DOI: 10.15514/ISPRAS-2025-37(5)-2

Применение кодов в модульной метрике для поиска k-соседей

А.Р. Шарапов, ORCID: 0000-0002-4794-0206 <arsharapov@hse.ru>
В.А. Давыдов, ORCID: 0000-0003-1316-3346 <v.davydov@hse.ru>

Национальный исследовательский университет «Высшая школа экономики», Россия, 101000, г. Москва, ул. Мясницкая, д. 20.

Аннотация. Рассматривается применение кодов, исправляющих ошибки в модульной метрике, для решения задачи идентификации объекта на множестве Q-ичных векторов размерности D методом k-соседей. В качестве предварительной обработки обучающей выборки используется метод кластеризации, использующий процедуру декодирования всех векторов обучающей выборки выбранным кодом в модульной метрике.

Ключевые слова: метод k-ближайших соседей; метрики; кластеризация; коды в модульной метрике; вектор.

Для цитирования: Шарапов А.Р., Давыдов В.А. Применение кодов в модульной метрике для поиска k-соседей. Труды ИСП РАН, том 37, вып. 5, 2025 г., стр. 33–42. DOI: 10.15514/ISPRAS–2025–37(5)–2.

Благодарности. Статья подготовлена в ходе проведения исследования в рамках Программы фундаментальных исследований Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ).

Application of Codes in Modular Metrics for Searching K-Neighbors

A.R. Sharapov, ORCID: 0000-0002-4794-0206 <arsharapov@hse.ru>
V.A. Davydov, ORCID: 0000-0003-1316-3346 <petrov@ispras.ru>

National Research University Higher School of Economics, 11, Pokrovksy Bulvar, Moscow, 109028, Russia.

Abstract. This paper is devoted to the application of suffix codes in the modular metric for solving clustering and k-nearest neighbors (KNN) problems. The advantages of using the modular metric over the Euclidean metric are considered, especially in high-dimensional spaces. The main emphasis is placed on the development of efficient clustering and k-nearest neighbors algorithms using codes that can correct errors in the modular metric. The proposed approach provides polynomial complexity with respect to the training sample dimension, which makes it promising for machine learning applications with large datasets and high-performance requirements.

Keywords: KNN (k-nearest neighbors) method; metrics; clustering; codes in module metric; vector.

For citation: Sharapov A.R., Davydov V.A. Application of codes in modular metrics for searching k-neighbors. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 5, 2025, pp. 33-42 (in Russian). DOI: 10.15514/ISPRAS-2025-37(5)-2.

Acknowledgements. The article was prepared during the research conducted within the framework of the Fundamental Research Program of the National Research University Higher School of Economics (HSE).

1. Введение

Классическая проблема поиска ближайшего соседа формулируется следующим образом: задано подмножество $\boldsymbol{V}$ вещественного векторного пространства размерности D с заданной метрикой ρ и состоящее из N векторов. Также имеется подмножество $\boldsymbol{M}$ векторного пространства $\boldsymbol{V}$ и элемент $\boldsymbol{b} \in \boldsymbol{V}$, требуется найти $\boldsymbol{a} \in \boldsymbol{M}$, ближайший к $\boldsymbol{b}$. Данная проблема и ее решение является центральной проблемой в вычислительной геометрии.

Быстрое вычисление ближайших соседей активно изучается в рамках научных направлений машинного обучения. Наиболее наивная реализация поиска соседей включает в себя грубое вычисление расстояний между всеми парами точек в наборе данных, этот подход имеет сложность $O(DN^2)$. Однако, поскольку количество выборок N растет, подход грубой силы быстро становится невозможным.

Для решения проблемы вычислительной неэффективности подхода грубой силы были изобретены различные древовидные структуры данных. Эффективное кодирование совокупной информации о расстоянии позволяет сократить необходимое количество вычислений расстояния в выборке благодаря использованию древовидных структур данных. Основная идея заключается в том, что если точка A очень далека от точки B, и точка B очень близка к C, то мы знаем, что точки A и C очень далеки, без необходимости явно вычислять их расстояние. Таким образом, вычислительная стоимость поиска ближайших соседей может быть снижена до $O(DN \log N)$ или лучше. Это значительное улучшение по сравнению с грубой силой для больших N.

В литературе также изучается модифицированная задача поиска приблизительного ближайшего соседа. В приблизительном поиске соседа на расстоянии r, структура данных должна сообщать о точке на расстоянии cr от заданной точки $\boldsymbol{b}$ для некоторой константы $c > 1$, но только в случае, если существует точка на расстоянии r от $\boldsymbol{b}$. Будем называть эту проблему (r, c) -ближайшим соседом (nearest neighbor, NN)

В работе [6] представлен LSH алгоритм поиска ближайшего соседа, который использует $O\left(DN^{1+1/c}\right)$ подготовительных операций, объем памяти $O(DN + N^{1+1/c})$ и $O(DN^{1/c})$ операций по поиску ближайшего сосед. Используя уменьшение размерности [7], число операций поиска ближайшего соседа можно дополнительно сократить до $O(D + N^{1/c})$, а сложность предварительной обработки сократить до $O\left(DN + N^{1+1/c}\right)$. Алгоритм LSH успешно использовался в нескольких прикладных сценариях, включая вычислительную биологию [8-9] и ссылки в них или [10: 414].

