ЧЕБЫШЕВСКИЙ СБОРНИК
Том 20. Выпуск 2.

УДК 519.72 DOI 10.22405/2226-8383-2019-20-2-93-107

Новый подход к поиску строковой медианы и визуализация строковых кластеров[1]

Д. В. Горбачев, Е. П. Офицеров

Горбачев Дмитрий Викторович — доктор физико-математических наук, профессор, кафедра прикладной математики и информатики, Тульский государственный университет (г. Тула).
e-mail: dvgmail@mail.ru
Офицеров Евгений Петрович — кафедра прикладной математики и информатики, Тульский государственный университет (г. Тула).
e-mail: eofitserov@gmail.com

Аннотация

Рассматривается следующая задача о поиске медианы набора строк:

$$c = \operatorname*{argmin}_{a \in U} \sum_{b \in D} d(a, b),$$

где $D \subset G^*$ — конечный набор строк над алфавитом G, $U \subseteq G^*$, d — редакционное расстояние Левенштейна. Эта задача имеет важные приложения, например в биоинформатике при анализе белковых последовательностей. Однако известно, что в общем случае для $U = G^*$ задача о медиане является NP-сложной. Поэтому для приближенного решения были предложены эвристические алгоритмы, в частности, жадный алгоритм. Мы предлагаем новый гибкий подход, базирующийся на гладкой аппроксимации расстояния Левенштейна $\tilde{d}$. В его основе лежит стохастическое кодирование символьных последовательностей и следующая формула для редакционного расстояния:

$$d(X_1, X_2) = \min_{(X_1', X_2'')_e \subseteq X_1 \times X_2} \left\{ \frac{1}{2} \|X_1' - X_2''\|_1 + |X_1| - |X_1'| + |X_2| - |X_2''| \right\},$$

где минимум берется по всем подпоследовательностям $(X_1', X_2'')_e$ равной длины. С одной стороны, стохастическое кодирование расширяет класс, на котором ищется экстремум. Однако наш основной результат показывает, что медиана не меняется. С другой стороны, теперь мы можем воспользоваться гладкими методами оптимизации, если заменить минимум в определении выше его гладким приближением. Мы приводим детали реализации на основе градиентного спуска, включая рекуррентные формулы расчета $\tilde{d}$. Эффективный расчет приближенной медианы позволяет, например, применить метод k-средних для кластеризации строк. Мы даем способ визуализации этих кластеров на основе метода стохастического вложения соседей tSNE.

Ключевые слова: редакционное расстояние Левенштейна, строковая медиана, гладкий минимум, градиентный спуск, метод k-средних, визуализация строкового кластера.

Библиография: 19 названий.

[1]Результаты исследования опубликованы при финансовой поддержке ТулГУ в рамках научного проекта №:НИР_2018_28.

Для цитирования:
Д. В. Горбачев, Е. П. Офицеров. Новый подход к поиску строковой медианы и визуализация строковых кластеров // Чебышевский сборник, 2019, т. 20, вып. 2, с. 93–107.

CHEBYSHEVSKII SBORNIK
Vol. 20. No. 2.

UDC 519.72 DOI 10.22405/2226-8383-2019-20-2-93-107

New approach to searching for string median and visualization of string clusters[2]

D. V. Gorbachev, E. P. Ofitserov

Gorbachev Dmitry Viktorovich — Doctor of physical and mathematical sciences, Professor, Department of Applied Mathematics and Computer Science, Tula State University (Tula).
e-mail: dvgmail@mail.ru
Ofitserov Evgenii Petrovich — Department of Applied Mathematics and Computer Science, Tula State University (Tula).
e-mail: eofitserov@gmail.com

Abstract

We consider the following problem of searching the median of a string set:

$$c = \operatorname*{argmin}_{a \in U} \sum_{b \in D} d(a, b),$$

where $D \subset G^*$ is the finite set of strings over the alphabet G, $U \subseteq G^*$, d is the Levenshtein edit distance. This problem has important applications, e.g., in bioinformatics in the analysis of protein sequences. However, it is known that in the general case for $U = G^*$ the median problem is NP-hard. Therefore, for an approximate solution, heuristics algorithms were proposed, in particular, the greedy algorithm. We give a new flexible approach based on smooth Levenshtein distance approximation $\tilde{d}$. It uses stochastic character encoding of sequences and the following formula for the edit distance:

$$d(X_1, X_2) = \min_{(X_1', X_2'')_e \subseteq X_1 \times X_2} \left\{ \frac{1}{2} \|X_1' - X_2''\|_1 + |X_1| - |X_1'| + |X_2| - |X_2''| \right\},$$

where the minimum is taken over all subsequences $(X_1', X_2'')_e$ of equal length. On the one hand, stochastic coding extends the class, over which the extremum is searched. However, our main result shows that the median is the same. On the other hand, now we can use smooth optimization methods, replacing the minimum in the definition above with a smooth approximation. We give implementation details based on the gradient descent, including the recursion formulas for calculating $\tilde{d}$. Effective calculation of the approximate median allows, e.g., to use the k-means method for string clustering. We propose an approach to visualize such clusters based on tSNE stochastic nesting method.

Keywords: Levenshtein edit distance, string median, smooth minimum, gradient descent, k-means, string cluster visualization.

Bibliography: 19 titles.

For citation:
D. V. Gorbachev, E. P. Ofitserov, 2019, "New approach to searching for string median and visualization of string clusters" , *Chebyshevskii sbornik*, vol. 20, no. 2, pp. 93–107.

[2]The results of the study are published with the financial support of Tulsu within the framework of the scientific project №:НИР_2018_28.

1. Введение

В этой работе мы приведем теоретическое обоснование метода гладкой аппроксимации редакционного расстояния Левенштейна в задаче поиска строковой медианы. Эти результаты продолжают исследования второго автора, связанные с применением методов дифференцируемого выравнивания в задачах машинного обучения [2, 3, 4, 14, 5]. Мы приведем основные факты численной реализации методами математического программирования. Также дается способ визуализации строкового кластера и его приближенной медианы.

Пусть $n \in \mathbb{N}$, $G_n = \{g_1, \ldots, g_n\}$ — алфавит, $|G| = n$ — число элементов, G_n^L — множество строк или последовательностей $a = a[1] \ldots a[L]$ над G_n длины $L = |a| \geqslant 0$ (при $L = 0$ имеем пустую строку λ), $G_n^* = \bigcup_{L \geqslant 0} G_n^L$.

Через $d(a_1, a_2)\colon G_n^* \times G_n^* \to \mathbb{Z}_+$ обозначим дискретное редакционное расстояние Левенштейна. Оно определяется как минимальное количество операций замены, вставки и удаления символа, необходимых для преобразования строки a_1 в строку a_2 [1, гл. 11]. Свойства d и разные варианты подсчета приводятся в пункте 2.1.

