21. *Lebedev B.K., Lebedev O.B., Lebedeva E.M.* Razbienie na klassy metodom al'ternativnoy kollektivnoy adaptatsii [Partition a class method alternative collective adaptation], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, No. 7 (180), pp. 89-101.
22. *Karpenko A.P.* Populyatsionnye algoritmy global'noy poiskovoy optimizatsii. Obzor novykh i maloizvestnykh algoritmov [Population algorithms of global search engine optimization. Overview of new and little-known algorithms], *Informatsionnye tekhnologii* [Information Technology], 2012, No. 7, pp. 1-32.
23. *Rodzin S., Rodzina L.* Theory of bioinspired search for optimal solutions and its application for the processing of problem-oriented knowledge, *Proceeding of the 8th IEEE International Conference «Application of Information and Communication Technologies»,* 2014, pp. 142-147.
24. *Kureychik V.V., Polupanova E.E.* Evolyutsionnaya optimizatsiya na osnove algoritma kolonii pchel [Artificial bee colony algorithm of evolutionary optimization], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2009, No. 12 (101), pp. 41-46.
25. *Kuliev E.V., Lezhebokov A.A., Dukkardt A.N.* Podkhod k issledovaniyu okrestnostey v roevykh algoritmakh dlya resheniya optimizatsionnykh zadach [Approach to research environs in swarms algorithm for solution of optimizing problems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 7 (156), pp. 15-25.
26. *Semenova A.V., Kureychik V.M.* Multi-objective particle swarm optimization for ontology alignment, *Proceeding of the 10th International Conference on «Application of Information and Communication Technologies»,* 2016, pp. 141-148.
27. *Bova V.V., Kureichik V.V., Zaruba D.V.* Data and knowledge classification in intelligence informational systems by the evolutionary method, *Proceeding of the 6th International Conference «Cloud System and Big Data Engineering (Confluence)»,* 2016, pp. 6-11.
28. *Mizzaro S., Robertson S.* HITS hits TREC - exploring IR evaluation results with network analysis, *SIGIR 2007. ACM, 2007,* pp. 479-486.

Статью рекомендовала к опубликованию д.т.н., профессор Л.С. Лисицына.

Бова Виктория Викторовна – Южный федеральный университет; e-mail: vvbova@yandex.ru; 347928, г. Таганрог, Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент.

Лещанов Дмитрий Валерьевич – e-mail: leshok.dimkaa@yandex.ru; кафедра систем автоматизированного проектирования; студент.

Bova Victoria Victorovna – Southern Federal University; e-mail: vvbova@yandex.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; associate professor.

Leshchanov Dmitriy Valeryevich – e-mail: leshok.dimkaa@yandex.ru; the department of computer aided design; student.

УДК 002.53:004.89

Ю.А. Кравченко, А.Н. Нацкевич

МОДЕЛЬ РЕШЕНИЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ ДАННЫХ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ БУСТИНГА АЛГОРИТМОВ АДАПТИВНОГО ПОВЕДЕНИЯ МУРАВЬИНОЙ КОЛОНИИ И К-СРЕДНИХ*

Рассмотрена разработка модели решения задачи кластеризации. Приведена постановка задачи. Рассматриваются классические (k-means) и современные (метод ядра, метод ансамблей, аффинное распределение) алгоритмы решения задачи кластеризации, выделяются их достоинства и недостатки. Аналитический обзор методов кластеризации показывает, что для

* Работа выполнена при финансовой поддержке РФФИ (проект № 17–07–00446).

большинства из них отсутствует программная реализация из-за сложности решения проблемы реализации полного перебора объектов обучающей выборки. В дальнейшем целесообразно применять такую модель системы кластеризации, в которой проблема полного перебора объектов обучающей выборки снимается, так как он осуществляется лишь один раз при формировании обобщенных образов классов. Данный подход, реализованный на принципах эволюционных вычислений, позволит увеличивать размерность обучающей выборки до тех пор, пока не будет достигнуто требуемое высокое качество кластеризации. Более того, необходимо учесть, что сложность математической модели экспоненциально увеличивает трудоемкость программной реализации системы и в такой же степени уменьшает вероятность того, что эта система будет практически работать. Таким образом, на рынке можно будет реализовать только такие программные системы, в основе которых лежат достаточно простые и «прозрачные» математические модели. Поэтому разработчик, заинтересованный в тиражировании своего программного продукта, подходит к вопросу о выборе математической модели с учетом возможностей программной реализации. Модель должна быть как можно более простой, а значит реализоваться с меньшими затратами и более качественно. В качестве примера решения задачи кластеризации данных представляется новая модель решения задач оптимизации, базирующаяся на использовании ориентированного двудольного графа и бустинга алгоритмов моделирования поведения колонии муравьев и классического алгоритма k-means. Предложен новый механизм решения задачи кластеризации. Эвристика алгоритма моделирования поведения колонии муравьев основана на двух техниках: базовой и итеративной. Базовый метод представляет собой алгоритм движения одного конкретного муравья по графу поиска решений. Итеративная техника предполагает последовательное построение решения каждым отдельным агентом колонии, последующая оценка решения и поиск лучшего полученного решения. Алгоритм k-meansреализует механизм итеративного поиска решения посредством использования средних точек класса (центроидов). Использование бустинга позволяет решить некоторые проблемы классических алгоритмов, такие, как начальный выбор параметра kдля алгоритма k-meansи проблему выбора начальной позиции центроидов. Проведенные исследования показали, что решения, полученные при помощи использования подхода бустинга алгоритмов, позволяют получать решения, не уступающие или превосходящие по качеству решения, полученные современными алгоритмами.

