УДК 004.9 DOI 10.18522/2311-3103-2021-6-73-88

Ф.С. Булыга, В.М. Курейчик

АЛГОРИТМЫ АГЛОМЕРАТИВНОЙ КЛАСТЕРИЗАЦИИ ПРИМЕНИТЕЛЬНО К ЗАДАЧАМ АНАЛИЗА ЛИНГВИСТИЧЕСКОЙ ЭКСПЕРТНОЙ ИНФОРМАЦИИ

Рассмотрены и представлены основные проблемы и принципы функционирования процесса кластеризации данных, в частности принципы и задачи кластеризации текстовых массивов лингвистической экспертной информации. В ходе выполнения данной работы были обозначены основные трудности возникающие при проектировании подобного рода систем, например: необходимость предварительной обработки данных, сокращение размерности исходной выборки и т.п. Для эффективного выполнения представленных задач реализованное решение должно обладать комплексным подходом учитывающим показатели эффективности методов направленных на решение отдельных подзадач, а также способностью обеспечить высокие показатели эффективности реализации каждого этапа процесса кластеризации. В представленной работе рассматриваются различные группы алгоритмов иерархической кластеризации, в частности была рассмотрена подгруппа алгоритмов агломеративной кластеризации применительно к задачам кластеризации лингвистической экспертной информации. В описываемой работе приведена формальная постановка задачи кластеризации текстов, а также определена основная группа реализованных решений основанных на принципах агломеративной кластеризации: ROCK, CURE, CHAMELEON. Проведен детальный обзор каждого из представленных алгоритмов, а также сформулированы основные достоинства и недостатки каждого из них. Преимуществом данной работы можно считать совокупность представленных данных об алгоритмах, а также результаты сравнительного анализа, позволяющие в дальнейшем оценить целесообразность и потенциальную вероятность применения указанных решения из представленной группы алгоритмов агломеративной кластеризации. Новизна данной работы заключается в формировании обзорного анализа существующих подходов в области иерархической кластеризации для решения задач кластерного анализа лингвистической экспертной информации, а также формирование результатов проведенного сравнительного анализа рассмотренных алгоритмов.

Кластеризация; иерархическая кластеризация; агломеративная кластеризация; интеллектуальный анализ данных; кластеризация лингвистической экспертной информации.

P.S. Bulyga, V.M. Kureichik

AGGLOMERATIVE CLUSTERIZATION ALGORITHMS FOR THE PROBLEMS OF ANALYSIS OF LINGUISTIC EXPERT INFORMATION

This article discusses and presents the main problems and principles of the data clustering process, in particular, the principles and tasks of clustering text arrays of linguistic expert information. In the course of this work, the main difficulties arising in the design of such systems were identified, for example: the need for preprocessing data, reducing the size of the initial sample, etc. To effectively perform the presented tasks, the implemented solution must have an integrated approach that takes into account the efficiency indicators of methods aimed at solving individual subtasks, as well as the ability to provide high efficiency indicators for the implementation of each stage of the clustering process. In the presented work, various groups of hierarchical clustering algorithms are considered, in particular, a subgroup of agglomerative clustering algorithms was considered in relation to the problems of clustering linguistic expert information. In the described work, a formal statement of the text clustering problem is given, and the main group of implemented solutions based on the principles of agglomerative clustering is determined: ROCK, CURE, CHAMELEON. A detailed review of each of the presented algorithms is carried out, and the main advantages and disadvantages of each of them are formulated. The advantage of this work can be considered the totality of the presented data on the algorithms, as well as the results of a compara-

tive analysis, which make it possible to further assess the feasibility and potential probability of using these solutions from the presented group of agglomerative clustering algorithms. The novelty of this work lies in the formation of an overview analysis of existing approaches in the field of hierarchical clustering for solving the problems of cluster analysis of linguistic expert information, as well as the formation of the results of the comparative analysis of the considered algorithms.

Clustering; hierarchical clustering; agglomerative clustering; data mining; clustering of linguistic expert information.

Введение. При решении задач различного назначения в научной области и не только, может возникать проблема обработки и анализа полученных данных. Обработка и анализ данных – процесс позволяющий структурировать, выделять характерные признаки, определять зависимости, обобщать и сортировать данные различной структуры. Однако, решение подобных задач в текущих условиях может повлечь за собой ряд проблем, обусловленных следующими причинами:

Одной из причин возникновения трудностей при обработке и анализе информации можно выделить слабую структурированность входных данных, что в свою очередь увеличивает временные, ресурсные и прочие затраты на выполнение процесса обработки.

Ко второй причине можно отнести высокие темпы увеличения количества данных. Так, согласно статистической информации компании Statista каждый год объем генерации данных увеличивается на 30 %, таким образом на данный момент, показатель находится в районе 74 зеттабайта за 2021 год [1].

Для выполнения процесса анализа и обработки информации было предложено и реализовано множество подходов и алгоритмов, отличающихся друг от друга алгоритмической сложностью, временными и ресурсными затратами, особенностями эксплуатации а также различными вариациями форматов выходных данных. Зачастую, при обработке данных применяется два типа подходов: классификация или кластеризация.

Классификация. Классификация – процесс упорядочивания данных по заранее заданным классам, основываясь на выделенных признаках данных, с целью отражения взаимосвязей между ними. Классы, сформированные при помощи классификации – это множество данных, объединенных по некоторому или некоторым общим признакам объектов, отличающих данную совокупность объектов от других.

В качестве параметра классификации могут приниматься различные признаки, удовлетворяющие задачам и целям проводимой классификации. За основу класса берутся наиболее важные признаки данных, в полной мере отвечающие задачам проводимой классификации.

Результатом классификации объекта выступает указание данному объекту некоторого параметра (наименование или номер), позволяющего однозначно определить принадлежность данного объекта к конкретному классу.

Кластеризация. Кластеризация – процесс слияния экземпляров некоторой выборки данных в кластеры, основываясь на признаках подобия экземпляров выборки так, чтобы объекты принадлежащие различными кластерам обладали существенным отличием, а экземпляры одного кластера были наиболее схожи.

Декомпозиция множества подобных объектов позволяет в дальнейшем упростить процесс обработки и принятия решений, предоставляя возможность применения к каждому кластеру собственного алгоритма анализа [2]. Начальной информацией для кластерного анализа выступает матрица наблюдений (1):

$$T = \begin{bmatrix} t_{11} & t_{21} & \dots & t_{1n} \\ t_{12} & t_{22} & \dots & t_{2n} \\ \dots & \dots & \dots & \dots \\ t_{m1} & t_{m2} & \dots & t_{mn} \end{bmatrix}, \quad (1)$$

где, каждая строка матрицы (1) представлена значением n признаков конкретного объекта кластеризации. Обычно в качестве метрики подобия объектов применяется метрика расстояния. При этом, координаты прототипов неизвестны, а их вычисление происходит единовременно с процессом декомпозиции данных на кластеры.

