

О. И. Нечаева

(Новосибирск)

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ
НЕЙРОСЕТЕВЫХ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ
СИМВОЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ***

Рассматриваются алгоритмы кластеризации символьных последовательностей с использованием расстояния Левенштейна, основанные на методе k -средних и нейронной сети Кохонена. Приводятся описание и сравнительная характеристика по времени и точности двух эвристических алгоритмов нахождения ядра в кластерах: алгоритма выбора ядра из кластера и алгоритма подсчета символов. Оценивается возможность применения метода k -средних с переходом к частотным словарям для кластеризации различных типов символьных последовательностей. Сравниваются по времени и качеству алгоритмы кластеризации векторов в евклидовом пространстве методом k -средних и с помощью нейронной сети Кохонена.

Введение. Процесс кластеризации представляет собой разбиение множества объектов на подмножества, называемые кластерами, элементы которых близки между собой в некотором смысле. В качестве объектов могут выступать, например, символьные последовательности либо векторы в евклидовом пространстве, характеризующие некоторые показатели в той или иной предметной области.

Кластеризация применяется в самых разнообразных областях, например: в медицине для систематики и диагностики заболеваний по известным симптомам, в археологии для установления таксономии различных находок, в музыке для выявления структуры мелодии или установления плагиата, в распознавании изображений и сигналов, в маркетинговых исследованиях и т. д. Одной из важнейших областей применения кластеризации является обработка генетических последовательностей, например выделение генов из длинных последовательностей ДНК [1].

Существуют различные подходы к кластеризации множеств объектов. Примерами могут служить: кластеризация, основанная на определении плотности объектов в некотором пространстве [2]; иерархическая кластеризация с поэтапным построением дерева кластеров [3]; кластеризация, основанная не на рассмотрении локальных связей между объектами, а на итера-

* Работа поддержана «Программой фундаментальных исследований РАН» (грант № 17, 2004).

тивной минимизации глобального критерия [4]. Один из важнейших подходов к кластеризации основан на использовании нейросетевых алгоритмов, к которым относятся «итерационный метод динамических ядер» [5], называемый также методом k -средних, нейронная сеть Кохонена [6], самоорганизующиеся карты свойств [6].

В представленной работе рассматриваются алгоритмы кластеризации символьных последовательностей методом k -средних с использованием расстояния Левенштейна [7] при двух способах вычисления ядра в кластере. Для кластеризации большого числа длинных последовательностей рассматривается также подход, основанный на переходе от символьных последовательностей к векторам евклидова пространства, называемым «частотными словарями» этих последовательностей [8]. Поиск ядра в евклидовом пространстве выполняется путем нахождения «среднего вектора», а также с помощью построения нейронной сети Кохонена [6]. Для всех этих методов разработаны алгоритмы и проведен ряд численных экспериментов. Целью исследования является определение областей параметров кластеризуемых последовательностей, в которых тот или иной алгоритм наиболее эффективен по времени и качеству.

Результаты исследования, во-первых, позволяют судить о том, какой из вышеперечисленных методов наиболее подходит для кластеризации последовательностей, характеризующих, например, нотный текст, с помощью которой выявляется глобальная структура мелодии в музыкальном произведении; во-вторых, иллюстрируют утверждение в [9] о том, что метод перехода к частотным словарям применим для достаточно длинных символьных последовательностей, какими являются генетические последовательности.

1. Задача кластеризации и алгоритм ее решения методом k -средних.

Пусть Y – это некоторый алфавит символов, Y^* – множество всевозможных последовательностей символов из Y . Рассмотрим множество $X \subseteq Y^*$ из m символьных последовательностей $X = \{x^1, \dots, x^m\}$, где $x^i = [y_1^i, y_2^i, \dots, y_{N_i}^i]$ ($y_j^i \in Y$, N_i – длина i -й последовательности); m – количество последовательностей в X .

Чтобы проводить кластеризацию множества символьных последовательностей, нужно определить меру близости между его элементами. Для символьных последовательностей одной из наиболее распространенных мер близости является расстояние Левенштейна [7].

Расстоянием Левенштейна $d(x^1, x^2)$ между двумя последовательностями $x^1 = [y_1^1, y_2^1, \dots, y_{N_1}^1]$ и $x^2 = [y_1^2, y_2^2, \dots, y_{N_2}^2]$ называется неотрицательное целое число, равное минимальному количеству операций вставок, удалений и замен символов в x^1 , необходимых, чтобы из x^1 получить x^2 . Заметим, что $d(x^1, x^2)$ определено для любой пары элементов из Y^* , симметрично и удовлетворяет неравенству треугольника для любых x^1 и x^2 . Вычислить расстояние Левенштейна можно, например, с помощью алгоритма Вагнера – Фишера из работы [10], который основан на методе динамического программирования. Его суть состоит в том, что вычисляются расстояния между все более и более длинными префиксами двух последовательностей на основании уже вычисленных до получения окончательного результата. В процессе вычисления расстояния между последовательностями $[y_1^1, \dots, y_i^1, \dots, y_{N_1}^1]$ и $[y_1^2, \dots, y_j^2, \dots, y_{N_2}^2]$ заполняется массив $\{d_{i,j}\}$ размера $(N_1 + 1) \times (N_2 + 1)$ по

рекуррентной формуле:

$$d_{i,j} = \begin{cases} i, & j=0; \\ j, & i=0; \\ \min \{d_{i-1,j} + 1, d_{i,j-1} + 1, d_{i-1,j-1} + \delta'(y_i^1, y_j^2)\}, & i>0, j>0, \end{cases} \quad (1)$$

где $d_{i,j} = d([y_1^1, y_2^1, \dots, y_i^1], [y_1^2, y_2^2, \dots, y_j^2])$, а $\delta'(\cdot, \cdot)$ – это отрицание символа Кронекера, т. е. $\delta'(y_i^1, y_j^2) = 1$, если $y_i^1 \neq y_j^2$, и $\delta'(y_i^1, y_j^2) = 0$ в противном случае. Искомым расстоянием Левенштейна между x^1 и x^2 является число, равное последнему элементу массива $\{d_{i,j}\}$, т. е. $d(x^1, x^2) = d_{N_1, N_2}$.