Строковой медианой или центроидом конечного набора строк $D \subseteq G_n^*$ на подмножестве $U \subseteq G_n^*$ называется строка $c = c(D, U) \in U$, для которой [12]

$$c = \operatorname*{argmin}_{a \in U} \sum_{b \in D} d(a, b). \tag{1}$$

Основными являются случаи $U = D$ и $U = G_n^*$. Назовем медиану $c(D, D)$ локальной, а $c(D, G_n^*)$ — глобальной.

Дискретная экстремальная задача (1) имеет большое прикладное значение (см., например, результаты, упомянутые в [9, 12, 10]). Однако, несмотря на то, что сложность вычисления расстояния Левенштейна по алгоритму Вагнера–Фишера полиномиальная (см. пункт 2.1), доказано, что задача поиска глобальной медианы набора D будет NP-сложной, если количество и длины строк в D могут быть произвольными. Для родственной задачи о множественном выравнивании этот факт был установлен в [19]. NP-сложность задачи (1) в работе [9] доказана при помощи сведения к NP-сложной задаче о нахождении длиннейшей общей подпоследовательности строк из D.

Были предложены различные методы для приближенного нахождения медианы. Самый простой способ состоит в поиске локальной медианы (см., например, [12]). Это задача сложности $O(L_{\max}^2 |D|^2)$, где $L_{\max}$ — максимальная длина строки из D. Однако результаты [12] показывают, что $c(D, D)$ может заметно отличаться от глобальной медианы $c(D, G_n^*)$.

В [8] был предложен жадный алгоритм приближенного поиска глобальной медианы (см. пункт 2.2). Он прост в реализации, быстр и дает неплохое приближение, которое имеет смысл использовать для сравнения методов. Сложность жадного алгоритма $O(L_{\max}^2 |D||G|)$ за итерацию (которых не более $2L_{\max}$ по лемме 1). Были предложены варианты этого алгоритма (см., например, [11]). Отметим вариант получения приближенной глобальной медианы модификацией локальной медианы, имеющий сложность $O(L_{\max}^3 |D||G|)$ за итерацию [12]. В [10] проблема (1) сводится к задаче целочисленного программирования.

В работе [15] применяется вероятностный подход на основе представления символов стохастическими векторами. Нам потребуется такое представление, поэтому напомним, что вектор $w = (w_1, \ldots, w_n)$ называется стохастическим (вероятностным) или весовым, если $w_1, \ldots, w_n \in [0, 1]$ и $w_1 + \ldots + w_n = 1$. Множество таких векторов обозначим через W_n.

Мы отсылаем к упомянутым выше работам за более подробным обзором проблемы.

В данной работе предлагается метод приближенного вычисления медианы на основе стохастического кодирования последовательностей и гладкой аппроксимацией расстояния Левенштейна $\tilde{d}$. Мы предполагаем, что максимальная длина $L_{\max}$ подпоследовательности из D много меньше $|D|$. Подробнее идея состоит в следующем.

Пусть $E_n = \{e_1, \dots, e_n\}$, где $e_j = (0, \dots, 0, 1, 0 \dots, 0) \in \{0,1\}^n$ — бинарный вектор с единицей на j-м месте, $j = \overline{1,n}$. Закодируем символ $g_j \in G_n$ вектором e_j. Тогда произвольная последовательность $a = a[1] \dots a[L]$ над алфавитом G_n биективно отображается в бинарную последовательность $B = B[1] \dots B[L]$ над алфавитом E_n.

Непустую последовательность B также можно интерпретировать как бинарную прямоугольную матрицу со строками $B[i] = (B[i,1], \dots, B[i,n])$, $i = \overline{1,L}$, и элементами

$$B[i,j] = \begin{cases} 1, & \text{если } a[i] = g_j, \\ 0 & \text{иначе,} \end{cases} \qquad j = \overline{1,n}.$$

В итоге имеем $E_n^* \cong G_n^*$ и можно не делать различия между бинарными последовательностями (матрицами) и символьными последовательностями.

Метрику Левенштейна можно перенести на бинарные матрицы из E_n^* с помощью следствия 1 из пункта 2.1:

$$d(B_1, B_2) = \min_{(B_1', B_2'')_e \subseteq B_1 \times B_2} \Big\{ \frac{1}{2} \|B_1' - B_2''\|_1 + |B_1| - |B_1'| + |B_2| - |B_2''| \Big\}. \tag{2}$$

Тогда задача о строковой медиане набора $D \subseteq E_n^*$ на подмножестве $U \subseteq E_n^*$ эквивалентно записывается как

$$S(D,U) = \min_{A \in U} \sum_{B \in D} d(A,B), \quad C(D,U) \in \arg S(D,U). \tag{3}$$

Множество E_n является множеством крайних точек W_n. Следуя [15] вложим дискретное множество E_n^* в непрерывное пространство W_n^* последовательностей (матриц) вида $X = X[1] \dots X[L]$, где $X[i]$ — стохастический вектор:

$$X[i] \in [0,1]^n, \quad \sum_{j=1}^{n} X[i,j] = 1, \quad i = \overline{1,L}, \quad L = |X| \geqslant 0$$

($L = 0$ отвечает пустой матрице Λ). Будем называть такие матрицы или последовательности весовыми.

Округлением $\operatorname{round}(w)\colon E_n \to W_n$ стохастического вектора $w \in W_n$ называется бинарный вектор e_j, в котором единица стоит на месте максимального w_j (если таких несколько, то выбирается последний). Соответственно, округлением $\operatorname{round}(X)\colon W_n^* \to E_n^*$ весовой матрицы X называется бинарная матрица, которая состоит из округлений строк X.

Формула (2) справедлива для произвольных весовых матриц $X_1, X_2 \in W_n^*$. Поэтому задачу (3) можно рассмотреть на более широком множестве $U = W_n^*$. Следующая основная теорема показывает, что минимум не изменится и решение (3) можно восстановить, применяя округление.

Теорема 1. *Для любого конечного набора $D \subseteq E_n^*$ имеем*

$$S(D, E_n^*) = S(D, W_n^*), \quad C(D, E_n^*) = \operatorname{round}(C(D, W_n^*)).$$

Вычисление $S(D, W_n^*)$ сводится к решению серии задач $\min \sum_{B \in D} d(X,B)$, где $X \in W_n^*$ и $|X|$ фиксировано. Пространство W_n^* непрерывно, однако функция $X \mapsto \sum_{B \in D} d(X,B)$ не везде дифференцируема из-за взятия минимума в (2). Это не позволяет применить эффективные методы математического программирования, в частности, градиентного спуска. Чтобы обойти этот момент мы воспользуемся гладкой аппроксимацией $\tilde{d}$ расстояния Левенштейна, основанной на применении гладкого минимума. Для эффективного вычисления функции $\tilde{d}$ и

ее градиента предлагаются рекуррентных формулы, являющиеся аналогами формулы динамического программирования для d. Данные результаты были получены вторым автором в работе [14].