Кластеризация; эволюционное моделирование; роевые алгоритмы; алгоритм роя муравьев; бустинг; k-means.

Yu.A. Kravchenko, A.N. Natskevich

THE MODEL FOR DATA CLUSTERIZATION PROBLEM SOLUTION BASED ON BUSTING OF ANT COLONY AND K-MEANS ALGORITHMS

The article presents new model for solving clustering problem. Problem definition is presented. Article describes some classic (k-means) and modern (kernel method, ensembles method, affinity propagation) algorithms for solving clustering problem. Overview of the research methods shows that most of them do not have a software implementation due to solving the problems of Brute-force of the sample objects. It is recommended to apply a model of the clustering system in which the objects of the training sample are completely processed only once at the step of creating initial class structure. This approach is based on the principles of evolutionary computation and allows increasing the dimension of the training sample until the required high quality of clustering is received. Moreover, the complexity of the mathematical model exponentially increases the complexity of the software implementation. Also the model complexity reduces the probability that this system will practically work. At the market can be implemented only systems, based on simple mathematical models. In this case, developer who interested in replicating his software product, creates a mathematical model, taking into account the possibilities of software implementation. The model should be as simple as possible, and implemented with lower costs and more qualitatively. The model proposed in this article is based on oriented bipartite graph and algorithms busting (for ant colony optimization algorithm and classic k-means algorithm). New approach for solving clustering problem is described. Ant colony optimization algorithm heuristic based on two techniques: common technique and iterative. Common technique is based on the single ant algorithm (for searching the graph path). Iterative technique involves the sequential construction of a solution by each individual colony agent,

the subsequent evaluation of the solution and the search for the best solution is obtained. K-means algorithm realized solution search by using the averages class points (centroids). The use of boosting allows solving some problems of classical algorithms, such as the initial choice of the parameter k for the k-means algorithm and the problem of choosing the initial position of the centroids.The conducted researches showed that the solutions obtained with the use of the algorithm boosting approach allow obtaining solutions with identical or more increased quality to the solutions obtained by modern algorithms.

Clustering problem; evolutionary modeling; swarm algorithms; ant colony optimization; k-means.

Введение. Одним из достаточно распространенных методов анализа данных является кластеризация. Изначально имеется некоторое число объектов и число кластеров. Число кластеров может задаваться заранее или определяться алгоритмом. Любой объект может быть отнесен к любому кластеру. Основная цель кластеризации – разделение данных на группы (кластеры), состоящие из наиболее идентичных элементов. Разбиение выборки на группы схожих объектов позволяет упростить дальнейшую обработку данных и принятие решений, применяя к каждому кластеру свой метод анализа.

Поскольку задача кластеризации данных относится к классу *NP*-полных задач, создание эффективных методов решения этой задачи является актуальной проблемой. Для решения данной задачи было разработано большое количество алгоритмов, которые отличаются друг от друга сложностью, временными затратами и эксплуатационными свойствами.

Как показано в [1], на текущий момент алгоритмы кластеризации подразделяются на традиционные алгоритмы и современные.

Современные алгоритмы кластеризации подразделяются еще на несколько групп, рассмотрим некоторые из них:

1. Методы, основанные на методе ядра (kernel method) (класс алгоритмов, самый популярный представитель – метод опорных векторов). Детальная информация об этих алгоритмах приведена в [2, 3]. Одни из наиболее часто используемых методов – Kernel k-means [4] и Approximate kernel k-means [5]. Основная идея этого алгоритма - перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Идея алгоритмов классификации, базирующихся на ярде – использование метода опорных векторов. Такой подход позволяет частично уменьшить влияние шума на результат кластеризации, но, как показывают эксперименты [1, 4, 5], время исполнения алгоритма по сравнению с традиционным *k*-means существенно повышается, что делает алгоритм менее применимым к данным большой размерности.

2. Алгоритмы, основанные на использовании Ансамблей. Одни из представителей – алгоритмы, базирующиеся на использовании генетического подхода [6] и алгоритмы, базирующиеся на применении теории нечетких множеств [7]. Основная идея данных алгоритмов заключается в генерации набора исходных результатов кластеризации по определенному методу. Итоговый результат кластеризации получается путем интеграции исходных результатов кластеризации различными алгоритмами. Преимущество такого подхода заключается в возможности распараллеливания используемых алгоритмов. Среди минусов можно выделить недостаточное понимание разницы между первичными результатами кластеризации. Также можно отметить сложность разработки общей целевой функции (concensus function) [1].

Как показывают эксперименты, не все современные эвристические методы способны дать оптимальное решение задачи кластеризации при приемлемых временных затратах. Таким образом, проблема разработки алгоритма, сочетающего в себе полиномиальную сложность решения проблемы при приемлемых временных затратах является актуальной.

В последние годы интенсивно разрабатываются методы, основанные на процессах, происходящих в биологической природной среде, многие из которых основываются на моделировании коллективного интеллекта, используя при этом понятие стигмергии (непрямое взаимодействие агентов путем преобразования агентами окружающей среды). Также стоит учитывать, что некоторые вероятностные методы способны давать оптимальные решения задачи кластеризации при средних временных затратах. Например, классическая версия алгоритм *k*-means.

Также активно развивается такое направление, как бустинг [8]. Бустинг – это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов.

В данной работе демонстрируется метод решения задачи кластеризации, основывающийся на последовательном примере алгоритма моделирования поведения колонии муравьев и алгоритма *k*-means, использующегося для последующей обработки результатов и построения наиболее точного решения поставленной задачи. Алгоритм моделирования поведения колонии муравьев использует в качестве пространства поиска решений двудольный граф, что позволяет более просто и точно производить поиск решения и его оценку [9, 10–13].

