УДК 004.424.4, 004.93.14, 004.021, 004.023, 004.912

# АЛГОРИТМ ВЫБОРА ХАРАКТЕРИСТИЧЕСКОГО ЭЛЕМЕНТА МНОЖЕСТВА В ЗАДАЧЕ КЛАСТЕРИЗАЦИИ

**Трофимов Б. И., Кольцов Ю. В., Гарнага В. В.**

ALGORITHM ABOUT SELECTION OF THE CHARACTERISTIC ELEMENT IN A CLUSTERING PROCESS'S SET

Trofimov B. I., Koltsov U. V., Garnaga V. V.

Kuban State University, Krasnodar, 350040, Russia

*Abstract.* People use classification for objects organization into groups since ancient times. In one of his articles Robert Sokal notes that classification is high level of intellectual activity and it helps to understand the nature. Clustering is result of software algorithms applying to classification. This approach allows deploying data mining to classified information.

The article describes an algorithm for a cluster characteristic element selection and its formal requirements definition. One of areas for the algorithm's applying is intellectual text search systems.

A main purpose of the article is description of an algorithm for characteristic element selection. The algorithm should have less asymptotic estimate operating time than enumeration of all elements.

A main idea based on the classical method of branches and borders. An original part of the algorithm is errors estimates comparison for selected characteristic element. Also, the article describes two algorithms for random test data generation. Showed results of these tests illustrate and explain advantages of the main algorithm in comparison with the enumeration algorithm. An empirical assessment of the proposed algorithm convergence demonstrates its better efficiency. We plan to use the article results in intellectual text search area. Clustering and neural networks are main approaches used in this area.

*Keywords:* branch and bound method, text search, graph models, Damerau–Lowenstein metric

## Введение

Человек использует классификацию для упорядочивания объектов в группы с древнейших времен. В одной из своих статей Р. Сокэл отмечает, что классификация является интеллектуальной деятельностью высокого уровня, необходимой нам для понимания природы [1]. Кластеризация как результат ее развития с применением средств автоматизации является одним из важных подходов, позволяющих произвести интеллектуальный анализ данных.

В качестве основных задача кластеризации выделяют:

– разработку типологии или классификации;

– исследование полезных концептуальных схем группирования объектов;

– порождение гипотез на основе исследования данных;

– проверку гипотез или исследования для определения присутствия, выделенных тем или иным способом, типов (групп) в имеющихся данных.

Большинство методов кластеризации используют три вида расстояния: между двумя объектами, между объектом и кластером и между двумя кластерами. Для вычисления расстояния между двумя объектами чаще всего используются меры, вычисление которых обычно не представляет алгоритмической сложности, будь то евклидова метрика или расстояние Левенштейна [2]. Основную

Трофимов Богдан Игоревич, аспирант кафедры информационных технологий Кубанского государственного университета; e-mail: bogdan.i.trofimov@mail.ru

Кольцов Юрий Владимирович, канд. физ.-мат. наук, заведующий кафедрой информационных технологий Кубанского государственного университета; e-mail: dean@fpm.kubsu.ru

Гарнага Валерий Владимирович, канд. физ.-мат. наук, доцент кафедры информационных технологий Кубанского государственного университета; e-mail: Garnaga.Valeriy@fpm.kubsu.ru

Работа выполнена при поддержке РФФИ (13-01-00807).

проблему, как правило, представляет вычисление расстояний между кластером и объектом и между двумя кластерами, поскольку при самом простом подходе на основе сравнения всех объектов, содержащихся в кластерах, асимптотическая оценка такого подхода равна $O(N \cdot M)$, где $N$, $M$ — количество объектов в первом и втором кластере соответственно. Такой вариант алгоритмически сложен и неприменим при работе с большим количеством объектов. В данных условиях можно выбрать один объект из кластера в качестве характеристического, такой объект должен быть в наибольшей степени похож на остальные объекты из кластера. В настоящей статье предлагается алгоритм выбора указанного характеристического объекта, при этом приводятся формальные условия определения этого объекта. Алгоритм основан на идее, схожей по смыслу с классическим методом ветвей и границ [3]. Предлагаемый алгоритм может использоваться, например, в задаче текстового поиска [5].