В работе [1] был представлен новый подход к решению задачи поиска K-ближайших соседей (KNN) [5] для случая модульной метрики, который предлагает использование кодовых конструкций. Предложенный метод свел задачу классификации к процедуре декодирования кодов в модульной метрике, которая была описана в работе [2]. Модульную метрику также называют метрикой Манхэттена или $l1$ метрикой.

Как было отмечено в [1], модульная метрика имеет ряд преимуществ по сравнению с метрикой Евклида для пространств с большой размерностью, что дает дополнительный аргумент для ее использования в задачах классификации на таких пространствах. Метод K-ближайших соседей, как отмечено в [4] является наиболее точным для решения задач классификации по сравнению с альтернативными методами, однако его применение ограничено алгоритмической сложностью, которая у известных вариантов реализации KNN зависит от объема обучающей выборки.

Предположим, что объем обучающей выборки $\boldsymbol{V}$ равен N. Для поиска ближайшего расстояния необходимо перебрать все объекты из обучающей выборки, рассчитать для каждого из них расстояние до тестового объекта $\boldsymbol{f}$ и затем найти минимум. Сложность такого поиска линейная по N и зависит от размерности пространства признаков D.

Предположим, что значение каждого признака имеет определенные границы т.е. минимальное значение и максимальное значение. Разницу между максимальным и минимальными значениями будем называть диапазоном признака. Проведем операцию нормирования каждого признака, то есть приведем минимальное значение к нулю, а максимальное к единице путем вычитания минимального значения из всех значений и деления результата на диапазон. Для перехода к целым числам будем считать, что значения отрезка от нуля до единицы разделены на Z равномерных интервалов. Пусть $\{0, 1, 2, \dots, Z-1\} \equiv Z$ Если $\boldsymbol{f} \in Z^D$, то сложность такого алгоритма поиска $O(ND)$. В типичной задаче машинного обучения количество признаков D может быть порядка 100, а размер выборки может исчисляться десятками и сотнями тысяч объектов. Такая сложность является сдерживающим фактором для реализации метода в приложениях промышленного интернета вещей на устройствах, где нужно малое время реакции системы, энергоэффективность и низкие требования к «железу». Всё это означает, что для решения проблемы KNN возникает необходимость в более быстрых методах поиска ближайших соседей, чем простой перебор.

Переход к кодам в модульной метрике, предложенный в работе [1], дает возможность решения задачи классификации методом KNN с полиномиальной сложностью от размерности обучающей выборки, что открывает новые перспективы для использования KNN в машинном обучении, особенно для больших объемов данных и больших размерностей выборки. Использование кодов в модульной метрике в конкретных приложениях для KNN зависит от того, насколько эффективность теоретического подхода, описанного в работе [1], будет сохранена при практической реализации. Для такой реализации в данной работе предлагается использование суффиксной конструкции кодов в модульной метрике, которая была описана в статье [3]

Сложность решения задачи поиска ближайшего соседа для выборки $\mathbf{V}$ размерности D с заданной метрикой ρ и состоящее из N векторов при больших значениях N может быть существенно снижена при проведении предварительной обработки выборки $\mathbf{V}$**.** В качестве такой обработки может быть использована кластеризация выборки $\mathbf{V}$**,** описание которой приводится в следующем разделе.

2. Задача кластеризации

Задача кластеризации (или обучения без учителя) заключается в следующем. Задано множество $\boldsymbol{V}$ размерности D с заданной метрикой ρ, и состоящее из N векторов. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ, а объекты разных кластеров существенно отличались. При этом каждому объекту $\boldsymbol{a} = (a_1, a_2, \dots, a_D) \in \boldsymbol{V}$ приписывается метка (номер) кластера $\boldsymbol{y_a} \in \boldsymbol{Y}$. Алгоритм кластеризации — это функция $\boldsymbol{V} \to \boldsymbol{Y}$, которая любому объекту $\boldsymbol{a} = (a_1, a_2, \dots, a_D) \in \boldsymbol{V}$ ставит в соответствие метку кластера $\boldsymbol{y_a} \in \boldsymbol{Y}$. Множество меток Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации.

Имеются многочисленные обзоры работ в области кластерного анализ, например, статьи [11-14]. Ляо в 2005 году [16] опубликовал обзор методов кластеризации для данных временных рядов.