Особенность используемого подхода заключается в объединении биоинспирированного подхода с вероятностными алгоритмами решения задачи кластеризации, что позволяет использовать сильные стороны алгоритма моделирования поведения колонии муравьев (полиномиальное время работы и стигмергия) [15–17], при этом делая его решение более точным с помощью последующего использования алгоритма *k*-means. Также, подобный подход позволяет решить сразу несколько проблем алгоритма *k*-means, а именно проблему начального выбора параметра *k* (т.к. используются результаты кластеризации, полученные с помощью алгоритма моделирования поведения колонии муравьев) и проблему выбора начальной позиции центроидов.

Для каждого алгоритма используется различный ряд критериев. Основные критерии, использующиеся в процессе выполнения алгоритма моделирования поведения колонии муравьев, – дальность нахождения текущего рассматриваемого объекта от выбранного центроида кластера и количество феромона на ребре графа. Таким образом, при распределении учитываются особенности задачи кластеризации и параметры, обусловленные стигмергией.

Основной критерий, использующийся в алгоритме *k*-means – среднее межкластерное расстояние. Этот критерий является наиболее часто применимым при оценки полученного решения задачи кластеризации.

1. Постановка задачи кластеризации. Пусть $X=\{x_i \mid i=1,2,\ldots,n\}$ – множество объектов, каждый объект описывается множеством атрибутов (признаков конкретного объекта) $A=\{a_j \mid j=1,2,\ldots,m\}$. $Y=\{y_l \mid l=1,2,\ldots,k\}$ – множество кластеров, по которым необходимо распределить объекты. Каждый кластер содержит центроид $c_l \in C$, описывающий средние параметры множества объектов, входящих в данный кластер. Задана функция расстояния между объектами $P(x_i, x_j)$. Требуется разбить множество объектов на непересекающиеся подмножества так, чтобы каждый кластер состоял из объектов, близких по метрике p. В процессе решения каждому объекту приписывается номер кластера l.

Алгоритм кластеризации – это функция $a: X \to Y$, которая любому объекту $x_i \in X$ ставит в соответствие номер кластера l. Количество кластеров при этом может быть известно заранее или определяться в процессе работы алгоритма.

В качестве метрики p была выбрана Евклидова метрика. Общая формула для подсчета расстояния между двумя объектами выглядит следующим образом:

$$d = \sqrt{\sum_{k=1}^{m}(x_i^k - x_j^k)^2}.$$

Для определения значений объектов используется шкалирование по методу Минимакс. Используется шкала [0, 1]. Общая формула для шкалирования выглядит следующим образом:

$$a_i' = \frac{(a_i - min(a_i))}{(max(a_i) - min(a_i))},$$

где a_i' – новое нормализованное значение *i* атрибута объекта, a_i – ненормализованное значение атрибута объекта,$\min a_i$ – минимальное значение *i* атрибута, $\max a_i$ – максимальное значение атрибутаa.

Рассмотрим небольшой пример. Пусть максимальное значение атрибута a_i=2.74, минимальное значение атрибута a_i=1.15. Тогда получим значение атрибута – 1.98. Применяя формулу, описанную выше, получаем нормализованное значение a_i'=0.522.

Для определения средних значений атрибутов объектов, входящих в состав конкретного кластера, используются центроиды. Центроид представляет собой вектор, состоящий из средних значений атрибутов всех объектов, входящих в каждый конкретный кластер.

Формула для определения центроида конкретного *j* кластера выглядит следующим образом:

$$c_j' = \frac{1}{S_j}\sum_{x_i \in S_j}\sum_{a^k \in x} a_i^k + c_j^k,$$

где $x^k{}_i$ – *k* атрибут *i* объекта *x*, $x^k{}_j$–*k* атрибут *j* объекта *x*.

Решением задачи кластеризации является множество $V'=\{Y'_j|l=1,2,\ldots,k\}$. Запланированным вариантом решения V' является разбиение множества объектов по множеству кластеров.

В качестве оценки решения V' рассматривается целевая функция, имеющая следующий вид:

$$F = \frac{P^o}{P^i} \to max,$$

где P^o – среднее межкластерное расстояние; P^i – среднее внутрикластерное расстояние.

Среднее внутрикластерное показывает среднее расстояние между объектами конкретного класса и определяется по следующей формуле:

$$F = \frac{1}{S_j}\sum_{i=1}^{n} p\left(x_i, c_j\right) \to min,$$

где *p* – расстояние, которое вычисляется по формуле выбранной метрики; $x \in X$ – текущий элемент; $c \in C$ – центроид данного кластера; *n*– количество элементов в конкретном *j* кластере.

Подсчет внутрекластерного расстояния для всех кластеров имеет вид:

$$P^i = \frac{1}{X}\sum_{j=1}^{n}\sum_{i=1}^{n} p\left(x_i, c_j\right) \to min,$$

где *p* – расстояние, которое вычисляется по формуле выбранной метрики; $x \in X$ – текущий элемент; $c \in C$ – центроид данного кластера; *k*– общее количество элементов; *l* – количество элементов в конкретном *j* кластере.

Среднее межкластерное расстояние описывает расстояние между центроидами всех классов и определяется по следующей формуле:

$$P^o = \frac{1}{U}\sum_{u \in U} p(u_i, u) \to max,$$

где *p* – расстояние с учетом выбранной метрики; u_i – рассматриваемый центроид; u – центроид, относительно которого вычисляется среднее межкластерное расстояние; n – общее количество кластеров.

