_2.

Параметр p_i оценивается как доля выборок из $\widetilde{\mathbf{S}}_i$, для которых выполняется неравенство:

$$Q^3(\widetilde{S}_t^f, R_o^f) \geqslant Q^3(\widetilde{S}_t, R_o). \tag{1}$$

Рассчитанные по сгенерированным множествам выборок $\widetilde{\mathbf{S}}_1$ и $\widetilde{\mathbf{S}}_2$ с использованием неравенства (1) параметры p_1 и p_2 выступают в качестве p-значений. Будем считать, что параметр p_1 описывает достоверность опровержения нулевой гипотезы о достаточности R_2 для описания взаимосвязи Y с X-переменными и, следовательно, характеризует достоверность необходимости использования в двумерной закономерности переменной X_1.

Параметр p_2 описывает достоверность опровержения нулевой гипотезы о достаточности R_1 для описания взаимосвязи Y с X-переменными и характеризует достоверность необходимости использования в двумерной закономерности переменной X_2.

Двумерная закономерность считается значимой на уровне α, если одновременно выполняются неравенства $p_1 < \alpha$ и $p_2 < \alpha$. Использование изложенного подхода в методе ОДР описано в работе [6]. В работе [12] рассматривалось аналогичное применение пере-

становочных тестов для оценки необходимости аппроксимации данных кусочно-линейной регрессионной моделью вместо более простой линейной.

3 Проблема множественного тестирования при поиске закономерностей

Метод ОДР эффективно оценивает достоверность закономерности, связывающей целевую переменную Y с сочетанием переменных X_1 и X_2. Важную информацию для изучения зависимости переменной Y от совокупности переменных $\widetilde{X} = \{X_1, :, X_n\}$ может дать анализ двумерных закономерностей, связывающих Y со всевозможными парными сочетаниями переменных из $\widetilde{X}$. Поиск таких закономерностей сводится, согласно содержанию предыдущего раздела, к проверке набора нулевых гипотез о независимости Y от переменной из $\widetilde{X}$ внутри квадрантов простых разбиений. В силу самой природы статистической верификации вероятность случайного достижения (или превышения) значения статистики критерия для хотя бы одной нулевых гипотез из $H_1, \ldots, H_r$ значительно больше вероятности такого события при проверке одной индивидуальной гипотезы. В случае ОДР вероятность случайного достижения функционала качества $Q^3(\widetilde{S}_t, R_0)$ хотя бы для одного из парных сочетаний переменных из $\widetilde{X}$ может значительно превышать p-значения, рассчитанные при верификации отдельной закономерности без учета эффекта множественного тестирования. Вследствие этого настоящий уровень значимости найденной закономерности оказывается хуже уровня значимости, рассчитанного с помощью простого применения перестановочного теста. Проблему необходимости использования существенно более жестких критериев отбора при тестировании большого числа исходных нулевых гипотез принято называть проблемой множественного тестирования (множественных сравнений).

Наиболее распространенными способами оценивания верхних границ для вероятности случайного отклонения хотя бы одной нулевой гипотезы является метод коррекции Бонферрони–Холма. В данном методе коррекция уровня значимости производится путем простого умножения исходного уровня значимости α, рассчитанного с помощью используемого нескорректированного статистического критерия C, на множитель $r - r_v + 1$, где r — общее число проверяемых нулевых гипотез; r_v — общее число проверяемых нулевых гипотез, которые были отвергнуты на уровне значимости α. Иными словами, при наличии r_v нулевых гипотез, отвергнутых C на уровне α, эти гипотезы следует считать достоверно отвергнутыми на уровне $\alpha(r - r_v + 1)$. При использовании двумерных моделей типа III из метода ОДР общее число проверяемых нулевых гипотез очевидно равно удвоенному значению различных пар X-переменных или $r = n(n-1)$ (см. разд. 1). В современных биомедицинских базах данных общее число разнообразных клинических, лабораторных или инструментальных показателей, которые могут рассматриваться в качестве X-переменных, нередко достигает 150–200 или даже более высоких значений. Таким образом, общее число тестируемых нулевых гипотез достигает $2 \cdot 10^4$–$4 \cdot 10^4$, а величин множителя $r - r_v + 1$ может существенно превышать 10^4. Для того чтобы закономерности можно было достоверно считать значимыми на уровне $p < 0{,}05$ или $p < 0{,}01$ согласно методу Бонферрони–Холма, необходимо, чтобы p-значения, рассчитанные согласно способу из разд. 2 не превышали 10^{-6}. Для корректной оценки столь низких p-значений требуется свыше 10^6 перестановок, что потребовало бы чрезвычайно высоких объемов вычислений. Кроме того, столь высокая значимость закономерностей достигается редко при наиболее распространенных объемах баз клинических данных, включающих порядка 10^2–10^3 случаев. Однако теория Бонферрони–Холма основана на завышенной оценке вероятности

ошибочного отклонения нулевых гипотез в условиях множественного тестирования, что, в свою очередь, приводит к существенному занижению уровня достоверности выявляемых закономерностей. Существенно более точную картину может дать использование методов, основанных на сравнении наборов закономерностей, найденных в реальной выборке с наборами закономерностей, найденных в случайных выборках, имеющих сходные с исходной выборкой структуру и объем. Одним из способов генерации случайных выборок может быть случайная перестановка позиций целевой переменной относительно фиксированных позиций X-переменных. Это означает, что для целей коррекции, связанной с проблемой множественного тестирования, может быть использована фактически та же самая схема, которая лежит в основе перестановочных тестов, используемых для тестирования отдельных закономерностей. Следует отметить, что перестановочный тест достаточно активно используется для оценки величины эффекта множественного тестирования. В этой связи могут быть упомянуты работы [13, 14].

4 Задача анализа связи VEGF с другими биологическими показателями

Целью исследования было исследование взаимосвязи уровня содержания в сыворотке крови эндотелиальныого фактора роста кровеносных сосудов белка VEGF с различнымми биологическими и биохимическими показателями. VEGF влияет на развитие новых кровеносных сосудов (ангиогенез) и развитие незрелых кровеносных сосудов (сосудистая поддержка), запуская сигнальный каскад, который в конечном итоге стимулирует рост эндотелиальных клеток сосуда, их функционирование и пролиферацию [15]. Для достижения более высокой устойчивости и наглядности анализа на предварительном этапе непрерывный показатель содержания VEGF в сыворотке крови переводился в бинарную форму, т. е. в качестве целевой переменной использовался бинарный показатель VEGF-bin, равный 1 при содержании VEGF менее 750 нг/л и равный 2 при содержании VEGF более 750 нг/л.

Метод ОДР использовался для изучения связи VEGF со стандартными биохимическими показателями, концентрацией гормонов щитовидной железы и половых гормонов, показателями коагуллограммы, концентрацией нейроспецифических белков, характеризующих повреждение мозговой ткани при ишемическом инсульте (ИИ). В качестве X-переменных рассматривались также уровни макро- и микроэлементов в сыворотке крови, а также значения показателей энергетического метаболизма мозга — уровня постоянного потенциала (УПП). В общем, изучалась взаимосвязь целевой переменной со 142 показателями.

В исследование была включена группа из 55 пациентов с возрастом от 40 до 88 лет, имеющих в анамнезе ИИ и группа из 33 пациентов с возрастом от 33 до 84 лет, имеющих в анамнезе случаи транзиторой ишемической атаки (ТИА).

Использование изложенного выше метода ОДР со вторым вариантом престановочного теста при использовании 2000 случайных престановок выявило следующее распределение закономерностей, описываемых разбиениями из $\widetilde{R}_3$, по уровню значимости:

– 158 двумерных закономерностей, для которых

$$\max(p_1, p_2) < 0{,}05;$$

– 24 двумерные закономерности, для которых

$$\max(p_1, p_2) < 0{,}01;$$

– 12 двумерных закономерностей, для которых

$$\max(p_1, p_2) < 0{,}005;$$

– 1 двумерная закономерность, для которой

$$\max(p_1, p_2) < 0{,}0005.$$

Таким образом, одна из найденных двумерных закономерностей имеет согласно второму варианту перестановочного теста значимость, определяемую неравествами $p_1 < 0{,}0005$ и $p_2 < 0{,}0005$. Данная закономерность связывает бинарный показатель VEGF-bin с концентрацией нейроспецифических белков S-100 и показателем насыщения (сатурации) крови кислородом (sO2). Белки S-100 — группа кальцийсвязывающих белков с низким молекулярным весом, участвующих в регуляции разнообразных внутриклеточных и межклеточных процессов. Известно, что уровень S-100 коррелирует с повреждением мозговой ткани при ИИ. Также в ранней фазе церебрального инфаркта S-100 является ответом мозговой ткани на ишемию [16]. Индекс сатурации sO2 представляет собой долю гемоглобина, связанного с кислородом, и является важным показателем, характеризующим снабжение тканей кислородом [17]. Закономерность графически представлена на рис. 4.

Случаи с уровнем VEGF выше 750 обозначены +, случаи с уровнем VEGF ниже 750 обозначены **O**. В каждом квадранте находится дробь, в числителе которой находится число случаев, обозначенных значком +, в знаменателе находится число случаев, обозначенных значком **O**. Квадранты пронумерованы римскими цифрами с возрастанием номера по

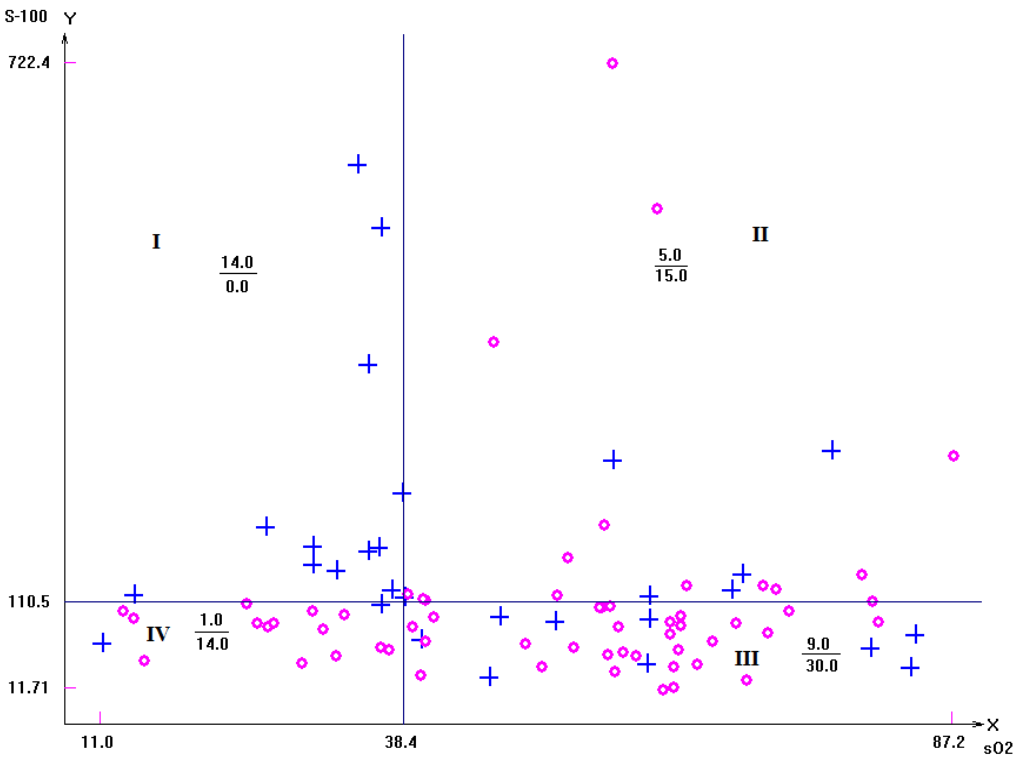


Рис. 4 Двумерная закономерность, связывающая коэффициент сатурации sO2 и S-100 с VEGF

ходу часовой стрелки. Нумерация начинается от верхнего левого квадранта. В квадранте I находятся только наблюдения с уровнем VEGF выше 750. В остальных квадрантах преобладают наблюдения из группы с низкими значениями VEGF. Наиболее сильное преобладание наблюдается в квадранте IV. Таким образом, низкий уровень сатурации sO2 в сочетании с высоким значением S-100 в преобладающем большинстве случаев соответствует высокому значению VEGF, что, возможно, связано с необходимостью компенсации недостаточного уровня снабжения головного мозга кислородом. Отметим, что связь содержания S-100 с VEGF выявляется также с использованием простейшей одномерной модели метода ОДР при $p = 0{,}002$.

Отметим, что метод ОДР позволил также выявить целый ряд двумерных закономерностей, описывающих связь между VEGF и S-100 в сочетании с целым рядом других показателей. В их число вошли общий уровень гемоглобина, фракции оксигемоглобина FO2Hb и дезоксигемоглобина FHHb в общем гемоглобине, общая железосвязывающая способность сыворотки (ОЖСС), парциальное давление кислорода в венозной крови (pO2), парциальное давление углекислого газа в венозной крови (pCO2), общее содержание Ca.

В ячейках табл. 1 для каждой закономерности даны значения вошедших в нее показателей, соответствующие границы и p-значения, рассчитанные с помощью второго варианта перестановочного теста. В двух правых колонках, озаглавленных «Распределение»,

Таблица 1 Двумерные закономерности, в которых одним из факторов является S-100

Показатели	Границы	p-значения	Распределение	
			VERF $>$ 750	VERF $<$ 750
Hg	127,5	0,012	0/1	12/5
S-100	146,348	0,002	9/4	8/49
ОЖСС	39,5	0,002	5/10	17/20
S-100	86,738	0,002	6/0	1/29
pCO2	44,0	0,013	2/2	16/11
S-100	114,445	0,002	5/0	6/46
pO2	40,0	0,01	17/10	1/2
S-100	116,268	$p < 0{,}0005$	6/47	5/0
sO2	38,4	$p < 0{,}0005$	14/0	5/15
S-100	110,54	$p < 0{,}0005$	1/14	9/30
FO2Hb	37,075	0,025	13/1	6/14
S-100	110,54	0,001	1/14	9/30
FHHb	54,6	0,007	3/11	15/1
S-100	116,268	$p < 0{,}0005$	7/28	4/19
Ca	2,255	$p < 0{,}0005$	2/16	11/2
S-100	114,44	0,007	28/68	18/5

представлено распределение случаев с VEGF > 750 и VEGF < 750. Дроби, приведенные в этих ячейках, имеют тот же смысл, что и дроби в квадрантах на рис. 4. Расположение ячеек в двух правых колонках таблицы совпадает с расположением соответствующих квадрантов.

Необходимо отметить, что почти все показатели, вошедшие в закономерности из табл. 1, непосредственно связаны со снабжением кислородом головного мозга.

5 Компьютерные эксперименты по оценке эффекта множественного тестирования

Изучение эффекта множественного тестирования основывается на сравнении достоверности закономерностей, найденных в случайных выборках, с достоверностью закономерностей, найденных в исходной выборке. При этом случайные выборки генерировались из исходной выборки путем случайных перестановок позиций значений целевой переменной относительно фиксированных позиций векторов X-переменных. Для оценивания достоверности двумерных закономерностей в случайных выборках использовался второй вариант перестановочного теста, т. е. для каждой закономерности вычислялись p-значения p_1 и p_2. Из-за высокой трудоемкости вычислений исследование ограничивалось 50 случайными выборками. Для набора уровней значимости α из отрезка $[0, 0{,}05]$ была рассчитана усредненная по всем 50 выборкам доля пар переменных, для которых были выявлены двумерные закономерности, удовлетворяющие условию $\max\{p_1, p_2\} \leqslant \alpha$. Указанные доли приведены в табл. 2. Значения уровней значимости приведены в столбцах таблицы, озаглавленных α. Соответствующая доля пар переменных приведена в соседнем столбце, озаглавленном ν. В верхней левой ячейке приведена доля пар переменных, удовлетворяющих условию $\max\{p_1, p_2\} < 0{,}0005$. Доли ν являются несмещенными и состоятельными оценками вероятности выполнения неравенства $\max\{p_1, p_2\} \leqslant \alpha$ (или неравенства $\max\{p_1, p_2\} < 0{,}0005$ для верхней левой ячейки) при выполнении условия независимости целевой переменной от вектора X-переменных. Из табл. 2 видно, что усредненная по всем 50 выборкам доля пар X-переменных, для которых выполнено условие

$$\max(p_1, p_2) < 0{,}0005\,, \tag{2}$$

составляет $1{,}18 \cdot 10^{-5}$. Используя данную долю в качестве оценки вероятности выполнения условия (2), получаем вероятность случайного появления хотя бы одной двумерной

Таблица 2 Доли пар переменных, для которых выполняется условие $\max\{p_1, p_2\} \leqslant \alpha$

α	ν	α	ν
$p < 0{,}0005$	$1{,}18 \cdot 10^{-5}$	0,007	$4{,}63 \cdot 10^{-4}$
0,0005	$3{,}35 \cdot 10^{-5}$	0,008	$5{,}57 \cdot 10^{-4}$
0,001	$5{,}71 \cdot 10^{-5}$	0,009	$6{,}28 \cdot 10^{-4}$
0,0015	$8{,}86 \cdot 10^{-5}$	0,01	$7{,}42 \cdot 10^{-4}$
0,002	$1{,}18 \cdot 10^{-4}$	0,012	$9{,}25 \cdot 10^{-4}$
0,0025	$1{,}52 \cdot 10^{-4}$	0,014	$1{,}18 \cdot 10^{-3}$
0,003	$1{,}77 \cdot 10^{-4}$	0,017	$1{,}53 \cdot 10^{-3}$
0,0035	$2{,}1 \cdot 10^{-4}$	0,02	$1{,}98 \cdot 10^{-3}$
0,004	$2{,}48 \cdot 10^{-4}$	0,025	$2{,}68 \cdot 10^{-3}$
0,0045	$2{,}8 \cdot 10^{-4}$	0,03	$3{,}48 \cdot 10^{-3}$
0,0055	$3{,}6 \cdot 10^{-4}$	0,05	$7{,}07 \cdot 10^{-3}$

«закономерности» среди 10011 пар:

$$1 - (1 - 1{,}18 \cdot 10^{-5})^{10011} \simeq 0{,}165\,.$$

Таким образом, «закономерность», удовлетворяющая условию (1), может возникнуть чисто случайно, по крайней мере для одной из пар переменных с вероятностью примерно 16,5%.

Следующей по уровню значимости в табл. 1 является двумерная закономерность, связывающая бинарный показатель VEGF-bin с показателями ОЖСС и S-100. Из табл. 1 видно, что все 6 случаев с ОЖСС $< 39{,}5$ и S-100 $< 86{,}738$ соответствуют высокому уровню VEGF. Наоборот, из 30 случаев с ОЖСС $> 39{,}5$ и S-100 $< 86{,}74$ высокому уровню VEGF соответствует только один случай.

Из табл. 1 также видно, что для данной закономерности $\max(p_1, p_2) = 0{,}002$. Согласно табл. 2 доля пар переменных, для которых выполнено условие

$$\max(p_1, p_2) \leqslant 0{,}002\,, \tag{3}$$

составляет $1{,}18 \cdot 10^{-4}$. Используя данную долю в качестве оценки вероятности выполнения условия (1), получаем вероятность случайного появления хотя бы одной двумерной «закономерности» среди 10011 пар:

$$1 - (1 - 1{,}18 \cdot 10^{-4})^{10011} \simeq 0{,}89\,.$$

Таким образом, «закономерность», удовлетворяющая условию (3), может возникнуть чисто случайно, по крайней мере для одной из пар переменных с вероятностью примерно 89%. Поэтому простое выполнение условий (2) и (3) не является сколь-либо убедительным свидетельством наличия соответствующих закономерностей, обладающих обобщающей способностью, если проводить исследование по полной совокупности наблюдаемых переменных. В целом из анализа табл. 2 можно сделать вывод, что при всеобъемлющем разведывательном анализе данных с размерностью выше 142 нельзя рассчитывать на достоверность даже тех двумерных закономерностей, для которых выполнено условие (2).

Однако нередко интересы исследователей ограничиваются существенно более узкой задачей, сводящейся к оценке характера связи с целевой величиной только какой-то определенной группы показателей или определенного набора сочетаний показателей. Например, двумерные закономерности, представленные в табл. 2, соответствуют оценке влияния на уровень VEGF содержания белков из группы S-100 в сочетании с другими биохимическими и биологическими показателями. Общее число парных сочетаний такого типа очевидно составляет 141. Вероятность случайного появления хотя бы одной двумерной закономерности, удовлетворяющей условию (2) среди 141 пар, будем оценивать точно так же, как ранее оценивалась аналогичная вероятность для 10011 пар, т. е.

$$1 - (1 - 1{,}18 \cdot 10^{-5})^{141} \simeq 0{,}0017 < 0{,}002\,.$$

Таким образом закономерность, связывающую VEGF с S-100 в сочетании с SO2, можно считать значимой на уровне $p < 0{,}02$ после проведения коррекции, учитывающей эффект множественного тестирования. Вероятность случайного появления хотя бы одной двумерной закономерности, удовлетворяющей условию (3) среди 141 пар, соответственно согласно таблице оценивается по формуле:

$$1 - (1 - 1{,}18 \cdot 10^{-4})^{141} \simeq 0{,}0165 < 0{,}02\,.$$

Закономерность, связывающую VEGF с S-100 в сочетании с ОЖСС, можно считать значимой на уровне $p < 0{,}02$ после проведения коррекции, учитывающей эффект множественного тестирования. К сожалению, учет эффекта множественного тестирования не позволяет сделать заключение о достоверности остальных закономерностей из табл. 2. Следует отметить, что простая коррекция по Бонферрони при размере множества пар переменных, в котором осуществляется поиск, равном 141, дает значимость закономерности, связывающей VEGF с S-100 в сочетании с sO2 всего лишь на уровне

$$p < 0{,}0005 \cdot 141 = 0{,}0705\,.$$

Значимость закономерности, связывающей VEGF с S-100 в сочетании с ОЖСС, оценивается на уровне

$$p = 0{,}002 \cdot 141 = 0{,}282\,.$$

Таким образом, обе закономерности оказываются незначимыми.

6 Заключение

Проведенные оценочные расчеты показывают, что двумерная закономерность, полученная с помощью метода ОДР и удовлетворяющая условию $\max(p_1, p_2) < 0{,}0005$, оказывается значимой на уровне $p < 0{,}002$, а двумерная закономерность, полученная с помощью метода ОДР и удовлетворяющая условию $\max(p_1, p_2) < 0{,}002$, оказывается значимой на уровне $p < 0{,}02$ с учетом эффекта множественного тестирования при переборе 141 пары переменных. При этом доля двумерных закономерностей, для которых исходная статистическая значимость, рассчитанная с помощью нескорректированного теста, не хуже фиксированного уровня α оказывается в 10–30 раз ниже величины αN_p, где N_p — число протестированных пар переменных. Для того чтобы анализ оставался достоверным для менее выраженных закономерностей, необходимо ограничивать число просматриваемых пар X-переменных, исходя из формулируемых на начальном этапе целей.

К биологическим результатам исследования следует отнести выявление взаимосвязи уровня белка VEGF с показателями, характеризующими снабжение тканей кислородом, которая, однако, проявляется только в сочетании с содержанием белков из группы S-100.

Литература

[1] *Dunn O. J.* Multiple comparisons among means // J. Am. Stat. Association, 1961. Vol. 56(293). P. 52–64.

[2] *Holm S.* A simple sequentially rejective multiple testprocedure // Scand. J. Stat., 1979. No. 6. P. 65–70.

[3] *Sidak Z. K.* Rectangular confidence regions for the means of multivariate normal distributions // J. Am. Stat. Association, 1967. No. 62(318). P. 626–633.

[4] *Hochberg Y.* A sharper Bonferroni procedure for multiple tests of significance // Biometrika, 1988. Vol. 75. P. 800–802.

[5] *Сенько О. В.* Перестановочный тест в методе оптимальных разбиений // Ж. вычисл. матем. матем. физ., 2003. Т. 43. № 9. С. 1422–1431.

[6] *Senko O., Kuznetsova A.* The optimal valid partitioning procedures // "InterStat" — Statistics in Internet, June 2006. No. 6. http://ip.statjournals.net.

[7] *Kuznetsova A., Kostomarova I., Senko O.* Modification of the method of optimal valid partitioning for comparison of patterns related to the occurrence of ischemic stroke in two groups of patients // Pattern Recogn. Image Anal., 2013. Vol. 22. No. 4. P. 10–25.

[8] *Kim H.-J., Fay M. P., Feuer E. J., Midthune D. N.* Permutation tests for joint point regression with applications to cancer rates // Stat. Medicine, 2000. Vol. 19. No. 3. P. 335–351.

[9] *Ernst M.* Permutation methods: A basis for exact inference // Stat. Sci., 2004. Vol. 19. No. 4. P. 676–685.

[10] *Good P. I.* Permutation, parametric and bootstrap tests of hypotheses. — Springer ser. in statistics. — 3rd ed. — Springer, 2005. 334 p.

[11] *Ojala M., Garriga G.* Permutation tests for studying classifier performance // J. Machine Learning Res., 2010. No. 11. P. 1833–1863.

[12] *Senko O. V., Dzyba D. S., Pigarova E. A., Rozhinskaya L. Ya., Kuznetsova A. V.* A method for evaluating validity of piecewise-linear models // KDIR, 2014. P. 437–443.

[13] *Tusher V. G., Tibshirani R., Chu G.* Significance analysis of microarrays applied to the ionizing radiation response // Proc. Natl. Acad. Sci. USA, 2001. Vol. 98. P. 5116–5121.

[14] *Dudoit S., Popper Shaffer J., Boldrick J. C.* Multiple hypothesis testing in microarray experiments // Stat. Sci., 2003. Vol. 18. No. 1. P. 71–103.

[15] *Sun Y., Jin K., Xie L., Childs J., Mao X. O., Logvinova A., Greenberg D. A.* VEGF-induced neuroprotection, neurogenesis, and angiogenesis after focal cerebral ischemia // J. Clin. Invest., 2003. Vol. 111. No. 12. P. 1843–1851, 976.

[16] *Marenholz I., Heizmann C. W., Fritz G.* S100 proteins in mouse and man: From evolution to function and pathology (including an update of the nomenclature) // Biochem. Biophys. Res. Commun., 2004. Vol. 322. No. 4. P. 1111–1122. doi:10.1016/j.bbrc.2004.07.096. PMID 15336958

[17] *Haymond S.* Oxygen saturation // Clinical Laboratory News, February 2006. No. 10-12. www.aacc.org.

Поступила в редакцию 15.10.2015

Evaluating of multiple testing effect in method of optimal valid partitioning*

***O. V. Senko*[1], *A. M. Morozov*[2], *A. V. Kuznetsova*[3], *and L. L. Klimenko*[4]**

senkoov@mail.ru, alxmopo3ov@gmail.com, azfor@narod.ru, klimenkoll@mail.ru

[1]Federal Research Center "Computer Science and Control" of RAS, 44/2 Vavilova st., Moscow, Russia

[2]Lomonosov Moscow State University, 1 Leninskie Gory, Moscow, Russia

[3]Emanuel Institute of Biochemical Physics RAS, 4 Kosygina st., Moscow, Russia

[4]Semenov Institute of Chemical Physics RAS, 4 Kosygina st., Moscow, Russia

Development of methods for statistically valid regularities discovery is one of the most important data mining problems. One of the possible techniques of regularities search is method of optimal valid partitioning (OVP), using permutation test for statistical verification. In high-dimensional tasks, verification becomes more complicated task due to the problem of multiple testing. Standard Bonferroni correction is based on very strong validity thresholds that rarely are practically achievable when dimension is greater than 100. Set of Monte-Carlo experiments was conducted to evaluate true validity of found regularities in the following biomedical task: study of relationship between vessels growth factor (VEGF) levels and wide set of biological indicators. Set of regularities found in initial data set was compared with sets of regularities

*The research was supported by the Russian Foundation for Basic Research, project No. 14-07-00819.

that were found in 50 random data sets. At that random data sets were generated from initial data set by random permutations of the target variable positions with fixed positions of explanatory variables vectors. It was shown in experiments that fraction of two-dimensional regularities that are valid at uncorrected significance level α is 10-30 times less than αN_p where N_p is the number of enumerated pairs of explanatory variables. Some ways to soft validity thresholds are discussed.

Keywords: *regularities; permutation test; multiple comparing*

DOI: 10.21469/22233792.2.1.03

References

[1] Dunn, O. J. 1961. Multiple comparisons among means. *J. Am. Stat. Association* 56(293):52–64.

[2] Holm, S. 1979. A simple sequentially rejective multiple testprocedure. *Scand. J. Stat.* 6:65–70.

[3] Sidak, Z. K. 1967. Rectangular confidence regions for the means of multivariate normal distributions. *J. Am. Stat. Association* 62(318):626–633.

[4] Hochberg, Y. 1988. A sharper Bonferroni procedure for multiple tests of significance. *Biometrika* 75:800–802.

[5] Senko, O. V. 2003. Perestanovochnyi test v metode optimalnych razbienii. *Zh. Vychisl. Matem. Matem. Fiz.* 43(9):1422–1431.

[6] Senko, O., and A. Kuznetsova. June 2006. The optimal valid partitioning procedures. *"InterStat" — Statistics in Internet* 6. http://ip.statjournals.net.

[7] Kuznetsova, A., I. Kostomarova, and O. Senko. 2013. Modification of the method of optimal valid partitioning for comparison of patterns related to the occurrence of ischemic stroke in two groups of patients. *Pattern Recogn. Image Anal.* 22(4):10–25.

[8] Kim, H.-,J., M. P. Fay, E. J. Feuer, and D. N. Midthune. 2000. Permutation tests for joint point regression with applications to cancer rates. *Stat. Medicine* 19(3):335–351.

[9] Ernst, M. 2004. Permutation methods: A basis for exact inference. *Stat. Sci.* 19(4):676–685.

[10] Good, P. I. 2005. *Permutation, parametric and bootstrap tests of hypotheses*. Springer ser. in statistics. 3rd ed. Springer. 334 p.

[11] Ojala, M., and G. Garriga. 2010. Permutation tests for studying classifier performance. *J. Machine Learning Res.* 11:1833–1863.

[12] Senko, O. V., D. S. Dzyba, E. A. Pigarova, L. Ya. Rozhinskaya, and A. V. Kuznetsova. 2014. A method for evaluating validity of piecewise-linear models. *KDIR* 437–443.

[13] Tusher, V. G., R. Tibshirani, and G. Chu. 2001. Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA* 98:5116–5121.

[14] Dudoit, S., J. Popper Shaffer, and J. C. Boldrick. 2003. Multiple hypothesis testing in microarray experiments. *Stat. Sci.* 18(1):71–103.

[15] Sun, Y., K. Jin, L. Xie, J. Childs, X. Mao, A. Logvinova, and D. A. Greenberg. 2003. VEGF-induced neuroprotection, neurogenesis, and angiogenesis after focal cerebral ischemia. *J. Clin. Invest.* 111(12):1843–1851, 976.

[16] Marenholz, I., C. .W. Heizmann, and G. Fritz. 2004. S100 proteins in mouse and man: From evolution to function and pathology (including an update of the nomenclature). *Biochem. Biophys. Res. Commun.* 322(4):1111–1122. doi:10.1016/j.bbrc.2004.07.096. PMID 15336958

[17] Haymond, S. February 2006. Oxygen saturation. *Clinical Laboratory News* 10-12. www.aacc.org.

Received October 15, 2015

Методы определения характеристик коагуляции и фибринолиза по последовательности изображений фибринового сгустка в плазме крови *in vitro*

***Ю. Д. Бернштейн*[1], *О. С. Брусов*[2], *И. А. Матвеев*[3]**
juliebernshtein@gmail.com, oleg.brusow@yandex.ru, matveev@ccas.ru
[1]Московский физико-технический институт, г. Долгопрудный, Институтский пер., 9
[2]ФГБНУ Научный центр психического здоровья, г. Москва, Москва, Каширское шоссе, 34
[3]ФИЦ «Информатика и управление» РАН, г. Москва, ул. Вавилова, 44/2

Решается задача количественного определения характеристик фибринового сгустка в методе тромбодинамики. Исходными данными являются последовательности цифровых снимков кюветы в регистраторе тромбодинамики, наполненной плазмой крови, в которой происходит рост и рассасывание сгустка фибрина от вставки-активатора, сделанных через равные интервалы времени. Определяются границы активатора, скорости роста и рассасывания фибринового сгустка, изменение его размеров и плотности во времени, а также момент отрыва сгустка от активатора. Для выделения сгустков на изображении применяются бинаризация, математическая морфология и метод проекций. Совокупность измеряемых параметров и их временная динамика могут быть использованы в целях медицинской диагностики потенциалов фибринолиза и коагуляции.

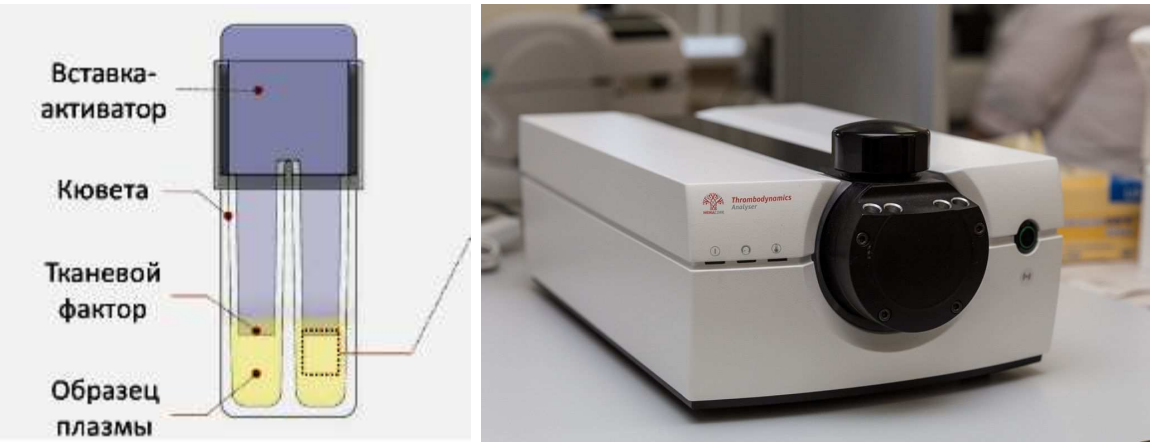
Ключевые слова: *тромбодинамика; математическая морфология; метод проекций изображения; алгоритм бинаризации с адаптивным порогом*

DOI: 10.21469/22233792.2.1.04

1 Введение

Важным методом медицинской диагностики является выявление нарушений в системе гемостаза [1, 2]. Одним из новых подходов здесь является *метод тромбодинамики* — изучение роста фибринового сгустка в плазме крови, разработанный в лаборатории физической биохимии ГНЦ РАМН [3, 4]. С помощью тромбодинамики можно выявить склонность свертывающей системы крови пациента к гипо- и гиперкоагуляции (кровоточивости и тромбообразованию) и осуществить персонализированный подбор антикоагулянтной терапии, необходимой при лечении и профилактике тромбозов [5].

Процесс тромбодинамики протекает в кювете (рис. 1, *а*), расположенной в передней части регистратора (рис. 1, *б*). В кювету помещается вставка, на торец которой нанесен *активатор свертывания* — тканевой фактор. Как только плазма крови соприкасается с активатором, стартует процесс свертывания, и от торца вставки в объем плазмы начинает расти фибриновый сгусток, так же как на поврежденной стенке сосуда в организме (*in vivo*). Очевидными характеристиками свертывания являются плотность сгустка (его средняя яркость на изображении) и скорость роста (распространения в пространстве). В перечисленных работах доказана связь этих параметров с состоянием системы тромбообразования. Однако не менее важным является тромболизис (фибринолиз) — процесс растворения сгустков фибрина [6], поэтому предлагается изучать не только процесс роста тромба, но и процесс его растворения. Для этого кювета разделена на два канала: в первом находится чистая плазма крови пациента, по которой изучается процесс коагуляции, в плазму во втором канале вводится *тканевой активатор плазминогена* (ТАП), запускающий процесс фибринолиза. В условиях эксперимента рост тромба в кювете начинается



(*а*) Кювета с активатором (*б*) Регистратор тромбодинамики

Рис. 1 Установка

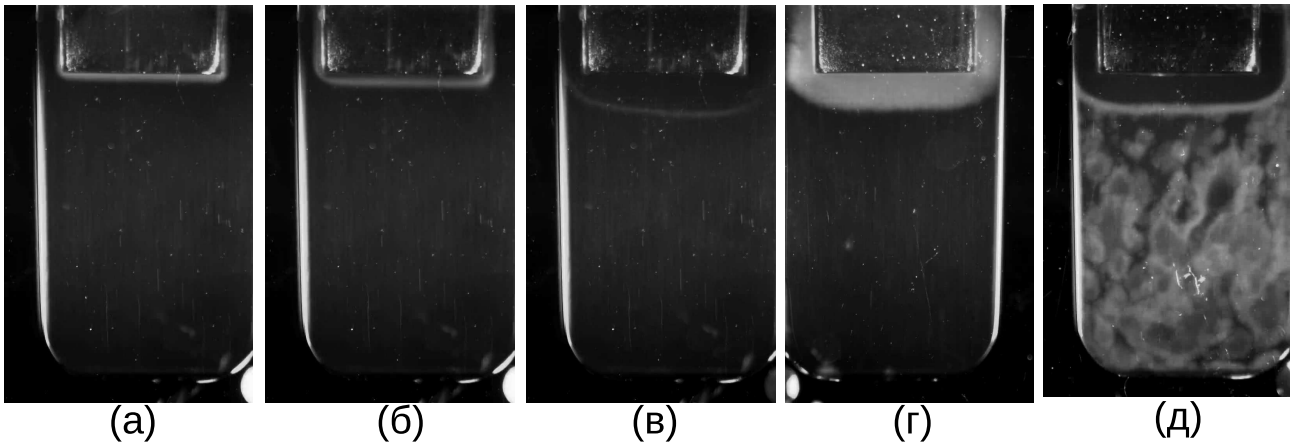


Рис. 2 Примеры изображений тромбов: (*а*) начало роста тромба (10-й кадр видеопоследовательности); (*б*) отрыв тромба от активатора в результате фибринолиза (70-й кадр); (*в*) движение тромба в виде слоя (400-й кадр); (*г*) изображение плазмы в первом канале (без ТАП, 400-й кадр); (*д*) возникновение спонтанных сгустков

от вставки-активатора (рис. 2, *а*). Растворение тромба начинается также от активатора (рис. 2, *б*). В результате в кювете формируется тромб в форме слоя некоторой толщины, с течением времени удаляющегося от активатора (рис. 2, *в*). В первом канале (без ТАП) процесс фибринолиза не запускается (рис. 2, *г*). Применение метода осложняется возникновением *спонтанных сгустков*, которые искажают и разрушают движение тромба (рис. 2, *д*).

Весь процесс тромбодинамики регистрируется цифровой камерой, делающей снимки через заданные (как правило, равные) интервалы времени. Камера и кювета неподвижны в течение всего эксперимента, который длится 30–60 мин. Таким образом, исходными данными являются последовательности изображений кюветы и находящегося в ней фибринового сгустка.

Методы цифровой обработки изображений активно и плодотворно применяются в медицине [7, 8] в различных областях диагностики. Задача автоматического определения плотности и скорости роста и рассасывания тромба по последовательности изображений

также является интересным приложением. Ранее в составе системы «Гемакор» [9] был разработан метод оценки величины коагуляционного потенциала, основанный на средней яркости изображения в первом канале. Это устойчивый к шумам, но недостаточно точный метод. Этим методом определяется лишь интегральная характеристика — масса тромба (как общая его яркость), в то время как более информативными представляются скорость роста и плотность. Кроме того, находятся лишь параметры коагуляции, но не фибринолиза. В данной работе предлагается метод автоматического определения перемещения переднего и заднего фронтов тромба, что позволяет получить раздельно сведения о скорости роста и плотности тромба, а также о скорости фибринолиза.

2 Постановка задачи

На вход поступает последовательность из $N+1$ монохромных изображений одинакового размера $W \times H$ пикселей. Зададим начало координат в верхнем левом углу изображения, оси абсцисс Ox и ординат Oy направлены вправо и вниз соответственно. Изображение с номером n является функцией двух дискретных целочисленных аргументов $I_n = I_n(x, y)$, $x, y \in \mathbf{Z}$, $x \in \overline{0; W-1}$, $y \in \overline{0; H-1}$, т. е. матрицей. Значениями функции (элементами матрицы) являются целые положительные числа — яркости соответствующих пикселей изображения. Последовательность изображений можно рассматривать как функцию трех дискретных целочисленных аргументов $I(x, y, n)$, $n \in \mathbf{Z}$, $n \in \overline{0; N}$. Тромб на изображении растет от активатора вниз, т. е. в сторону увеличения ординаты.

Требуется определить следующие характеристики $I(x, y, n)$:

- положение переднего фронта тромба $y_F(n)$ (ординату) на каждом кадре последовательности;
- момент отрыва тромба от активатора, т. е. номер кадра n_d, на котором появляется зазор между активатором и тромбом;
- положение заднего фронта тромба $y_B(n)$ на каждом кадре последовательности начиная с момента отрыва, $n > n_d$;
- получить значения положений заднего и переднего фронтов на каждом кадре последовательности, выраженные в микронах и полученные умножением соответствующих ординат на изображениях на некий линейный коэффициент;
- среднее значение яркости $b(n)$ между передним и задним фронтами на каждом кадре.

Для точного решения этих задач также требуется определить положение активатора в кадре. На любой заданной последовательности активатор неподвижен, т. е. имеет одинаковые координаты на всех ее изображениях. Однако на разных последовательностях его положение может меняться, поэтому положение активатора определяется для каждой последовательности по ее первому кадру. Часть активатора, видимая на изображении, представляет собой прямоугольник в верхней его четверти со сторонами, приблизительно параллельными осям изображения. Обозначим координаты активатора как x_L и x_R — левая и правая стороны, y_{act} — нижняя сторона (рис. 3). Красными линиями на рисунке обозначены границы активатора, синей и зеленой — положения заднего и переднего фронтов роста сгустка соответственно.

Также на изображениях видна граница кюветы: правая — в первом канале (см. рис. 2, *г*) и левая — во втором канале (см. рис. 2, *а* – 2, *в* и 2, *д*). Однако в отличие от активатора границу кюветы не требуется определять с высокой точностью, поэтому выбраны два фиксированны значения $x_{\text{cuv}}^{\text{left}} = 120$ и $x_{\text{cuv}}^{\text{right}} = 570$. Содержимое изображений первого канала с координатами больше $x_{\text{cuv}}^{\text{right}}$ и второго канала с координатами меньше $x_{\text{cuv}}^{\text{left}}$

Рис. 3 Определяемые величины на кадре последовательности

игнорируется. Далее будем считать ширину изображений W и все операции с абсциссой для изображений, урезанных согласно этому правилу.

3 Метод решения

При обработке набора изображений выполняются следующие действия:

- поиск границ активатора на начальном изображении;
- определение момента отрыва;
- последовательное определение положений переднего и заднего фронтов сгустка на каждом изображении набора снимков, или детектирование ситуации появления спонтанных сгустков, приводящей к невозможности обработки оставшихся изображений набора;
- определение яркости сгустка.