Задача кластеризации множества X заключается в нахождении разбиения этого множества на подмножества P_1, \dots, P_k , называемые *кластерами*, такие что $P_1 \cup P_2 \cup \dots \cup P_k = \{x^1, \dots, x^m\}$, $k \leq m$, $P_i \cap P_j = \emptyset$, $i \neq j$, и элементы каждого кластера P_i , $i=1, \dots, k$, близки между собой. Наиболее распространенным методом решения этой задачи является метод k -средних [5]. Для его описания введем следующие понятия.

Ядром кластера P_i называется последовательность $a^i \in Y^*$, которая удовлетворяет условию

$$a^i = \arg \min_{a \in Y^*} \sum_{x \in P_i} d(a, x), \quad i=1, \dots, k. \quad (2)$$

Следует заметить, что a^i является аналогом центра тяжести кластера P_i и играет роль его «типичного представителя». Введение ядра позволяет формализовать понятие близости элементов кластера и сводит распределение последовательностей по кластерам к сравнению их с ядрами всех кластеров и выбору ближайшего.

Математическая постановка метода k -средних заключается в том, чтобы найти набор ядер a^1, \dots, a^k и разбить множество X на кластеры $\{x^1, \dots, x^m\} = P_1 \cup P_2 \cup \dots \cup P_k$, минимизирующие величину, которая далее называется *критерием качества*:

$$D = \sum_{i=1}^k \sum_{x \in P_i} d(a^i, x). \quad (3)$$

Из формулы (3) видно, что в результате кластеризации множество X разобьется на такие подмножества-кластеры P_1, \dots, P_k , что для каждого элемента $x \in P_i$ расстояние от x до ядра a^i кластера P_i меньше расстояний до ядер остальных кластеров.

Заметим, что изначально может быть неизвестно число кластеров, так как отсутствует априорная информация о данных и их природе. Поэтому обычно k выбирается исходя из условий конкретной задачи. После разбиения в случае необходимости нужно осуществить слияние некоторых кластеров и уже с новым числом кластеров повторить кластеризацию.

Далее приведены основные шаги итерационного алгоритма кластеризации методом k -средних. Обозначим $a^i(n)$ ядро на итерации с номером n .

А л г о р и т м 1.

0. Фиксируется начальный набор ядер $a^1(0), \dots, a^k(0)$ произвольно либо по какому-нибудь эвристическому правилу, и полагается $P_i = \emptyset, i=1, \dots, k$.

1. На итерации с номером n для каждой последовательности $x^j, j=1, \dots, m$, вычисляются расстояния до всех k ядер, и выбирается среди них минимальное: $d(x^j, a^i(n)) = \min \{d(x^j, a^1(n)), \dots, d(x^j, a^k(n))\}$. Последовательность x^j добавляется в кластер P_i . В случае, когда минимум достигается при нескольких значениях i , выбор между ними может быть сделан произвольно. В результате получается некоторое предварительное разбиение P_1, \dots, P_k .

2. В каждом кластере P_i вычисляется новое ядро $\hat{a}^i, i=1, \dots, k$, и делается переход к шагу 1 с найденными ядрами $a^i(n+1) := \hat{a}^i, i=1, \dots, k$.

Схема одной итерации показана на рис. 1.

На каждой итерации алгоритма уменьшается критерий качества D , отсюда следует сходимость алгоритма: после конечного числа шагов разбиение $\{x^1, \dots, x^m\} = P_1 \cup P_2 \cup \dots \cup P_k$ уже не меняется.

Главная вычислительная сложность в методе k -средних – это поиск ядра в кластере на шаге 2 алгоритма 1. Задача нахождения ядра с использованием расстояния Левенштейна, вообще говоря, является NP-полной. Рассмотрим вычисление ядра в отдельно взятом кластере P . Пусть t – количество последовательностей в кластере $P, t \leq m; N$ – максимальная длина последовательностей из $P; \Sigma$ – множество символов, встречающихся в последовательностях кластера, $\Sigma \subseteq Y; |\Sigma|$ – количество этих символов. Тогда если искать ядро полным перебором, то сложность будет иметь порядок $O(tN^2|\Sigma|^N)$. Для ее уменьшения предлагается использовать два эвристических алгоритма.

1.1. *Алгоритм выбора ядра из кластера.* Наиболее простой способ нахождения ядра в кластере основан на его выборе среди последовательностей кластера.

Пусть x^1, \dots, x^t – последовательности кластера P . Тогда ядром кластера будем называть $x^p \in P$, сумма расстояний от которого до всех элементов кластера минимальна, и формула (2) принимает вид

$$x^p = \arg \min_{x \in P} \sum_{i=1}^t d(x, x^i). \quad (4)$$

В программе это реализуется двойным циклом, перебирающим всевозможные пары входных последовательностей. Сложность нахождения ядра таким способом равна $O(N^2 t^2)$, так как вычисление расстояния Левенштей-

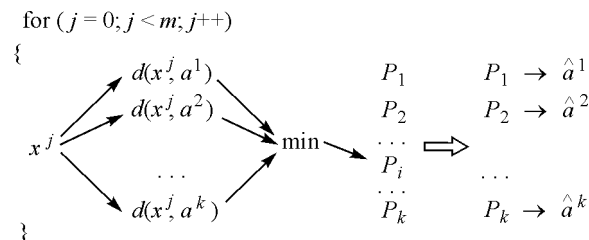


Рис. 1. Схема итерации алгоритма кластеризации методом k -средних

на для двух последовательностей x^i и x^j имеет сложность $O(N_i N_j)$.

