УДК 002.53:004.89 DOI 10.23683/2311-3103-2018-5-120-131

Ю.А. Кравченко, А.Н. Нацкевич, И.О. Курситыс

МОДЕЛЬ БУСТИНГА БИОИНСПИРИРОВАННЫХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ КЛАССИФИКАЦИИ И КЛАСТЕРИЗАЦИИ*

Рассмотрены способы применения моделей бустинга для решения задач кластеризации и классификации, описаны сравнительные характеристики этих моделей. Также разработана модель бустинга для решения задачи кластеризации. Приведена постановка задачи. Представлен аналитический обзор некоторых перспективных разработок среди современных и классических алгоритмов кластеризации, оценены их достоинства и недостатки. Представлен модифицированный алгоритм бустинга для решения задачи кластеризации. Проведено сравнение подходов бустинга и бэггинга, получена оценка достоинств и недостатков рассмотренных подходов. Приведен обзор алгоритмов, использующихся в процессе бустинга. В качестве примера решения задачи кластеризации данных представляется новая модель решения задач оптимизации, базирующаяся на использовании взвешенного множества алгоритмов кластеризации и их бустинга, основанного на идеях биоинспирированных алгоритмов. Эвристика предложенного алгоритма бустинга заключается в использовании матрицы вероятностей, что позволяет проводить взвешенную оценку качества результата обучающихся алгоритмов для получения наиболее высокого качества решения задачи кластеризации, а также использовать взвешенные наборы данных, содержащие информацию о вероятности вхождения каждого отдельного элемента в определенный кластер. Проведенные исследования показали, что решения, полученные при помощи использования подхода бустинга алгоритмов, позволяют получать результаты, не уступающие или превосходящие по качеству варианты, полученные известными алгоритмами.

Бустинг; кластеризация; классификация; эволюционное моделирование; роевые алгоритмы; машинное обучение; биоинспирированные алгоритмы.

Yu.A. Kravchenko, A.N. Natskevich, I.O. Kursitys

THE MODEL OF BOOSTING BIOINSPIRED ALGORITHMS FOR SOLVING PROBLEMS OF CLASSIFICATION AND CLUSTERING

In the article methods of application of boosting models for solving clustering and classification problems are considered, comparative characteristics of these models are described. A boosting model has also been developed to solve the clustering problem. The statement of the problem is given. An analytical review of some promising developments among modern and classical clustering algorithms is presented, their advantages and disadvantages are estimated. A modified boosting algorithm for solving the clustering problem is presented. The approaches of boosting and bagging are compared, the merits and drawbacks of the approaches considered are estimated. The review of algorithms used in the process of boosting is given. As an example of solving the problem of data clustering, a new model for solving optimization problems is presented, based on the use of clustering algorithms weighted set and their boosting based on the ideas of bioinspired algorithms. The heuristic of the proposed boosting algorithm is the use of a probability matrix, which allows a weighted estimation of the learning algorithms quality result to obtain the highest quality of the solution to the clustering problem, and also use weighted data sets containing information on the probability of each individual element occurrence in a particular cluster. The conducted researches showed that the solutions obtained by using the algorithm boosting approach allow to obtain results that are not inferior or superior in quality to the variants obtained by the known algorithms.

Boosting; clustering; classification; evolutionary modeling; swarm algorithms; machine learning; bioinspired algorithms.

* Работа выполнена при поддержке РФФИ (проекты: № 17-07-00446 и 18-07-00050).

Введение. При решении многих задач научного общества и различных областей бизнеса часто возникает необходимость анализа данных. Решение данной проблемы становится достаточно затруднительным в связи с тем, что одной из наиболее сильно выраженных тенденций развития общества является постоянный рост объемов данных и их слабая структурированность [1]. В качестве примера можно привести статистику компании IBM, согласно которой минимум 2.5 эксабайта данных генерируется каждый год [2].

Данная проблема обосновывает актуальность создания новых масштабируемых алгоритмов анализа данных, которые способны выдать хорошие результаты кластеризации при условии оптимальных временных затрат. Одним их наиболее часто используемых методов анализа данных является кластеризация, что обосновывается необходимостью деления огромного количества постоянно растущего объема данных на кластеры [1] для последующего упрощения их обработки с целью выделения информации и решения различных научных проблем. Изначально имеется множество объектов, которые необходимо разбить на множество кластеров таким образом, чтобы каждая отдельная группа включала объекты, являющиеся наиболее схожими друг с другом в соответствии с используемой метрикой. При этом количество кластеров может как определяться автоматически самим алгоритмом, так и задаваться пользователем заранее. Каждый элемент входного набора данных может быть определен в любой из кластеров.