Решение задачи кластеризации принципиально неоднозначно, и тому есть несколько причин. Во-первых, не существует однозначно наилучшего критерия качества кластеризации. Известен целый ряд достаточно разумных критериев, а также ряд алгоритмов, не имеющих чётко выраженного критерия, но осуществляющих достаточно разумную кластеризацию «по построению». Все они могут давать разные результаты. Во-вторых, число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным

критерием. В-третьих, результат кластеризации существенно зависит от метрики ρ, выбор которой, как правило, также субъективен и определяется экспертом.

Для решения в дальнейшем задачи поиска ближайшего соседа мы будем использовать такие цели кластеризации как:

- упростить дальнейшую обработку данных и принятия решений, работая с каждым кластером по отдельности;
- сократить объём хранимых данных в случае сверхбольшой выборки $\boldsymbol{V}$, оставив по одному наиболее типичному представителю от каждого кластера.

В первом случае число кластеров стараются сделать поменьше. Во втором случае важнее обеспечить высокую степень сходства объектов внутри каждого кластера, а кластеров может быть сколько угодно.

Во всех этих случаях может применяться иерархическая кластеризация, когда крупные кластеры дробятся на более мелкие, те в свою очередь дробятся ещё мельче и так далее. Такие задачи называются задачами таксономии (taxonomy). Результатом таксономии является не простое разбиение множества объектов на кластеры, а древообразная иерархическая структура. Вместо номера кластера объект характеризуется перечислением всех кластеров, которым он принадлежит, от крупного к мелкому. Мы будем рассматривать алгоритмы иерархической кластеризации, позволяющие автоматизировать процесс построения таксономий.

Задачу кластеризации можно ставить как задачу дискретной оптимизации: необходимо так приписать номера кластеров $\boldsymbol{y_a} \in \boldsymbol{Y}$ каждому объекту $\boldsymbol{a} = (a_1, a_2, \dots, a_D) \in \boldsymbol{V}$, чтобы значение выбранного функционала качества приняло наилучшее значение. Существует много разновидностей функционалов качества кластеризации, но нет «самого правильного» функционала.

Если алгоритм кластеризации вычисляет центры кластеров $\mu(\boldsymbol{y_a})$, $\boldsymbol{y_a} \in \boldsymbol{Y}$, то можно определить функционал суммы средних внутрикластерных расстояний:

$$\Phi_0(\mu, V) = \sum_{y \in Y} \frac{1}{|K_y|} \sum_{a|y_a=y} \rho(a, \mu(y_a)) \to min$$

где $K_y = \{a \in V | y_a = y\}$ — кластер с номером y.

Также можно определить функционал суммы межкластерных расстояний:

$$\Phi(\mu) = \sum_{z_1, z_2 \in \mu(Y)} \rho(z_1, z_2)$$

На практике вычисляют отношение пары функционалов, чтобы учесть, как межкластерные, так и внутрикластерные расстояния:

$$\frac{\Phi_0}{\Phi_1} \to min, \qquad \frac{\Phi_1}{\Phi_0} \to max$$

Будем в дальнейшем считать, что проведена дискретизация компонент векторов $\boldsymbol{a} = (a_1, a_2, \dots, a_D) \in \boldsymbol{V}$ на $\boldsymbol{Z}$ уровней. Это позволяет использовать мощный алгебраический аппарат конечных полей и строить кодовые конструкции, описанные в следующем разделе.

3. Описание суффиксной конструкции кодов в модульной метрике для решения задачи классификации методом KNN

Пусть задано множество $\boldsymbol{V}$ состоящее из N векторов $\boldsymbol{a} = (a_1, a_2, \dots, a_D) \in \mathbf{V}, \boldsymbol{b} = (b_1, b_2, \dots, b_D) \in \boldsymbol{V}$. Каждый вектор множества $\boldsymbol{V}$ состоит из D компонентов над

подмножеством целых чисел $a_i \in \{0,1,2,\dots Z-1\} \equiv \boldsymbol{Z}$, $b_i \in \{0,1,2,\dots Z-1\} \equiv \boldsymbol{Z}$. Расстояние в модульной метрике между векторами $\boldsymbol{a}$ и $\boldsymbol{b}$ определяется по формуле

$$d_M(\boldsymbol{a},\boldsymbol{b}) = \sum_{i=1}^{D} |a_i - b_i|$$

Определим отношение $\boldsymbol{a} \gg \boldsymbol{b}: \{a_i \geq b_i \mid 1 \leq i \leq D\}$ и отношение $\boldsymbol{a} \ll \boldsymbol{b}: \{a_i \leq b_i \mid 1 \leq i \leq D\}$. Если $d_M(\boldsymbol{a},\boldsymbol{b}) = t$ и $\boldsymbol{a} \gg \boldsymbol{b}$ ($\boldsymbol{a} \ll \boldsymbol{b}$) будем говорить, что вектор $\boldsymbol{a}$ получен из вектора $\boldsymbol{b}$ путем t увеличивающих (уменьшающих) вес ошибок в модульной метрике. Такие ошибки будем называть однонаправленными ошибками.