1.2. *Алгоритм подсчета символов.* Другой способ нахождения ядра в кластере основан на подсчете частот встречаемости символов в последовательностях кластера и предложен в [11] для «решения модельных задач распознавания символьных последовательностей и реальных задач распознавания слов произвольного диктора». Этот способ предусматривает посимвольное построение ядра. Для его представления введем некоторые понятия.

Массив F – это прямоугольная таблица символов из Y размера $t \times N$, $N = \max N_i$, строками которой являются $x^1, \dots, x^t \in P$. При этом если $N_i < N$, то в i -й строке оставшиеся $N - N_i$ элементов полагаются равными пустому символу e .

Текущий массив $F_{\text{тек}}$ – это таблица символов из Y размера $t \times N$, которая состоит из t строк и обрабатывается в течение работы алгоритма. Перед вычислением алгоритма все элементы текущего массива равны пустому символу.

Алгоритм подсчета символов для нахождения ядра в кластере P выглядит следующим образом.

А л г о р и т м 2.

0. Первый столбец текущего массива $F_{\text{тек}}$ полагается равным первому столбцу массива F .

1. Для каждого $i = 1, \dots, N$ выполняются следующие действия.

1.1. Для каждого непустого символа u в текущем массиве $F_{\text{тек}}$ вычисляется величина ω_y , равная количеству строк $F_{\text{тек}}$, в которых u встречается хотя бы один раз. Из этих символов выбирается u с наибольшей ω_y .

1.2. Если $\max_y(\omega_y) > 0,5t$, то u включается очередным символом в искомое ядро и проводится коррекция текущего массива $F_{\text{тек}}$, которая заключается в следующем. В каждой строке $F_{\text{тек}}$ символ u , а также символы, находящиеся левее самого левого вхождения u , заменяются на e . Если u не входит в какую-нибудь строку, то она остается без изменения. В случае когда условие $\max_y(\omega_y) > 0,5t$ выполняется для нескольких символов u , то для включения в ядро выбирается символ с минимальной суммой номеров позиций в строках текущего массива $F_{\text{тек}}$. Если в строке несколько символов данного типа, то суммируется номер позиции самого левого вхождения символа. Переход на 1.1.

1.3. Если $\max_y(\omega_y) \leq 0,5t$, то очередной столбец текущего массива $F_{\text{тек}}$ полагается равным столбцу массива F с тем же номером. Переход на 1.1.

Сложность алгоритма 2 в худшем случае, когда ни на одном шаге не проводится коррекция текущего массива, оценивается $O(N^2 t)$.

2. Метод k -средних с переходом к частотным словарям. Важнейшим направлением исследований в молекулярной биологии и биофизике является изучение взаимосвязи между структурой генетической цепочки и информацией, зашифрованной в различных ее частях. Одним из подходов к решению такого рода задач является кластеризация. При этом огромные размерности генетических цепочек делают практически неприменимым расстояние Левенштейна. В качестве альтернативы в [8] предлагается осуществлять кластеризацию не последовательностей, а частотных словарей этих последовательностей. В этом случае последовательности считаются близкими, если близки их частотные словари в евклидовой метрике.

Рассмотрим множество $X = \{x^1, \dots, x^m\}$, $x^i = [y_1^i, y_2^i, \dots, y_{N_i}^i]$. Для каждой последовательности x^i частотный словарь толщины q , $N_i \geq q$, строится следующим образом. Нумеруются всевозможные подпоследовательности длины q всех элементов множества X . Если число таких подпоследовательностей равно s , то для x^i частотный словарь $W_i = (f_1, f_2, \dots, f_s)$ – это вектор размерности s , j -я координата которого равна частоте встречаемости j -й подпоследовательности в x^i , т. е.

$$f_j = \frac{\alpha_j}{N_i - q + 1}; \quad 0 \leq f_j \leq 1 \quad (j=1, \dots, s); \quad \sum_{j=1}^s f_j = 1, \quad (5)$$

где α_j – количество повторений подпоследовательности с номером j в x^i , а число всех подпоследовательностей длины q в x^i равно $N_i - q + 1$.

Каждый словарь можно рассматривать как точку в s -мерном евклидовом пространстве и проводить кластеризацию словарей $\{W_1, \dots, W_m\}$ алгоритмом 1 метода k средних с использованием обычной евклидовой метрики. Ядро a^i в кластере P_i с использованием евклидовой метрики вычисляется по формуле

$$a^i = \frac{1}{|P_i|} \sum_{W \in P_i} W. \quad (6)$$

Так как каждой последовательности x^i соответствует свой частотный словарь W_i толщины q , то, кластеризовав множество словарей $\{W_1, \dots, W_m\}$, получим кластеризацию исходного множества последовательностей $\{x^1, \dots, x^m\}$. Следует отметить, что результаты кластеризации существенно зависят от выбора толщины q словаря [8].

Основным преимуществом этого подхода является линейная сложность вычислений по s и m на каждой итерации, что позволяет применять метод для обработки длинных генетических последовательностей.