2. Бустинг алгоритмов моделирования поведения колонии муравьев и классического алгоритма k-средних. В данной работе для решения задачи кластеризации используется бустинг алгоритмов моделирования поведения муравьиной колонии и классического алгоритма *k*-средних. Алгоритм моделирования поведения колонии муравьев основывается на комбинации двух техник: базовой и итеративной. Основу базовой техники составляет конструктивный алгоритм построения муравьем некоторой конкретной интерпретации решения базовой задачи. Итеративный метод заключается в реализации итеративной процедуры поиска лучшего решения.

Поиск решений осуществляется в ориентированном двудольном графе (рис. 1) на основе итерационного выполнения базовой техники алгоритма поиска лучшего решения. Работа поисковой процедуры начинается с построения в соответствии со спецификой решаемой задачи графа поиска решений.

Граф поиска решений представлен в виде выражения $H^l=(E \cup W, U^l)$, где $E=\{e_i|i=1,2,...,n\}$ – первая доля, описывающая множество объектов для кластеризации, а $W=\{w_j|j=1,2,...,m\}$ – вторая доля, описывающая множество кластеров. Ребро U_{ij} связывает вершину $e_i \in E$ с $W_j \in W$.

$U^l = \{u_j|j=1,2, ..., m\}$ – множество ребер, связывающих вершины множества E с вершинами множества W, l – номер агента, получившего текущее решение. Ребро указывает на возможность принадлежности текущего объекта кластеру, которое в дальнейшем учитывается при работе алгоритма *k*-means. Для наглядности представления конкретного решения V^l сгруппируем множество заданий W в подмножества.

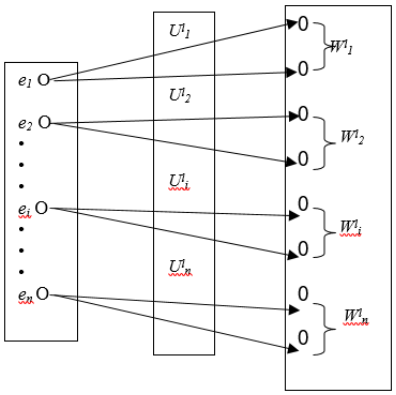

Рис. 1. Графовая модель поиска решений

В работе поиск решения V^l сводится к поиску на полном двудольном графе H_{nm} такого решения H^l, для которого оценка F^l имеет минимальное значение.

Первоначально, для каждого объекта выбираются объекты, атрибуты которых считаются начальными атрибутами центроидов каждого класса. Объекты выбираются таким образом, чтобы согласно метрики p они были наиболее далекими друг от друга. Сами объекты на этапе определения центроида не распределяются.

На каждой итерации l выполняется три этапа. На первом этапе каждый муравей строит решение (разбиение на кластеры), на втором этапе откладывается феромон, на третьем этапе осуществляется испарение феромона. Рассмотрим каждый из этих этапов более подробно.

Как было сказано выше, на первом этапе осуществляется последовательное построение решения каждым агентом. За каждым агентом закрепляется стартовая вершина. Стартовая вершина может выбираться тремя различными способами:

1. В качестве стартовой вершины может выбираться начальная вершина $w_1 \epsilon W$.
2. Выбор стартовой вершины может инкрементироваться согласно итерации $w_i \epsilon W$.
3. Стартовая вершина может выбираться случайным образом.

Далее каждый агент осуществляет последовательное случайное распределение вершин в кластеры. При случайном распределении элементов формируется начальный список вершин $X(t)$, которые могут быть распределены в один из кластеров. На первом шаге $X(0) = X$. Также определяется список табу, содержащий элементы уже распределенные в один из кластеров. На первом шаге выполняется условие $X'(0)=\emptyset$. Для этого списка выполняется условие $X(0) \notin X'(0)$ Объекты, находящиеся в этом списке в процессе построения решения не могут быть распределены снова. После распределения элементы x_i выполняются следующие условия: $xi \notin X(l)$; $X(l) \neq X$; $xi \epsilon X'(l)$; $X'(l) \neq \emptyset$. Распределение продолжается до тех пор, пока $X(l)$ не станет пустым. На последнем шаге выполняются условия: $X'(l)=X$; $X(l)=\emptyset$.

Моделирование поведения муравьёв связано с анализом количества феромона на ребрах графа H_{nm} и близости к центроиду кластера. На начальном этапе на всех ребрах U графа H_{nm} откладывается одинаковое (небольшое) количество феромона Q/v, где $v=|U|$. Параметр Q задается априорно. Формируется список объектов. Небольшое изначальное количество феромона задается с целью ограничения вероятности попадания алгоритма в локальный оптимум. Будем обозначать граф H_{nm} после отложения на нем на итерации t феромона, как $H_{nm}(t)$. После начального отложения – $H_{nm}(0)$. На первом этапе каждый муравей осуществляет распределение объектов по кластерам. Для этого муравей просматривает последовательно множество объектов и на каждом шаге для рассматриваемого объекта случайным образом выбирается кластер. Вероятность распределения объекта в кластер, базируется на подсчете веса каждого ребра. Вес конкретного ребра определяется по следующей формуле:

$$f_{i,j} = \left(\alpha d\left(x_i, y_j\right) + \beta \tau_{i,j}\right),$$

где α – коэффициент, определяющий вес критерия расстояния конкретного объекта от центроида кластера при распределении; p – расстояние с учетом используемой метрики; $x \in X$– текущий i объект для распределения; $c \in C$ – центроид j кластера; β – коэффициент, определяющий вес критерия количества отложенного феромона; $\tau_{i,j}$ – количество феромона отложенного на ребре $u_{i,j}$.

