

Научная статья

УДК 519.872

<https://doi.org/10.31854/1813-324X-2025-11-2-84-100>

EDN:SIYZKF



Модель представления решающих функций для метрических методов распознавания образов

Игорь Васильевич Александров ✉ aleksandrov.iv@sut.ru

Роман Михайлович Вивчарь, vivchar.rm@sut.ru

Руслан Валентинович Киричек, kirichek@sut.ru

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация

Аннотация

В настоящее время ключевую роль поддержки принятия решений в различных отраслях играет интеллектуальный анализ данных. Его важной составной частью является машинное обучение, **актуальная задача** которого на практике – осуществление классификации объектов в режиме реального времени. Она может быть достигнута за счет распараллеливания алгоритмов обработки данных как по входным данным, так и по данным решающих функций. Для повышения эффективности распараллеливания методов машинного обучения разработана унифицированная модель представления решающих функций. **Целью** настоящей статьи является представление унифицированной модели решающих функций для алгоритмов машинного обучения и функций, обеспечивающих ее распараллеливание как по входным данным, так и по данным решающих функций.

Сущность представленного подхода заключается в том, что на основе анализа особенностей работы метрических методов машинного обучения выявлены независимые данные для обработки, представленные в виде разных категорий анализируемого свойства, разработана модель представления решающих функций, описывающая выявленные характерные черты объектов входных данных и данных решающих функций в виде соответствующих наборов унифицированных элементов и включающая в себя функции, обеспечивающие их параллельную обработку на основе группового распараллеливания объектов. Предложенный подход базируется на использовании методов анализа алгоритмов и вычислительной сложности, математической статистики и методологии проектирования параллельных алгоритмов.

Эксперименты показали, что распараллеливание предложенной модели решающих функций на примере метода потенциальных функций позволяет повысить эффективность классификации как для одного объекта за счет возможности использования дополнительных вычислительных ресурсов, так и для группы объектов при наличии ограничений объема памяти компьютеров или горизонта планирования.

Научная новизна предложенного подхода заключается в том, что модель отличается от существующих единым способом формализации объектов и их свойств в виде унифицированных элементов для обучающих и классификационных данных и имеет структуру и функции, ориентированные на ее параллельную обработку методами распознавания образов на основе решающих функций в рамках группового распараллеливания объектов.

Теоретическая значимость: модель имеет унифицированный характер и может использоваться при распараллеливании других методов распознавания образов, которые могут быть описаны сходными параметрами, архитектурой и классификационными признаками.

Практическая значимость предложенного подхода заключается в том, что модель позволяет осуществить декомпозицию задачи классификации образов на отдельные подзадачи поиска закономерностей между входными данными и данными решающих функций.

Ключевые слова: распознавание образов, метод решающих функций, машинное обучение

Ссылка для цитирования: Александров И.В., Вивчарь Р.М., Киричек Р.В. Модель представления решающих функций для метрических методов распознавания образов // Труды учебных заведений связи. 2025. Т. 11. № 2. С. 84–100. DOI:10.31854/1813-324X-2025-11-2-84-100. EDN:SIYZKF

Original article
<https://doi.org/10.31854/1813-324X-2025-11-2-84-100>
EDN:SIYZKF

Decision Functions Model for Metric Methods of Pattern Recognition

 **Igor V. Alexandrov** ✉, aleksandrov.iv@sut.ru
 **Roman M. Vivchar**, vivchar.rm@sut.ru
 **Ruslan V. Kirichек**, kirichек@sut.ru

The Bonch-Bruевич Saint Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

Annotation

Currently, data mining based on machine learning plays a key role in decision support in various industries. An important practical problem of machine learning is the implementation of object classification in real time, which can be achieved by parallelizing data processing algorithms both for input data and for decision function data. To improve the efficiency of parallelizing machine learning methods, a unified decision function model has been developed. **The Relevance** of this research is to present a unified decision function model in the framework of machine learning algorithms and functions for its parallelization both in terms of input data and decision function data.

The essence of the presented approach is that the features of the operation of metric methods of machine learning are analyzed, independent data for processing are presented using different categories of the analyzed property, developed decision function model describes the object features for input data and decision function data using standardized elements and including functions for their parallel processing based on group parallelization of objects. The proposed approach is based on the use of methods for analyzing algorithms and computational complexity, mathematical statistics and the methodology of designing parallel algorithms.

Experiments have shown that parallelization of the proposed decision function model for the potential function method allows increasing the classification efficiency for one object using additional computing resources, and for a group of objects within the limits of the computer's memory size or planning horizon.

The novelty of the proposed approach is that the model differs from existing ones in a method of formalizing objects and their features using unified elements for training and classification objects and has a structure and functions oriented towards its parallel processing by pattern recognition methods based on decision functions within the framework of group parallelization of objects.

Theoretical significance: the model is unified and can be used to parallelize other pattern recognition methods that can be described by similar parameters, architecture, and classification features.

The practical significance of the proposed approach is that the model allows decomposing the pattern classification problem into separate subtasks of finding regularities between input data and decision function data.

Keywords: pattern recognition, decision function method, machine learning

For citation: Aleksandrov I.V., Vivchar R.M., Kirichек R.V. Decision Functions Model for Metric Methods of Pattern Recognition. *Proceedings of Telecommunication Universities*. 2025;11(2):84–100. (in Russ.) DOI:10.31854/1813-324X-2025-11-2-84-100. EDN:SIYZKF

Введение

Наиболее перспективным направлением информационных технологий, используемым для организации поддержки принятия решений [1, 2] в таких областях, как экономика, образование, медицина и телекоммуникации, является интеллектуальный анализ данных [3, 4]. Одним из актуальных направлений исследований в области интел-

лектуального анализа данных являются вычислительные системы машинного обучения [5, 6]. Для решения задач машинного обучения применяется группа методов решающих функций [7, 8], основанных на алгоритмах метрической классификации и оценивания сходства объектов. Наличие обучающих данных является обязательным для использования методов решающих функций, по-

этому эти методы также называют алгоритмами, основанными на прецедентах [9]. Главным достоинством методов решающих функций является их высокая эффективность с точки зрения достоверности результата, а главным недостатком – вычислительноемкие алгоритмы обработки данных. Особенно это актуально при работе с большими объемами данных.

В последнее десятилетие повышение производительности вычислительных узлов связано с развитием многоядерных процессоров и сети Интернет, что привело к распространению облачных систем. Однако современное программное обеспечение для решения задач машинного обучения значительно отстает от аппаратной части компьютерной техники и часто неэффективно использует предоставляемые вычислительные ресурсы [10]. Данная ситуация связана со следующими проблемами:

- 1) трудоемкость задачи распараллеливания вычислительных алгоритмов;
- 2) разнообразие типов данных решающих функций;
- 3) необходимость получения информации в режиме реального времени.

Способом решения рассмотренных проблем является разработка унифицированной модели представления решающих функций для метрических методов распознавания образов. В работах [11–13] представлена модель знаний для алгоритмов интеллектуального анализа данных, обеспечивающая распараллеливание вычислений только по входным данным. Предлагаемая модель отличается от вышеуказанной возможностью распараллеливания вычислений не только по входным данным, но и по данным решающих функций, что обеспечивает повышение эффективности классификации объекта в режиме реального времени.

Формальное описание данных

Входными данными для классификации образов являются множества однородных объектов $X = \{x_1, \dots, x_s\}$ – событий, явлений, сделок и др. – и меток их классов $Y = \{y_1, \dots, y_s\}$ – категорий, на которые делятся объекты. Каждый анализируемый объект описывается в виде набора из n атрибутов (длина, вес, возраст и др.). На практике наиболее часто используемой формой представления объекта является вещественный вектор числовых значений его атрибутов. Поэтому в работе ограничимся следующим способом представления объекта x :

$$x = (v_1^{Object}, \dots, v_i^{Object}, \dots, v_n^{Object}), \quad (1)$$

где $v_i^{Object} \in R$ – числовое значение i -го атрибута объекта x .

Набор допустимых классов U объекта x определим как конечное множество неповторяющихся меток из m элементов:

$$U = (u_1, \dots, u_m), \quad (2)$$

$$u_k \cap u_j = \emptyset \forall k \neq j (k, j \in \{1..m\}),$$

где u_j – j -я метка из множества U .

Процесс определения класса u_j объекта x_i происходит на основе анализа его черт $\{q_1, \dots, q_w\}$, отражающих необходимые свойства данных для отделения объектов одного класса от другого в рамках применяемого метода распознавания образов. Проанализировав особенности работы рассматриваемых метрических методов распознавания образов, следует отметить, что каждый искомый параметр для объекта x_i также можно описать в виде вещественного вектора числовых значений анализируемых атрибутов:

$$q = (v_1^{Property}, \dots, v_i^{Property}, \dots, v_m^{Property}), \quad (3)$$

где $v_i^{Property} \in R$ – числовое значение i -го анализируемого атрибута особенности данных.

Дополнительно определим следующие технические и алгоритмические элементы данных.

1) Технический элемент *null*, обозначающий отсутствие какой-либо информации для рассматриваемого объекта данных.

2) Технический элемент *id*, обозначающий идентификационный номер объекта или класса в рамках всего множества данных. Например, идентификационным номером класса u является его порядковый номер id_u во множестве U .

3) Параметр p определяет особенность работы метода распознавания (количество искомых объектов, число итерации и др.) и представляет собой следующий кортеж:

$$p = \langle n_{Parameter}, v_{Parameter} \rangle, \quad (4)$$

где $n_{Parameter}$ – имя параметра; $v_{Parameter}$ – значение параметра.

4) Множество параметров:

$$P = \{p_1, \dots, p_g\}, \quad (5)$$

где p_i – i -й параметр работы алгоритма.

Когда множество параметров $\{p_1, \dots, p_g\}$ фиксировано и однозначно определено, множество P может быть упрощено до списка значений его параметров $P \equiv \{v_1^{Parameter}, \dots, v_g^{Parameter}\}$. В дальнейшем под множеством параметров P будет пониматься только набор их числовых значений.

Рассматриваемая задача распознавания образов состоит из следующих двух этапов [14].