Так как сделан переход к евклидовой метрике, то кластеризацию можно проводить с помощью нейронной сети Кохонена [6]. В отличие от метода k -средних в этом случае ядро корректируется по мере поступления очередной последовательности из множества X . Поэтому не требуется применять на каждой итерации при вычислении ядра в соответствующем кластере формулу (4).

Далее приведены основные шаги кластеризации с помощью нейронной сети Кохонена.

Алгоритм 3.

0. Фиксируется начальный набор ядер $a^1(0), \dots, a^k(0)$ произвольно либо по какому-нибудь эвристическому правилу, и полагается $P_i = \emptyset$, $i=1, \dots, k$.

1. На итерации с номером n для каждого $j=1, \dots, m$ выполняются следующие действия.

1.1. Для вектора W^j вычисляются расстояния до всех k ядер, и выбирается ближайшее к W^j ядро $a^i(n)$:

$$\|W^j - a^i(n)\| = \min \left\{ \|W^j - a^1(n)\|, \dots, \|W^j - a^k(n)\| \right\}.$$

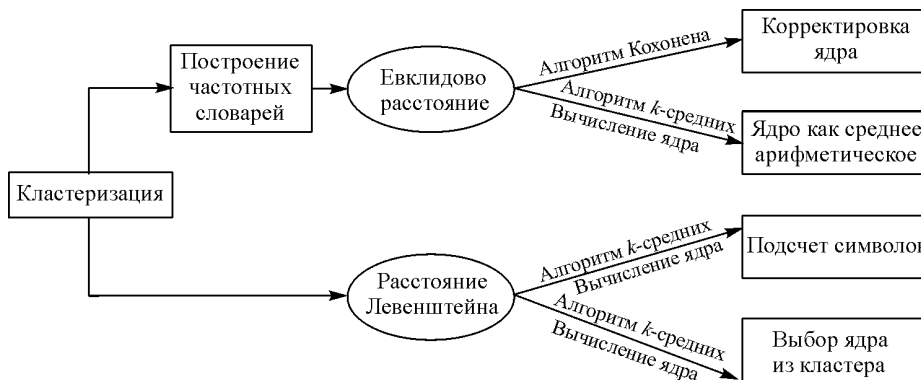


Рис. 2. Схема отношений исследуемых алгоритмов

В случае, когда минимум достигается при нескольких значениях i , выбор между ними может быть сделан произвольно.

1.2. Проводится корректировка ядра:

$$a^i(n+1) = a^i(n) + \eta(n)(W^j - a^i(n)), \quad a^p(n+1) = a^p(n), \quad p=1, \dots, k, \quad p \neq i.$$

Вектор W^j добавляется в кластер P_i .

2. Повторяется первый шаг до тех пор, пока разбиение не перестанет меняться. Здесь последовательность $\eta(n)$ – это шаг обучения, который должен удовлетворять условиям

$$\sum_{n=1}^{\infty} \eta(n) = \infty; \quad \sum_{n=1}^{\infty} (\eta(n))^2 < \infty. \quad (7)$$

Например, можно взять в качестве такой последовательности $\eta(n) = 1/n$, или $\eta(n) = \eta_0 n^{-\beta}$, или $\eta(n) = \eta_0(1 - \beta n)$, где $\beta \leq 1$.

3. Численные эксперименты. Алгоритмы для всех описанных выше методов реализованы в среде Visual C++ 6.0 с использованием библиотеки MFC. На рис. 2 показаны отношения между исследованными алгоритмами. Было проведено несколько сотен численных экспериментов, в которых сравнивались по времени и точности два алгоритма нахождения ядра в кластере (разд. 3.1) для их использования в методе k -средних с расстоянием Левенштейна (разд. 3.2), исследовалась зависимость качества кластеризации с использованием частотных словарей от различных параметров последовательностей, сравнивались результаты кластеризации методом k -средних и нейронной сетью Кохонена в евклидовом пространстве (разд. 3.3).

В экспериментах использовались случайно сгенерированные последовательности символов с заданными значениями длины и мощности алфавита (см. разд. 3.1 и 3.2) и последовательность ДНК бактерии *Methanococcus marisaludis* [12] длиной 1661137 символов (см. разд. 3.3), при этом проводилась кластеризация ее отрезков.

Во всех экспериментах последовательности множества $X = \{x^1, \dots, x^m\}$ имеют одинаковую длину N .

3.1. Экспериментальное исследование двух алгоритмов нахождения ядра в кластере с использованием расстояния Левенштейна. Здесь экспери-

ментально определяются и сравниваются вычислительные характеристики двух алгоритмов нахождения ядра в отдельно взятом кластере P : выбора из кластера и подсчета символов. Пусть кластер P состоит из t последовательностей x^1, \dots, x^t длины N , $t \leq m$; Σ – множество символов, встречающихся в последовательностях кластера, $\Sigma \subseteq Y$. Вычисляется ядро a кластера двумя алгоритмами и сравнивается их работа по точности вычисления ядра и времени.

Для сравнения алгоритмов по точности вычисления ядра рассматривается среднее расстояние Левенштейна \bar{d} от найденного ядра a до всех последовательностей кластера, которое вычисляется по формуле

$$\bar{d} = \frac{1}{t} \sum_{i=1}^t d(a, x^i). \quad (8)$$

Из формулы (2) следует, что точное ядро кластера – это последовательность, реализующая минимум среднего расстояния до последовательностей кластера. Поэтому считается, что один алгоритм точнее другого вычислил ядро, если соответствующее среднее расстояние меньше. Пусть \bar{d}' и \bar{d}'' – это средние расстояния от ядер a' и a'' до последовательностей кластера, найденных с помощью алгоритма выбора из кластера и алгоритма подсчета символов соответственно.