Структура работа следующая. В разделе 2 приведены основные утверждения относительно редакционного расстояния Левенштейна: необходимые для понимания классические результаты (пункт 2.1), сведения о жадном алгоритме (пункт 2.2), доказательство основной теоремы 1 (пункт 2.3). В разделе 3 даются основные факты для дифференцируемого редакционного расстояния (ДРР) $\tilde{d}$: определение ДРР (пункт 3.1), рекуррентные формулы для расчета (пункт 3.2), детали градиентного спуска (пункт 3.3). В разделах 4 и 5 приведены краткие сведения о визуализации кластеров строк и возможном развитии темы соответственно.

2. Основные утверждения

2.1. Редакционное расстояние Левенштейна

Мы воспользуемся хорошо известными результатами, связанными с расстоянием Левенштейна d, взятыми из [1, гл. 11].

Для строки $a = a[1]a[2]\ldots a[L] \in G_n^*$ через

$$a[\mathbf{i}] = a[i_1, \ldots, i_\ell] = a[i_1]\ldots a[i_\ell] \subseteq a, \quad 1 \leqslant i_1 < \ldots < i_\ell \leqslant L,$$

обозначается подпоследовательность длины ℓ (пустая при $\ell = 0$ или $L = 0$), определяемая индексами $\mathbf{i} = i_1, \ldots, i_\ell$, $a[i\!:\!j] = a[i]a[i+1]\ldots a[j]$ — подстрока от индекса i до j (пустая при $i > j$), $a[1\!:\!i]$ — префикс длины $i = \overline{0, L}$.

Пусть далее $a_1 = a_1[1]\ldots a_1[L_1]$, $a_2 = a_2[1]\ldots a_2[L_2]$ и

$$\delta(i, j) = d(a_1[1\!:\!i], a_2[1\!:\!j]), \quad i = \overline{0, L_1}, \quad j = \overline{0, L_2}.$$

Для вычисления расстояния Левенштейна применяется следующая рекуррентная формула:

$$d(a_1, a_2) = \delta(L_1, L_2),$$

где $\delta(i, 0) = i$, $\delta(0, j) = j$ и при $i, j \geqslant 1$

$$\delta(i, j) = \min\{\delta(i, j-1) + 1, \delta(i-1, j) + 1, \delta(i-1, j-1) + \theta\}, \quad \theta = \begin{cases} 1, & a_1[i] \neq a_2[j], \\ 0, & a_1[i] = a_2[j]. \end{cases} \tag{4}$$

Эта формула лежит в основе алгоритма динамического программирования Вагнера–Фишера и позволяет за $O(L_1 L_2)$ операций вычислить $\delta(L_1, L_2)$, а также оптимальное редакционное предписание — список длины $d(a_1, a_2)$ необходимых замен, вставок и удалений.

Расстояние Левенштейна удовлетворяет всем аксиомам метрики и

$$||a_1| - |a_2|| \leqslant d(a_1, a_2) \leqslant \max\{|a_1|, |a_2|\}. \tag{5}$$

Можно дать другое выражение для расстояния Левенштейна, используя термин выравнивание последовательностей [1, гл. 11]. Глобальным выравниванием двух строк называется вставка в строки пробелов (в том числе и на их концах) до равной длины так, чтобы i-й символ или пробел первой строки отвечал i-му символу или пробелу второй строки для всех $i = 1, 2, \ldots$ (пара пробел–пробел не допускается). Приведем пример выравнивания из [1] для строк `vintner` и `writers`:

$$\begin{pmatrix} \mathtt{v} & \mathtt{_} & \mathtt{i} & \mathtt{n} & \mathtt{t} & \mathtt{n} & \mathtt{e} & \mathtt{r} & \mathtt{_} \\ \mathtt{w} & \mathtt{r} & \mathtt{i} & \mathtt{_} & \mathtt{t} & \mathtt{_} & \mathtt{e} & \mathtt{r} & \mathtt{s} \end{pmatrix}.$$

По выравниванию легко получить редакционное предписание и наоборот: совпадающая пара символ–символ пропускается, несовпадающая пара символ–символ отвечает замене, пара пробел–символ — вставке и символ–пробел — удалению. Для примера выше имеем предписание длины 5, которое является оптимальным.

Произвольное выравнивание последовательностей a_1, a_2 можно построить, задав список пар символ–символ. Символы этих пар образуют подпоследовательности одинаковой длины. Для примеры выше имеем `viter` и `witer`. Оставшиеся символы будут отвечать пробелам. В связи с этим введем следующее определение (где «e» в индексе означает equal length).

Определение 1. (1) *Соответствием* $(a_1', a_2'')_e \subseteq a_1 \times a_2$ *последовательностей* a_1, a_2 *называется произвольная пара подпоследовательностей* $a_1' = a_1[i_1', \ldots, i_\ell']$, $a_2'' = a_2[i_1'', \ldots, i_\ell'']$ *равной длины* $\ell = |a_1'| = |a_2''|$, *где* $0 \leqslant \ell \leqslant \min\{L_1, L_2\}$ ($\ell = 0$ *отвечает пустому соответствию). Здесь символ* $a_1[i_s']$ *первой последовательности отвечает символу* $a_2[i_s'']$ *второй последовательности,* $s = \overline{1, \ell}$.

(2) *Также положим* $a_1' = a_1'[1] \ldots a_1'[\ell]$, $a_2'' = a_2''[1] \ldots a_2''[\ell]$, *где* $a_1'[s] = a_1[i_s']$, $a_2''[s] = a_2[i_s'']$.

(3) *Соответствие* $(a_1', a_2'')_e$ *однозначно определяется индексами подпоследовательностей* $(\mathbf{i}', \mathbf{i}'')_e \in \overline{1, L_1} \times \overline{1, L_2}$, *где* $\mathbf{i}' = i_1', \ldots, i_\ell'$, $\mathbf{i}'' = i_1'', \ldots, i_\ell''$ *(для пустого соответствия имеем пустой набор индексов).*

В выравнивании, построенном по соответствию $(a_1', a_2'')_e$, $L_1 - \ell$ символов первой строки, не вошедших в соответствие, должны отвечать пробелам второй строки, а $L_2 - \ell$ символов второй строки — пробелам первой. Например, для соответствия $(\texttt{vine}, \texttt{rite})_e$ имеем выравнивание

$$\begin{pmatrix} \texttt{_} & \texttt{v} & \texttt{i} & \texttt{n} & \texttt{t} & \texttt{n} & \texttt{e} & \texttt{r} & \texttt{_} & \texttt{_} \\ \texttt{w} & \texttt{r} & \texttt{i} & \texttt{t} & \texttt{_} & \texttt{_} & \texttt{e} & \texttt{_} & \texttt{r} & \texttt{s} \end{pmatrix}.$$

Несмотря на то, что пробелы расставляются неоднозначно, длина предписания будет всегда равна 8. Она складывается из 2 различий в соответствии (замены), 3 пробелов вверху (вставки) и 3 пробелов внизу (удаления).