Кластеризация, рассматриваемая как самый важный и перспективный в плане изучения подход неконтролируемого обучения (обучения без учителя) [3]. Поскольку кластеризация входит в класс NP-сложных задач, использование метода полного перебора может привести к большим временным затратам, также становится слабоэффективным использование различных точных методов. Таким образом, создание методов, которые позволят получить эффективные решения при условии оптимальных временных затрат является актуальной научной проблемой.

Для решения данной задачи было разработано большое количество алгоритмов, которые отличаются друг от друга временными затратами, алгоритмической сложностью и различными особенностями эксплуатации. Также были предприняты различные попытки классификации разработанных алгоритмов.

Например, многие методы кластеризации были классифицированы такими учеными, как Donkuan, X. Yingjie T., анализ результатов исследований которых представлен в работе [4]. Они разделили имеющиеся алгоритмы кластеризации на две большие группы: классические и современные методы.

Среди современных методов одними из достаточно эффективных являются методы, использующие ансамбли алгоритмов. Например, существует ряд алгоритмов, базирующихся на использовании эволюционного моделирования [5–11]. Также часто применяют алгоритмы, использующие теорию нечетких множеств. В процессе работы алгоритмы этого класса генерируют сразу несколько решений с помощью каждого отдельного алгоритма, входящего в ансамбль. Итоговое решение задачи получается путем интеграции набора решений определенным методом. Например, методом голосования. Основное преимущество этого класса алгоритмов заключается в обеспечении возможности использования моделей распараллеливания. Основным недостатком является невозможность выработки функции консенсуса (concensus function) [4]. Также стоит отметить, что итоговая алгоритмическая сложность ансамбля может оказаться достаточно большой в случае вхождения в ансамбль сложных алгоритмов.

Среди часто применяемых методов также стоит выделить бустинг, что обусловлено его достаточно активным развитием и высоким качеством получаемых решений. Бустинг также базируется на методе ансамблей. Идея бустинга заключа-

ется в построении очереди применяемых алгоритмов. При этом каждый следующий алгоритм работает с решениями, полученными предыдущим алгоритмом, что позволяет более точно обрабатывать ошибки и получать более точные решения выбранной оптимизационной задачи [4, 12].

Например, при решении задачи кластеризации, бустинг позволяет учитывать возможность специфики организации каждого отдельного набора данных и выбирать наиболее эффективный алгоритм их обработки.

Рассмотрим концепцию бустинга и основные идеи, на которых она базируется более подробно.

1. Концепция бустинга для решения задачи классификации. Один из подходов, применяющихся при решении различных задач обучения – комбинирование различных моделей и алгоритмов. Две основные идеи, которые часто применяются при использовании данного подхода – бэггинг (bagging от bootstrap aggregating) и бустинг. Основная идея бэггинга заключается в построении множества независимых между собой моделей с дальнейшим принятием общего решения методом голосования или усреднения [13]. Этот подход используется в таких алгоритмах, как Random Trees и Random Forest. Бустинг использует подход, противоположный бэггингу. Модели применяются по очереди и при построении решения каждая последующая модель имеет доступ к результатам и ошибкам, полученным в ходе работы предыдущей модели.

Изначально бустинг был разработан для решения задачи классификации. Идея, на которой базируется бустинг заключается в том, что при решении задачи классификации применение комбинации слабых классификаторов может дать результат наиболее точный чем применение только одного классификатора. При этом считается, что слабый классификатор способен провести классификацию поступившего на вход элемента с вероятностью ошибки строго меньше (но не сильно), чем случайное принятие решений (0.5 в двоичном случае) [14].

В самом простом случае бустинг, также как и бэггинг, базируется на идее построения ансамбля. В случае решения задачи классификации подобный ансамбль представляет собой алгоритм решения определенной задачи обучения (например, классификации или кластеризации), включающий в себя несколько слабых алгоритмов решения этой задачи [14]. Алгоритмы комбинируются определенным образом с целью получения одного сильного алгоритма. Идея была представлена в [15]. В качестве обоснования рациональности подхода авторы предполагали, что может быть проще обучить ансамбль простых алгоритмов, чем обучить один крайне сложный алгоритм.