Этап 1. Обучение

В рамках этого этапа определяются необходимые свойства данных для наборов обучающих объектов в рамках соответствующих решающих функций. Решающие функции могут содержать такие особенности обучающих данных, как усред-

ненный эталон для набора объектов одного класса, гиперплоскости, разделяющие объекты одного класса от других, подмножества объектов обучающей выборки.

Этап 2. Классификация

Целью этого этапа является нахождение искомых параметров тестового объекта x_i^{Test} для определения его класса y_i^{Test} из множества U на основе данных решающих функции. В качестве искомых параметров тестового объекта могут выступать такие параметры данных, как расстояние от тестового объекта до усредненного эталона группы объектов i -го класса, K ближайших соседей в обучающей выборке и др.

Обобщим способы представления объекта x , его класса u и черты q (1–3) для вышеуказанных этапов работы рассматриваемых методов распознавания образов в виде понятия унифицированного факта следующим образом:

$$w = \{id_{Fact}, y_{Fact}, V_{Fact}^{Data}, V_{Fact}^{Category}\}, \quad (6)$$

где id_{Fact} – идентификационный номер факта w в рамках множества объектов анализируемой характерной черты данных; y_{Fact} – класс описываемых фактом w данных; V_{Fact}^{Data} – описываемые фактом w данные (объект, разделяющая плоскость, расстояние, счетчик объектов и др.); $V_{Fact}^{Category}$ – набор соответствующих категорий для всех анализируемых свойств данных в рамках применяемого метода распознавания образов (более подробно будут рассмотрены в следующем разделе).

Обобщенная структура факта w реализуется на основе возможности назначения любому его аргументу вышеуказанного технического элемента *null*.

На основе сформулированного понятия факта (6) определим основные элементы данных.

1) Представим неделимый набор фактов в виде понятия параметра данных следующим образом:

$$k = \{w_1, \dots, w_d\}, \quad (7)$$

где w_j – j -й факт параметра k .

2) Независимое i -е множество объектов для анализа:

$$X_i = \{x_{i,1}, \dots, x_{i,a}\}, \quad (8)$$

где $x_{i,j}$ – j -й объект множества X_i .

3) Набор соответствующих классов для X_i :

$$K_i^{Class} = \{k_{i,1}^{Class}, \dots, k_{i,a}^{Class}\}, \quad (9)$$

где $k_{i,j}^{Class} = \{w_{i,j,1}^{Class}\}$ – класс j -го объекта $x_{i,j}$ множества X_i .

4) Набор выявленных однотипных черт для X_i :

$$K_i^{Property} = \{k_{i,1}^{Property}, \dots, k_{i,b}^{Property}\}, \quad (10)$$

где $k_{i,j}^{Property} = \{w_{i,j,1}^{Property}\}$ – j -я черта множества X_i .

Количество черт в $K_i^{Property}$ в обобщенном виде может не совпадать с числом объектов в X_i . Так, например, для каждого набора объектов одного класса в рамках метода эталонов существует только один усредненный эталон.

5) Множество объектов X_i , определенное на основе фактов:

$$K_i^{Object} = \{k_{i,1}^{Object}, \dots, k_{i,a}^{Object}\}, \quad (11)$$

где $k_{i,j}^{Object} = \{w_{i,j,1}^{Object}\}$ – j -й объект множеств X_i и K_i^{Object} .

В рамках разработки обобщенной модели представления решающих функции объекты K_i^{Object} множества X_i показаны в виде его свойства по аналогии с классами K_i^{Class} и параметрами $K_i^{Property}$.

6) Обобщенная особенность множества X_i :

$$K_i = \{k_{i,1}, \dots, k_{i,q}\} = \begin{cases} K_i^{Object}, & \text{если описываются объекты} \\ K_i^{Class}, & \text{если описываются классы} \\ K_i^{Property}, & \text{если описываются параметры} \end{cases}, \quad (12)$$

где $k_{i,j}$ – j -я особенность множества X_i .

Таким образом, под обрабатываемыми данными в дальнейшем будем понимать особенности (объекты, классы и параметры) анализируемых объектов.

7) Характеристика анализируемых объектов, описывающая множество однотипных особенностей:

$$c_i = \{K_{i,1}, \dots, K_{i,w}\}, \quad (13)$$

где $K_{i,j}$ – j -я особенность данных.

8) Набор характеристик, описывающих единую специфику данных, в виде характерной черты:

$$C_i = \{c_{i,1}, \dots, c_{i,e}\}, \quad (14)$$

где $c_{i,j}$ – j -я характеристика данных.

9) Множество взаимосвязанных характерных черт:

$$d_k = \{C_{k,1}, \dots, C_{k,t}\}, \quad (15)$$

где $C_{k,j}$ – j -я характерная черта данных.

10) Множество наборов характерных черт:

$$D_i = \{d_{i,1}, \dots, d_{i,r}\}, \quad (16)$$

где $d_{i,j}$ – j -й независимый набор характерных черт данных.

Формальное представление свойств независимых данных

В работе [11] была предложена концепция представления шагов работы алгоритмов интеллектуального анализа данных в виде функции поиска $f_{Characteristic}$ соответствующих закономерностей между входными и целевыми данными в рамках применяемого метода их обработки. Обобщим рассматриваемую концепцию для анализируемых метрических методов распознавания образов на основе вышеуказанных элементов данных (4–14):

$$C_{New} = f_{Characteristic}(C_{Input}, C_{Goal}, P_{Characteristic}), \quad (17)$$

где C_{New} – множество s искомым особенностей анализируемых данных:

$$C_{New} = \{c_1^{New} = \{K_{1,1}^{New}, \dots, K_{1,c}^{New}\}\};$$

C_{Input} – исходное множество z особенностей входных данных в виде единого свойства анализируемой характерной черты данных:

$$C_{Input} = \{c_1^{Input} = \{K_{1,1}^{Input}, \dots, K_{1,z}^{Input}\}\};$$

C_{Goal} – исходное множество x особенностей целевых данных в виде единого свойства анализируемой характерной черты данных:

$$C_{Goal} = \{c_1^{Goal} = \{K_{1,1}^{Goal}, \dots, K_{1,x}^{Goal}\}\};$$

$P_{Characteristic}$ – параметры функции $f_{Characteristic}$.

Для рассматриваемых методов распознавания образов в качестве входных данных выступают особенности либо обучающей, либо тестовой выборки. Целевыми являются данные решающих функций в виде соответствующих анализируемых особенностей обучающих материалов. Под искомыми понимаются особенности входных данных для поиска. Одни и те же данные C_{Input} и C_{Goal} могут быть использованы разными алгоритмами обработки материалов для вышеуказанных этапов обучения и классификации тестовых объектов. Работы разных алгоритмов будут отличаться друг от друга типами обрабатываемых независимых характеристик (группами особенностей) для анализируемых особых черт входных и целевых данных. Поэтому необходимо разработать единый способ представления независимых характеристик для разных их видов в рамках множества особенностей анализируемых данных.

Для группирования соответствующих элементов по необходимым v категориям i -го свойства $v_{Feature.i}$ определяют следующие функции:

1) для группировки фактов по параметрам:

$$K_{Data.Parametr.i} = f_{Group.Parametr}(K_{Data.Parametr.Old.p}, v_{Feature.i}, flag), \quad (18)$$

где $K_{Data.Parametr.Old.p}$ – данные, у которых факты сгруппированы по категориям произвольного свойства $v_{Feature.p}$:

$$K_{Data.Parametr.Old.p} = \{k_1^{Data.Feature.Old.p}, \dots, k_s^{Data.Feature.Old.p}\};$$

$flag$ – параметр, принимающий значение True (истина), если факты необходимо группировать по b категориям свойства $v_{Feature.i}$, и False (ложь), когда необходимо объединить все факты в единое множество; $K_{Data.Parametr.i}$ – данные $K_{Data.Parametr.Old.p}$, у которых факты сгруппированы по категориям анализируемого свойства $v_{Feature.i}$:

$$K_{Data.Parametr.i} = \{k_1^{Data.Feature.i}, \dots, k_v^{Data.Feature.i}\};$$

2) для группировки особенностей:

$$C_{Data.Feature.i} = f_{Group.Property}(C_{Data.Feature.Old.p}, v_{Feature.i}, flag), \quad (19)$$

где $C_{Data.Feature.Old.p}$ – множество данных, у которых факты сгруппированы по категориям произвольного свойства $v_{Feature.p}$:

$$C_{Data.Feature.Old.p} = \{k_1^{Data.Feature.Old.p}, \dots, k_v^{Data.Feature.Old.p}\};$$

$C_{Data.Feature.i}$ – множество данных $C_{Data.Feature.Old.p}$, у которых факты сгруппированы по категориям анализируемого свойства $v_{Feature.i}$:

$$C_{Data.Feature.i} = \{k_1^{Data.Feature.i}, \dots, k_z^{Data.Feature.i}\};$$

3) для группировки характеристик:

$$C_{Data.Feature.i} = f_{Group.Characteristic}(C_{Data.Feature.Old.j}, v_{Feature.i}, flag), \quad (20)$$

где $C_{Data.Feature.Old.j}$ – характерная черта данных, у которой особенности сгруппированы по категориям произвольного свойства данных $v_{Feature.Old.j}$:

$$C_{Data.Feature.Old.j} = \{c_1^{Data.Feature.Old.j}, \dots, c_a^{Data.Feature.Old.j}\};$$

$C_{Data.Feature.i}$ – характерная черта данных $C_{Data.Feature.Old.j}$, у которой особенности сгруппированы по b категориям нового свойства данных $v_{Feature.i}$:

$$C_{Data.Feature.i} = \{c_1^{Data.Feature.i}, \dots, c_b^{Data.Feature.i}\};$$

4) для группировки характерных черт (15):

$$d_{Data.Feature.i} = f_{Group.Characteristics}(d_{Data.Feature.Old}, v_{Feature.i}, flag), \quad (21)$$

где $d_{Data.Feature.Old}$ – множество из z произвольных характерных черт данных, у которых особенности сгруппированы по категориям соответствующих свойств данных $\{v_{Feature.Old.1}, \dots, v_{Feature.Old.z}\}$:

$$d_{Data.Feature.Old} = \{C_1^{Data.Feature.Old.1}, \dots, C_z^{Data.Feature.Old.z}\};$$

$d_{Data.Feature.i}$ – множество из z характерных черт данных, у которых особенности сгруппированы по b категориям анализируемого нового свойства $v_{Feature.i}$:

$$d_{Data.Feature.i} = \{C_1^{Data.Feature.i}, \dots, C_z^{Data.Feature.i}\};$$

$Flag$ – набор параметров для соответствующих z характерных черт $d_{Data.Feature.Old}$:

$$Flag = \{flag_1, \dots, flag_z\}.$$

В основе обработки вышеуказанных функций лежат логические операции:

- если $flag = True$, то элементы группируются по соответствующему k -му отличительному признаку (категории) некоторого анализируемого свойства $v_{Feature.i}$, которое применяемый метод распознавания образов учитывает при определении искомым данных в рамках работы функции $f_{Characteristic}$:

- если $flag = False$, то элементы объединяются в одно множество.