Формула для определения вероятности $p_{i,j}$ распределения объекта x_i в кластер $y_j \in Y$ выглядит следующим образом:

$$P_{i,j} = \frac{(\alpha d(x_i, y_j) + \beta \tau_{i,j})}{\sum_{j=1}^{n} (\alpha d(x_i, y_j) + \beta \tau_{i,j})}.$$

В разработанном алгоритме используется евклидова метрика. В связи с этим формула для определения расстояния от объекта x_i до центроида c_j записывается следующим образом:

$$d = \sqrt{\sum_{k=1}^{m} (a_i^k - c_j^k)^2},$$

где $a \in A$ – атрибут объекта, который необходимо распределить; $c \in C$ – центроид класса; n – количество атрибутов у объекта.

Таким образом, последовательно, для каждого элемента, муравей определяет кластер, в который необходимо распределить каждый конкретный объект. Каждый l-й агент формирует на ребрах графа $H_{nm}(t-1)$ свой собственный граф – решение $H^l(t)$, определяется решение $V^l(t)$, соответствующее графу – решению $H^l(t)$, и оценка решения $F^l(t)$.

После того, как каждый l агент построил свое решение, происходит подсчет атрибутов центроидов для каждого кластера.

Формула формирования центроидов кластеров:

$$c_j' = \frac{1}{S_j}\sum_{x_i \in S_j} x_i,$$

где $c' \in C$ – новое значение центроида j кластера; x_i – объект, входящий в состав кластера S, при этом $x \in S$, n – количество объектов в j кластере.

После подсчета атрибутов центроидов происходит оценка решения, полученного каждым агентом. Для оценки решения используется подсчет среднего внутрекластерного расстояния между объектами, входящими в состав конкретного кластера и межкластерного расстояния.

На втором этапе итерации t, каждый агент откладывает феромон на рёбрах графа $H_{nm}(t\text{-}1)$, соответствующих ребрам построенного граф – решения $H^l(t)$, в количестве пропорциональном функции качества $F^l(t)$. Чем меньше $F^l(t)$, тем больше феромона откладывается на ребрах построенного граф – решения $H^l(t)$.

Формула, определяющая количество феромона, которое должен отложить каждый конкретный муравей, выглядит следующим образом:

$$q_{i,j} = \frac{q_{i,j} * F^b}{F^c},$$

где q – количество феромона, которое должен отложить муравей; F^b – лучшее найденное решение; F^c – текущее полученное муравьем решение.

После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе итерации t происходит общее испарение феромона на ребрах двудольного графа $H_{nm}(t)$. Испарение проходит на всех ребрах множества U, при этом коэффициент испарения A задается априорно.

Формула для определения количества оставшегося после испарения феромомна на конкретном ребре имеет вид:

$$\tau_{i,j} = \tau_{i,j} * (1 - A),$$

где A – коэффициент испарения; τ – количество феромона на i ребре.

Данный процесс происходит определенное количество итераций, после чего происходит смена алгоритмов и начинает работу классический алгоритм k-средних.

На вход алгоритма поступает результат работы алгоритма моделирования поведения колонии муравьев, представляющий собой двудольный граф H_{nm}, содержащий информацию о лучшем найденном решении. Количество кластеров k задается в соответствии с этим решением $k = Y$. Также наследуется информация о полученных атрибутах центроидов. Общая формула работы алгоритма k-средних выглядит следующим образом.

Для каждого конкретного объекта $x \in X$ происходит определение ближайшего центроида по следующей формуле:

$$f_{i,j} = \sum_{y_j \in Y} d(x_i, y_j) \text{-> min},$$

где x_i – текущий распределяемый объект; y_j – расматриваемый кластер; $d(x_i, y_j)$ – функция, определяющая расстояние между объектом и центроидом согласно используемой метрике.

После чего происходит перерасчет центроидов в соответствии с объектами, входящими в его состав. Перерасчет значений центроидов распределяется по следующей формуле:

$$c_j' = \frac{1}{S_j}\sum_{x_i \in S_j} x_i.$$

Процесс происходит только некоторое (небольшое), количество итерации с целью улучшения полученного решения. Процесс продолжается в случае, если $P^i_c > P^i_b$. Где P^i_c – оценка решения, полученная алгоритмом *k*-средних, P^i_b оценка лучшего решения, полученного с помощью алгоритма моделирования поведения колонии муравьев. В случае, если $P^i_c \leq P^i_b$, работа алгоритма прекращается.

4. Экспериментальные исследования. Основной цель проведения экспериментальных исследований является проверка эффективности работы модели бустинга алгоритма муравьиной колонии и классического алгоритма *k*-средних для решения задачи кластеризации данных. Для этих целей была использована процедура проверки алгоритма с использованием контрольных примеров с заранее известным оптимумом. Первой целью являлось исследование влияния управляющих операторов, таких как: размер популяции муравьев; количество итераций и параметров, управляющих отложением и испарением феромона. Для определения однозначной достоверной оценки работы модели была проведена серия экспериментов.

Временная сложность алгоритма моделирования поведения колонии муравьев зависит от времени жизни колонии t_1 (число итераций), количества исполнителей n и количества элементов для кластеризации m, и определяется как $O(t*n*2*m)$. Временная сложность алгоритма *k*-средних зависит от количества итераций t_2 и количества элементов для кластеризации. Временная сложность алгоритма *k*-средних определяется как $O(t_2*m)$. Модель бустинга предполагает последовательное использование алгоритмов, следовательно, общая алгоритмическая сложность определяется, как $O(t_1*n*2*m+t_2*m)$. Эксперименты показали, что в 98 % случаев пространство синтезированных решений включает глобальное оптимальное решение.