Например, вместо того, чтобы обучить одну огромную нейронную сеть, можно обучить несколько нейронных сетей меньшей размерности, после чего скомбинировать их результат решения поставленной задачи. Модель организации подобного подхода представлена на рис. 1. Из рисунка – $H_m : X \to \{-1, +1\}$ – m-й бинарный классификатор (для $m = 1, \ldots, M$), а $x \in X$ – входной элемент, который необходимо классифицировать. Основная идея подобного ансамбля заключается в комбинировании решений, полученных каждым отдельным классификатором $H_m(x)$. При этом выходное решение ансамбля представлено в виде $H(x)$ и может быть получено различными методами. Например, методом голосования.

Основная идея бустинга заключается в постоянном многократном использовании слабых алгоритмов, получая при этом последовательность работы алгоритмов, результаты которых могут быть объединены, как показано на рис. 1. При этом оценка каждого полученного решения на каждом шаге алгоритма базируется на точности решения, полученного каждым отдельным алгоритмом, что позволяет алгоритму бустинга фокусироваться на тех объектах данных, которые классифи-

цированы неправильно. Некоторые алгоритмы бустинга могут содержать отдельные критерии выбора каждого отдельного слабого алгоритма, что позволяет на выходе получить более качественное решение.

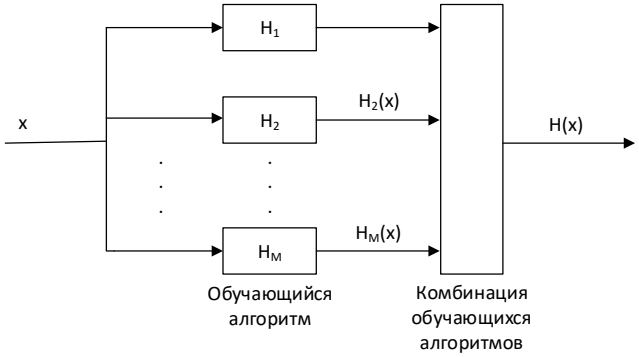

Рис. 1. Модель ансамбля классификаторов

После определенной работы над концепцией бустинга, Fraud и Schapire в нескольких своих работах [16, 17] предложили алгоритм адаптивного бустинга (AdaBoost). Основная идея данного алгоритма заключается в использовании взвешенной версии определенного набора. Такой набор данных используется многократно, что предопределяет необходимость большого размера данного набора, как было описано в алгоритмах, изложенных выше.

Сейчас данный алгоритм является хорошо изученным и часто используется для построения ансамблей классификаторов за оптимальное время. Алгоритм обучает определенный набор слабых обучающихся алгоритмов для построения слабого классификатора, используя модель, показанную на рис. 1. Слабый классификатор получается путем последовательного использования повторно взвешенного набора данных, содержащего веса, полученные основываясь на точности работы предыдущих классификаторов. При этом каждый раз используется один и тот же набор данных с экземплярами, взвешенными в соответствии с их правильной или неправильной классификацией, проведенной предыдущими используемыми классификаторами. Это позволяет слабым обучающимся алгоритмам на каждой итерации концентрировать свое внимание на тех данных, которые были на предыдущей итерации классифицированы неправильно. Также одной из задач является выбор слабого обучающегося алгоритма для получения базового классификатора таким образом, чтобы не производилось уменьшение весов объектов, правильно классифицированных на предыдущем шаге. Если базовый обучающийся алгоритм достаточно точный, он может производить классификацию, оставляя значительный вес только выбросам и шумовым экземпляром с целью более точного их изучения на следующих итерациях алгоритма. Более подробно структура алгоритма представлена на рис. 2.

Как показывают аналитические обзоры, выполненные такими учеными, как Dongkuan Xu, Yingjie Tian [4] и Ka-Chun Wong [1], многие алгоритмы бустинга, в том числе, AdaBoost, который был представлен выше, способны получить оптимальные решения при решении задачи классификации. Однако не все современные разработанные алгоритмы бустинга приспособлены для решения задач обучения без учителя. Таким образом, становится актуальной проблема разработки алгоритма, который способен получить достаточно точные решения поставленной проблемы при условии оптимальных временных затрат.