Таким образом, в общем случае длина предписания, отвечающего соответствию

$$(a_1', a_2'')_e \subseteq a_1 \times a_2,$$

будет содержать $|a_1| - |a_1'|$ позиций для вставок и $|a_2| - |a_2''|$ для удалений. При этом число замен определяется числом различий в подпоследовательностях a_1', a_2'', которое можно задать с помощью расстояния Хемминга

$$d_H(a_1', a_2'') = |\{i \colon a_1'[i] \neq a_2''[i],\ i = \overline{1, \ell}\}|.$$

(равно нулю при $\ell = 0$). В итоге, получаем следующее

Предложение 1. *Расстояние Левенштейна между строками* $a_1, a_2 \in G_n^*$ *равно*

$$d(a_1, a_2) = \min_{(a_1', a_2'')_e \subseteq a_1 \times a_2} \left\{ d_H(a_1', a_2'') + |a_1| - |a_1'| + |a_2| - |a_2''| \right\}.$$

Отметим, что мы не встречали это утверждение в литературе.

Предложение 1 следующим образом обобщается на бинарное представление последовательностей.

Следствие 1. *Пусть* $B_1, B_2 \in E_n^*$. *Тогда*

$$d(B_1, B_2) = \min_{(B_1', B_2'')_e \subseteq B_1 \times B_2} \left\{ \frac{1}{2} \|B_1' - B_2''\|_1 + |B_1| - |B_1'| + |B_2| - |B_2''| \right\}, \tag{6}$$

где для $\ell = |B_1'| = |B_2''|$

$$\|B_1' - B_2''\|_1 = \sum_{i=1}^{\ell} \left\|B_1'[i] - B_2''[i]\right\|_1 = \sum_{i=1}^{\ell}\sum_{j=1}^{n} \left|B_1'[i,j] - B_2''[i,j]\right|$$

(нуль при $\ell = 0$*).*

Доказательство. Достаточно показать, что $\|B_1' - B_2''\|_1 = 2d_H(B_1', B_2'')$. Это легко следует из следующего равенства для строк матриц:

$$\left\|B_1'[i] - B_2''[i]\right\|_1 = \begin{cases} 0, & B_1'[i] = B_2''[i], \\ 2, & B_1'[i] \neq B_2''[i], \end{cases} \quad i = \overline{1,\ell}.$$

Оно верно, так как в каждом из бинарных векторов $B_1'[i]$, $B_2''[i]$ только по одной единице. □

Следствие 2. (1) *Формула* (6) *позволяет определить редакционное расстояние* $d(X_1, X_2)$ *для произвольных весовых матриц* $X_1, X_2 \in W_n^*$.

(2) *Функция* $d(X_1, X_2)$ *по-прежнему удовлетворяет неравенствам* (5).

Доказательство. Часть 1 следствия очевидна.

Часть 2 вытекает из (6). Пусть для определенности $|X_1| \leqslant |X_2|$. Соответствием максимальной длины будет (X_1, X_2''), $|X_1| = |X_2''|$, поэтому

$$d(X_1, X_2) \leqslant \frac{1}{2}\|X_1 - X_2''\|_1 + |X_1| - |X_1| + |X_2| - |X_2''| \leqslant \frac{1}{2}\left(\|X_1\|_1 + \|X_2''\|_1\right) + |X_2| - |X_2''|.$$

Имеем

$$\|X_1\|_1 = \sum_{i=1}^{|X_1|}\sum_{j=1}^{n} |X_1[i,j]| = \sum_{i=1}^{|X_1|} 1 = |X_1|,$$

аналогично $\|X_2''\|_1 = |X_2''|$. Поэтому

$$d(X_1, X_2) \leqslant |X_2''| + |X_2| - |X_2''| = |X_2|.$$

С другой стороны, для любых соответствий

$$\frac{1}{2}\|X_1' - X_2''\|_1 + |X_1| - |X_1'| + |X_2| - |X_2''| \geqslant |X_1| + |X_2| - 2|X_1'| = |X_2| - |X_1|,$$

откуда следует, что $d(X_1, X_2) \geqslant |X_2| - |X_1|$ и, аналогично, $d(X_1, X_2) \geqslant |X_1| - |X_2|$. □

Для доказательства теоремы 1 нам потребуется следующая лемма (где «$\cdot$» обозначает скалярное произведение векторов).

Предложение 2. *Пусть* $X \in W_n^*$, $B \in E_n^*$. *Тогда*

$$d(X, B) = |X| + |B| - \max_{(\mathbf{i},\mathbf{i}'')_e \subseteq \overline{1,|X|}\times\overline{1,|B|}} \sum_{i=1}^{|X|} \tilde{B}_{\mathbf{i}',\mathbf{i}''}[i] \cdot X[i], \tag{7}$$

где $\mathbf{i}' = i_1', \ldots, i_\ell'$, $\mathbf{i}'' = i_1'', \ldots, i_\ell''$ *— индексы соответствий,* $0 \leqslant \ell \leqslant \min\{|X|, |B|\}$,

$$\tilde{B}_{\mathbf{i}',\mathbf{i}''}[i,j] = \begin{cases} B[i_s'', j] + 1, & i = i_s',\ s \in \overline{1,\ell}, \\ 0 & \textit{иначе}, \end{cases} \quad j = \overline{1,n}.$$

Доказательство. Используя индексы $(\mathbf{i}', \mathbf{i}'')_e$ для задания соответствия

$$(X', B'')_e, \quad X' = X[\mathbf{i}'], \quad B'' = B[\mathbf{i}''],$$

по формуле (6) находим

$$d(X, B) = \min_{(\mathbf{i}', \mathbf{i}'')_e} \Big\{ \frac{1}{2} \|X' - B''\|_1 + |X| - |X'| + |B| - |B''| \Big\}.$$