Будем называть подмножество векторов $\mathbb{C} \subset \boldsymbol{V}$ кодом, исправляющим t увеличивающих (уменьшающих) вес ошибок в модульной метрике если выполняются условия $\forall\, \boldsymbol{a} \in \mathbb{C},\ \forall\, \boldsymbol{b} \in \mathbb{C}, \nexists\, \boldsymbol{c} \in \boldsymbol{V}: \boldsymbol{c} \gg \boldsymbol{a}, \boldsymbol{c} \gg \boldsymbol{b}\ (\boldsymbol{c} \ll \boldsymbol{a}, \boldsymbol{c} \ll \boldsymbol{b}), d_M(\boldsymbol{a},\boldsymbol{c}) \leq t, d_M(\boldsymbol{b},\boldsymbol{c}) \leq t$

Пусть задано конечное поле из Q элементов $GF(\mathrm{Q})$ и выполняется условие $\mathrm{Q} > Z$, т.е. мультипликативная группа поля должна содержать не менее Z элементов. Пусть заданы множество различных ненулевых элементов поля $L = \{l_1, l_2, \dots, l_D\} \subset GF(\mathrm{Q})$ которое будем называть множеством локаторов и множество различных ненулевых элементов поля $S = \{s_1, s_2, \dots, s_T\} \subset GF(\mathrm{Q})$ которое будем называть множеством суффиксов. Для $1 \leq t \leq \mathrm{T}$ будем обозначать подмножества мощности $t: S_t = \{s_1, s_2, \dots, s_t\} \subseteq \{s_1, s_2, \dots, s_T\}$. Будем считать, что $L \cap S = \emptyset$. Определим отображение вектора $\boldsymbol{u} = (u_1, u_2, \dots, u_D) \in \boldsymbol{V}$ в локаторный полином $u(x)$

$$u(x) = \mathcal{F}(\boldsymbol{u}) \triangleq \prod_{i=1}^{D} \left(1 - \frac{x}{l_i}\right)^{u_i}$$

Для подмножества суффиксов $S_t = \{s_1, s_2, \dots, s_t\}$ определим суффиксный полином

$$s_t(x) \triangleq x \prod_{i=1}^{t} \left(1 - \frac{x}{s_i}\right)$$

Обозначим $supp\big(s(x)\big)$ – множество корней полинома $s(x)$. Из определения полиномов $u(x)$ и $s_t(x)$ следует, что $supp(u(x)) \cap supp\big(s_t(x)\big) = \emptyset$. Обозначим $f(x)$ фиксированный полином от формальной переменной x с коэффициентами над полем $GF(\mathrm{Q})$, который является остатком некоторого локаторного полинома $u(x)$ с коэффициентами над полем $GF(\mathrm{Q})$ по модулю полинома $s_t(x)$.

$$f(x) = u(x)\ mod\ s_t(x)$$

В работе [3] доказываются две теоремы, важные для дальнейших рассуждений.

Теорема 1.

Код $\mathbb{C}_{\boldsymbol{f}} = \big\{\boldsymbol{u} \in \boldsymbol{V}: u(x) \equiv f(x)\ mod\ s_t(x), f(x) \in \boldsymbol{\mathcal{F}_Q}/s_t(x)\big\}$ исправляет не менее t однонаправленных ошибок в модульной метрике.

Теорема 2.

Код $\boldsymbol{\mathcal{B}_\delta}$

$$\boldsymbol{\mathcal{B}_\delta} = \left\{\boldsymbol{c} \in \mathbb{C}: \sum_{i=1}^{D} |\mathrm{c}_i| \ \equiv\ \delta\ mod\ (2t+1), 0 \leq\ \delta \leq 2t\right\}$$

исправляет не менее t ошибок в модульной метрике.

Из определения суффиксного полинома $s_t(x)$, локаторного полинома $u(x)$, а также китайской теоремы об остатках следует, что соответствующий $f(x)$ однозначно определяется

t вычетами полинома $u(x)$ по модулю полиномов $(1 - xs_i^{-1})$, $1 \le i \le t$. Каждый такой вычет является ненулевым элементом поля $GF(\mathrm{Q})$. Нулевой вычет невозможен, поскольку что $L \cap S = \emptyset$. Будем обозначать для полинома $u(x)$ такие ненулевые вычеты $u_i^* \equiv u(x) \bmod (1 - xs_i^{-1})$, $1 \le i \le t$. Таким образом получаем отображение $\mathfrak{J}$ произвольного вектора $\boldsymbol{u} = (u_1, u_2, \dots, u_D) \in \boldsymbol{V}$ в вектор ненулевых вычетов $\boldsymbol{u}^* = (u_1^*, u_2^*, \dots, u_t^*)$.