## 1. Описание алгоритма

Обозначим кластеризуемую выборку объектов через $X$. Алгоритм построен, исходя из предположения,что используемая метрика оценки расстояния между объектами кластеризуемой выборки удовлетворяет трем основный аксиомам расстояния:

$$\rho(a,b) = 0 \;\Leftrightarrow a = b,$$

$$\rho(a,b) \geqslant 0, \;\forall\; a, b,$$

$$\rho(a,b) \leqslant \rho(a,c) + \rho(b,c).$$

Как известно, третья аксиома носит также название аксиомы треугольника.

Кроме того, расстояние должно удовлетворять условию коммутативности

$$\rho(a,b) = \rho(b,a).$$

Введем функцию $e(a)$

$$e(a) = \sum_{\forall\, b \in X} \rho(a,b), \tag{1.1}$$

где $a, b \in X$, назовем ее ошибкой выбора центра кластера.

Характеристический, «средний», «оптимальный» элемент $\bar{a}$ может быть определен по следующему условию:

$$e(\bar{a}) = \min_{\forall\, a \in X} e(a).$$

Это условие означает, что выбирается объект с минимальной ошибкой из всех возможных.

Остановимся на некоторых свойствах функции $e(a)$, необходимых для дальнейшего изложения. Пусть известно значение функции $e(a)$ для некоторого объекта $a$. Рассмотрим выражение для ошибки некоторого объекта $b$

$$e(b) = \rho(a,b) + \sum_{\forall\, c \in X \setminus \{a\}} \rho(b,c).$$

Воспользовавшись аксиомой треугольника, можно получить следующее выражение:

$$\rho(b,c) + \rho(a,c) \geqslant \rho(a,b).$$

Откуда следует, что

$$\rho(b,c) \geqslant \rho(a,b) - \rho(a,c).$$

Подставляя полученное выражение в выражение ошибки объекта $b$, получаем

$$e(b) \geqslant \rho(a,b) + \sum_{\forall c \in X \setminus \{a\}} (\rho(a,b) - \rho(a,c)). \tag{1.2}$$

Выносим в (1.2) из-под знака суммы $\rho(a,b)$

$$e(b) \geqslant \rho(a,b)\,|X| - \sum_{\forall c \in X \setminus \{a\}} \rho(a,c). \tag{1.3}$$

Очевидно, что выражение справа от знака разности представляет собой ошибку объекта $a$. Тогда соотношение (1.3) может быть представлено как

$$e(b) \geqslant \rho(a,b)\,|X| - e(a). \tag{1.4}$$

Значение справа от знака неравенства в (1.4) для удобства дальнейшего изложения обозначим как

$$\mu(b) = \rho(a,b)\,|X| - e(a). \tag{1.5}$$

Таким образом, получили нижнюю оценку $\mu(b)$ ошибки произвольного объекта $b$, исходя из известной заранее ошибки объекта $a$. Эта формула является ключевой для данного алгоритма. Исходя из нее можно оценить ошибки всех объектов в кластере, вычислив точное значение лишь одной, и сразу отсеять все ошибки, нижняя оценка которых превышает уже вычисленную, поскольку их ошибка, очевидно будет больше последней.

Введем следующие обозначения: $Y$ — множество объектов, ошибки которых уже вычислены; $Z$ — множество, содержащее объекты, точные значения ошибок которых еще не вычислены, но нижние оценки ошибок не превышают ни одной из ошибок для объектов из множества $Y$. Кроме того, на каждой итерации алгоритма каждому элементу $a \in Z$ ставится в соответствие число $\varphi_i(a)$, смысл и способ вычисления которого будет пояснен далее, где $i$ номер очередной итерации поиска оптимального элемента.