Учтем, что X — весовая матрица, а B — бинарная. Поэтому $X[i'_s, j] \in [0, 1]$,

$$\sum_{j=1}^{n} X[i'_s, j] = 1$$

и $B[i''_s] = e_{j_s}$ для некоторого j_s. Отсюда следует, что

$$\frac{1}{2} \|X' - B''\|_1 - |X'| - |B''| = \frac{1}{2} \sum_{s=1}^{\ell} \sum_{j=1}^{n} \big| X[i'_s, j] - B[i''_s, j] \big| - 2\ell =$$

$$= \frac{1}{2} \sum_{s=1}^{\ell} \Big(\sum_{j \neq j_s} X[i'_s, j] + 1 - X[i'_s, j_s] \Big) - 2\ell = \frac{1}{2} \sum_{s=1}^{\ell} (2 - 2X[i'_s, j_s]) - 2\ell = - \sum_{s=1}^{\ell} (1 + X[i'_s, j_s]) =$$

$$= - \sum_{s=1}^{\ell} \Big(\sum_{j=1}^{n} X[i'_s, j] + \sum_{j=1}^{n} B[i''_s, j] X[i'_s, j] \Big) = - \sum_{s=1}^{\ell} \sum_{j=1}^{n} X[i'_s, j] (B[i''_s, j] + 1) =$$

$$= - \sum_{i=1}^{|X|} \sum_{j=1}^{n} X[i, j] \tilde{B}_{\mathbf{i}', \mathbf{i}''}[i, j] = - \sum_{i=1}^{|X|} \tilde{B}_{\mathbf{i}', \mathbf{i}''}[i] \cdot X[i].$$

Таким образом,

$$\frac{1}{2} \|X' - B''\|_1 + |X| - |X'| + |B| - |B''| = |X| + |B| - \sum_{i=1}^{|X|} \tilde{B}_{\mathbf{i}', \mathbf{i}''}[i] \cdot X[i],$$

откуда следует утверждение леммы. □

Как хорошо известно максимум конечного числа линейных функций — выпуклая кусочно-линейная функция (см. [7, пример 3.5]). Отсюда и из предложения 2 немедленно вытекает

Следствие 3. *Для каждого $i = \overline{1, L}$ функция $X[i] \mapsto d(X[1] \dots X[i] \dots X[L], B)$ на множестве W_n является выпуклой (вниз) кусочно-линейной функцией.*

2.2. Жадный алгоритм

Жадный алгоритм поиска строковой медианы был предложен в работе [8]. Напомним, что решается дискретная задача оптимизации

$$c = \operatorname*{argmin}_{a \in G_n^*} F(a), \quad F(a) = \sum_{b \in D} d(a, b), \quad D \subseteq G_n^*.$$

Для известного a функция $F(a)$ вычисляется эффективно по алгоритму Вагнера–Фишера.

Лемма 1. *Пусть $\bar{L}_D = \frac{1}{|D|} \sum_{b \in D} |b|$ — средняя длина строки в наборе D. Тогда для длины медианы справедлива оценка*

$$|c| \leqslant 2\bar{L}_D. \tag{8}$$

Доказательство. Для пустой строки имеем $F(\lambda) = \sum_{b\in D} |b|$.

Если $|c| > \frac{2}{|D|}\sum_{b\in D}|b|$, то в силу (5) находим

$$F(c) = \sum_{b\in D} d(c,b) \geqslant \sum_{b\in D} ||c|-|b|| \geqslant \Big|\sum_{b\in D}(|c|-|b|)\Big| = \Big||D||c| - \sum_{b\in D}|b|\Big| > \sum_{b\in D}|b| = F(\lambda).$$

Поэтому c не может быть медианой и (8) верно. □

Шаги жадного алгоритма. Начинаем с присваиваний $c_0 = \lambda$ и $F_0 = F(c_0) = \sum_{b\in D}|b|$.

Далее для $k = 0, 1, \ldots$ решается задача

$$c_{k+1} = \operatorname{argmin}\{F(a)\colon a = c_k g,\ g \in G\}, \quad F_{k+1} = F(c_{k+1}),$$

т.е. перебором по $g \in G_n$ к текущей строке c_k добавляется символ g и среди таких строк $a = c_k g$ вычисляются минимальное значение F_{k+1} функции $F(a)$ и соответствующая оптимальная строка c_{k+1}. Сложность этого шага $O((k+1)L_{\max}|G||D|)$, где $L_{\max} = \max_{b\in D}|b|$.

Если $F_{k+1} < \min_{i=\overline{0,k}} F_i$, то итерации продолжаются и c_{k+1} новое приближение медианы. Иначе алгоритм останавливается и строка c_k ответ.

Число итераций в силу леммы 1 ограничено $2\bar{L}_D$.

2.3. Доказательство теоремы 1

Пусть $D, U \subseteq E_n^*$, $F_D(X) = \sum_{B\in D} d(X,B)$. Напомним, что в матричной постановке медиана $C = C(D,U)$ определяется из решения дискретной задачи

$$S(D,U) = \min_{X\in U} F_D(X), \quad C = C(D,U) \in \arg S(D,U).$$

Покажем, что

$$S(D,E_n^*) = S(D,W_n^*).$$

По лемме 1 с учетом следствия 2 находим

$$\min_{X\in U} F_D(X) = \min_{0\leqslant L\leqslant 2\bar{L}_D} \min_{\substack{X\in U\\ |X|=L}} F_D(X), \tag{9}$$

поэтому достаточно доказать

$$S(D,E_n^L) = S(D,W_n^L). \tag{10}$$

При $L = 0$ это равенство очевидно и реализуется на пустой матрице. Поэтому пусть $L \geqslant 1$.

Очевидно, что минимум в непрерывной задаче $S(D,W_n^L)$ достигается на некоторой весовой матрице $\hat{X} \in W_n^L$. Этот минимум можно искать следующим образом:

$$S(D,W_n^L) = \min_{X[1]\in W_n} \ldots \min_{X[L]\in W_n} F_D(X)$$

В силу следствия 3 функция $X[i] \mapsto F_D(X[1]\ldots X[i]\ldots X[L])$ выпуклая на множестве W_n. Поэтому ее минимум достигается в некоторой крайней точке (бинарном векторе) $\hat{X}[i] = e_{j_0} \in E_n$. Это доказывает (10).

В общем случае минимум может также достигаться в выпуклой комбинации крайних точек $\hat{X}[i] = \sum_{s=1}^m w_{j_s} e_{j_s}$, $1 \leqslant m \leqslant n$. Тогда по определению округления вектор $\operatorname{round}(\hat{X}[i])$ совпадает с одним из e_{j_s} и также дает решение. Следовательно, $\operatorname{round}(\hat{X})$ — решение $S(D,E_n^L)$. Теорема 1 доказана.

3. Гладкая аппроксимация расстояния Левенштейна

Детали гладкой аппроксимации расстояния Левенштейна подробно даны в работе [14]. Здесь мы приведем основные факты.

3.1. Гладкое расстояние

По теореме 1 медиану $C \in E_n^*$ дискретного набора $D \subset E_n^*$ можно искать в непрерывном пространстве весовых матриц W_n^*. Для этого с учетом (9) для каждого $0 \leqslant L \leqslant 2\bar{L}_D$ ищется медиана C_L заданной длины L:

$$C_L = \operatorname{round}(\hat{C}_L), \quad \hat{C}_L = \operatorname*{argmin}_{X \in W_n^L} F_D(X), \tag{11}$$

где, напомним, $F_D(X) = \sum_{B \in D}^m d(X, B)$. Глобальная медиана C определяется как лучшая из медиан C_L.