В качестве примера приведем следующие категории данных, позволяющие отличить друг от друга такие однотипные особенности, как источник данных (класс анализируемых объектов, порядковый номер объекта и др.), иерархические уровни, на которые делятся объекты и их свойства (уровни эталонов в методе дробящихся эталонов), порядковый номер этапа обработки данных (например, i -й случайный выбор объекта из обучающего множества).

Таким образом, для представления независимых характеристик анализируемой характерной черты данных C_{Data} в рамках i -го свойства $v_{Feature.i}$ необходимо:

- присвоить каждому p -му факту C_{Data} требуемую категорию $v_p^{Category.Feature.i}$ свойства $v_{Feature.i}$;
- сгруппировать особенности (наборы фактов) характерной черты C_{Data} по b категориям свойства $v_{Feature.i}$.

Важно отметить, что категории свойства присваиваются фактам, а группируются особенности (наборы фактов). Это возможно благодаря тому, что особенности являются минимальным неделимым набором данных для обработки (например, множество K ближайших соседей). Поэтому все факты должны иметь единую категорию при группировании по особенностям.

Представим множество категории свойства $v_{Feature.i}$ в виде набора неповторяющихся целых положительных чисел следующим образом:

$$V_{Values.Feature.i} = \{1, 2, 3, \dots, b\}. \quad (22)$$

Отметим, что произвольная особенность $K_{i,j}^{Data}$ характерной черты C_{Data} может повторяться в нескольких характеристиках разных категорий $\{C_1^{Data.Feature.i}, \dots, C_b^{Data.Feature.i}\}$ одного свойства $v_{Feature.i}$. Например, в рамках этапа обучения метода опорных векторов объекты i -го класса участвуют в построении гиперплоскостей решающих функций для всех комбинаций пар классов с участием i -го класса. Поэтому необходимо для каждого факта данных k_p устанавливать не одну категорию $v_p^{Category.Feature.i}$, а целый набор значений для определения принадлежности его каждой категории $\{1, 2, 3, \dots, b\}$ (22) свойства $v_{Feature.i}$. Это целесообразно делать следующим образом:

$$V_p^{Category.Feature.i} = \{v_{p,1}^{Category.Feature.i}, \dots, v_{p,b}^{Category.Feature.i}\}, \quad (23)$$

где $v_{p,j}^{Category.Feature.i}$ – значение принадлежности k_p факта j -й категории свойства $v_{Feature.i}$: 0 – если не принадлежит, 1 – если принадлежит.

Таким образом, любой метод распознавания образов должен иметь для всякой искомой характерной черты алгоритм, который будет определять для каждого p -го факта входных, целевых и искомым данных необходимые категории для группировки:

- по особенностям в рамках свойства $v_{Feature.j}$:

$$v_p^{Category.Feature.j};$$

- по категориям свойства $v_{Feature.i}$ (23):

$$V_p^{Category.Feature.i}.$$

Формальное представление взаимосвязей между данными

В качестве анализируемых метрических методов распознавания образов были выбраны и рассмотрены следующие шесть алгоритмов:

- метод K ближайших соседей;
- метод потенциальных функций;
- метод эталонов;
- метод коррекционных приращений;
- метод наименьших среднеквадратических ошибок;
- метод наивный Байесовский.

При реализации алгоритмов этапов обучения и классификации для вышеуказанных методов распознавания в рамках функции $f_{Characteristic}$ (17) одни и те же однотипные объекты и их характеристики могут группироваться и обрабатываться по-разному для искомым особенностей входных данных. Так, например, усредненные эталоны соответствующих q классов обрабатываются двумя способами:

1) *отдельно друг от друга* на этапе обучения:

$$C_1^{Etalon} = \{c_{1,1}^{Etalon} = \{K_{1,1,1}^{Etalon}\}, \dots, c_{1,q}^{Etalon} = \{K_{1,q,1}^{Etalon}\}\},$$

где C_1^{Etalon} – эталоны классов в виде независимых составляющих $\{c_{1,1,1}^{Etalon}, \dots, c_{1,1,q}^{Etalon}\}$ одной характерной черты C_1^{Etalon} ; $K_{1,i,1}^{Etalon} = \{k_{1,1,i,1}^{Etalon}\}$ – представление эталона i -го класса в виде особенности;

$$C_1^{Class} = \{c_{1,1}^{Class} = \{K_{1,1,1}^{Class}\}, \dots, c_{1,q}^{Class} = \{K_{1,q,1}^{Class}\}\},$$

где C_1^{Class} – соответствующие эталонам наборы обучающих объектов одного класса в виде независимых составляющих $\{c_{1,1,1}^{Class}, \dots, c_{1,1,q}^{Class}\}$ одной характерной черты C_1^{Class} ; $K_{1,i,1}^{Class} = \{k_{1,1,i,1}^{Class}, \dots, k_{1,1,i,x}^{Class}\}$ – представление набора обучающих объектов i -го класса в виде особенности;

2) *в виде единого набора данных* на этапе классификации тестового объекта в рамках поиска минимального расстояния до них от тестового объекта:

$$C_1^{Etalon} = \{c_{1,1}^{Etalon} = \{K_{1,1,1}^{Etalon}, \dots, K_{1,q,1}^{Etalon}\}\},$$

где C_1^{Etalon} – эталоны классов в виде одного свойства $c_{1,1}^{Etalon}$ характерной черты C_1^{Etalon} ;

$$C_1^{Test} = \{c_{1,1}^{Test} = \{K_{1,1,1}^{Test}, \dots, K_{1,w,1}^{Test}\}\},$$

где C_1^{Test} – тестовые объекты в виде одной характеристики $c_{1,1}^{Test}$ репрезентативной черты C_1^{Test} ; $K_{1,i,1}^{Test} = \{k_{1,1,i,1}^{Test}\}$ – представление i -го тестового объекта для классификации в виде особенности.

Чтобы учесть, какие входные данные являются независимыми, а какие – общими, необходимо представить работу алгоритма, основанного на функции $f_{Characteristic}$, как процесс обработки иерархических данных. Для этого представим входные, целевые и искомые закономерности в виде сгруппированных фактов по необходимым свойствам данных следующим образом:

$$\begin{aligned} d_{Input} &= \{C_1^{Input}, \dots, C_z^{Input}\}, \\ d_{Goal} &= \{C_1^{Goal}, \dots, C_z^{Goal}\}, \\ d_{New} &= \{C_1^{New}, \dots, C_z^{Goal}\}, \end{aligned}$$

где d_{Input} , d_{Goal} , d_{New} – исходные множества анализируемых характерных черт входных, целевых и искомого данных.

Для вышеуказанных методов распознавания свойственны следующие особенности представления данных d_{Input} , d_{Goal} и d_{New} :

1) множества характерных черт закономерностей d_{Input} , d_{Goal} и d_{New} сгруппированы по z категориям одного анализируемого свойства $v_{Feature.All.Feature}$ на основе функции $f_{Group.Characteristics}$ (21), в рамках которых определяются независимые данные для обработки;

2) множество элементов $\{c_{1,1}^{Input}, \dots, c_{1,a}^{Input}\}$ для каждой характерной черты C_i^{Input} данных d_{Input} сгруппировано по a категориям анализируемого свойства $v_{Feature.Input.Characteristic}$ в рамках функции $f_{Group.Characteristic}$ (20); с помощью числа a категорий свойства $v_{Feature.Input.Characteristic}$ определяется общий или индивидуальный характер искомого данных относительно входных;

3) множество элементов $\{c_{1,1}^{Goal}, \dots, c_{1,b}^{Goal}\}$ для каждой характерной черты C_i^{Goal} данных d_{Goal} сгруппировано по b категориям анализируемого свойства $v_{Feature.Goal.Characteristic}$; аналогично предыдущей особенности общий или индивидуальный характер искомого данных относительно целевых данных определяется по категориям свойства $v_{Feature.Goal.Characteristic}$;

4) для каждой характерной черты C_i^{New} данных d_{New} множество искомого элементов $\{c_{i,1,1}^{New}, \dots, c_{i,1,b}^{Goal}, \dots, c_{i,a,1}^{Goal}, \dots, c_{i,a,b}^{Goal}\}$ сгруппировано по $a * b$ категориям анализируемых свойств $v_{Feature.Input.Characteristic}$ и $v_{Feature.Goal.Characteristic}$;

5) множества особенностей для каждой характеристики закономерностей d_{Input} , d_{Goal} и d_{New} сгруппированы по категориям анализируемого свойства $v_{Feature.Property}$ на основе функции $f_{Group.Property}$ (19) в рамках определения общих и независимых данных для обработки;

6) множества фактов для каждой особенности закономерностей d_{Input} , d_{Goal} и d_{New} сгруппированы по категориям анализируемого свойства $v_{Feature.Parametr}$ на основе функции $f_{Group.Parametr}$ (18) в рамках определения неделимых наборов данных для обработки.