В качестве используемых параметров алгоритма моделирования поведения колонии муравьев были использованы следующие: $t_1 = 100$ – количество итераций, $n = 50$ – размерность колонии. Размер данных для кластеризации использовался в соответствии с базой объектов для кластеризации. Количество итераций для алгоритма *k*-means = 50. Такое небольшое число использовалось исходя из цели уточнения решения, полученного алгоритмом моделирования поведения колонии муравьев.

Для сравнительного анализа было выбрано несколько алгоритмов. Рассмотрим последовательно каждый отдельный из них.

Один из алгоритмов – Approximate kernel k-means (AKKM) [5]. Это один из методов, основанных на соединении метода ядер с алгоритмом *k*-means. Алгоритм работает в два этапа. Первый этап – вычисление ядра (kernel computation), второй этап – кластеризация. Особенность алгоритма заключается в использовании модифицированной матрицы ядер [5, 18–20]. Авторы предлагают разбивать матрицу на отдельные подматрицы для облегчения вычислительного процесса определения значений ядер. Вычислительная сложность базируется на анализе этапа построения матрицы ядер и на оценки сложности кластеризации и вычисляется как $O(m^3+m^2n+mnCl)$, где m – количество элементов для построения первичной матрицы ядер (m<n)) и n – количество элементов для кластеризации, содержащиеся в наборе данных, т.е. числа, определяющие размерность матрицы, C – количество кластеров, на которые необходимо разбить элементы l – количество итераций. Приведем табл. 1, показывающую сравнительные результаты работы алгоритмов.

Таблица 1

Сравнительные результаты работы алгоритмов по времени

Размерность базы данных	Время вычисления ядра для AKKM	Время кластеризации для AKKM	Время кластеризации для разработанной модели бустинга
100	1.40	17.70	17.30
200	1.64	22.57	21.64
500	3.82	28.56	26.48
1000	11.14	55.01	52.05
2000	22.80	134.68	131.86
5000	64.11	333.31	329.34

Вторым алгоритмом является алгоритм взвешенного согласования кластеров с помощью ядер [21]. Алгоритм основан на использовании метода ансамблей и базируется на нескольких шагах:

Шаг генерации – на этом шаге происходит генерация решений с помощью использования любого алгоритма кластеризации.

На втором шаге происходит определение наиболее общего набора данных с помощью использования функции ядра. Для нахождения функции консенсуса используется алгоритм имитации отжига.

Вычислительная сложность алгоритма – $O(n*m*rMax)$, где $rMax$ – максимально возможное число итераций, n – число объектов, которые необходимо кластеризовать, m – число кластеров.

Алгоритмы сравниваются по критерию *ici* (Incorrectly Clustered Instances). Процент неверно классифицированных объектов согласно лучшему решению для каждого набора данных представлен в табл. 2.

Таблица 2

Сравнительные результаты работы алгоритмов по качеству решения

Набор данных	Число кластеров	k-means (ici)	WPCK (ici)	Модель бустинга (ici)
Ионосфера	10	28.5	17.6	7.2
Ионосфера	20	26.3	16.5	6.4
Ирис	10	22.3	9.3	5.7
Ирис	20	18.6	7.3	4.4
Ирис	30	15.1	5.3	2.1

Заключение. В работе предлагается новая модель, базирующаяся на бустинге алгоритмов моделирования поведения колонии муравьев и алгоритме *k*-средних. Визуализация модели представлена в виде двудольного графа. Предложены новые механизмы решения задач кластеризации. В отличие от канонической парадигмы муравьиного алгоритма на графе поиска решений строится двудольный граф. Такой подход является эффективным способом поиска рациональных решений для задач оптимизации, допускающих интерпретацию решений в виде двудольных графов. Алгоритм оптимизации может быть успешно применен для решения сложных комплексных задач оптимизации.

Было проведено сравнение с другими известными алгоритмами. Результаты работы алгоритмов по времени, представленные в табл. 1, показали, что использование модели бустинга дает незначительное временное преимущество (в пределах 1%), однако, стоит учесть, что время работы по сравнению с другими алгоритмами может уменьшаться с ростом количества объектов для кластеризации, что объясняется использованием вероятностного подхода.

Сравнительный анализ качества работы алгоритмов, результаты которого приведены в табл. 2 показал, что решения, полученные с помощью использования подхода бустинга, отличаются в лучшую сторону по сравнению с аналогами, имеющими меньшую алгоритмическую сложность. Также стоит отметить уменьшение процента неверно классифицированных объектов с ростом числа кластеров.

Источником усовершенствования может стать более корректный подбор управляющих параметров и учет уже полученных решений при использовании алгоритма k-means.

Также стоит отметить тот факт, что скорость работы алгоритмов может быть увеличена путем использования параллельной парадигмы программирования [22]. В качестве метода декомпозиции можно выбрать как декомпозицию по данным, так и декомпозицию по функциональным особенностям алгоритма.