Используя введенные обозначения, опишем предлагаемый алгоритм выбора характеристического элемента:

1°. Множество $Y$полагается равным $\varnothing$, множество $Z$ совпадает с множеством $X$, значения $\varphi_0(a)$ не определены.

2°. Из множества $Z$ выбирается произвольный элемент $a$, вычисляется его ошибка $e(a)$, а также нижние оценки ошибок $\mu(b)$, для всех элементов $b \in Z \setminus \{a\}$, значения $\varphi_0(b)$ полагаются равными $\mu(b)$. Элемент $a$ переносится в множество $Y$; из множества $Z$ удаляются все элементы $p$, такие, что

$$\varphi_0(p) > e(a).$$

Далее выполняется конечное число итераций формирования множества $Y$. Номер очередной итерации будем обозначать через $i$. Принимаем $i$ равным единице.

3°. На шаге $i$ из множества $Z$ выбирается элемент $a$ с наименьшим значением функции $\varphi(a)$ и вычисляется точное значение его ошибки $e(a)$. Элемент $a$ переносится в множество $Y$. Вычисляются значения $\varphi(b)$ для всех $b \in Z \setminus \{a\}$ по формуле

$$\varphi_i(b) = \max\bigl(\varphi_{i-1}(b), \mu(b)\bigr),$$

где $\mu(b)$ определяется из соотношения(1.5)

$$\mu(b) = \rho(a,b)\,|X| - e(a);$$

Из множества $Z$ удаляются все элементы $p$, для которых

$$\exists\, c \in Y : \varphi_0(p) \geqslant e(c).$$

4°. Если $Z \neq \varnothing$, то $i = i + 1$ и алгоритм возвращается к шагу 3°.

5°. Если $Z = \varnothing$, то из множества $Y$ выбирается элемент с наименьшей оценкой, это и есть характеристический элемент кластера.

## 2. Тестирование работы алгоритма

На основе предлагаемого алгоритма была написана компьютерная программа и проведено сравнение результатов ее работы с реализацией, находящей характеристический элемент полным перебором.

Для генерации исходных данных (множества $X$) предлагается два подхода:

1) на основе генерации связей элементов в полносвязном графе с симметричной матрицей весов;
2) на основе генерации случайных слов.

Остановимся подробнее на описании предлагаемых алгоритмов генерации исходных данных.

### *2.1. Подход на основе генерации весов связей в полносвязном графе*

В предлагаемом подходе расстояния между элементами множества $X$ соответствуют весам взвешенного полносвязного графа, вершинам которого ставятся в соответствие элементы множества $X$. Генерация весов связей происходит итеративно, с учетом ограничений, накладываемых аксиомой треугольника. Далее опишем процесс генерации весов связей указанного графа.

Пронумеруем элементы множества $X$ от 1 до $n$, где $n = |X|$. Рассмотрим матрицу весов $M$, соответствующую генерируемому графу, где элементы первого столбца и первой строки соответствуют весам связей первого элемента множества $X$с самим собой ($M[1,1]$), вторым ($M[1,2]$ и $M[2,1]$ соответственно), третьим элементом ($M[1,3]$ и $M[3,1]$ соответственно) и т. д. Матрица весов симметрична, значит, $M[i,j] = M[j,i]$, поэтому рассматриваются только верхний треугольник матрицы, выше и правее главной диагонали, последняя не включается в рассмотрение.

Допустим, что процесс генерации будет проходить столбец за столбцом слева направо, и сверху вниз в каждом столбце. Сначала элементы матрицы весов с индексами $M[1,j]$, $j = 1,\ldots,n$, генерируются случайным образом в диапазоне, заданном до начала эксперимента.