3.1 Поиск границ активатора

Для определения границ активатора используется поиск максимума проекций бинаризованного изображения. Бинаризация осуществляется с порогом, выбранным по гистограмме яркостей. На начальном изображении последовательности $I_0(x, y)$ выделяется область — верхняя четверть, в которой находится активатор: $y \in [0; H/4]$. В этой области собирается интегральная гистограмма яркости:

$$H(b) = |\{(x, y) : I_0(x, y) \leqslant b\}| \,.$$

Порог устанавливается как величина, отсекающая 10% самых ярких точек верхней четверти изображения:

$$\theta : H(\theta) = 0{,}9\frac{WH}{4} = 0,9H(L)\,,$$

где L — максимальная яркость изображения. Также можно использовать порог, вычисляемый методом Оцу [10]. Бинаризация изображения по порогу θ:

$$I_0^{\text{Bin}}(x, y) = \begin{cases} 1, & \text{если } I_0(x, y) \geqslant \theta\,; \\ 0, & \text{если } I_0(x, y) < \theta\,. \end{cases} \tag{1}$$

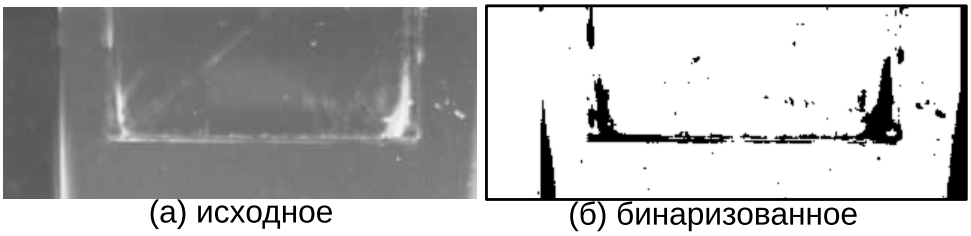


Рис. 4 Бинаризация с порогом $\theta = 15000$ для изображения, значения интенсивности которого лежат в промежутке $[0; 65535]$

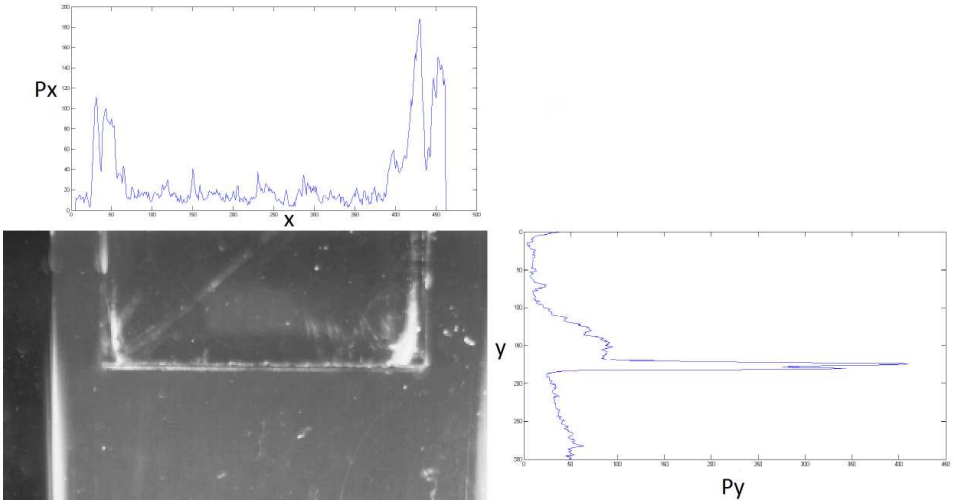


Рис. 5 Фрагмент изображения I_0 и его проекции

Пример преобразования с помощью бинаризации показан на рис. 4, на бинаризованном изображении здесь и далее для наглядности белый цвет соответствует нулевым элементам, черный — единичным.

Для бинаризованного изображения строим горизонтальную проекцию (на ось ординат)

$$P_x(x) = \sum_{y=0}^{H/4-1} I_0^{\text{Bin}}(x, y)$$

и вертикальную проекцию (на ось абсцисс)

$$P_y(y) = \sum_{x=0}^{W-1} I_0^{\text{Bin}}(x, y)\,. \tag{2}$$

Пример изображения и построенных проекций дан на рис. 5.

Проекции сглаживаются усреднением в скользящем окне с полушириной $w = 5$:

$$P^{\text{Blur}}(x) = \frac{1}{2w+1} \sum_{\xi=-w}^{w} P(x+\xi) I(x+\xi \geqslant 0) I(x+\xi \leqslant W-1)\,. \tag{3}$$

Координаты левой и правой границ активатора определяются как положения максимумов в левой и правой половинах сглаженной горизонтальной проекции:

$$x_L = \arg \max_{x \in [0; W/2)} P_x^{\text{Blur}}(x)\,; \;\; x_R = \arg \max_{x \in (W/2; W]} P_x^{\text{Blur}}(x)\,,$$

координата нижней границы — как положение максимума сглаженной аналогичным образом вертикальной проекции:

$$y_{\text{act}} = \arg \max_{y \in (0;H/4)} P_y^{\text{Blur}}(y)\,. \tag{4}$$

3.2 Определение момента отрыва

При анализе исходных данных было отмечено, что отрыв сгустка совпадает по времени со значительным уменьшением его яркости. При этом все время до отрыва яркость (как средняя, так и общая) сгустка нарастает. Используя эту особенность, удалось построить следующий простой способ определения момента отрыва. В прямоугольной области изображения, ограниченной сверху нижней границей активатора y_{act}, справа и слева — положениями его краев x_L и x_R и имеющей некоторую заданную (эмпирически выбранную) высоту $h = 10$, вычисляется средняя яркость:

$$b(n) = \frac{1}{(h+1)(x_R - x_L + 1)} \sum_{y=y_{\text{act}}}^{y_{\text{act}}+h} \sum_{x=x_L}^{x_R} I_n(x,y)\,.$$

Значение $b(n)$ монотонно возрастает с ростом n вплоть до момента отрыва. В момент отрыва n_d это значение падает: $b(n_d) < b(n_d - 1)$.

3.3 Определение положений фронтов

В течение эксперимента условия съемки остаются постоянными, поэтому все изменения изображений последовательности обусловлены ростом и растворением тромбов, появлением спонтанных сгустков, а также шумами. В начальный момент времени (на первом в последовательности изображении) тромба нет, и это изображение можно использовать как фоновое, т. е. получать изображение тромба, вычитая его из всех последующих (рис. 6, *а*):

$$\tilde{I_n}(x,y) = I_n(x,y) - I_0(x,y). \tag{5}$$

Бинаризуя (5) согласно (1), получаем новую последовательность изображений $\tilde{I}_n^{\text{Bin}}(x,y)$, область тромба, а также малые шумовые области (рис. 6, *б*).

К каждому изображению из полученной последовательности применяется морфологическая операция размыкания (opening) по примитиву SE, являющимся квадратом размерами 11×11, заполненный единицами. Морфологическое размыкание состоит из двух базовых операций: дилатации и эрозии.

Определение 1. *Для множеств* $\tilde{I}_n^{\text{Bin}}$ *и* SE *эрозия* $\tilde{I}_n^{\text{Bin}}$ *по* SE *определяется как*

$$\tilde{I}_n^{\text{Bin}}(x,y) \ominus \text{SE} = \min_{i \in [-k,k]} \tilde{I}_n^{\text{Bin}}(x+i, y+i)\,.$$

Определение 2. *Для множеств* $\tilde{I}_n^{\text{Bin}}$ *и* SE *дилатация* $\tilde{I}_n^{\text{Bin}}$ *по* SE *определяется как*

$$\tilde{I_n}^{\text{Bin}} \oplus \text{SE} = \max_{i \in [-k,k]} \tilde{I}_n^{\text{Bin}}(x+i, y+i)\,.$$

Определение 3. *Размыкание множества* $\tilde{I_n}^{\text{Bin}}$ *по* SE *определяется как*

$$\tilde{I}_n^{\text{Open}} = \tilde{I}_n^{\text{Bin}} \circ \text{SE} = (\tilde{I}_n^{\text{Bin}} \ominus \text{SE}) \oplus \text{SE}\,.$$

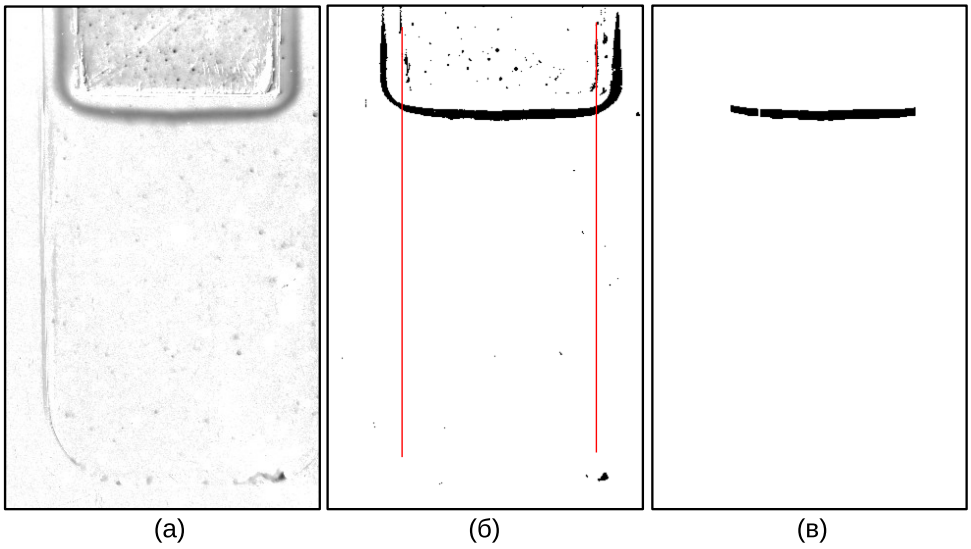


Рис. 6 Пример применения морфологических операций: (*а*) разность текущего изображения (60-й кадр) и начального; (*б*) бинаризованное разностное изображение и границы x_L и x_R; (*в*) морфологическая фильтрация, вырезание области внутри границ

При помощи морфологической операции размыкания шумы на изображениях удаляются. Таким образом, получили последовательность изображений $\tilde{I}_n^{\text{Open}}$. Также необходимо отметить, что движение тромба представляет интерес в центральной части кюветы, а именно: в прямоугольнике, ограниченном с боков размерами активатора, поэтому на изображении вырезается область $\Omega = \{(x, y) : x \in [x_L; x_R]\}$. Результат этих операций показан на рис. 6, *в*. Применив вертикальную проекцию (2), сглаживание (3) и поиск максимума, аналогичный (4), получаем приближенно абсциссу центра тромба y_0.

На рис. 6 видно, что тромб, даже ограниченный областью Ω, слегка изогнут, поэтому непосредственным применением метода проекций затруднительно получить точные положения переднего и заднего фронтов. С целью получить точные значения сделана следующая модификация метода проекций. Анализ исходных данных показал, что в начале развития тромба его границы являются практически прямыми линиями, повторяя прямую линию нижней стороны активатора. Однако с удалением тромба от начального положения он все больше изгибается. Точная модель этой деформации не построена (это одно из направлений дальнейшей работы). В данной работе принята приближенная модель параболической формы изгиба. Предполагается, что центральная линия тромба имеет форму:

$$y = y_0 - a(y_0)\left(x - \frac{x_R + x_L}{2}\right)^2. \tag{6}$$

Вершина параболы находится в точке $((x_L + x_R)/2, y_0)$, расположенной на оси симметрии активатора. Коэффициент уравнения параболы $a(y_0) = 0,5 \cdot 10^{-4}(y_0 - y_{\text{act}})$ линейно зависит от расстояния тромба от активатора, коэффициент пропорциональности подобран эмпирически. Применив вертикальную проекцию (2), где значение y в правой части берется из (6), к изображению вертикальных градиентов $V(x, y) = I(x, y + 1) - I(x, y - 1)$, получим в окрестности y_0 локальный минимум на месте переднего фронта (концентрация отрицательных вертикальных градиентов на переходе от яркого тромба к темному фону) и локальный максимум на месте заднего фронта.

3.4 Определение яркости сгустка

Яркость тромба определяется как средняя яркость пикселей области, ограниченной вертикальными прямыми, проходящими по левому и правому краям активатора и параболическими приближениями переднего и заднего фронтов, т. е. удовлетворяющих условиям:

$$\begin{cases} x \geqslant x_L\,; \\ x \leqslant x_R\,; \\ y \geqslant y_B + a(y_B)\left(x - \dfrac{x_R + x_L}{2}\right)^2\,; \\ y \leqslant y_F + a(y_F)\left(x - \dfrac{x_R + x_L}{2}\right)^2\,. \end{cases}$$

4 Заключение

Разработаны алгоритмы автоматического определения характеристик тромба по его изображениям. Определяются такие характеристики, как положения фронтов сгустка, средняя яркость сгустка, моменты начала фибринолиза и появления спонтанных сгустков. Разработан алгоритм определения границы активатора. Автоматическая обработка данных тромбодинамики позволит заменить ручную оценку на количественную автоматизированную, основанную на вычисляемых характеристиках. Задача выделения обоих фронтов сгустка в эксперименте с ТАП поставлена и решена впервые. Расчеты произведены для 69 последовательностей изображений тромбодинамики, в последовательностях от 300 до 450 изображений. Ручная проверка правильности автоматического определения изучаемых характеристик не выявила расхождений с мнением человека-эксперта.

Пример проведенных вычислений для последовательности из 450 снимков показан на рис. 7.

Момент отрыва сгустка от активатора $n_d = 54$.

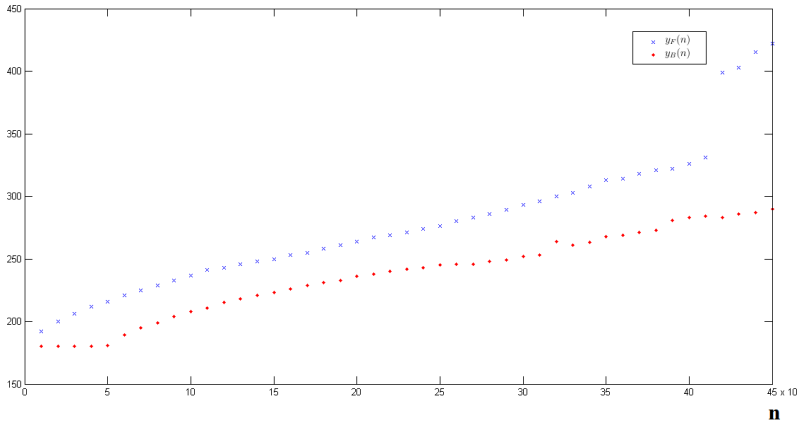


Рис. 7 График зависимости положений фронтов сгустка ($а$) и яркости снимков ($б$) от номера кадра

Литература

[1] He S., Antovic M., Blomback M. A simple and rapid laboratory method for determination of haemostatic potential in plasma. Modification for use in routine laboratories and research work // Thromb. Res., 2001. Vol. 103. No. 5. P. 255–361.

[2] Curnow J., Morel-Kopp M.-C., Roddie C., Aboud M., Ward C. M. Reduced fibrinolytic and increased fibrin generation can be detected in hypercoagulable patients using the overall hemostatic potential assay // J. Thromb. Haemost., 2007. Vol. 5. No. 3. P. 528–534.

[3] Hemker H., Ataullakhanov F. Good mathematical practice: Simulation of the hemostatic-thrombotic mechanism, a powerful tool but one that must be used with circumspection // Pathophysiol. Haemo. Thromb., 2005. Vol. 43. No. 2-3. P. 55–57.

[4] Soshitova N., Karamzin S., Balandina A., et al. Predicting prothrombotic tendencies in sepsis using spatial clot growth dynamics // Blood Coagul. Fibrin., 2012. Vol. 23. No. 6. P. 498–507.

[5] Panteleev M., Balandina A., Lipets E. Taskoriented modular decomposition of biological networks: Trigger mechanism in blood coagulation // Biophys. J., 2010. Vol. 98. P. 1751–1761.

[6] Antovic A. Screening haemostasis — looking for global assays: The overall haemostasis potential (ohp) method — a possible tool for laboratory investigation of global haemostasis in both hypo- and hypercoagulable conditions // Curr. Vasc. Pharmacol., 2008. Vol. 6. No. 3. P. 173–185.

[7] Gonzalez R. C., Woods R. E. Digital image processing. — 2nd Ed. — Prentice Hall, 2002. URL: `http://www.imageprocessingplace.com/DIP/dip_book_description/book_description.htm`.

[8] Angeneat S., Pichon E., Tannenbaum A. Mathematical methods in medical image processing // Bull. Amer. Math. Soc., 2006. No. 43. P. 365–396.

[9] Hemacore. `http://www.hemacore.com/`.

[10] *Otsu N.* A threshold selection method from gray-level histograms // IEEE Trans. Syst. Man Cyb., 1979. Vol. 9. No. 1. P. 62–66. doi: 10.1109/TSMC.1979.4310076.

Поступила в редакцию 05.08.2015

Methods for in vitro determination of coagulation and fibrinolysis characteristics using the blood plasma images sequence

***J. D. Bernshtein*[1], *O. S. Brusov*[2], *and I. A. Matveev*[3]**

`juliebernshtein@gmail.com, oleg.brusow@yandex.ru, matveev@ccas.ru`

[1]Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow, Russia

[2]Mental Health Research Center, 34 Kashirskoye shosse, Moscow, Russia

[3]Federal Research Center "Computer Science and Control" of RAS, 44/2 Vavilova st., Moscow, Russia

The problem of quantifying the characteristics of a fibrin clot in the thrombodynamics method is being solved. The initial data of the method are the sequences of digital images of the cell filled with blood plasma and located in the thrombodynamics registrar, in which the clot is growing and resorbing from the activator, made at regular intervals. Activator's boundaries, speed of growth and resorption of the fibrin clot, time changes of its size and density, and the moment of the clot's separation from the activator are determined. Methods of binarization, mathematical morphology, and image projections are used to select clots in the image. The set of measured parameters and their temporal dynamics may be used for medical diagnostic potential of fibrinolysis and coagulation.

Keywords: *thrombus dynamics; mathematical morphology; image projection method; binarization algorithm with adaptive threshold*

DOI: 10.21469/22233792.2.1.04

References

[1] He, S., M. Antovic, and M. Blomback. 2001. A simple and rapid laboratory method for determination of haemostatic potential in plasma. Modification for use in routine laboratories and research work. *Thromb. Res.* 103(5):255–361.

[2] Curnow, J., M.-C. Morel-Kopp, C. Roddie, M. Aboud, and C. M. Ward. 2007. Reduced fibrinolytic and increased fibrin generation can be detected in hypercoagulable patients using the overall hemostatic potential assay. *J. Thromb. Haemost.* 5(3):528–534.

[3] Hemker, H., and F. Ataullakhanov. 2005. Good mathematical practice: Simulation of the hemostatic-thrombotic mechanism, a powerful tool but one that must be used with circumspection. *Pathophysiol. Haemo. Thromb.* 34(2-3):55–57.

[4] Soshitova, N., S. Karamzin, A. Balandina, *et al.* 2012. Predicting prothrombotic tendencies in sepsis using spatial clot growth dynamics. *Blood Coagul. Fibrin.* 23(6):498–507.

[5] Panteleev, M., A. Balandina, and E. Lipets. 2010. Taskoriented modular decomposition of biological networks: Trigger mechanism in blood coagulation. *Biophys. J.* 98:1751–1761.

[6] Antovic, A. 2008. Screening haemostasis — looking for global assays: The overall haemostasis potential (ohp) method — a possible tool for laboratory investigation of global haemostasis in both hypo- and hypercoagulable conditions. *Curr. Vasc. Pharmacol.* 6(3):173–185.

[7] Gonzalez, R. C., and R. E. Woods. 2002. *Digital image processing.* 2nd ed. Prentice Hall. Available at: `http://www.imageprocessingplace.com/DIP/dip_book_description/book_description.htm` (accessed May 19, 2016).

[8] Angeneat, S., E. Pichon, and A. Tannenbaum. 2006. Mathematical methods in medical image processing. *Bull. Amer. Math. Soc.* 43:365–396.

[9] Hemacore. Available at: `http://www.hemacore.com/` (accessed May 19, 2016).

[10] Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cyb.* 9(1):62–66. doi: 10.1109/TSMC.1979.4310076.

Received August 5, 2015

Построение полного решающего дерева с использованием гетерогенной системы на основе технологии CUDA*

И. Е. Генрихов
ingvar1485@rambler.ru
«Мобайл парк ИТ», г. Химки, ул. Панфилова, 21/1

Статья посвящена исследованию алгоритмов классификации на основе полных решающих деревьев (ПРД). Рассматриваемая конструкция решающего дерева (РД) позволяет в каждой специальной вершине дерева учитывать все признаки, удовлетворяющие критерию ветвления. Основным недостатком ПРД является существенно большее время синтеза дерева по сравнению с классическим РД. Рассмотрены вопросы снижения времени построения ПРД с использованием технологии CUDA (Compute Unified Device Architecture). Данная технология позволяет использовать большое число ядер графического процессора для ускорения выполнения сложных вычислений. Приведены результаты тестирования на модельных и реальных задачах. Показано, что применение технологии CUDA позволяет заметно снизить время синтеза ПРД (более чем в 10 раз) только в тех случаях, когда обучающая выборка содержит большое число признаков и/или обучающих объектов и при этом информация либо вещественнозначная, либо целочисленная большой значности.

Ключевые слова: *задача распознавания по прецедентам; полное решающее дерево; CUDA; гетерогенная система*

DOI: 10.21469/22233792.2.1.05

1 Введение

Одним из известных инструментов для решения задач обучения по прецедентам [1] являются РД. Процедура построения классического РД представляет собой итерационный процесс, на каждом шаге которого для построения очередной внутренней вершины РД выбирается признак, наилучшим образом удовлетворяющий некоторому критерию ветвления. По каждой ветви, исходящей из построенной внутренней вершины, осуществляется спуск и строится либо лист дерева, либо новая внутренняя вершина. Каждому листу в РД приписан один из классов и, как правило, в листе содержится вся информация, позволяющая сделать вывод о принадлежности распознаваемого объекта классу, который приписан данному листу.

Очевидным недостатком классической модели РД является то, что на очередной итерации для построения внутренней вершины среди всех признаков, удовлетворяющих выбранному критерию ветвления в равной или почти равной мере, выбирается только один признак (и выбирается этот признак фактически случайным образом). При этом в зависимости от выбранного признака построенные деревья могут существенно отличаться по своим распознающим качествам. Для решения данной проблемы в [2] предложен новый подход к синтезу РД. При возникновении ситуации, когда два или более признака удовлетворяют критерию ветвления в равной или почти равной мере, предлагается проводить ветвление по каждому из этих признаков независимо. Полученная в результате конструкция названа ПРД.

*Работа выполнена при финансовой поддержке РФФИ, проект № 13-01-00787.

Таким образом, в отличие от классического РД в ПРД на каждой итерации строится специальная вершина, называемая полной вершиной, которой соответствует набор признаков X. Далее по аналогии с классическим РД проводится ветвление по каждому из признаков, входящих в X. Конструкция ПРД позволяет более существенно использовать имеющуюся информацию, при этом описание распознаваемого объекта может порождаться не одной ветвью, как в классическом РД, а несколькими ветвями. Каждая такая ветвь участвует в процедуре голосования (является голосующей).

Первоначальная модель ПРД [2] предназначалась для обработки целочисленной информации, при этом использовались логические критерии ветвления и самый простой вид коллективного голосования (голосования по большинству). Более совершенные модели ПРД с энтропийным критерием ветвления, использующие взвешенное голосование по голосующим ветвям дерева, были построены и исследованы в [3, 4]. На их основе были построены алгоритмы синтеза ПРД для обработки вещественнозначной информации с наличием пропусков в признаковых описаниях объектов и неравномерного распределения объектов по классам в обучающей выборке (в этом случае можно указать пару классов таких, что число обучающих объектов в одном из них существенно больше числа обучающих объектов в другом). В [3, 4] получены теоретические и экспериментальные оценки, характеризующие высокую обобщающую способность ПРД по сравнению с обобщающей способностью классического РД. Счет на реальных задачах показал, что классификаторы на основе ПРД не уступают по качеству другим современным классификаторам на основе РД, например таким, как «бустинг» над РД и «баггинг» над РД, а иногда показывают и более высокое качество.

Основным недостатком ПРД является существенно большее время синтеза дерева по сравнению с классическим РД. Теоретические оценки времени синтеза ПРД в худшем и в некоторых важных частных случаях были получены в [3]. Поэтому актуальной задачей является снижение временно́й сложности построения ПРД.

В данной работе рассмотрены вопросы снижения времени синтеза ПРД с применением параллельных вычислений на основе технологии CUDA — программно-аппаратная архитектура унифицированных вычислений [5, 6] от компании NVIDIA. Указанная технология позволяет выполнить операции, не требующие длительного времени, на центральном процессоре (CPU) компьютера, а все сложные операции (в вычислительном плане) — на графическом процессоре компьютера (GPU). При таком способе реализации алгоритма принято говорить о применении гетерогенной системы, так как используются ресурсы CPU и GPU для ее выполнения. Графический процессор состоит из однородных вычислительных элементов (мультипроцессоров) с общей памятью. Каждый мультипроцессор способен исполнять параллельно тысячи вычислительных «нитей». Нити могут быть сгруппированы в вычислительные потоки, имеющие общий кэш и быструю разделяемую память для обмена данными между нитями потока. Применение гетерогенных вычислений с использованием GPU наиболее эффективно при решении задач, обладающих параллелизмом по данным, число арифметических операций в которых велико по сравнению с операциями над памятью. Существует много таких задач в различных областях: обработка сигналов, физика, data mining, deep learning, machine learning, вычислительная биология, медицина, биоинформатика, вычислительная гидродинамика, компьютерное видение и работа с изображениями, медиа и развлечения, медицинская визуализация, молекулярная динамика, численный анализ, квантовая химия и др. Также основными достоинствами технологии CUDA является свободный доступ к программным инструментам, позволяющим

реализовать алгоритмы с применением технологии CUDA, и доступность графических ускорителей.

В разд. 2 введены основные понятия и описана общая схема построения ПРД.

В разд. 3 описаны разработанные алгоритмы синтеза ПРД с применением технологии CUDA.

В разд. 4 на модельных данных и на реальных задачах из репозитория UCI [7] и коллекции задач [1], собранной в отделе математических проблем распознавания и методов комбинаторного анализа Вычислительного центра им. А. А. Дородницына РАН Федерального исследовательского центра «Информатика и управление» РАН (ВЦ РАН), протестированы разработанные алгоритмы синтеза ПРД на основе гетерогенных вычислений и выявлены особенности их применения на различных типах задач по сравнению с алгоритмом синтеза ПРД, полностью исполняемом на CPU.

2 Основные понятия

Рассматривается задача распознавания по прецедентам с системой признаков $\{x_1, \ldots, x_n\}$, с непересекающимися классами $K_i,\ i \in I = \{1, \ldots, l\},$ и множеством обучающих объектов $T = \{S_1, \ldots, S_m\}$, где $S_r = (a_{r1}, \ldots, a_{rn}),\ a_{rj} \in \{\mathbb{R}, \text{«*»}\},\ r \in \{1, \ldots, m\},$ $j = 1, \ldots, n$. Если $a_{rj} =$«*», то значение признака x_j для объекта S_r не определено. Пусть далее $S = (b_1, \ldots, b_n)$ — распознаваемый объект и $b_j \in \{\mathbb{R}, \text{«*»}\},\ j = 1, \ldots, n.$

Опишем структуру ПРД. Пусть $\hat{T}$ — подмножество обучающих объектов и $X(\hat{T})$ — подмножество признаков, рассматриваемых на текущем шаге построения дерева. На первом шаге $\hat{T} = T,\ X(\hat{T}) = \{x_1, \ldots, x_n\}$.

При построении ПРД могут встречаться три типа вершин: висячие, полные и обычные вершины.

Определение 1. *Ветвью в ПРД называется путь, начинающийся в корне дерева и заканчивающийся в вершине ПРД.*

Определение 2. *Вершины ПРД, не имеющие выходящих дуг, называются висячими вершинами или листьями.*

Определение 3. *Обычной вершиной в ПРД называется внутренняя вершина ПРД, для которой выполняются следующие условия:*

1. *Данной вершине соответствует ровно один признак $x \in \{x_1, \ldots, x_n\}$.*
2. *В данную вершину входит одна дуга и выходит не менее двух дуг, помеченных разными числами.*
3. *Каждая дуга, выходящая из данной вершины, входит либо в висячую вершину, либо в полную вершину ПРД.*

Определение 4. *Полной вершиной в ПРД называется внутренняя вершина ПРД, для которой выполняются следующие условия:*

1. *Данной вершине ν соответствует набор различных признаков $X_\nu = \{x_{j_1}, \ldots, x_{j_q}\}, X_\nu \subseteq$ $\subseteq X, q \geqslant 1$.*
2. *В данную вершину ν входит одна дуга и выходит ровно q дуг с метками, равными номерам признаков из X_ν.*
3. *Каждая дуга с меткой $t, t \in \{j_1, \ldots, j_q\}$, выходящая из данной вершины, входит в обычную вершину, соответствующую признаку $x_t, x_t \in X_\nu$.*

Определение 5. *Глубиной ветви в ПРД называется число обычных вершин, которые содержит эта ветвь, исключая концевую вершину ветви.*

Определение 6. *Ярусом i-го уровня (i-м ярусом) в ПРД называется совокупность полных и обычных вершин, порожденных ветвями с глубиной $i-1$, а также совокупность листьев дерева, порожденных ветвями с глубиной i.*

На каждом шаге синтеза ПРД строится либо висячая вершина дерева, либо формируется набор из различных признаков $X_\nu, X_\nu \subseteq X$, образующий полную вершину ν. Далее из полной вершины ν строится ровно $|X_\nu|$ дуг с метками $j_1, \ldots, j_q$. Дуга с меткой $t, t \in \{j_1, \ldots, j_q\}$, входит в обычную вершину, соответствующую признаку $x_t, x_t \in X_\nu$. При ветвлении из обычной вершины, соответствующей признаку x_t, происходит удаление признака x_t из $X(\hat{T})$ и удаление некоторых объектов из $\hat{T}$.

В данной работе рассматривается задача распознавания с веществennозначными признаками, поэтому используется следующий способ ветвления из обычной вершины [3]. Для ветвления из обычной вершины, соответствующей признаку $x_t, x_t \in X(\hat{T})$, осуществляется бинарная перекодировка текущих значений признака x_t с помощью «оптимального» порога $d(x_t)$. Рассматриваемая вершина помечается парой $(x_t, d(x_t))$. Спуск из вершины $(x_t, d(x_t))$ происходит по двум ветвям, при этом левая ветвь помечается 0, а правая — 1. При спуске из вершины $(x_t, d(x_t))$ по левой (правой) ветви происходит удаление признака x_t из $X(\hat{T})$ и удаляются те объекты из $\hat{T}$, для которых значение признака x_t больше (не больше) $d(x_t)$.

Рассмотрим решающее правило при классификации распознаваемого объекта S с помощью ПРД.

Пусть v — висячая вершина, ей может быть приписана пара $\left(B_v, \left\{\omega_v^1, \ldots, \omega_v^l\right\}\right)$ [3], где B_v — элементарная конъюнкция (э.к.) над переменными $x_1, \ldots, x_n$; ω_v^i — оценка принадлежности объекта S классу $K_i, i \in I$, вносимая вершиной v.

В данной работе используется следующий способ вычисления вектора оценок в висячей вершине v [3]. Пусть m_v^i — число объектов класса K_i, описание которых попадает в интервал истинности конъюнкции B_v; m^i — число объектов класса K_i в исходной обучающей выборке. Тогда $\omega_v^i = (m_v^i + 1)/(m^i + l), i \in I$.

Замечание 1. Причина применения указанного способа вычисления вектора оценок в висячей вершине v заключается в том, что в этом случае повышается качество распознавания [3, 4].

Пусть висячая вершина v порождена ветвью дерева с обычными вершинами $x_{j_1}, \ldots, x_{j_r}$ и $\sigma_i, i \in 1, \ldots, r,$ — метка дуги, выходящая из вершины x_{j_i}. Под э.к. B_v для висячей вершины v подразумевается конъюнкция вида $[x_{j_1} > d(x_{j_1})]^{\sigma_1} \cdots [x_{j_r} > d(x_{j_r})]^{\sigma_r}$, где $[x_{j_i} > d(x_{j_i})]^{\sigma_i} = 1$, если $x_{j_i} > d(x_{j_i})$ при $\sigma_i = 1$ или $x_{j_i} \leqslant d(x_{j_i})$ при $\sigma_i = 0$, иначе $[x_{j_i} > d(x_{j_i})]^{\sigma_i} = 0,\ i \in 1, \ldots, r$.

Под интервалом истинности N_v э.к. B_v будем понимать множество наборов вида $(\alpha_1, \ldots, \alpha_n)$, где $\alpha_{j_i} = \sigma_i$ при $i = 1, \ldots, r$, и $\alpha_j \in \{0, 1\}, j \notin \{j_1, \ldots, j_r\}$.

Описанием объекта $S = (b_1, \ldots, b_n)$ в вершине v будем называть вектор $S(v) = (\beta_1, \ldots, \beta_n)$, в котором $\beta_{j_i} = 1$, если $b_{j_i} > d(x_{j_i})$, иначе $\beta_{j_i} = 0$ при $i = 1, \ldots, r$, и $\beta_j = 0$ при $j \notin \{j_1, \ldots, j_r\}$.

Определение 7. *Висячая вершина v называется голосующей для S, если $S(v) \in N_v$.*

При синтезе ПРД, в отличие от классического РД, описание распознаваемого объекта S может попасть в разные листья дерева, т. е. $S(v_1) \in N_{v_1}$ и $S(v_2) \in N_{v_2}$ при $v_1 \neq v_2$.

Пусть $Q(S)$ — множество всех голосующих висячих вершин для S. Для каждого $i \in I$ вычисляется оценка принадлежности объекта S классу K_i, имеющая вид:

$$\Gamma(S, K_i) = \sum_{v \in Q(S)} \omega_v^i, \;\; i \in I.$$

Объект S зачисляется в класс K_i, если $\Gamma(S, K_i) = \max\limits_{j \in I} \Gamma(S, K_j)$, $i \in I$, $\Gamma(S, K_i) \neq \neq \Gamma(S, K_j)$ при $i \neq j$, $j \in I$.

Если классов с максимальной оценкой несколько, то среди них выбирается только один, а именно тот, который имеет наибольшее число объектов в обучающей выборке, иначе происходит отказ алгоритма от классификации объекта S.

В случае вещественнозначной информации важной является задача нахождения такого порога $d(x_t)$, который наилучшим образом разделяет объекты из $\hat{T}$ по признаку x_t, принадлежащие разным классам. Опишем способ выбора порога для перекодировки текущих значений признака $x_t \in X(\hat{T})$, применяемый в данной работе.

Пусть $\{c_1, \ldots, c_u\}$, $u \leqslant m$, — множество различных значений по признаку x_t, $c_{i+1} > c_i$, $1 \leqslant i \leqslant u-1$. Пусть объекты $S_{i_1} = (a_{i_1 1}, \ldots, a_{i_1 n})$, $S_{i_2} = (a_{i_2 1}, \ldots, a_{i_2 n})$ из $\hat{T}$ принадлежат разным классам. Если $a_{i_1 t} = c_i$ и $a_{i_2 t} = c_{i+1}$, тогда число $k_{t_i} = (c_i + c_{i+1})/2$, $1 \leqslant i \leqslant u-1$, является порогом признака x_t.

Обозначим через $G_t = \{k_{t_1}, \ldots, k_{t_j}\}$ множество порогов признака x_t. Порог $k \in G_t$ разбивает множество $\hat{T}$ на два подмножества $\left\{T_k^{(1)}, T_k^{(2)}\right\}$, где $T_k^{(1)}$ $\left(T_k^{(2)}\right)$ состоит из объектов множества $\hat{T}$, для которых $a_{rt} \leqslant k$ $(a_{rt} > k)$, $r = 1, \ldots, m$.

Здесь применена идея корректного перекодирования вещественнозначной информации, предложенная Ю. И. Журавлевым и используемая при построении логических процедур распознавания для дискретизации исходной информации и понижения значности целочисленных данных [8]. Данный способ определения порога для признака $x_t \in X(\hat{T})$, позволяет сократить число порогов и делает эту процедуру более корректной по сравнению со способом определения порога для признака в алгоритме C4.5 [9].

Для каждого найденного порога признака $x_t \in X(\hat{T})$ определяется «информативность», и в качестве оптимального порога $d(x_t)$ берется тот порог, для которого эта информативность максимальна.

Опишем критерий выбора оптимального порога для признака $x_t \in X(\hat{T})$.

Обозначим через $f\left(K_i, \hat{T}\right)$, $i \in I$, число объектов из множества $\hat{T}$, относящихся к классу K_i, и через R_t — множество объектов из $\hat{T}$, для которых значение признака x_t не определено. Вероятность $P_t^i(\hat{T})$ того, что случайно выбранный объект из множества $\hat{T}$ будет принадлежать классу K_i, равна $f(K_t, \hat{T} \backslash R_t) / \left|\hat{T} \backslash R_t\right|$.

Величина, вычисляемая по формуле

$$\mathrm{Info}(\hat{T})_t = -\sum_{i=1}^{l} P_t^i(\hat{T}) \log_2 P_t^i(\hat{T}),$$

называется количеством информации (энтропией) по признаку x_t, необходимое для определения класса, которому принадлежит объект из множества $\hat{T}$.

Величина, вычисляемая по формуле

$$\mathrm{Info}(x_t)_k = \frac{\left|T_k^{(1)}\right|}{\left|\hat{T} \setminus R_t\right|} \mathrm{Info}\left(T_k^{(1)}\right)_t + \frac{\left|T_k^{(2)}\right|}{\left|\hat{T} \setminus R_t\right|} \mathrm{Info}\left(T_k^{(2)}\right)_t,$$

называется количеством информации, необходимым для определения класса, которому принадлежит объект из множества $\hat{T}$ после разбиения $\hat{T}$ по порогу k признака x_t.

Информационный выигрыш (information gain) после выбора порога k признака x_t вычисляется по формуле $\mathrm{Gain}(x_t)_k = \mathrm{Info}(\hat{T})_t - \mathrm{Info}(x_t)_k$.

Величина, вычисляемая по формуле

$$\mathrm{SplitInfo}(x_t)_k = -\frac{\left|T_k^{(1)}\right|}{\left|\hat{T} \setminus R_t\right|} \log_2 \frac{\left|T_k^{(1)}\right|}{\left|\hat{T} \setminus R_t\right|} - \frac{\left|T_k^{(2)}\right|}{\left|\hat{T} \setminus R_t\right|} \log_2 \frac{\left|T_k^{(2)}\right|}{\left|\hat{T} \setminus R_t\right|},$$

определяет потенциальную информацию, получаемую при разбиении множества $\hat{T}$ по порогу k признака x_t.

Оптимальным порогом в G_t для признака x_t считается порог k, для которого нормированный информационный выигрыш

$$\mathrm{GainRatio}(x_t)_k = \frac{\mathrm{Gain}(x_t)_k}{\mathrm{SplitInfo}(x_t)_k}$$

принимает свое наибольшее значение.

Таким образом, для каждого найденного текущего порога определяется информативность признака $x_t \in X(\hat{T})$ по описанному выше критерию и в качестве оптимального порога $d(x_t)$ берется тот порог, для которого эта информативность максимальна. Данная процедура повторяется для каждого признака из $X(\hat{T})$. Далее вызывается процедура выбора набора признаков для ветвления $X_\nu \in \{x_{j_1}, \ldots, x_{j_q}\}$, $X_\nu \subseteq X(\hat{T})$ (см. разд. 3) и осуществляется ветвление из полной вершины $\left(\left\{x_{j_1}, \ldots, x_{j_q}\right\}, \left\{d(x_{j_1}), \ldots, d(x_{j_q})\right\}\right)$.

Замечание 2. Описанная выше модификация энтропийного критерия была применена в работах [3, 4]. Отличие от аналогичного критерия, применяемого в алгоритме C4.5, заключается в используемой методике учета пропущенных данных в признаковых описаниях обучающих объектов при ветвлении из обычной вершины дерева. Различие методик учета пропусков описано ниже.

В описанных критериях при вычислении информативности разбиения текущего множества $\hat{T}$ по порогу k признака x_t пропущенные значения признака x_t для объектов из $\hat{T}$ не принимаются во внимание. Если в описании обучающего объекта значение признака x_t пропущено, то при ветвлении из обычной вершины, соответствующей признаку x_t, этот объект удаляется. Применяемая методика обработки пропусков направлена на сохранение исходной информации в полном объеме.

В алгоритме C4.5 применяется другая методика: предполагается, что пропущенные значения признака x_t вероятностно распределены пропорционально частоте появления встречающихся значений. Поэтому в алгоритме C4.5 если в описании обучающего объекта значение признака x_t пропущено, то такой объект не удаляется и при ветвлении из обычной вершины, соответствующей признаку x_t, его описание попадает и в левую, и в правую ветвь с определенными весами, которые учитываются при классификации. Использование методики алгоритма C4.5 вносит шум в обучающие данные. Если бы на месте пропущенного значения признака x_t находилось какое-либо реальное число, полученное в процессе сбора данных, то оно могло бы существенно повлиять на выбор оптимального порога признака x_t, что могло бы изменить структуру и качество решающего дерева.

Описание других методик, используемых при решении задачи классификации с пропусками, представлено в работах [10, 11]. Большинство из них основано на замене

пропущенного значения одним из допустимых. Это значение может быть вычислено различными способами: как среднее по существующим значениям признака; как наиболее вероятное значение для признака; случайно выбрано из существующих значений; получено с помощью методов k-ближайших соседей, регрессионного или кластерного анализа. Также существует методика, основанная на удалении объектов с пропущенными значениями из обучающей выборки до начала построения дерева. Такой подход может применяться в случае, когда число объектов с пропущенными значениями невелико по сравнению с числом всех обучающих объектов. Недостаток данного подхода состоит в том, что теряется полезная информация, содержащаяся в удаленных объектах. Иногда применяется методика, заключающаяся в построении дополнительной ветви, выходящей из обычной вершины, соответствующей признаку x_t, в которую «попадают» все обучающие объекты, в описании которых значение признака x_t не определено [12].

В случае если значение признака x_t для распознаваемого объекта S не определено, то признак x_t исключается из исходного набора признаков. Далее строится ПРД для объекта S, т. е. при ветвлении из обычной вершины строится только та ветвь, по которой будет осуществлен «спуск» описания объекта S. Таким образом, при построении ПРД для классификации объекта S учитываются только те признаки, для которых значения в S определены. Данный способ учета пропусков в распознаваемом объекте был применен в [3, 4].