$$\mathfrak{J}(\mathbf{u}) \triangleq (u_1^*, u_2^*, \dots, u_t^*)$$

Из утверждения Теоремы 1 следует, что вектору ненулевых вычетов $\boldsymbol{u}^* = (u_1^*, u_2^*, \dots, u_t^*)$ соответствует либо свой код $\mathbb{C}$**,** исправляющий не менее t однонаправленных ошибок в модульной метрике, поскольку каждый $f(x) \in \boldsymbol{\mathcal{F}_Q}/s_t(x)$ однозначно описывается своим вектором вычетов, либо вектору ненулевых вычетов $\boldsymbol{u}^* = (u_1^*, u_2^*, \dots, u_t^*)$ соответствует пустое множество $\emptyset$. Аналогично из утверждения Теоремы 2 следует, что каждому вектору ненулевых вычетов $\boldsymbol{u}^* = (u_1^*, u_2^*, \dots, u_t^*)$ и целому числу $0 \le \delta \le 2t$ соответствует либо свой код $\boldsymbol{\mathcal{B}}$**,** исправляющий не менее t произвольных ошибок в модульной метрике, либо пустое множество $\emptyset$. Сформулируем данные утверждения в виде Следствий.

Следствие из Теоремы 1.

Пусть задан вектор $(u_1^*, u_2^*, \dots, u_t^*), u_i^* \in GF(\mathrm{Q}), u_i^* \ne 0,\ 1 \le i \le t$. Тогда либо $\mathbb{C} = \{\boldsymbol{u} \in \boldsymbol{V}: \mathfrak{J}(\mathbf{u}) \triangleq (u_1^*, u_2^*, \dots, u_t^*)\}$ исправляет не менее t однонаправленных ошибок в модульной метрике, либо $\mathbb{C} \equiv \emptyset$.

Следствие из Теоремы 2.

Пусть задан вектор $(u_1^*, u_2^*, \dots, u_t^*), u_i^* \in GF(\mathrm{Q}), u_i^* \ne 0,\ 1 \le i \le t$ и задано целое число $0 \le \delta \le 2t$. Тогда либо $\boldsymbol{\mathcal{B}} = \{\boldsymbol{u} \in \boldsymbol{V}: \mathfrak{J}(\mathbf{u}) \triangleq (u_1^*, u_2^*, \dots, u_t^*),\ \sum_{i=1}^{D} |u_i| \equiv \delta \bmod (2t + 1)\}$ исправляет не менее t произвольных ошибок в модульной метрике, либо $\boldsymbol{\mathcal{B}} \equiv \emptyset$.

Сформулируем конструкцию для метода KNN с использованием суффиксных кодов в модульной метрике, согласно Следствию из Теоремы 2.

4. Использование суффиксной конструкции кодов в модульной метрике для обработки тестовой выборки

Пусть N объем тестовой выборки $\boldsymbol{V}$. Выборка содержит C различных классов объектов. Каждый вектор $\boldsymbol{V}$ состоит из D компонентов над подмножеством целых чисел $\{0,1,2, \dots Z - 1\} \equiv \boldsymbol{Z}$**.** Такую выборку будем обозначать (N, D, Z, C). Зададим максимальную величину ошибок T, которую будет исправлять кодовая конструкция. Как вариант, можно считать, что $T \approx max\{D, Z\}$. Пусть задано $GF(Q)$ и выполняется неравенство $Q > max\{D + T, Z\}$. Данное условие позволяет обеспечить формирование множества локаторов $L = \{l_1, l_2, \dots, l_D\} \subset GF(\mathrm{Q})$ и суффиксов $S = \{s_1, s_2, \dots, s_T\}$ для которых выполняется условие $L \cap S = \emptyset$.

Будем строить конструкцию кодов, для исправления t ошибок в модульной метрике, где $1 \le t \le \mathrm{T}$. Поставим в соответствие каждой из D позиций векторов $\boldsymbol{v} = (v_1, v_2, \dots, v_D) \in \boldsymbol{V}(N, D, Z, C)$ ненулевой элемент поля из множества локаторов $L = \{l_1, l_2, \dots, l_D\} \subset GF(\mathrm{Q})$.

Для любого $1 \le t \le \mathrm{T}$ зададим отображение $W(\boldsymbol{v}, t)$ вектора $\boldsymbol{v} = (v_1, v_2, \dots, v_D)$ в локаторный полином степени t от формальной переменной $\boldsymbol{x}$ над полем $GF(Q)$ по формуле

$$W(\boldsymbol{v}, t) = \prod_{i=1}^{D} \left(1 - \frac{x}{l_i}\right)^{v_i} mod \left(x \prod_{i=1}^{t} \left(1 - \frac{x}{s_i}\right)\right)$$

Значения s_i выбираются из множества суффиксов $S_t = \{s_1, s_2, \dots, s_t\} \subset \{s_1, s_2, \dots, s_T\}$. Заметим, что младший коэффициент $\boldsymbol{W}(\boldsymbol{v}, t)$ всегда равен 1 для любых значений $t > 1$ и любого вектора $\boldsymbol{v} = (v_1, v_2, \dots, v_D) \in \boldsymbol{V}$. Число различных полиномов $\boldsymbol{W}(\boldsymbol{v}, t)$ не превосходит величины $(\boldsymbol{Q} - \boldsymbol{1})^{\boldsymbol{t}}$.