Обозначим произвольную закономерность d в виде структуры дерева и определим для нее следующие обозначения:

$$\begin{aligned} - e_i^{Feature} &= d. C_i - i\text{-я характерная черта;} \\ - e_{i,j}^{Characteristic} &= d. C_i. C_j - j\text{-я характеристика } d. C_i; \\ - e_{i,j,k}^{Property} &= d. C_i. C_j. K_k - k\text{-я особенность } d. C_i. C_j; \\ - e_{i,j,k,p}^{Parametr} &= d. C_i. C_j. K_k. k_p - p\text{-й параметр } d. C_i. C_j. K_k. \end{aligned}$$

Представим все элементы одного вида закономерности d в виде следующих объединений $n_{Parts} = q$ множеств, где в каждом i -м подмножестве $n_i^{Elements} = w$ элементов:

для *Feature*:

$$\begin{aligned} E_{Feature} &= \{e_1^{Feature}, \dots, \cup, \dots, e_z^{Feature}\} = \\ &= \{E_1^{Feature}, \dots, \cup, \dots, E_q^{Feature}\}, \end{aligned}$$

для *Characteristic*:

$$\begin{aligned} E_{Characteristic} &= \{e_{1,1}^{Characteristic}, \dots, \cup, \dots, e_{z,s}^{Characteristic}\} = \\ &= \{E_1^{Characteristic}, \dots, \cup, \dots, E_q^{Characteristic}\}, \end{aligned}$$

для *Property*:

$$E_{Property} = \{e_{1,1,1}^{Property}, \dots, \cup, \dots, e_{z,s,g}^{Property}\} = \{E_1^{Property}, \dots, \cup, \dots, E_q^{Property}\},$$

для *Parametr*:

$$E_{Parametr} = \{e_{1,1,1,1}^{Parametr}, \dots, \cup, \dots, e_{z,s,g,k}^{Parametr}\} = \{E_1^{Parametr}, \dots, \cup, \dots, E_q^{Parametr}\}.$$

Определим обобщенные функции копирования и разделения множества однотипных элементов для всех уровней иерархии произвольной закономерности d :

$$D = f_{Copy.Elements}(d, n_{Parts}),$$

$$D = f_{Split.Elements}(d, type, n_{Parts}, N_{Elements}), \quad (24)$$

где d – произвольная закономерность для обработки; $n_{Parts} = q$ – число копий закономерности d ; $type = \{Feature, Characteristic, Property, Parametr\}$ – виды элементов, которые функция может обрабатывать; $n_{Parts} = q$ – число частей, на которые разбивается единое множество элементов вида $type$ в рамках d ; $N_{Elements} = \{n_1^{Elements}, \dots, n_q^{Elements}\}$ – число элементов в соответствующих $n_{Parts} = q$ частях; $D = \{d_1, \dots, d_q\}$ – итоговое множество $n_{Parts} = q$ закономерностей, содержащее соответствующие части единого набора разбиваемых элементов.

Функция $f_{Split.Elements}$ разделяет закономерность d на множество $n_{Parts} = q$ неполных частей, которые содержат непересекающиеся данные вида $type$:

$$d_1 \cdot E_{Type}^1, \dots, \cup, \dots, d_q \cdot E_{Type}^q = d \cdot E_{Type}$$

$$d_1 \cdot E_{Type}^1, \dots, \cap, \dots, d_q \cdot E_{Type}^q = \emptyset. \quad (25)$$

Представим обобщенный алгоритм распараллеливания вычисления для вышеуказанных методов распознавания, основанный на использовании функции $f_{Characteristic}$, в виде последовательной обработки иерархических независимых данных следующим образом:

Эман 1. Обработка независимых характерных черт

Все характерные черты в рамках одной категории (входные, целевые и искомые данные) обрабатываются отдельно друг от друга, а характерные черты с одним индексом разных категорий – вместе (рисунок 1, где красная линия отображает направление потока данных для обработки в рамках i -й характерной черты). Разделим с помощью алгоритма, основанного на использовании функции $f_{Split.Elements}$ (24), данные входной, целевой и искомой закономерностей на независимые части вида D (16), состоящих из одной характерной черты, в рамках z категорий свойства $v_{Feature.All.Feature}$ следующим образом:

$$D_{Input-Feature} = \{d_1^{Input-Feature}, \dots, d_z^{Input-Feature}\},$$

где $D_{Input-Feature}$ – набор z частей характерных черт закономерности входных данных d_{Input} ;

$$D_{Goal-Feature} = \{d_1^{Goal-Feature}, \dots, d_z^{Goal-Feature}\},$$

где $D_{Goal-Feature}$ – набор z частей характерных черт закономерности целевых данных d_{Goal} ;

$$D_{New-Feature} = \{d_1^{New-Feature}, \dots, d_z^{New-Feature}\},$$

$D_{New-Feature}$ – набор z частей характерных черт закономерности искомых данных d_{New} .

Эман 2. Обработка независимых характеристик данных

Все характеристики входных и целевых данных в рамках своей категории обрабатываются отдельно друг от друга, а характеристики вида $(c_{i,j}^{Input}, \{c_{i,1}^{Goal}, \dots, c_{i,p}^{Goal}\})$ – вместе (см. рисунок 1, где зелеными линиями показана обработка каждой входной характеристики со всеми целевыми). Пара характеристик вида $(c_{i,j}^{Input}, c_{i,p}^{Goal})$ однозначно определяет искомую характеристику $c_{i,j,p}^{New}$, т.е. число искомых характеристик равно количеству входных, умноженному на число целевых характеристик.

Определим следующие данные для i -й характерной черты, полученные на предыдущем этапе – входные, целевые и искомые:

$$d_i^{Input-Feature} = \{c_{i,1}^{Input}, \dots, c_{i,a}^{Input}\},$$

$$d_i^{Goal-Feature} = \{c_{i,1}^{Goal}, \dots, c_{i,b}^{Goal}\},$$

$$d_i^{New-Feature} = \{c_{1,1,1}^{New}, \dots, c_{1,1,b}^{Goal}, \dots, c_{1,a,1}^{Goal}, \dots, c_{1,a,b}^{Goal}\}.$$

Разделим характеристики на примере входной, целевой и искомой закономерности ($d_i^{Input-Feature}$, $d_i^{Goal-Feature}$ и $d_i^{New-Feature}$) на независимые части данных с помощью $v_{Feature.Input.Characteristic}$ и $v_{Feature.Goal.Characteristic}$ следующим образом:

$$D_i^{Input-Characteristic} = \{d_1^{Input-Characteristic}, \dots, d_a^{Input-Characteristic}\},$$

где $D_i^{Input-Characteristic}$ – набор a закономерностей с одной характеристикой от $d_i^{Input-Feature}$;

$$D_i^{New-Characteristic} = \{d_{i,1}^{New-Characteristic.1}, \dots, d_{i,r}^{New-Characteristic.1}\},$$

где $D_i^{New-Characteristic}$ – набор a закономерностей, содержащих по b характеристик $d_i^{New-Feature}$;

$$D_i^{Goal-Characteristic} = \{d_1^{Goal-Characteristic}, \dots, d_b^{Goal-Characteristic}\},$$

где $D_i^{Goal-Characteristic}$ – набор b закономерностей с одной характеристикой от $d_i^{Goal-Feature}$;

$$D_{i,j}^{New-Characteristic} =$$

$$\{d_{i,j,1}^{New-Characteristic}, \dots, d_{i,j,b}^{New-Characteristic}\},$$

где $D_{i,j}^{New-Characteristic}$ – набор b закономерностей с одной характеристикой от $d_{i,j}^{New-Characteristic.1}$.

В рамках рассматриваемого этапа независимыми данными для обработки будут закономерности следующего вида:

$$(d_{i,j}^{Input-Characteristic}, d_{i,k}^{Goal-Characteristic}, d_{i,j,k}^{New-Characteristic}).$$

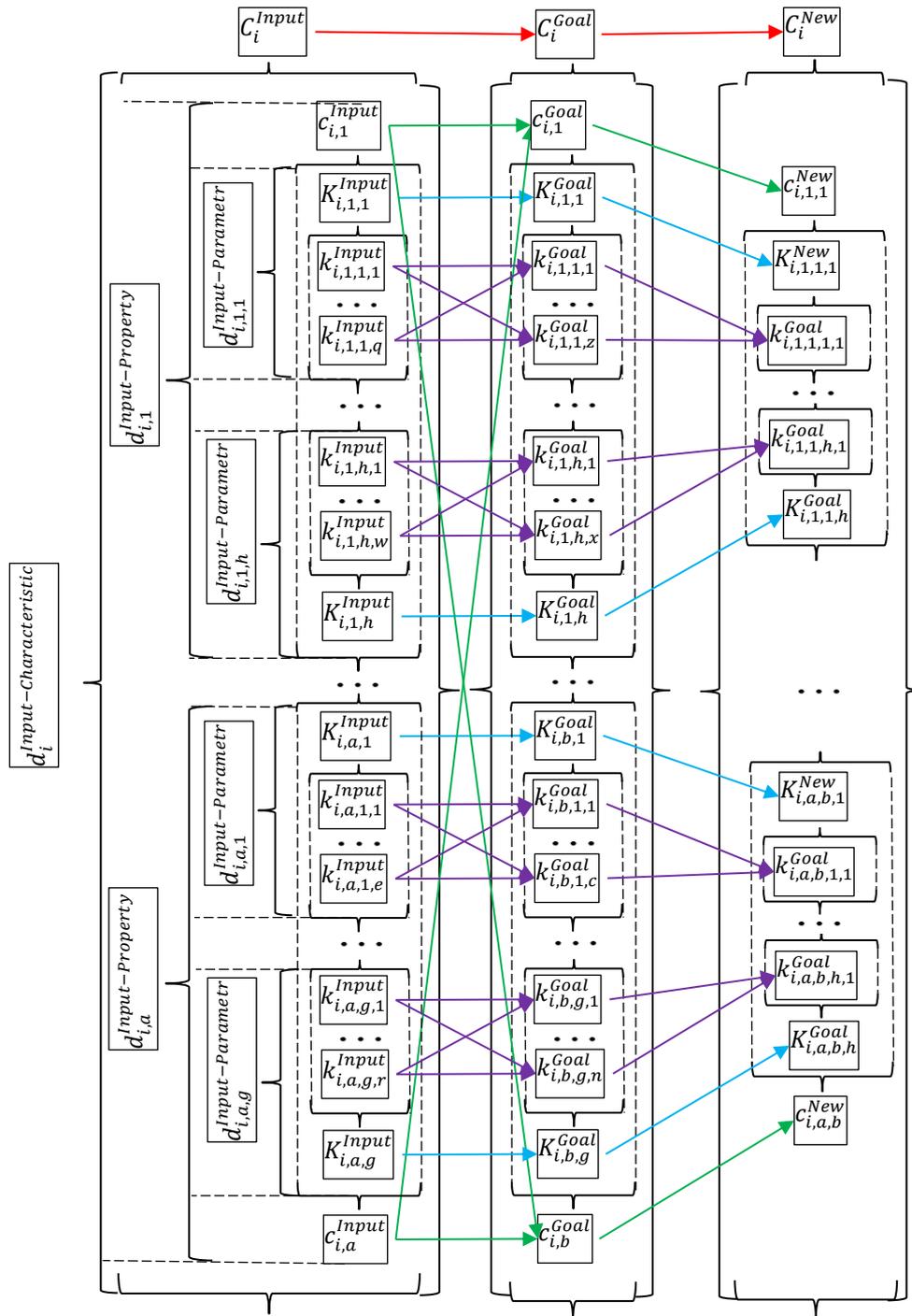


Рис. 1. Общий принцип обработки иерархических данных

Fig. 1. General Principle of Processing Hierarchical Data

Эман 3. Обработка независимых особенностей данных.