В свою очередь, методы кластеризации можно разделить на две основные группы: четкие и нечеткие методы кластеризации. Четкие методы кластеризации декомпозируют множество экземпляров X на некоторое количество не пересекаемых подмножеств, при этом, каждый объект может принадлежать только одному кластеру. Нечеткие методы кластеризации предусматривают возможность одновременной принадлежность объекта к различным кластерам.

1. Задачи кластерного анализа. Задачи, в которых возможно применение кластерного анализа весьма разнообразны, и в большей степени зависят от постановки задачи и преследуемых целей. Можно выделить следующие основные задачи, достигаемые при проведении кластерного анализа:

1. Формирование структуры некоторого множества экземпляров X_m , при помощи декомпозиции множества на кластеры объектов по некоторым признакам схожести, тем самым упрощая дальнейший процесс обработки и принятия решений, с возможностью применения отдельных алгоритмов к разным кластерам.
2. Сжатие объема получаемых данных. Данный процесс необходим при поступлении на вход выборки данных большой размерности, при этом процесс сжатия будет основываться на сохранении наиболее типичных агентов каждого кластера.
3. Определение новизны путем детектирования и выделения нетривиальных объектов не принадлежащих по своим свойствам ни к одному из сформировавшихся кластеров.

В первом из рассмотренных случаев количество кластеров необходимо свести к минимум, что позволит проводить анализ выборки малой размерности, экземпляры которой при этом будут обладать схожими параметрами.

Во втором случае принципиально получение кластеров с высоким уровнем сходства объектов. Количество кластеров при этом может быть не ограничено, вплоть до равного значения числа полученных кластеров и элементов выборки.

В третьем случае ставится задача определения нетривиальных объектов, которые по своим свойствам не могут быть отнесены ни к одному из полученных кластеров. Таким образом, наибольший интерес будут представлять одинокие кластеры, обладающие наибольшим расстоянием к соседям.

Для выполнения всех вышеперечисленных задач возможно применение различных методов кластеризации. В данной работе будет рассмотрена одна и групп кластеризационных методов, а именно алгоритмы иерархической кластеризации. Иерархическая кластеризация – процесс поэтапного слияния множества объектов в единый кластер, либо представление множества в качестве единого кластера, с последующим дроблением, на более мелки части. Результатом данного процесса выступает не только декомпозиция множества или его объединение в единый кластер, но также формирование древообразной иерархической структуры.

1.1. Кластеризация экспертной лингвистической информации. Процесс кластеризации экспертной лингвистической информации является одним из наиболее важных и часто встречаемых этапов при решении большинства прикладных задач в области обработки и анализа естественного языка. Например, к наиболее частым задачам кластеризации текстовых массивов можно отнести: разработка и проектирование интеллектуальных ассистентов, разработка рекомендательных систем, и т.п.

В свою очередь, процесс кластеризации текстовых массивов данных можно разделить на два основных этапа:

1. Выделение основных признаков из текстовых массивов с последующим формированием множества векторов, выступающих в качестве математической модели кластеризуемого массива.

2. Основываясь на различных метриках схожести векторов, происходит определение кластеров текстовых данных.

Методы позволяющие выделять основные признаки из текстовых массивов разделяются на следующие группы:

1. Методы основанные на семантических признаках [3].
2. Методы основанные на Latent Semantic Analysis («LSA», «LDA» [4]).
3. Методы основанные на Word embeddings («word2vec», «BERT» [5]).
4. Методы основанные на «Мешке слов» («bag-of-words» [6]).

Отличительной особенностью перечисленных методов, является необходимость предварительного обучения. Подобное обучение осуществляется на текстовых массивах, которые включают в себя термины, применяемые в кластеризируемых текстах, с целью определения «семантической близости» слов, используемых при проектировании векторной модели текста.

Для некоторых прикладных задач анализ и обработка текстовой информации является достаточно сложной и трудозатратой процедурой. Например в задачах кластеризации научных текстов или аннотаций. Основной проблемой данного процесса является отсутствие достаточного количества текстов для обучения алгоритмов в некоторых областях научных исследований. В связи с вышеперечисленными проблемами, возникающими при решении различных прикладных задачах, разработка и проектирование методов и алгоритмов кластеризации лингвистической экспертной информации является актуальной и востребованной задачей.

2. Постановка задачи кластеризации. Общая постановка задачи кластеризации выглядит следующим образом: пусть задано некоторое множество $\boldsymbol{X}$ и функция, вычисляющая расстояние между экземплярами данного множества $\boldsymbol{d: X^2 \rightarrow [0, \infty)}$. Далее, необходимо разделить выборку объектов $\boldsymbol{X_m}$ принадлежащих данному множеству на кластеры так, чтобы расстояние между экземплярами выборки одного кластера, было минимальными, а расстояние между экземплярами различных кластеров – максимальным.

Алгоритм кластеризации – это представление $\boldsymbol{a:X \rightarrow Y}$, где $\boldsymbol{Y}$ – неопределенный каскад кластеров. Следовательно, некоторому экземпляру, принадлежащему выборке $\boldsymbol{x_i \in X_m}$ ставится в соответствии некоторый кластер $\boldsymbol{y_i \in Y}$. Обычно, размерность каскада кластеров $\boldsymbol{Y}$ заранее не определена, и для того, чтобы получить оптимальную размерность каскада, необходимо решить задачу вычисления количества кластеров, соответствующих некоторому критерию эффективности (качеству) кластеризации.

В свою очередь постановка задачи кластеризации документов выглядит немного иным образом, так данную постановку задачи можно сформулировать на основе задачи распознавания по Ю.И. Журавлеву [7]. Пусть задано множество $\boldsymbol{M}$ объектов $\boldsymbol{\{w_i\}}$; на данном множестве M проведем декомпозицию для получения конечного значения классов $\boldsymbol{\Psi_v, v = 1, \dots, g;\ \cup_v^g \Psi_v = M}$. При этом, объекты определяются значением признаков $\boldsymbol{x_j,\ j=1,\dots,n}$. Множество значений указанных признаков x_j определяет представление объекта $\boldsymbol{I(w)=\{x_1, \dots, x_n\}}$. Данные о принадлежности объекта w к какому-либо классу определяются в векторной форме: $\boldsymbol{\{I_1(w),\dots,I_g(w)\}}$, где $\boldsymbol{I_h(w)}$ содержит в себе данные о принадлежность объекта w к классу $\boldsymbol{\Psi_h}$ (2):

$$I_h(w) = \begin{cases} 1, \text{если } w \in \Psi_h \\ 0, \text{если } w \notin \Psi_h \\ \Delta, \text{если неопределен} \end{cases} \quad (2)$$

Заключение о принадлежность объекта $\boldsymbol{w}$ к классу $\boldsymbol{\Psi_h}$ принимается основываясь на значении сравнительного анализа расстояния между объектом и классом. При этом пороговые значения расстояния предопределенны заранее.