5. Обработка обучающей выборки путем иерархической кластеризации

Процедура обработки множества слов $\boldsymbol{V}(N,D,Z,C)$ производится последовательно для подмножеств, соответствующих каждому классу из $\boldsymbol{Y}=\{1,2,\dots C\}$. Для дальнейших рассуждений выберем класс номер 1 из множества таких классов $\boldsymbol{Y}=\{1,2,\dots C\}$. Обработка $\boldsymbol{V}(N,D,Z,C)$ для каждого класса осуществляется последовательно для всех значений $1\le t\le T$, начиная с минимального.

Для векторов данного класса из множества слов $\boldsymbol{V}(N,D,Z,C)$ и заданного значения t рассмотрим полином $\boldsymbol{W}(\boldsymbol{a},\boldsymbol{t})$ для каждого $\boldsymbol{a}=(a_1,a_2,\dots,a_D)\in\boldsymbol{V}$ и заданного множества суффиксов $S_t=\{s_1,s_2,\dots,s_t\}\subset\{s_1,s_2,\dots,s_T\}$. Выберем наиболее часто встречающийся вариант. Обозначим число векторов такого варианта n_{1t}. Обозначим такой полином $\boldsymbol{V}(\boldsymbol{x},\boldsymbol{t},\boldsymbol{1})$. Согласно Следствию из Теоремы 2, подмножество слов $\boldsymbol{V}(N,D,Z,C)$, имеющих одинаковый $\boldsymbol{V}(\boldsymbol{x},\boldsymbol{t},\boldsymbol{1})$ является кодом, с исправлением t однонаправленных ошибок в модульной метрике. Будем обозначать такой код $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$.

Проведем процедуру исправления t ошибок во всех векторах $\boldsymbol{V}(N,D,Z,C)$, относящихся к первому классу, чей полином не совпадает с $\boldsymbol{V}(\boldsymbol{x},\boldsymbol{t},\boldsymbol{1})$, т.е. найдем ошибки, которые показывают, насколько слово кода $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$ отличается от слова $\boldsymbol{a}=(a_1,a_2,\dots,a_D)\in\boldsymbol{V}(N,D,Z,C)$. Для этого необходимо последовательно предположить, что число t произошедших ошибок распределилось между ошибками, увеличивающими вес и ошибками, уменьшающими вес, т.е. рассмотреть $2t+1$ вариантов.

Производя последовательно $2t+1$ декодирований, получаем различные варианты кодового вектора, ближайшего к анализируемому слову $\boldsymbol{a}=(a_1,a_2,\dots,a_D)\in\boldsymbol{V}$, не являющегося кодовым. Из полученных вариантов выбираем то кодовое слово, которое находится на минимальном расстоянии к анализируемому вектору $\boldsymbol{a}=(a_1,a_2,\dots,a_D)\in\mathbf{V}(N,D,Z,C)$. В результате вместо вектора $\boldsymbol{a}=(a_1,a_2,\dots,a_D)\in\mathbf{V}(N,D,Z,C)$ для текущего значения $1\le t\le$ T получаем вектор $\boldsymbol{a}^*=(a_1^*,a_2^*,\dots,a_D^*)\in\mathbf{V}_1^*(n_{1t},D,Z,C,t)$. Заметим, что

$$d_M(\boldsymbol{a}^*,\boldsymbol{a})=\sum_{i=1}^{D}|a_i^*-a_i|\le t$$

В одно слово кода $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$ может быть декодировано несколько слов $\mathbf{V}(N,D,Z,C)$, относящихся к классу 1. Для каждого слова $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$ фиксируется сколько слов $\boldsymbol{V}(N,D,Z,C)$ из класса 1 было декодировано в данное слово. Будем называть такое число мощностью слова $\boldsymbol{a}^*=(a_1^*,a_2^*,\dots,a_D^*)\in\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$. Некоторые слова кода, соответствующего полиному $\boldsymbol{V}(\boldsymbol{x},\boldsymbol{t},\boldsymbol{1})$ могут иметь нулевую мощность. Это означает, что в них не было декодировано ни одного слова из $\boldsymbol{V}(N,D,Z,C)$, относящегося к классу 1.