На все последующие элементы столбцов $(2,\ldots,j-1)$ со 2-го по $n$-й накладываются

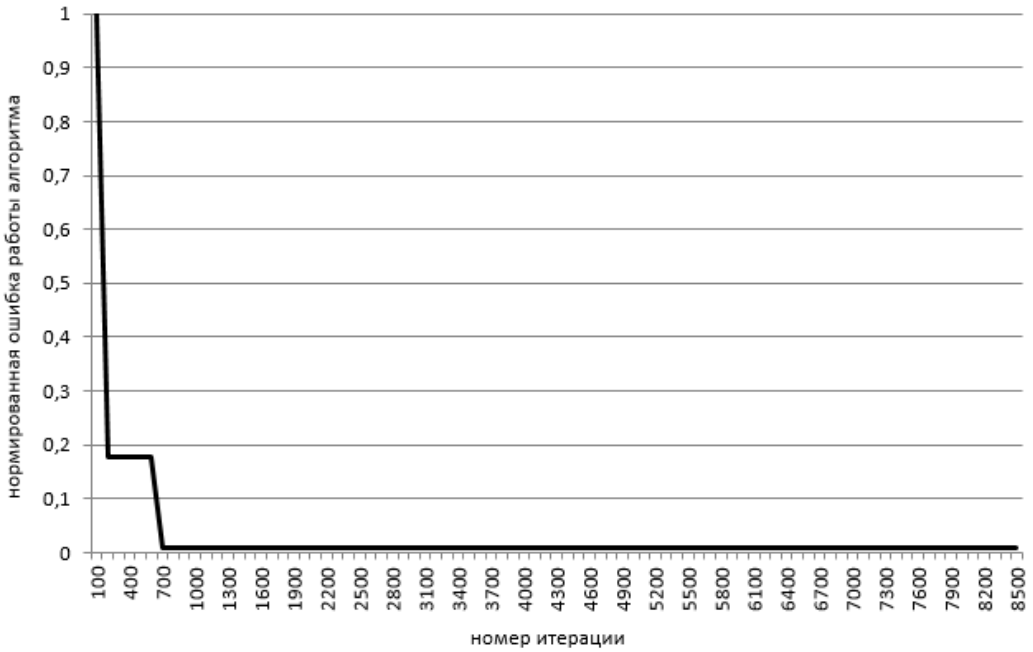


Рис. 1. Зависимость результата работы предлагаемого алгоритма от номера итерации

ограничения аксиомой треугольника

$$\max_{k=0,\ldots,i-1}\Big(\max\left(M\left[k,i\right],M\left[k,j\right]\right)- \\ -\min\left(M\left[k,i\right],M\left[k,j\right]\right)\Big)\leqslant M\left[i,j\right], \quad (2.1)$$

$$M\left[i,j\right]\leqslant\min_{k=0,\ldots,i-1}\left(M\left[k,i\right]+M\left[k,j\right]\right).$$

Генерация проводится в диапазоне, заданном этими ограничениями.

### *2.2. Подход на основе генерации случайных слов*

В этом подходе элементами множества $X$ являются случайно сгенерированные слова различной длины. В данной работе использовались слова длины от 3 до 30 символов. В качестве расстояния между сгенерированными словами используется метрика Дамерау–Левенштейна [4], удовлетворяющая указанным выше требованиям к метрике.

## 3. Экспериментальные исследования

На основе предложенных подходов была проведена практическая апробация предлагаемого алгоритма. Статистические исследования сравнения авторского алгоритма и метода полного перебора показали, что предлагаемый алгоритм выигрывает по времени лишь в 3 % случаев при $|X| = 100$, 2 % случаев при $|X| = 1\,000$. Однако дальнейшее исследование показало что данная проблема обусловлена отсутствием критерия сходимости алгоритма. На рис. 1 представлена зависимость результата работы описанного алгоритма от номера итерации. По оси абсцисс отложен номер итерации, по оси ординат — приведенный результат работы алгоритма. Как видно из рис. 1, после некоторого количества итераций алгоритм сходится к верному результату, при этом на протяжении многих (более 90 %) итераций продолжается отсеивание оставшихся вариантов.