При группировании экземпляров кластера, необходимо вычислить сходство данных экземпляров в соответствии с метрикой расстояния, вычисляемой по следующим формулам [8]:

1. Манхэттенское расстояние, вычисляемое по формуле:

$$d(X,Y) = \sum_i |x_i - y_i|. \quad (3)$$

2. Евклидово расстояние, вычисляемое по формуле:

$$d_E(X,Y) = \sqrt{\sum_i (x_i - y_i)^2}\,. \quad (4)$$

3. Расстояние Чебышева, вычисляемое по формуле:

$$d(X,Y) = \max(|x_i - y_i|). \quad (5)$$

4. Квадрат евклидова расстояния, вычисляемое по формуле:

$$d_E(X,Y) = \sum_i (x_i - y_i)^2. \quad (6)$$

Рассмотрев возможные варианты вычисления метрики расстояния можно сделать заключение о высокой вариативности вычисления параметра схожести различных кластеров. Однако, для формирования более полного представления о методах кластеризации, необходимо представить классификацию методов.

Представление классификации методов кластеризации достаточно затруднительно в силу вероятности пересечения некоторых критериев классификации, а также возможность некоторых методов обладать функциями, которые могут быть отнесены к нескольким классам. Так, в работе [9] была предложена классификация методов кластеризации с точки зрения разработчика алгоритмов, результатом данной работы стало дерево методов кластерного анализа, представленное на рис. 1.

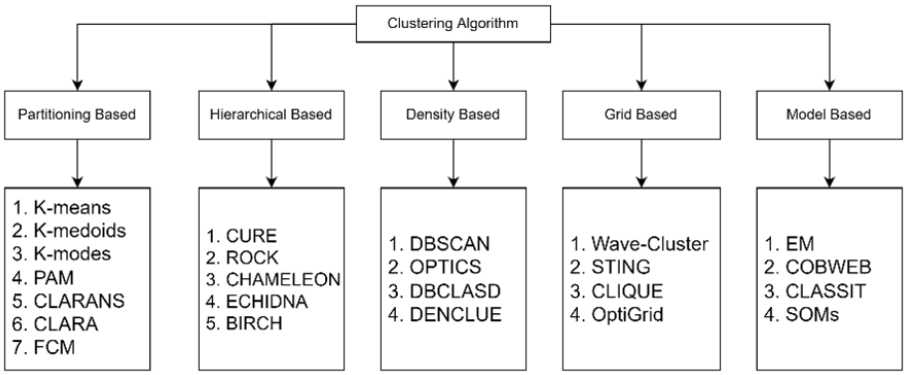


Рис.1. Дерево классификации методов кластеризации

В данной работе более подробно будут рассмотрены методы иерархической кластеризации в качестве одного из подходов решения задачи кластерного анализа лингвистической экспертной информации.

3. Иерархическая кластеризация. Алгоритмы иерархической кластеризации базируются на основном принципе поэтапной иерархической декомпозиции массива объектов. Подобные алгоритмы принято представлять в виде двух от-

дельных групп, разделяемых направлением построения иерархии: агломеративная кластеризация – объединение малых кластеров в общий; дивизионная кластеризация – представление всех экземпляров массива единым кластером, с последующим разделением на малые. Результатом работы данного типа методов кластерного анализа выступает система вложенных разбиений, визуализируемая при помощи дендрограмм.

В данной работе будет рассмотрена агломеративная кластеризация, также именуемые восходящей. В общем виде агломеративная кластеризация выполняется в соответствии следующим этапам [10]:

1. Инициализация некоторого множества кластеров $\boldsymbol{C_1}$, где каждый элемент массива данных $\boldsymbol{x_i}$ выступает кластером $\boldsymbol{C_q = \{x_1, \dots, x_n\}; q=\{1, \dots, n\}}$ (q – шаг итерации);
2. Определяем в $\boldsymbol{C_{q-1}}$ пару кластеров с минимальным расстоянием друг от друга $\boldsymbol{B}$ и $\boldsymbol{F}$, при условии, что $\boldsymbol{B \neq F}$;
3. Проводим этап слияние обнаруженных кластеров $\boldsymbol{B}$ и $\boldsymbol{F}$ в единый кластер $\boldsymbol{W = B \cup F}$.
4. Вычисление расстояния от сформированного кластера $\boldsymbol{W}$ к иному кластеру $\boldsymbol{S}$, данное значение определяется по расстояниям $\boldsymbol{(B,F)}$ $\boldsymbol{(B,S)}$ и $\boldsymbol{(F,S)}$.

Основным отличием большинства иерархических алгоритмов выступает метод расчета расстояния между кластеров. Ниже приведены основные и часто применяемые функции вычисления межкластерного расстояния [11]:

- Расстояние дальнего соседа:
$$R(W,S) = \max_{w \in W,\ s \in S} \rho(w,s). \quad (7)$$
- Расстояние ближнего соседа:
$$R(W,S) = \min_{w \in W,\ s \in S} \rho(w,s). \quad (8)$$
- Расстояние Уорда:
$$R(W,S) = \frac{|S|W}{|S|+|W|}\ \rho^2\left(\sum_{w \in W} \frac{w}{|W|}, \sum_{s \in S} \frac{s}{|S|}\right). \quad (9)$$
- Расстояние между центрами:
$$R(W,S) = \rho^2\left(\sum_{w \in W} \frac{w}{|W|}, \sum_{s \in S} \frac{s}{|S|}\right). \quad (10)$$
- Среднее расстояние:
$$R(W,S) = \frac{1}{|W||S|} \sum_{w \in W} \sum_{s \in S} \rho(w,s). \quad (11)$$

Большинство методов расчета расстояния можно представить в виде единой формулы, предложенной Г. Лансом и У. Уильямсом в 1983 году [12]:

$$R(B \cup F, S) = a_B R(B,S) + a_F R(F,S) + \beta(B,F) + \gamma|R(B,S) - R(F,S)|, \quad (12)$$

где, $\boldsymbol{B \cup F}$ – кластер, сформированный слиянием двух кластеров на предшествующей итерации, $\boldsymbol{S}$ – кластер, к которому необходимо рассчитать расстояние, $\boldsymbol{a_B, a_F, \beta, \gamma}$ – коэффициенты, определяющие разные функции расстояний.

В табл. 1 представлены вариации значений коэффициентов, определяющие соответствующие алгоритмы.

Таблица 1

Значение коэффициентов алгоритмом вычисления расстояния

Коэффициенты	Расстояние дальнего соседа	Расстояние ближнего соседа	Расстояние Уорда	Расстояние между центрами	Среднее расстояние
a_B	0,5	0,5	$\frac{\lvert B\rvert}{\lvert W\rvert}$	$\frac{\lvert B\rvert}{\lvert W\rvert}$	$\frac{\lvert S\rvert + \lvert B\rvert}{\lvert S\rvert + \lvert W\rvert}$
a_F	0,5	0,5	$\frac{\lvert F\rvert}{\lvert W\rvert}$	$\frac{\lvert F\rvert}{\lvert W\rvert}$	$\frac{\lvert S\rvert + \lvert F\rvert}{\lvert S\rvert + \lvert W\rvert}$
β	0	0	0	$-a_B\, a_F$	$\frac{-\lvert S\rvert}{\lvert S\rvert + \lvert W\rvert}$
γ	-0,5	0,5	0	0	0

Общее описание принципов и свойств алгоритмов агломеративной кластеризации, дает представление об их функционировании, и является необходимым минимумом для дальнейшего рассмотрения существующих алгоритмов группы агломеративной иерархической кластеризации.