3 Алгоритмы синтеза полного решающего дерева с помощью гетерогенных вычислений

В данной работе в качестве базового алгоритма синтеза ПРД применяется алгоритм AGI.Bias [3].

Опишем алгоритм AGI.Bias.

Алгоритм AGI.Bias является рекурсивным. Пусть $T(a_{rj})$ — матрица, задаваемая обучающей выборкой T, где $r = 1, \ldots, m,\ j = 1, \ldots, n$, a_{rj} — значение признака x_j для обучающего объекта S_r. Обозначим через $\tilde{T}$ матрицу, рассматриваемую на текущем шаге алгоритма, $\tilde{X} = \{x_j \in X_T\}$ — множество всех признаков на текущем шаге. На первом шаге $\tilde{T} = T(a_{rj}),\ X_T = \{x_1, \ldots, x_n\}$. Шаг рекурсии в алгоритме AGI.Bias представляет собой последовательность действий 1–3, описанных ниже.

1. Просматриваются все столбцы матрицы $\tilde{T}$. Если в столбце нет хотя бы двух различных значений, то этот столбец вычеркивается из $\tilde{T}$ и признак, соответствующий данному столбцу, удаляется из $\tilde{X}$. Если $\tilde{X} = \varnothing$, то переходим к третьему действию, иначе осуществляется переход к следующему действию.
2. Для каждого признака $x_j \in \tilde{X}$ вычисляется значение критерия $\mathrm{GainRatio}(x_j)_k$. Если $G_j = \varnothing$, то $\mathrm{GainRatio}(x_j)_k = 0$. Если значение $\mathrm{GainRatio}(x_j)_k = 0$ для всех признаков из $\tilde{X}$, то переходим к третьему действию. Иначе вызывается процедура формирования набора признаков для ветвления, описанная ниже. Пусть в результате сформирован набор признаков $X_\nu = \{x_{j_1}, \ldots, x_{j_q}\}, 1 \leqslant q \leqslant n$. Создается полная вершина с меткой $\left(\{x_{j_1}, \ldots, x_{j_q}\}, \{d(x_{j_1}), \ldots, d(x_{j_q})\}\right)$. Далее осуществляется ветвление по каждому признаку $x_t \in X_\nu, t \in \{j_1, \ldots, j_q\}$.

   Для каждого признака $x_t \in X_\nu, t \in \{j_1, \ldots, j_q\}$, по оптимальному порогу $k = d(x_t)$ строятся две дуги, выходящие из вершины (x_t, k). Если матрица $\tilde{T}$ состоит из одного столбца, то при построении подматриц $\tilde{T}_k^{(1)}$ и $\tilde{T}_k^{(2)}$ этот столбец не удаляется. Для левой (правой) дуги вершины (x_t, k) строится подматрица $\tilde{T}_k^{(1)}$ ($\tilde{T}_k^{(2)}$) матрицы $\tilde{T}$, полученная удалением столбца, соответствующего признаку x_t, и строк S_r, в которых

$a_{rt} > k$ ($a_{rt} \leqslant k$), $r = 1, \ldots, m$. Если подматрица $\tilde{T}_k^{(1)}$ ($\tilde{T}_k^{(2)}$) содержит объекты одного класса или состоит из одного столбца, то переходим к третьему шагу, иначе $\tilde{T} = \tilde{T}_k^{(1)}$ ($\tilde{T} = \tilde{T}_k^{(2)}$), $\tilde{X} = \tilde{X} \backslash \{x_t\}$ и осуществляется рекурсивный переход к первому действию.

3. Пусть $\tilde{T}$ содержит m_v^i объектов класса $K_i, i \in I$. Строится висячая вершина v с меткой $\left(B_v, \{\omega_v^1, \ldots, \omega_v^l\}\right)$, B_v — конъюнкция соответствующая данной вершине, $\omega_v^i = (m_v^i + 1)/(m^i + l)$, где m^i — число объектов класса K_i в исходной обучающей выборке, $i \in I$.

Для того чтобы построить ПРД для классификации объекта S в случае наличия пропусков в описании объекта $S = (b_1, \ldots, b_n)$, достаточно положить на первом рекурсивном шаге $\tilde{X} = \{x_j \in X_T | b_j \neq «*»\}$.

Замечание 3. Для сокращения времени классификации объекта S строятся только голосующие за S листья ПРД [3]. Поэтому при спуске из обычной вершины $(x_t, d(x_t))$ если $b_t \leqslant d(x_t)$, то строится левая дуга, иначе строится правая дуга.

Процедура выбора набора признаков X_ν для ветвления представляет собой следующую последовательность шагов.

1. Пусть $\tilde{T}$ содержит w столбцов, соответствующих признакам $x_{j_1}, \ldots, x_{j_w}$. Тогда $Y = \{x_{j_1}, \ldots, x_{j_w}\}$.
2. Вычисляется средний информационный выигрыш $q = \sum\limits_{i=1,\ldots,w} \text{GainRatio}(x_{j_i})_k / w$.
3. Определяется число признаков, для которых информационный выигрыш выше среднего или равен ему $n = \sum\limits_{i=1,\ldots,w} c_{j_i}$, где $c_{j_i} = 1$, если $\text{GainRatio}(x_{j_i})_k \geqslant q$, иначе $c_{j_i} = 0$. Признаки x_{j_i}, для которых $\text{GainRatio}(x_{j_i})_k < q$, $i = 1, \ldots, w$, удаляются из Y.
4. Вычисляется $h = \min\limits_{x_i \in Y} \text{GainRatio}(x_i)_k$.
5. Если $(q/n) + h \geqslant \max\limits_{x_i \in Y} \text{GainRatio}(x_i)_k$, то происходит выход из процедуры и возвращается итоговый набор признаков Y, иначе осуществляется переход к третьему шагу процедуры, положив $q := (q/n) + h$.

Смысл указанной процедуры формирования набора признаков X_ν для ветвления на текущем шаге синтеза дерева заключается в поиске признаков, информативность которых совпадает с максимальным значением информативности среди признаков из Y, а также признаки, информативность которых близка к максимальному значению информативности. Близость признаков из Y по информативности вычисляется на основе среднего нормированного информационного выигрыша.

Таким образом, в процессе синтеза ПРД наибольшее время (в основном от 93%–99% от всего времени синтеза ПРД) (см. разд. 4) тратится на поиск оптимальных порогов для признаков на каждом шаге синтеза дерева, так как для каждого признака из $\tilde{X}$ требуется выделить все различные значения признака, далее отсортировать их, после чего найти возможные пороги для бинарной перекодировки, по описанному в разд. 2 модифицированному энтропийному критерию вычислить информативность для каждого найденного порога и выбрать порог с максимальной информативностью. Поэтому в приведенных ниже алгоритмах синтеза ПРД с использованием технологии CUDA указанные вычисления были полностью «перенесены» на GPU.

В первом из разработанных алгоритмов — PAGI.Bias (Parallel AGI.Bias) — реализована функция поиска оптимального порога для признака $x_t \in \tilde{X}$ на GPU — FindOptimalThreshold. На вход данной функции передаются: массив значений признака x_t, текущий массив меток обучающих объектов и начальная информативность при-

знака x_t ($\mathrm{Info}(\hat{T})_t$ (см. разд. 2)). На выходе функции FindOptimalThreshold: оптимальный порог и значение максимальной информативности для признака x_t. При инициализации функции FindOptimalThreshold указывается величина p — максимальное число вычислительных «нитей», которые могут быть использованы GPU для вычислений. В описанной функции последовательно выполняются следующие шаги:

1. Определяются все «уникальные» (различные) значения признака x_t. Данный блок выполняется параллельно p вычислительными нитями GPU. Пусть a_i^j — i-е значение признака x_t, которое проверяется на уникальность j-й нитью. Просматриваются все значения признака x_t, индекс которых меньше i, и если нет ни одного значения, равного a_i^j, то a_i^j считается уникальным для признака x_t. После проверки на уникальность одного значения вычислительная нить переходит к проверке следующего значения. При этом каждая нить анализирует значения признака x_t на уникальность, индексы которых во входном массиве не совпадают со значениями индексов, анализируемыми другими нитями GPU.
2. Сортируется массив уникальных значений признака x_t, полученный на шаге 1. В качестве метода сортировки используется вариант сортировки подсчетом, суть которой заключается в определении числа значений, которые меньше текущего значения. Данный блок выполняется параллельно p вычислительными нитями. Пусть c_i^j — i-е уникальное значение признака x_t, которое сортируется j-й нитью. Определяется s_i^j — число уникальных значений признака x_t, меньшее c_i^j. Число s_i^j однозначно определяет индекс значения c_i^j в результирующем массиве. После определения индекса для одного уникального значения признака x_t вычислительная нить переходит к поиску индекса для следующего уникального значения. Здесь так же, как и на первом шаге, каждая нить определяет индексы для набора уникальных значений признака x_t, который не пересекается ни с одним набором, анализируемым другими нитями.
3. Выделяются возможные пороги на основе отсортированного массива уникальных значений признака, полученного на шаге 2. Для проверки того, что полусумма двух соседних значений из упорядоченного множества уникальных значений является порогом, необходимо просмотреть метки обучающих объектов, в описании которых встречается одно из двух значений, и если найдутся хотя бы два таких объекта, принадлежащих разным классам, то данное число является порогом (см. разд. 2). Данный блок также выполняется p вычислительными нитями GPU. Это значит, что каждая нить осуществляет описанную проверку только по одной полусумме соседних значений из входного массива для данного шага. Аналогично предыдущим шагам после проверки одной полусуммы нить переходит к проверке следующей полусуммы соседних значений. Наборы анализируемых значений полусумм для вычислительных нитей GPU не пересекаются между собой.
4. Вычисляются значения информативности для каждого порога, полученного на шаге 3. Данный блок также выполняется параллельно p вычислительными нитями. Пусть k_i^j — i-й порог признака x_t, для которого вычисляется информативность j-й нитью. Для порога k_i^j нитью с индексом j определяется значение модифицированного энтропийного критерия $\mathrm{GainRatio}(x_t)_{k_i^j}$ (см. разд. 2). После расчета информативности порога вычислительная нить переходит к вычислению информативности следующего порога из входного массива для данного шага. Наборы порогов, для которых вычисляется информативность нитями GPU, не пересекаются между собой.
5. Для признака x_t вычисляется оптимальный порог $d(x_t)$ — порог с максимальной информативностью. Поиск оптимального порога осуществляется редукцией [5, 6] масси-

ва информативности порогов, полученного на шаге 4, с помощью p вычислительных нитей.

Таким образом, из описания алгоритма PAGI.Bias следует, что реализованный способ вычислений на GPU «позволяет» с линейной сложностью отсортировать массив текущих значений признака, построить все пороги и вычислить их информативность (для входного массива значений признака из m элементов, если число параллельно работающих нитей больше m), а затем с логарифмической сложностью осуществить поиск оптимального порога. Для лучшей загрузки ядер GPU на каждом шаге синтеза ПРД динамически определяется максимальное число вычислительных нитей GPU, которые могут быть задействованы при поиске оптимального порога для признака.

Замечание 4. В описанном выше алгоритме PAGI.Bias говорится, что блок вычислений на каждом шаге функции FindOptimalThreshold будет выполняться параллельно p вычислительными нитями. Здесь следует понимать, что выполнение не подразумевает, что данный блок будет действительно выполнен параллельно p нитями, так как следует учитывать программно-аппаратные ограничения максимального числа параллельно работающих нитей и особенности работы вычислительных нитей с учетом задержек, связанных с операциями доступа к глобальной памяти GPU [5, 6].

Второй разработанный алгоритм — Dynamic PAGI.Bias — отличается от алгоритма PAGI.Bias тем, что поиск оптимального порога осуществляется параллельно для всех признаков на текущем шаге синтеза ПРД. Для этого применяется динамический параллелизм (возможность динамически порождать новые вычислительные потоки без возврата к коду, исполняемом на CPU) [6]. Если на текущем шаге синтеза ПРД имеется m признаков, то с помощью динамического параллелизма создается m вычислительных потоков, где поток с индексом i осуществляет поиск оптимального порога для признака x_i, т. е. внутри потока инициализируется и вызывается функция FindOptimalThreshold для соответствующего признака. Описанное отличие позволяет еще больше распараллелить вычисления и уменьшить временны́е издержки, связанные с необходимостью копировать данные между памятью GPU и оперативной памятью CPU. В алгоритме PAGI.Bias на одном шаге синтеза дерева при расчете оптимальных порогов для m признаков и n обучающих объектов требуется передать $4(mn + n + 2)$ байт и получить $8m$ байт данных. В алгоритме Dynamic PAGI.Bias требуется передать $4(2n + m + 2)$ байт и получить $8m$ байт данных.

Третий разработанный алгоритм — Deep Dynamic PAGI.Bias — отличается от алгоритма Dynamic PAGI.Bias тем, что динамический параллелизм применяется и на более низких вычислительных уровнях — при реализации блоков вычислений 1–4 функции FindOptimalThreshold. Например, в блоке поиска уникальных значений признака x_t формируются вычислительные потоки, каждый из которых с помощью z вычислительных нитей параллельно проверяет на уникальность не более z значений признака, после анализа выбранного набора значений вычислительный поток прекращает свою работу. Отличие заключается в том, что описанные потоки независимы друг от друга (за исключением того, что один поток может начать свое выполнение раньше (если потоки попали в одну очередь потоков GPU) или чуть раньше другого потока (если потоки попали в разные очереди потоков GPU) [5, 6]), а в алгоритме Dynamic PAGI.Bias данный блок выполняется в одном вычислительном потоке, тем самым накладывая определенные ограничения на параллелизм вычислений нитями этого потока. Для лучшей загрузки мультипроцессоров GPU на каждом шаге синтеза ПРД в алгоритме Deep Dynamic PAGI.Bias динамически определяется z — число вычислительных нитей GPU, которые могут быть

задействованы при работе каждого вычислительного потока, выполняемых в блоках 1–4 функции FindOptimalThreshold.

4 Результаты численного эксперимента

Исследование времени поиска оптимальных порогов для признаков в процессе построения ПРД алгоритмами AGI.Bias, PAGI.Bias, Dynamic PAGI.Bias и Deep Dynamic PAGI.Bias, описанных в разд. 3, осуществлялось на модельных данных. Каждая модель представляет собой один шаг синтеза ПРД указанными алгоритмами, на котором рассматриваются разные наборы значений по одному или нескольким признакам. Отличие моделей друг от друга заключается в распределении значений признака x и в числе обучающих объектов. Число классов равно двум, число обучающих объектов каждого класса одинаково, т. е. $|K_1| = |K_2|$. Для каждой модели рассматривались варианты с $1, 2, 4, 6, 8, 12, 16$ и 20 признаками. При этом значения всех признаков модели совпадают, т. е. если в описании обучающего объекта значение признака j равно 1, то и значение признака k равно $1, \forall k \neq j$. Таким образом, исследуется влияние размерности и типа задачи на время расчета оптимальных порогов разработанными алгоритмами.

Опишем модели и полученные результаты.

Модель 1 — имеется 2500 объектов первого класса и 2500 объектов второго класса. Значения признака x представляют собой массив последовательных значений от 0 до $m - 1$ включительно, где m — общее число обучающих объектов. Метки обучающих объектов чередуются, т. е. первый объект имеет метку класса 1, второй объект — метку класса 2, третий объект — метку класса 1 и т. п.

Модель 2 и модель 3 аналогичны модели 1 за исключением того, что в этих случаях рассматривается ситуация, когда в первом классе и во втором классе по 250 объектов и 25 объектов соответственно.

Модели 1–3 позволяют исследовать время расчета оптимальных порогов для признаков алгоритмами, приведенных в разд. 3, в худшей ситуации (целочисленная информация большой значности). В этом случае для каждого признака с описанным распределением m значений по m обучающим объектам будет найдено m различных значений, отсортировано ровно m значений, будет определено $m - 1$ порогов, для каждого из найденного порога будет вычислена информативность по модифицированному энтропийному критерию (см. разд. 2), и будет найден оптимальный порог редукцией $m - 1$ значения информативности найденных порогов.

Модель 4 по сути является моделью 1, но только в этой модели значения признака для i-ого объекта равно $(i - 1) \mod 15$, т. е. первые 15 значений последовательно заполнены числами от 0 до 14 включительно, следующие 15 значений тоже последовательно заполнены числами от 0 до 14 включительно и т. п.

Модель 5 и модель 6 аналогичны модели 4, но только в этих случаях рассматривается ситуация, когда в первом классе и во втором классе по 250 объектов и 25 объектов соответственно.

Модели 4–6 позволяют проанализировать время поиска оптимальных порогов для признаков алгоритмами AGI.Bias, PAGI.Bias, Dynamic PAGI.Bias и Deep Dynamic PAGI.Bias в лучшей ситуации (целочисленная информация малой значности). В этом случае для каждого признака с таким распределением 15-ти значений по m обучающим объектам будет найдено 15 различных значений, отсортировано ровно 15 значений, будет определено 14 порогов, для каждого из этих порогов будет вычислена информативность по модифицированному энтропийному критерию и будет найден оптимальный порог редукцией 14 значений информативности найденных порогов.

Значения времени поиска оптимальных порогов для всех рассматриваемых наборов признаков каждой модели и каждого алгоритма приведены на рис. 25. По оси абсцисс — 75-й процентиль времени поиска оптимальных порогов для данного набора признаков, по оси ординат — число признаков. Время указано в миллисекундах. Для расчета 75-го процентиля вычисление оптимальных порогов для каждого набора признаков каждой модели и каждого алгоритма осуществлялось 40 раз. На графиках введены обозначения: CPU — AGI.Bias, GPU 1 — PAGI.Bias, GPU 2 — Dynamic PAGI.Bias, GPU 3 — Deep Dynamic PAGI.Bias.

Из приведенных графиков на рис. 1, *г*–1, *е* видно, что в моделях с небольшим числом различных значений признака (модели 4–6) параллельные варианты алгоритма AGI.Bias показывают плохие результаты по сравнению с алгоритмом AGI.Bias, за исключением ситуации, когда имеется большое число признаков и большое число обучающих объектов — алгоритм Deep Dynamic PAGI.Bias показал сопоставимые с алгоритмом AGI.Bias результаты. Также следует отметить, что при увеличении размерности по числу обучающих объектов (при переходе от модели 5 к модели 4) скорость роста времени расчета оптимальных порогов алгоритмом AGI.Bias в несколько раз больше по сравнению с разработанными параллельными вариантами этого алгоритма (в среднем в 2,5 раза больше).

Сравнение параллельных алгоритмов между собой на графиках рис. 1, *г*–1, *е* показывает следующее.

1. Алгоритм PAGI.Bias медленнее (в среднем в 7,5 раза) алгоритмов Dynamic PAGI.Bias и Deep Dynamic PAGI.Bias за счет того, что во время расчета оптимальных порогов требуется передать больше данных в память GPU, и за счет того, что поиск оптимальных порогов осуществляется последовательно по одному признаку.
2. Алгоритм Dynamic PAGI.Bias быстрее (в среднем в 1,9 раза) алгоритма Deep Dynamic PAGI.Bias на моделях с небольшим и малым числом обучающих объектов (модели 5 и 6) за счет того, что алгоритму Deep Dynamic PAGI.Bias требуется больше данных (различных значений признака и порогов признака) для лучшей загрузки вычислительных потоков на более низких вычислительных уровнях (см. разд. 3), иначе время, затрачиваемое на инфраструктурные расходы, оказывается больше выигрыша по времени, получаемого от распараллеливания вычислений на этих уровнях.
3. Алгоритм Deep Dynamic PAGI.Bias быстрее (в среднем в 1,5 раза) алгоритма Dynamic PAGI.Bias на модели с большим числом обучающих объектов (модель 4).

Из графиков на рис. 1, *а*–1, *в* следует, что в моделях с большим числом обучающих объектов и с большим числом различных значений признака (модель 1) параллельные варианты алгоритма AGI.Bias показывают существенное ускорение расчета оптимальных порогов для признаков (до 20 раз) по сравнению с алгоритмом AGI.Bias. Если имеется меньшее число обучающих объектов с большим числом различных значений признака (модель 2), то параллельные алгоритмы показывают ускорение до 5 раз. Если осуществляется расчет оптимальных порогов для признаков с малым числом обучающих объектов и с большим числом различных значений признака (модель 3), то параллельные алгоритмы оказываются медленнее алгоритма AGI.Bias. Следует также отметить, что при увеличении размерности по числу обучающих объектов (при переходе от модели 2 к модели 1) скорость роста времени расчета оптимальных порогов алгоритмом AGI.Bias в несколько раз больше по сравнению с разработанными параллельными вариантами этого алгоритма (в среднем в 4 раза больше).

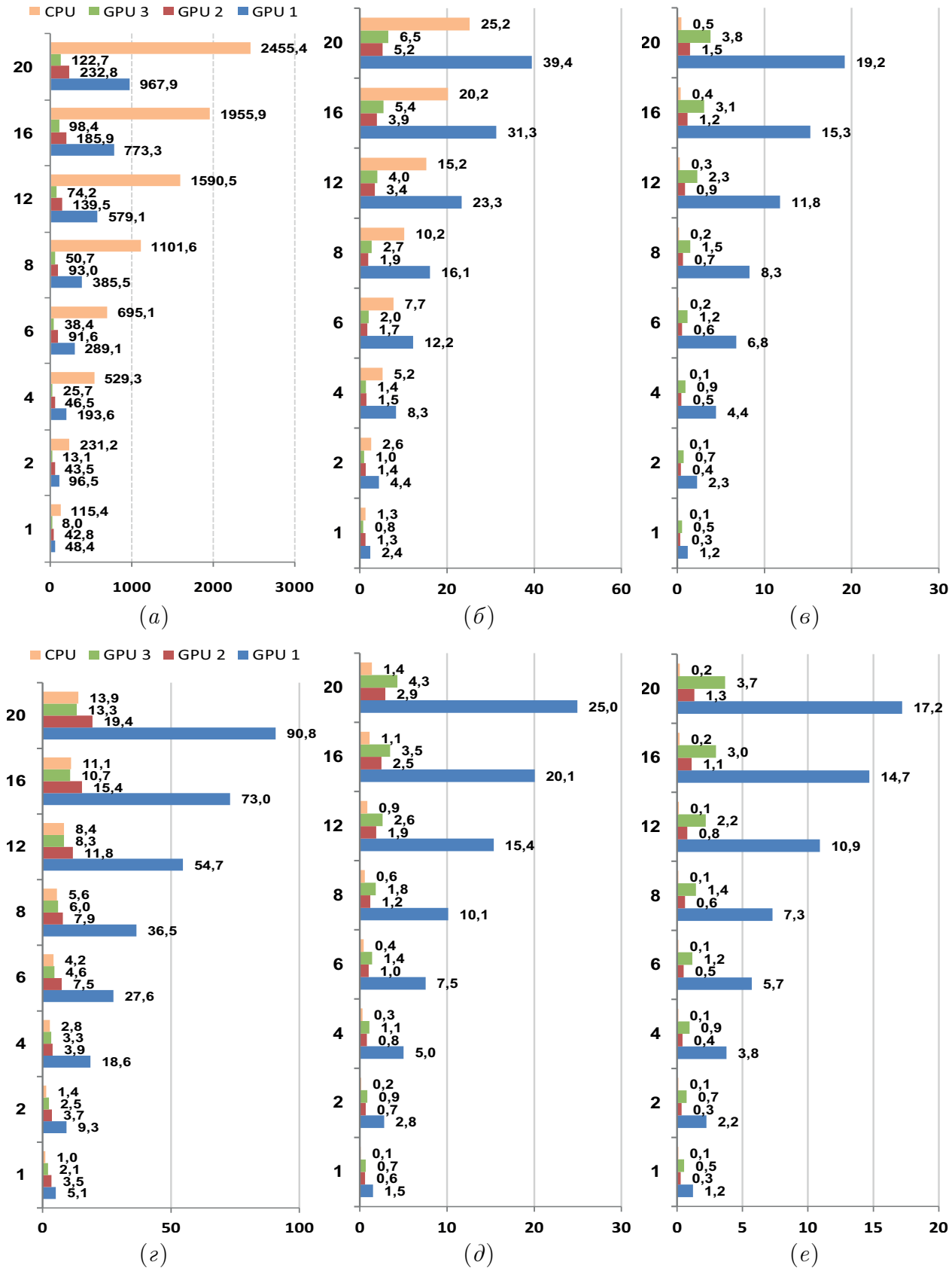


Рис. 1 Изменение времени вычисления информативности признаков от числа признаков для различных алгоритмов в моделях 1 (*а*), 2 (*б*), 3 (*в*), 4 (*г*), 5 (*д*) и 6 (*е*)

Сравнение параллельных алгоритмов между собой на графиках рис. 1, *а*–1, *в* показывает следующее.

1. Алгоритм PAGI.Bias медленнее (в среднем в 8 раз) алгоритмов Dynamic PAGI.Bias и Deep Dynamic PAGI.Bias за счет того, что во время расчета оптимальных порогов требуется передать больше данных из памяти CPU в память GPU, и за счет того, что поиск оптимальных порогов осуществляется последовательно по одному признаку.
2. Алгоритм Dynamic PAGI.Bias быстрее алгоритма Deep Dynamic PAGI.Bias (в среднем в 1,6 раза) на моделях с небольшим и малым числом обучающих объектов (модели 2 и 3) за счет того, что алгоритму Deep Dynamic PAGI.Bias требуется больше данных (различных значений признака и порогов для признака), чтобы время, затрачиваемое на инфраструктурные расходы дополнительного распараллеливания вычислений на более низких вычислительных уровнях (см. разд. 3), было меньше выигрыша по времени, получаемого от этого дополнительного распараллеливания.
3. Алгоритм Deep Dynamic PAGI.Bias быстрее алгоритма Dynamic PAGI.Bias (в среднем в 2,6 раз) на модели с большим числом обучающих объектов (модель 1).

Таким образом, на модельных данных наилучшие результаты в случае малого числа обучающих объектов или признаков с малой значностью показывает алгоритм AGI.Bias, в случае большого числа обучающих объектов с большой значностью признаков наилучшие результаты показывает алгоритм Deep Dynamic PAGI.Bias (до 20 раз быстрее алгоритма AGI.Bias). В ситуации небольшого числа обучающих объектов с большой значностью признаков наиболее быстрым оказался алгоритм Dynamic PAGI.Bias (до 5 раз быстрее алгоритма AGI.Bias и до 1,4 раза быстрее алгоритма Deep Dynamic PAGI.Bias).

Исследование времени синтеза ПРД алгоритмами AGI.Bias, PAGI.Bias, Dynamic PAGI.Bias и Deep Dynamic PAGI.Bias, описанных в разд. 3, осуществлялось на 24 реальных задачах: 5 задач из репозитория ВЦ РАН [1] и 19 задач (Abalone (задача № 12), EEG Eye State (задача № 13), Adult (задача № 14), Bank Marketing (задача № 15), Image Segmentation (задача № 16), MAGIC Gamma Telescope (задача № 17), Statlog Image Segmentation(задача № 19), Seismic-Bumps (задача № 20), Statlog Shuttle (задача № 21), Wine Quality Red and White (задачи № 22 и № 23), LED Display (задача № 24), Wilt (задача № 11), Pima Indians Diabetes (задача № 8), Credit Approval (задача № 9), Banknote Authentication (задача № 7), Hepatitis (задача № 6), Glass Identification (задача № 5), Wine (задача № 3)) из репозитория UCI [7].

На задачах №№ 12–24 каждым алгоритмом полностью строилось по одному ПРД (без применения процедуры, описанной в замечании 3). На задачах №№ 1–11 рассматривалась процедура скользящего контроля («leave-one-out» (LOO)) [13]. Для этой процедуры один шаг алгоритма заключается в удалении одного из объектов обучающей выборки, построении классификатора (синтез ПРД с применением процедуры, описанной в замечании 3) и распознавании удаленного объекта.

Опишем полученные результаты.

В табл. 1 представлены результаты алгоритмов для задач №№ 1–11. Для алгоритма AGI.Bias и каждой задачи приведены две величины: общее время работы процедуры LOO и количество процентов от общего времени, затраченное на вычисление оптимальных порогов. Для параллельных версий алгоритма AGI.Bias в табл. 1 приведены значения величины

$$q = \begin{cases} -\dfrac{t_G}{t_C}, & \text{если } t_G > t_C; \\ \dfrac{t_C}{t_G}, & \text{если } t_C \geqslant t_G, \end{cases}$$

Таблица 1 Эффективность алгоритмов по времени синтеза ПРД в процедуре LOO

Описание задачи		AGI.BIAS	PAGI.BIAS	Dynamic PAGI.BIAS	Deep Dynamic PAGI.Bias
№	$(\|K_1\|,\ldots,\|K_l\|,n)$				
1	$(48,23,8)*$	245,0 (92,91%)	−45,6	−20,9	−15,0
2	$(47,30,7)$	54,2 (93,57%)	−37,2	−11,1	−8,2
3	$(59,71,48,13)*$	2374,2 (97,94%)	−24,9	−4,5	−4,1
4	$(51,218,24)$	3755,4 (86,99%)	−106,1	−26,4	−31,2
5	$(70,76,17,13,9,29,9)*$	46915,6 (98,69%)	−9,3	−4,6	−2,2
6	$(32,123,19)$	397,4 (93%)	−69,1	−12,6	−14,1
7	$(762,610,4)*$	85248,1 (92,91%)	−2,2	1,3	15,2
8	$(500,268,8)*$	58757,9 (99,88%)	−5,3	−2,1	1,8
9	$(307,383,15)$	12414,4 (96,3%)	−24,9	−8,2	−6,3
10	$(11,47,15)*$	91,5 (95,79%)	−29,6	−6,2	−14,6
11	$(74,4265,5)*$	3649297,0 (99,95%)	1,8	5,0	14,3

Таблица 2 Эффективность алгоритмов по времени синтеза одного ПРД

Описание задачи		AGI.BIAS	PAGI.BIAS	Dynamic PAGI.BIAS	Deep Dynamic PAGI.Bias
№	$(\|K_1\|,\ldots,\|K_l\|,n)$				
12	$(1528,1307,1342,8)*$	36488,6 (99,69%)	1,2	2,0	5,4
13	$(8257,6723,14)*$	3375061,3 (99,39%)	−1,2	1,6	3,2
14	$(24720,7841,14)$	486958,7 (99,17%)	−1,8	−1,5	7,6
15	$(39922,5289,16)$	59926,0 (96,67%)	−4,3	−3,0	1,3
16	$(30,30,30,30,30,30,30,14)*$	77797,3 (87,81%)	−77,9	−35,1	−29,5
17	$(12332,6688,10)*$	17322443,8 (99,97%)	3,1	5,3	26,3
18	$(96,104,963)*$	18398853,7 (99,97%)	3,3	5,6	27,5
19	$(330,330,330,330,330,330,330,19)*$	862045,5 (97,54%)	−12,1	−6,1	−3,5
20	$(2414,170,18)$	2185,8 (92,83%)	−14,9	−8,1	−4,2
21	$(34108,37,132,6748,2458,6,11,9)$	3609,9 (95,88%)	−10,6	−4,0	1,6
22	$(20,163,1457,2198,880,175,5,11)*$	5170,2 (99,69%)	−1,9	1,1	2,5
23	$(10,53,681,638,199,18,11)*$	2650,1 (99,23%)	−2,4	−1,1	1,5
24	$(300,330,309,315,310,269,302,304,276,285,7)$	70,3 (98,95%)	−17,7	−15,7	−12,3

где t_C — общее время процедуры LOO при использовании алгоритма AGI.Bias; t_G — общее время процедуры LOO для соответствующего варианта параллельной версии алгоритма AGI.Bias. В табл. 2 представлены результаты алгоритмов для задач №№ 12–24. Смысл значений, приведенных в табл. 2, аналогичен смыслу значений табл. 1 за исключением того, что вместо процедуры LOO для задачи полностью строилось одно ПРД. Звездочкой помечены задачи, в которых большинство признаков с вещественнозначной или целочисленной информацией большой значности. Коричневым цветом выделены значения, когда достигается ускорение алгоритма PAGI.Bias относительно алгоритма AGI.Bias, бирюзовым — значения ускорения алгоритма Dynamic PAGI.Bias относительно AGI.Bias, зеленым — значения ускорения алгоритма Deep Dynamic PAGI.Bias относительно AGI.Bias.

На рис. 2 и 3 для некоторых задач приведены нормированные графики изменения времени расчета оптимальных порогов на каждом ярусе дерева при построении ПРД исследуемыми алгоритмами.

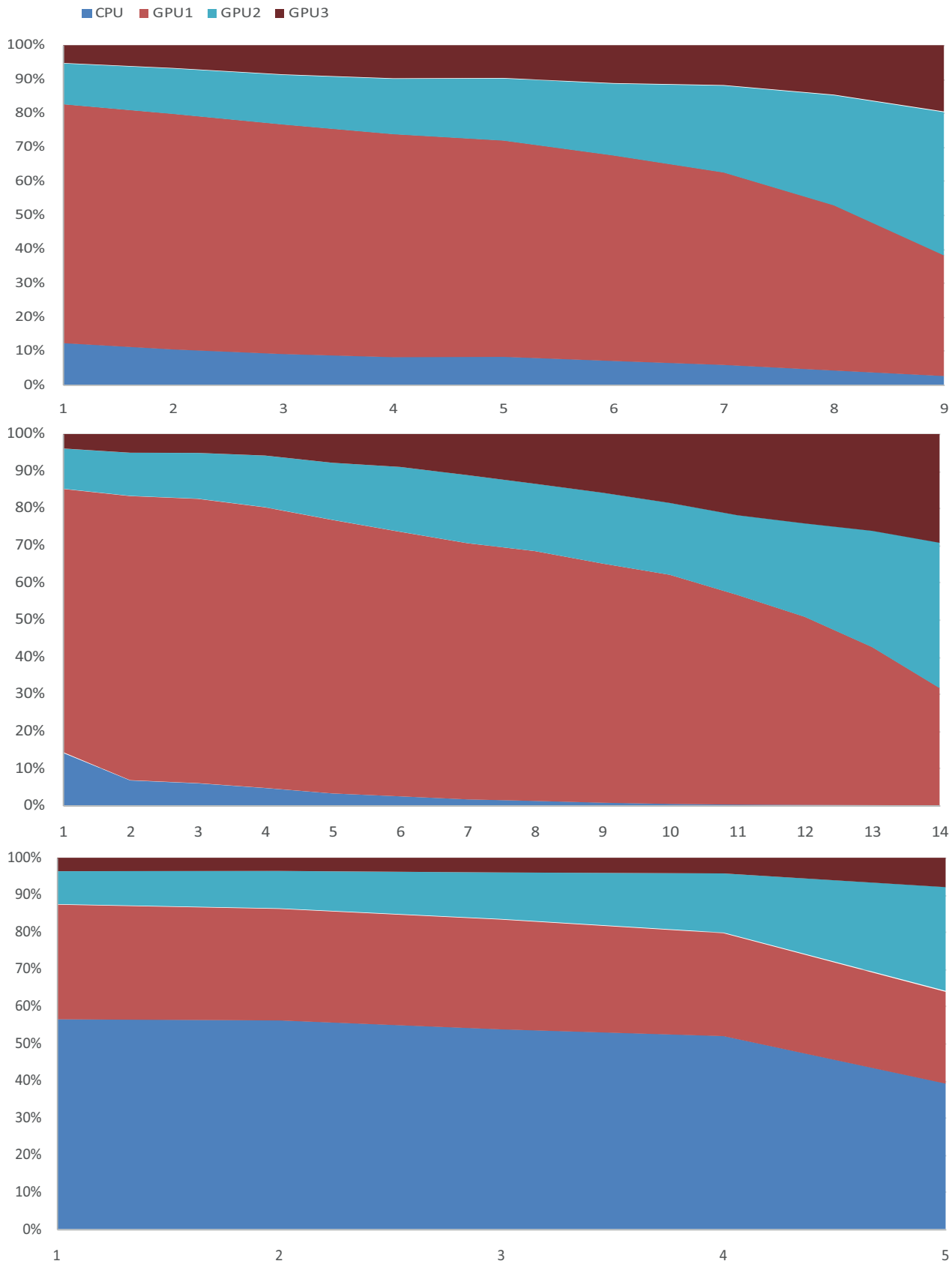


Рис. 2 Нормированные графики изменения времени расчета оптимальных порогов в процедуре LOO для каждого яруса и каждого алгоритма в задачах № 5 (сверху), № 8 (по центру) и № 11 (снизу)

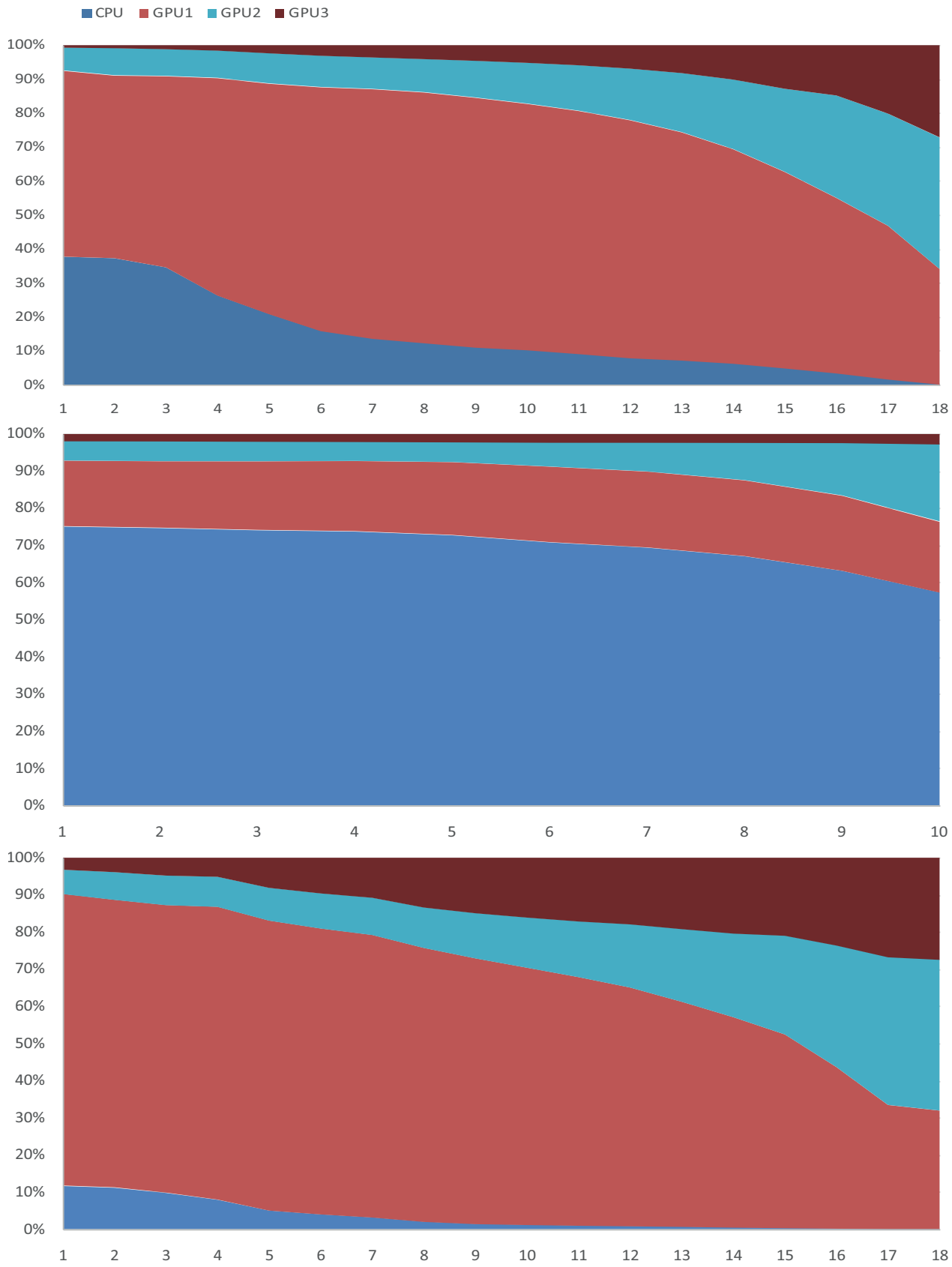


Рис. 3 Нормированные графики изменения времени расчета оптимальных порогов в процессе синтеза ПРД для каждого яруса и каждого алгоритма в задачах №19 (сверху), №18 (по центру) и №16 (снизу)

Наиболее трудоемким этапом синтеза ПРД алгоритмом AGI.Bias является поиск оптимальных порогов для бинарной перекодировки значений признаков. На указанные вычисления приходится в среднем 96,3% от всего времени синтеза ПРД.

Анализ полученных результатов по табл. 1 и 2 показал, что наиболее быстрым из описанных в разд. 3 параллельных алгоритмов оказался алгоритм Deep Dynamic PAGI.Bias, который позволяет сократить время построения ПРД (до 27 раз) по сравнению с алгоритмом AGI.Bias только в тех случаях, когда обучающая выборка содержит большое число признаков и/или обучающих объектов и при этом информация либо вещественнозначная, либо целочисленная большой значности (задачи №№ 7, 8, 11–15, 17, 18, 21–23). В случае небольших обучающих выборок с вещественнозначной или целочисленной информацией небольшой значности уменьшение времени (в среднем в 8,5 раз) наблюдалось только при построении первых трех–пяти ярусов ПРД (задачи №№ 5, 9, 16, 19 и 20). При построении остальных ярусов ПРД происходило увеличение времени за счет того, что копирование данных между GPU и CPU занимало большее время, чем получаемый выигрыш в вычислениях на GPU по сравнению с CPU. Данный факт наглядно показан на рис. 2 и 3. На задачах с малой значностью признаков или малой обучающей выборкой наблюдалось существенное увеличение времени при построении ПРД параллельными версиями алгоритма AGI.Bias. Данный факт объясняется, во-первых, тем, что выигрыша в вычислениях на GPU по сравнению с CPU практически нет. Во-вторых, тем, что на перенос необходимой информации в память GPU требуется намного больше времени, чем получаемый выигрыш в вычислениях на GPU.

Сравнение параллельных версий алгоритма AGI.Bias между собой показало следующее:

1. Применение в алгоритме Dynamic PAGI.Bias динамического параллелизма для поиска оптимальных порогов по нескольким признакам параллельно и передача меньшего объема данных между CPU и GPU позволило уменьшить время построения ПРД до 5,5 раз (в среднем в 2,7 раза) по сравнению с PAGI.Bias.
2. Применение динамического параллелизма на более «низких» вычислительных уровнях (см. разд. 3) в алгоритме Deep Dynamic PAGI.Bias позволило ускорить синтез ПРД до 11 раз (в среднем в 3 раза) по сравнению с алгоритмом Dynamic PAGI.Bias.

5 Заключение

В данной работе разработаны алгоритмы синтеза ПРД с применением параллельных вычислений на основе технологии CUDA — PAGI.Bias, Dynamic PAGI.Bias и Deep Dynamic PAGI.Bias. В качестве базового алгоритма был использован алгоритм AGI.Bias.