Отметим, что для каждого значения $1\le t\le$ T и для каждого слова кодов $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$ определяется свое значение мощности. Множества суффиксов удовлетворяет условию

$$S_1\subset\cdots\subset S_{t-1}\subset S_t\subset S_{t+1}\subset\cdots\subset S_T$$

Из данного условия следует, что код, исправляющий $t+1$ ошибок, является подмножеством кода с исправлением t ошибок, т.е. выполняется условие $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)\supset\boldsymbol{V}_1^*\left(n_{1,t+1},D,Z,C,t+1\right)$, то каждое слово кода $\boldsymbol{V}_1^*\left(n_{1,t+1},D,Z,C,t+1\right)$ имеет список из $t+1$ мощностей для всех кодов, в которые данное слово входит.

Поскольку множество слов $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$ является кодом, исправляющим однонаправленные t ошибок, возможна ситуация, когда один и тот же вектор $(a_1,a_2,\dots,a_D)\in\boldsymbol{V}(N,D,Z,C)$ декодируется сразу в несколько слов $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$. Это возможно, если произошли разнонаправленные ошибки. В этом случае будем выбирать то кодовое слово из $\boldsymbol{V}_1^*(n_{1t},D,Z,C,t)$, расстояние до которого меньше, а если несколько слов кода находятся на одинаковом расстоянии – выбирать кодовое слово с большей мощностью.

Процедуру, описанную выше, проведем для всех классов из множества таких классов $\boldsymbol{Y} = \{1, 2, \dots C\}$. В результате для каждого значения $1 \le t \le T$ получим множества $\boldsymbol{V}_1^*(n_{1t}, D, Z, C, t)$, $\boldsymbol{V}_2^*(n_{2t}, D, Z, C, t)$, … $\boldsymbol{V}_C^*(n_{Ct}, D, Z, C, t)$. Объединение данных множеств дает адаптированную выборку $\boldsymbol{V}^*(N^*, D, Z, C, t)$. Другими словами, выполняется условие

$$\boldsymbol{V}^*(N^*, D, Z, C, t) = \bigcup_{i=1}^{C} \boldsymbol{V}_i^*(n_{it}, D, Z, C, t)$$

Заметим, что для числа слов полученных кодов с ненулевой мощностью для каждого $1 \le t \le T$ выполняется неравенство.

$$N \ge \sum_{i=1}^{C} n_{it} = N^*$$

Выбор наиболее часто встречающегося варианта полинома $\boldsymbol{V}(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{1})$ эмпирически определяет наиболее подходящий код, как код, для которого надо исправлять ошибки в меньшем числе слов. Другими словами, у такого кода максимальное число кодовых слов совпало с числом векторов $\boldsymbol{V}(N, D, Z, C)$, относящимся к первому классу объектов. Выбор может произведен и по другим критериям. Например, по минимальному расстоянию от выбранных кодовых слов до ближайших векторов $\boldsymbol{V}(N, D, Z, C)$.

В дальнейшем будем для определения k ближайших соседей для вектора $\boldsymbol{f} = (f_1, f_2, \dots, f_D)$ использовать $\boldsymbol{V}_1^*(n_{1t}, D, Z, C, t)$, $\boldsymbol{V}_2^*(n_{2t}, D, Z, C, t)$, … $\boldsymbol{V}_C^*(n_{Ct}, D, Z, C, t)$ для каждого значения $1 \le t \le T$, последовательно переходя от меньших значений t к большим. Признаком остановки будет являться получение кодовых слов с суммарной мощностью не менее *k*.

6. Поиск ближайших k соседей к новому объекту

Выбираем начальное значение $t = 1$ и для каждого класса из множества классов $\mathbf{Y} = \{1,2, \dots C\}$, проведем декодирование вектора $\boldsymbol{f} = (f_1, f_2, \dots, f_D)$ в ближайшие кодовые слова кодов $\boldsymbol{V}_1^*(n_{1t}, D, Z, C, t)$, $\boldsymbol{V}_2^*(n_{2t}, D, Z, C, t)$, … $\boldsymbol{V}_C^*(n_{Ct}, D, Z, C, t)$.

Полученные слова каждого кода имеют свою мощность (т.е. число слов исходной обучающей выборки, которые были декодированы в данное слово). Если сумма мощностей таких слов больше или равна k, то выбранный параметр t достаточен. Если сумма мощностей меньше k – необходимо увеличить параметр до t+1 и повторить действие алгоритма для кодов $\boldsymbol{V}_1^*(n_{1,t+1}, D, Z, C, t+1)$, $\boldsymbol{V}_2^*(n_{2,t+1}, D, Z, C, t+1)$, … $\boldsymbol{V}_C^*(n_{C,t+1}, D, Z, C, t+1)$.

Для классифицируемого объекта $\boldsymbol{f} = (f_1, f_2, \dots, f_D)$ выбирается тот класс из множества $\boldsymbol{Y} = \{1,2, \dots C\}$, для которого мощность полученного кодового слова больше. Если два слова разных классов имеют одинаковую мощность – то вычисляется расстояние в модульной метрике от вектора $\boldsymbol{f} = (f_1, f_2, \dots, f_D)$ до полученных кодовых слов, имеющих одинаковую мощность, и выбирается тот класс, расстояние до которого меньше.