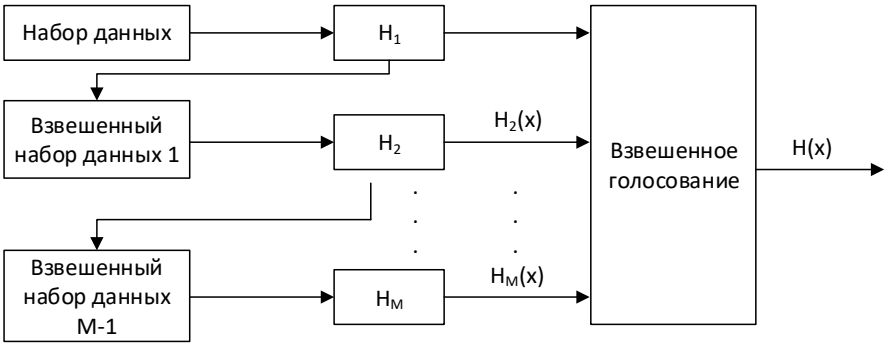

Рис. 2. Модель работы алгоритма AdaBoost

2. Сравнительная оценка процедур бустинга для решения задач классификации и кластеризации. Наиболее широкое распространение метод бустинга получил в сфере машинного обучения для решения задач обучения с учителем таких, как задача классификации. Основная идея решения подобных задач – обучение определенного алгоритма на уже классифицированных (labeled) данных с целью создания достоверных прогнозов для неклассифицированных данных [14]. Обучение с учителем – частная дисциплина, входящая в состав машинного обучения, которая также включает в себя обучение без учителя, которое базируется исключительно на анализе неклассифицированных данных.

Также отдельно на основе этих двух подходов выделяют такую отдельную область, как полу-контролируемое обучение с учителем. Эта область включает элементы из обеих дисциплин [18], поскольку подразумевается, что в набор данных входят как элементы, которые уже были классифицированы заранее, так и элементы, класс которых еще не определен.

В исследовательской работе [14] был также рассмотрен ряд алгоритмов бустинга, специализирующихся на решении этого класса проблем. Одно из решений, предлагаемых авторами – назначение элементов в псевдо-классы.

В рамках данной статьи, одним из возможных решений проблемы может быть сведение задачи к кластеризации.

В случае решения задач обучения с учителем алгоритмы, как правило, получают некую обобщающую функцию *h(.)*, которая содержит решение задачи классификации. При этом основная цель решения задачи классификации – провести категоризацию объектов в предопределенный набор классов.

Также отметим, что в самом общем случае выходные данные решения задачи классификации представлены в качестве *Y*, содержащей информацию о двух классах, которые кодированы, как *{-1, 1}*.

При этом основная задача машины – обучиться на наборе тренировочных данных $(y_1, x_1), \ldots, (y_n, x_n)$, которые уже классифицированы для последующего прогнозирования классификации новых объектов x_{new}. Где $x_1, \ldots, x_n$ – прогнозы классификации, а n – количество экземпляров набора тренировочных данных. В данном случае основной задачей алгоритма является разработка такого правила прогнозирования *h(.)*, при котором будет осуществляться максимально корректная классификация поступающих на вход данных.

$$(y_1, x_1), \ldots, (y_n, x_n) - (superviced\ learning) \rightarrow h(x_{new}) = y_{new}. \qquad (1)$$

В случае решения задач обучения без учителя, набор тренировочных данных отсутствует, а в качестве некой обобщающей функции выступает целевая функция оценки качества полученного решения. При этом основная задача машины – разработать изначальное правило распределения множества объектов на кластеры. Формула выглядит следующим образом:

$$(x_1, x_n) - (unsuperviced\ learning) \rightarrow (y_1, x_1), \dots, (y_n, x_n). \quad (2)$$

В случае решения задачи кластеризации может оцениваться среднее внутрикластерное расстояние или среднее межкластерное расстояние.

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

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

$$F = \frac{P^o}{P^i} \rightarrow max, \quad (3)$$

где P_o – среднее межкластерное расстояние, P_i – среднее внутрикластерное расстояние.

Рассмотрим подробнее механизм организации бустинга с помощью использования биоинспирированного алгоритма.

При этом формула для подсчета внутрикластерного расстояния имеет следующий вид:

$$P^i = \frac{1}{X}\sum_{j=1}^{n}\sum_{i=1}^{n} p\left(x_i, c_j\right) \rightarrow min, \quad (4)$$

где p – расстояние между объектами с учетом метрики, $x \in X$ – обрабатываемый элемент, $c \in C$ – центроид кластера, l – количество элементов в конкретном j кластере.