3.1. Алгоритм CURE (Clustering Using Representatives). Алгоритм кластеризации CURE представляет каждый элемент входной выборки в виде отдельного кластера (точки), с последующим поэтапным слиянием всей выборки в единый кластер [13].

Принцип работы данного алгоритма заключается в следующем: задается константное число точек, находящихся на максимальном удалении от центра тяжести кластера. Далее воспроизводится процесс группирования данных точек к центру тяжести на параметр ***J***. После завершения процесса группирования, данные точки будут определены в качестве представителей кластера, которому они принадлежат. Кластеры с наиболее схожими наборами репрезентативных точек буду объединены на следующей итерации алгоритма. За счет применения точек-представителей, алгоритм CURE менее подвержен выбросам и способен определять кластеры вариативной формы и различной размерности.

Описание этапов работы алгоритма CURE:

1. На первой итерации алгоритма необходимо случайным образом отобрать часть точек, с последующим размещением их в памяти. Далее проводится объединение точек с подобными параметрами при помощи иерархического метода, в заранее заданное количество кластеров.
2. У каждого кластера определяется ***C*** точек-представителей, которые максимально удалены друг от друга, при этом значение ***C*** остается неизменным.
3. Производим группирование и объединение кластеров, с наиболее схожими каскадами точек-представителей. В случае, если чисто кластеров не было достигнуто, необходимо вернуться к шагу 2. Для того, чтобы во время процесса группирования кластеров, не нужно было каждый раз производить отбор из всех ***C*** точек-представителей, необходимо производить отбор из ***2C*** точек объединенных кластеров.
4. Отобранные точки сдвигаются на последующем шаге на параметр ***J*** к центроиду кластера.

Входными данными для данного алгоритма выступает: матрица $\boldsymbol{x} \in \boldsymbol{R^{m*n}}$, ***h*** – количество кластеров, ***c*** – количество точек-представителей, параметр ***α, γ, m*** – количество точек для выполнения кластеризации, ***n***-размерность.

α – данный параметр предназначен для сжатия точек-представления к центроиду кластера, сами авторы в своей работе рекомендовали задавать данный параметр в диапазоне [0.2; 0.7], поскольку данный диапазон позволяет обнаружить кластеры несферической формы и подавлять нечастые выбросы.

γ – данный параметр определяется в качестве некоторой выборки от m точек.

c – данный параметр отвечает за количество точек-представлений. Значение данного параметра будет выбираться в зависимости от исходных данных.

Сложность данного алгоритма будет рассчитываться следующим образом. Так как входными данными для алгоритма CURE выступает n репрезентативных точек в d-мерном пространстве, а также константное число кластеров h, тогда, каждой точке требуется рассчитать расстояние к другой. Сложность данного этапа будет составлять: $O(n)$. В случае поиска соседних кластеров способом перебора, сложность данного поиска также будет составлять $O(n)$, но при эффективном поиске, сложность можно сократить до $(log_2 n)$ [14]. По завершению этапа поиска, каждую точку требуется сместить к центроиду, сложность данной операции составит $\mathbf{O(n)}$. Следовательно, наиболее длительное время выполнения алгоритма составит $\mathbf{O(n^2 log_2 n)}$. В случае двумерного пространства временная сложность составит: $\mathbf{O(n^2)}$. В заключении можно констатировать, что алгоритм CURE выполняется за полиномиальное время [15].

Математическое описание алгоритма:

Пусть задана некоторая матрица $x \in R^{m*n}$, число кластеров равное $\mathbf{h}$, количество точек-представителей с, параметры α,γ. Любая строка из матрицы x представляет собой некоторую точку в d-мерном пространстве.

Вычисление расстояния между точками $a = (a_1, \ldots, a_n)$ и $b = (b_1, \ldots, b_n)$ будет производиться по формуле:

$$d(a,b) = \sqrt{\sum_{i=1}^{d}(a_i - b_i)^2}. \quad (13)$$

Этап «Инициализации». Отбирается γm строк из матрицы x некоторым случайным образом, в памяти выделяется «куча», часть которой будет выделена для хранения дерева.

Этап «Кластеризация». В начале каждой итерации производится представление каждой отобранной точки в кластер с центроидом в этой самой точке. Далее производим выполнение следующих шагов до тех пор, пока количество кластеров не будет равно h.

1. Для i-го кластера производятся вычисления центроида μ_i в соответствии с формулой:

$$\mu_i = \frac{1}{s_i}\sum_{j \in S_i} x_j \, , \quad (14)$$

где S_i – порядковый номер строк матрицы x, принадлежащие кластеру i; s_i – число экземпляров множества S_i; x_j – j-я строка матрицы x.

2. Производится детектирование двух кластеров подходящих для слияния. Критерием выбора в данном случае может выступать: минимальное расстояние между экземплярами, принадлежащими данной паре кластеров или минимальное расстояние между центроидами двух кластеров с последующим объединением данных кластеров.

Этап «Смещение к центроиду». Каждая точка-представитель x^y k-го кластера выполняет смещение к центроиду μ_k в α раз, по формуле:

$$\widehat{x^y} = \alpha\mu_k + (1-\alpha)x^y \quad (15)$$

тем самым получая множество $\widehat{R_k}$ из с точек, которые в дальнейшем будут использоваться в качестве представителя кластера.

Этап «Присваивания». Данный этап присваивает неиспользованные точки к кластерам, путем высчитывания расстояния до каждой точки-представителя.

В заключении можно сказать следующее: алгоритм CURE осуществляет иерархическую кластеризацию при помощи наборов определяющих точек, применение данного алгоритма возможно только в Евклидовом пространстве и для наборов данным большой размерности.

К преимуществам данного алгоритма можно отнести: выполнение кластеризации на высоком уровне; определение кластеров вариативной формы и различной размерности; обладание линейной зависимости к условиям места хранения данных и квадратичной сложности для данных высокой размерности.

Недостатки: потребность в определении пороговых значений, значений количества кластеров; возможность работы исключительно с числовыми данными; эффективные показатели только с данными малой размерности.

3.2. Алгоритм ROCK (Robust Clustering using links). Характерной особенностью данного алгоритма является то, что в нем применяются взаимосвязи между точками, в отличие от алгоритмов основанных на применении различных метрик [16]. Данный подход в кластеризации позволяет совершенствовать некоторые глобальные зависимости, а также является более результативным при обработке данных, свойства которых определяются небольшим конечным объемом параметров.

К базовым понятиям алгоритма можно отнести «соседство пары точек». Так, пусть задана некоторая функция схожести $sim(p_i, p_j)$, которая принимает значение в диапазоне [0; 1], выражающая близость или подобность точек p_i и p_j, полагая что 0 – абсолютная отдаленность, 1 – абсолютная близость. Тогда при наличии границы θ между 0 и 1, при условии что $sim(p_i, p_j) \geq \theta$, точки p_i и p_j будут считаться близкими. При этом, выбор параметра граничного значения θ и функции sim будет зависеть от особенности реализации и входных данных.