Для задачи оптимизации на непрерывном пространстве W_n^L естественно воспользоваться гладкими методами оптимизации. Однако функция $X \mapsto d(X, B)$ в силу следствия (3) является только кусочно-дифференцируемой. Поэтому в [14] было предложено заменить в определении (6) минимум на его гладкую аппроксимацию, называемую гладким минимумом. Этот прием известен в выпуклой оптимизации [7].

Для конечного множества $Y \subset \mathbb{R}$ гладкий τ-минимум определяется равенством

$$\operatorname{smin}(Y; \tau) = \frac{\sum_{y \in Y} y e^{\tau y}}{\sum_{y \in Y} e^{\tau y}}, \tag{12}$$

где $\tau < 0$ — параметр аппроксимации (при $\tau > 0$ имеем гладкий максимум). Нетрудно показать, что

$$\min Y \leqslant \operatorname{smin}(Y; \tau) \leqslant \min Y + C e^{\tau \Delta}, \tag{13}$$

где $C = |Y|(\operatorname{mean} Y - \min Y)$ и $\Delta = \min_{y' \neq y'' \in Y} |y' - y''|$. Отсюда следует, что гладкий минимум стремится к обычному экспоненциально быстро. Далее параметр τ часто опускается, если это не вызывает недоразумений.

Определение 2. *Пусть $X_1, X_2 \in W_n^*$, $\tau < 0$. Гладкой аппроксимацией расстояния Левенштейна называется функция*

$$\tilde{d}(X_1, X_2; \tau) = \operatorname*{smin}_{(X_1', X_2'')_e \subseteq X_1 \times X_2} \Big\{ \frac{1}{2} \|X_1' - X_2''\|_1 + |X_1| - |X_1'| + |X_2| - |X_2''| \Big\}. \tag{14}$$

Из определения гладкого минимума следует, что функция $\tilde{d}(X_1, X_2)$ будет дифференцируемой по аргументам, понимая под градиентом функции $\|X\|_1 = \sum_{s=1}^L \sum_{t=1}^n |X[s,t]|$ матричную функцию

$$\nabla \|X\|_1 = (\operatorname{sign} X[s,t]), \quad s = \overline{1, L}, \quad t = \overline{1, n}. \tag{15}$$

Поэтому $\tilde{d}$ также называется дифференцируемым редакционным расстоянием (ДРР).

ДРР обладает свойствами симметричности и неотрицательности. Однако, в отличии от оригинального редакционного расстояния, $\tilde{d}(X, X) \neq 0$. Поэтому также вводится следующее

Определение 3. *Несмещенным ДРР называется функция*

$$\tilde{d}_0(X_1, X_2) = \tilde{d}(X_1, X_2) - \frac{1}{2} \big(\tilde{d}(X_1, X_1) + \tilde{d}(X_2, X_2) \big).$$

Из (13) следует

Предложение 3. *При $\tau \to -\infty$ экспоненциально быстро $\tilde{d}_0(X_1, X_2; \tau) \to d(X_1, X_2)$.*

3.2. Рекуррентные формулы

Расчет ДРР $\tilde{d}$ напрямую по формуле (14) затруднителен из-за перебора соответствий, число которых растет экспоненциально с ростом $|X_1|$ и $|X_2|$. Напомним, что для классического редакционного расстояния d данная проблема решается при помощи рекуррентной формулы (4) и основанной на ней алгоритме динамического программирования Вагнера–Фишера, который позволяет вычислить d за полиномиальное время.

В [14] аналогичные рекуррентные формулы установлены для $\tilde{d}$. Пусть $X_1, X_2 \in W_n^*$, $|X_1| = L_1$, $|X_2| = L_2$, $i = \overline{0, L_1}$, $j = \overline{0, L_2}$, Ω_{ij} — множество всевозможных соответствий $(X_1', X_2'')_e \in X_1[1:i] \times X_2[1:j]$ и для таких соответствий длины $k = |X_1'| = |X_2''|$

$$R_{ij}(X_1', X_2'') = \frac{1}{2}\|X_1' - X_2''\|_1 + i + j - 2k.$$

Положим

$$\alpha_{ij} = \alpha_{ij}(X_1, X_2) = \sum_{(X_1', X_2'')_e \in \Omega_{ij}} R_{ij}(X_1', X_2'') e^{\tau R_{ij}(X_1', X_2'')},$$

$$\beta_{ij} = \beta_{ij}(X_1, X_2) = \sum_{(X_1', X_2'')_e \in \Omega_{ij}} e^{\tau R_{ij}(X_1', X_2'')},$$

что отвечает числителю и знаменателю в гладком минимуме (12). Тогда

$$\tilde{d}(X_1, X_2) = \frac{\alpha_{L_1 L_2}}{\beta_{L_1 L_2}}.$$

Теорема 2 ([14]). *Для* $i = \overline{1, L_1}$, $j = \overline{1, L_2}$

$$\alpha_{ij} = (\alpha_{i-1,j} + \beta_{i-1,j} + \alpha_{i,j-1} + \beta_{i,j-1})e^{\tau} + (\alpha_{i-1,j-1} + \beta_{i-1,j-1}\delta_{ij})e^{\tau\delta_{ij}} - (\alpha_{i-1,j-1} + 2\beta_{i-1,j-1})e^{2\tau},$$

$$\beta_{ij} = (\beta_{i-1,j} + \beta_{i,j-1})e^{\tau} + \beta_{i-1,j-1}(e^{\tau\delta_{ij}} - e^{2\tau}),$$

где

$$\alpha_{i0} = ie^{\tau i}, \quad \alpha_{0j} = je^{\tau j}, \quad \beta_{i0} = e^{\tau i}, \quad \beta_{0j} = e^{\tau j}, \quad \delta_{ij} = \frac{1}{2}\|X_1[i] - X_2[j]\|_1.$$

Рекуррентные формулы из теоремы 2 дают эффективный способ вычисления $\tilde{d}(X_1, X_2)$ за полиномиальное время. Дифференцирование этих формул с учетом (15) позволяет получить рекуррентные соотношения для эффективного расчета градиента $\nabla_{X_1}\tilde{d}(X_1, X_2)$. Отсюда получаем значение и градиент несмещенного ДРР.

3.3. Градиентный спуск

Рассмотрим задачу приближенного вычисления медианы C_L на множестве W_n^L (см. (11)). Из определения W_n^L имеем задачу условной оптимизации. Возможность вычисления градиента несмещенного ДРР позволяет воспользоваться методом проекции градиента.