Среднее межкластерное расстояние описывает расстояние между объектами, входящими в состав различных кластеров и определяется по следующей формуле:

$$P^o = \frac{1}{U}\sum_{u \in U} p(u_i, u) \rightarrow max, \quad (5)$$

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

Приведенные данные позволяют делать вывод о том, что процедуры бустинга могут быть использованы одинаково эффективно как для решения задач классификации, так и для решения задач кластеризации. Также существует ряд исследований [12], показывающий высокую эффективность применения алгоритмов бустинга для решения задач полу-контролируемого обучения.

3. Бустинг биоинспирированных алгоритмов для решения задачи кластеризации. В данной работе для решения задачи кластеризации используется модель бустинга биоинспирированных алгоритмов. Алгоритм основывается на алгоритме AdaBoost, который был описан выше. Основная идея разработанного алгоритма заключается в использовании взвешенной версии определенного набора алгоритмов и множество вероятностей, определяющее вхождения каждого отдельного объекта в конкретный кластер. Данный набор алгоритмов используется многократно, что позволяет подобрать алгоритм, наиболее хорошо подходящий для кластеризации каждого конкретного набора данных. Схема работы алгоритма представлена на рис. 3.

Рассмотрим работу алгоритма более подробно. На первом шаге происходит случайный выбор алгоритма, производящего кластеризацию, после чего происходит оценка его решения в соответствии с целевой функцией.

Входные параметры. Набор данных, состоящих из множества объектов, которые необходимо кластеризовать $X=\{x_i \mid i=1,2,\ldots,n\}$, где n – общие число объектов для кластеризации. Множество алгоритмов, которые могут быть примененины для кластеризации объектов $A=\{a_j \mid j=1,2,\ldots,m\}$, где m – общее число алгоритмов. Количество итераций T.

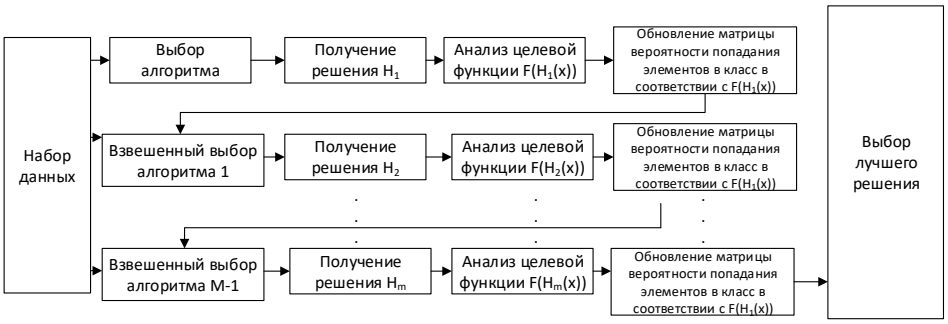

Рис. 3. Схема работы алгоритма бустинга

Стоит отметить одну из особенностей алгоритма – поскольку в наборе алгоритмов, которые могут быть применены в процессе бустинга используются биоинспирированные алгоритмы, появляется возможность использования множества, определяющего вероятность попадания каждого отдельного элемента в кластер, что может быть использовано, например, муравьиным алгоритмом.

В процессе работы алгоритма выбор каждого отдельного биоинспирированного алгоритма кластеризации осуществляется по следующей формуле:

$$f_j = \left(\alpha \frac{V_b}{V_t} + \beta u_j\right), \tag{6}$$

где V_b – результат вычисления целевой функции лучшего полученного решения, V_t – результат вычисления целевой функции текущим алгоритмом на текущей итерации, $u \in U$ – вероятность выбора i-го алгоритма из множества алгоритмов, которые могут быть использованы для кластеризации, α – коэффициент, определяющий значимость критерия улучшения показателя целевой функции, β – коэффициент, определяющий вес критерия вероятности выбора каждого конкретного алгоритма.

Также в процессе работы алгоритма при получении итогового решения каждым отдельным биоинспирированным алгоритмом производится обновление матрицы вероятностей вхождения каждого элемента набора данных в отдельный кластер для обеспечения последовательного улучшения решения каждым следующим алгоритмом. Обновление весов вероятностей происходит по следующей формуле:

$$f_{i,j} = \left(\alpha d\left(x_i, y_j\right) + \beta \tau_{i,j}\right), \tag{7}$$

где α – коэффициент, определяющий вес критерия расстояния конкретного объекта от центроида кластера при распределении, d – расстояние с учетом используемой метрики, $x \in X$ – текущий i объект для распределения, $y \in Y$ – кластер j кластера, β – коэффициент, определяющий вес критерия вероятности распределения элемента в j кластер, $\tau_{i,j}$ – порог вхождения элемента в кластер.