Второе базовое понятие – связь между точками. Так, пусть задана некоторая функция связи $link(p_i, p_j)$ представляемая как численное значение общих соседей точек p_i и p_j. Из вышеописанного определения можно сделать заключение, что чем больше значение связи, тем выше вероятность того, что рассматриваемые точки принадлежат единому для них кластеру. Данный подход кластеризации является более обширным, в сравнении с применяемой в иных алгоритмах метрикой расстояния.

Алгоритм ROCK включает в себя два основных этапа. Так, первоначально задано n точек и k – кластеров (желаемое количество кластеров). На первом шаге алгоритм производит вычисление значений связей $link(p_i, p_j)$ между всеми парами точек, при этом каждая точка будет считаться отдельным кластером [17]. Каждому кластеру i формируется локальная куча $q[i]$, которая включает в себя все кластеры j, с которыми связь не равна нулю. Одновременно с этим формируя глобальную «кучу» Q, включающую в себя все кластеры. Второй шаг является циклом, на каждой итерации которого происходит слияние двух кластеров с наибольшими значениями функции полезности $g(i, j)$, после чего происходят соответствующие преобразования в куче. Завершение алгоритма возможно в двух случаях: все связи между оставшимися кластера нулевые, либо в остатке получено k кластеров.

Сложность данного алгоритма будет вычисляться следующим образом. Вычисление значений связей можно представить в виде перемножения пары матриц размерности n, что соответствует $O(n^{2.37})$. При этом затраты в памяти на хранение связей не должно быть более чем $n(n+1)/2$, в случае соседства пары точек. Впрочем в подавляющей части случаев медианное количество соседей составляет m_a а максимальное число m_m что в свою очередь меньше n, поэтому оценка сложности зависит от величин представленных параметров. Сложность формирования массива соседей оценивается как $O(n^2)$. Так, после завершения формирования массива для любой точки, алгоритм рассматривает все соседние пары, при этом, для каж-

дой пары формируется единственная связь. Если m_i – количество соседних точек i, тогда необходимо для нее увеличить число связей на m_i^2 раз. Следовательно сложность алгоритма составит $\sum m_i^2$ и оценивается как $O(n\, m_m\, m_a)$. Так как любая точка i может обладать не более чем $min\{m_m\, m_i,\, n\}$ связями, тогда полные затраты памяти не превышают $O(min\{n\, m_m\, m_a,\, n^2\})$ [18].

Условные обозначения используемые в расчете сложности алгоритма: m_a – медианное количество соседей; m_m – максимально возможное количество соседей; m_i – количество соседей точки i; n – размерность входных данных.

Математическое описание алгоритма. Пусть дано некоторое множество S состоящее из n экземпляров и некоторое число кластеров k. Для всех $p_i, p_j \in S$ задана функция схожести: $sim(p_i, p_j)$ в диапазоне [0, 1], также задано пороговое значение θ в интервале [0, 1] и функция $f(\theta)$.

Необходимо декомпозировать множество S на k не пересекаемых кластеров C_k таким образом, чтобы значение целевой функции E_l было максимальным. Полагая что $p_i, p_j \in S$ определим следующие условия функций:

1. Функция определения соседства пары точек:

$$neib(p_i, p_j) = \begin{cases} 1, sim(p_i, p_j) \geq \theta \\ 0, sim(p_i, p_j) < \theta \end{cases}. \quad (16)$$

2. Функция определения числа связей между точками:

$$link(p_i, p_j) = \sum_{s \in S} neib(p_i, s) neib(p_j, s). \quad (17)$$

3. Целевая функция, предполагая что n_i количество элементов в C_i

$$E_l = \sum_{i=1}^{k} n_i \sum_{p_q, p_r \in C_i} \frac{link(p_q, p_r)}{n_i^{1+2f(\theta)}}. \quad (18)$$

4. Функция связи между кластерами C_i, C_j:

$$link[C_i, C_j] = \sum_{p_q \in C_i, p_r \in C_j} link(p_i, p_j). \quad (19)$$

5. Функция полезности, позволяет оценить целесообразность слияния кластеров C_i, C_j:

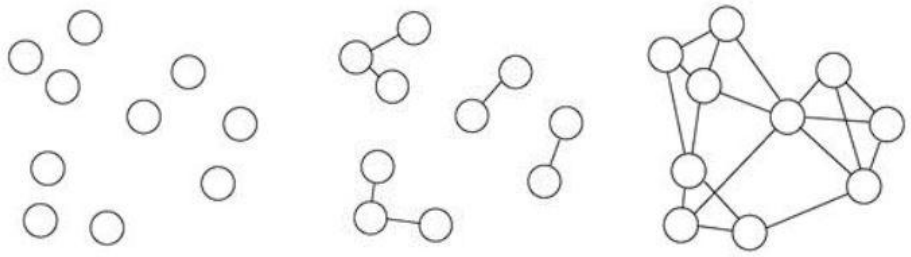
$$g(C_i, C_j) = \frac{link[C_i, C_j]}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}}. \quad (20)$$

В алгоритме ROCK изначально каждый элемент разбивается на n подмножеств, совершая $n\text{-}k$ итераций, на каждой из которых происходит слияние множеств, для которых показатель функции полезности $g(i, j) \rightarrow max$. В заключении можно констатировать, что алгоритм является достаточно устойчивым, и детерминированным.

3.3. Алгоритм CHAMELEON (Hierarchical Clustering Using Dynamic Modeling). Принцип работы данного подхода заключается в итеративном слиянии пары соседних кластеров. На первоначальном этапе данный алгоритм применяет декомпозицию графа с целью получения каскада кластеров, относительно малой величины. На следующем этапе с целью объединения кластеров полученных на первом этапе применяется агломеративная кластеризация. Таким образом, рассматриваемый алгоритм CHAMELEON по своей сути является гибридным алгоритмом включающим в себя особенности классических иерархических методов и графо-ориентированных методов [19].

Так, на первом этапе, основываясь на принципах графо-ориентированного подхода, применяется формирование графа базируясь на матрице сходства по принципу k-ближних соседей. Вершины подобного графа соединены ребрами, в

случае если объект соответствующий какой-либо из данных вершин попадает в k-ближайших объектов. На рис. 2 представлены: (а) некоторые объекты в пространстве; (б) граф построенный по принципе 1-го ближнего соседа; (в) граф построенный по принципу 3-х ближних соседей.



(а) (б) (в)

Рис. 2. Процесс формирования графа алгоритмом CHAMELEON

В последствии CHAMELEON последовательно декомпозирует сформированный граф на некоторое множество малых подграфов. На каждой итерации определяется подграф, включающий в себя наибольшее количество вершин. Данный граф будет разделен на два подграфа таким образом, чтобы разделить графа был минимальным, и каждый из вновь сформированных графов содержал некоторый процент вершин исходного графа. Этап разделения будет завершен только тогда, когда исходный граф будет содержать количество вершин не меньшее чем заданное пороговое значение. Результатом данных итераций должно быть множество связных графов, которые будут интерпретироваться множеством начальных кластеров.