Эксперимент 1. Цель эксперимента – показать зависимости \bar{d}' и \bar{d}'' от количества t последовательностей в кластере. Для каждого значения t (от 2 до 60) при фиксированных длине $N = 20$ и мощности множества символов $|\Sigma| = 50$ производились следующие действия:

- двумя алгоритмами вычислялись соответственно два варианта ядра кластера a'_t и a''_t ;
- для обоих найденных ядер вычислялись средние расстояния \bar{d}'_t и \bar{d}''_t по формуле (8).

Данный эксперимент проводился также для значений длин последовательностей N , равных 191, 382, 573, 764.

На рис. 3 изображены пять пар кривых для пяти запусков эксперимента 1 с различными значениями N . Из рисунка видно, что при малых t алгоритм выбора из кластера точнее вычисляет ядро по сравнению с алгоритмом подсчета символов. В точке t_0 точность обоих алгоритмов совпадает, а при $t > t_0$ точность алгоритма подсчета символов становится несколько лучше точности алгоритма выбора из кластера.

Эксперимент 2. Цель эксперимента – получить данные о времени вычисления ядра алгоритмом выбора из кластера и алгоритмом подсчета символов. С помощью функции `clock()` измерялось время вычисления в миллисекундах ядра a в кластере обоими алгоритмами при фиксированном N и различных t .

Результаты измерения времени приведены в таблице, которая подтверждает аналитическую оценку, данную в разд. 1.

Из вышеизложенного следует, что алгоритм подсчета символов несколько превосходит по качеству и скорости вычисления ядра алгоритм выбора из кластера при достаточно большом количестве t последовательностей в кластере.

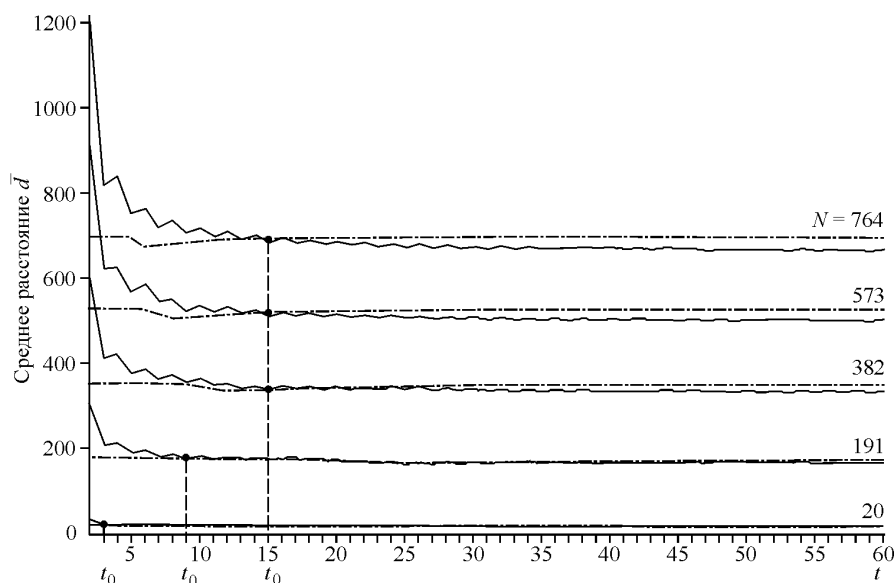


Рис. 3. Зависимости среднего расстояния \bar{d} от числа последовательностей t в кластере P при различных N и фиксированном $|\Sigma|$. На графиках отмечены точки пересечения t_0 для каждой пары кривых (штрихпунктирная кривая – алгоритм выбора из кластера, сплошная – алгоритм подсчета символов)

3.2. *Экспериментальное исследование кластеризации методом k -средних с применением двух алгоритмов нахождения ядра в кластере.* Экспериментально определяются и сравниваются вычислительные характеристики двух алгоритмов кластеризации, которые различаются способом нахождения ядер в кластерах. Проводится кластеризация множества символьных последовательностей $X = X_m = \{x^1, \dots, x^m\}$ методом k -средних с использованием двух алгоритмов нахождения ядра в кластере: подсчета символов и выбора из кластера, а также сравниваются полученные результаты по качеству кластеризации и времени вычисления.

Пусть D' и D'' – критерии качества, вычисленные по формуле (3), для кластеризации X методом k -средних с использованием алгоритма подсчета символов и алгоритма выбора из кластера соответственно. Считается, что одна кластеризация качественнее другой, если ее критерий качества меньше.

Эксперимент 3. Цель эксперимента – показать зависимости относительных критериев качества D'/m и D''/m от количества m последовательностей в X_m . Для каждого значения m от 30 до 290 при фиксированных длине $N = 20$, мощности множества символов $|\Sigma| = 50$ и количестве кластеров $k = 20$ производились следующие действия:

$N = 40$	Алгоритм выбора ядра из кластера (время, мс)	Алгоритм подсчета символов (время, мс)
$t = 40$	1203	141
$t = 200$	29609	703
$t = 700$	361750	2609

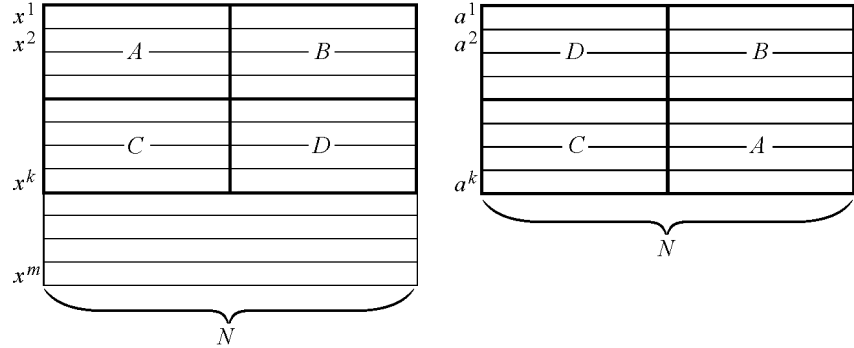


Рис. 4. Выбор начальных ядер на основе входных последовательностей

– выполнялась кластеризация множества X_m двумя способами: методом k -средних (алгоритм 1) с алгоритмом выбора из кластера и методом k -средних с алгоритмом подсчета символов;

– для обоих вариантов кластеризации вычислялись критерии качества D' и D'' по формуле (3).