В результате принятия решения по классу нового объекта, увеличивается мощность того кодового слова, в которое был декодирован классифицируемый вектор $\boldsymbol{f} = (f_1, f_2, \dots, f_D)$. Далее алгоритм поиска ближайших k соседей повторяется для классификации нового вектора и так далее.

7. Заключение

Предложенная конструкция по использованию суффиксных кодов в модульной метрике для решения задачи иерархической кластеризации и задачи поиска k соседей позволяет реализовать решение данных задач с полиномиальной сложностью от размерности обучающей выборки. Конструкция, при выборе достаточно большого значения T не требует повторной процедуры обработки обучающей выборки при переходе к большему значению k при решении задачи поиска k соседей.

Список литературы / References

[1]. В. А. Давыдов. Использование кодов в модульной метрике для решения задачи KNN. (в публикации).

[2]. В. А. Давыдов. Коды, исправляющие ошибки в модульной метрике, метрике Ли и ошибки оператора // Пробл. передачи информ. 1993. Глава 29:3. С. 209–217

[3]. V. Davydov, N. Zeulin, I. Pastushok, A. Turlikov. Coding Scheme for High-Order QAM Modulations in the Manhattan Metric // IEEE Communications Letters. 2020. Vol. 24. N. 11. P. 2387–2391.

[4]. T. Cover, P. Hart. Nearest neighbor pattern classification // IEEE Transactions on Information Theory. 1967. Vol. 13. N. 1. P. 21–27.

[5]. Fix. E, Hodges, J.L. Discriminatory analysis. Nonparametric discrimination; consistency properties // USAF School of Aviation Medicine. 1951. Technical Report 4.

[6]. P. Indyk, R. Motwani. Approximate nearest neighbor: towards removing the curse of dimensionality // Proceedings of the Symposium on Theory of Computing. 1998.

[7]. E. Kushilevitz, R. Ostrovsky, Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces // Proceedings of the Thirtieth ACM Symposium on Theory of Computing. 1998. P. 614– 623.

[8]. J. Buhler, M. Tompa. Finding motifs using random projections // Proceedings of the Annual International Conference on Computational Molecular Biology. 2001.

[9]. J. Buhler. Provably sensitive indexing strategies for biosequence similarity search // Proceedings of the Annual International Conference on Computational Molecular Biology. 2002.

[10]. N. C. Jones, P. A. Pevzner. An Introduction to Bioinformatics Algorithms // The MIT Press Cambridge. 2004.

[11]. M. Namratha. IOSR Journal of Computer Engineering. 2012. Vol. 4(6). P 23–30.

[12]. A. V. Kumar, J. C. Selvaraj. Journal of Recent Research and Applied Studies. 2016. Vo_103.

[13]. M. Omran, A. Engelbrecht, A. A. Salman. Intelligent Data Analysis. 2007. Vol. 11(6). P. 583–605.

[14]. M. Wegmann, D. Zipperling, J. Hillenbrand, J. Fleischer. A review of systematic selection of clustering algorithms and their evaluation. 2021.

[15]. J. Gu. Journal of Physics: Conference Series. 2021. Vol. 1.

[16]. T. W. Liao. Pattern Recognition. 2005. Vol. 38(11). P. 1857–1874.

[17]. T. Gupta, S. Panda. International Journal of Engineering & Technology. 2018. Vol. 7(4). P. 4766–4768.

Информация об авторах / Information about authors

Александр Рауилович ШАРАПОВ – является аспирантом московского института электроники и математики им. А.Н. Тихонова, департамент электронной инженерии. Темой исследования является «Разработка метода анализа данных с использованием кодов, исправляющих ошибки в метрике l1». Сфера научных интересов: статистика, анализ данных, кодирование информации, сети и телекоммуникации.

Alexander Rauilovich SHARAPOV is a postgraduate student at the Moscow Institute of Electronics and Mathematics named after A.N. Tikhonov, Department of Electronic Engineering. The topic of his research is "Development of a method for analyzing data using codes that correct errors in the l1 metric". His research interests include statistics, data analysis, information coding, networks and telecommunications.

Вячеслав Анатольевич ДАВЫДОВ – кандидат технических и экономических наук. Он является приглашенным преподавателем московского института электроники и математики им. А.Н. Тихонова, кафедра информационной безопасности киберфизических систем. Сфера научных интересов: алгебраическая и комбинаторная теория кодирования, теория активных систем, технология блокчейн.

Vyacheslav Anatolyevich DAVYDOV – Cand. Sci. (Tech., Econ.). He is a visiting lecturer at the Moscow Institute of Electronics and Mathematics named after A.N. Tikhonov, Department of Information Security of Cyber-Physical Systems. His research interests include algebraic and combinatorial coding theory, active systems theory, blockchain technology.