На следующем этапе происходит поэтапное слияние кластеров, применяя значения их взаимосвязанности и относительного подобия. В случае если оба перечисленных значения у обоих кластеров достаточно большие, будет произведено слияние. Относительная взаимосвязанность двух кластеров определяется абсолютной взаимосвязанностью кластеров, нормализованных с учетом внутренних связей каждого кластера. Нормализация необходима для того, чтобы исключить возможность возникновения тенденции к приоритетному слиянию кластеров большой размерности [20]. На рис. 3 представлена схема работы алгоритма.

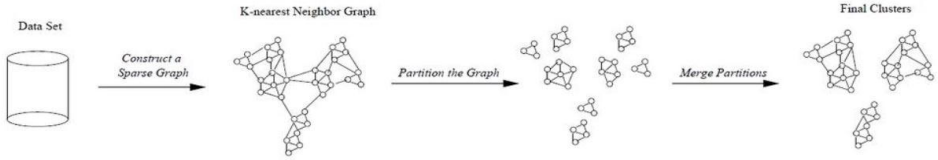


Рис. 3. Схема работы CHAMELEON

Вычисление сложности алгоритма: первоначально определим условные обозначения, используемые в ходе описания сложности алгоритма: $\boldsymbol{m}$ – количество исходных кластеров сформированных после второго шага алгоритма, $\boldsymbol{n}$ – размерность исходной выборки данных; для упрощения анализа положим размерность любого кластера равной $\boldsymbol{n/m}$.

Формирование графа k-ближних соседей. Для определения k-ближних соседей некоторой единой фиксированной вершины, в общем случае потребуется $\boldsymbol{O(n)}$ шагов, соответственно, для выполнения данного этапа необходимо $\boldsymbol{O(n^2)}$ шагов.

Формирование стартового каскада кластеров. Пусть задан некоторый граф ***G=(V,E).*** Первая итерация алгоритма – декомпозиция кластера максимальной размерности на пару подкластеров. Данное действие может быть выполнено за ***O(|V| +|E|)*** шагов. Поскольку первоначальной точкой данной итерации выступает граф k-ближних соседей ***|E|=O(|V|),*** соответственно, шаг может быть завершен за ***O(|V|)=O(n)*** шагов. Итоговая сложность данного этапа может равняться ***O(n log(n/m)) = O(n log(n)),*** поскольку по итогу будет получено m кластеров.

Определение сложности третьего этапа всецело будет зависеть от вычислительной сложности относительных взаимных связностей. При этом, время затрачиваемое для декомпозиции каждого из первоначальных ***m*** кластеров будет равняться ***O(n/m),*** в итого потребуется ***O(n).*** После этого, необходимо выполнить ***m-1*** слияний ***m*** первоначальных кластеров, следовательно, итоговая сложность вычислений схожести составит: ***O(nm).*** Исходя из всего вышесказанного, можно сделать заключение о том, что итоговая сложность алгоритма составит: $\boldsymbol{O(n^2 + nm + n\log n + m^2\log(m))}$.

Математическое описание алгоритма CHAMELEON. Этапы алгоритма:

♦ Определение набора данных в виде графа G=(V, E) при помощи алгоритма k-ближних соседей: $\boldsymbol{e = (v_i, v_j) \in E}$, при условии $\boldsymbol{dist(v_i, v_j) \in sorted(\{r | r = dist(v_i, v_k),\ k = 1, .., n\})\ [1 \ldots k]}$, причем ***weight(e)*** = $\boldsymbol{dist(v_i, v_j)}$;

♦ Декомпозиция графа на малые кластеры. Для выполнения данного этапа потребуется сформировать множество C = G и выполнить цикл со следующими условиями на каждой итерации:

- Обнаружить $C_{greatest}$ таким, чтобы $|C_{greatest}| = \max_{1 \le j \le |C|} |C_j|$;
- В случае если $\frac{|C_{greatest}|}{|V|} > minsize$, тогда необходимо декомпозировать $C_{greatest}$ на пару кластеров C_1 и C_2. В противном случае, цикл завершается;

♦ Формирование стартового каскада кластеров $C_{current}$ = C. Далее необходимо выполнить новый цикл со следующими условиями на каждой итерации:

- Каждой паре кластеров C_i, $C_j \in C_{current}$ необходимо рассчитать следующие значения:

$EC(C_i, C_j) = \sum_{e \in con\ n(C_i, C_j)} weight(e)$, где $con\ n(C_i, C_j) = \{(v, \omega) | v \in C_i, \omega \in C_j\}$;

$EC(C_i) = \sum_{e \in mincut(C_i)} weight(e)$, где $mincut(C_i)$ – минимально возможный разрез C_i, точно также происходит вычисление $EC(C_j)$;

$$\bar{S}_{EC(C_i, C_j)} = \frac{EC(C_i, C_j)}{|con\ n(C_i, C_j)|};$$

$$\bar{s}_{EC(C_i)} = \frac{EC(C_i)}{|mincut(C_i)|};$$

- Выполнение процесса расчета взаимосвязанности и относительного взаимного сходства пары нынешних кластеров:

$$RI(C_i, C_j) = \frac{|EC(C_i, C_j)|}{|EC(C_i)| + |EC(C_j)|};$$

$$RC(C_i, C_j) = \frac{\bar{S}_{EC(C_i, C_j)}}{\frac{|C_i|}{|C_i| + |C_j|} \bar{S}_{EC(C_i)} + \frac{|C_j|}{|C_i| + |C_j|} \bar{S}_{EC(j)}};$$

♦ В случае если $RI(C_i, C_j)^{\alpha} RC(C_i, C_j) > t$, тогда необходимо выполнение объединение данной пары кластеров, если наоборот – перейти к следующему этапу. На данном этапе $\boldsymbol{\alpha}$ – параметр, при помощи которого можно выставить приоритет одному из показателей, ***t*** – параметр, демонстрирующий минимальную степень сходства кластеров, при котором возможно объединение. В случае если на одном из этом не было объединения кластеров, цикл необходимо завершить;

В завершение можно сказать следующее: CHAMELEON разделяется на части, которые исполняются поэтапно и обособленно друг от друга, в рамках каждой из частей возможна параллельная реализация.

Представленное описание алгоритмов агломеративной кластеризации позволяют провести сравнительный анализ, необходимый для определения целесообразности применения какого-либо из представленных алгоритмов, в качестве решения задачи кластеризации текстовых данных. Результаты сравнительного анализа представлены в табл. 2.