Особенности входных, целевых и искомых характеристик в рамках своей категории обрабатываются независимо друг от друга (см. рисунок 1, где синими линиями показаны отдельные обработки входной и целевой характеристик). Разделим данные входной, целевой и искомой закономерностей с помощью категорий свойства $v_{Feature.Property}$ на независимое количество v частей, содержащих по одной особенности, используя алгоритм (25):

$$D_{i,j}^{Input-Property} = \{d_{i,j,1}^{Input-Property}, \dots, d_{i,j,v}^{Input-Property}\},$$

где $D_{i,j}^{Input-Property}$ – набор z частей особенностей закономерности входных данных $d_{i,j}^{Input-Property}$;

$$D_{i,k}^{Goal-Property} = \{d_{i,k,1}^{Goal-Property}, \dots, d_{i,k,v}^{Goal-Property}\},$$

где $D_{i,k}^{Goal-Property}$ – набор z частей особенностей закономерности целевых данных $d_{i,k}^{Goal-Property}$;

$$D_{i,j,k}^{New-Property} = \{d_{i,j,k,1}^{New-Property}, \dots, d_{i,j,k,v}^{New-Property}\},$$

где $D_{i,j,k}^{New-Property}$ – набор z частей особенностей закономерности искомых данных $d_{i,j,k}^{New-Property}$.

Эман 4. Обработка независимых параметров данных.

Каждый параметр входных данных, как и на этапе 2, обрабатывается с целым набором параметров целевых данных в рамках категорий свойства $v_{Feature.Parametr}$ (см. рисунок 1, где фиолетовыми линиями показана общая обработка входных и целевых параметров). Параметры, в отличие от вышеуказанных этапов, необходимо разделять по заданному в них количеству объектов. Первые два этапа отвечают за учет общих и индивидуальных входных данных для целевых данных, принадлежащих разным классам (межклассовые взаимосвязи), а третий и четвертый этапы определяют общие и индивидуальные взаимосвязи между входными и целевыми данными внутри одного класса (внутриклассовые взаимосвязи).

Формальное представление модели решающих функций

В основе разработанной модели лежит принцип повторного использования входных данных для определения множества искомых закономерностей между ними и необходимыми для этого дан-

ными решающих функций. В рамках предлагаемого подхода определим модель решающих функций.

Модель m решающих функции хранит выявленные характерные черты обучающих данных d_{Goal} и искомые d_{New} входных данных в виде соответствующих наборов данных:

$$m = \langle d_{Goal}, d_{New} \rangle, \tag{26}$$

где $d_{Goal} = \{c_1^{Goal}, \dots, c_w^{Goal}\}$ – множество независимых наборов однотипных характеристик обучающей выборки, описывающих необходимые данные для определения заданного набора искомых характерных черт; $d_{New} = \{c_1^{New}, \dots, c_e^{New}\}$ – множество наборов искомых однотипных характеристик входных данных.

С помощью функции $f_{Characteristic}$ обобщим рассматриваемую концепцию для анализируемых метрических методов распознавания образов на основе предложенной модели (26) следующим образом:

$$m_e = f_{Features}(d_{Input}, m_0, P_{Features}), \tag{27}$$

где $m_e = (d_{Goal}, d_e^{New})$ – конечное состояние модели после определения всех e искомых характерных черт данных; $m_0 = (d_{Goal}, d_0^{New})$ – начальное состояние модели до определения всех o искомых характерных черт данных; d_{Input} – исходное множество обрабатываемых характерных черт входных данных:

$$d_{Input} = \{c_1^{Input}, \dots, c_i^{Input} = \{c_{i,1}^{Input} = \{K_{i,1}^{Input}, \dots, K_{i,z}^{Input}\}}, \dots, c_q^{Input}\};$$

d_{Goal} – исходное множество анализируемых характерных черт целевых данных:

$$d_{Goal} = \{c_1^{Goal}, \dots, c_i^{Goal} = \{c_{i,1}^{Goal} = \{K_{i,1}^{Goal}, \dots, K_{i,x}^{Goal}\}}, \dots, c_w^{Goal}\};$$

d_w^{New} – конечное состояние множества искомых характерных черт для входных данных:

$$d_w^{New} = \{c_{1,End}^{New}, \dots, c_{i,End}^{New} = \{c_{i,1}^{New} = \{K_{i,1}^{New}, \dots, K_{i,c}^{New}\}}, \dots, c_{e,End}^{New}\};$$

d_{New} – исходное множество искомых характерных черт для входных данных:

$$d_{New} = \{c_{1,0}^{New}, \dots, c_{i,0}^{New} = \{c_{i,1}^{New} = \{K_{i,1}^{New}, \dots, K_{i,c}^{New}\}}, \dots, c_{e,0}^{New}\};$$

$f_{Features}$ – соответствующая функция поиска искомых закономерностей (свойств, классов, параметров данных) m_w между d_{Input} и m_0 ; $P_{Features}$ – параметры работы функции $f_{Features}$.

Так как искомые характерные черты $\{c_1^{New}, \dots, c_e^{New}\}$ входных данных определяются последовательно, то процесс реализации алгоритма, использующего функцию $f_{Features}$ (27), представим в виде набора шагов по определению соответствующей h -й искомой характерной черты C_h^{New} :

$$m_h = f_h^{Feature}(d_{Input}, m_{h-1}, P_{Features}), \quad (28)$$

где $f_h^{Feature}$ – соответствующая функция поиска искомой особенности C_h^{New} между входными данными d_{Input} и текущими выявленными характерными чертами входных и обучающих данных m_{h-1} .

Выделим и рассмотрим следующие особенности этого алгоритма (28).

Во-первых, множества d_{Input} и d_{Goal} описывают достаточное количество данных для определения всех искомым характерных черт d_{New} . Однако для поиска h -й искомой особенности C_h^{New} не всегда требуется анализировать все характерные черты входных d_{Input} и целевых d_{Goal} данных.

Во-вторых, для определения h -й искомой характерной черты C_h^{New} функция $f_h^{Feature}$ также может использовать полученные на предыдущих этапах искомые особенности $\{c_1^{New}, \dots, c_{h-1}^{New}\}$ в качестве целевых данных.

Анализ вышеуказанных особенностей показывает, что для корректной работы алгоритма необ-

ходимо разработать способ представления данных, который позволит определять, какие характерные черты входных и целевых данных участвуют в получении искомой особенности C_h^{New} , а также – выявлять общие характерные черты для особенности C_h^{New} , которые необходимо обрабатывать целиком, без разбиения.

В интересах решения рассматриваемой проблемы дополнительно введем в набор значений категорий каждого свойства следующие технические индексы:

– индекс 0, отвечающий за отсутствие какой-либо связи между заданной характерной чертой и анализируемым свойством данных, т. е. характерная черта не участвует в обработке функции $f_h^{Features}$ для получения особенности C_h^{New} ;

– индекс All для группы однотипных особенностей, показывающий, что набор данных является общим для всех категорий свойства, и его необходимо обрабатывать целиком без разбиения.

Пусть метод распознавания образов в рамках модели m анализирует e различных искомым характерных черт данных с помощью функций вида $f_h^{Features}$. Сформируем в обобщенном виде для каждого p -го факта данных k_p собственный набор значений категорий соответствующих свойств (29) для всех e функции поиска искомым данных (30).

$$V_{p,h}^{Category.Feature.All} = \{V_{p,h}^{Category.Feature.All.Feature}, V_{p,h}^{Category.Feature.Input.Characteristic}, V_{p,h}^{Category.Feature.Goal.Characteristic}, V_{p,h}^{Category.Feature.Property}, V_{p,h}^{Category.Feature.Parametr}\}. \quad (29)$$

$$V_p^{Category.All.Feature.All} = \{V_{p,1}^{Category.Feature.All}, \dots, V_{p,e}^{Category.Feature.All}\}. \quad (30)$$

Операции над элементами модели знаний для параллельной обработки

Основным подходом к распараллеливанию вычислений для вышеуказанных методов распознавания образов является разделение данных на части по заданному в них количеству объектов. Тем не менее, как было показано выше, характерная черта может состоять как из общих, так и независимых частей данных. Таким образом, при распараллеливании по объектам необходимо учитывать принадлежность каждого объекта своей группе. Например, при формировании независимой части из $n * t$ объектов одного класса в рамках этапа обучения метода эталонов на вычислительный узел необходимо передать только усредненный эталон соответствующего класса. А для случая, когда формируется n частей разных классов по t объектов, необходимо передать n усредненных эталонов анализируемых классов. Таким образом,

объемы вычислений для двух рассматриваемых случаев равны, а объемы передаваемых данных – нет.

Также отметим, что на объем передаваемых данных влияет вышеуказанная возможность дублирования одной особенности для нескольких характеристик. Поэтому необходимо обеспечить распараллеливание вычислений по объектам без учета их зависимостей между собой, влияющих на объем передаваемых данных. Для решения этой задачи предлагается использовать подход, заключающийся в следующем:

1) представить все данные каждой категории (входной, целевой и искомой) на всех уровнях в виде непересекающихся подмножеств данных;

2) распараллеливать вычисления по независимым группам особенностей с целью получения только общих входных и целевых данных.