Алгоритм PAGI.Bias отличается от алгоритма AGI.Bias тем, что поиск оптимального порога для бинарной перекодировки текущих значений признака при синтезе ПРД полностью перенесен на GPU. Указанное изменение ускорило построение ПРД алгоритмом PAGI.Bias (до 3,3 раз на реальных задачах) по сравнению с алгоритмом AGI.Bias.

Алгоритм Dynamic PAGI.Bias отличается от алгоритма PAGI.Bias в применении динамического параллелизма (поиск оптимального порога осуществляется по нескольким признакам параллельно) и в уменьшении объема передаваемых данных между памятью GPU и оперативной памятью CPU. Данные изменения позволили снизить время синтеза дерева алгоритмом Dynamic PAGI.Bias (до 5,5 раз на реальных задачах) по сравнению с алгоритмом PAGI.Bias.

Алгоритм Deep Dynamic PAGI.Bias отличается от алгоритма Dynamic PAGI.Bias тем, что динамический параллелизм применяется и на более «низких» вычислительных уров-

нях (например, при расчете информативности порогов признака по различным наборам порогов параллельно). Показано, что применение алгоритма Deep Dynamic PAGI.Bias позволяет заметно снизить время синтеза ПРД (до 11 раз на реальных задачах) по сравнению с алгоритмом Dynamic PAGI.Bias.

На реальных задачах показано, что наиболее быстрым из разработанных алгоритмов оказался алгоритм Deep Dynamic PAGI.Bias, который позволяет сократить время построения ПРД (до 27 раз) по сравнению с алгоритмом AGI.Bias только в тех случаях, когда обучающая выборка содержит большое число признаков и/или обучающих объектов и при этом информация либо вещественнозначная, либо целочисленная большой значности. В случае небольших обучающих выборок с вещественнозначной или целочисленной информацией небольшой значности уменьшение времени (в среднем в 8,5 раз) наблюдалось только при построении первых трех–пяти ярусов ПРД. При построении остальных ярусов ПРД происходило увеличение времени за счет того, что копирование данных между GPU и CPU занимало большее время, чем получаемый выигрыш в вычислениях на GPU по сравнению с CPU. На задачах с малой значностью признаков или малой обучающей выборкой наблюдалось увеличение времени на всех ярусах ПРД по той же самой причине.

Таким образом, разработанные алгоритмы синтеза ПРД с применением параллельных вычислений на основе технологии CUDA позволяют существенно снизить время синтеза дерева (более чем в 10 раз) только для задач распознавания по прецедентам, содержащим большое число признаков и/или обучающих объектов с вещественнозначной информацией или целочисленной информацией большой значности.

Литература

[1] *Журавлев Ю. И., Рязанов В. В., Сенько О. В.* Распознавание. Математические методы. Программная система. Практические применения. — М.: ФАЗИС, 2006. 176 с.

[2] *Djukova E. V., Peskov N. V.* A classification algorithm based on the complete decision tree // J. Pattern Recogn. Image Anal., 2007. Vol. 17. No. 3. P. 363–367.

[3] *Генрихов И. Е.* Построение и исследование полных решающих деревьев для задач классификации по прецедентам: Дисс. ... канд. физ.–мат. наук. — М., 2013. 169 с.

[4] *Генрихов И. Е.* Исследование обобщающей способности полного решающего дерева // Ж. вычисл. мат. мат. физ., 2014. Т. 54. № 6. С. 1033–1047.

[5] *Боресков А. В., Харламов А. А., Марковский Н. Д.* Параллельные вычисления на GPU. Архитектура и программная модель CUDA. — М.: Изд-во Московского ун-та, 2012. 336 с.

[6] *Cheng J., Grossman M., McKercher T.* Professional CUDA C programming. — New York, NY, USA: Wrox, 2014. 528 p.

[7] *Lichman M.* UCI Machine Learning Repository. Irvine, CA, USA: University of California, School of Information and Computer Science, 2013. `http://archive.ics.uci.edu/ml`.

[8] *Дюкова Е. В., Журавлев Ю. И., Песков Н. В., Сахаров А. А.* Обработка вещественнозначной информации логическими процедурами распознавания // Искусственный интеллект, 2004. № 2, С. 80–85.

[9] *Quinlan J. R.* C4.5: Programs for machine learning. San Mateo, CA, USA: Morgan Kaufmann, 1993. 302 p.

[10] *Peng L., Lei L.* A review of missing data treatment methods // Int. J. Intelligent Information Management Syst. Technol., 2005. Vol. 1. No. 3. P. 412–419.

[11] *Marlin B. M.* Missing data problems in machine learning. Department of Computer Science, University of Toronto, 2008. PhD Thesis. 156 p.

[12] *Kohavi R., Kunz C.* Option decision trees with majority votes // Conference (International) on Machine Learning, 1997. P. 161–169.

[13] *Kuncheva L. I.* Combining pattern classifiers methods and algorithms. — Hoboken, NJ, USA: John Wiley & Sons, Inc., 2004. 350 p.

Поступила в редакцию 20.12.2015

Synthesis of full decision tree with using heterogeneous systems on the basis of CUDA technology

I. E. Genrikhov

ingvar1485@rambler.ru

Mobile park IT, 21/1 Panfilova st., Khimki, Moscow Region, Russia

The article is devoted to classification algorithms based on full decision trees. Due to the decision tree construction under consideration, all the features satisfying a branching criterion are taken into account at each special vertex. The main drawback of full decision tree is significantly larger the time for synthesis of tree compared with the classical decision tree. The issues are considered of reducing the time for synthesis of full decision tree using CUDA technology. This technology allows one to use a large number of GPU cores to speed up the execution of complex computing. The results of the testing are given on model and real tasks. It is shown that the use of CUDA technology allows one to significantly reduce the time for synthesis of the full decision tree (more than 10 times) only in cases where the training set contains large number of attributes and/or learning objects and the information are either real-valued or integer large complexity.

Keywords: *precedent-based pattern recognition problem; full decision tree; CUDA; heterogeneous system*

DOI: 10.21469/22233792.2.1.05

References

[1] Zhuravlev, Yu. I., V. V. Ryazanov, and O. V. Senko. 2006. *Raspoznavanie. Matematicheskie metody. Programmnaya sistema. Prakticheskoe primenenie* [Recognition. Mathematical methods. Software system. Practical applications]. Moscow: Phasis. 176 p.

[2] Djukova, E. V., and N. V. Peskov. 2007. A classification algorithm based on the complete decision tree. *J. Pattern Recogn. Image Anal.* 17(3):363–367.

[3] Genrikhov, I. E. 2013. Postroenie i issledovanie polnykh reshayuschcikh derev'ev dlya zadach klassifikatsii po pretsedentam [Construction and research the full decision trees for classification problems by precedents]. Moscow. PhD Thesis. 169 p.

[4] Genrikhov, I. E. 2014. Issledovanie oboshchayushchey sposobnosti polnogo reshayushchego dereva [Analysis of the generalization ability of a full decision tree]. *J. Comput. Math. Math. Phys.* 54(6):1046–1059.

[5] Boreskov, A. V., A. A. Harlamov, and N. D. Markovskii. 2012. *Parallel'nye vychisleniya na GPU. Arkhitektura i programmnaya model' CUDA* [Parallel computing on the GPU. The architecture and programming model of CUDA]. Moscow: Moscow University Publs. 336 p.

[6] Cheng, J., M. Grossman, and T. McKercher. 2014. *Professional CUDA C programming*. New York, NY: Wrox. 528 p.

[7] Lichman, M. 2013. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Available at: `http://archive.ics.uci.edu/ml` (accessed August 14, 2016).

[8] Djukova, E. V., Yu. I. Zhuravlev, N. V. Peskov, and A. A. Saharov. 2004. Obrabotka veshchestvennoznachnoy informatsii logicheskimi protsdurami raspoznavaniya [Processing a real-valued information with logical recognition procedures]. *Artificial Intelligence* 2: 80–85.

[9] Quinlan, J. R. 1993. *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann. 302 p.

[10] Peng, L., and L. Lei. 2005. A review of missing data treatment methods. *Int. J. Intelligent Information Management Syst. Technol.* 1(3):412–419.

[11] Marlin, B. M. 2008. *Missing data problems in machine learning*. Department of Computer Science University of Toronto. PhD Thesis. 156 p.

[12] Kohavi, R., and C. Kunz. 1997. Option decision trees with majority votes. *Conference (International) on Machine Learning*. 161–169.

[13] Kuncheva, L. I. 2004. *Combining pattern classifiers methods and algorithms*. Hoboken, NJ: John Wiley & Sons, Inc. 350 p.

Received December 20, 2015

Многоклассовое распознавание образов в пространстве представлений с многоуровневым разрешением*

М. М. Ланге, С. Н. Ганебных, А. М. Ланге
lange_mm@ccas.ru, sng@ccas.ru
ФИЦ «Информатика и управление» РАН, г. Москва, ул. Вавилова, 44/2

Рассматривается метрическая схема распознавания образов, порождаемых многоклассовым источником изображений, в пространстве древовидных представлений с многоуровневым разрешением. Для различных уровней разрешения на множестве представлений введено семейство мер различия образов и разделяющих функций в форме правдоподобий по классам. Решение базируется на критерии голосования значений разделяющих функций и допускает отказ. Разработана процедура обучения, включающая отбор эталонов и оптимизацию параметров. Построен параметрический решающий алгоритм, включающий стратегии иерархического и переборного поиска решения в многоуровневой сети эталонов. Получена аналитическая оценка вычислительной сложности решающего алгоритма. Эффективность классификаторов с различными параметрами демонстрируется ROC-кривыми и эмпирическими зависимостями доли ошибок от вычислительной сложности для объединенного источника изображений подписей, жестов руки и лиц.

Ключевые слова: *древовидное представление; разделяющая функция; многоклассовое распознавание образов; многоуровневая сеть эталонов; ROC-кривая; иерархический поиск; вычислительная сложность*

DOI: 10.21469/22233792.2.1.06

1 Введение

Во многих приложениях эффективность методов распознавания образов целесообразно оценивать в терминах соотношения характеристик качества и трудоемкости, которые определяются вероятностью ошибок и вычислительной сложности алгоритмов принятия решения. Увеличение информации о распознаваемых объектах, как правило, позволяет уменьшить вероятность ошибки, но сопряжено с ростом вычислительной сложности решающих алгоритмов, что приводит к снижению быстродействия распознающих устройств (классификаторов). Острота проблемы существенно возрастает с увеличением числа классов и, в частности, в случае распознавания биометрических объектов, когда число классов определяется количеством персон, данные о которых хранятся в памяти классификатора, и может составлять десятки и сотни тысяч. Поэтому в рамках многоклассового распознавания важной задачей является уменьшение вычислительной сложности решающего алгоритма при сохранении малой вероятности ошибок. Для решения этой задачи естественно использовать структурные методы представления данных с многоуровневым разрешением, которые позволяют строить решающие алгоритмы на основе иерархических процедур направленного поиска решения.

Структурные представления изображений относятся к описаниям, которые ориентированы на ускорение процедур анализа изображений. Среди таких описаний особое место

*Работа выполнена при частичной финансовой поддержке РФФИ, проекты №№ 15-01-04671, 15-07-09324 и 15-07-07516.

занимают древовидные и пирамидальные представления с многоуровневым разрешением [1–3], которые позволяют ускорить процедуры сегментации и выделения информативных объектов на изображениях. Древовидные и пирамидальные представления растровых изображений не инвариантны к аффинным преобразованиям объектов изображения и избыточны, поскольку содержат описания как информативных объектов (образов), так и фона.

Отмеченные недостатки, как правило, отсутствуют у представлений образов, выделенных на изображениях. Известны представления геометрических форм в форме графов [4] и скелетов [5]. Такие представления инвариантны к преобразованиям поворота и смещения объекта в поле изображения. Однако существует проблема построения меры на множестве таких представлений, связанная с неопределенностью установления соответствия фрагментов представлений. Развитие структурных представлений образов для эффективного распознавания в терминах соотношения «качество–сложность» заключается в построении инвариантных древовидно-структурированных описаний образов, допускающих быстрое вычисление мер различия и сходства таких представлений. В работе [6] предложен метод представления образов, заданных полутоновыми изображениями, в форме деревьев эллиптических примитивов и введена мера различия образов на множестве таких представлений. Показано, что рассмотренные древовидные представления позволяют построить эффективный метрический классификатор на основе голосования эталонов с использованием решающего алгоритма, вычислительная сложность которого существенно меньше сложности переборного алгоритма. Сокращение вычислительной сложности решающего алгоритма достигнуто за счет многоуровневого разрешения используемых представлений и сужения зоны поиска решения (вычисления меры) на последовательных уровнях базы эталонов. Быстрый алгоритм вычисления меры с использованием деревьев рассмотрен также в работе [7].

В наиболее полной постановке, задача многоклассового распознавания с возможностью отказа исследована в работе [8]. В указанной работе для серии источников образов приведены сравнительные доли ошибок при различных методах распознавания в векторных пространствах признаков. Однако в этой работе отсутствуют оценки вычислительной сложности решающих алгоритмов, которые использованы в рассмотренных методах. Метод многоклассового распознавания образов в пространстве древовидных представлений с метрическим решающим правилом, не предусматривающим отказа, исследован авторами в работе [9] в терминах соотношения «качество–сложность». В настоящей работе дано обобщение этого метода для решающего правила с отказом и продемонстрирована его эффективность для распознавания лиц, жестов и подписей. Характеристики эффективности приведены в терминах ROC-кривых (Receiver Operating Characteristics) [10] и зависимостей доли ошибок от вычислительной сложности решающего алгоритма.

2 Источник образов и модель распознавания

Пусть источник порождает множество изображений, а выделенные на них образы образуют множество $\mathbf{X} = \{\mathbf{x}\}$ объектов, которые могут быть представлены в форме векторов, деревьев, графов и т. п. Множество $\mathbf{X}$ содержит объекты, относящиеся к множеству классов $\Omega = \{\omega_0, \omega_1, \ldots, \omega_c\}$, $c \geqslant 1$, в котором каждый положительный класс из подмножества $\Omega_c = \{\omega_i,\ i = 1, \ldots, c\}$ включает объекты с одинаковой семантикой, определяемой номером i, а нулевой класс ω_0 включает все прочие объекты. На множестве классов Ω задано априорное распределение $\{p(\omega_i) \geqslant 0, i = 0, \ldots, c : \sum_{i=0}^{c} p(\omega_i) = 1\}$, которое дает вероятности $p(\Omega_c) = \sum_{i=1}^{c} p(\omega_i) = \theta \geqslant 0$ и $p(\omega_0) = 1 - \theta$; на множестве объектов $\mathbf{X}$ определена

мера различия $d(\mathbf{x}, \hat{\mathbf{x}}) \geqslant 0$ любой пары объектов $\mathbf{x} \in \mathbf{X}$, $\hat{\mathbf{x}} \in \mathbf{X}$. Предполагается, что распределение $\{\theta, 1-\theta\}$ неизвестно, но известно условное распределение на подмножестве $\Omega_c : \{p(\omega_i|\Omega_c) = p(\omega_i)/\sum_{i=1}^{c} p(\omega_i),\ i = 1, \ldots, c\}$.

Будем считать, что задано обучающее множество

$$\mathbf{X}^{\text{train}} = \left\{\mathbf{X}_i^{\text{train}} = \{\mathbf{x}_{ij}\}_{j=1}^{m_i}\right\}_{i=0}^{c} \subset \mathbf{X}$$

мощности $\|\mathbf{X}^{\text{train}}\| = \sum_{i=0}^{c} m_i = m$, в котором кластеры с положительными номерами образуют подмножество $\mathbf{X}_{\Omega_c}^{\text{train}} = \{\mathbf{X}_i^{\text{train}}\}_{i=1}^{c}$, а кластер с нулевым номером дает подмножество $\mathbf{X}_{\omega_0}^{\text{train}} = \mathbf{X}_0^{\text{train}}$. На подмножестве $\mathbf{X}_{\Omega_c}^{\text{train}}$ отобраны наборы эталонных объектов, образующие множество

$$\hat{\mathbf{X}} = \left\{\hat{\mathbf{X}}_i = \{\hat{\mathbf{x}}_{ij}\}_{j=1}^{\hat{m}_i} \subseteq \mathbf{X}_i^{\text{train}}\right\}_{i=1}^{c} \tag{1}$$

мощности $\|\hat{\mathbf{X}}\| = \sum_{i=1}^{c} \hat{m}_i = \hat{m}_i$, где $\hat{m}_i \leqslant m_i$ и $\hat{m} \leqslant m$. Для пары объектов $\mathbf{x} \in \mathbf{X}$ и $\hat{\mathbf{x}} \in \hat{\mathbf{X}}$ вводится ядро

$$K_d(\mathbf{x}, \hat{\mathbf{x}}) = e^{-d((\mathbf{x}, \hat{\mathbf{x}})} \tag{2}$$

по мере $d(\mathbf{x}, \hat{\mathbf{x}})$. Ядро (2) и нормированные по классам подмножества Ω_c весовые коэффициенты эталонов

$$\mathbf{W} = \{\mathbf{W}_i = \{w(\hat{\mathbf{x}}_{ij}) \geqslant 0\}_{j=1}^{\hat{m}_i} : \sum_{j=1}^{\hat{m}_i} w(\hat{\mathbf{x}}_{ij}) = 1\}_{i=1}^{c} \tag{3}$$

порождают разделяющие функции [11]

$$g_i(\mathbf{x}) = \sum_{j=1}^{\hat{m}_i} w(\hat{\mathbf{x}}_{ij}) K_d(\mathbf{x}, \hat{\mathbf{x}}_{ij}),\ i = 1, \ldots, c\,, \tag{4}$$

которые эквивалентны потенциалам в точке $\mathbf{x}$, создаваемым наборами эталонов $\hat{\mathbf{X}}_i$, $i = 1, \ldots, c$ с весами (3). Разделяющие функции (4) и набор порогов $\boldsymbol{\Delta} = \{\Delta_i,\ i = 1, \ldots, c\}$ дают следующее решающее правило определения номера класса для объекта $\mathbf{x} \in \mathbf{X}$:

$$i^*(\mathbf{x}) = \max_{i=1}^{c} [g_i(\mathbf{x}) \geqslant \Delta_i] \arg\max_{i=1}^{c} \left(g_i(\mathbf{x})[g_i(\mathbf{x}) \geqslant \Delta_i]\right), \tag{5}$$

где $[f]$ – индикатор, принимающий значение 1, если условие f выполняется, и значение 0 в противном случае.

В общем случае решение (5) соответствует критерию голосования разделящих функций [11], дополненному отказом от распознавания. В настоящей работе исследуются классификаторы по критерию (5), в котором используются наборы эталонов (1) и весовые коэффициенты

$$w(\hat{\mathbf{x}}_{ij}) = [\hat{\mathbf{x}}_{ij} = \arg\min_{k=1}^{\hat{m}_i} d(\mathbf{x}, \hat{\mathbf{x}}_{ik})],\ i = 1, \ldots, c\,,$$

которые приводят к модификациям критерия ближайшего соседа. В любом случае при выбранном множестве весов эталонов $\mathbf{W}$ вида (3), качество решения (5) определяется условными вероятностями ошибок $\varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\Omega_c\right)$ и $\varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\omega_0\right)$ соответственно по «своим» (из подмножества классов Ω_c) и «чужим» (из класса ω_0) объектам. Полная вероятность ошибки определяется математическим ожиданием

$$\varepsilon_\theta\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}\right) = \theta\varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\Omega_c\right) + (1-\theta)\varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\omega_0\right)\,. \tag{6}$$

Полагая, что θ является случайной величиной с плотностью $p(\theta) = 1, 0 \leqslant \theta \leqslant 1$, и интегрируя функцию (6) по θ с указанной плотностью, получим функционал ошибок

$$\varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}\right) = \int_0^1 \varepsilon_\theta\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}\right) p(\theta) d\theta = \frac{1}{2}\left(\varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\Omega_c\right) + \varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\omega_0\right)\right). \quad (7)$$

В рамках рассматриваемой модели, процедура обучения сводится к отбору множества эталонов $\hat{\mathbf{X}}$ вида (1) с параметрами $\hat{m}_i < m_i$, $i = 1, \ldots, c$, и выбору пороговых значений $\boldsymbol{\Delta}$, которые минимизируют функционал ошибок $\varepsilon^{\text{train}}(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W})$ вида (7) на объектах обучающего множества $\mathbf{X}^{\text{train}}$. Оптимизация классификатора сводится к нахождению пары

$$(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*) = \arg \min_{\hat{\mathbf{X}}, \boldsymbol{\Delta}: \hat{m} < m} \varepsilon^{\text{train}}(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}). \quad (8)$$

Множество эталонов и набор порогов, удовлетворяющие соотношению (8), определяют оптимальный классификатор с решающим правилом (5), которое допускает точную или приближенную реализацию на алгоритме a_α с параметром α. Эффективность классификатора с оптимальными параметрами (8) и алгоритмом a_α определяется двумя характеристиками: вероятностью ошибок $\varepsilon_\alpha^{\text{test}}\left(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}\right)$ вида (7), вычисляемой на тестовом множестве $\mathbf{X}^{\text{test}} = \mathbf{X} \setminus \mathbf{X}^{\text{train}}$, и вычислительной сложностью $C_\alpha(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W})$, измеряемой количеством операций, затрачиваемых алгоритмом a_α на один объект. Полагая, что с увеличением сложности $C_\alpha\left(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}\right)$ (путем варьирования параметром α) вероятность ошибок $\varepsilon^{\text{test}}\left(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}\right)$ не возрастает и исключая α при заданной сложности $C^* > 0$, алгоритм a_α порождает функцию «качество–сложность»

$$\varepsilon\left(C^*|\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}\right) = \varepsilon_\alpha^{\text{test}}\left(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}\right) : \alpha = \arg\left(C_\alpha\left(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}\right) = C^*\right) \quad (9)$$

для классификатора с параметрами $\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}$.

Ниже исследуется классификатор с решающим правилом (5), реализующим критерий ближайшего соседа в пространстве древовидных представлений объектов с многоуровневым разрешением. Дано описание способа построения древовидных представлений объектов, и на множестве представлений построена мера различия объектов, используемая в ядре (2). В заданном пространстве представлений определены разделяющие функции вида (4) и построен параметрический решающий алгоритм, который допускает существенное уменьшение вычислительной сложности по сравнению с алгоритмом полного перебора при достаточно малом увеличении доли ошибок. Для источников изображений лиц, жестов руки и подписей при различных способах отбора эталонов построены оценки функции «качество–сложность» (9) путем варьирования параметром решающего алгоритма.

3 Представление образов и решающий алгоритм

В этом разделе рассматривается способ древовидного представления образов с многоуровневым разрешением, вводится семейство мер различия образов и соответствующие этим мерам разделяющие функции на различных уровнях разрешения. Используя разделяющие функции всех уровней, строится иерархический алгоритм, реализующий решающее правило (5).

3.1 Многоуровневое описание образов примитивами

Множество объектов источника $\mathbf{X}$ включает образы, выделенные на полутоновых изображениях и удовлетворяющие некоторым ограничениям. Налагаемые ограничения формируют множество допустимых образов, удовлетворяющих следующему определению.

Определение 1. Образ, заданный на изображении односвязным или многосвязным набором пикселей, яркости которых соответствуют их «массам», считается допустимым, если набор пикселей объекта образует компактное или распределенное двумерное твердое тело с однозначно идентифицируемой декартовой системой собственных координат.

Для множества образов $\mathbf{X}$, удовлетворяющих определению 1, в работе [6] предложен способ их представления бинарными деревьями эллиптических примитивов. Способ базируется на дихотомическом разбиении любого образа $\mathbf{x} \in \mathbf{X}$ на сегменты (наборы пикселей) и аппроксимации каждого сегмента с номером n эллиптическим примитивом Q_n, который соответствует вершине бинарного дерева. Каждый делимый сегмент с номером n порождает пару новых сегментов следующего уровня с номерами $2n{+}1$ и $2n{+}2$ и соответствующую пару аппроксимирующих примитивов. Нумерация сегментов и примитивов производится с учетом ориентации оси разбиения делимого сегмента.

Корневой примитив Q_0 аппроксимирует исходный образ $\mathbf{x}$ и имеет номер $n = 0$. Концевые вершины дерева примитивов соответствуют неделимым сегментам с числом пикселей не более заданного значения (например, сегменты, состоящие из одного пикселя). Аппроксимирующие примитивы строятся в декартовой системе собственных координат аппроксимируемых сегментов. В случае отсутствия у сегмента с номером n однозначно идентифицируемой системы собственных координат примитив Q_n строится в системе координат сегмента-родителя с номером $\lfloor (n-1)/2 \rfloor$. Таким образом, для построения дерева примитивов достаточно существования идентифицируемой системы собственных координат исходного образа, который является корневым сегментом с номером $n = 0$.

Введем следующие характеристики дихотомического разбиения образа $\mathbf{x}$: N_l — множество номеров концевых вершин на l-м шаге; $x'_l = \{P_n, n \in \mathrm{N}_l : \cup_n P_n = \mathbf{x}\}$ — слой сегментов; $x_l = \{Q_n, n \in \mathrm{N}_l\}$ — слой аппроксимирующих примитивов на l-м шаге. Базовыми операциями над сегментом $P_n \in x'_l$ являются: операция формирования сегментов следующего уровня (segmentation)

$$F_{\mathrm{seg}}(P_n) = \begin{cases} (P_{2n+1}, P_{2n+2}), & \text{если } P_n \text{ — делимый;} \\ P_n, & \text{если } P_n \text{ — неделимый,} \end{cases}$$

и операция аппроксимации

$$F_{\mathrm{app}}(P_n) = Q_n,$$

которая сохраняет примитив, если он выбран для сегмента на предыдущем шаге, либо строит примитив на текущем шаге, если сегмент не имел аппроксимации на предыдущем шаге. Введенные понятия позволяют сформулировать следующий рекурсивный алгоритм представления (representation) объекта $\mathbf{x} \in \mathbf{X}$.

Алгоритм $a_{\mathbf{rep}}$. Начальные условия при $l = 0$:

$$P_0 = \mathbf{x},\ x'_0 = P_0; \quad Q_0 = F_{\mathrm{app}}(P_0), \quad x_0 = Q_0; \ \ L \geqslant 1.$$

Вычисление слоев сегментов и примитивов на шагах $l = 1, \ldots, L-1$:

$$x'_l = \{F_{\mathrm{seg}}(P_n), \forall P_n \in x'_{l-1}\}; \quad x_l = \{F_{\mathrm{app}}(P_n), \forall P_n \in x'_l\}.$$

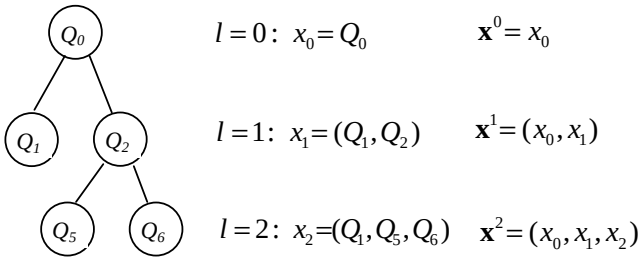


Рис. 1 Пример полного бинарного дерева глубины $L = 2$

Алгоритм a_{rep} строит для объекта $\mathbf{x} \in \mathbf{X}$, удовлетворяющего определению 1, многоуровневое представление

$$a_{\text{rep}}(\mathbf{x}) = \mathbf{x}^L = (x_0, \ldots, x_l, \ldots, x_L) \tag{10}$$

в форме полного бинарного дерева глубины L (бинарное дерево называется полным, если любая его промежуточная вершина имеет две исходящие ветви). В (10) любое поддерево $\mathbf{x}^l = (x_0, \ldots, x_l)$ глубины $l = 0, \ldots, L$ задано последоватеьльностью слоев примитивов, в которой слой x_l образован концевыми вершинами этого поддерева. Такое представление обеспечивает растущее с увеличением l разрешение, определяемое числом примитивов в слое x_l. Пример структуры вида (10) дан на рис. 1 для случая $L = 2$.

Каждый эллиптический примитив определяется набором параметров

$$Q_n = (\mathbf{r}_n, \mathbf{u}_n, \mathbf{v}_n, z_n), \tag{11}$$

где n — номер вершины дерева, соответствующий сегменту образа; $\mathbf{r}_n$ — вектор центра эллипса; $\mathbf{u}_n$ и $\mathbf{v}_n$ — векторы большой и малой полуосей; z_n — средний уровень яркости пикселей в аппроксимируемом сегменте. Координаты центра, определяющие вектор $\mathbf{r}_n$, и радиусы, определяющие векторы $\mathbf{u}_n$ и $\mathbf{v}_n$, вычисляются как параметры эллипса рассеяния соответствующего сегмента. Векторы $\mathbf{r}_n$, $\mathbf{u}_n$ и $\mathbf{v}_n$ задаются в собственной системе координат объекта $\mathbf{x}$. Разбиение делимого сегмента с номером $n \geqslant 0$ производится осью, проходящей через центр этого сегмента и параллельной одной из собственных осей сегмента с номером $n = 0$. Ось разбиения на каждом шаге l фиксирована и меняется с увеличением номера шага. При указанной нумерации примитив Q_n находится в полном дереве на уровне $l = \lfloor \log_2(n+1) \rfloor$. Параметры каждого примитива (11) с номером $n \geqslant 0$ нормируются относительно корневого примитива с номером $n = 0$.

Примеры представлений подписи, жеста руки и лица слоями эллиптических примитивов, которые соответствуют уровням завершенного бинарного дерева глубины $L = 8$ (с числом вершин 2^l, $l = 0, \ldots, L$) даны на рис. 2.

Построение дерева примитивов в собственных координатах объекта и нормировка их параметров позволяют сформулировать следующее свойство представления (10).

Утверждение 1. При достаточно малом размере пикселей и большом числе уровней квантования яркостей бинарное дерево эллиптических примитивов с точностью до размера пикселя и уровня квантования инвариантно к преобразованиям поворота, смещения, изменения масштаба и уровня яркости представляемого объекта.

3.2 Мера различия объектов

В этом подразделе вводится семейство мер различия объектов на множестве представлений вида (10). Для определения меры различия объектов $\mathbf{x} \in \mathbf{X}$, $\hat{\mathbf{x}} \in \hat{\mathbf{X}}$, представленных поддеревьями $\mathbf{x}^l \in \mathbf{x}^L$, $\hat{\mathbf{x}}^l \in \hat{\mathbf{x}}^L$ глубины l, потребуются следующие определения.

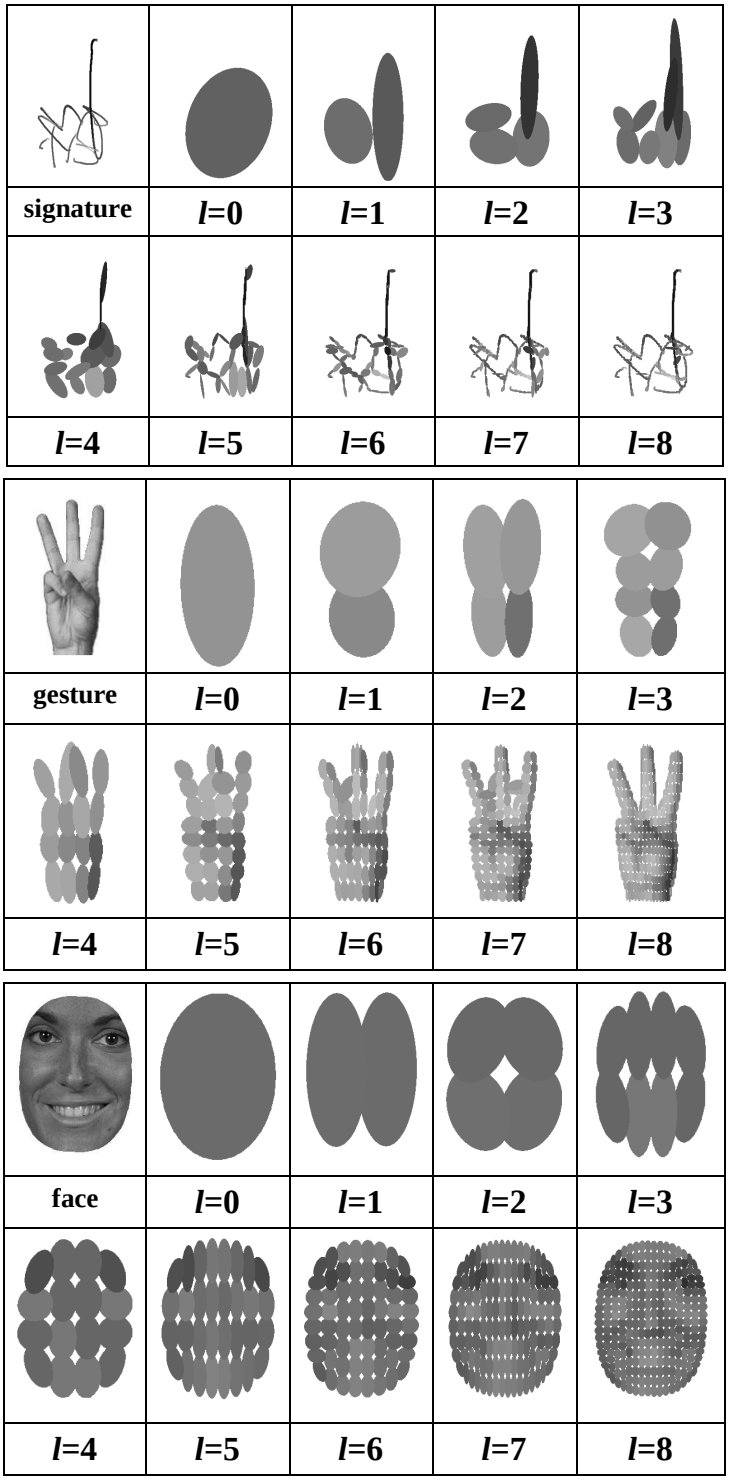


Рис. 2 Примеры представлений подписи, жеста руки и лица слоями эллиптических примитивов, образующими завершенные бинарные деревья глубины $L = 8$

Определение 2. Примитивы $Q_n \in \mathbf{x}^l$ и $\hat{Q}_n \in \hat{\mathbf{x}}^l$ в n-х вершинах поддеревьев $\mathbf{x}^l$ и $\hat{\mathbf{x}}^l$ называются соответственными. Множество пар соответственных примитивов в поддеревьях $\mathbf{x}^l$ и $\hat{\mathbf{x}}^l$ образует пересечение $\mathbf{x}^l \cap \hat{\mathbf{x}}^l$.

С учетом определения 2 введем для любой пары поддеревьев $\mathbf{x}^l$, $\hat{\mathbf{x}}^l$, $l = 1, \ldots, L$ следующие функции различия порядка l по параметрам соответственных примитивов:

$$\left.\begin{aligned}\rho_r(\mathbf{x},\hat{\mathbf{x}}) &= \sum_{n:(Q_n,\hat{Q}_n)\in\mathbf{x}^l\cap\hat{\mathbf{x}}^l} w_n\|\mathbf{r}_n-\hat{\mathbf{r}}_n\|;\\ \rho_{uv}(\mathbf{x},\hat{\mathbf{x}}) &= \sum_{n:(Q_n,\hat{Q}_n)\in\mathbf{x}^l\cap\hat{\mathbf{x}}^l} w_n(\|\mathbf{u}_n-\hat{\mathbf{u}}_n\|+\|\mathbf{v}_n-\hat{\mathbf{v}}_n\|)\,;\\ \rho_z(\mathbf{x},\hat{\mathbf{x}}) &= \sum_{n:(Q_n,\hat{Q}_n)\in\mathbf{x}^l\cap\hat{\mathbf{x}}^l} w_n\|z_n-\hat{r}_n\|,\end{aligned}\right\} \tag{12}$$

где w_n — весовой коэффициент соответственных примитивов с номером n:

$$w_n=\frac{\lfloor\log_2(n+1)\rfloor 2^{-\lfloor\log_2(n+1)\rfloor}}{\sum_{n:(Q_n,\hat{Q}_n)\in\mathbf{x}^l\cap\hat{\mathbf{x}}^l}\lfloor\log_2(n+1)\rfloor 2^{-\lfloor\log_2(n+1)\rfloor}}\,.$$

Нормы в (12) вычисляются в метрике L1. Функции (12) дают средние отклонения объектов обучающего множества $\mathbf{X}^{\text{train}}$ относительно объекта $\hat{\mathbf{x}}$:

$$\left.\begin{aligned}\sigma_r(\hat{\mathbf{x}}^l) &= \frac{1}{m-1}\sum_{\mathbf{x}^l:\,\mathbf{x}\in\mathbf{X}^{\text{train}}}\rho_r(\mathbf{x}^l,\hat{\mathbf{x}}^l)\,;\\ \sigma_{uv}(\hat{\mathbf{x}}^l) &= \frac{1}{m-1}\sum_{\mathbf{x}^l:\,\mathbf{x}\in\mathbf{X}^{\text{train}}}\rho_{uv}(\mathbf{x}^l,\hat{\mathbf{x}}^l)\,;\\ \sigma_z(\hat{\mathbf{x}}^l) &= \frac{1}{m-1}\sum_{\mathbf{x}^l:\,\mathbf{x}\in\mathbf{X}^{\text{train}}}\rho_z(\mathbf{x}^l,\hat{\mathbf{x}}^l),\end{aligned}\right\} \tag{13}$$

где m — число объектов в множестве $\mathbf{X}^{\text{train}}$. С учетом соотношений (12) и (13) мера различия порядка $l=1,\ldots,L$ для пары объектов $\mathbf{x}$, $\hat{\mathbf{x}}$ определяется функцией:

$$d^l(\mathbf{x},\hat{\mathbf{x}})=\frac{\rho_r(\mathbf{x}^l,\hat{\mathbf{x}}^l)}{\sigma_r(\hat{\mathbf{x}}^l)}+\frac{\rho_{uv}(\mathbf{x}^l,\hat{\mathbf{x}}^l)}{\sigma_{uv}(\hat{\mathbf{x}}^l)}+\frac{\rho_z(\mathbf{x}^l,\hat{\mathbf{x}}^l)}{\sigma_z(\hat{\mathbf{x}}^l)}, \tag{14}$$

которая является аналогом махаланобисовой меры [11] для трехмерного пространства признаков с метрикой L1.

3.3 Разделяющие функции и решающий алгоритм

Набор мер $\{d^l(\mathbf{x},\hat{\mathbf{x}}),\,l=1,\ldots,L\}$ вида (14) порождает набор разделяющих функций

$$\left\{g_i^l(\mathbf{x}),\,l=1,\ldots,L\right\}_{i=1}^{c} \tag{15}$$

вида (4), где l — порядок функции $g_i^l(\mathbf{x})$. Функции (15) используются для построения алгоритма поиска решения вида (5) с использованием параметрической стратегии сужения зоны поиска на последовательных уровнях $l=1,\ldots,L$. Предполагается, что наборы эталонов по классам в множестве (1) заданы сетью представлений

$$\left\{\hat{\mathbf{X}}_i^l,\,l=1,\ldots,L\right\}_{i=1}^{c}, \tag{16}$$

в которой $\hat{\mathbf{X}}_i^l$ — набор представлений поддеревьями глубины l для набора эталонов $\hat{\mathbf{X}}_i$. Стратегия сужения зоны поиска решения в сети (16) задается экспоненциальной функцией

$$c_l=\lfloor c2^{-\alpha(l-1)}\rfloor,\;l=1,\ldots,L\,, \tag{17}$$

где $\alpha \geqslant 0$ — свободный параметр, а c_l — число классов, для которых вычисляются разделяющие функции порядка l на соответствующем уровне сети (16). Стратегия (17) дает следующий алгоритм, реализующий решение (5).

Алгоритм a_α. На последовательных уровнях $l = 1, \ldots, L-1$ сети (16) вычисляются разделяющие функции (15) для c_l классов и среди них отбираются c_{l+1} классов с наибольшими значениями разделяющих функций; решение (5) принимается на уровне $l^* = (l < L)[c_{l+1} < 2] + (l = L)[c_L \geqslant 2]$, где $[f]$ — индикатор f.

В алгоритме a_α параметр $\alpha > 0$ соответствует стратегии направленного поиска решения (guided search algorithm $a^{\text{gs}}_{\alpha>0}$), а параметр $\alpha = 0$ — стратегии поиска решения на основе полного перебора (exhaustive search algorithm $a^{\text{es}}_{\alpha=0}$).

Вычислительная сложность C_α алгоритма a_α определяется количеством элементарных операций, затрачиваемых на вычисление меры различия $d^l(\mathbf{x}, \hat{\mathbf{x}})$ вида (14) между предъявляемым объектом $\mathbf{x}$ и всеми эталонными объектами $\hat{\mathbf{x}} \in \hat{\mathbf{X}}^*$, на уровнях $l = 1, \ldots, L$ сети (16). Множество эталонов $\hat{\mathbf{X}}^*$ отбирается согласно (8) на этапе обучения. Вычислительные затраты на вычисление разделяющих функций, их сортировку и отбор наибольших значений, а также затраты на сравнение с порогами малы по сравнению с затратами на вычисление меры и поэтому не учитываются. Элементарной операцией считается сравнение пары соответственных примитивов предъявляемого и эталонного объекта. Поскольку число примитивов в l-м слое x_l представления (10) не превосходит 2^l, сложность алгоритма a_α, реализующего стратегии (17), удовлетворяет оценке

$$C_\alpha \leqslant \sum_{l=1}^{L} 2^l \sum_{i=1}^{c} \hat{m}_i [i \in \mathrm{N}_{c_l}] \leqslant 2c \max_{i=1}^{c} \hat{m}_i \sum_{l=1}^{L} 2^{(1-\alpha)(l-1)}, \tag{18}$$

где $\hat{m}_i$ — число эталонов i-го класса; N_{c_l} — множество номеров классов, отобранных в сети (16) на l-м уровне ($\|\mathrm{N}_{c_l}\| = c_l$); $[f]$ — индикатор f. Модификация оценки (18) получена в работе [9].