В данном эксперименте начальные ядра выбирались на основе входных последовательностей следующим образом:

$$a^i = \underbrace{[y_{\lfloor N/2 \rfloor + 1}^{i+\lfloor k/2 \rfloor}, \dots, y_N^{i+\lfloor k/2 \rfloor}]}_D, \underbrace{[y_{\lfloor N/2 \rfloor + 1}^i, \dots, y_N^i]}_B, \text{ если } i=1, \dots, \lfloor k/2 \rfloor,$$

$$a^i = \underbrace{[y_1^i, \dots, y_{\lfloor N/2 \rfloor}^i]}_C, \underbrace{[y_1^{i-\lfloor k/2 \rfloor}, \dots, y_{\lfloor N/2 \rfloor}^{i-\lfloor k/2 \rfloor}]}_A, \text{ если } i=\lfloor k/2 \rfloor + 1, \dots, k.$$

Такой выбор ядер схематически иллюстрирует рис. 4. Эксперимент 3 проводился также для числа кластеров $k=30$.

На рис. 5, a, b изображены два графика зависимостей относительных критериев качества D'/m и D''/m для двух значений k : $k=20$ (a), $k=30$ (b). Из графиков видно, что при малых значениях m выполняется соотношение $D' > D''$, а при увеличении m разница сглаживается: $D' \approx D''$. Это объясняется тем, что если количество последовательностей в X много больше числа кластеров, то с большой вероятностью в каждый кластер попадает достаточное количество последовательностей, чтобы алгоритм подсчета символов давал приемлемое качество вычисления ядер в кластерах (см. разд. 3.1).

Следует заметить, что в процессе кластеризации входные последовательности могут распределяться по кластерам неравномерно. Это ведет к тому, что в кластерах с малым числом последовательностей вычисление ядра алгоритмом подсчета символов дает плохое качество, и использование таких ядер на следующей итерации не ведет к минимизации D' . Вследствие этого для кластеризации с алгоритмом подсчета символов в рассмотренном эксперименте требовалось в среднем в 10 раз больше итераций, чем с алгоритмом выбора из кластера. Поэтому преимущества алгоритма подсчета символов по времени вычисления ядра (см. разд. 3.1) теряют свою актуальность.

Итак, метод k -средних с алгоритмом выбора ядра из кластера эффективнее метода k -средних с алгоритмом подсчета символов по времени из-за су-

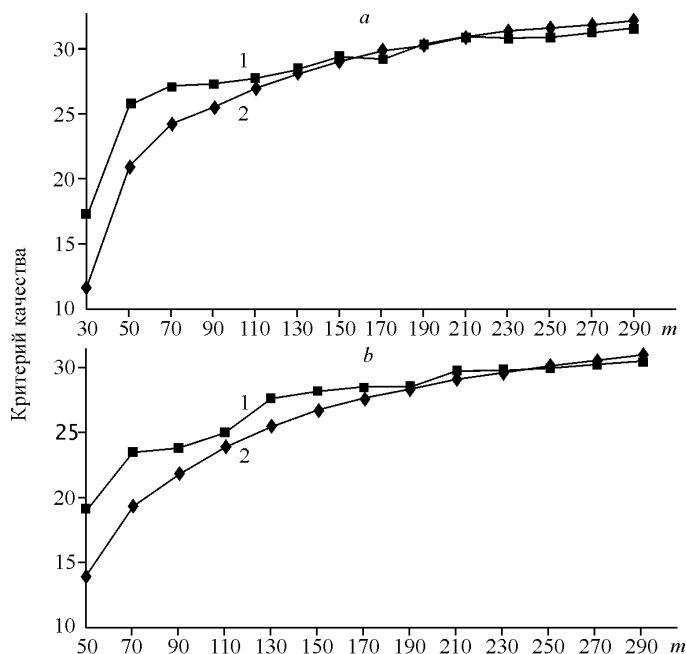


Рис. 5. Зависимости относительных критериев качества D'/m (кривая 1) и D''/m (кривая 2) кластеризаций с использованием алгоритмов подсчета символов и выбора из кластера соответственно от количества кластеризуемых последовательностей при фиксированных N и $|\Sigma|$

щественной разницы в числе итераций и в среднем сравним с ним по качеству, если количество последовательностей в X много больше числа кластеров.

3.3. *Экспериментальное исследование кластеризации методом k -средних частотных словарей символьных последовательностей в евклидовом пространстве.* Экспериментально определяются и сравниваются вычислительные характеристики двух алгоритмов кластеризации множества частотных словарей $\{W_1, \dots, W_m\}$ [8] в евклидовом пространстве: нейронная сеть Кохонена и метод k -средних.