При этом, если отсутствует возможность разбиения данных на непересекающиеся части, особенности должны быть представлены в виде единого набора данных.

Так, например, в рамках операции построения гиперплоскостей:

$$C_{Goal} = \{c_{1,1,2}^{Hyper}, \dots, c_{1,n-1,n}^{Test}\}$$

для всех комбинации пар n классов метода опорных векторов множество наборов сгруппированных по классам обучающих объектов:

$$C_{Input} = \{K_{1,1}^{Class}, \dots, c_{1,n}^{Class}\}$$

невозможно разделить на непересекающиеся части относительно гиперплоскостей соответствующих решающих функций (рисунок 2а). Поэтому данные одного из множеств C_{Input} или C_{Goal} должны быть представлены в виде одной характеристики (рисунок 2б).

На практике количество целевых данных меньше, чем входных, поэтому целесообразно ограничиваться передачей только целевых данных, а для входных и искомым использовать распараллеливание, т. е. необходимо исключить учет категорий свойств $v_1^{Feature.Goal.Property}$ и $v_1^{Feature.Characteristic}$ из распараллеливания.

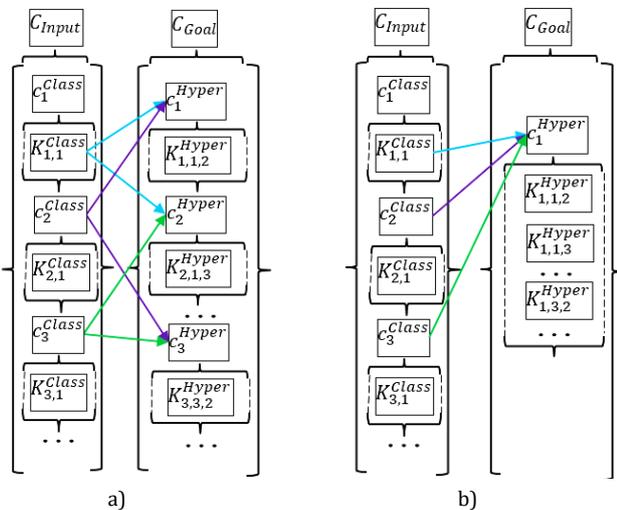


Рис. 2. Виды зависимостей между входными и целевыми данными метода опорных векторов: а) пересекающиеся; б) непересекающиеся

Fig. 2. Types of Dependencies between Input and Target Data of the Support Vector Machine: a) Intersecting; b) Non-Intersecting

Выделим два этапа предлагаемого подхода к распараллеливанию вычислений по объектам без учета их зависимостей между собой.

Этап 1. Распараллеливание вычислений на основе разделения входных, целевых и искомым данных на независимые части в рамках предложенного разбиения для последовательной обработки.

Этап 2. Распараллеливание вычислений в рамках разделения входных и целевых данных на части по заданному в них количеству объектов.

Рассмотрим выделенные этапы предлагаемого подхода к распараллеливанию вычислений на примере метода потенциальных функций более подробно (рисунок 3).

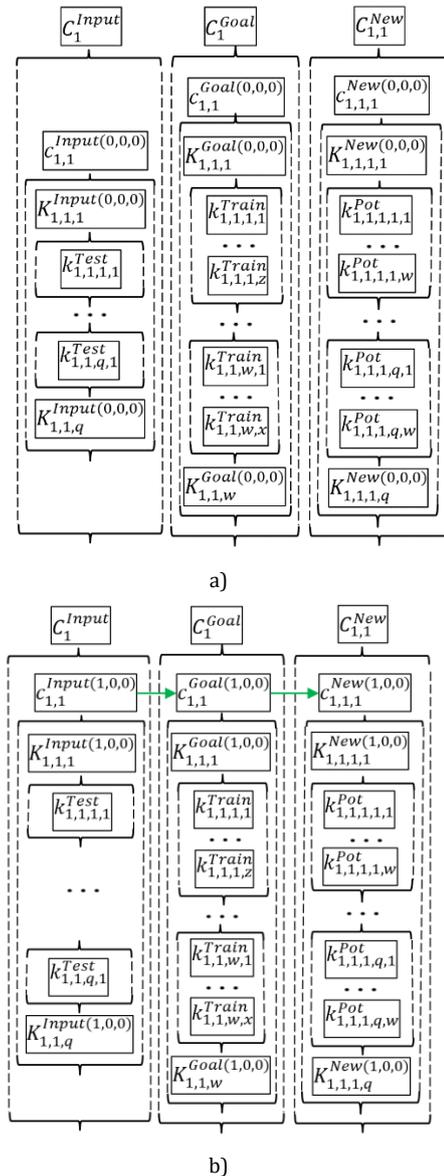


Рис. 3. Этапы распараллеливания метода потенциальных функций: а) этап инициализации данных; б) этап распараллеливания по свойству $v_1^{Feature.All.Feature}$

Fig. 3. Parallelization Stages of the Potential Function Method: a) Initialization Stage; b) Parallelization Stage by Property $v_1^{Feature.All.Feature}$

Входными данными для метода потенциальных функции являются следующие элементы (см. рисунок 3а):

$$d_{Input} = \left\{ C_1^{Input} = \left\{ c_{1,1}^{Input} = \left\{ K_{1,1,1}^{Input} = \right. \right. \right. \\ \left. \left. \left. = \left\{ k_{1,1,1,1}^{Test} \right\}, \dots, K_{1,1,q}^{Input} = \left\{ k_{1,1,q,1}^{Test} \right\} \right\} \right\} \right\},$$

d_{Input} – исходное множество q объектов для классификации, где $k_{1,1,i,1}^{Test}$ – i -й тестовый объект;

$$m_0 = (d_{Goal}, d_0^{New}),$$

где m_0 – начальное состояние модели до определения всех q классов тестовых объектов d_{Input} ;

$$d_{Goal} = \left\{ C_1^{Goal} = \left\{ c_{1,1}^{Goal} = \left\{ K_{1,1,1}^{Goal} = \right. \right. \right. \\ \left. \left. \left. = \left\{ k_{1,1,1,1}^{Train}, \dots, k_{1,1,1,z}^{Train} \right\}, \dots, K_{1,1,w}^{Goal} = \left\{ k_{1,1,w,1}^{Train}, \dots, k_{1,1,w,x}^{Train} \right\} \right\} \right\} \right\},$$

где d_{Goal} – исходное множество наборов обучающих объектов одного класса для w соответствующих классов;

$$d_0^{New} = \left\{ C_{1,1}^{New} = \left\{ c_{1,1,1}^{New} = \left\{ K_{1,1,1,1}^{New}, \dots, K_{1,1,1,i}^{New} = \right. \right. \right. \\ \left. \left. \left. = \left\{ k_{1,1,1,i,1}^{Pot}, \dots, k_{1,1,1,i,w}^{Pot} \right\}, \dots, K_{1,1,1,q}^{New} \right\} \right\} \right\},$$

где d_0^{New} – исходное множество наборов значений потенциалов w классов для соответствующих q тестовых объектов.

В рамках метода потенциальных функции входные, целевые и искомые данные содержат только по одной характерной черте данных, поэтому функция вида $f_h^{Feature}$ тоже одна: $f_1^{Feature}$. Как следствие, данные будут группироваться в рамках независимых множеств элементов только по следующим пяти свойствам $V_{p,h}^{Category.Feature.All}$ (29):

- $v_1^{Feature.All.Feature}$;
- $v_1^{Feature.Input.Characteristic}$;
- $v_1^{Feature.Goal.Characteristic}$;
- $v_1^{Feature.Property}$;
- $v_1^{Feature.Parametr}$.

Для простоты изложения последующего материала добавим индексы первых трех свойств к элементам обрабатываемых данных:

$$K_i^{(IG,IN,GN)} C_i^{(IG,IN,GN)} \text{ и } C_i^{(IG,IN,GN)},$$

где IG – индекс свойства $v_1^{Feature.All.Feature}$; IN – индекс свойства $v_1^{Feature.Input.Characteristic}$; GN – индекс свойства $v_1^{Feature.Goal.Characteristic}$.

Этап обработки независимых характерных черт для распараллеливания отличается от вышеуказанного этапа последовательной обработки тем, что необходимо представлять характерные черты в виде непересекающихся групп данных. Если это невозможно осуществить, целевые и искомые данные передаются целиком, а входные распараллеливаются. А в случае возможности представления данных в виде непересекающихся

групп допускается передавать искомые данные целиком. Тогда входные и целевые данные становятся общими относительно друг друга и, как следствие, можно сразу перейти к распараллеливанию по объектам.

Метод потенциальных функций имеет всего одну характерную черту в каждом наборе данных. Таким образом, выделим следующие особенности распараллеливания по $v_1^{Feature.All.Feature}$ (рисунок 3б).

Во-первых, тестовые объекты (входные данные) в рамках метода потенциальных функций являются общими для обучающих объектов (целевые данные), сгруппированным по классам, поэтому существует только одна индивидуальная категория для свойства $v_1^{Feature.All.Feature} = [1]$, и у всех особенностей обучающих данных первый индекс в скобках равен 1.

Во-вторых, аналогично входным данным у всех особенностей целевых данных первый индекс в скобках тоже равен 1.

В-третьих, аналогично входным и целевым данным у всех особенностей искомых данных первый индекс в скобках тоже равен 1.

Этап обработки независимых характеристик данных по категориям свойств $v_1^{Feature.Input.Characteristic}$ и $v_1^{Feature.Goal.Characteristic}$ отличается от соответствующего этапа последовательной обработки той же спецификой, что и этап обработки характерных черт.

Выделим следующие особенности распараллеливания по $v_1^{Feature.Input.Characteristic}$ (рисунок 4а).

Во-первых, входные данные содержат независимые друг от друга характеристики относительно искомых, поэтому существует q категорий свойства $v_1^{Feature.Input.Characteristic}$. Таким образом, входные данные разделяются на q групп, и второй индекс равен номеру соответствующей группы.

Во-вторых, целевые данные не участвуют в группировке по свойству $v_1^{Feature.Input.Characteristic}$, поэтому в рамках обобщения для всех целевых особенностей второй индекс равен *All*.

В-третьих, вторым индексам искомых данных аналогично входным также было присвоено соответствующее q значение.