Алгоритм сходится примерно на 700-й итерации. Результат работы предлагаемого алгоритма оценивался по нормированной формуле ошибки кластеризации, вычисляемой с помощью соотношения (1.1). Нормировка ошибки производилась по формуле

$$\bar{e}\left(a\right)=\frac{e\left(a\right)-\min\limits_{\forall b\in X}e\left(b\right)}{\max\limits_{\forall b\in X}e\left(b\right)-\min\limits_{\forall b\in X}e\left(b\right)}.$$

При такой динамике процесса предлагаемый алгоритм выигрывает у алгоритма полного перебора по времени более чем в 3 раза, что иллюстрирует рис. 2. Как было показано выше, предлагаемый алгоритм достиг правильного решения на 700-й итерации.

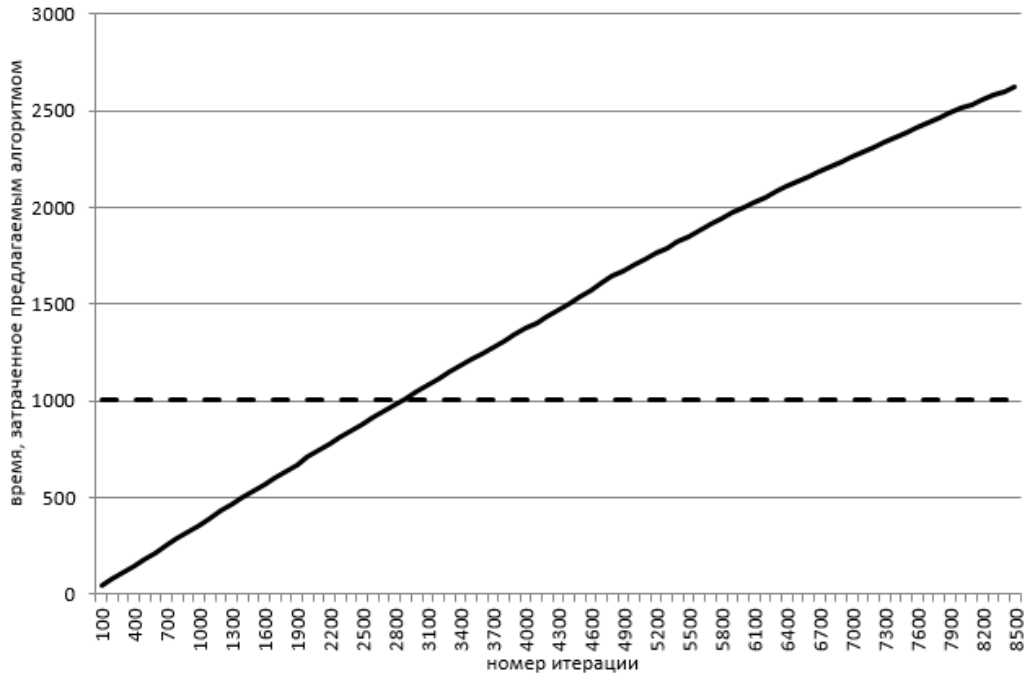


Рис. 2. Время, затраченное предлагаемым алгоритмом в зависимости от номера итерации. По оси абсцисс — номер итерации, по оси ординат — затраченное время, в мс. Пунктирной линией обозначено время, затраченное алгоритмом перебора, сплошной — динамика времени, затрачиваемого предлагаемым алгоритмом

Аналогичные результаты наблюдались в большинстве других экспериментов с различными значениями $|X|$: 100, 1 000, 5 000, 10 000 и 20 000. Количество экспериментов для этих данных, соответственно: 10 000, 1000, 200, 100 и 50. При количестве итераций менее 10 % от значения $|X|$ предлагаемый алгоритм сходился к верному ответу. Поэтому предлагается использовать эмпирическую оценку максимального количества итераций в 10%. При таком выборе предлагаемый алгоритм выигрывает по времени работы в 82% случаев при $|X| = 100$, в 98 % случаев при $|X| = 1\,000$, в 95 % случаев при $|X| = 5\,000$. На бóльших выборках статистические исследования не проводились ввиду весьма продолжительной по времени генерации исходного графа связей между элементами множества $X$.