Для оценки качества кластеризации используются критерии качества D' и D'' для нейронной сети Кохонена и метода k -средних соответственно, вычисляющиеся по формуле, аналогичной (3), но с использованием евклидова расстояния:

$$D = \sum_{i=1}^k \sum_{W \in P_i} d_E(a^i, W), \quad (3')$$

где $P_1 \cup P_2 \cup \dots \cup P_k = \{W_1, \dots, W_m\}$; d_E – евклидово расстояние.

Эксперимент 4. Цель эксперимента – показать зависимости критериев качества D' и D'' от длины N последовательностей. Для каждого значения N от 50 до 1050 при фиксированном количестве последовательностей $m=300$ в X , количестве кластеров $k=30$, мощности множества символов $|\Sigma|=4$, толщине словарей $q=5$, шаге обучения для сети Кохонена $\eta(n)=1/n$ производились следующие действия:

- по входным последовательностям $\{x^1, \dots, x^m\}$ строилось множество частотных словарей $\{W_1, \dots, W_m\}$, как описано в разд. 2;

– выполнялась кластеризация множества $\{W_1, \dots, W_m\}$ двумя способами: методом k -средних с евклидовым расстоянием и с помощью нейронной сети Кохонена;

– для обоих вариантов кластеризации вычислялись соответственно критерии качества D' и D'' по формуле (3').

В данном эксперименте в качестве начальных ядер брались первые k векторов множества частотных словарей: $a^1 := W_1, \dots, a^k := W_k, k \leq m$.

На рис. 6 изображены графики зависимостей критериев качества D' и D'' от длины N входных последовательностей. Из графиков видно, что при увеличении длины последовательностей оба критерия качества убывают. Поэтому приемлемое качество кластеризации достигается для достаточно длинных символьных последовательностей [9]. Заметим, что при фиксированном $m=300$ критерии качества обоих алгоритмов мало отличаются. В следующем эксперименте показано, что при увеличении числа m последовательностей эта разница увеличивается.

Эксперимент 5. Цель эксперимента – показать зависимости критериев качества D' и D'' , а также времени вычислений при кластеризации $\{W_1, \dots, W_m\}$ двумя алгоритмами от количества m последовательностей в X . Для каждого значения m от 50 до 1000 при фиксированных длине $N=1000$, количестве кластеров $k=30$, мощности множества символов $|\Sigma|=4$, толщине словарей $q=5$, шаге обучения для сети Кохонена $\eta(n)=1/n$ производились следующие действия:

– по входным последовательностям $\{x^1, \dots, x^m\}$ строилось множество частотных словарей $\{W_1, \dots, W_m\}$, как описано в разд. 2;

– выполнялась кластеризация множества $\{W_1, \dots, W_m\}$ двумя способами: методом k -средних с евклидовым расстоянием и с помощью нейронной сети Кохонена;

– фиксировалось время выполнения кластеризации обоими алгоритмами;

– для обоих вариантов кластеризации вычислялись критерии качества D' и D'' по формуле (3').

В данном эксперименте в качестве начальных ядер брались первые k векторов множества частотных словарей: $a^1 := W_1, \dots, a^k := W_k, k \leq m$.

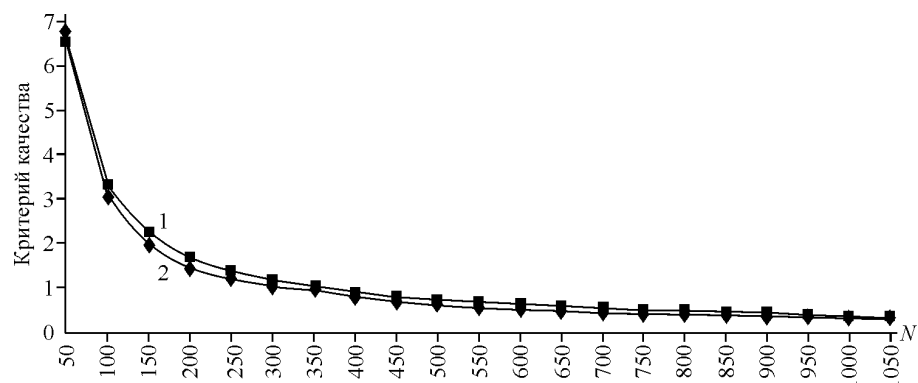


Рис. 6. Зависимости критериев качества D' (кривая 1) и D'' (кривая 2) кластеризации множества $\{W_1, \dots, W_m\}$ с помощью нейронной сети Кохонена и методом k -средних соответственно от длины N последовательностей