БИБЛИОГРАФИЧЕСКИЙСПИСОК

1. *Donkuan X. Yingjie T.* A comprehensive survey of clastering algorithms // Annals of Data Science. – 2015. – Vol. 2, Issue 2. – P. 165-193.
2. *Müller K., Mika S., Rätsch G.* An introduction to kernel-based learning algorithms // IEEE Trans Neural Netw. – 2001. – No. 12. – P. 181-201.
3. *Filippone M., Camastra F., Masulli F.* A survey of kernel and spectral methods for clustering // Pattern Recognit. – 2008. – Vol. 41. – P. 176-190.
4. *Schölkopf B., Smola A., Müller K.* Nonlinear component analysis as a kernel eigenvalue problem // Neural Comput. – 1998. – Vol. 10. – P. 1299-1319.
5. *Radha C., Rong J., Timothy C.H., Anil K.J.* Scalable Kernel Clustering: Approximate Kernel k-means // Computer Vision and Pattern Recognition. – 2014.
6. *Yoon H., Ahn S., Lee S., Cho S., Kim J.* Heterogeneous clustering ensemble method for combining different cluster results // In: Data mining for biomedical applications. – 2006. – P. 82-92.
7. *Punera K., Ghosh J.* Consensus-based ensembles of soft clusterings // ApplArtifIntell. – 2008. – Vol. 22. – P. 780-810.
8. Бустинг. Применение в области машинного обучения. – URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%91%D1%83%D1%81%D1%82%D0%B8%D0%BD%D0%B3 (дата обращения 28.4.2017).
9. *Лебедев Б.К., Лебедев О.Б.* Моделирование адаптивного поведения муравьиной колонии при поиске решений, интерпретируемых деревьями // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 27-34.
10. *Лебедев Б.К., Нацкевич А.Н.* Решение однородной распределительной задачи методом моделирования поведения муравьиной колонии // Информатика, вычислительная техника и инженерное образование. – 2015. − № 4 (24). – С. 7-15.
11. *Gladkov L.A., Kravchenko Y.A., Kureichik V.V.* Evolutionary Algorithm for Extremal Subsets Comprehension in Graphs // World Applied Sciences Journal. – 2013. – Vol. 27 (9). – P. 1212-1217.
12. *Курейчик В.М., Кажаров А.А.* Использование шаблонных решений в муравьиных алгоритмах // Известия ЮФУ. Технические науки. – 2013. – № 7 (144) – С. 11-17.
13. *Курейчик В.М.* Особенности построения систем поддержки принятия решений // Известия ЮФУ. Технические науки. – 2012. – № 7 (132). – С. 92-98.
14. *Курейчик В.В., Родзин С.И.* О правилах представления решений в эволюционных алгоритмах // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 13-21.
15. *Bova V.V., Kravchenko Y.A., Kureichik V.V.* Decision Support Systems for Knowledge Management // Software Engineering in Intelligent Systems. Proceedings of the 4th Computer Science On-line Conference 2015 (CSOC2015). Springer International Publishing AG Switzerland. – Vol. 3. – P. 123-130.
16. *Гладков Л.А., Курейчик В.М., Курейчик В.В.* Генетические алгоритмы. – М.: Физматлит, 2006. – 320 с.
17. *Курейчик В.В., Бова В.В., Курейчик Вл.Вл.* Комбинированный поиск при проектировании // Образовательные ресурсы и технологии. – 2014. – № 2 (5). – С. 90-94.
18. *Курейчик В.В., Курейчик Вл.Вл.* Бионический поиск при проектировании и управлении // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 178-183.

19. *Zaporozhets D.Yu., Zaruba D.V., Kureichik V.V.* Hybrid bionic algorithms for solving problems of parametric optimization // World Applied Sciences Journal. – 2013. – No. 23 (8). – P. 1032-1036.
20. *Курейчик В.В., Запорожец Д.Ю.* Роевой алгоритм в задачах оптимизации // Известия ЮФУ. Технические науки. – 2010. – № 7 (108). – С. 28-32.
21. *Sandro V.P., Jyrko C.M., José R.S.* Weighted Partition Consensus via Kernels // Pattern Recognition. – 2010. – Vol. 43(8). – P. 2712-2724.
22. *Карпов В.Е.* Введение в распараллеливание алгоритмов и программ // Компьютерные исследование и моделирование. – 2010. – Т. 2, № 3. – С. 231-272.