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

$$P_{i,j} = \frac{(\alpha d(x_i, y_j) + \beta \tau_{i,j})}{\sum_{j=1}^{n} (\alpha d(x_i, y_j) + \beta \tau_{i,j})}. \tag{8}$$

Опишем шаги алгоритма более подробно:

1. Инициализировать множества вероятностей выбора каждого отдельного алгоритма кластеризации $U=\{uj \mid j=1,2,\ldots,m\}$.
2. Пока $t < T$:
   a. Запустить один из алгоритмов кластеризации. Выбор алгоритма осуществить по формуле (6);
   b. Произвести кластеризацию элементов с помощью выбранного алгоритма, после чего оценить проведенную кластеризацию по формуле (3), внутрикластерное и межкластерное расстояние оценить по формулам (4), (5);
   c. Присвоить веса вероятности вхождения каждого отдельного элемента в каждый отдельный кластер в соответствии с результатом работы выбранного алгоритма по формуле (8);
   d. Для конкретного выбранного алгоритма обновить вероятность выбора;
   e. Инкрементировать номер итерации $t <- t+1$.
3. Закончить цикл
4. Выбрать лучшее решение в соответствии с целевой функцией.

Рассмотрим более подробно некоторые из пунктов алгоритма. В пункте (a) приведена формула для оценки качества работы определенного алгоритма кластеризации. Поскольку на первой итерации данных о предыдущей оценке целевой функции нет, данный критерий не учитывается, и формула выглядит следующим образом:

$$f_j = (\beta u_j). \quad (9)$$

Представим результаты экспериментальных исследований, доказывающих эффективность применения описанного алгоритма.

4. Экспериментальные исследования. Целью проведения исследования эффективности была проверка качества полученных решений модели бустинга биоинспирированных при решении задачи кластеризации данных. Для этого была использована процедура проверки алгоритма с использованием бэнчмарок с заранее известным оптимумом. Первой задачей было исследование влияния управляющих операторов, таких как количество итераций алгоритма бустинга и значения весов параметров в матрице вероятности выбора алгоритмов и матрице вероятности вхождения элементов в кластер. В процессе оценки модели была проведена серия экспериментов.

Алгоритмическая сложность бустинга зависит от временной сложности алгоритмов, использующихся при проведении бустинга. Изначально временная сложность бустинга – *O(n)*. Результаты проведенных экспериментальных исследований показали, что 98 % полученных алгоритмом решений содержат глобальное оптимальное решение.

В качестве используемых параметров алгоритма бустинга были использованы следующие: количество итераций самого цикла бустинга – 50; количество итераций для построения решения каждого отдельного алгоритма, входящего в ансамбль – 100. Для осуществления бустинга были использованы алгоритмы роя муравьев, стаи серых волков и алгоритм светлячков.

В процессе исследования результаты алгоритма бустинга сравнивались с результатами таких алгоритмов, как Approximate kernel k-means (AKKM) [19] и классический k-means. AKKM основан на использовании метода ядра.

Особенность алгоритма заключается в использовании модифицированной матрицы ядер [19]. Алгоритмическая сложность базируется на этапе построения матрицы ядра и оценки алгоритмической сложности кластеризации. Общая фор-

мула сложности: $O(m^3+m^2n+mnCl)$, где m (количество элементов для построения первичной матрицы ядер (m<n)) и n (количество элементов для кластеризации), C – число кластеров, l – количество итераций. Приведем табл. 1, показывающую временное сравнение алгоритмов.

Таблица 1

Временное сравнение алгоритмов

Размерность базы данных	Время вычисления ядра для AKKM	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

Кроме временных параметров алгоритмы сравнивались по критерию процентного количество неверно кластеризованных решений ici (incorrectly clustered instances). Результаты сравнения приведены в табл. 2.