При использовании совершенных представляющих деревьев вида (10), в которых уровень с номером $l = 1, \ldots, L$ содержит 2^l вершин-примитивов, оценка (18) позволяет оценить сверху отношение сложностей алгоритмов $a^{\text{gs}}_{\alpha>0}$ и $a^{\text{es}}_{\alpha=0}$. В случае применения деревьев указанного типа двойная сумма в (18) при $\alpha = 0$ дает точное значение сложности переборного решающего алгоритма $a^{\text{es}}_{\alpha=0}$:

$$C_{\alpha=0} = \sum_{l=1}^{L} 2^l \sum_{i=1}^{c} \hat{m}_i = 2(2^L - 1)c\hat{m}_{\text{mean}}\,, \tag{19}$$

где $\hat{m}_{\text{mean}} = c^{-1} \sum_{i=1}^{c} \hat{m}_i$. При $\alpha > 0$ неравенство (18) дает оценку сверху для сложности иерархического решающего алгоритма $a^{\text{gs}}_{\alpha>0}$:

$$C_{\alpha>0} \leqslant 2\frac{2^{(1-\alpha)L} - 1}{2^{1-\alpha} - 1} c\hat{m}_{\max}\,, \tag{20}$$

где $\hat{m}_{\max} = \max\limits_{i=1}^{c} \hat{m}_i$. Доля сложности алгоритма $a^{\text{gs}}_{\alpha>0}$ относительно сложности алгоритма $a^{\text{es}}_{\alpha=0}$ определяется отношением $\gamma = C_{\alpha>0}/C_{\alpha=0}$, которое с учетом (19) и (20) удовлетворяет оценке

$$\gamma \leqslant \frac{2^{(1-\alpha)L} - 1}{2^{(1-\alpha)} - 1} \, \frac{1}{2^L - 1} \, \frac{\hat{m}_{\max}}{\hat{m}_{\text{mean}}}\,. \tag{21}$$

Утверждение 2. Доля вычислительной сложности решающего алгоритма с параметром $\alpha \geqslant 1$ относительно сложности переборного алгоритма с параметром $\alpha = 0$ удовлетворяет оценке

$$\frac{C_{\alpha\geqslant 1}}{C_{\alpha=0}} \leqslant \frac{L}{2^L - 1} \frac{\hat{m}_{\max}}{\hat{m}_{\text{mean}}}.$$

Сформулированное утверждение следует из неравенства (21), в котором при $\alpha \geqslant 1$ точное значение суммы L членов убывающей геометрической прогрессии заменено оценкой сверху $(2^{(1-\alpha)L} - 1)/(2^{(1-\alpha)} - 1) \leqslant L$. Утверждение 2 демонстрирует экспоненциально растущий вычислительный выигрыш алгоритма $a^{\text{gs}}_{\alpha\geqslant 1}$ относительно алгоритма $a^{\text{es}}_{\alpha=0}$ с увеличением глубины представляющих деревьев.

4 Обучение классификатора

Процедура обучения предполагает отбор наборов эталонов по классам $\hat{\mathbf{X}}^*_i$, $i = 1, \ldots, c$, и получение оценок порогов Δ^*_i, $i = 1, \ldots, c$, которые образуют множества $\hat{\mathbf{X}}^* = \left\{\hat{\mathbf{X}}^*_i\right\}^c_{i=1}$ и $\Delta^* = \{\Delta^*_i\}^c_{i=1}$, удовлетворяющие условию (8). Строится оценка функционала ошибок (7) в форме зависимости от наборов эталонов $\hat{\mathbf{X}}_i$ и порогов Δ_i по классам с номерами $i = = 1, \ldots, c$. Предлагается рекурсивная процедура построения наборов эталонов $\hat{\mathbf{X}}_i$ и параметрическая стратегия выбора порогов Δ_i по классам подмножества Ω_c для нахождения оптимальных параметров $\left\{\hat{\mathbf{X}}^*_i, \Delta^*_i\right\}^c_{i=1}$ путем минимизации оценки скользящего контроля функционала ошибок.

4.1 Оценка функционала ошибок

Разделяющие функции (15), вычисляемые согласно (4) по мерам порядка $l = 1, \ldots, L$, принимают значения на отрезке [0,1]. Пусть $\overline{g}^L_i$ — средние значения разделяющих функций по S порядковым статистикам:

$$\overline{g}^L_i = \frac{1}{S} \sum_{s=1}^{S} g^L_{is},\ i = 1, \ldots, c\,, \tag{22}$$

где g^L_{is} — s-е наибольшее значение (порядковая статистика) для функции $g^L_i(\mathbf{x})$ на обучающем множестве $\mathbf{X}^{\text{train}}$. Средние значения (22) используются в стратегии выбора порогов $\Delta_i = \delta \overline{g}^L_i$, $i = 1, \ldots, c$, с параметром $\delta \in [0, 1]$. Ниже предлагается оценка функционала ошибок (7) в виде зависимости от наборов эталонов $\left\{\hat{\mathbf{X}}_i\right\}^c_{i=1}$ и порогового параметра δ.

Процедура построения оценок вероятностей ошибок классификации по своим и чужим объектам базируется на схеме принятия решений, заданной графом на рис. 3. Состояния графа соответствуют классам $\omega_0, \omega_1, \ldots, \omega_c$, а дуги — решениям, принимаемым

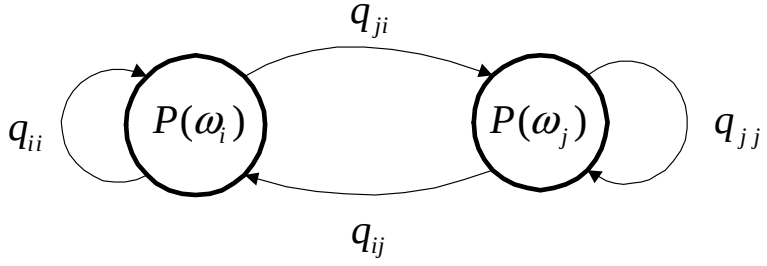


Рис. 3 Схема принятия решений

на классе ω_j по объектам класса ω_i. Состояния графа помечены безусловными априорными вероятностями $p(\omega_i)$, $i = 0, \ldots, c$, а дуги — условными вероятностями ошибочных q_{ji}, $j = 0, \ldots, c$, $j \neq i$, или верных q_{ii} решений по объектам из класса $\omega_i \in \Omega$.

С учетом схемы на рис. 3 условные вероятности ошибок в функционале (7) по своим и чужим объектам равны

$$\varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\Omega_c\right) = \sum_{i=1}^{c}(1 - q_{ii})p(\omega_i|\Omega_c)\,; \tag{23}$$

$$\varepsilon\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\omega_0\right) = \sum_{i=1}^{c} q_{i0}\,. \tag{23'}$$

Полагая, что вероятности q_{ii} и q_{i0} являются функциями переменных $\hat{\mathbf{X}}_i$ и δ, зависящими от весов эталонов $\mathbf{W}_i$, $i = 1, \ldots, c$, в (3) и значений $\overline{g}_i^L$ вида (22), получаем следующие оценки вероятностей верных и ложных решений на классе ω_i, $i = 1, \ldots, c$, в терминах true positive rate (TPR) и false positive rate (FPR) [10]:

$$\hat{q}_{ii} = \varepsilon_{\text{tpr}}\left(\hat{\mathbf{X}}_i, \delta, \mathbf{W}_i\right); \quad \hat{q}_{i0} = \varepsilon_{\text{fpr}}\left(\hat{\mathbf{X}}_i, \delta, \mathbf{W}_i\right).$$

Применение этих оценок в формулах (23) дает при равномерном распределении $p(\omega_i|\Omega_c) = 1/c$, $i = 1, \ldots, c$, следующие оценки условных вероятностей ошибок:

$$\hat{\varepsilon}\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\Omega_c\right) = \frac{1}{c}\sum_{i=1}^{c}\left(1 - \varepsilon_{\text{tpr}}\left(\hat{\mathbf{X}}, \delta, \mathbf{W}_i\right)\right) = 1 - \text{TPR}\,; \tag{24}$$

$$\hat{\varepsilon}\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}|\omega_0\right) = \sum_{i=1}^{c}\varepsilon_{\text{fpr}}\left(\hat{\mathbf{X}}, \delta, \mathbf{W}_i\right) = \text{FPR}\,. \tag{24'}$$

Утверждение 3. В рамках принятых допущений оценка функционала ошибок (7) с учетом соотношений (24) имеет вид:

$$\hat{\varepsilon}\left(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W}\right) = \sum_{i=1}^{c}\varepsilon\left(\hat{\mathbf{X}}_i, \delta, \mathbf{W}_i\right), \tag{25}$$

где $\varepsilon\left(\hat{\mathbf{X}}_i, \delta, \mathbf{W}_i\right)$ — оценка доли ошибок элементарного классификатора, образованного парой классов $\{\omega_0, \omega_i\}$:

$$\varepsilon\left(\hat{\mathbf{X}}_i, \delta, \mathbf{W}_i\right) = \frac{1}{2}\left(\frac{1}{c}\left(1 - \varepsilon_{\text{tpr}}\left(\hat{\mathbf{X}}_i, \delta, \mathbf{W}_i\right)\right) + \varepsilon_{\text{fpr}}\left(\hat{\mathbf{X}}_i, \delta, \mathbf{W}_i\right)\right). \tag{26}$$

4.2 Отбор эталонов

Отбор эталонов в каждом кластере $\mathbf{X}_i^{\text{train}} \subset \mathbf{X}^{\text{train}}$, $i = 1, \ldots, c$, обучающего множества выполняется независимо с использованием жадного алгоритма, предложенного в работе [9]. Алгоритм отбора эталонов базируется на процедуре дихотомического разбиения кластера $\mathbf{X}_i^{\text{train}}$ мощности m_i на фрагменты и выборе в каждом фрагменте эталонного объекта $\hat{\mathbf{x}}$, относительно которого объекты этого фрагмента имеют наименьшее рассеяние. Рассеяние вычисляется по мере вида (14) наибольшего порядка L.

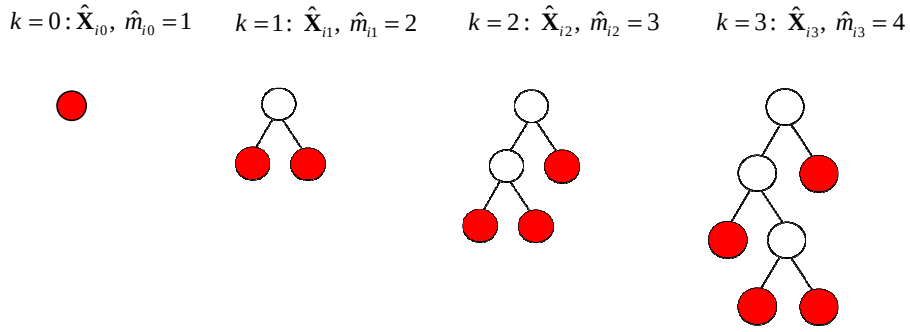


Рис. 4 Пример наборов эталонов в четырехслойном бинарном дереве

На шаге с номером $k=0$ в кластере $\mathbf{X}_i^{\text{train}}$ выбирается единственный эталон. На каждом последующем шаге дихотомии с номером $k=1,\ldots,m_i-1$ делимым фрагментом является фрагмент, объекты которого имеют наибольшее рассеяние относительно своего эталона (на первом шаге делимым сегментом является весь кластер). В результате, на последовательных шагах рекурсивной процедуры формируются наборы эталонов $\hat{\mathbf{X}}_{ik}$ мощности $\|\hat{\mathbf{X}}_{ik}\| = \hat{m}_{ik} = k+1$, $k=0,\ldots,m_i-1$. Набор эталонов $\hat{\mathbf{X}}_{ik}$, формируемый на k-м шаге, соответствует слою концевых вершин в бинарном дереве глубины k. Пример бинарного дерева, содержащего четыре слоя наборов эталонов, дан на рис. 4.

Формализация построения дерева наборов эталонов состоит в следующем. Пусть N_{ik} — множество номеров концевых вершин на k-м шаге дихотомического разбиения кластера $\mathbf{X}_i^{\text{train}}$, $\mathbf{X}_{ik} = \{\mathrm{X}_{in},\, n \in \mathrm{N}_{ik} : \bigcup\limits_n \mathrm{X}_{in} = \mathbf{X}_i^{\text{train}}\}$ — набор фрагментов; $\hat{\mathbf{X}}_{ik} = \{\hat{\mathbf{x}}_{in} \in \mathrm{X}_{in},\, n \in \mathrm{N}_{ik}\}$ — набор эталонов k-го шага. Эталон $\hat{\mathbf{x}}_{in} \in \mathrm{X}_{in}$ выбирается из условия:

$$\hat{\mathbf{x}}_{in} = \arg\min_{\hat{\mathbf{x}}\in \mathrm{X}_{in}} \max_{\mathbf{x}\in \mathrm{X}_{in}} d^L(\mathbf{x},\hat{\mathbf{x}})\,, \tag{27}$$

которое обеспечивает минимальное рассеяние $D(X_{in}) = \max\limits_{\mathbf{x}\in \mathrm{X}_{in}} d^L(\mathbf{x},\hat{\mathbf{x}}_{in})$ объектов фрагмента X_{in} относительно выбранного эталона $\hat{\mathbf{x}}_{in}$. В наборе $\mathbf{X}_{ik}$ фрагментов k-го шага дихотомии делимым является фрагмент $\mathrm{X}_{in} \subset \mathbf{X}_{ik}$ с наибольшим рассеянием:

$$D(\mathrm{X}_{in}) = \max_{\tilde{\mathrm{X}}_{in}\subset \mathbf{X}_{ik}} D(\tilde{\mathrm{X}}_{in}).$$

Для разбиения делимого фрагмента $\mathrm{X}_{in} \to (\mathrm{X}_{i2n+1}, \mathrm{X}_{i2n+2})$ в нем выбираются опорные объекты $(\mathbf{x}', \mathbf{x}'') \in \mathrm{X}_{in}$, которые образуют пару наиболее удаленных друг от друга объектов по мере $d^L(\mathbf{x}', \mathbf{x}'')$. Фрагменты $(\mathrm{X}_{i2n+1}, \mathrm{X}_{i2n+2})$ формируются на наборах, образованных наиболее близкими объектами к соответствующим опорным объектам $(\mathbf{x}', \mathbf{x}'')$ по указанной мере.

Базовыми операциями над фрагментом $\mathrm{X}_{in} \subset \mathbf{X}_{ik}$ являются: операция формирования фрагментов следующего уровня (fragmentation)

$$f_{\text{frag}}(\mathrm{X}_{in}) = \begin{cases} (\mathrm{X}_{i2n+1}, \mathrm{X}_{i2n+2}), & \mathrm{X}_{in} \text{ — делимый;} \\ \mathrm{X}_{in}, & \mathrm{X}_{in} \text{ — неделимый} \end{cases}$$

и операция отбора (selection) эталона

$$f_{\text{sel}}(\mathrm{X}_{in}) = \hat{\mathbf{x}}_{in}$$

в соответствии с условием (27). Указанные операции позволяют сформулировать рекурсивный алгоритм построения наборов эталонов в заданном классе.

Алгоритм $a_{\mathbf{sel}}$. Начальные условия при $k = 0$:

$$\mathrm{X}_{i0} = \mathbf{X}_i^{\mathrm{train}}; \quad \mathbf{X}_{i0} = \mathrm{X}_{i0}; \quad \hat{\mathbf{x}}_{i0} = f_{\mathrm{sel}}(\mathrm{X}_{i0}); \quad \hat{\mathbf{X}}_{i0} = \hat{\mathbf{x}}_{i0}; \quad m_i \geqslant 1.$$

Построение наборов фрагментов и наборов эталонов на шагах $k = 1, \ldots, m_i - 1$:

$$\mathbf{X}_{ik} = \{f_{\mathrm{frag}}(\mathrm{X}_{in}), \forall \mathrm{X}_{in} \subset \mathbf{X}_{i(k-1)}\};$$
$$\hat{\mathbf{X}}_{ik} = \{f_{\mathrm{sel}}(\mathrm{X}_{in}), \forall \mathrm{X}_{in} \subset \mathbf{X}_{ik}\}.$$

Алгоритм a_{sel} строит на кластере $\mathbf{X}_i^{\mathrm{train}}$ последовательность наборов эталонов

$$a_{\mathrm{sel}}(\mathbf{X}_i^{\mathrm{train}}) = (\hat{\mathbf{X}}_{i0}, \ldots, \hat{\mathbf{X}}_{ik}, \ldots, \hat{\mathbf{X}}_{i(m_i-1)}), \tag{28}$$

в которой набор $\hat{\mathbf{X}}_{ik} = \{\hat{\mathbf{x}}_{in}, \forall n \in \mathrm{N}_{ik}\}$ мощности $\hat{m}_{ik} = k + 1$ образует k-й слой в бинарном дереве. Выбор из последовательности (28) оптимального набора $\hat{\mathbf{X}}_{lk^*} = \hat{\mathbf{X}}_i^*$ мощности $\hat{m}_{ik^*} = \hat{m}_i^*$ осуществляется путем минимизации оценки скользящего контроля для доли ошибок вида (26).

4.3 Оптимизация параметров

Аддитивная форма оценки функционала ошибок, полученная в утверждении 3, сводит минимизацию доли ошибок $\hat{\varepsilon}^{\mathrm{train}}(\hat{\mathbf{X}}, \boldsymbol{\Delta}, \mathbf{W})$ вида (25) на заданном обучающем множестве $\mathbf{X}^{\mathrm{train}}$ к независимой минимизации долей ошибок $\varepsilon^{\mathrm{train}}(\hat{\mathbf{X}}_i, \delta, \mathbf{W}_i)$ вида (26) по наборам эталонов $\hat{\mathbf{X}}_i$, $i = 1, \ldots, c$, при фиксированных значениях $\delta \in [0, 1]$, взятых с шагом $\Delta\delta > 0$. Для элементарного классификатора на паре классов $\{\omega_0, \omega_i\}$ выбор оптимального набора $\mathbf{X}_i^\delta$ при фиксированном δ выполняется на кластере $\mathbf{X}_i^{\mathrm{train}}$ своих объектов из класса ω_i. Оптимизация набора эталонов в классе ω_i сводится к нахождению в последовательности (28) набора

$$\hat{\mathbf{X}}_i^\delta = \arg\min_{k=0}^{m_i-1} \varepsilon^{\mathrm{train}}(\hat{\mathbf{X}}_{ik}, \delta, \mathbf{W}_{ik}), \tag{29}$$

где $\varepsilon^{\mathrm{train}}(\hat{\mathbf{X}}_{ik}, \delta, \mathbf{W}_{ik})$ — доля ошибок вида (26) по всем объектам из кластеров $\mathbf{X}_0^{\mathrm{train}} \subset \subset \mathbf{X}^{\mathrm{train}}$ и $\mathbf{X}_i^{\mathrm{train}} \subset \mathbf{X}^{\mathrm{train}}$, реализуемая на наборе эталонов $\hat{\mathbf{X}}_{ik}$ с весами $\mathbf{W}_{ik}$ и пороговым параметром δ.

Оптимальный параметр δ^*, одинаковый для всех классов долей ошибок, реализуемых на наборах эталонов $\hat{\mathbf{X}}_i^\delta$, $i = 1, \ldots, c$, вида (29) и равен

$$\delta^* = \arg\min_\delta \sum_{i=1}^{c} \varepsilon^{\mathrm{train}}(\hat{\mathbf{X}}_i^\delta, \delta, \mathbf{W}_i). \tag{30}$$

Построение древовидно-структурированных наборов эталонов выполняется с помощью алгоритма a_{sel} на кластерах $\mathbf{X}_i^{\mathrm{train}} \subset \mathbf{X}^{\mathrm{train}}$, $i = 1, \ldots, c$. Вычисление используемых в (29) и (30) долей ошибок $\varepsilon^{\mathrm{train}}(\hat{\mathbf{X}}_{ik}, \delta, \mathbf{W}_{ik})$ и $\varepsilon^{\mathrm{train}}(\hat{\mathbf{X}}_i^\delta, \delta, \mathbf{W}_i)$ элементарных классификаторов $\{\omega_0, \omega_i\}$, $i = 1, \ldots, c$, выполняется по схеме «класс против всех»: кластер $\mathbf{X}_i^{\mathrm{train}}$ представляет объекты класса ω_i, а множество $\mathbf{X}^{\mathrm{train}} \setminus \mathbf{X}_i^{\mathrm{train}}$ — объекты класса ω_0. При любом фиксированном $\delta \in [0, 1]$ и значениях $i = 1, \ldots, c$ решение элементарного классификатора $\{\omega_0, \omega_i\}$ по объектам кластера $\mathbf{X}_i^{\mathrm{train}}$ принимается согласно правилу

$$i^*(\mathbf{x}) = [g_i^L(\mathbf{x}) \geqslant \delta \overline{g}_i^L] i\,.$$

Здесь разделяющая функция $g_i^L(\mathbf{x})$ вычисляется по модифицированной мере

$$\tilde{d}^L(\mathbf{x},\hat{\mathbf{x}}) = [\mathbf{x}=\hat{\mathbf{x}}] \operatorname*{mean}_{(\mathbf{x}',\mathbf{x}'')\in\mathbf{X}_i^{\text{train}}} d^L(\mathbf{x}',\mathbf{x}'') + [\mathbf{x}\neq\hat{\mathbf{x}}]d^L(\mathbf{x},\hat{\mathbf{x}}),$$

где $\operatorname*{mean}_{(\mathbf{x}',\mathbf{x}'')\in\mathbf{X}_i^{\text{train}}} d^L(\mathbf{x}',\mathbf{x}'')$ — среднее значение меры различия $d^L(\mathbf{x}',\mathbf{x}'')$ объектов кластера $\mathbf{X}_i^{\text{train}}$, а $[f]$ — индикатор f.

Результатом операций (29) и (30) являются оптимальные наборы эталонов $\{\hat{\mathbf{X}}_i^*\}_{i=1}^c$ с мощностями $\{\hat{m}_i^* = \|\hat{\mathbf{X}}_i^*\|\}_{i=1}^c$ и оптимальные пороги $\{\Delta_i^* = \delta^*\overline{g}_i^L\}_{i=1}^c$.

5 Тестирование классификатора

5.1 Состав и схема эксперимента

Разработанный классификатор протестирован на составном источнике полутоновых изображений (8 бит/пиксель) подписей (40 классов по 20 изображений) [12], жестов руки (25 классов по 40 изображений) [13] и лиц (25 классов по 40 изображений) [14]. Множество изображений $\mathbf{X}$ содержало 2800 объектов; подмножество Ω_c включало $c = 90$ семантически однородных классов; класс ω_0 эмулирован на объектах подмножества Ω_c; для представления объектов использованы деревья примитивов глубины $L = 10$. Результаты тестирования представлены ROC-кривыми в терминах зависимостей TPR от FPR, полученных при различных наборах эталонов по классам, и кривыми зависимости средней доли ошибок $((1-\text{TPR})+\text{FPR})/2$ от вычислительной сложности C_α решающего алгоритма a_α, построенными для классификаторов с различными наборами эталонов и оптимизированными порогами.

Реализованы две схемы скользящего контроля, каждая из которых базируется на многократном разбиении множества объектов источника $\mathbf{X}$ на обучающую $\mathbf{X}^{\text{train}}$ и тестовую $\mathbf{X}^{\text{test}}$ выборки. На каждой выборке $\mathbf{X}^{\text{train}}$ строится классификатор: согласно (29) отбираются наборы эталонов по классам $\hat{\mathbf{X}}_i^\delta$, $i = 1,\ldots,c$, при различных δ и согласно (30) выбирается оптимальный параметр δ^*. Построенному классификатору предъявляется выборка $\mathbf{X}^{\text{test}}$, на которой регистрируются ошибочные решения по своим или чужим объектам.

Первая схема скользящего контроля использует процедуру leave-one-out для регистрации ошибочных решений по своим объектам. В этой схеме выборки $\mathbf{X}^{\text{test}}$ образованы всевозможными одиночными объектами из подмножества классов Ω_c; классификаторы строятся на выборках $\mathbf{X}^{\text{train}} = \mathbf{X}\setminus\mathbf{X}^{\text{test}}$ и содержат наборы эталонов для c классов; предъявление этим классификаторам выборок $\mathbf{X}^{\text{test}}$ дает долю ошибок скользящего контроля $\varepsilon_{\text{loo}}^{\text{test}}(\hat{\mathbf{X}},\boldsymbol{\Delta},\mathbf{W}|\Omega_c)$, которая является оценкой величины $\hat{\varepsilon}(\hat{\mathbf{X}},\boldsymbol{\Delta},\mathbf{W}|\Omega_c)$, определенной в (24).

Вторая схема скользящего контроля использует процедуру leave-class-out для регистрации ошибочных решений по чужим объектам. В этой схеме выборки $\mathbf{X}^{\text{test}}$ образованы одиночными кластерами $\mathbf{X}_i$, $i = 1,\ldots,c$, эмулирующими объекты из класса ω_0; классификаторы строятся на выборках $\mathbf{X}^{\text{train}} = \mathbf{X}\setminus\mathbf{X}^{\text{test}}$ и содержат наборы эталонов для $c-1$ классов; предъявление классификаторам выборок $\mathbf{X}^{\text{test}}$ дает долю ошибок скользящего контроля $\varepsilon_{\text{lco}}^{\text{test}}(\hat{\mathbf{X}},\boldsymbol{\Delta},\mathbf{W}|\omega_0)$, которая является оценкой величины $\hat{\varepsilon}(\hat{\mathbf{X}},\boldsymbol{\Delta},\mathbf{W}|\omega_0)$, определенной в (**??**').

В рассмотренных схемах скользящего контроля построение классификатора на любой выборке $\mathbf{X}^{\text{train}}$ выполняется с применением алгоритма отбора наборов эталонов по классам a_{sel} и операций оптимизации параметров, заданных соотношениями (29) и (30).

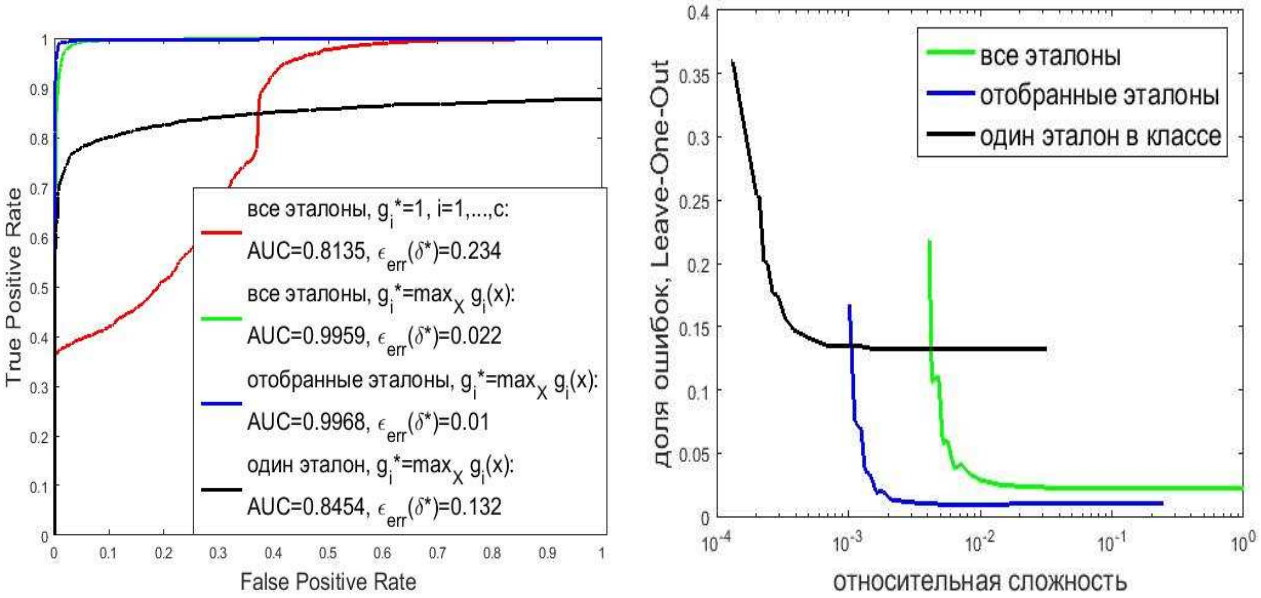
5.2 Результаты эксперимента

В ходе эксперимента получено параметрическое множество оценок скользящего контроля

$$\mathrm{TPR}_{\alpha=0}(\delta) = 1 - \varepsilon^{\mathrm{test}}_{\mathrm{loo},\alpha=0}\left(\hat{\mathbf{X}}^{\delta}, \boldsymbol{\Delta}^{\delta}, \mathbf{W}|\Omega_c\right); \quad \mathrm{FPR}_{\alpha=0}(\delta) = \varepsilon^{\mathrm{test}}_{\mathrm{lco},\alpha=0}\left(\hat{\mathbf{X}}^{\delta}, \boldsymbol{\Delta}^{\delta}, \mathbf{W}|\omega_0\right) \qquad (31)$$

с параметром $\delta \in [0,1]$, где $\hat{\mathbf{X}}^{\delta}$, $\boldsymbol{\Delta}^{\delta}$ — множества эталонов и порогов при фиксированном значении δ. Множество оценок (31) дает ROC-кривую в терминах зависимости TPR от FPR. Весовые коэффициенты эталонов по классам, образующие множество $\mathbf{W}$, принимали значения $\{0,1\}$: значение 1 для ближайшего эталона в каждом классе и значение 0 для всех других эталонов. Оценки (31) вычислены на основе решающего правила (5) с применением переборного алгоритма $a^{\mathrm{es}}_{\alpha=0}$ по всем разделяющим функциям наибольшего порядка $L = 10$. Множество порогов $\boldsymbol{\Delta}^{\delta} = \{\Delta_i = \delta\overline{g}^L_i\}^c_{i=1}$ строилось для двух случаев: с использованием значений $\{\overline{g}^L_i = 1\}^c_{i=1}$, которые порождают одинаковые пороги для всех классов, и для значений $\{\overline{g}^L_i = g^L_{i1}\}^c_{i=1}$, которые определяются первыми порядковыми статистиками в формуле (22) и приводят к адаптивным по классам порогам. Семейство ROC-кривых, полученных при различных схемах выбора эталонов по классам, дано на рис. 5, *а*. На графиках указаны площади AUC (Area Under Curve) под соответствующими кривыми.

«Наихудшую» ROC-кривую (красная кривая, AUC=0,8135) демонстрирует классификатор с наборами эталонов по классам, заданными всеми объектами обучающей выборки (наборами наибольшей мощности), и одинаковыми порогами по классам; «наилучшую» (синяя кривая, AUC=0,9968) — классификатор с наборами эталонов, отобранными с применением алгоритма a_{sel}, и адаптивными порогами. Для сравнения приведены две ROC-кривые для классификаторов с адаптивными порогами: с одним эталоном в каждом классе, отобранным алгоритмом a_{sel} (черная кривая, AUC=0,8454) и с наборами эталонов наибольшей мощности (зеленая кривая, AUC = 0,9959).



(*а*) $\alpha = 0,\ 0 \leqslant \delta \leqslant 1$ (*б*) $\alpha \geqslant 0,\ \delta = \delta^*$

Рис. 5 ROC-кривые (*а*) и функции «качество–сложность» (*б*)

Для рассмотренных классификаторов с адаптивными порогами $\boldsymbol{\Delta}^* = \{\Delta_i = \delta^* \overline{g}_i^L\}_{i=1}^c$ (при оптимальных значениях δ^* вида (30)) вычислены оценки долей ошибок скользящего контроля

$$\varepsilon_\alpha^{\text{test}}(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}) = \frac{1}{2}\left(\varepsilon_{\text{loo},\alpha}^{\text{test}}(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}|\Omega_c) + \varepsilon_{\text{lco},\alpha}^{\text{test}}(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}|\omega_0)\right)$$

с применением решающего алгоритма a_α с параметром $0 \leqslant \alpha \leqslant 5{,}5$. Здесь $\hat{\mathbf{X}}^*$ — множество эталонов, полученное либо на наборах (29) с оптимальным параметром (30), либо на одиночных эталонах по классам, отобранных алгоритмом a_{sel}, либо на наборах наибольшей мощности, заданных соответствующими кластерами обучающей выборки. Используя оценку (18), для классификаторов с указанными параметрами вычислены оценки вычислительной сложности $C_\alpha(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W})$ и доли сложности

$$\gamma_\alpha(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W}) = \frac{C_\alpha(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W})}{\max\limits_{\hat{\mathbf{X}}^*} C_{\alpha=0}(\hat{\mathbf{X}}^*, \boldsymbol{\Delta}^*, \mathbf{W})}.$$

Для трех классификаторов с «наилучшими» ROC-кривыми зависимости $\varepsilon_\alpha^{\text{test}}$ от γ_α даны на рис. 5, *б*. Полученные зависимости демонстрируют наименьшие доли ошибок $\varepsilon_{\alpha,\min}^{\text{test}}$, достигаемые на решающих алгоритмах $a_{\alpha>0}^{\text{gs}}$ и $a_{\alpha=0}^{\text{es}}$, и соответствующие этим ошибкам значения вычислительного выигрыша $1/\gamma_\alpha$ алгоритма $a_{\alpha>0}^{\text{gs}}$ относительно алгоритма $a_{\alpha=0}^{\text{es}}$. Для указанных классификаторов получены следующие характеристики: при наборах эталонов наибольшей мощности (зеленая кривая) наименьшая доля ошибок $\varepsilon_{\alpha,\min}^{\text{test}} = 0{,}022$, вычислительный выигрыш $1/\gamma_\alpha = 13{,}7$; при оптимальных наборах эталонов (синяя кривая) $\varepsilon_{\alpha,\min}^{\text{test}} = 0{,}010$, $1/\gamma_\alpha = 20{,}9$; при одном эталоне в каждом классе (черная кривая) $\varepsilon_{\alpha,\min}^{\text{test}} = 0{,}0132$, $1/\gamma_\alpha = 416{,}7$. Заметное увеличение достоверности распознавания и вычислительного выигрыша достигнуто на оптимальных наборах эталонов по сравнению с наборами эталонов наибольшей мощности. Использование одиночных эталонов по классам обеспечивает существенное увеличение вычислительного выигрыша по сравнению с оптимальными наборами, но приводит к снижению качества распознавания.

6 Заключение

В пространстве древовидных представлений образов с многоуровневым разрешением разработан многоклассовый метрический классификатор с решающим правилом голосования наборов эталонов по классам, которое допускает отказ от классификации. Введена однопараметрическая ROC-функция в терминах зависимости доли верных решений TPR по «своим» объектам, относящимся к положительным классам, от доли ложных решений FPR по «чужим» объектам, которые не относятся к положительным классам и допускают верное решение в виде отказа от классификации. Параметром ROC-функции является параметр пороговых значений, используемых в решающем правиле. Процедура обучения базируется на построении древовидно-структурированных наборов эталонов по классам и оптимизации таких наборов на основе минимизации долей ошибок соответствующих элементарных классификаторов типа «класс против всех». Обучение включает также оценивание порогового параметра на основе минимизации средней доли ошибок $((1-\text{TPR})+{}+\text{FPR})/2$ в многоклассовой схеме. Используя сеть эталонов с многоуровневым разрешением, предложен иерархический решающий алгоритм на основе параметрической стратегии сужения зоны поиска решения на последовательных уровнях разрешения. Выявлена область значений параметра решающего алгоритма, которая обеспечивает экспоненциально растущий вычислительный выигрыш по сравнению с алгоритмом полного перебора.

Эффективность предложенного метода распознавания образов продемонстрирована на множестве изображений лиц, жестов и подписей, содержащих объекты из 90 классов. В режиме скользящего контроля эмулированы классы «своих» и «чужих» объектов. Для классификаторов с различными наборами эталонов и различными стратегиями выбора порогов построены ROC-функции и вычислены характеристики AUC. Для наборов эталонов по классам, заданных одиночными объектами, всеми объектами обучающей выборки и оптимальными наборами, отобранными путем минимизации функционала ошибок, построены соотношения «качество–сложность» в терминах зависимости доли ошибок от вычислительной сложности решающего алгоритма. Наилучшее соотношение «качество–сложность» получено для классификатора с оптимальными наборами эталонов по классам: при наименьшей доле ошибок порядка 1% иерархический решающий алгоритм демонстрирует 20-кратное увеличение быстродействия по сравнению с переборным алгоритмом.

Дальнейшее развитие предложенной модели классификации предполагает разработку и исследование многоклассовых схем с использованием наборов элементарных классификаторов и, в частности, элементарных SVM-классификаторов (Support Vector Machine), обучаемых по схеме «класс против всех». Предполагается также исследовать схемы комплексирования на основе голосования решений классификаторов по отдельным источникам (Majority Voting) и на основе обобщенной меры на ансамбле источников (General Measure).

Литература

[1] *Rosenfeld A.* Quadtrees and pyramids: Hierarchical representation of images. — University of Maryland, Computer Science, 1982. 14 p.

[2] *Samet H.* The quadtree and related hierarchical data structures // ACM Comput. Surv., 1984. Vol. 16. No. 2. P. 187–260. doi: 10.1145/356924.356930.

[3] *Elfiky N. M., Khan F. S., van de Weijer J., Gonzalez J.* Discriminative compact pyramids for object and scene ecognition // Pattern Recogn., 2012. Vol. 45. No. 4. P. 1627–1636. http://dx.doi.org/10.1016/j.patcog.2011.09.020.

[4] *Torsello A., Jiang X., Ferrer M.* Editorial for the special issue on graph-based representations in pattern recognition // Pattern Recogn. Lett., 2012. Vol. 33. No. 15. P. 1957. http://dx.doi.org/10.1016/j.patrec.2012.08.016.

[5] *Mestetskiy L., Semenov A.* Binary image skeleton — continuous approach // 3rd Conference (International) on Computer Vision Theory and Applications Proceedings. — INSTI CC, 2008. Vol. 1. P. 251–258.

[6] *Ganebnykh S. N., Lange M. M., Stepanov D. Y.* Metric classifier using multilevel network of templates // Pattern Recogn. Image Anal., 2012. Vol. 22. No. 2. P. 265–277. doi: 10.1134/S1054661812020034.

[7] *Pelillo M., Hidovic-Rowe D., Torsello A.* Polynomial-time metrics for attributed trees // IEEE Trans. Pattern Anal. Mach. Intell., 2005. Vol. 27. No. 7. P. 1087–1099. doi: 10.1109/TPAMI.2005.146.

[8] *Tax D. M. J., Duin R. P. W.* Growing a multi-class classifier with a reject option // Pattern Recogn. Lett., 2008. Vol. 29. No. . P. 1565–1570. http://dx.doi.org/10.1016/j.patrec.2008.03.010.

[9] *Lange M. M., Ganebnykh S. N.* An efficiency of hierarchical classification in terms of fidelity-complexity ratio // Machine Learning Data Anal., 2014. Vol. 1. No. 8. P. 1126–1136. http://jmlda.org/papers/doc/2014/no8/Lange2014Efficiency.pdf.

[10] *Theodoridis S., Koutroumbas K.* Pattern recognition. — 4th ed. — Elsevier, 2009. 978 p. http://www.sciencedirect.com/science/book/9781597492720.

[11] *Duda R., Hart P., Stork D.* Pattern classification. — 2nd ed. — Wiley, 2001. 680 p. http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471056693.html.

[12] Database of signatures: First international signature verification competition. 2004. www.cse.ust.hk/svc2004/download.html.

[13] *Moeslund T.* Gesture recognition database. 2002. www-prima.inrialpes.fr/FGnet/data/12-MoeslundGesture/database.html.

[14] *Stepanov D. Yu.* Face database. — MIREA, 2013. http://stepanovd.com/news_2015_10_facedben.html?lang=EN.

Поступила в редакцию 29.07.2016

Multiclass pattern recognition in a space of multiresolution representations*

M. M. Lange, S. N. Ganebnykh, and A. M. Lange

lange_mm@ccas.ru, sng@ccas.ru

Federal Research Center "Computer Science and Control" of RAS, 44/2 Vavilova st., Moscow, Russia

For a multiclass source of patterns given by images, a metric classification scheme in a space of tree-structured pattern representations is suggested. At the successive resolution levels, in a set of the pattern representations, both a family of dissimilarity measures at the successive levels and discriminant functions (class likelihoods) by the appropriate measures are defined. A decision of the multiclass classifier is made by voting the values of the discriminant functions. Also, a reject is available. A learning procedure that includes a selection of the template patterns in classes as well as an optimization of the classifier parameters is developed. A parametric decision algorithm that includes both hierarchical and exhaustive search strategies for the decision in the multilevel network of the templates is constructed. Analytical estimation of a computational complexity of the algorithm is obtained. For a composite source of the patterns given by signatures, hand gestures, and faces, an efficiency of the classifiers with different parameters is shown by the appropriate ROC curves as well as by empirical dependencies of the error rate on the computational complexity of the decision algorithm.

Keywords: *tree-structured representation; discriminant function; multiclass pattern recognition; multilevel network of templates; ROC-curve; hierarchical search; computational complexity*

DOI: 10.21469/22233792.2.1.06

References

[1] Rosenfeld, A. 1982. *Quadtrees and pyramids: Hierarchical representation of images.* University of Maryland, Computer Science. 14 p.

[2] Samet, H. 1984. The quadtree and related hierarchical data structures. *ACM Comput. Surv.* 16(2):187–260. doi: 10.1145/356924.356930.

[3] Elfiky, N. M., F. S. Khan, J. van de Weijer, and J. Gonzalez. 2012. Discriminative compact pyramids for object and scene ecognition. *Pattern Recogn.* 45(4):1627–1636. Available at: http://dx.doi.org/10.1016/j.patcog.2011.09.020 (accessed October 15, 2016).

*The research was partially supported by the Russian Foundation for Basic Research (grants 15-01-04671, 15-07-09324, and 15-07-07516).

[4] Torsello, A., X. Jiang, and M. Ferrer. 2012. Editorial for the special issue on graph-based epresentations in pattern recognition. *Pattern Recogn. Lett.* 33(15):1957. Available at: http://dx.doi.org/10.1016/j.patrec.2012.08.016 (accessed October 15, 2016).

[5] Mestetskiy, L., and A. Semenov. 2008. Binary image skeleton — continuous approach. *3rd Conference (International) on Computer Vision Theory and Applications Proceedings*. INSTICC. 1:251–258.

[6] Ganebnykh, S. N., M. M. Lange, and D. Y. Stepanov. 2012. Metric classifier using multilevel network of templates. *Pattern Recogn. Image Anal.* 22(2):265–277. doi: 10.1134/S1054661812020034.

[7] Pelillo, M., D. H. Hidovic-Rowe, and A. Torsello. 2005. Polynomial-time metrics for attributed trees. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(7):1087–1099. doi: 10.1109/TPAMI.2005.146.

[8] Tax, D. M. J., and R. P. W. Duin. 2008. Growing a multi-class classifier with a reject option. *Pattern Recogn. Lett.* 29(10):1565–1570. Available at: http://dx.doi.org/10.1016/j.patrec.2008.03.010 (accessed October 15, 2016).

[9] Lange, M. M., and S. N. Ganebnykh. 2014. An efficiency of hierarchical classification in terms of fidelity-complexity ratio. *Machine Learning Data Anal.* 1(8):1126–1136. Available at: http://jmlda.org/papers/doc/2014/no8/Lange2014Efficiency.pdf (accessed October 15, 2016).

[10] Theodoridis, S., and K. Koutroumbas. 2009. *Pattern recognition*. 4th ed. Elsevier. Available at: http://www.sciencedirect.com/science/book/9781597492720 (accessed October 15, 2016).

[11] Duda, R., P. Hart, and D. Stork. 2001. *Pattern classification*. 2nd ed. Wiley. 680 p. Available at: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471056693.html (accessed October 15, 2016).

[12] Database of signatures: First international signature verification competition. 2004. Available at: www.cse.ust.hk/svc2004/download.html (accessed June 14, 2016).

[13] Moeslund, T. 2002. Gesture recognition database. Available at: www-prima.inrialpes.fr/FGnet/data/12-MoeslundGesture/database.html (accessed June 14, 2016).

[14] Stepanov, D. Yu. 2013. *Face database*. MIREA. Available at: http://stepanovd.com/news_2015_10_facedben.html?lang=EN (accessed June 14, 2016).