В качестве начального приближения X_0 берется случайная матрица из W_n^L. Очередная итерация имеет вид

$$X_{k+1/2} = X_k - \gamma_k \frac{1}{|D|} \sum_{B \in D} \nabla_X(\tilde{d}_0(X, B))(X_k), \quad X_{k+1} = \mathrm{Pr}\,(X_{k+1/2}), \tag{16}$$

где $\gamma_k > 0$ — шаг метода и проекция $\mathrm{Pr}\colon \mathbb{R}^{L\times n} \to W_n^L$ определяется равенством

$$\mathrm{Pr}\,(Y) = \operatorname*{argmin}_{X \in W_n^L} \|Y - X\|_2^2.$$

Для практических приложений важен случай, когда число строк $|D|$ велико. В этом случае можно применить стохастический градиентный спуск, когда в (16) вместо D берется случайная подвыборка из D много меньшего размера [16].

Было проведено сравнение предложенного подхода с жадным алгоритмом. Для этого проведен вычислительный эксперимент на синтетических данных. Генерировалось 100 наборов случайных строк D над алфавитом из 4 букв длинной от 5 до 20 символов. Далее на этом множестве выполнялся поиск медианы двумя алгоритмами. Оценивалась величина

$$\delta(D) = \sum_{B \in D} (d(C, B) - d(C_g, B)),$$

где C — медиана, найденная предложенным алгоритмом, а C_g — жадным. Было установлено, что для наборов из 10 строк $\operatorname{mean} \delta = -0.23$, $\operatorname{std} \delta = 0.034$ на уровне значимости $p = 0.99$. Это означает, что в данном примере точность алгоритма чуть выше жадного. Однако с увеличением числа строк они сравниваются по точности, поэтому требуется дальнейшее исследование проблемы.

4. Визуализация кластеров строк

Одно из интересных и важных приложений строковой медианы является визуализация строковых кластеров. Эта задача имеет прикладное значение, например, в биоинформатике. При этом часто достаточно знать медиану приближенно, например, при визуализации больших данных.

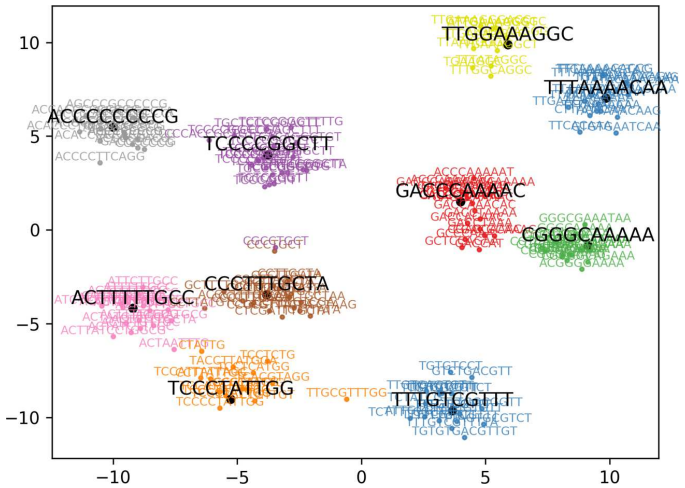
Рассмотрим данную задачу. Имеется набор строк S. Требуется разбить S на непересекающиеся кластеры $S = \bigcup_{k=1}^{K} S_k$, так чтобы строки каждого кластера были близки по редакционному расстоянию к соответствующим медианам $c(S_k)$. В случае векторных данных для решения этой задачи применяется популярный алгоритм k-средних (см., например, [6, гл. 3]), позволяющий находить заданное число кластеров. Он основан на нахождении центроидов. Для строк центроидом является медиана. Поэтому для адаптации метода k-средних к строкам важно эффективно находить медиану, пусть приближенно. Результаты данной работы позволяют это сделать. Отметим еще алгоритм k-медоидов, в котором среднее ищется среди точек кластера. В случае строк это будет отвечать локальной медиане. Однако этот метод имеет свои ограничения, в частности, он требует знания матрицы расстояний $(d(B_i, B_j))_{i,j=1}^{|S|}$. Кроме того, как уже отмечалось, локальная медиана может быть далека от глобальной, что может привести к смещению кластеров.

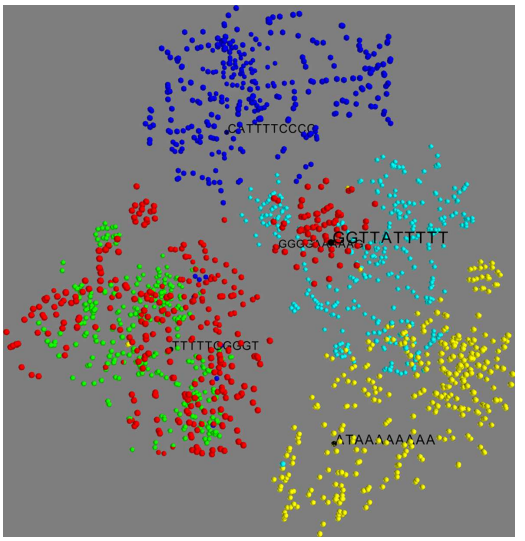
Чтобы проанализировать получившиеся кластеры, интересно их визуализировать. Основная проблема состоит в большой размерности данных. Стандартным подходом в этом случае является проекция данных в пространство низкой размерности (два или три) с сохранением метрических свойств (см. классические работы [18, 17]).

Популярным методом визуализации, активно используемом в машинном обучении, является метод стохастического вложения соседей с t-распределением (tSNE), предложенный в работе [13]. Данный метод позволяет визуализировать многомерные данные путем их нелинейной проекции на пространство малой размерности. В отличии от классических линейных алгоритмов снижения размерности [18] данный метод лучше отображает структуру многомерных данных. Он позволяет рассчитать координаты проекций строк на основе матрицы парных расстояний. В данном случае вычисление такой матрицы не является проблемой, поскольку для визуализации достаточно использовать небольшие подвыборки кластеров.

На рисунке приведен пример визуализации кластеров строк. Он сделан в Python с помощью класса TSNE из пакета машинного обучения Sklearn. Двумерный график (слева) получен функцией scatter из пакета Matplotlib, предназначенного для построения 2D-графиков. Для

вывода трехмерного графика (справа) применялась функция points3d из пакета научной 3D-визуализации Mayavi.





5. Заключение

В работе предложено теоретическое обоснование нового подхода для поиска строковой медианы с помощью гладких методов оптимизации. Основным преимуществом данного подхода по сравнению с существующими дискретными методами является его гибкость. Это позволяет применять его в задачах поиска медианы с ограничениями, а также во многих других связанных проблемах строковой оптимизации. Благодаря тому, что ДРР является диффренцируемой функцией ее можно использовать для сравнения последовательностей в составе различных дифференцируемых моделях машинного обучения, включая нейронные сети. Такие задачи строковой оптимизации в том числе играют важную роль при визуализации строковых данных. Планируется исследовать перечисленные направления.

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. Гасфилд Д. Строки, деревья и последовательности в алгоритмах: Информатика и вычислительная биология. СПб.: Невский Диалект, БХВ-Петербург, 2003.