Таблица 2

Сравнительный анализ алгоритмов агломеративной кластеризации

Алгоритм	**Тип моделирования**	**Преимущества**	**Недостатки**
ROCK	Статическое моделирование	Надежность алгоритма; Применим для обработки больших массивов данных;	Низкая скорость работы; Слабая масштабируемость для множества объектов большой размерности;
CURE	Статическое моделирование	Устойчив к выбросам; Применим для обработки больших массивов данных; Процесс выполнения без ущерба качества кластеризации; Легкость распознавания кластеров произвольной формы;	Необходимо задание порогового значения плотности объектов и числа кластеров; Возможность работы только с числовыми значениями;
CHAMELEON	Динамическое моделирование	Эффективность применения для больших массивов данных; Возможность детектирования кластеров различной формы;	Не применим для пространств большой размерности; Высокая временная сложность в пространстве большой размерности;

Комментируя представленные результаты сравнительного анализа можно заявить следующее. Рассматриваемый алгоритм CURE не предназначен для работы с данными текстового формата, соответственно применение данного алгоритма для решения поставленной задачи невозможно.

В свою очередь алгоритм ROCK несмотря на выделенные недостатки связанные со скоростью выполнения кластерного анализа обладает достаточным потенциалом для дальнейших исследований в качестве возможного решения задачи кластеризации текстовых данных. Поскольку обладает свойством возможности работы с большими массивами и достаточной устойчивостью к выбросам.

Алгоритм CHAMELEON также обладает достаточным потенциалом для дальнейших исследований, поскольку возможен в применении для массивов больших данных, а также способен обнаруживать кластеры различной формы, что является достаточно частым явлением в задачах кластеризации текстовых данных.

Заключение. Данная работа посвящена обзорному анализу методов кластерного анализа в рамках кластеризации экспертной лингвистической информации. По итогам написания данной работы были определены основные положения и за-

дачи процесса кластеризации, а также приведены основные выводы касающиеся существующих решений в данной области в рамках группы методов агломеративной кластеризации. По результатам сравнительного анализа методов CURE, ROCK, CHAMELEON было сделано заключение о нецелесообразности дальнейшего исследования метода CURE в силу отсутствия поддержки данного метода работы с текстовыми данными. Для дальнейших исследований по данному вопросу предлагается к рассмотрению реализация и сравнение методов ROCK и CHAMELEON. Реализация данных методов позволит провести экспериментальные исследования с целью получения показателей эффективности и определить подходящий метод для дальнейшего применения в будущих исследованиях.

Данная работа выступаем теоретическим базисом для начала исследований в области кластеризации текстовым массивов данных, а также экспертной лингвистической информации, и носит обзорный характер. Новизна данной работы заключается в рассмотрении основных методов и подходов к решению подобного рода задачи, а также выделения основных достоинств и недостатков каждого из представленных методов.

Отличием данной работы от существующих научных трудов является проведенный аналитический обзор развития теории кластерного анализа, а также уточнение теоретических конструкций по данной тематике. Представлен к рассмотрению сравнительный анализ методов агломеративной кластеризации с последующим выделением основных достоинств не недостатков рассмотренных методов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Volume of data/information created, captured, copied and consumed worldwide from 2021 year. – https://www.statista.com/statistics/871513/worldwide-data-created/ (дата обращения 22.12.2021).
2. *Заргарян Ю.А., Затылкин В.В.* Классификация и нечеткая кластеризация в задачах принятия решений // Известия ЮФУ. Технические науки. – 2010. – № 1 (102). – С. 140-144.
3. *Staab S., Hotho A.* Ontology-based text document clustering // Proc. International Intelligent Information System // Intelligent Information Processing and Web Mining Conference (IIS: IIPWM'03). – 2003. –P. 451-452.
4. *Hofmann T.* Probabilistic latent semantic indexing // Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999). – 1999. – P. 50-57.
5. *Devlin J., Chang M., Lee K.* BERT: Pretraining of deep bidirectional transformers for language understanding // ArXiv. – 2018. – P. 42-48.
6. *Whissell J.S., Clarke C.L.* Improving document clustering using Okapi BM25 feature weighting // Information Retrieval. – 2011. – Vol. 14, No. 5. – P. 466-487.
7. *Журавлев Ю.И.* Об алгебраическом подходе к решению задач распознавания или классификации // Проблемы кибернетики. – 1978. – Т. 33. – С. 5-68.
8. *Ермоченко С.А.* Концепция применения Mapreduce в иерархической агломеративной кластеризации // Вестнік Віцебскага дзяржаунага універсітэта. – 2019. – № 3 (104). – С. 28-37.
9. *Махрусе Н.* Современные тенденции методов интеллектуального анализа данных: метод кластеризации // Московский экономический журнал. – 2019. – № 6. – С. 359-377.
10. *Бильгаева Л.П., Заиграева Е.В.* Оценка качества агломеративной кластеризации // Приложение математики в экономических и технических исследованиях. – 2020. – № 1 (10). – С. 43-53.
11. *Кирпичников А.П., Ризаев И.С. Тахавова Э.Г. и др.* Разработка эффективного алгоритма иерархической кластеризации // Вестник Технологического университета. – 2019. – Т. 22, № 10. – С. 117-122.
12. *Уиллиамс У.Т., Ланс Дж.Н.* Методы иерархической классификации // Статистические методы для ЭВМ / под ред. К. Энслейна, Э. Рэлстона, Г.С. Уилфа. – М.: Наука, 1986. – С. 269-300.

13. *Гладилин А.В., Гамазина В.С.* Иерархические методы кластеризации данных и их характеристики // Информационные технологии в экономических и технических задачах. – Пенза: Пензенский государственный технологический университет, 2016. – С. 200-202.
14. *Sudipto G., Rajeev R., Kyuseok S.* CURE: an efficient clustering algorithm for large databases // SIGMOD '98 Pro. of the 1998 ACM SIGMOD international conference on Management of data. – 1998. – P. 73-84.
15. *Дубаков А.А., Воробьев А.М.* Разработка алгоритма иерархической агломеративной кластеризации для анализа текстовых документов Воробьев // Математическое и информационное моделирование. – Тюмень: Тюменский государственный университет, 2018. – С. 246-255.
16. *Давыдов О.А.* Анализ существующих алгоритмов кластеризации (Ч. 1) // Вестник Тихоокеанского государственного университета. – 2020. – № 1 (56). – С. 27-36.
17. *Михайлов А.С., Шабанов В.Ю.* Разработка алгоритм кластеризации номинальных данных // Информационные технологии. – Новосибирск: Новосибирский национальный исследовательский государственный университет, 2019. – С. 101-107.
18. *Холда О.С., Извозчикова В.В.* Разработка алгоритма обработки больших массивов данных // Глобализация науки и техники в условиях кризиса. – Ростов-на-Дону: Изд-во ВВМ», 2021. – С. 48-53.
19. *Безверхий О.А., Самохвалова С.Г.* Кластеризация большого объема текстовых поисковых запросов // Ученые заметки ТОГУ. – 2016. – Т. 7, № 3-1. – С. 104-110.
20. *Шатовская Т.Б., Заремская А.А.* Экспериментальные результаты исследования качества кластеризации разнообразных наборов данных с помощью модифицированного алгоритма хамелеона // ScienceRise. – 2015. – Т. 3, № 2 (8). – С. 11-16.