Received July 29, 2016

Моделирование и анализ вариаций космических лучей в периоды повышенной солнечной и геомагнитной активности*

О. В. Мандрикова, Т. Л. Заляев, Ю. А. Полозов, И. С. Соловьев
oksanam1@mail.ru, tim.aka.geralt@mail.ru, up_agent@mail.ru, kamigsol@yandex.ru
Институт космофизических исследований и распространения радиоволн ДВО РАН
Россия, Камчатский край, Елизовский район, п. Паратунка, ул. Мирная, 7

Описан способ анализа вариаций космических лучей (КЛ), позволяющий выделять аномальные изменения и получать количественные оценки о моментах их возникновения, временной длительности и интенсивности. Способ включает декомпозиции данных нейтронных мониторов на основе вейвлет-преобразования и их аппроксимацию на основе адаптивных нейронных сетей переменной структуры. На основе применения способа выполнен анализ вариаций КЛ в периоды повышенной солнечной и геомагнитной активности и выделены аномальные изменения, возникающие за несколько часов до геомагнитных бурь, во время бурь происходили длительные и глубокие Форбуш-понижения (анализировались данные нейтронных мониторов станций Апатиты и Мыс Шмидта). Совместно с данными КЛ анализировались вариации геомагнитного поля и ионосферные параметры, обработка которых выполнялась на основе методов, предложенных авторами.

Ключевые слова: *космические лучи; магнитная буря; ионосферные параметры; анализ данных; вейвлет-преобразование; нейронные сети*

DOI: 10.21469/22233792.2.1.07

1 Введение

Наблюдаемые на поверхности Земли вариации КЛ являются интегральным результатом различных солнечных, гелиосферных и атмосферных явлений и имеют сложную структуру [1, 2]. Наиболее существенные изменения в параметрах КЛ вызывают выбросы коронарной массы и следующие за ними изменения в параметрах межпланетного поля и солнечного ветра [3, 4]. Их интенсивность зависит от метеорологических параметров, электромагнитной обстановки в Солнечной системе и физических условий в Галактике [1]. Также в вариациях КЛ находит отражение 11-летний цикл и 27-дневный солнечный период вращения [2] и присутствует суточный ход, обусловленный асимметрией формы магнитосферы, которая изменяется во времени при изменении параметров солнечного ветра [5].

Наблюдения КЛ используются при проведении ряда фундаментальных и прикладных исследований, связанных с мониторингом и прогнозом космической погоды [1, 3]. Анализ КЛ позволяет получать ценную информацию о состоянии околоземного космического пространства в периоды экстремальных солнечных событий. Весьма актуальной задачей является выделение аномальных изменений в динамике КЛ накануне сильных геомагнитных бурь [3, 6, 7]. Данные КЛ имеют сложную структуру, и традиционные методы обработки статистических данных не являются достаточно эффективными для их исследования [4, 5, 8, 9] Для изучения вариаций КЛ в настоящее время получают развитие методы адаптивной аппроксимации [4], вейвлет-преобразование [4, 9–14] и нейронные сети

*Работа выполнена при финансовой поддержке гранта Российского научного фонда №14-11-00194.

(НС) [8, 14–16]. Использование НС при первичной обработке данных нейтронных мониторов позволило повысить эффективность процедуры подавления шума, по сравнению с медианными методами [8]. На основе совмещения вейвлет-преобразования с методом разложения на эмпирические моды в долгосрочных временны́х изменениях хода КЛ выделены доминирующие временны́е масштабы (периоды 11 лет, 22 года, 6 лет и двухлетние колебания) и определена их физическая природа [4]. Исследования данной работы основаны на совместном применении методов вейвлет-преобразования и адаптивных НС переменной структуры. Вейвлет-преобразование позволяет выполнять детальный анализ локальных структур данных [17, 18] и является эффективным средством изучения сложных нестационарных процессов [10–13, 19–21]. Преимущество нейросетевого представления аппроксимируемой функции заключается в большой гибкости базовых функций и их способности к адаптации [22, 23]. Совместное применение кратномасштабного вейвлет-преобразования с НС, впервые предложенное в работе [14] для анализа КЛ, показало эффективность данного подхода в задачах изучения их структуры и выделения аномальных изменений в периоды повышенной солнечной и геомагнитной активности. Данная статья является продолжением этой работы. Для получения более детальной и достоверной информации об изменениях в параметрах КЛ совместно с разработанными решениями в статье предложено использовать непрерывное вейвлет-преобразование и пороговые функции.

2 Описание способа

2.1 Моделирование вариаций космических лучей на основе совмещения кратномасштабного вейвлет-преобразования и нейронных сетей

1. На основе кратномасштабного анализа (КМА) вейвлет-преобразования до уровня разложения получаем представление вариации КЛ в виде [17, 24]:

$$f_0(t) = \sum_{j=-1}^{-m} f^d[2^j t] + f^a[2^{-m} t]\,.$$

Здесь $f^d[2^j t] = \sum_n d_{j,n}\Psi_{j,n}(t)$, где $d_{j,n} = \langle f, \Psi_{j,n}\rangle$ — разномасштабные детализирующие компоненты, $f^d[2^j t] \in W_j$, $W_j = \mathrm{clos}_{L^2(R)}(2^{j/2}\Psi(2^j t - n) : n \in Z)$, Ψ — базисный вейвлет, j — разрешение; $f^a[2^{-m}t] = \sum_k c_{-m,n}\theta_{-m,n}(t)$, где $c_{-m,n} = \langle f, \theta_{-m,n}\rangle$ — сглаженная составляющая, $f^a[2^{-m}t] \in V_{-m}$, $V_j = \mathrm{clos}_{L^2(R)}(2^{j/2}\theta(2^j t - n) : n \in Z)$, θ — сглаживающая скэйлинг-функция. Представление вариации КЛ в вейвлет-пространстве на основе КМА до 6-го масштабного уровня разложения показано на рис. 1.

2. Используя обратное вейвлет-преобразование, восстанавливаем исходное разрешение компонент:

$$f_0^{a,(-m)}(t) = \sum_n c_{0,n}^{(-m)}\theta_{0,n}(t),\ f_0^{d,j}(t) = \sum_n d_{0,n}^j \Psi_{0,n}(t),$$

верхние индексы $(-m)$ и j соответствуют уровню разложения и разрешению компонент до выполнения операции обратного вейвлет-преобразования. Получаем вариацию КЛ в виде:

$$f_0(t) = f_0^{a,(-m)}(t) + \sum_{j=-1}^{-m} f_0^{d,j}(t). \qquad (1)$$

В соответствии с представлением (1) вариация КЛ включает сглаженную составляющую $f_0^{a,(-m)}(t)$, характеризующую уровень КЛ, и разномасштабные детализирующие компоненты $f_0^{d,j}(t)$, характеризующие локальные вариации относительно уровня.

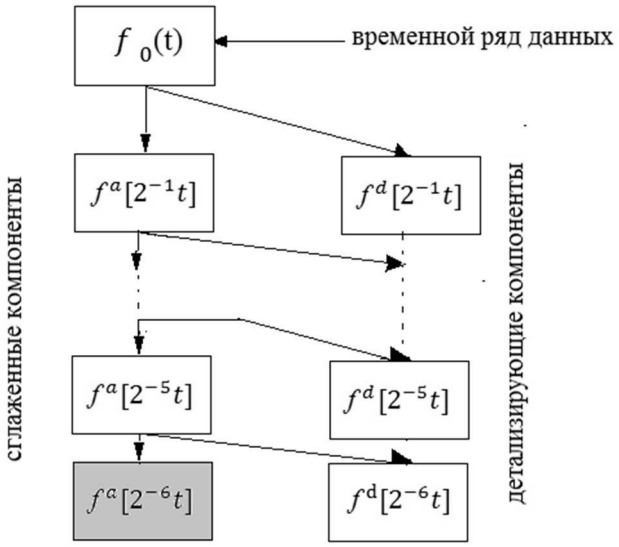


Рис. 1 Схема разложения данных КЛ на основе КМА

3. Для сглаженной составляющей КЛ на основе НС переменной структуры строим отображение (задача статистической экстраполяции, [9])

$$y\colon f_0^{a,(-m)} \to f_0^{*a,(-m)}, \tag{2}$$

где $f_0^{a,(-m)}$ — вход НС; $f_0^{*a,(-m)}$ — выход НС. При подаче на вход обученной НС значений функции $f^{a,(-m)}$ из интервала $(l-Q+1, l)$ сеть вычисляет упрежденные ее значения на временно́м интервале $(l+1, l+I)$, где l — текущий дискретный момент времени; I — длина интервала упреждения. Ошибка НС определяется как разность между желаемым $f_0^{*a,(-m)}$ и действительным $\hat{f}_0^{*a,(-m)}$ выходными значениями функции:

$$e(t) = \hat{f}_0^{*a,(-m)}(t) - f_0^{*a,(-m)}(t).$$

Алгоритмы выбора уровня разложения КМА m (см. п. 1) и построения нейросетевой схемы представлены в работах [9, 14]. При выполнении данной работы аппроксимирующие нейросетевые схемы, выполняющие отображение (2), строились отдельно для каждой станции регистрации данных КЛ. Учитывая долгосрочные временны́е изменения хода КЛ (периоды 11 лет, 22 года, 6 лет и двухлетние колебания), данные за разные годы моделировались отдельно. Поскольку динамика КЛ существенно зависит от электромагнитной обстановки в Солнечной системе и в периоды аномальных изменений находит отражение в геомагнитном поле [25], с целью экстраполяции характерного (фонового) хода КЛ в оценках использовались данные за временны́е интервалы относительно спокойного геомагнитного поля (интервалы, в которые суммарный за сутки K-индекс не превышает значения 18). При выполнении КМА использовались вейвлеты семейства Койфлеты порядка 3, которые были определены путем минимизации погрешности аппроксимации. Были построены нейросетевые схемы для выделенной сглаженной составляющей масштабного уровня $m = -6$ (данная составляющая показана на рис. 1 серым цветом), их архитектура представлена на рис. 2. Полученные нейросетевые схемы выполняют преобразование данных вида:

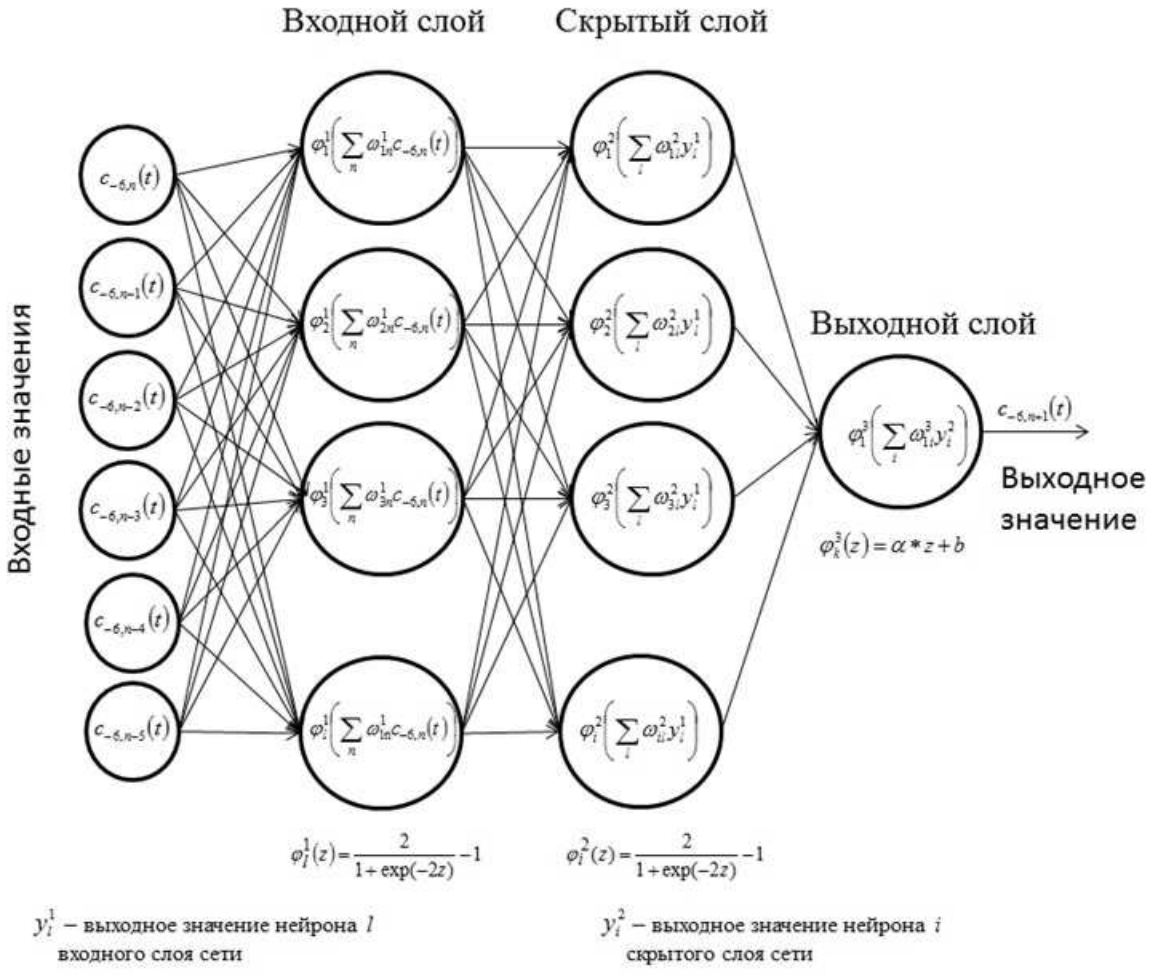


Рис. 2 Архитектура построенных нейросетевых схем

$$c_{-6,n+1}(t) = \varphi_k^3\left(\sum_i \omega_{ki}^3 \varphi_i^2\left(\sum_l \omega_{il}^2 \varphi_l^1\left(\sum_n \omega_{ln}^1 c_{-6,n}(t)\right)\right)\right),$$

где ω_{ln}^1 — весовые коэффициенты нейрона l входного слоя сети; ω_{il}^2 — весовые коэффициенты нейрона i скрытого слоя сети; ω_{ki}^3 — весовые коэффициенты нейрона k выходного слоя; $\varphi_l^1(z) = \varphi_i^2(z) = 2/(1+\exp(-2z))) - 1$; $\varphi_k^3(z) = az + b$.

Применение нейросетевой схемы позволяет воспроизводить характерные вариации сглаженной составляющей КЛ $f_0^{a(-m)}$ (аппроксимирует характерный уровень вариаций КЛ). В период аномальных изменений временно́го хода КЛ абсолютные значения ошибок обученной НС возрастут, поэтому операция их выделения может быть основана, например, на проверке следующего условия:

$$|e(t)| > T,$$

где T — пороговое значение, определяющее наличие аномалии.

На рис. 3 в качестве примера показаны результаты работы построенной нейросетевой схемы для станции Апатиты за 2013 г. Анализ рис. 3 показывает хорошие аппроксими-

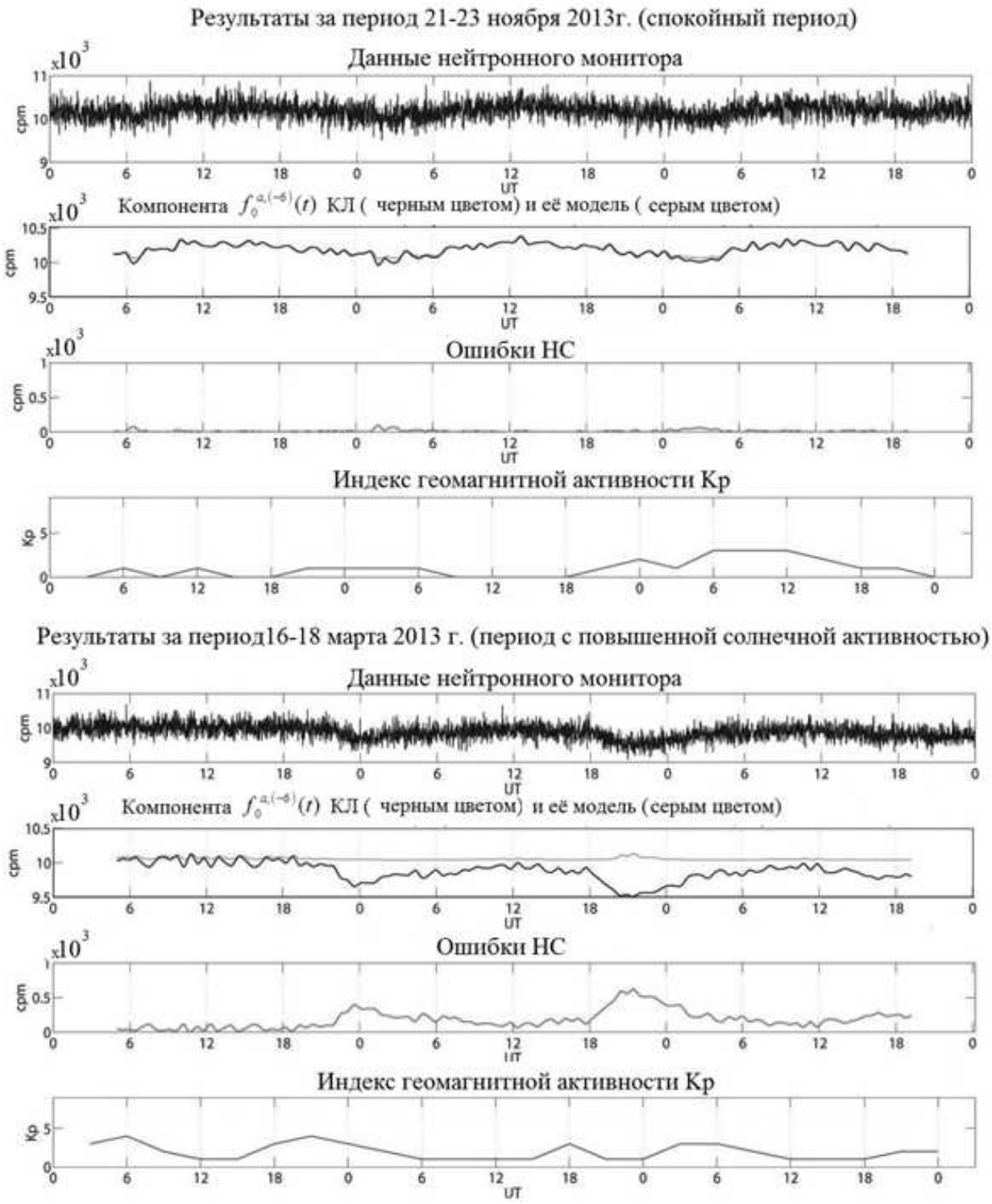


Рис. 3 Результаты работы нейросетевой схемы для станции Апатиты за 2013 г.

рующие свойства НС, ее ошибки в период спокойного геомагнитного поля (21–23 ноября 2013 г.) не превышают значения $0{,}1 \cdot 10^3$. В период возрастания геомагнитной активности (16–18 марта 2013 г.) наблюдается изменение хода КЛ, и ошибки сети возрастают.

2.2 Выделение локальных разномасштабных аномалий в вариациях космических лучей и оценка их параметров на основе непрерывного вейвлет-преобразования

Аномалии в регистрируемых вариациях КЛ могут содержать трендовые изменения, возникающие в периоды длительных Форбуш-эффектов, а также могут содержать локальные кратковременные особенности, характерные для локальных повышений и понижений КЛ. Описанная выше нейросетевая схема позволяет выделять трендовые изменения вариаций КЛ (изменения уровня КЛ). Выделение локальных аномалий может быть выполнено на основе более детального непрерывного вейвлет-преобразования, которое определяется формулой [17, 24]:

$$W_\Psi f_{b,a} := |a|^{-1/2} \int\limits_{-\infty}^{\infty} f(t)\Psi\left(\frac{t-b}{a}\right) dt, \tag{3}$$

где $f \in L^2(R)$; $a, b \in R$, $a \neq 0$, параметр a характеризует масштаб, b — время; Ψ — базисный вейвлет.

При уменьшении масштаба a коэффициенты $W_\Psi f_{b,a}$ характеризуют свойства функции f в окрестности b. На малых масштабах a абсолютные значения коэффициентов $|W_\Psi f_{b,a}|$ являются малыми за исключением окрестностей, содержащих локальные особенности функции f [18, 26]:

$$|W_\Psi f_{b,a}| \leqslant A a^{\alpha+1/2}\,, \tag{4}$$

где A — некоторое положительное число; α — показатель Липшица функции f в окрестности b.

Как следует из соотношения (4), постепенное уменьшение масштаба a позволяет фокусироваться на локальных свойствах сложной функции и детально исследовать ее структуру. Основываясь на этом свойстве вейвлет-преобразования, для выделения локальных аномалий во временно́м ходе КЛ в работе использовалась пороговая функция вида:

$$P_{T_a}(W_\Psi f_{b,a}) = \begin{cases} W_\Psi f_{b,a}, & \text{если } (W_\Psi f_{b,a} - W_\Psi f_{b,a}^{\mathrm{med},l}) \geqslant T_a\,; \\ 0, & \text{если } |W_\Psi f_{b,a} - W_\Psi f_{b,a}^{\mathrm{med},l}| < T_a\,; \\ -W_\Psi f_{b,a}, & \text{если } (W_\Psi f_{b,a} - W_\Psi f_{b,a}^{\mathrm{med},l}) < -T_a\,. \end{cases} \tag{5}$$

Здесь $W_\Psi f_{b,a}^{\mathrm{med},l}$ — медианное значение, рассчитанное в скользящем временно́м окне длины l; $T_a = U\mathrm{St}_a^l$ — пороговая функция, где $\mathrm{St}_a^l = \sqrt{(l-1)^{-1}\sum_{k=1}^{l}(W_\Psi f_{b,a} - \overline{W_\Psi f_{b,a}})}$ — стандартное отклонение, рассчитанное в скользящем временном окне длины l; $\overline{W_\Psi f_{b,a}}$ — среднее значение; U — пороговый коэффициент.

Длина скользящего временно́го окна $l = 1440$ отсчетов, что соответствует одним суткам (определена с учетом суточного хода КЛ).

В качестве базисного вейвлета использовались Койфлеты порядка 1. Выбор базисного вейвлета основывался на критерии минимизации погрешности вычислений [18]: в словаре ортонормированных базисов $D = \cup_{\lambda\in\Lambda} B^\lambda$ базис $B^\alpha = \{q_z^\beta\}_{1\leqslant z\leqslant N}$ лучше, чем базис $B^\gamma = \{q_z^\gamma\}_{1\leqslant z\leqslant N}$ при аппроксимации функции f, если он дает меньшую погрешность при одинаковом числе аппроксимирующих слагаемых, т. е. при всех $Z \geqslant 1$

$$\varepsilon^\beta[Z] \leqslant \varepsilon^\gamma[Z]\,,$$

где $\varepsilon[Z]$ — погрешность аппроксимации, которая определяется как

$$\varepsilon^{\lambda}[Z] = \sum_{Z \notin I_z^{\lambda}} |\langle f, q_z^{\lambda} \rangle|^2 = \|f\|^2 - \sum_{z \in I_z^{\lambda}} |\langle f, q_z^{\lambda} \rangle|^2$$

(I_z — множество индексов мощности Z).

В силу случайной природы данных использование любого порога T_a, определяющего наличие либо отсутствие аномалии, неминуемо связано с возможностью ошибочных решений. В работе в качестве критерия выбора порога использовался критерий наименьшей частоты ошибок (оценивался и минимизировался апостериорный риск [27]), который при располагаемых априорных данных представляет наиболее полную о них информацию. При оценке апостериорного риска для определения состояния межпланетной среды и околоземного пространства (характеризующих динамику КЛ) использовались параметры солнечного ветра, данные B_z компоненты межпланетного магнитного поля (получены на основе проекта ACE [http://www.srl.caltech.edu/ACE/ASC/]) и индекс геомагнитной активности K. Результаты оценок показали, что наименьшую погрешность обеспечивает пороговый коэффициент $U = 2{,}5$.

Применение операции (5) позволяет фиксировать периоды аномальных повышений и аномальных понижений КЛ. При оценке аномального периода необходимо учитывать носитель базисного вейвлета Ψ. Если базисный вейвлет Ψ имеет компактный носитель, равный $[-\Omega, \Omega]$, то множество пар точек (b, a) таких, что точка ξ содержится в носителе $\Psi_{b,a}$, определяют конус влияния точки ξ [18]. Так как носитель $\Psi_{b,a}$ на масштабе a равен $[b - \Omega a, b + \Omega a]$, то конус влияния точки ξ на масштабе a определяется неравенством

$$|b - \xi| \leqslant \Omega a.$$

Для оценки интенсивности аномалии в момент времени $t = b$ использовалась величина

$$Y_b = \sum_a P_{T_a}(W_\Psi f_{b,a}), \tag{6}$$

которая в случае локального повышения КЛ будет положительной, а в случае локального понижения КЛ — отрицательной.

Для детального анализа ионосферных параметров использовались вычислительные решения, основанные на непрерывном вейвлет-преобразовании (см. (3)) [28]:

$$P_{T_a}(W_\Psi f_{b,a}) = \begin{cases} W_\Psi f_{b,a}, & \text{если } |W_\Psi f_{b,a} - W_\Psi f_{b,a}^{\mathrm{med}}| \geqslant T_a\,; \\ 0, & \text{если } |W_\Psi f_{b,a} - W_\Psi f_{b,a}^{\mathrm{med}}| < T_a\,. \end{cases}$$

Здесь порог $T_a = U\mathrm{St}_a$ определяет наличие аномалии на масштабе a вблизи точки ξ, содержащейся в носителе $\Psi_{b,a}$; U — коэффициент порога, $\mathrm{St}_a = \sqrt{(\Phi - 1)^{-1} \sum_{u=1}^{\Phi} (W_\Psi f_{b,a} - \overline{W_\Psi f_{b,a}})^2}$; $\overline{W_\Psi f_{b,a}}$ и $W_\Psi f_{b,a}^{\mathrm{med}}$ — среднее значение и медиана, определяемые в скользящем временно́м окне длины Φ.

Интенсивность аномалии в момент времени $t = b$ оценивалась в работе как

$$I_b = \sum_a \frac{|P_{T_a}(W_\Psi f_{b,a})|}{\|W_\Psi f_{b,a}\|_2},$$

где норма $\|W_\Psi f_{b,a}\|_2 = \sqrt{\sum_{N_a} (P_{T_a}(W_\Psi f_{b,a}))^2}$, N_a — длина ряда на масштабе a.

3 Результаты анализа данных

В обработке использовались минутные данные нейтронных мониторов станций Мыс Шмидта (Россия) и Апатиты (Россия) и ионосферные данные станций Ленинград (Россия) и King Solmon (США). Для оценки состояния геомагнитного поля использовались данные магнитометров станций, расположенных вблизи анализируемых районов: Barrow (Аляска) и Furstenfeldbruck (Германия).

Анализируемый период 19.10.2003–26.10.2003 (рис. 4 и 5) содержит несколько солнечных событий. Наиболее сильная магнитная буря класса G3 (Kp индекс 7) произошла 24 октября 2003 г. Данное событие вызвал приход к Земле облака CME от вспышки класса X1, произошедшей 22 октября 2003 г. (http://www.nws.noaa.gov/os/assessments/pdfs/SWstorms_assessment.pdf). Результаты обработки данных КЛ демонстрируют общий характер их динамики на анализируемых станциях (см. рис. 4 и 5). Моделирование данных на основе НС (см. рис. 4, d, 4, e, 5 d и 5, e) показывает незначительное понижение их уровня 22 октября (короткий Форбуш-эффект) и более существенное и длительное понижение, возникшее к концу суток 24 октября (длительный Форбуш-эффект). В моменты понижений в данных КЛ возникают локальные особенности, которые фиксируются на основе применения пороговых функций (соотношения (5) и (6), показаны на рис. 4, b, 4, c, 5, b и 5, c черным цветом). Накануне Форбуш-эффектов, в начале суток 21 октября и в начале суток 24 октября, наблюдаются локальные возрастания КЛ (соотношения (5) и (6), показаны на рис. 4, b, 4, c, 5, b и 5, c серым цветом). Аномальное возрастание 21 октября происходило в период повышенной геомагнитной активности, имело максимальную интенсивность примерно в 20:00 UT и характеризовалось плавным нарастанием интенсивности и резким смещением спектра в область высоких частот, которое, вероятно, связано с ускорением КЛ по мере приближения межпланетного возмущения. Аномальное возрастание 24 октября возникло за несколько часов до начала магнитной бури, наибольшая его интенсивность на всех анализируемых станциях наблюдается за 3–4 ч до прихода ударной волны (SSC). Анализ режима ионосферы (см. рис. 4, $k-4$, m и 5, $k-5$, m) показывает, что накануне бури 24 октября 2003 г. на анализируемых станциях происходили колебания электронной плотности ионосферы (на рис. 4, l и 5, l показаны серым цветом — повышение концентрации относительно фона, черным цветом — понижение концентрации) за несколько часов до начала бури наблюдается аномальное повышение фона. В период бури произошло существенное понижение электронной концентрации и возникла отрицательная фаза ионосферной бури, которая наибольшей интенсивности достигла на станции Ленинград 25 октября в 19:00 UT, на станции King Solmon — 26 октября в 11:00 UT. Сопоставление результатов обработки с данными межпланетной среды показывает в выделенные аномальные периоды наличие возмущений в B_z компоненте межпланетного магнитного поля.

4 Выводы

Выполненный анализ вариаций КЛ показал общий характер их поведения в периоды повышенной солнечной активности и магнитных бурь. Во время магнитных бурь уровень КЛ существенно понижался и возникали глубокие и длительные Форбуш-эффекты. Представляют интерес выделенные аномальные предповышения КЛ, возникающие за несколько часов до начала магнитных бурь. Подобные аномальные изменения отмечены авторами работ [6, 7, 29]. В этих работах показано, что возникающие в последние часы перед ударной волной, а иногда задолго до ее прихода, аномальные предповышения КЛ (а в некоторых случаях предпонижения КЛ) могут являться предвестниками сильных геомагнитных воз-

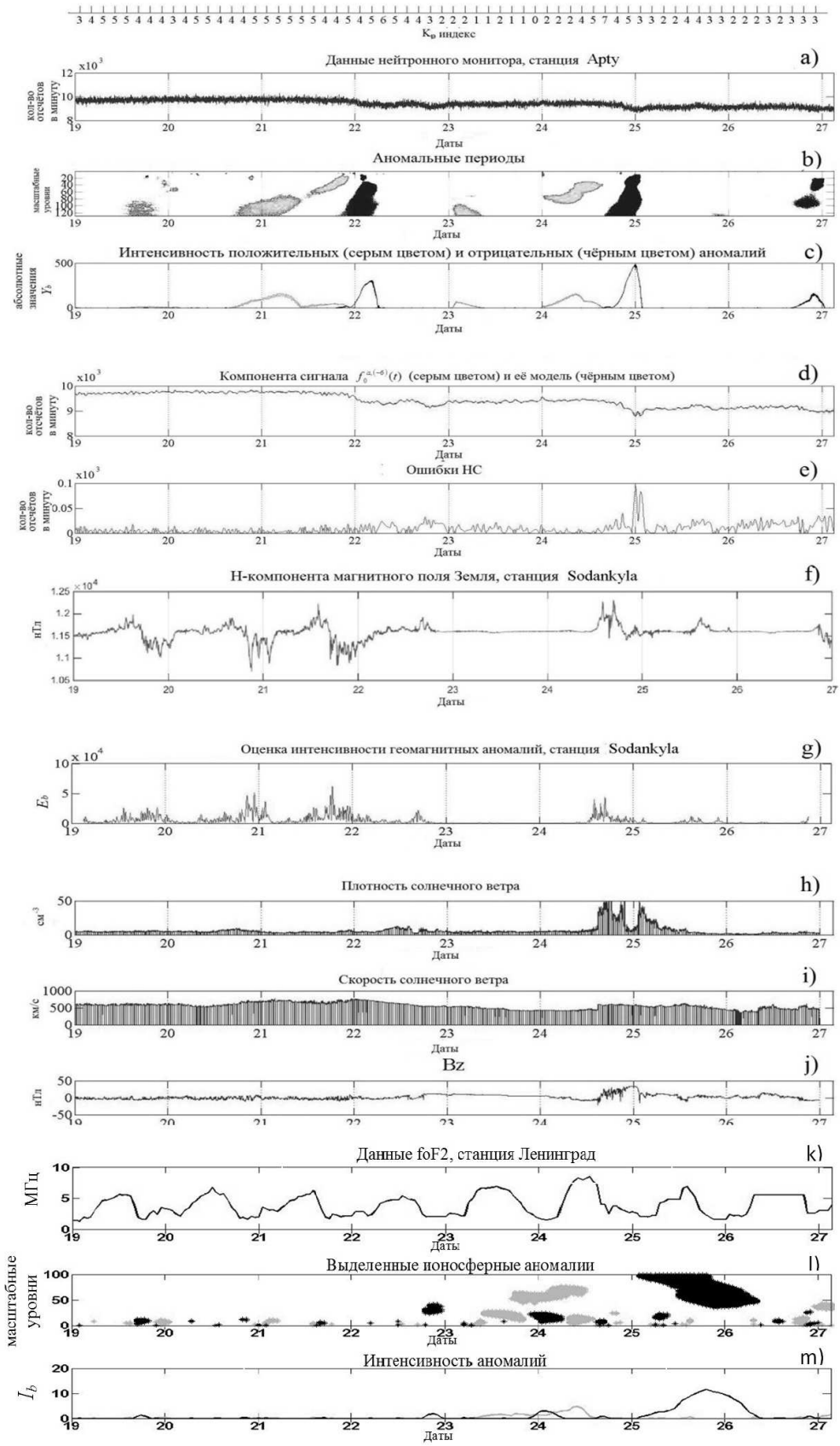


Рис. 4 Результаты обработки данных за период 19.10.2003—27.10.2003, станция Апатиты

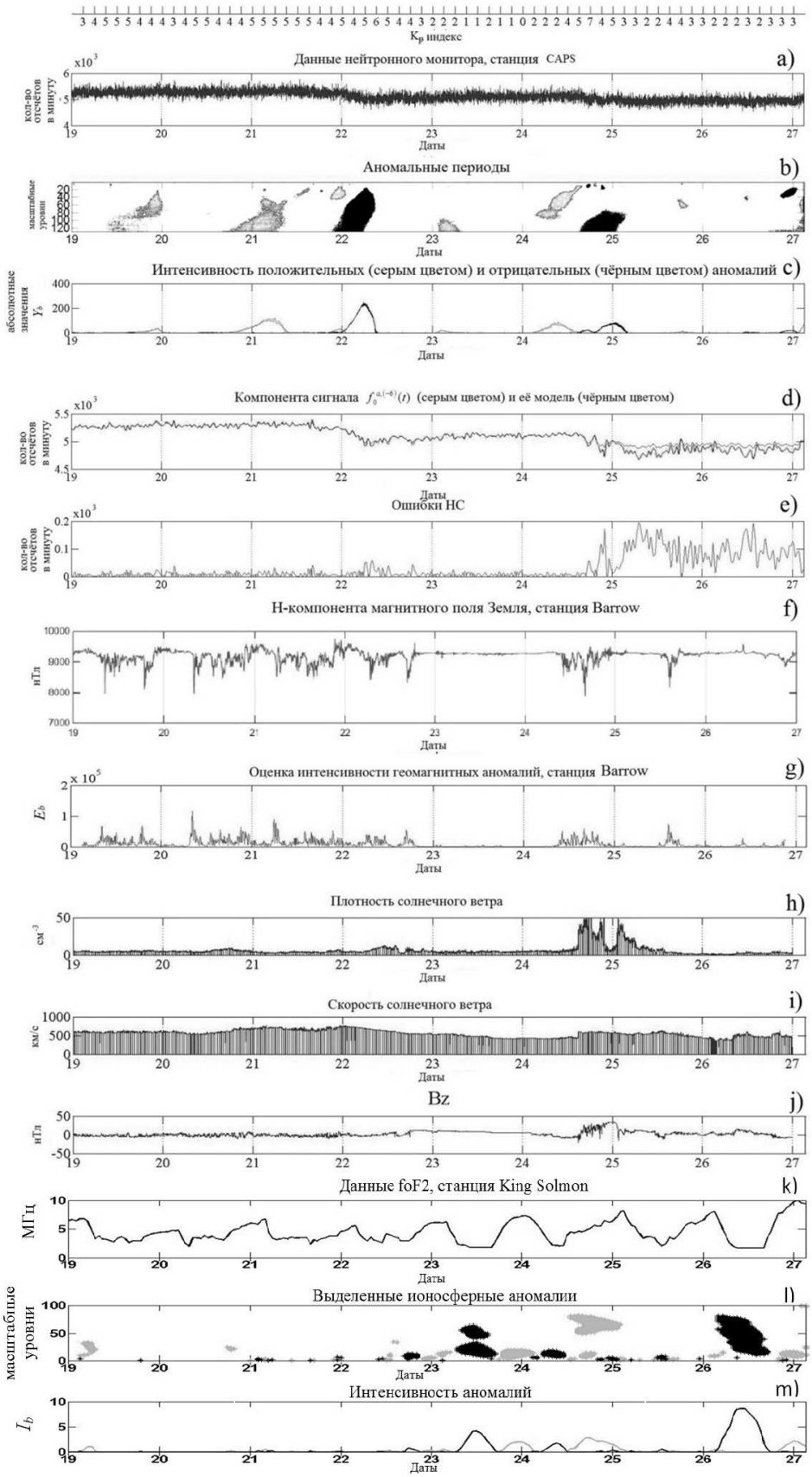


Рис. 5 Результаты обработки данных за период 19.10.2003–27.10.2003, станция Мыс Шмидта

мущений и имеют важное прикладное значение. Результаты данной работы служат подтверждением вышесказанному и показывают эффективность предложенного способа для детального анализа КЛ и выделения подобных эффектов.

5 Благодарности

Работа поддержана грантом РНФ № 14-11-00194. Авторы благодарят институты, поддерживающие станции регистрации данных, которые были использованы в исследовании, а также выражают признательность сотрудникам Института земного магнетизма, ионосферы и распространения радиоволн РАН, разработавшим интерактивную среду и программное обеспечение баз данных, обеспечивающих оперативное получение информации о параметрах состояния космического пространства.

Литература

[1] *Топтыгин И. Н.* Космические лучи в межпланетных магнитных полях. — М.: Наука, 1983. 301 с.

[2] *Тясто М. И., Данилова О. А., Дворников В. М., Сдобнов В. Е.* Большие снижения геомагнитных порогов космических лучей в период возмущений магнитосферы // Известия РАН, сер. физическая, 2009. Т. 73. № 3. С. 385–388.

[3] *Зеленый Л. М., Веселовский И. С., Бреус Т. К., и др.* Плазменная гелиогеофизика / Под общ. ред. Л. М. Зеленого, И. С. Веселовского. — М.: Физматлит, 2008. Т. 2. 560 с.

[4] *Vecchio A., Laurenza M., Storini M., Carbone V.* New insights on cosmic ray modulation through a joint use of nonstationary data-processing methods // Adv. Astronomy, 2012. doi: `http://dx.doi.org/10.1155/2012/834247.2012`.

[5] *Kóta J., Somogyi A.* Some problems of investigating periodicities of cosmic rays // Acta Physica Academiae Scientiarum Hungaricae, 1969. Vol. 27. P. 523—548.

[6] *Ruffolo D.* Transport and acceleration of energetic charged particles near an oblique shock // Astrophys. J., 1999. No. 515. P. 787-800.

[7] *Belov A. V., Bieber J. W., Eroshenko E. A., Evenson P., Pyle R., Yanke V. G.* Cosmic ray anisotropy before and during the passage of major solar wind disturbances // Adv. Space Res., 2003. Vol. 31. No. 4. P. 523—548.

[8] *Paschalis P., Sarlanis C., Mavromichalaki H.* Artificial neural network approach of cosmic ray primary data processing // Solar Phys., 2013. Vol. 1. No. 182. P. 303–318.

[9] *Мандрикова О. В., Заляев Т. Л.* Моделирование вариаций космических лучей на основе совмещения кратномасштабных вейвлет-разложений и нейронных сетей переменной структуры // Цифровая обработка сигналов, 2015. № 1. С. 11–16.

[10] *Козлов В. И.* Оценка скейлинговых свойств динамики флуктуаций космических лучей в цикле солнечной активности // Геомагнетизм и аэрономия, 1999. Т. 39. № 1. С. 100–104.

[11] *Козлов В. И., Марков В. В.* Вейвлет-образ тонкой структуры 11-летнего цикла по исследованию флуктуаций космических лучей в 20—23 циклах // Геомагнетизм и аэрономия, 2007. Т. 47. № 1. С. 47–55.

[12] *Козлов В. И., Марков В. В.* Вейвлет-образ гелиосферной бури в космических лучах // Геомагнетизм и аэрономия, 2007. Т. 47. № 1. С. 56–65.

[13] *Козлов В. И., Козлов В. В.* Новый индекс солнечной активности — индекс мерцаний космических лучей // Геомагнетизм и аэрономия, 2008. Т. 48. № 4. С. 1–9.

[14] *Mandrikova O. V., Solovev I. S., Zalyaev T. L.* Methods of analysis of geomagnetic field variations and cosmic ray data // Earth Plan. Space, 2014. Vol. 66. No. 148. doi: `http://dx.doi.org/10.1186/s40623-014-0148-0`.

[15] *Zarrouk N., Bennaceur R.* Neural network and wavelets in prediction of cosmic ray variability: The North Africa as study case // Acta Astronautica, 2010. No. 66. P. 1008–1016.

[16] *Мандрикова О. В., Заляев Т. Л.* Моделирование вариаций космических лучей и выделение аномалий на основе совмещения вейвлет-преобразования с нейронными сетями // Машинное обучение и анализ данных, 2014. Т. 1. № 9. С. 1154–1167.

[17] *Daubechies I.* Ten lectures on wavelets. — Philadelphia: SIAM, 1992. 357 p.

[18] *Mallat S.* A wavelet tour of signal processing: The sparse way. — 3rd ed. — USA: Academic Press, 2008. 832 p.

[19] *Ротанова Н. М., Бондарь Т. Н., Иванов В. В.* Вейвлет-анализ вековых геомагнитных вариаций // Геомагнетизм и аэрономия, 2004. № 2. С. 276–282.

[20] *Zaourar N., Hamoudi M., Mandea M., Balasis G., Holschneider M.* Wavelet-based multiscale analysis of geomagnetic disturbance // Earth Planets Space, 2013. Vol. 65. No. 12. P. 1525–1540.

[21] *Mandrikova O. V., Solovev I., Geppener V., Taha Al-Kasasbehd R., Klionskiy D.* Analysis of the Earth's magnetic field variations on the basis of a wavelet-based approach // Digital Signal Proc., 2013. Vol. 23. P. 329-339.

[22] *Haykin S.* Neural networks: A comprehensive foundation. — 2nd ed. — New York, NY, USA: Prentice Hall, 1999. 842 p.