REFERENCES

1. *Donkuan X. Yingjie T.* A comprehensive survey of clastering algorithms, *Annals of Data Science*, 2015, Vol. 2, Issue 2, pp. 165-193.
2. *Müller K., Mika S., Rätsch G.* An introduction to kernel-based learning algorithms, *IEEE Trans Neural Netw.*, 2001, No. 12, pp. 181-201.
3. *Filippone M., Camastra F., Masulli F.* A survey of kernel and spectral methods for clustering, *Pattern Recognit*, 2008, Vol. 41, pp. 176-190.
4. *Schölkopf B., Smola A., Müller K.* Nonlinear component analysis as a kernel eigenvalue problem, *Neural Comput.*, 1998, Vol. 10, pp. 1299-1319.
5. *Radha C., Rong J., Timothy C.H., Anil K.J.* Scalable Kernel Clustering: Approximate Kernel k-means, *Computer Vision and Pattern Recognition*, 2014.
6. *Yoon H., Ahn S., Lee S., Cho S., Kim J.* Heterogeneous clustering ensemble method for combining different cluster results, *In: Data mining for biomedical applications*, 2006, pp. 82-92.
7. *Punera K., Ghosh J.* Consensus-based ensembles of soft clusterings, *ApplArtifIntell*, 2008, Vol. 22, pp. 780-810.
8. Busting. Primenenie v oblasti mashinnogo obucheniya [Boosting. Application in the field of machine learning]. Available at: http://www.machinelearning.ru/wiki/index.php?title=%D0%91%D1%83%D1%81%D1%82%D0%B8%D0%BD%D0%B3 (accessed 28 April 2017).
9. *Lebedev B.K., Lebedev O.B.* Modelirovanie adaptivnogo povedeniya murav'inoy kolonii pri poiske resheniy, interpretiruemykh derev'yami [Modelling of an ant colony adaptive behaviour by search of the decisions interpreted by trees], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 27-34.
10. *Lebedev B.K., Natskevich A.N.* Reshenie odnorodnoy raspredelitel'noy zadachi metodom modelirovaniya povedeniya murav'inoy kolonii [The solution of the homogeneous distribution of tasks by modeling the behavior of ant colonies], *Informatika, vychislitel'naya tekhnika i inzhenernoe obrazovanie* [Informatics, computer science and engineering education], 2015, No. 4 (24), pp. 7-15.
11. *Gladkov L.A., Kravchenko Y.A., Kureichik V.V.* Evolutionary Algorithm for Extremal Subsets Comprehension in Graphs, *World Applied Sciences Journal*, 2013, Vol. 27 (9), pp. 1212-1217.
12. *Kureychik V.M., Kazharov A.A.* Ispol'zovanie shablonnykh resheniy v murav'inykh algoritmakh [Template using for ant colony algorithms], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2013, No. 7 (144), pp. 11-17.
13. *Kureychik V.M.* Osobennosti postroeniya sistem podderzhki prinyatiya resheniy [Features of decision making support system design], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 7 (132), pp. 92-98.
14. *Kureychik V.V., Rodzin S.I.* O pravilakh predstavleniya resheniy v evolyutsionnykh algoritmakh [On the rules for the submission decisions in evolutionary algorithm], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 7 (108), pp. 13-21.
15. *Bova V.V., Kravchenko Y.A., Kureichik V.V.* Decision Support Systems for Knowledge Management, *Software Engineering in Intelligent Systems. Proceedings of the 4th Computer Science On-line Conference 2015 (CSOC2015).* Springer International Publishing AG Switzerland, Vol. 3, pp. 123-130.
16. *Gladkov L.A., Kureychik V.M., Kureychik V.V.* Geneticheskie algoritmy [Genetic algorithms]. Moscow: Fizmatlit, 2006, 320 p.
17. *Kureychik V.V., Bova V.V., Kureychik Vl.Vl.* Kombinirovannyy poisk pri proektirovanii [Combined search in the design], *Obrazovatel'nye resursy i tekhnologii* [Educational resources and technology], 2014, No. 2 (5), pp. 90-94.

18. *Kureychik V.V., Kureychik Vl.Vl.* Bionicheskiy poisk pri proektirovanii i upravlenii [Search inspired by natural systems, for the design and management], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 178-183.
19. *Zaporozhets D.Yu., Zaruba D.V., Kureichik V.V.* Hybrid bionic algorithms for solving prob-lems of parametric optimization, *World Applied Sciences Journal*, 2013, No. 23 (8), pp. 1032-1036.
20. *Kureychik V.V., Zaporozhets D.Yu.* Roevoy algoritm v zadachakh optimizatsii [Swarm algorithm in optimisation promlems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 7 (108), pp. 28-32.
21. *Sandro V.P., Jyrko C.M., José R.S.* Weighted Partition Consensus via Kernels, *Pattern Recognition*, 2010, Vol. 43(8), pp. 2712-2724.
22. *Karpov V.E.* Vvedenie v rasparallelivanie algoritmov i programm [Introduction to parallel algorithms and software], *Komp'yuternye issledovanie i modelirovanie* [Computer research and modeling], 2010, Vol. 2, No. 3, pp. 231-272.

Статью рекомендовал к опубликованию д.т.н., профессор А.Г. Коробейников.

Кравченко Юрий Алексеевич – Южный федеральный университет; e-mail: yakravchenko@sfedu.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371651; кафедра систем автоматизированного проектирования; доцент.

Нацкевич Александр Николаевич – e-mail: natskevich.a.n@gmail.com; кафедра систем автоматизированного проектирования; аспирант.

Kravchenko Yury Alekseevich – Southern Federal University; e-mail: yakravchenko@sfedu.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371651; the department of computer aided design; associate professor.

Natskevich Alexander Nikolaevich – e-mail: natskevich.a.n@gmail.com; the department of computer aided design; graduate student.

УДК 004.912

В.С. Корнилов, В.М. Глушань, А.Ю. Лозовой

ОЦЕНКА КАЧЕСТВА МАШИННОГО ПЕРЕВОДА ТЕКСТА С ИСПОЛЬЗОВАНИЕМ МЕТОДА АНАЛИЗА НЕЧЕТКИХ ДУБЛИКАТОВ*

Статья посвящена разработке способов оценки качества и улучшения результатов машинного перевода. Машинный перевод рассматривается как полностью автоматический перевод текста на основе правил. Существующие методы и средства машинного перевода имеют как преимущества так и недостатки, заключающиеся в потере семантической целостности при переводе одного и того же текста с одного естественного языка на другой естественный язык, в итоге результат перевода в большинстве случаев некорректен. Целью работы является создание системы автоматической корректировки машинного перевода, результатом работы которой будет текст на уровне публикации. Научная новизна заключается в использовании процедуры получения обратного перевода, его сравнения с оригинальным текстом для численной оценки качества машинного перевода, а также поиска несоответствий с применением системы выявления нечетких дубликатов в параллельных корпусах и последующей их корректировки. В настоящее время существует широкий набор методов и приложений для оценки машинного и автоматизированного переводов с использованием обработки параллельных корпусов. Недостатком данных методов является невозможность отследить ошибки в конкретном параллельном корпусе. Существует широкий набор методов для анализа совпадений в неструктурированном тексте, применяемых для поиска плагиата в различных сферах. Классические методы анализа текста подразделяются на синтаксические методы анализа последовательностей, со-

* Работа выполнена при финансовой поддержке РФФИ (проект № 15–01–05669).