Сформулируем особенности распараллеливания по $v_1^{Feature.Goal.Characteristic}$ (рисунок 4б).

Во-первых, входные данные не участвуют в группировке по свойству $v_1^{Feature.Goal.Characteristic}$, поэтому в рамках обобщения третий индекс всех входных особенностей равен техническому индексу *All*.

Во-вторых, целевые данные содержат независимые друг от друга множества обучающих объек-

тов для w классов ($v_1^{Feature.Goal.Characteristic} = \{1, \dots, w\}$). Таким образом, целевые данные разделяются на w групп и второй индекс равен номеру соответствующей группы.

Кроме того, третьим индексам искомым данных аналогично целевым также было присвоено соответствующее w значение.

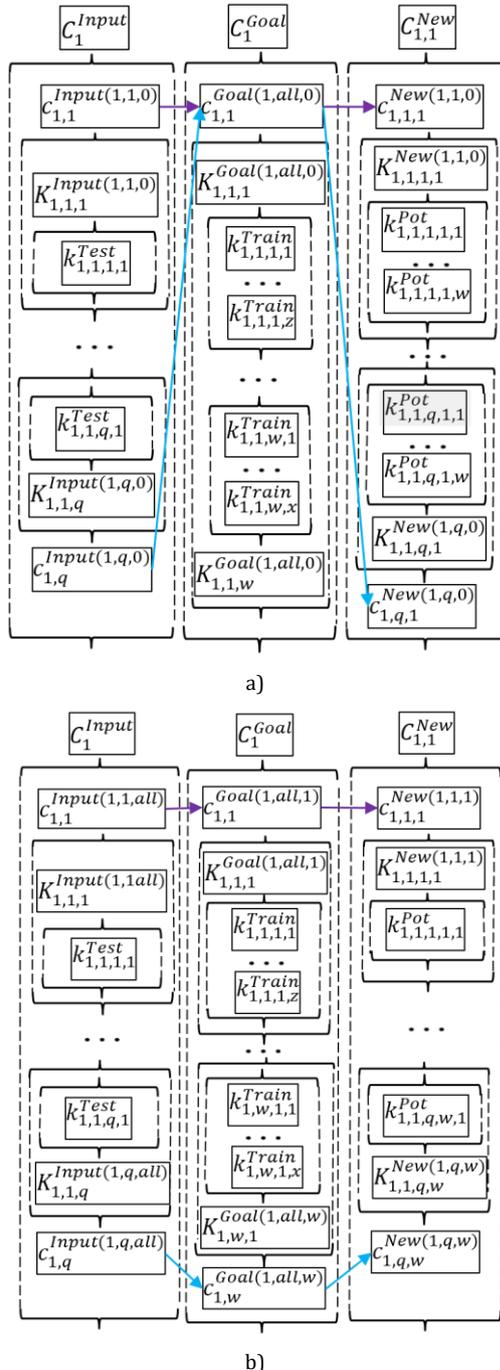


Рис. 4. Этапы распараллеливания метода потенциальных функций: а) по свойству $v_1^{Feature.Input.Characteristic}$; б) по свойству $v_1^{Feature.Goal.Characteristic}$
 Fig. 4. Parallelization Stages of the Potential Function Method: а) by Property $v_1^{Feature.Input.Characteristic}$; б) by property $v_1^{Feature.Goal.Characteristic}$

Этап обработки независимых особенностей данных: в рамках вышеуказанных ограничений на непересекаемость групп данных между собой их независимые особенности могут быть либо общими друг для друга, либо индивидуальными в рамках одинакового количества особенностей в характеристиках. Для метода потенциальных функций рассматриваемый этап отсутствует, так как в каждой характеристике находится только одна особенность (свойство $v_1^{Feature.Property}$).

Этап обработки независимых параметров данных с помощью $v_1^{Feature.Parametr}$: в силу неделимости фактов некоторых параметров получить точное число элементов в каждой части не всегда является возможным. Например, в рамках метода K ближайших соседей число фактов в каждой независимой части целевых данных в обобщенном виде всегда будет кратно числу k .

Проанализировав особенности распараллеливания вычислений в рамках этапов последовательной обработки независимых данных, определим на основе ранее разработанных функций обработки закономерностей следующие обобщенные функции:

1) функция копирования модели:

$$M = f_{Copy.Models}(m, n_{parts}) = \{m_1, \dots, m_q\}, \quad (31)$$

где $n_{parts} = q$ – число копий модели m ; $M = \{m_1, \dots, m_q\}$ – выходное множество моделей;

2) функция разделения модели:

$$M = f_{split.Models}(m, type, n_{parts}, N_{Elements}) = \{m_1 \cdot E_{Type}^1, \dots, m_q \cdot E_{Type}^q\}, \quad (32)$$

$type = \{Feature, Characteristic, Property, Parametr\}$ – виды элементов, которые функция может обрабатывать; $n_{parts} = q$ – число частей, на которые разбивается единое множество элементов вида $type$ в рамках m ; $N_{Elements} = \{n_1^{Elements}, \dots, n_q^{Elements}\}$ – число элементов в соответствующих $n_{parts} = q$ частях;

3) функция агрегирования набора моделей в единую модель:

$$m = f_{Union.Models}(M, type) = \{m \cdot \{d_1 \cdot E_{type}^1, \dots, \oplus, \dots, d_q \cdot E_{type}^q\}\}, \quad (33)$$

где \oplus – операция агрегирования для соответствующих элементов вида $type$.

Разработанные функции (31–33) обеспечивают распараллеливание вычислений процесса классификации объекта как по входным данным, так и по данным решающих функций.

Результаты эксперимента по распараллеливанию метода потенциальных функций

Алгоритм распараллеливания как по входным данным, так и по данным предложенной модели решающих функций метода потенциальных функций был протестирован на различных частях данных задачи *Customer Churn DataSet* [15]. Задача заключается в разделении множества тестируемых клиентов (объектов) на $z = 2$ класса (останется или уйдет клиент), где каждый классифицируемый объект имеет $e = 10$ атрибутов (столбцов) и идентификационный номер. Три атрибута имеют строковый тип с ограниченным числом принимаемых значений вида $\{a_1, a_2, \dots, a_q\}$ и поэтому были приведены к числовому типу на основе следующего принципа: $a_1 = 1, \dots, a_q = q$.

Определим следующие параметры:

c – количество компьютеров (клиентов);

TN_i – время передачи i -ых входных данных D_i и модели решающих функции M_i по сети между сервером и i -ым клиентом;

TP_i – время построения i -й модели M_i с учетом входных данных D_i на i -м клиенте.

В качестве критерия эффективности выступает усредненный коэффициент полезного действия (КПД) построения одной части модели решающих функций на c компьютерах с учетом времени передачи данных по сети:

$$\text{КПД} = \frac{1}{c} * \frac{\sum_{i=1}^c TP_i}{\sum_{i=1}^c (TP_i + TN_i)}. \quad (34)$$

В качестве вычислительного ресурса выступает Grid-система с переменным количеством клиентов (от 1 до 10) и одним сервером, которые имеют следующие одинаковые аппаратные характеристики: процессор *Intel(R) Core (TM) i3-4130 CPU 3.40 GHz*, ОЗУ 4 Гб.

Входные параметры данных, алгоритма, конфигурации Grid-системы, на основе которых тестировалось программное обеспечение.

1) Число c клиентов в Grid-системе менялось следующим образом: 1, 3, 5, 7, 10.

2) Выборка обучающих объектов оставалась неизменной и содержала $w = 3000$ объектов (строк), каждый ее элемент имел по $e = 10$ атрибутов (столбцов) плюс один столбец класса.

3) Данные модели решающих функций метода потенциальных функций в виде набора обучающих объектов из w элементов распараллеливались по соответствующему числу используемых клиентов на следующие количества частей: 1 ($w = 3000$); 3 ($w = 1000$); 5 ($w = 600$); 7 ($w = 428$); 10 ($w = 300$).

4) Количество тестируемых объектов для распознавания в рамках распараллеливания по вход-

ным данным изменялось следующим образом: 1000, 2000, 3000, 4000, 5000.

5) Число классов равняется двум, что не оказывает существенного влияния на время передачи и обработки данных, и поэтому в рамках эксперимента будут учитываться только зависимости времени обработки и передачи данных от количества обучающих и тестируемых объектов.

Результаты эксперимента представлены в таблице 1, а графическое представление ее данных – на рисунке 5.

Обобщенные выводы

Во-первых, наличие только одного клиента ($c = 1$) отвечает за случай, когда распараллеливаются входные данные, а не модель решающих функций.

Во-вторых, распараллеливание предложенной модели решающих функций позволяет создавать подзадачи меньшего размера и задействовать дополнительные вычислительные ресурсы с целью минимизации времени обработки одного объекта относительно случая, когда распараллеливаются вычисления только по входным данным.

В-третьих, уменьшение размера модели решающих функций приводит к уменьшению КПД обработки, так как время передачи данных по сети прямо пропорционально количеству передаваемых обучающих и тестовых объектов, а время построения модели зависит от произведения числа обучающих объектов и количества тестовых объектов (поиск расстояний производится от каждого из них до всех обучающих). Таким образом, существует нелинейная зависимость КПД обработки от количества обучающих и тестовых объектов.

В-четвертых, в случаях наличия ограничений объема памяти компьютеров для хранения данных решающих функций или времени горизонта планирования возможность распараллеливания предложенной модели решающих функций позволяет более эффективно использовать вычислительные ресурсы за счет учета вышеуказанной нелинейной зависимости КПД от размеров обрабатываемых данных.