[23] *Агеев А. Д., Балухта А. Н., Бычков А. В. и др.* Нейроматематика / Под общ. ред. А. И. Галушкина. — М.: ИПРЖР, 2002. 448 с.

[24] *Chui C. K.* An introduction in wavelets. — New York, NY, USA: Academic Press, 1992. 264 p.

[25] *Akasofu S. I., Chapman S.* Solar–terrestrial physics. — Oxford: Oxford University Press, 1972. 891 p.

[26] *Jaffard S.* Pointwise smoothness, two-microlocalization, and wavelet coefficients // Publications Mathématiques, 1991. Vol. 35. P. 155-168.

[27] *Левин Б. Р.* Теоретические основы статистической радиотехники. — 2-е изд. — М.: Сов.радио, 1975. 392 с.

[28] *Mandrikova O. V., Fetisova N. V., Polozov Y. A., Solovev I. S., Kupriyanov M. S.* Method for modeling of the components of ionospheric parameter time variations and detection of anomalies in the ionosphere // Earth Planet Space, 2015. Vol. 67. doi: `http://dx.doi.org/10.1186/s40623-015-0301-4`.

[29] *Munakata K., Bieber J. W., Yasue S., Kato C., Koyama M., Akahane S., Fujimoto K., Fujii Z., Humble J. E., Duldig M. L.* Precursors of geomagnetic storms observed by muon detector network // J. Geophys. Res., 2000. No. 105. P. 27457–27468.

Поступила в редакцию 15.06.2016

Modeling and analysis of cosmic ray variations during periods of increased solar and geomagnetic activity*

O. V. Mandrikova, T. L. Zalyaev, Yu. A. Polozov, and I. S. Solovev

oksanam1@mail.ru, tim.aka.geralt@mail.ru, up_agent@mail.ru, kamigsol@yandex.ru

Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS

7 Mirnaya Str., Paratunka, Elizovskiy District, Kamchatka Region, Russia

A method for analysis of cosmic ray variations, which allows one to allocate anomalous changes and to obtain quantitative estimates of their occurrence time, duration, and intensity is described. The method includes decomposition of neutron monitor data based on wavelet transform and their approximation based on adaptive variable structure neural networks. Using this method, an analysis of cosmic ray variations during the periods of increased solar and geomagnetic activity has been performed and anomalous changes that occurred a few hours before geomagnetic storms have been allocated. Long and deep Forbush decreases took place during the storms (neutron monitor data from Apatity and Cape Schmidt stations have been analyzed). Cosmic ray data have been analyzed together with geomagnetic field variations and ionospheric parameters, which processing has been performed on the basis of methods proposed by the authors.

Keywords: *cosmic rays; magnetic storm; ionospheric parameters; data analysis; wavelet transform; neural networks*

DOI: 10.21469/22233792.2.1.07

References

[1] Toptygin, I. N. 1983. *Kosmicheskie luchi v mezhplanetnykh magnitnykh polyakh* [Cosmic rays in the interplanetary magnetic field]. Moscow: Nauka. 301 p.

[2] Tyasto, M. I., O. A. Danilova, V. M. Dvornikov, and V. E. Sdobnov. 2009. Bol'shie snizheniya geomagnitnykh porogov kosmicheskikh luchey v period vozmushcheniy magnitosfery [Large reduction of geomagnetic thresholds of cosmic rays in the magnetosphere during disturbances]. *Izvestiya RAN, ser. fizicheskaya* [Izvestiya RAN, physical ser.] 73(3):385–388.

[3] Zelenyy, L. M., I. S. Veselovskiy, T. K. Breus, *et al.* 2008. *Plazmennaya geliogeofizika* [Plasma geliogeofizika]. Eds. L. M. Zelenyy and I. S. Veselovsky. Moscow: Fizmatlit. Vol. 2. 560 p.

[4] Vecchio, A., M. Laurenza, M. Storini, and V. Carbone. 2012. New insights on cosmic ray modulation through a joint use of nonstationary data-processing methods. *Adv. Astronomy.* doi: http://dx.doi.org/10.1155/2012/834247.2012.

[5] Kóta, J., and A. Somogyi. 1969. Some problems of investigating periodicities of cosmic rays. *Acta Physica Academiae Scientiarum Hungaricae* 27:523–548.

[6] Ruffolo, D. 1999. Transport and acceleration of energetic charged particles near an oblique shock. *Astrophys. J.* 515:787–800.

[7] Belov, A. V., J. W. Bieber, E. A. Eroshenko, P. Evenson, R. Pyle, and V. G. Yanke. 2003. Cosmic ray anisotropy before and during the passage of major solar wind disturbances. *Adv. Space Res.* 31(4):523–548.

[8] Paschalis, P., C. Sarlanis, and H. Mavromichalaki. 2013. Artificial neural network approach of cosmic ray primary data processing. *Solar Phys.* 1(182):303–318.

*The research was supported by the grant of the Russian Science Foundation No. 14-11-00194.

[9] Mandrikova, O. V., and T. L. Zalyaev. 2015. Modelirovanie variatsiy kosmicheskikh luchey na osnove sovmeshcheniya kratnomasshtabnykh veyvlet-razlozheniy i neyronnykh setey peremennoy struktury [Modeling variations of cosmic rays on the basis of combination of multiresolution wavelet decomposition and neural networks with variable structure]. *Tsifrovaya obrabotka signalov* [Digital signal processing] 1:11–16.

[10] Kozlov, V. I. 1999. Otsenka skeylingovykh svoystv dinamiki fluktuatsiy kosmicheskikh luchey v tsikle solnechnoy aktivnosti [Estimation of the scaling properties of the dynamics of fluctuations of cosmic rays in the solar activity cycle]. *Geomagnetizm i aeronomiya* [Geomagnetism and Aeronomy] 39(1):100–104.

[11] Kozlov, V. I., and V. V. Markov. 2007. Veyvlet-obraz tonkoy struktury 11-letnego tsikla po issledovaniyu fluktuatsiy kosmicheskikh luchey v 20–23 tsiklakh [Wavelet image of the fine structure of the 11-year cycle to study the fluctuations of cosmic rays in 20-23 cycles]. *Geomagnetizm i aeronomiya* [Geomagnetism and Aeronomy] 47(1):47–55.

[12] Kozlov, V. I., and V. V. Markov. 2007. Veyvlet-obraz geliosfernoy buri v kosmicheskikh luchakh [Wavelet image of heliospheric storm in cosmic rays]. *Geomagnetizm i aeronomiya* [Geomagnetism and Aeronomy] 47(1):56–65.

[13] Kozlov, V. I., and V. V. Kozlov. 2008. Novyy indeks solnechnoy aktivnosti — indeks mertsaniy kosmicheskikh luchey [New solar activity index — the index of cosmic ray scintillation]. *Geomagnetizm i aeronomiya* [Geomagnetism and Aeronomy] 48(4):1–9.

[14] Mandrikova, O. V., I. S. Solovev, and T. L. Zalyaev. 2014. Methods of analysis of geomagnetic field variations and cosmic ray data. *Earth Plan. Space* 66(148). doi: `http://dx.doi.org/10.1186/s40623-014-0148-0`.

[15] Zarrouk, N., and R. Bennaceur. 2010. Neural network and wavelets in prediction of cosmic ray variability: The North Africa as study case. *Acta Astronautica* 66:1008–1016.

[16] Mandrikova, O. V., and T. L. Zalyaev. 2014. Modelirovanie variatsiy kosmicheskikh luchey i vydelenie anomaliy na osnove sovmeshcheniya veyvlet-preobrazovaniya s neyronnymi setyami [Simulation of cosmic ray variations and anomalies in the allocation on the basis of combining wavelet transform with neural networks]. *Mashinnoe obuchenie i analiz dannykh* [Machine Learning Data Anal.] 1(9):1154–1167.

[17] Daubechies, I. 1992. *Ten lectures on wavelets.*. Philadelphia: SIAM. 357 p.

[18] Mallat, S. 2008. *A wavelet tour of signal processing. The sparse way.* 3rd ed. USA: Academic Press. 832 p.

[19] Rotanova, N. M., T. N. Bondar', and V. V. Ivanov. 2004. Veyvlet-analiz vekovykh geomagnitnykh variatsiy [Wavelet analysis of secular geomagnetic variations]. *Geomagnetizm i aeronomiya* [Geomagnetism and Aeronomy] 2:276–282.

[20] Zaourar, N., M. Hamoudi, M. Mandea, G. Balasis, and M. Holschneider. 2013. Wavelet-based multiscale analysis of geomagnetic disturbance. *Earth Planets Space* 65(12):1525–1540.

[21] Mandrikova, O. V., I. Solovev, V. Geppener, R. Taha Al-Kasasbehd, and D. Klionskiy. 2013. Analysis of the Earth's magnetic field variations on the basis of a wavelet-based approach. *Digital Signal Proc.* 23:329–339.

[22] Haykin, S. 1999. *Neural networks: A comprehensive foundation.* 2nd ed. New York, NY: Prentice Hall. 842 p.

[23] Ageev, A. D., A. N. Balukhta, A. V. Bychkov, *et al.* 2002. *Neyromatematika.* [Neuromathematic]. Ed. A. I. Galushkin. Moscow: IPRZhR. 448 p.

[24] Chui, C. K. 1992. *An introduction in wavelets.* New York, NY: Academic Press. 264 p.

[25] Akasofu, S. I., and S. Chapman. 1972. *Solar–terrestrial physics.* Oxford: Oxford University Press. 891 p.

[26] Jaffard, S. 1991. Pointwise smoothness, two-microlocalization, and wavelet coefficients. *Publications Mathématiques* 35:155–168.

[27] Levin, B. R. 1975. *Teoreticheskie osnovy statisticheskoy radiotekhniki* [Theoretical foundations of statistical radio engineering]. Moscow: Sov. Radio. 392 p.

[28] Mandrikova, O. V., N. V. Fetisova, Y. A. Polozov, I. S. Solovev, and M. S. Kupriyanov. 2015. Method for modeling of the components of ionospheric parameter time variations and detection of anomalies in the ionosphere. *Earth Planet Space* 67. doi: `http://dx.doi.org/10.1186/s40623-015-0301-4`.

[29] Munakata, K., J. W. Bieber, S. Yasue, C. Kato, M. Koyama, S. Akahane, K. Fujimoto, Z. Fujii, J. E. Humble, and M. L. Duldig. 2000. Precursors of geomagnetic storms observed by muon detector network // *J. Geophys. Res.* 105:27457–27468.

Received June 15, 2016

Agent-based simulation modeling for regional ecological-economic systems. A case study of the Republic of Armenia*

***L. A. Beklaryan*[1], *A. S. Akopov*[2], *A. L. Beklaryan*[2], *and A. K. Saghatelyan*[3]**
beklar@cemi-rssi.ru, aakopov@hse.ru, abeklaryan@hse.ru, ecocentr@sci.am
[1]Central Economics and Mathematics Institute of the Russian Academy of Sciences
47 Nachimovski prosp., Moscow, Russia
[2]National Research University Higher School of Economics
33 Kirpichnaya st., Moscow, Russia
[3]Center for Ecological-Noosphere Studies of the National Academy of Sciences of RA
68 Abovyan st., Yerevan, Armenia

Actual problems of modeling of ecologic-economic systems on the example of the Republic of Armenia (RA) are considered. Based on the methods of agent modeling and system dynamics, the simulation model of ecological-economic system which has allowed constructing the RA Ecological Map was created. The important purpose of the suggested approach is search of scenarios of rational modernization of the agent-enterprises, which are the main sources of emissions with simultaneous definition of effective strategy of the government regulation. The bi-criterial optimization problem for the ecological-economic system of RA is formulated and solved with the help of the developed genetic algorithm.

Keywords: *ecological-economic system; simulation modeling; system dynamics; agent-based modeling; multiobjective optimization*

DOI: 10.21469/22233792.2.1.08

1 Introduction

At the present time, important direction in the field of sustainable development of ecological-economic systems, which is known as "Ecological economics" [1], is developing significantly.

The main feature of this direction is studying of long-term dynamics of the ecological-economic system, taking into account interaction of key economic agents: the production and refining enterprises being the main sources of emissions; consumers (both internal, and external), the government, which is carrying out the regulating functions, in particular, regarding the enterprises — main sources of emissions; vehicles and green companies, which are carrying out pure ecological products, human resources, people and environment, including minerals, water resources, power sources, land, forest, etc.

Complexity of seeking the best scenarios of long-term development of ecological-economic system is caused by not only the large-scale of such systems and their elements, but, firstly, an availability of internal nonlinear multiple feedbacks which are both balanced and reinforcing that leads to occurrence of difficult predicted multiplicative effect. For example, restriction of mining obviously reduces the level of emissions to the atmosphere, however, also leads to decrease in cumulative profit and tax revenues in the budget of the region. On the other hand, improvement of the general ecological situation in the region attracts the development of green branches of the economy, in particular, such as tourism, sectors of high technologies, green agriculture, etc. which can substitute nonecological economy industries.

*The research was partially supported by the Russian Foundation for Basic Research (grant 15-51-05011 Arm_a).

Therefore, for seeking scenarios of long-term development of similar complex ecological-economic systems, it is necessary to use the methods of system dynamics [2] and agent-based modeling, allowing to investigate the multiplicative influence of internal feedbacks, dynamical flows of natural resources in a complex interdependence between key economic agents (in particular, between enterprises and the government) and possibilities of effective government regulation.

The idea of use of methods of system dynamics and agent-based modeling for research of ecological-economic systems is not new. In particular, possibilities of recovery of the ecological-economic system of South Africa [3], potential of development of ecological agriculture of China [4], and loudspeakers of stocks of water resources of China [5], loudspeakers of hunting natural resources of South America [6], etc. were investigated.

In this article, the concept of the agent-oriented model of ecological-economic system of RA is provided. Not detailed ecological map of Armenia on which dynamics of transition of manufacturing agents from initial conditions of the main sources of emissions of harmful substances to target final state — environmentally friendly productions is visualized is constructed.

The purpose of the work consists in the system analysis of the major characteristics of ecological-economic system on the example of RA taking into account the available interrelations between key economic agents and environment and forming of strategy of government regulation for manufactures in order to motivate them to transit to environmentally friendly production.

2 Model of ecological-economic system of the Republic of Armenia

The common simulation model of ecological-economic system of RA is developed using the methods of system dynamics and agent-based modeling. In this model, enterprises, which are the main sources of pollution of the atmosphere, need to be modernized for minimization of emissions level of harmful substances, have been identified. Such approach has allowed estimating forecast dynamics of emissions reduction of the main harmful substances at the atmosphere, such as carbon oxides (COx), hydrocarbons (CH), sulfur dioxide (SO_2), flying organic substances (NMVOC), nitrogen oxides (NOx), and other substances. At the same time, there is mechanism of the government regulation of activity of enterprises by means of the penalties and subsidies directed to refusal of use of the technologies polluting environment in favor of environmentally friendly technologies. The state chart, which is the algorithm of behavior of manufacturing agents considering influence of the government regulation in the form of penalties (for exceeding of limits of emissions of harmful substances) and subsidies (at essential decrease in emissions level) is represented in Fig. 1.

In Fig. 1, it is shown that agents-enterprises can have four possible final states:

1. Not ecological manufacturing (not ecological production) — production by which the enterprise remains to be one of the main sources of emissions of harmful substances.
2. Closing of the company — complete elimination of the enterprise as a result of which the source of emissions of harmful substances is eliminated.
3. Partial modernization — the enterprise is essential (in comparison with previous, the temporary period) reduces emissions level of harmful substances.
4. Ecological manufacturing (environmentally friendly production) — production by which the enterprise stops being a source of emissions of harmful substances.

Let us provide the formal description of the main part of the developed agent-oriented model of ecological-economic system of RA relating to behavior of agents enterprises with the mechanism of the government regulation. Let us note that the model of dynamics of

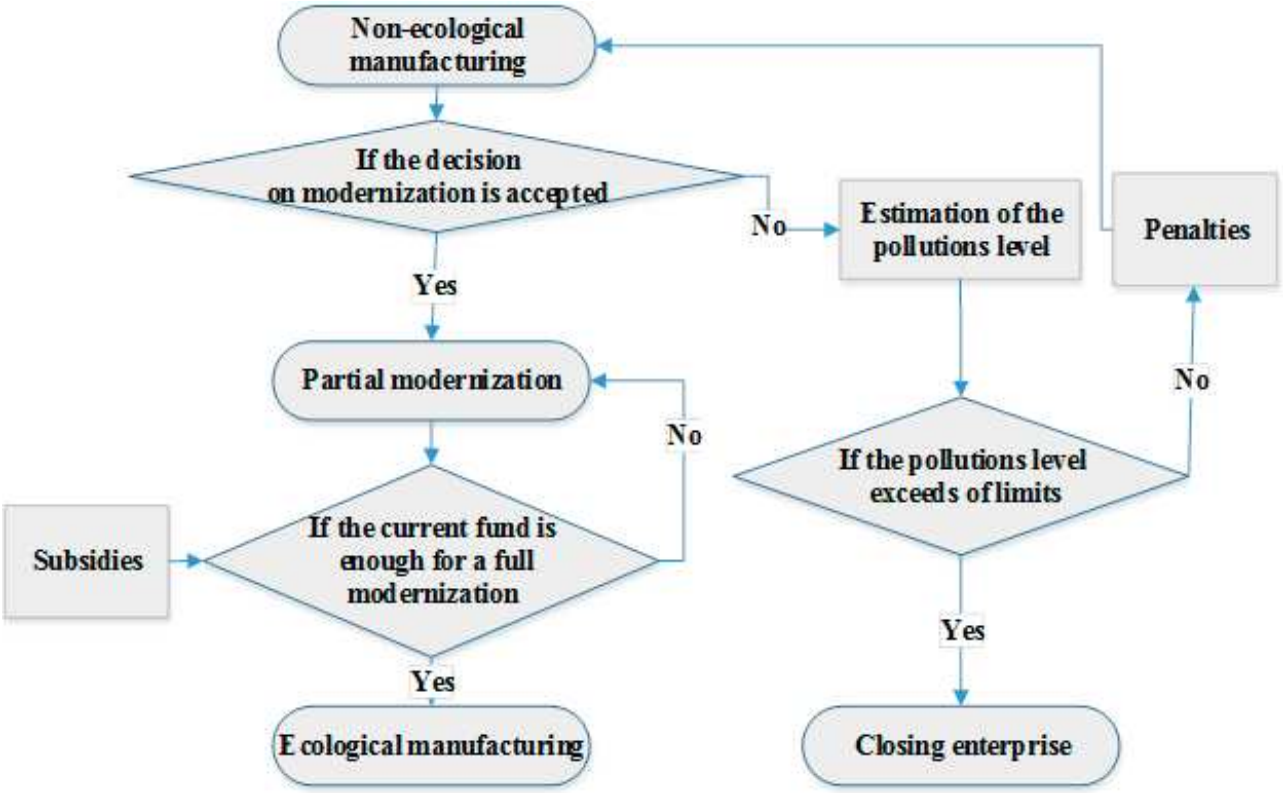


Figure 1 Algorithm of behavior of agents-enterprises

environment taking into account characteristics of all agents occupying it demands separate detailed consideration and is beyond this article.

Let us enter the following designations:

$t \in \{t_0, \ldots, t_0 + T\}$ — time by years (t_0 is the initial time moment and T is the horizon of strategic planning);

$J(t)$ — set of indexes of the agents enterprises which are the sources of emissions of harmful substances; $j \in J(t)$ — index of the agent enterprise at the time moment t; $i \in \{1, \ldots, I\}$ — kinds of harmful substances: SO_2, NOx, NMVOC, CH, etc.;

$\mathrm{st}_j(t) \in \{1, 2, 3, 4\}$ — possible states of jth-agent enterprise: $\mathrm{st}_j(t) = 1$ — initial condition of not ecological production; $\mathrm{st}_j(t) = 2$ — partial modernization; $\mathrm{st}_j(t) = 3$ — environmentally friendly production; $\mathrm{st}_j(t) = 4$ — closing the enterprise caused by the violation of the ecological legislation;

$\gamma_j(t)$ — the coefficient determining the volume of emissions of harmful substances in the atmosphere depending on state of jth-agent enterprise:

$$\gamma_j(t) = \begin{cases} 1, & \text{if } \mathrm{st}_j(t) = 1; \\ 0.5, & \text{if } \mathrm{st}_j(t) = 2; \\ 0, & \text{if } \mathrm{st}_j(t) = 3 \text{ or } \mathrm{st}_j(t) = 4; \end{cases}$$

$E_j(t)$ — total volume of emissions of harmful substances in the atmosphere of jth-agent enterprise:

$$E_j(t) = \sum_{i=1}^{I} \gamma_j(t) e_{ij}$$

where e_{ij} is the average volume of emissions of ith-harmful substances in the atmosphere of jth-agent enterprise;

$C_j(t)$ — costs of modernization or elimination of jth-agent enterprise:

$$C_j(t) = \begin{cases} p_j(t)(1-\gamma_j(t)), & \text{if } \mathrm{st}_j(t) \neq 4; \\ \overline{p}_j(t), & \text{if } \mathrm{st}_j(t) = 4 \end{cases}$$

where $p_j(t)$ is the modernization cost of jth-agent enterprise and $\overline{p}_j(t)$ is the elimination cost of jth-agent enterprise;

$V_j(t)$ — release total volume of the jth-agent enterprise, it is calculated with the help of well-known production function of Kobb–Douglas:

$$V_j(t) = \begin{cases} A_{0,j}(t)(L_j(t))^{\alpha_j(t)}(K_j(t)\gamma_j(t))^{\beta_j(t)}, & \text{if } \mathrm{st}_j(t) = 1 \text{ or } \mathrm{st}_j(t) = 2; \\ A_{0,j}(t)(L_j(t))^{\alpha_j(t)}(\underline{K}_j(t))^{\beta_j(t)}, & \text{if } \mathrm{st}_j(t) = 3; \\ 0, & \text{if } \mathrm{st}_j(t) = 4 \end{cases}$$

where $A_{0,j}(t)$ is the factor of scientific and technical progress for jth-agent enterprise; $L_j(t)$ is the number of human resource of jth-agent enterprise; $K_j(t)$ is the fixed assets of jth-agent enterprise; $\underline{K}_j(t)$ is the minimum necessary fixed assets of jth-agent enterprise by environmentally friendly production; $\alpha_j(t)$ and $\beta_j(t)$ are the parameters of production function of Kobb–Douglas, for which $\alpha_j(t) + \beta_j(t) = 1$ for all $j \in \{1, \ldots, J(t)\}$;

$D_j(t)$ — grants for modernization of jth-agent enterprise from the state within the suggested model are paid only to the enterprises which are already partially modernized and on condition of deficit of own means necessary for full modernization and to transition to environmentally friendly production:

$$D_j(t) = \begin{cases} \lambda_j(t)C_j(t), & \text{if } \mathrm{st}_j(t-1) = 2 \text{ and } C_j(t) - P_j(t-1) > 0; \\ 0, & \text{if } \mathrm{st}_j(t-1) \neq 2 \text{ or } C_j(t) - P_j(t-1) \leqslant 0 \end{cases}$$

where $\lambda_j(t)$ is the control parameter of system of the government regulation defining share of the costs of modernization subsidized from the government ($0 < \lambda_j(t) \leqslant 1$);

$\overline{D}(t)$ — the maximum permissible volume of subsidies, which can be directed on modernization of all agents enterprises from the government;

$F_j(t)$ — penalties of jth-agent enterprise from the government, caused by essential violation of the ecological legislation:

$$F_j(t) = \begin{cases} \eta_j(t)P_j(t-1), & \text{if } E_j(t) \geqslant \overline{E}_j(t) \text{ and } \mathrm{st}_j(t-1) \neq 3; \\ 0, & \text{if } E_j(t) \leqslant \overline{E}_j(t) \text{ or } \mathrm{st}_j(t-1) = 3 \end{cases}$$

where $\eta_j(t)$ is the the control parameter of system of the government regulation defining the share of the profit got by the agent enterprise in the previous period directed on payment of penalties ($0 < \eta_j(t) \leqslant 1$) and $\overline{E}_j(t)$ is the maximum permissible level of harmful substances in the atmosphere for this enterprise; and

$P_j(t)$ — profit of jth-agent enterprise:

$$P_j(t) = \tilde{p}_j(t)V_j(t) - C_j(t) + D_j(t) - F_j(t) - \mathrm{Const}_j(t)$$

where $\tilde{p}_j(t)$ is the average prices for products of jth-agent enterprise and $\mathrm{Const}_j(t)$ is the cumulative constant expenses of jth-agent enterprise (including labor costs, taxes, depreciation of fixed assets, etc.).

Let us note that within this model, two types of agents are considered. Agents enterprises treat the first type, being the main sources of emissions of harmful substances in the atmosphere, maximizing in each time moment t own profit. The government aiming at minimization of cumulative emissions level of harmful substances in the atmosphere at the expense of choice of the optimum strategy of the government regulation (differentiated in relation to agents enterprises) treats the second type. As a result, it is possible to formulate bi-criterial optimization problem for the considered ecological-economic system.

3 Problem definition

Problem 1. It is required to maximize profit of each jth-agent enterprise under minimum possible cumulative emissions of harmful substances in the atmosphere in each time moment t

$$\begin{cases} \max\limits_{\mathrm{st}_j(t)} P_j(t) \\ \min\limits_{\lambda_j(t),\eta_j(t)} E_j(t) \end{cases} \tag{1}$$

under restrictions:

$$\sum_{j\in J(t)} D_j(t) \leqslant \overline{D}(t), \quad P_j(t) \geqslant 0; \tag{2}$$

$$\mathrm{st}_j(t) \in \{1,2,3,4\},\ 0 < \eta_j(t) \leqslant 1,\ j \in J(t), \tag{3}$$

and other restrictions making clear economic sense.

As a result of the solution of the problems (1)–(3), the subset of optimal decisions is uniformly distributed along Pareto front. In order to get such subset with required level of a time effectivity, we applied genetic algorithms, which are similar to that described in works [7–10] and based on the class of Strength Pareto Evolutionary Algorithms (SPEA, SPEA2, etc.). The scheme of the developed genetic algorithm (GA) is presented in Fig. 2.

The developed GA is based on the classic evolution approach, which requires applying operators of selection, crossover, and mutation (see Fig. 2). The main feature of the developed GA is taking into account internal restrictions, which are implemented on the agent level. In particular, transitions from the finite states *"closing of the company"* and *"ecological manufacturing"* to other states are impossible. The transition from the state of a *"partial modernization"* to the state of an *"ecological manufacturing"* is possible only if the current fund is enough for a full modernization (when $P_j(t) \geqslant 0$).

Therefore, the GA considers only those states of agents, which are dynamical coordinated. Thus, GA touches possible trajectories of modernization of agent enterprises estimating them impact on the ecological-economic system.

It should be noted that there are some important blocks in the developed GA (see Fig. 2):

- *Forming initial population of decisions for each jth-agent.* As a rule, the initial state is defined as not ecological manufacturing ($\mathrm{st}_j(t_0) = 1, j \in J(t_0)$). However, over time, the state of agents can be changed. First of all, the GA provides forming some possible values of $\mathrm{st}_j(t)$ in each time moment $t \in \{t_0, \ldots, t_0 + T\}$. The choice of such values can be random under restrictions;
- *Loading input data to the simulation.* The simulation model of the ecological-economic system was implemented in AnyLogic system. The simulation model deals with real datasets having data about characteristics of agents enterprises of RA. Therefore, the developed

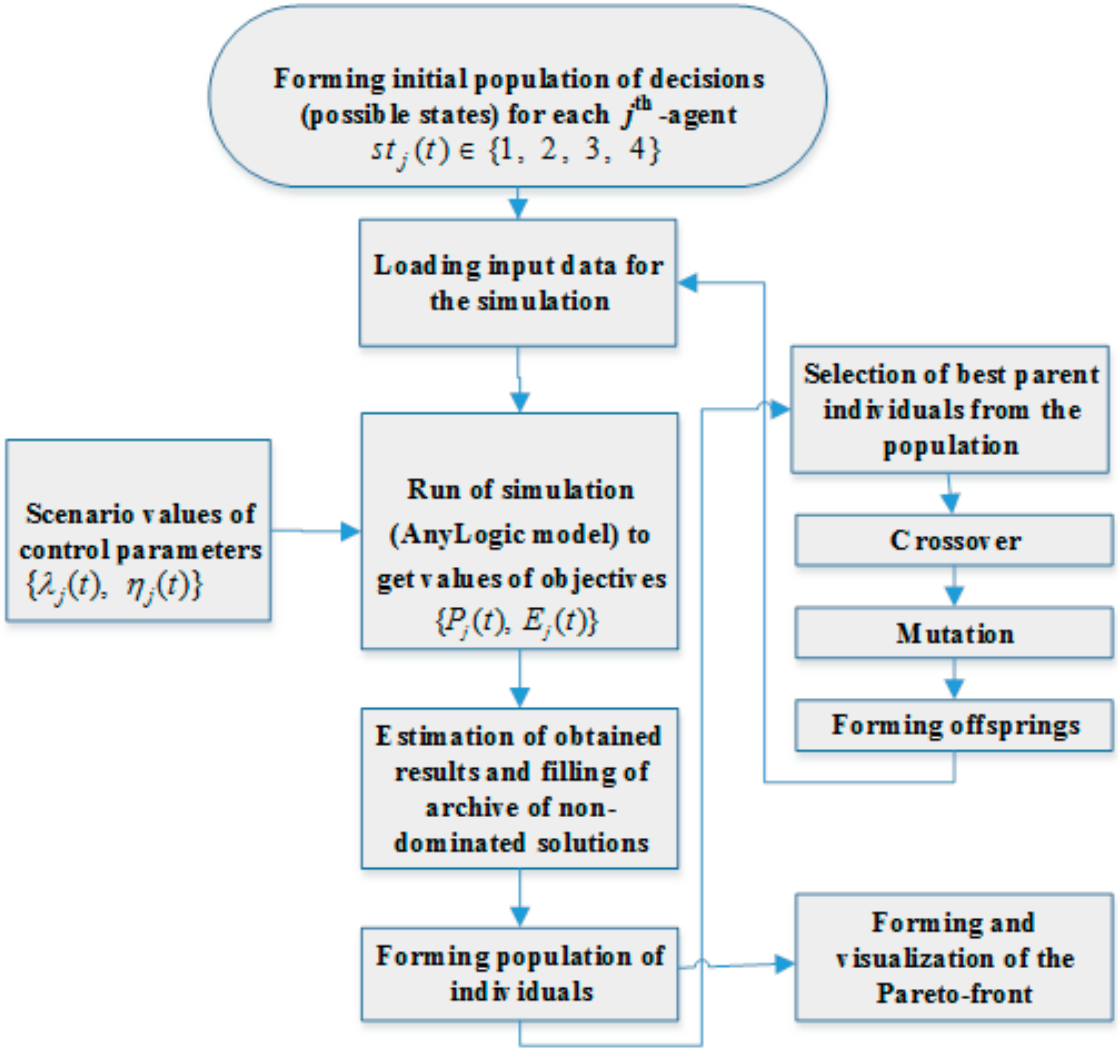


Figure 2 The scheme of the developed GA

model was integrated with the database and input data are being loaded in the model in a simulation process. Input data consist of control parameters, in particular, such as $\{\lambda_j(t), \eta_j(t)\}$ and exogenous variables such as parameters of Kobb–Duglas production function $\{A_{0,j}(t), \alpha_j(t), \beta_j(t)\}$, the predictive dynamics of resources $\{L_j(t), K_j(t)\}$, and other exogenous variables;

– *Run of simulation (AnyLogic model) to get values of objectives.* The simulation model is intended to obtain values of objective functions $\{P_j(t), E_j(t)\}$ under different values of control parameters;
– *Estimation of obtained results and filling of archive of nondominated solutions.* The filling of archive of nondominated solutions is implemented with the help of SPEA2-algorithm [8]. The algorithm is based on weighting of solutions (individuals) in proportionality to amount of its dominated solutions. Hence, individuals with smaller weights have more priority under the selection. It should be noted that such algorithm deals with convex objective functions only;

– *Forming population of individuals* is based in iteration moving individuals towards the Pareto-front. For this purpose, individuals having highest Pareto ranking will be assigned larger priority to including in the population; and
– *Selection of best parent individuals from the population, crossover, mutation, and forming offsprings* are the standard operators of the GA.

The choice of the final scenario of modernization of the enterprises taking into account aspiration to maximizing their profit with simultaneous minimization of cumulative emissions is carried out taking into account the additional preferences created based on the analysis of possible effects of implementation of the corresponding strategy of modernization. Thus, the structure of emissions of substances in the atmosphere (on each large enterprise) and their contribution to dynamics of incidence of the population is estimated. Let us note that different chemical elements have the differentiated impact on state of health of the population. For example, excess concentration of carbon oxide (CO) leads to organism intoxication (at certain concentration, by lethal outcome), systematic sulfur dioxide emissions lead to growth of pulmonary diseases, and nitrogen oxide — to growth of respiratory diseases. Each type of diseases renders different social and economic effects, which need to be considered at decision-making, including not eco-friendly enterprises relating to closing.

4 Simulation model of ecological-economic system

Further on, based of the suggested algorithm of behavior of agents enterprises (see Fig. 1) taking into account dynamics of environment (see Fig. 2) and bi-criteria optimization problems (1)–(3), the simulation model of ecological-economic system implemented in AnyLogic system (Fig. 3) has been developed. Important advantages of AnyLogic system are as follows:

– possibility of the combined use of different methods of simulation modeling, in particular, system dynamics and agent-based modeling;
– ability to aggregate the simulation model with databases;

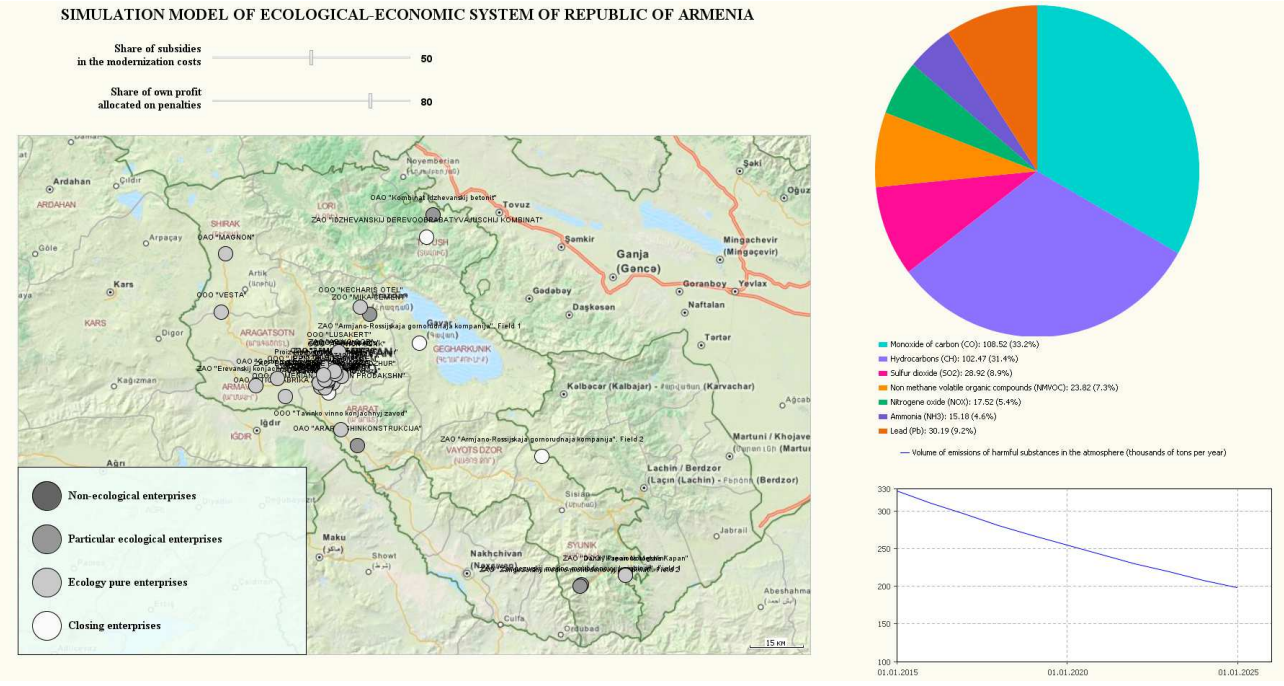


Figure 3 Simulation model of ecological-economic system of RA in AnyLogic system

– ability to aggregate the simulation model with geographic information systems (GIS), including possibility of dynamic visualization of condition of agents on map (see Fig. 3); and
– ability to aggregate the simulation model with the developed genetic optimization algorithm, in particular, providing forming of subset of optimum decisions across Pareto for the considered problem of ecological-economic system. This algorithm is described in [9, 10].

Let us note that the developed simulation model is integrated with the subject-oriented database containing up-to-date information about stationary sources of emissions — the RA enterprises, created with use of statistical data of the Center for Ecological-Noosphere Studies of the National Academy of Sciences of RA provided by the Center[1].

5 Results of the simulation modeling of ecological-economic system of the Republic of Armenia

The results of simulation modeling are unloaded in database and visualized on maps (Figs. 4–6). Forecast dynamics of possible states is presented on the maps (till 2025), the modernized agents enterprises being the main sources of emissions of harmful substances in RA. With black color, the enterprises which are in condition of not eco-friendly production, dark gray color — the partial modernized enterprises, light-gray color — the enterprises which have become environmentally friendly owing to full modernization, white color — the enterprises which are desirable for closing are allocated. Let us note that on the maps, not the legal but the actual addresses of the enterprises (including multiple) which are directly relating to sources of the corresponding emissions are visualized (for example, for the extracting enterprises, there are the coordinates of mining fields).

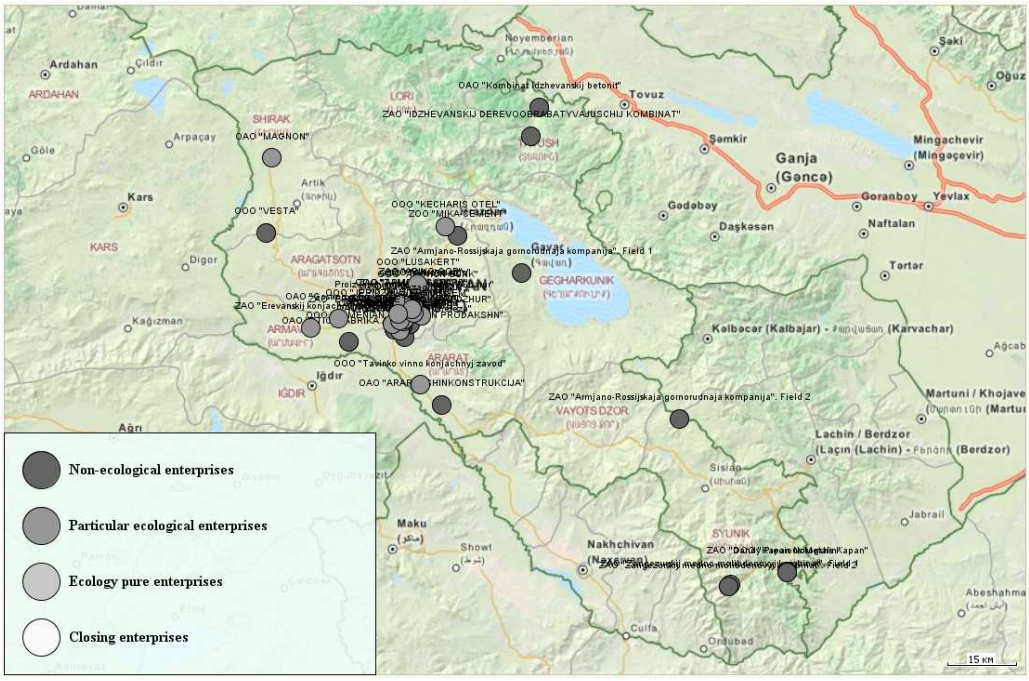


Figure 4 Initial condition of agents enterprises, being the main sources of emissions of harmful substances, visualized on the RA card for 2015

[1] http://ecocentre.am/.

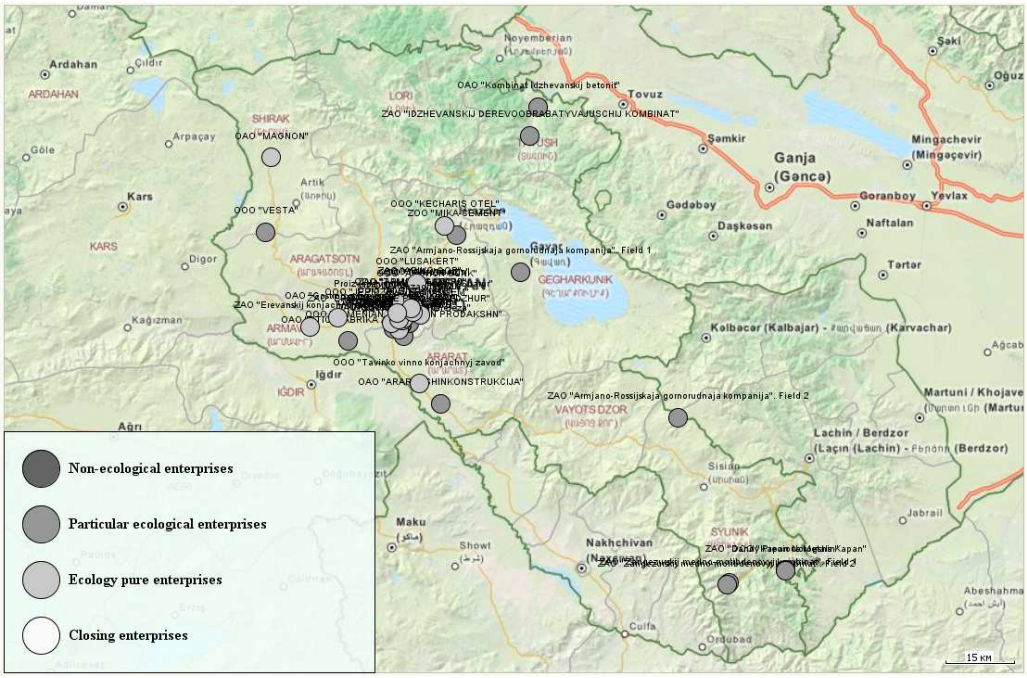


Figure 5 The intermediate condition of agents enterprises calculated by means of the developed simulation model for 2020

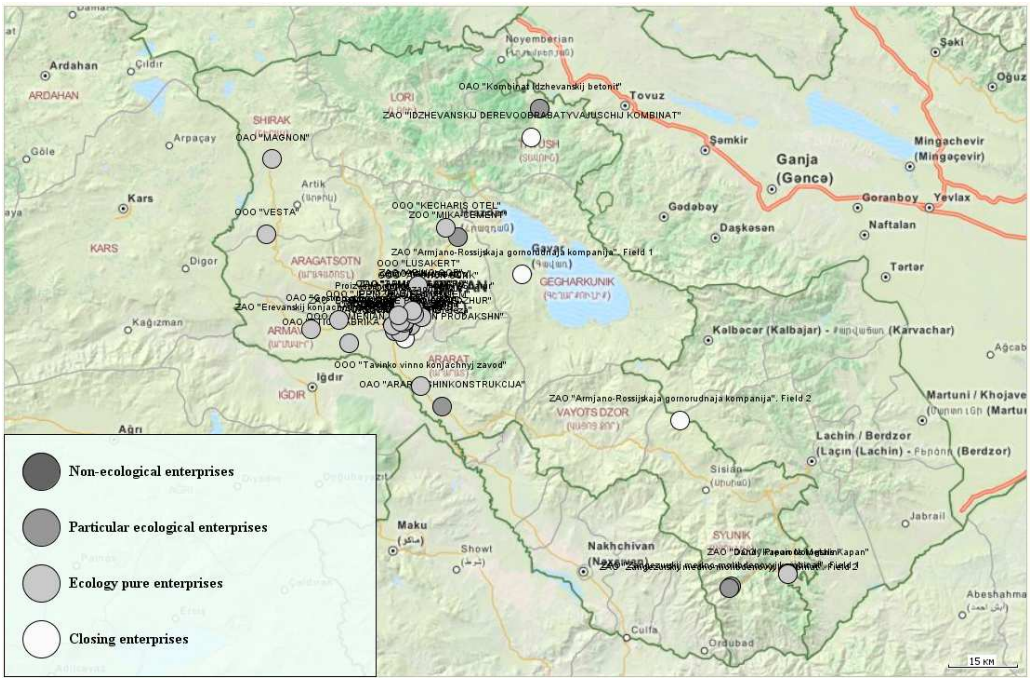


Figure 6 Optimum final condition of agents enterprises for 2025

Let us note that the developed decentralized agent-oriented model of ecological-economic systems is allowed forming the optimum plan of modernization of agent-enterprises through the scenario control of parameters of the government regulation such as penalties and subsidies. The final suggested scenario of modernization is chosen among subset of all received Pareto-optimal decisions as the most preferable to social and economic system. Thus, optimum values of parameters of the government regulation are found: $\lambda_j(t)$ (the share of costs of modernization

subsidized by the government) and $\eta_j(t)$ (the share of profit of the enterprise received for previous year, directed on penalties) for all considered agents enterprises $j \in J(t)$. Let us note that average optimum values $\tilde{\lambda}_j(t) \approx 50\%$ (for the separate enterprises of rather 10% subsidizing) and $\tilde{\eta}_j(t) \approx 80\%$ (for the separate enterprises penalties have to make up to 100% of the got profit).