## Заключение

Итак, в данной работе был предложен алгоритм выбора характеристического элемента множества в задаче кластеризации, разработан эмпирический критерий сходимости предлагаемого алгоритма. Также были предложены два алгоритма генерации исходных данных для проведения экспериментальных исследований по сравнению результатов работы и быстродействия предлагаемого алгоритма, и на их основе проведено сравнение с алгоритмом полного перебора. Планируется дальнейшее использование результатов данного исследования в задаче текстового поиска на основе кластеризации и применения нейросетевых технологий [5].

## Литература

1. *Сокэл Р.Р.* Кластер-анализ и классификация: предпосылки и основные направления. В кн.: Классификация и кластер / Под ред. Дж. Вэн Райзина. М.: Мир, 1980. С. 7–19.
2. *Левенштейн В. И.* Двоичные коды с исправлением выпадений, вставок и замещений символов // ДАН СССР. 1965. Т. 163. № 4. С. 845–848.
3. *Land A.H., Doig A.G.* An automatic method of solving discrete programming problems // Econometrica. 1960. С. 497–520. doi: 10.1.1.308.7332.
4. *Bard G.V.* Spelling-error tolerant, order-independent pass-phrases via the Damerau–Levenshtein string-edit distance metric // Proc. of ACSW '07 Proceedings of the Fifth Australasian symposium on ACSW frontiers. 2007. Vol. 68. P. 117–124.

5. *Гарнага В.В., Кольцов Ю.В., Трофимов Б.И.* Построение механизма нейросетевого поиска на основе алгоритма расширяющегося нейронного газа // Известия вузов. Северо-Кавказский регион. Технические науки. 2014. Вып. 6. С. 12–17. doi: 10.17213/0321-2653-2014-6-12-17

## References

1. Sokel R.R. Klaster-analiz i klassifikatsiya: predposylki i osnovnye napravleniya [Cluster analysis and classification: the background and the main directions]. In Dzh. Ven Rayzina (Ed.). *Klassifikatsiya i klaster* [Classification and cluster]. Moscow, Mir Publ., 1980, pp. 7–19.
2. Levenshteyn V. I. Dvoichnye kody s ispravleniem vypadeniy, vstavok i zameshcheniy simvolov [Binary codes with correction for deletions, insertions and substitutions of characters]. *Doklady Akademii nauk SSSR* [Rep. of the Academy of Sciences of the USSR], 1965, vol. 163, no. 4, pp. 845–848.
3. Land A.H., Doig A.G. An automatic method of solving discrete programming problems. *Econometrica*, 1960, pp. 497–520. doi: 10.1.1.308.7332.
4. Bard G.V. Spelling-error tolerant, order-independent pass-phrases via the Damerau–Levenshtein string-edit distance metric. *Proc. of ACSW '07 Proceedings of the Fifth Australasian symposium on ACSW frontiers*, 2007, vol. 68, pp. 117–124.
5. Garnaga V.V., Kol'tsov Yu.V., Trofimov B.I. Postroenie mekhanizma neyrosetevogo poiska na osnove algoritma rasshiryayushchegosya neyronnogo gaza [The construction of the mechanism of neural network based search algorithm expanding neural gas]. *Izvestiya vuzov. Severo-Kavkazskiy region. Tekhnicheskie nauki* [Proc. of the universities. North Caucasus region. Technical science], 2014, iss. 6, pp. 12–17. doi: 10.17213/0321-2653-2014-6-12-17

Статья поступила 19 ноября 2015 г.