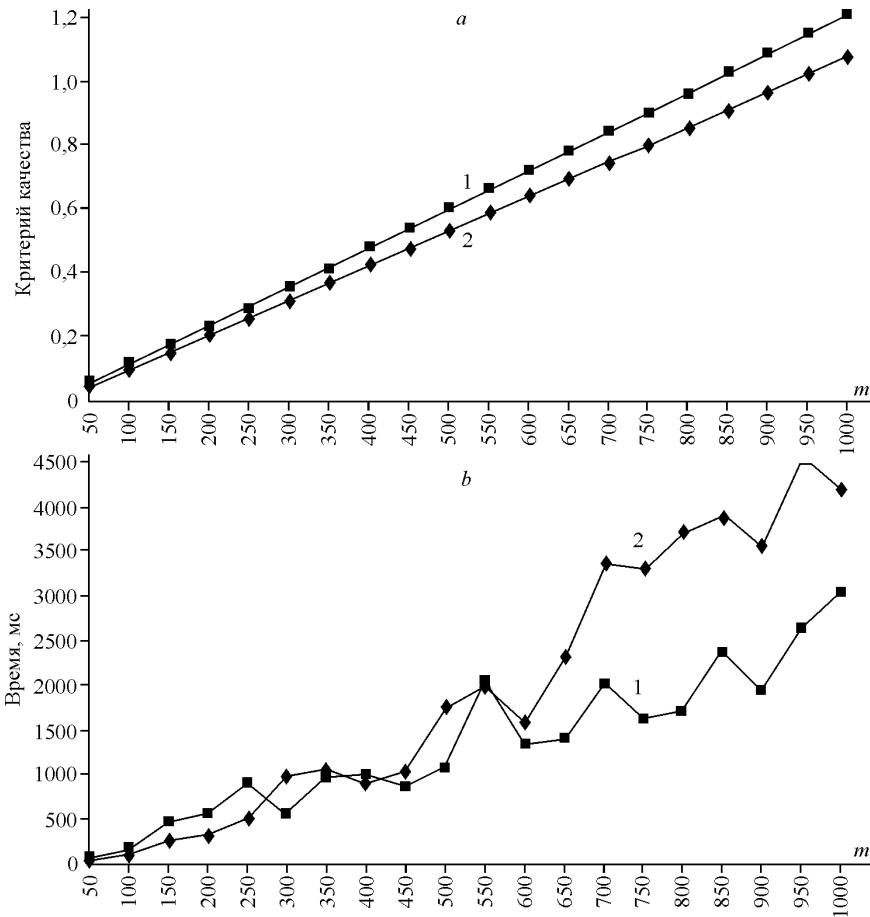


Рис. 7. Зависимости критериев качества D' (кривая 1) и D'' (кривая 2) и времени вычислений при кластеризации множества $\{W_1, \dots, W_m\}$ с помощью нейронной сети Кохонена и методом k -средних соответственно от количества m последовательностей

На рис. 7, *a* показаны зависимости критериев качества D' и D'' от m , а на рис. 7, *b* – зависимости времени вычислений при кластеризации $\{W_1, \dots, W_m\}$ двумя алгоритмами от m . Из графиков видно, что при данном шаге обучения качество кластеризации с помощью нейронной сети Кохонена немного хуже, однако сходимость этого алгоритма происходит быстрее сходимости метода k -средних. При подборе шага обучения можно добиться улучшения качества кластеризации сетью Кохонена с небольшой потерей скорости сходимости.

Заключение. В представленной работе приводится описание различных модификаций метода k -средних с расстоянием Левенштейна, которые в той или иной степени позволяют ускорить вычисления при кластеризации множества символьных последовательностей. Исследованы временные и качественные характеристики этих алгоритмов и сделаны следующие выводы.

1. В методе k -средних с расстоянием Левенштейна для вычисления ядра в кластерах лучше использовать алгоритм выбора ядра из кластера, так как он имеет преимущества по времени вычислений и в среднем сравним по качеству с алгоритмом подсчета символов.

2. Метод перехода к частотным словарям при кластеризации символьных последовательностей применим только для достаточно длинных последовательностей, какими являются, например, генетические последовательности. При этом, используя нейронную сеть Кохонена с подходящим шагом обучения, можно ускорить кластеризацию с некоторой потерей качества.

3. При использовании кластеризации для выявления структуры мелодии в музыкальном произведении следует применять метод k -средних с алгоритмом выбора из кластера для нахождения ядер. Переход к частотным словарям не дает качественных результатов из-за большой мощности алфавита и малых длин последовательностей, характеризующих нотный текст.

4. Использование метода перехода к частотным словарям при кластеризации генетических последовательностей позволяет избежать многократно и достаточно трудоемкого вычисления расстояния Левенштейна, что является существенным для длинных последовательностей.

СПИСОК ЛИТЕРАТУРЫ

1. **Льюин Б.** Гены. М: Мир, 1987.
2. **Sander J., Ester M., Kriegel H.-P., Xu X.** Density-based clustering in spatial databases: the algorithm GDBSCAN and its applications // Data Mining and Knowledge Discovery. 1998. **2**, N 2. P. 169.
3. **Guha S., Rastogi R., Shim K.** ROCK: A robust clustering algorithm for categorical attributes // Information System. 2000. **25**, N 5. P. 345.
4. **Yang Y., Guan X., You J.** CLOPE: A fast and effective clustering algorithm for transactional data // Proc. SIGKDD'02. Edmonton, Alberta, Canada, 2002. P. 682.
5. **McQueen J.** Some methods for classification and analysis of multivariate observations // Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Berkeley: University of California Press, 1967. Vol. 1. P. 281.
6. **Kohonen T. K.** Self-Organization and Associative Memory. N. Y.: Springer Verlag, 1989.
7. **Levenshtein V. I.** Binary codes capable of correcting deletions, insertions, and reversals // Cybernetics and Control Theory. 1966. **10**, N 8. P. 707.
8. **Gorban A. N., Popova T. G., Sadovsky M. G.** Classification of symbol sequences over their frequency dictionaries: towards the connection between structure and natural taxonomy // Open Syst. & Inform. Dyn. 2000. N 7. P. 1.
9. **Sadovsky M. G.** The method to compare nucleotide sequences based on the minimum entropy principle // Bull. Math. Biol. 2003. N 65. P. 309.
10. **Wagner R. A., Fischer M. J.** The string-to-string correction problem // Journ. of the ACM. 1974. **21**, N 1. P. 168.
11. **Кузнецов П. Г.** Обучение при распознавании символьных последовательностей // Вычислительные системы. 1994. № 150. С. 164.
12. <http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi?db=nucleotide&val=45357563>

*Институт вычислительной математики
и математической геофизики СО РАН,
E-mail: nechaeva@ssd.sccc.ru*

*Поступила в редакцию
19 января 2004 г.*