6 Concluding Remarks

The simulation model of ecological-economic system of RA considering features of system of the government ecological regulation has been developed with the help of the methods of system dynamics and agent-based modeling. The results of simulation modeling allow drawing conclusion on availability of basic possibility of modernization of the majority of the RA enterprises which activity is followed essential to emissions of harmful substances in the atmosphere. Thus, transition to model of ecological economy is possible at the expense of own means of the enterprises and subsidies from the government. Unfortunately, some enterprises, in particular, which are carrying out production and processing of copper molybdenum concentrate, cannot be given to environmentally friendly production because of the existing technology restrictions and, therefore, reasonably gradually to preserve them with simultaneous creation of new environmentally friendly productions.

References

[1] Costanza, R., J. H. Cumberland, H. Daly, R. Goodland, and R. B. Norgaard. 1997. *An introduction to ecological economics*. CRC Press. 275 p.

[2] Forrester, J. W. 2013. *Industrial dynamics*. Martino Fine Books.

[3] Crookes, D. J., J. N. Blignaut, M. de Wit, and K. J. Esler. 2013. System dynamic modeling to assess economic viability and risk trade-offs for *J. Environmental Management* 120:138–147. doi: http://dx.doi.org/10.1016/j.jenvman.2013.02.001.

[4] Shi, T., and R. Gill. 2005. Developing effective policies for the sustainable development of ecological agriculture in China: Case study of Jinshan County with a systems dynamics model. *Ecological Economics* 53(2):223–246. doi: http://dx.doi.org/10.1016/j.ecolecon.2004.08.006.

[5] Zhang, Z., W. X. Lu, Y. Zhao, and W. B. Song. 2014. Development tendency analysis and evaluation of the water ecological carrying capacity in the siping area of Jilin Province in China based on system dynamics and analytic hierarchy process. *Ecological Modelling* 275:9–21. doi: http://dx.doi.org/10.1016/j.ecolmodel.2013.11.031.

[6] Iwamura, T., E. F. Lambin, K. M. Silvius, J. B. Luzar, and J. M. V. Fragoso. 2014. Agent-based modeling of hunting and subsistence agriculture on indigenous lands: Understanding interactions between social and ecological systems. *Environmental Modelling Software* 58:109–127. doi: http://dx.doi.org/10.1016/j.envsoft.2014.03.008.

[7] Zitzler, E., and L. Thiele. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evolutionary Comput.* 3(4):257–271. doi: http://dx.doi.org/10.1109/4235.797969.

[8] Bleuer, S., M. Brack, L. Thiele, and E. Zitzler. 2001. Multiobjective genetic programming: Reducing bloat using spea2. *Congress on Evolutionary Computation Proceedings*. Seoul: IEEE. 1:536–543.

[9] Akopov, A. S., and M. A. Hevencev. 2013. A multi-agent genetic algorithm for multi-objective optimization. *IEEE Conference (International) on Systems, Man and Cybernetics Proceedings*. Manchester: IEEE. 1391–1395.

[10] Akopov, A. S. 2014. Parallel genetic algorithm with fading selection. *Int. J. Comput. Appl. Technol.* 49(3/4):325–331. doi: http://dx.doi.org/10.1504/IJCAT.2014.062368.

Received June 15, 2016

Агентное моделирование региональной эколого-экономической системы. Тематическое исследование для Республики Армения*

***Л. А. Бекларян*[1], *А. С. Акопов*[2], *А. Л. Бекларян*[2], *А. К. Сагателян*[3]**

beklar@cemi-rssi.ru, aakopov@hse.ru, abeklaryan@hse.ru, ecocentr@sci.am

[1]Центральный экономико-математический институт РАН

Россия, г. Москва, Нахимовский проспект, 47

[2]Национальный исследовательский университет «Высшая школа экономики»

Россия, г. Москва, ул. Кирпичная, 33

[3]Центр эколого-ноосферных исследований Национальной академии наук Республики Армения

Армения, г. Ереван, ул. Абовяна, 68

Рассматриваются актуальные вопросы моделирования эколого-экономической системы на примере Республики Армения (РА). Основываясь на методах агентного моделирования и системной динамики, создана имитационная модель эколого-экономической системы, позволившая построить Экологическую карту РА. Важной целью предлагаемого подхода является поиск сценариев рациональной модернизации предприятий, являющихся основными источниками выбросов вредных веществ с одновременным определением эффективной стратегии государственного регулирования. Сформулирована и решена бикритериальная задача оптимизации характеристик эколого-экономической системы на примере РА.

Ключевые слова: *эколого-экономическая система; имитационное моделирование; системная динамика; агентно-ориентированное моделирование; многокритериальная оптимизация*

DOI: 10.21469/22233792.2.1.08

Литература

[1] *Costanza R., Cumberland J. H., Daly H., Goodland R., Norgaard R. B.* An introduction to ecological economics. — CRC Press, 1997. 275 p.

[2] *Forrester J. W.* Industrial dynamics. — Martino Fine Books, 2013.

[3] *Crookes D. J., Blignaut J. N., de Wit M., Esler K. J.* System dynamic modeling to assess economic viability and risk trade-offs for ecological restoration in south africa // J. Environmental Management, 2013. Vol. 120. P. 138–147. doi: http://dx.doi.org/10.1016/j.jenvman.2013.02.001.

[4] *Shi T., Gill R.* Developing effective policies for the sustainable development of ecological agriculture in China: Case study of Jinshan County with a systems dynamics model // Ecological Economics, 2005. Vol. 53. No. 2. P. 223–246. doi: http://dx.doi.org/10.1016/j.ecolecon.2004.08.006.

*Работа выполнена при частичной финансовой поддержке РФФИ, проект №15-51-05011 Арм_а.

[5] *Zhang Z., Lu W. X., Zhao Y., Song W. B.* Development tendency analysis and evaluation of the water ecological carrying capacity in the siping area of Jilin Province in China based on system dynamics and analytic hierarchy process // Ecological Modelling, 2014. Vol. 275. P. 9–21. doi: http://dx.doi.org/10.1016/j.ecolmodel.2013.11.031.

[6] *Iwamura T., Lambin E. F., Silvius K. M., Luzar J. B., Fragoso J. M. V.* Agent-based modeling of hunting and subsistence agriculture on indigenous lands: Understanding interactions between social and ecological systems // Environmental Modelling Software, 2014. Vol. 58. P. 109–127. doi: http://dx.doi.org/10.1016/j.envsoft.2014.03.008.

[7] *Zitzler E., Thiele L.* Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach // IEEE Trans. Evolutionary Computation, 1999. Vol. 3. No. 4. P. 257–271. doi: http://dx.doi.org/10.1109/4235.797969.

[8] *Bleuer S., Brack M., Thiele L., Zitzler E.* Multiobjective genetic programming: Reducing bloat using spea2 // Congress on Evolutionary Computation Proceedings. — Seoul: IEEE, 2001. Vol. 1. P. 536–543.

[9] *Akopov A. S., Hevencev M. A.* A multi-agent genetic algorithm for multi-objective optimization // IEEE Conference (International) on Systems, Man and Cybernetics Proceedings. — Manchester: IEEE, 2013. P. 1391–1395.

[10] *Akopov A. S.* Parallel genetic algorithm with fading selection // Int. J. Comput. Appl. Technol., 2014. Vol. 49. No. 3/4. P. 325–331. doi: http://dx.doi.org/10.1504/IJCAT.2014.062368.

Поступила в редакцию 15.06.2016

О полных регрессионных решающих деревьях*

И. Е. Генрихов, Е. В. Дюкова, В. И. Журавлёв
ingvar1485@rambler.ru, edjukova@mail.ru, vadim091294@gmail.com
[1]ООО «Мобайл парк ИТ», г. Химки, ул. Панфилова, 21/1
[2]ФИЦ «Информатика и управление» РАН, г. Москва, ул. Вавилова, 44/2
[3]МГУ им. М. В. Ломоносова, г. Москва, Ленинские горы, 1

Рассматривается одна из центральных задач машинного обучения — задача восстановления регрессии. Предлагается качественно новая модель регрессионного решающего дерева (РРД), базирующаяся на понятии полного решающего дерева (ПРД). Ранее аналогичная конструкция решающего дерева (РД) была успешно апробирована на задаче классификации по прецедентам, которая по постановке близка к рассматриваемой задаче. Приведены результаты тестирования построенной модели полного РРД (ПРРД) на реальных данных.

Ключевые слова: *задача восстановления регрессии; регрессионные деревья; полное решающее дерево*

DOI: 10.21469/22233792.2.1.09

1 Введение

Одной из основных задач машинного обучения является задача обучения по прецедентам. Рассматривается следующая постановка этой задачи.

Исследуется множество объектов M. Объекты из M описываются системой признаков $\{x_1, \ldots, x_n\}$. Каждый объект S из M представим вектором длины n, в котором j-я координата равна значению признака x_j для объекта S. Задано некоторое числовое множество «ответов» Y и дана выборка объектов $T = \{S_1, ..., S_m\}$ из M такая, что для каждого объекта $S_i \in T$ известен «ответ» $y_i, y_i \in Y$. Объекты из T называются прецедентами или обучающими объектами. Требуется по выборке T построить алгоритм $A_T : M \to Y$, ставящий в соответствие каждому объекту S из M значение y из Y.

Актуальность рассматриваемой задачи заключается в том, что она возникает в целом ряде прикладных областей, таких как биология, геология, медицина, экономика, техника, банковская деятельность и др.

Выделяют два основных типа задач обучения по прецедентам.

1. Задача классификации (classification). В этом случае «ответ» y для объекта S из M называется меткой класса. Возможны следующие варианты:
   - $Y = \{-1; +1\}$ — классификация на 2 класса;
   - $Y = \{1, ..., N\}$ — классификация с N классами.
2. Задача восстановления регрессии (regression). В данном случае $Y = \mathbb{R}$ и «ответ» y для объекта S из M называется значением целевой переменной.

Одним из известных инструментов для решения задач обучения по прецедентам являются деревья решений.

Процедура построения классического РД представляет собой итерационный процесс. Как правило, для построения очередной вершины дерева выбирается признак, наилучшим

*Работа выполнена при финансовой поддержке РФФИ, проект № 16-01-00445.

образом удовлетворяющий некоторому критерию ветвления. По значениям этого признака и осуществляется ветвление, далее указанная процедура повторяется для каждой из ветвей. Однако если при построении дерева несколько признаков удовлетворяют критерию ветвления в равной или почти равной мере, то выбирается один из них (фактически случайным образом). При этом в зависимости от выбранного признака построенные деревья могут существенно отличаться как по составу используемых признаков, так и по своим распознающим качествам. Указанного недостатка лишена модель ПРД [1, 2]. В ПРД на каждой итерации строится так называемая полная вершина, которой соответствует набор признаков $\{x_{j_1}, \ldots, x_{j_q}\}, q \leqslant n$, где каждый признак удовлетворяет критерию ветвления. Затем для каждого признака $x_{j_i}, i \in \{1, ..., q\}$, строится «простая» внутренняя вершина, из которой осуществляется ветвление. По сравнению с классической конструкцией конструкция ПРД позволяет более полно использовать имеющуюся информацию, при этом описание распознаваемого объекта может порождаться не одной ветвью, как в классическом дереве, а несколькими ветвями. Каждая такая ветвь участвует в процедуре голосования (является голосующей).

Модель классического РД используется для решения обоих типов задач обучения по прецедентам. Модель ПРД разработана сравнительно недавно для решения задач классификации.

В настоящей работе рассматривается задача восстановления регрессии.

Одним из первых алгоритмов, использующих РРД, является алгоритм CART (classification and regression trees). Этот алгоритм строит бинарное РРД с критерием ветвления, основанным на вычислении статистик [3]. Похожую на CART конструкцию имеет алгоритм M5P. Алгоритм M5P, так же как и алгоритм CART, выполняет построение бинарных РД [4]. Более сложную конструкцию имеют алгоритмы классификации и восстановления регрессии Random Forest [4], REPTree [5] и Decision Stump [6]. Алгоритм Decision Stump базируется на построении k-арных РД, остальные алгоритмы строят бинарные РД.

Основной целью данной работы является построение и исследование ПРРД для задач с целочисленными данными.

В работе построены и протестированы алгоритмы NBRTree (nonbinary regression tree) и NBFRTree (nonbinary full regression tree), строящие k-арные регрессионные деревья, где k – максимальное число ребер, выходящих из простых вершин дерева. Алгоритм NBRTree строит классическое k-арное РРД. Алгоритм NBFRTree строит k-арное ПРРД. В обоих алгоритмах используется критерий ветвления, являющийся модификацией критерия ветвления алгоритма CART на случай k-арного дерева [3].

Проведено тестирование алгоритмов NBRTree и NBFRTree на реальных задачах. Показано, что на большинстве рассмотренных в работе задач алгоритм NBFRTree работает лучше других алгоритмов восстановления регрессии, участвовавших в тестировании, среди которых алгоритмы Random Forest, REPTree, M5P, CART, Decision Stump и NBRTree.

2 Основные понятия

Рассмотрим основные понятия, используемые при построении РРД, на примере бинарного РРД (БРРД).

Обозначим через $\breve{T}$ и $X(\breve{T}) \subseteq \{x_1, ..., x_n\}$ рассматриваемые на текущей итерации (шаге) построения РРД подмножество обучающих объектов и подмножество признаков соответственно.

На первом шаге $\breve{T} = T, X(\breve{T}) = \{x_1, ..., x_n\}$. На текущем шаге построения дерева для каждого признака x из $X(\breve{T})$ и каждого значения a признака x проводится разбиение $\breve{T}$

на две подвыборки (подвыборку $\breve{T}_R(x,a)$, для объектов которой выполняется неравенство $x \geqslant a$, и подвыборку $\breve{T}_L(x,a)$, для объектов которой выполняется неравенство $x < a$) и вычисляется оценка качества этого разбиения.

`Определение 1.` Оптимальным разбиением называется разбиение с наилучшей оценкой качества.

`Определение 2.` Признак удовлетворяет критерию ветвления, если оптимальное разбиение для этого признака имеет максимальную оценку качества разбиения среди оптимальных разбиений для всех других признаков.

Среди всех признаков, удовлетворяющих критерию ветвления, выбирается только один признак.

Различные алгоритмы БРРД отличаются критерием ветвления, а также правилом останова ветвления. Сложность построения БРРД очень велика при большом числе признаков, особенно если признаки многозначны.

На рис. 1 приведен пример ветвления из вершины x в БРРД. В этом дереве $x \in X(\breve{T}), a \in \{0,1,\ldots,k-1\}$, a — значение x.

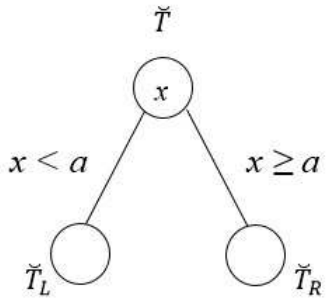


Рис. 1 Пример ветвления из x в БРРД

Алгоритм CART строит БРРД и при выборе оптимального разбиения использует статистический подход к оценке качества разбиения (критерий ветвления, основанный на вычислении статистик) [3]. Опишем этот критерий.

Пусть $\breve{T} = \{S_{i_1},\ldots,S_{i_u}\}, \breve{T}_R(x,a) = \{S_1^R,...,S_q^R\}, \breve{T}_L(x,a) = \{S_1^L,...,S_p^L\}$. При данном разбиении в правое и левое поддеревья попадает q и p объектов соответственно.

Пусть далее y_i^L и y_i^R — значения целевых переменных для объектов $S_i^L, i = 1,\ldots,q$, и $S_j^R, j = 1,...,p$, соответственно. Введем обозначения:

$$\bar{y}_{\breve{T}} = \frac{1}{u}\sum_{t=1}^{u} y_{i_t}\,;$$

$$V = \frac{1}{u}\sum_{t=1}^{u}(y_{i_t}^2) - \Big[\frac{1}{u}\sum_{t=1}^{u}(y_{i_t})\Big]^2\,;$$

$$\mathrm{SE}\,(x) = \frac{1}{u}\Big\{\sum_{i=1}^{p}(y_i^L - \bar{y}_{\breve{T}_L})^2 + \sum_{j=1}^{q}(y_j^R - \bar{y}_{\breve{T}_R})^2\Big\}\,;$$

$$C(x) = V - \mathrm{SE}\,(x)$$

Оптимальным считается разбиение с максимальным значением величины $C(x)$.

Построение очередной ветви в алгоритме CART прекращается, если величина $C(x)$ не превосходит наперед заданного числа δ.

Похожую на CART конструкцию имеет алгоритм M5P. Алгоритм M5P, так же как и алгоритм CART, выполняет построение бинарных РД, но использует энтропийный критерий ветвления [7].

Более сложную конструкцию имеют алгоритмы классификации и восстановления регрессии Random Forest [4], REPTree [5] и Decision Stump [6]. Алгоритм Decision Stump базируется на построении k-арных РД, остальные алгоритмы строят бинарные РД. Среди перечисленных алгоритмов наиболее используемым является алгоритм Random Forest.

Random Forest — алгоритм машинного обучения, заключающийся в использовании комитета (ансамбля) РД (предложен Л. Брейманом и А. Катлер в 2001 г.). В алгоритме Random Forest используется энтропийный критерий ветвления и процедура «бэггинг». Процедура бэггинга над РД заключается в использовании композиции РД, каждое из которых строится независимо. Для построения очередного дерева композиции случайным образом выбирается (с возвращением) некоторое подмножество обучающих объектов из исходной выборки. Результат распознавания определяется путем усреднения значений целевой переменной по всем построенным РД. Таким образом, деревья компенсируют ошибки друг друга.

3 Алгоритмы восстановления регрессии NBRTree и NBFRTree

3.1 Построение алгоритмов NBRTree и NBFRTree

Алгоритм NBRTree строит классическое РРД. Главная особенность алгоритма NBRTree — это его k-арная структура. Ветвление по выбранному признаку x разбивает обучающие объекты на k подвыборок, где k — число различных значений признака.

Рассмотрим более подробно схему ветвления из вершины $x, x \in X(\breve{T})$, в алгоритме NBRTree.

Не ограничивая общности, будем считать, что признак x имеет значения из $\{0, 1, \ldots, k-1\}, k \geqslant 2$. В этом случае при построении дерева решений из вершины x выходят k дуг, помеченные числами из $\{0, 1, \ldots, k-1\}$. Пусть σ — метка одной из дуг, выходящих из вершины $x, \sigma \in \{0, 1, \ldots, k-1\}$. Для формирования нового текущего подмножества объектов и нового текущего множества признаков удаляются те объекты из $\breve{T}$, для которых значение признака x не равно σ, а также из множества признаков удаляется сам признак x. Для улучшения качества распознавания при построении ветви используется правило останова, которое описано в конце данного параграфа.

Положим

$$x^{\sigma} = \begin{cases} 1, & \text{если } x = \sigma\,; \\ 0, & \text{если } x \neq \sigma. \end{cases}$$

Пусть v — висячая вершина, порожденная ветвью дерева с внутренними вершинами $x_{j_1}, \ldots, x_{j_r}$ и пусть дуга, выходящая из вершины $x_{j_i}, i \in \{1, \ldots, r\}$, имеет метку σ. Пусть далее $\breve{T}(v)$ — текущее множество объектов, которые попали в вершину v. Вершине v ставится в соответствие пара $(B, w(v))$, где $w(v)$ равно среднему значению целевой переменной по всем объектам из $\breve{T}(v)$, а B — элементарная конъюнкция вида $x_{j_1}^{\sigma_1}, \ldots, x_{j_r}^{\sigma_r}$. Интервал истинности элементарной конъюнкции B обозначим через N_B.

Пусть S — распознаваемый объект. Для каждой висячей вершины $(B, w(v))$ выполняется проверка принадлежности описания распознаваемого объекта S интервалу истинности N_B. Если описание S принадлежит N_B, то объекту S ставим в соответствие значение

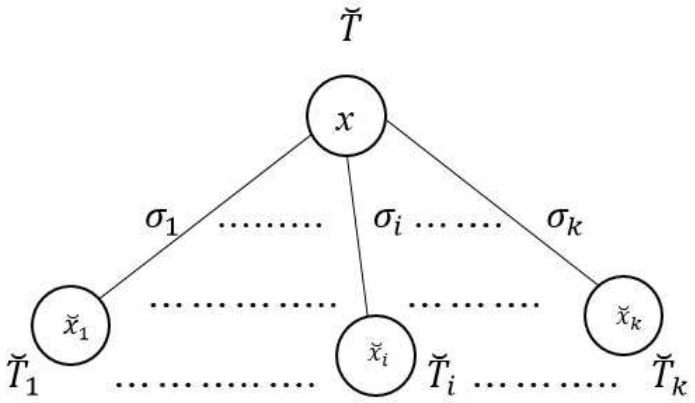


Рис. 2 Ветвление из вершины x в алгоритме NBRTree

целевой переменной $w(v)$. По построению описание может попасть только в одну из висячих вершин.

На рис. 2 показано ветвление из вершины x в алгоритме NBRTree для $\breve{T} = \breve{T}_1 \cup \breve{T}_2 \cup ... \cup \breve{T}_k$, где k — множество различных значений признака x.

В алгоритме NBFRTree используется идея ПРД, т. е. при возникновении ситуации, когда два или более признака удовлетворяют критерию ветвления в равной или почти равной мере, то ветвление проводится по каждому из этих признаков независимо.

Процедура распознавания объекта выполняется следующим образом. Пусть $V = v_1, \ldots, v_l$ — множество висячих вершин построенного дерева с соответствующими парами $(B_i, w(v_i)), i = 1, 2, ..., l, l \geqslant 1$. Для каждой висячей вершины v_i осуществляется проверка принадлежности описания объекта S интервалу истинности N_{B_i}.

Положим

$$I_{B_i} = \begin{cases} 1, & \text{если описание объекта } S \in N_{B_i}; \\ 0 & \text{в противном случае.} \end{cases}$$

Объекту S ставится в соответствие значение целевой переменной

$$W = \frac{\sum_{i=1}^{l} w(v_i) I_{B_i}}{\sum_{i=1}^{l} I_{B_i}}.$$

На рис. 3 показано ветвление из полной вершины $\{x_{j_1}, \ldots, x_{j_r}\}$ в алгоритме NBFRTree. Ветвление из простых вершин $x_{j_1}, ..., x_{j_r}$ производится как в алгоритме NBRTree.

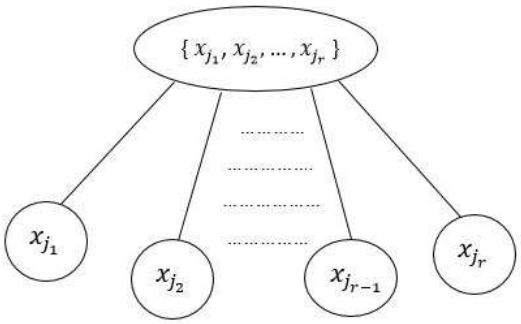


Рис. 3 Ветвление из полной вершины $\{x_{j_1}, \ldots, x_{j_r}\}$ в алгоритме NBFRTree

Опишем критерий ветвления, используемый в алгоритмах NBRTree и NBFRTree.

Пусть $\breve{T}_i = S^i_1, \ldots, S^i_{u_i}, y^i_t$ — значение целевой переменной обучающего объекта $S^i_t, t \in \{1, 2, \ldots, u_i\}$, и пусть рассматриваемый признак x принимает k значений. Обучающая выборка $\breve{T}_i$ разбивается по этому признаку на k подвыборок $\breve{T}_{i_1}, \ldots, \breve{T}_{i_k}$. Вычисляются величины

$$\mathrm{SE}\,(x,k) = \frac{1}{u_i}\Big\{ \sum_{S^i_t \in \breve{T}_{i_1}} (y^i_t - \bar{y}_{\breve{T}_{i_1}})^2 + \cdots + \sum_{S^i_t \in \breve{T}_{i_k}} (y^i_t - \bar{y}_{\breve{T}_{i_k}})^2 \Big\};$$

$$C(k,x) = V - \mathrm{SE}\,(x,k)\ .$$

При $k = 2$ описанный критерий совпадает с критерием ветвления алгоритма CART (см. разд. 2).

В алгоритме NBRTree для ветвления выбирается один признак, для которого $C(k,x) = \max\limits_{x \in X(\breve{T})} C(k,x)$.

В алгоритме NBFRTree признаки для ветвления выбираются иначе. Пусть $C_{\min} = \min\limits_{x \in X(\breve{T})} C(k,x)$ и $C_{\max} = \max\limits_{x \in X(\breve{T})} C(k,x)$. Сначала для каждого признака $x \in X(\breve{T}_i)$ вычисляется величина $C(k,x) = V - \mathrm{SE}\,(x,k)$. Далее значение $C(k,x)$ нормируется и вычисляется

$$C^*(k,x) = \frac{C(k,x) - C_{\min}}{C_{\max} - C_{\min}}\,.$$

Для построения полной вершины выбираются те признаки из $X(\breve{T}_i)$, для которых $0{,}75 \leqslant C^*(k,x) \leqslant 1$. В случае, когда $C_{\max} = C_{\min}$, разбиение производится по всем признакам из $X(\breve{T}_i)$.

Построение ветви прекращается, если разность между минимальной и максимальной целевыми переменными в данной вершине не превосходит наперед заданного ε (параметр останова).

4 Тестирование алгоритмов NBRTree и NBFRTree

Алгоритмы были протестированы на 18 реальных задачах из ресурса UCI [8]. Список задач, на которых производилось тестирование алгоритмов: Data1 — Servo; Data2 — Computer Hardware; Data3 — Yacht Hydrodynamics; Data4 — Concrete Slump Test; Data5 — Fertility; Data6 — Breast Cancer Wisconsin breast-cancer-wisconsin; Data7 — Concrete Compressive Strength; Data8 — Housing; Data9 — Airfoil Self-Noise; Data10 — Combined Cycle Power Plant; Data11 — Forest Fires; Data12 — White Wine Quality; Data13 — Red Wine Quality; Data14 — Student Performance; Data15 — Geographical Original of Music Data Set Geographical Original of Music Data Set latitude; Data16 — Geographical Original of Music Data Set longitude; Data17 — Breast Cancer Wisconsin wdbc; Data18 — Breast Cancer Wisconsin wpbc.

В задачах, в которых присутствовали признаки, принимающие вещественнозначные значения, была применена процедура перекодирования вещественнозначных значений признака в целочисленные. Производилась она следующим образом.

Пусть $\{c_1, \ldots, c_u\}$ – множество различных значений признака $x, c_{i+1} \geqslant c_i, 1 \leqslant i \leqslant u-1$. Выбирается t порогов, $5 \leqslant t \leqslant 10$, для признака x, делящих обучающую выборку по этому признаку на t равных частей.

Таблица 1 Оптимальное значение ε

Data	ε	Data	ε
Data1	0,6	Data10	0,5
Data2	0	Data11	2
Data3	0,1	Data12	1
Data4	0,5	Data13	1
Data5	0,7	Data14	17
Data6	1,1	Data15	60
Data7	0,5	Data16	170
Data8	0,1	Data17	0
Data9	0	Data18	0

Значение параметра останова ε для каждой задачи определялось эмпирически. Для разных значений ε производилась кросс-валидация. В результате выбиралось то значение ε, при котором достигался наилучший результат алгоритма. В табл. 1 приведено оптимальное значение ε для каждой из рассмотренных задач.

Для оценки качества работы алгоритмов была применена кросс-валидация по k фолдам. Исходные данные разбивались на k подвыборок, $k \geqslant 2$. Затем на $k-1$ подвыборке производилось обучение алгоритма, а оставшаяся подвыборка использовалась для тестирования. Процедура повторялась k раз. В итоге каждая из k подвыборок использовалась для тестирования.

Для оценки эффективности алгоритмов использовались величины MAE (Mean Absolute Error — средняя абсолютная ошибка) и RMSE (Root Mean Squared Error — корень среднеквадратичной ошибки), вычисляемые соответственно следующим образом:

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}|y_i - h_i|\,; \qquad \mathrm{RMSE} = \frac{1}{\sqrt{m}}\sqrt{\sum_{i=1}^{m}(y_i - h_i)^2},$$

где y_i — значения целевых переменных, а h_i — значения, выданные алгоритмом.

Алгоритмы NBRTree и NBFRTree сравнивались с алгоритмами CART и Random Forest из библиотеки sklearn языка Python, а также с алгоритмами Decision Stump, M5P и REPTree из свободного программного обеспечения для анализа данных WEKA.

Если число объектов в выборке не превышало 350, использовалась кросс-валидация Leave One Out. Для выборок, в которых больше 350 объектов, применялась кросс-валидация по 10 фолдам. Для большей надежности эксперимента кросс-валидация по 10 фолдам производилась 10 раз, после каждой итерации выборка перемешивалась.

В табл. 2–5 приведены результаты тестирования. В табл. 2 и 3 приведены результаты тестирования по методу Leave One Out на шести реальных задачах. В табл. 4 и 5 приведены результаты кросс-валидации (10 раз по 10 фолдам) на 12 задачах.

Из табл. 2–5 видно, что на 11 из 18 реальных задач с функционалом качества MAE наилучшие результаты показал алгоритм NBFRTree, на 4-х — Random forest, на двух — CART и на одной — NBRTree.

На 7 из 18 реальных задач с функционалом качества RMSE наилучшие результаты показал алгоритм NBFRTree, на 7 — Random Forest, на одной — алгоритмы CART, Desicion Stump, NBRTree и M5P.

Таблица 2 Качество работы оценивается функционалом качества MAE

Задачи	Размер $m \times n$	NBRTree	NBFRTree	Decision Stump	M5P	REPTree	CART	Random Forest
Data1	167×4	0,277	0,277	0,645	0,500	0,356	**0,181**	0,219
Data2	209×5	8,391	**7,313**	15,311	14,162	13,306	8,352	8,220
Data3	308×6	0,886	0,662	4,940	2,277	0,800	0,672	**0,511**
Data4	103×7	3,476	3,392	6,013	4,751	4,029	3,313	**2,902**
Data5	100×10	0,150	**0,120**	0,199	0,213	0,219	0,265	0,215
Data6	198×33	0,354	**0,237**	0,336	0,339	0,329	0,298	0,248

Таблица 3 Качество работы оценивается функционалом качества RMSE

Задачи	Размер $m \times n$	NBRTree	NBFRTree	Decision Stump	M5P	REPTree	CART	Random Forest
Data1	167×4	0,505	0,511	1,013	0,840	0,750	**0,402**	0,448
Data2	209×5	22,254	**16,563**	25,770	23,407	26,483	21,480	18,378
Data3	308×6	1,417	**0,877**	7,240	4,218	1,567	1,521	1,060
Data4	103×7	5,160	4,795	7,633	6,108	5,384	4,823	**4,160**
Data5	100×10	0,387	0,346	**0,318**	0,328	0,347	0,512	0,369
Data6	198×33	0,595	0,487	0,421	**0,411**	0,417	0,546	0,498

Таблица 4 Качество работы оценивается функционалом качества MAE

Задачи	Размер $m \times n$	NBRTree	NBFRTree	Decision Stump	M5P	REPTree	CART	Random Forest
Data7	1030×7	4,932	4,672	11,572	6,876	5,613	**4,489**	5,731
Data8	506×13	3,492	**3,106**	5,203	3,607	3,415	3,467	3,857
Data9	1503×5	2,651	**2,647**	5,018	3,318	2,753	2,670	3,404
Data10	9568×4	**3,710**	3,786	7,494	3,871	3,746	3,718	3,723
Data11	517×7	18,597	**18,563**	19,342	18,653	18,626	27,151	30,065
Data12	4898×11	0,490	**0,433**	0,671	0,582	0,563	0,499	0,467
Data13	1599×11	0,436	**0,396**	0,560	0,523	0,510	0,463	0,440
Data14	649×30	2,157	**2,073**	2,201	2,137	2,132	2,783	2,091
Data15	1059×68	16,844	13,895	13,989	14,246	13,929	15,932	**12,768**
Data16	1059×68	42,901	37,466	40,074	37,788	38,996	44,816	**34,166**
Data17	699×9	0,136	**0,113**	0,282	0,244	0,163	0,123	0,125
Data18	569×30	0,076	**0,065**	0,182	0,148	0,105	0,073	0,074

Наилучшие результаты показали алгоритмы NBFRTree и Random Forest. На задачах, на которых наилучшие результаты показал NBFRTree, алгоритм NBFRTree оказался лучше в среднем на 23% (функционал MAE) и на 22% (функционал RMSE) по сравнению с алгоритмом Random Forest. На задачах, на которых наилучшие результаты показал Random Forest, алгоритм NBFRTree оказался хуже в среднем на 16% (функционал MAE) и на 15% (функционал RMSE) по сравнению с алгоритмом Random Forest.

5 Заключение

Разработаны новые алгоритмы для задачи восстановления регрессии, основанные на построении РД (алгоритмы NBRTree и NBFRTree). В обоих алгоритмах используется критерий ветвления, который является модификацией критерия, используемого в хорошо из-

Таблица 5 Качество работы оценивается функционалом качества RMSE

Задачи	Размер $m \times n$	NBRTree	NBFRTree	Decision Stump	M5P	REPTree	CART	Random Forest
Data7	1030×7	7,334	**6,24**	14,508	8,755	7,454	6,698	7,484
Data8	506×13	5,390	**4,664**	6,949	5,216	5,113	5,355	5,205
Data9	1503×5	3,394	**3,369**	6,341	4,210	3,520	3,403	4,252
Data10	9568×4	**4,812**	4,876	9,135	4,966	4,855	4,817	4,838
Data11	517×7	45,738	**45,681**	64,018	63,825	64,470	86,587	77,133
Data12	4898×11	0,852	0,766	0,813	0,745	0,747	0,869	**0,668**
Data13	1599×11	0,778	0,665	0,734	0,670	0,681	0,787	**0,616**
Data14	649×30	2,965	**2,819**	2,908	2,900	2,933	3,794	2,833
Data15	1059×68	23,237	17,435	17,451	17,645	17,675	23,290	**16,766**
Data16	1059×68	54,920	47,427	50,263	47,948	50,844	61,821	**44,370**
Data17	699×9	0,479	0,446	0,549	0,421	0,449	0,484	**0,358**
Data18	569×30	0,272	0,242	0,325	0,236	0,244	0,272	**0,188**

вестном алгоритме CART, на случай k-арного дерева. Алгоритм NBRTree строит классическое k-арное дерево, в котором ветвление проводится только по одному признаку.

Алгоритм NBFRTree строит k-арное ПРРД, в котором ветвление на каждом шаге синтеза проводится по всем признакам, удовлетворяющим критерию ветвления в равной или почти равной мере. Проведено сравнение алгоритмов NBRTree и NBFRTree с алгоритмами CART и Decision Stump, M5P, а также с алгоритмами, имеющими более сложную конструкцию: Random Forest и REPTree. Тестирование проводилось на 18 реальных задачах из ресурса UCI. Эффективность алгоритмов оценивалась стандартными функционалами качества MAE и RMSE. Показано, что качество алгоритма NBFRTree выше качества алгоритмов NBRTree, Decision Stump, REPTree, M5P и CART и не уступает качеству алгоритма Random Forest, а в некоторых случаях показывает и лучшие результаты.

Таким образом, исследованный в данной работе подход к синтезу РД для решения задачи восстановления регрессии — построение ППРД — может быть успешно использован наравне с другими известными подходами к синтезу РД.

Литература

[1] *Djukova E. V., Peskov N. V.* A classification algorithm based on the complete decision tree // J. Pattern Recogn. Image Anal., 2007. Vol. 17. P. 363–367.

[2] *Генрихов И. Е., Дюкова Е. В.* Классификация на основе полных решающих деревьев // Ж. вычисл. мат. мат. физ., 2012. Т. 52. № 4. С. 750–761.

[3] *Breiman L., Friedman J. H., Olshen R. A., Stone C. J.* Classification and regression trees. — CRC Press, 1984. 368 p.

[4] *Breiman L.* Random forests // Machine Learning, 2001. Vol. 45. Iss. 1. P. 5–32.

[5] *Elomaa T., Kääriäinen M.* An analysis of reduced error pruning // J. Artificial Intelligence Res., 2001. Vol. 15. P. 163–187.

[6] *Ai W. I., Langley P.* Induction of one-level decision trees // 9th Conference (International) on Machine Learning Proceedings. — Morgan Kaufmann, 1992. P. 233–240.

[7] *Quinlan J. R.* Learning with continuous classes // Australian Joint Conference on Artificial Intellegence Proceedings. — World Scientific, 1992. P. 343–348.

[8] *Lichman M.* UCI machine learning repository, 2013. `http://archive.ics.uci.edu/ml`.

Поступила в редакцию 18.06.2016

About full regression decision trees*

I. E. Genrikhov, E. V. Djukova, and V. I. Zhuravlyov
ingvar1485@rambler.ru, edjukova@mail.ru, vadim091294@gmail.com
[1]LLC Mobile park IT, 21/1 Panfilova Str., Khimki, Moscow region, Russia
[2]FRC "Computer Science and Control"of RAS, 44/2 Vavilova Str., Moscow, Russia
[3]Lomonosov Moscow State University, 1 Leninskie Gory, Moscow, Russia

Background: The regression restoration problem is considered. The approach based on the construction of regression trees is highlighted among the existing approaches. The most known among algorithms of regression trees synthesis (e. g., algorithms CART and Random Forest) are based on use of the elementary trees, namely, binary regression trees. Rarely, k-ary regression trees are used. Only one feature which is meeting the selected criteria of branching is selected in the synthesis of such trees, and the branching is carried out using this feature. However, only one feature is chosen (randomly) in case when several features are equally or almost equally meeting the selected criteria in construction of regression trees. Thus, the constructed trees depending on the selected feature can significantly vary both on structures of the used features and on its recognition qualities.

Methods: A new approach to the construction of regression trees based on the construction of the so-called full decision tree is applied. Originally, the approach to the synthesis of full decision trees was investigated only on the precedents classification problems and presented improved quality in comparison with the known methods of synthesis of decision trees. So-called full node is built on each iteration in the full decision tree. A set of features corresponds to the full node, and each feature meets the selected branching criterion. Then, the simple internal node from which the branching is carried out is built for each feature of this set. In comparison with the classical construction, the full decision tree allows to use more fully the available information. Herewith, the description of the recognizable object may be generated not by only one branch, as in a classical tree, but by several branches.

Results: Two synthesis algorithms of regression trees — NBFRTree and NBRTree — are developed. The NBRTree algorithm builds classic k-ary regression tree using a statistical criterion selection feature. The NBFRTree algorithm is an improvement of the NBRTree with the approach to the full decision trees synthesis — at each step, a set of features, which are equally or nearly equally meet a statistical criterion, on which branching is carried out is selected. It is shown that the best results were received when the NBFRTree algorithm was used. A comparison of 18 real problems of NBFRTree and NBRTree algorithms with known regression trees synthesis algorithms, such as the Random Forest, Decision Stump, REPTree, and CART, is carried out. It is shown that the quality of the NBFRTree algorithm is higher than the quality of the Decision Stump, REPTree, and CART algorithms, and it does not concede on quality to the Random Forest algorithm and, in some cases, also shows the best results.

Concluding Remarks: It is shown that the applied in this work approach to the regression trees synthesis for the solving of the regression restoration problem — full regression trees — can be successfully used on an equal basis with other known approaches to regression trees synthesis.

Keywords: *regression restoration problem; regression trees; full decision tree*

DOI: 10.21469/22233792.2.1.09

*The research was supported by the Russian Foundation for Basic Research (grant 16-01-00445.).

References

[1] Djukova, E. V., and N. V. Peskov. 2007. A classification algorithm based on the complete decision tree. *J. Pattern Recogn. Image Anal.* 17:363–367.

[2] Genrikhov, I. E., and E. V. Djukova. 2012. Klassifikatsiya na osnove polnykh reshayushchikh derev'ev [Classification based on full decision trees]. *Zh. Vychisl. Matem. Matem. Fiz.* [J. Comput. Math. Math. Phys.] 52(4):653–663.

[3] Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and regression trees.* CRC Press. 368 p.

[4] Breiman, L. 2001. Random forests. *Machine Learning* 45(1):5–32.

[5] Elomaa, T., and M. Kääriäinen. 2001. An analysis of reduced error pruning. *J. Artificial Intelligence Res.* 15:163–187.

[6] Ai, W. I., and P. Langley. 1992. Induction of one-level decision trees. *9th Conference (International) on Machine Learning Proceedings.* Morgan Kaufmann. 233–240.

[7] Quinlan, J. R. 1992. Learning with continuous classes. *Australian Joint Conference on Artificial Intellegence Proceedings.* World Scientific. 343–348.

[8] Lichman, M. 2013. *UCI machine learning repository.* Available at: `http://archive.ics.uci.edu/ml` (accessed August 14, 2016).

Received June 18, 2016