2. Офицеров Е.П. Статистическая модель периферической селекции Т-клеточных рецепторов // Известия ТулГУ. Технические науки. 2017. Вып. 2. С. 138–143.

3. Офицеров Е.П. Глубокая модель селекции Т-клеточных рецепторов // Известия ТулГУ. Технические науки. 2017. Вып. 12. Ч. 2. С. 350–355.

4. Офицеров Е.П. Классификация последовательностей на основе коротких мотивов. Чебышевский сборник. 2018. Том 19, №:1. С. 187–199. doi: 10.22405/2226-8383-2018-19-1-187-199

5. Офицеров Е.П. Программный комплекс для решения задач машинного обучения на строковых данных при помощи дифференцируемого редакционного расстояния // Известия ТулГУ. Технические науки. 2019. Вып. 5. С. 370–376.

6. Ту Дж., Гонсалес Р. Принципы распознавания образов. М.: Мир, 1978.

7. Boyd S., Vandenberghe L. Convex Optimization. Cambridge University Press, 2004.

8. Casacuberta F., de Antonio M. A greedy algorithm for computing approximate median strings // In: VII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes. 1997. P. 193–198.

9. De la Higuera C., Casacuberta F. Topology of strings: Median string is NP-complete // Theoretical Computer Science. 2000. Vol. 230, no. 1–2. P. 39–48. doi: 10.1016/s0304-3975(97)00240-5

10. Hayashida M., Koyano H. Finding median and center strings for a probability distribution on a set of strings under Levenshtein distance based on integer linear programming // In: Fred A., Gamboa H. (eds) Biomedical Engineering Systems and Technologies. BIOSTEC 2016. Communications in Computer and Information Science. Springer, Cham. 2017. Vol 690. P. 108–121.

11. Kruzslicz F. Improved greedy algorithm for computing approximate median strings // Acta Cybernetica. 1999. Vol. 14, no. 2. P. 331–339.

12. Martínez-Hinarejos C.D., Juan A., Casacuberta F. Use of median string for classification // Proc. 15th Int. Conf. on Pattern Recognition. ICPR-2000, Barcelona, Spain. 2000. Vol. 2. P. 903–906. doi: 10.1109/ICPR.2000.906220

13. Van der Maaten L., Hinton G. Visualizing data using t-SNE // J. Mach. Learn. Res. 2008. Vol. 9. P. 2579–2605.

14. Ofitserov E., Tsvetkov V., Nazarov V. Soft edit distance for differentiable comparison of symbolic sequences // arXiv:1904.12562. 2019.

15. Olivares-Rodríguez C., Oncina J. A stochastic approach to median string computation // In: da Vitoria Lobo N. et al. (eds) Structural, Syntactic, and Statistical Pattern Recognition. SSPR /SPR 2008. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 2008. Vol. 5342. P. 431–440.

16. Ruder S. An overview of gradient descent optimization algorithms // arXiv:1609.04747. 2016.

17. Sammon J.W. A nonlinear mapping for data structure analysis // IEEE Transactions on Computers, 18(5):401–409, 1969.

18. Torgerson W.S. Multidimensional scaling I: Theory and method // Psychometrika. 1952. Vol. 17. P. 401–419.

19. Wang L., Jiang T. On the complexity of multiple sequence alignment // J. Computat. Biol. 1994. Vol. 1, no. 4. P. 337–348.

REFERENCES

1. Gusfield D. Algorithms on strings, trees, and sequences: Computer science and computational biology. New York: Cambridge University Press, 1997.

2. Ofitserov E.P. Statistical model of T-cell receptor pereferic selection // Izvestiya of Tula State University. Technical sciences. 2017. No. 2. P. 138–143.

3. Ofitserov E.P. Deep model of T-cell receptor selection // Izvestiya of Tula State University. Technical sciences. 2017. No. 12, part 2. P. 350–355.

4. Ofitserov E.P. Motif based sequence classification // Chebyshevskii Sbornik. 2018. Vol. 19, no. 1. P. 187–199. (In Russ.) https://doi.org/10.22405/2226-8383-2018-19-1-187-199

5. Ofitserov E.P. Software package for solving machine learning problems on string data using soft edit distance // Izvestiya of Tula State University. Technical sciences. 2019. No. 5. P. 370–376.

6. Tou J.T., Gonzalez R.C. Pattern recognition principles. Addison-Wesley Publishing Company, 1974.

7. Boyd S., Vandenberghe L. Convex Optimization. Cambridge University Press, 2004.

8. Casacuberta F., de Antonio M. A greedy algorithm for computing approximate median strings // In: VII Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes. 1997. P. 193–198.

9. De la Higuera C., Casacuberta F. Topology of strings: Median string is NP-complete // Theoretical Computer Science. 2000. Vol. 230, no. 1–2. P. 39–48. doi: 10.1016/s0304-3975(97)00240-5

10. Hayashida M., Koyano H. Finding median and center strings for a probability distribution on a set of strings under Levenshtein distance based on integer linear programming // In: Fred A., Gamboa H. (eds) Biomedical Engineering Systems and Technologies. BIOSTEC 2016. Communications in Computer and Information Science. Springer, Cham. 2017. Vol 690. P. 108–121.

11. Kruzslicz F. Improved greedy algorithm for computing approximate median strings // Acta Cybernetica. 1999. Vol. 14, no. 2. P. 331–339.

12. Martínez-Hinarejos C.D., Juan A., Casacuberta F. Use of median string for classification // Proc. 15th Int. Conf. on Pattern Recognition. ICPR-2000, Barcelona, Spain. 2000. Vol. 2. P. 903–906. doi: 10.1109/ICPR.2000.906220

13. Van der Maaten L., Hinton G. Visualizing data using t-SNE // J. Mach. Learn. Res. 2008. Vol. 9. P. 2579–2605.

14. Ofitserov E., Tsvetkov V., Nazarov V. Soft edit distance for differentiable comparison of symbolic sequences // arXiv:1904.12562. 2019.

15. Olivares-Rodríguez C., Oncina J. A stochastic approach to median string computation // In: da Vitoria Lobo N. et al. (eds) Structural, Syntactic, and Statistical Pattern Recognition. SSPR /SPR 2008. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 2008. Vol. 5342. P. 431–440.

16. Ruder S. An overview of gradient descent optimization algorithms // arXiv:1609.04747. 2016.

17. Sammon J.W. A nonlinear mapping for data structure analysis // IEEE Transactions on Computers, 18(5):401–409, 1969.

18. Torgerson W.S. Multidimensional scaling I: Theory and method // Psychometrika. 1952. Vol. 17. P. 401–419.

19. Wang L., Jiang T. On the complexity of multiple sequence alignment // J. Computat. Biol. 1994. Vol. 1, no. 4. P. 337–348.

Получено 18.05.2019 г.
Принято в печать 12.07.2019 г.