ТАБЛИЦА 1. Результаты КПД построения одной модели решающих функций для разного числа клиентов

TABLE 1. Efficiency Results on Building one Decision Functions Model for a Different Number of Clients

	$c = 1,$ $w = 3000$	$c = 3,$ $w = 1000$	$c = 5,$ $w = 600$	$c = 7,$ $w = 428$	$c = 10,$ $w = 300$
$q = 1000$	88,2	83,3	78,9	74,9	69,7
$q = 2000$	92,3	86,9	82,1	77,9	72,2
$q = 3000$	93,7	88,2	83,3	78,9	73,1
$q = 4000$	94,4	88,8	83,9	79,4	73,6
$q = 5000$	94,9	89,2	84,2	79,7	73,8

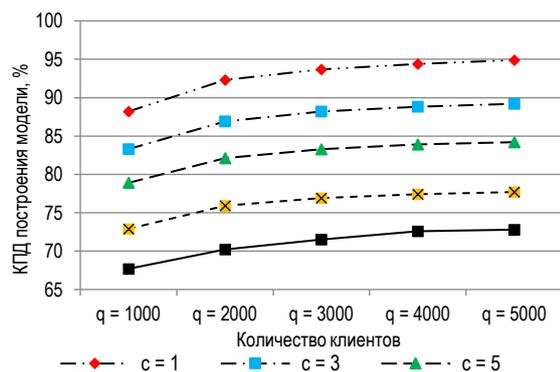


Рис. 5. Результаты КПД построения одной модели решающих функций для разного числа клиентов

Fig. 5. Efficiency Results on Building one Decision Functions Model for a Different Number of Clients

Заключение

В результате проведенных исследований была разработана унифицированная модель решающих функций для алгоритмов машинного обучения и функции, обеспечивающей ее распараллеливание как по входным данным, так и по данным решаю-

щих функций. Приведены примеры, показывающие, что предложенный алгоритм распараллеливания вычислений на основе вышеуказанной модели повышает эффективность классификации как для одного объекта, так и для группы объектов.

Научная новизна предложенного подхода заключается в том, что модель описывает объекты и их свойства единым способом в виде унифицированных элементов как для обучающих, так и классификационных данных. Кроме того, модель имеет структуру и функции, ориентированные на ее параллельную обработку методами распознавания образов на основе решающих функций в рамках группового распараллеливания объектов. Это позволяет осуществить классификацию объекта в режиме реального времени.

Предложенная модель носит также унифицированный характер и применима при распараллеливании других методов распознавания образов, которые могут быть описаны сходными параметрами, архитектурой и классификационными признаками.

Список источников

1. Балашова И.В., Терещенко Т.А. Системы поддержки принятия решений // The Scientific Heritage. 2021. № 79-4(79). С. 3–7. DOI:10.24412/9215-0365-2021-79-4-3-7. EDN:VBGSEN
2. Конев К.А. Машинное обучение для поддержки принятия решений в сфере качества на промышленном предприятии // Экономика. Информатика. 2023. Т. 50. № 3. С. 689–703. DOI:10.52575/2687-0932-2023-50-3-689-703. EDN:EOVNCU
3. Макшанов А.В., Журавлев А.Е. Технологии интеллектуального анализа данных: учебное пособие. СПб.: Лань, 2019. 212 с.
4. Замятин А.В. Интеллектуальный анализ данных: учебное пособие. Томск: Национальный исследовательский Томский государственный университет, 2020. 196 с. EDN:RODEPA
5. Платонов А.В. Машинное обучение: учебное пособие для вузов. М.: Издательство Юрайт, 2025. 89 с.
6. Лимановская О.В., Алферьева Т.И. Основы машинного обучения: учебное пособие. Екатеринбург: Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, 2020. 88 с. EDN:YLJQGT
7. Фомин В.В., Миклуш В.А. Интеллектуальные информационные системы: учебное пособие. СПб.: Российский государственный гидрометеорологический университет, 2013. 150 с. EDN:UPSHTP
8. Кочубей Р.И., Бычковский М.М., Зайкин Н.Н., Фатьянова Е.В., Свидло А.В., Чуприков О.В. Обоснование метода распознавания для решения задачи контроля содержимого данных о состоянии техники связи специального назначения // Известия Тульского государственного университета. Технические науки. 2023. № 12. С. 533–535. DOI:10.24412/2071-6168-2023-12-533-534. EDN:HVOKOQ
9. Дюк В.А., Малыгин И.Г., Прицкер В.И. Распознавание транспортных средств по силуэтам – трехкаскадный метод машинного обучения в системах технического зрения // Морские интеллектуальные технологии. 2022. № 2-1(56). С. 162–167. DOI:10.37220/MIT.2022.56.2.022. EDN:WQMNAG
10. Зайнидинов Х.Н., Каршиев З.А. Особенности параллельного выполнения алгоритмов интеллектуального анализа данных // Автоматика и программная инженерия. 2020. № 1(31). С. 83–91. EDN:YRWQDE
11. Холод И.И. Модели и методы построения параллельных алгоритмов анализа распределенных данных. Дис. ... докт. техн. наук. СПб.: Санкт-Петербургский государственный электротехнический университет “ЛЭТИ” им. В.И. Ульянова (Ленина), 2018. 351 с. EDN:HKSFMU
12. Kholod I.I., Shorov A.V. Unification of mining model for parallel processing // Proceedings of the Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus, St. Petersburg and Moscow, Russian Federation, 01–03 February 2017). IEEE, 2017. PP. 450–455. DOI:10.1109/EIConRus.2017.7910588. EDN:XMZHNX
13. Холод И.И. Метод определения возможностей параллельного выполнения функций алгоритмов анализа данных // Программные продукты и системы. 2018. № 2. С. 268–274. DOI:10.15827/0236-235X.122.268-274. EDN:UYSUBG
14. Bishop C. Pattern Recognition and Machine Learning. Berlin: Springer, 2006. 738 p.
15. Customer Churn DataSet // Kaggle. URL: <https://www.kaggle.com/datasets/muhammadshahidazeem/customer-churn-dataset> (Accessed 10.04.2025)

References

1. Balashova I., Tereshchenko T. Decision Support Systems. *The Scientific Heritage*. 2021;79-4(79):3–7. (in Russ.) DOI:10.24412/9215-0365-2021-79-4-3-7. EDN:VBGSEN
2. Konev K.A. Machine Learning for Quality Decision Support in an Industrial Enterprise. *Economics. Information technologies*. 2023;50(3):689–703. (in Russ.) DOI:10.52575/2687-0932-2023-50-3-689-703. EDN:E0VNCU
3. Makshanov A.V., Zhuravlev A.E. *Technologies of Intellectual Data Analysis*. St. Petersburg: Lan' Publ.; 2019. 212 p. (in Russ.)
4. Zamyatin A.V. *Intelligent Data Analysis*. Tomsk: National Research Tomsk State University Publ.; 2020. 196 p. (in Russ.) EDN:RODEPA
5. Platonov A.V. *Machine Learning*. Moscow: Yurajt Publ.; 2025. 89 p. (in Russ.)
6. Limanovskaya O.V., Alferyeva T.I. *Basics of Machine Learning*. Ekaterinburg: Ural Federal University Publ.; 2020. 88 p. (in Russ.) EDN:YLJQGT
7. Fomin V.V., Miklush V.A. *Intelligent Information Systems*. St. Petersburg: Russian State Hydrometeorological University Publ.; 2013. 150 p. (in Russ.) EDN:UPSHTP
8. Kochubey R.I., Bychkovsky M.M., Zaikin N.N., Fatyanova E.V., Svidlo A.V., Chuprikov O.V. Justification of the Recognition Method for Solving the Problem of Monitoring the Content of Status Data Special Purpose Communication Technicians. *Izvestiya Tula State University*. 2023;12:533–535. (in Russ.) DOI:10.24412/2071-6168-2023-12-533-534. EDN:HVOKOQ
9. Duke V.A., Malygin I.G., Pritsker V.I. Vehicle Recognition by Silhouettes – A Three-Stage Machine Learning method in Computer Vision Systems. *Marine Intelligent Technologies*. 2022;2-1(56):162–167. (in Russ.) DOI:10.37220/MIT.2022.56.2.022. EDN:WQMNAG
10. Zaynidinov X.N., Karshiev Z.A. Features of Parallel Execution of Data Mining Algorithms. *Automatics & Software Enginery*. 2020;1(31):83–91. (in Russ.) EDN:YRWQDE
11. Kholod I.I. *Models and Methods of Building Parallel Algorithms for Analyzing Distributed Data*. D.Sc Thesis. St. Petersburg: Saint Petersburg Electrotechnical University "LETI" Publ.; 2018. 351 p. (in Russ.) EDN:HKSFMU
12. Kholod I.I., Shorov A.V. Unification of mining model for parallel processing. *Proceedings of the Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus, 01–03 February 2017, St. Petersburg and Moscow, Russian Federation*. IEEE; 2017. p.450–455. DOI:10.1109/EIConRus.2017.7910588. EDN:XMZHNX
13. Kholod I.I. A Method for Determining the Capabilities of Parallel Execution of Data Mining Algorithm Functions. *Software & Systems*. 2018;2:268–274. (in Russ.) DOI:10.15827/0236-235X.122.268-274. EDN:UYSUBG
14. Bishop C. *Pattern Recognition and Machine Learning*. Berlin: Springer, 2006. 738 p.
15. *Kaggle*. Customer Churn DataSet. URL: <https://www.kaggle.com/datasets/muhammadshahidazeem/customer-churn-dataset> [Accessed 10.04.2025]

Статья поступила в редакцию 20.02.2025; одобрена после рецензирования 03.04.2025; принята к публикации 10.04.2025.

The article was submitted 20.02.2025; approved after reviewing 03.04.2025; accepted for publication 10.04.2025.

Информация об авторах:

**АЛЕКСАНДРОВ
Игорь Васильевич**

аспирант кафедры программной инженерии и вычислительной техники Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
 <https://orcid.org/0000-0002-6258-8191>

**ВИВЧАРЬ
Роман Михайлович**

кандидат технических наук, доцент кафедры программной инженерии и вычислительной техники Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
 <https://orcid.org/0000-0003-3865-9102>

**КИРИЧЕК
Руслан Валентинович**

доктор технических наук, профессор, ректор Санкт-Петербургского государственного университета телекоммуникаций им. проф. М.А. Бонч-Бруевича
 <https://orcid.org/0000-0002-8781-6840>

Киричек Р.В. является главным редактором журнала «Труды учебных заведений связи» с 2023 г., но не имеет никакого отношения к решению опубликовать эту статью. Статья прошла принятую в журнале процедуру рецензирования. Об иных конфликтах интересов авторы не заявляли.

Kirichek R.V. has been an editor-in-chief of the journal "Proceedings of Telecommunication Universities" since 2023, but has nothing to do with the decision to publish this article. The article has passed the review procedure accepted in the journal. The authors have not declared any other conflicts of interest.