REFERENCES

1. Volume of data/information created, captured, copied and consumed worldwide from 2021 year. Available at: https://www.statista.com/statistics/871513/worldwide-data-created/ (accessed 22 December 2021).
2. *Zargaryan Yu.A., Zatylkin V.V.* Klassifikatsiya i nechetkaya klasterizatsiya v zadachakh prinyatiya resheniy [Classification and fuzzy clustering in decision-making tasks], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 1 (102), pp. 140-144.
3. *Staab S., Hotho A.* Ontology-based text document clustering // Proc. International Intelligent Information System, *Intelligent Information Processing and Web Mining Conference (IIS: IIPWM'03),* 2003, pp. 451-452.
4. *Hofmann T.* Probabilistic latent semantic indexing, *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999),* 1999, pp. 50-57.
5. *Devlin J., Chang M., Lee K.* BERT: Pretraining of deep bidirectional transformers for language understanding, *ArXiv*, 2018, pp. 42-48.
6. *Whissell J.S., Clarke C.L.* Improving document clustering using Okapi BM25 feature weighting, *Information Retrieval*, 2011, Vol. 14, No. 5, pp. 466-487.
7. *Zhuravlev Yu.I.* Ob algebraicheskom podkhode k resheniyu zadach raspoznavaniya ili klassifikatsii [On an algebraic approach to solving problems of recognition or classification], *Problemy kibernetiki* [Problems of Cybernetics], 1978, Vol. 33, pp. 5-68.
8. *Ermochenko S.A.* Kontseptsiya primeneniya Mapreduce v ierarkhicheskoy aglomerativnoy klasterizatsii [The concept of using Mapreduce in hierarchical agglomerative clustering], *Vestnik Vitsebskaga dzyarzhaunaga universiteta* [Vestnik Vitsebskaga dzyarzhaunaga universiteta], 2019, No. 3 (104), pp. 28-37.
9. *Makhruse N.* Sovremennye tendentsii metodov intellektual'nogo analiza dannykh: metod klasterizatsii [Modern trends in data mining methods: clustering method], *Moskovskiy ekonomicheskiy zhurnal* [Moscow Economic Journal], 2019, No. 6, pp. 359-377.
10. *Bil'gaeva L.P., Zaigraeva E.V.* Otsenka kachestva aglomerativnoy klasterizatsii [Assessment of the quality of agglomerative clustering], *Prilozhenie matematiki v ekonomicheskikh i tekhnicheskikh issledovaniyakh* [Application of Mathematics in Economic and Technical Research], 2020, No. 1 (10), pp. 43-53.

11. *Kirpichnikov A.P., Rizaev I.S. Takhavova E.G., and others.* Razrabotka effektivnogo algoritma ierarkhicheskoy klasterizatsii [Development of an effective hierarchical clustering algorithm], *Vestnik Tekhnologicheskogo universiteta* [Bulletin of the Technological University], 2019, Vol. 22, No. 10, pp. 117-122.
12. *Uilliams U.T., Lans Dzh.N.* Metody ierarkhicheskoy klassifikatsii [Methods for hierarchical classification // Statistical methods for computers], *Statisticheskie metody dlya EVM* [Statistical Methods for Computers], ed. by K. Ensleyna, E. Relstona, G.S. Uilfa. Moscow: Nauka, 1986, pp. 269-300.
13. *Gladilin A.V., Gamazina V.S.* Ierarkhicheskie metody klasterizatsii dannykh i ikh kharakteristiki [Hierarchical methods of data clustering and their characteristics], *Informatsionnye tekhnologii v ekonomicheskikh i tekhnicheskikh zadachakh* [Information Technologies in Economic and Technical Problems]. Penza: Penzenskiy gosudarstvennyy tekhnologicheskiy universitet, 2016, pp. 200-202.
14. *Sudipto G., Rajeev R., Kyuseok S.* CURE: an efficient clustering algorithm for large databases, *SIGMOD '98 Pro. of the 1998 ACM SIGMOD international conference on Management of data*, 1998, pp. 73-84.
15. *Dubakov A.A., Vorob'ev A.M.* Razrabotka algoritma ierarkhicheskoy aglomerativnoy klasterizatsii dlya analiza tekstovykh dokumentov Vorob'ev [Development of an algorithm for hierarchical agglomerative clustering for the analysis of text documents Vorobiev], *Matematicheskoe i informatsionnoe modelirovanie* [Mathematical and Information Modeling]. Tyumen': Tyumenskiy gosudarstvennyy universitet, 2018, pp. 246-255.
16. *Davydov O.A.* Analiz sushchestvuyushchikh algoritmov klasterizatsii [Analysis of existing clustering algorithms (Part 1)], *Vestnik Tikhookeanskogo gosudarstvennogo universiteta* [Bulletin of the Pacific State University], 2020, No. 1 (56), pp. 27-36.
17. *Mikhaylov A.S., SHabanov V.Yu.* Razrabotka algoritm klasterizatsii nominal'nykh dannykh [Development of an algorithm for clustering nominal data], *Informatsionnye tekhnologii* [Information Technologies]. Novosibirsk: Novosibirskiy natsional'nyy issledovatel'skiy gosudarstvennyy universitet, 2019, pp. 101-107.
18. *Kholda O.S., Izvozchikova V.V.* Razrabotka algoritma obrabotki bol'shikh massivov dannykh [Development of an algorithm for processing large data arrays], *Globalizatsiya nauki i tekhniki v usloviyakh krizisa* [Globalization of Science and Technology in a Crisis]. Rostov-on-Donu: Izd-vo VVM», 2021, pp. 48-53.
19. *Bezverkhiy O.A., Samokhvalova S.G.* Klasterizatsiya bol'shogo ob''ema tekstovykh poiskovykh zaprosov [Clustering of a large volume of text search queries], *Uchenye zametki TOGU* [Scientific Notes of PNU], 2016, Vol. 7, No. 3-1, pp. 104-110.
20. *Shatovskaya T.B., Zaremskaya A.A.* Eksperimental'nye rezul'taty issledovaniya kachestva klasterizatsii raznoobraznykh naborov dannykh s pomoshch'yu modifitsirovannogo algoritma khameleona [Experimental results of studying the quality of clustering of various data sets using a modified chameleon algorithm], *ScienceRise,* 2015, Vol. 3, No. 2 (8), pp. 11-16.

Статью рекомендовал к опубликованию д.т.н., профессор Н.И. Витиска.

Булыга Филипп Сергеевич – Южный федеральный университет; e-mail: bulyga@sfedu.ru; г. Таганрог, Россия; тел.: +79001330866; кафедра САПР; аспирант.

Курейчик Виктор Михайлович – e-mail: vmkureychik@sfedu.ru; тел.: +79282132730; кафедра САПР; д.т.н.; профессор.

Bulyga Philip Sergeevich – Southern Federal University; e-mail: bulyga@sfedu.ru; Taganrog, Russia; phone: +79001330866; the department of computer-aided design; graduate student.

Kureichik Viktor Mikhailovich – e-mail: vmkureychik@sfedu.ru; phone: +79282132730; the department of computer-aided design; dr. of eng. sc; professor.