Таблица 2

Сравнения качества полученных решений

Набор данных	Число кластеров	k-means (ici)	AKKM (ici)	Модель бустинга (ici)
Ионосфера	10	28.5	17.6	7.2
Ионосфера	20	26.3	16.5	5.4
Ирис	10	22.3	9.3	4.7
Ирис	20	18.6	7.3	3.4
Ирис	30	15.1	5.3	1.1

В процессе проведения экспериментальных исследований было выявлено, что алгоритм бустинга дает небольшое временное преимущество (в пределах 1%), которое может быть увеличено или уменьшено при условии изменения списка алгоритмов, входящих в ансамбль.

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, что показатель результаты работы алгоритма бустинга могут сильно варьироваться в зависимости от алгоритмов, использующихся в процессе осуществления бустинга.

Заключение. В результате проведенных исследований можно сделать вывод о том, что процедуры бустинга могут быть использованы одинаково эффективно как для решения задач классификации, так и для решения задач кластеризации.

В качестве доказательства был разработан алгоритм бустинга для решения задачи кластеризации и проведено его сравнение с аналогами по таким параметрам, как время работы и процент некорректно кластеризованных объектов данных (*ici*).

Результаты сравнения показали, что алгоритм бустинга дает незначительное преимущество по временным затратам и более значительное преимущество при оценке качества полученных в результате работы алгоритма решений.

Стоит отметить, что для улучшения качества кластеризации и уменьшения итоговых временных затрат можно произвести перебор различных алгоритмов кластеризации для осуществления бустинга. Кроме того, время работы алгоритма может быть улучшено в случае использования параллельных парадигм программирования [20–22]. Также результаты работы конкретных алгоритмов, входящих в ансамбль, могут меняться при изменении параметров модификации матрицы вероятностей вхождения элемента в конкретный кластер.

Подобное исследование планируется провести в будущем.

Также стоит отметить, что разработанный алгоритм бустинга подходит для решения задач полу-контролируемого обучения с учителем. Например, для классифицированных элементов можно использовать модифицированную матрицу попадания элемента в конкретный кластер и увеличить вероятность вхождения в тот кластер, в который элемент уже был определен. Другой вариант решения проблемы – сведение задачи к задаче кластеризации и игнорирование существующей классификации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Ka-Chun Wong.* A Short Survey on Data Clustering Algorithms // IEEE Second International Conference on Soft Computing and Machine Intelligence, 2015.
2. IBM Consumer products industry blog. Industry insights. Электронный ресурс: https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/ (Дата обращения – 20.05.2018).
3. *Mayr A., Binder H., Gefeller O., Schmid M.* The Evolution of Boosting Algorithms – From Machine Learning to Statistical Modelling // Methods Inf Med. – 2014. – Vol. 53: – P. 419-427.
4. *Donkuan X. Yingjie T.* A comprehensive survey of clastering algorithms // Annals of Data Science. – 2015. – Vol. 2, Issue 2. – P. 165-193.
5. *Зайцев А.А., Курейчик В.В., Полупанов А.А.* Обзор эволюционных методов оптимизации на основе роевого интеллекта // Известия ЮФУ. Технические науки. – 2010. – № 12 (113). – С. 7-12.
6. *Kureichik V.V., Kravchenko Y.A.* Bioinspired algorithm applied to solve the travelling salesman problem // World Applied Sciences Journal. – 2013. – Vol. 22, No. 12. – P. 1789-1797.
7. *Gladkov L.A., Kureichik V.V., Kravchenko Y.A.* Evolutionary algorithm for extremal subsets comprehension in graphs // World Applied Sciences Journal. – 2013. – Vol. 27, No. 9. – P. 1212-1217.
8. *Курейчик В.В., Курейчик В.М., Сороколетов П.В.* Анализ и обзор моделей эволюции // Известия Российской академии наук. Теория и системы управления. – 2007. – № 5. – С. 114-126.
9. *Родзин С.И., Курейчик В.В.* Состояние, проблемы и перспективы развития биоэвристик // Программные системы и вычислительные методы. – 2016. – № 2. – С. 158-172.
10. *Курейчик В.В., Бова В.В., Курейчик Вл.Вл.* Комбинированный поиск при проектировании // Образовательные ресурсы и технологии. – 2014. – № 2 (5). – С. 90-94.
11. *Курейчик В.В., Курейчик Вл.Вл.* Биоиспирированный поиск при проектировании и управлении // Известия ЮФУ. Технические науки. – 2012. – № 11 (136). – С. 178-183.
12. Бустинг. Особенности применения 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: 10.06.2018).
13. *Дружков П.Н., Золотых Н.Ю., Половинкин А.Н.* Программная реализация алгоритма градиентного бустинга деревьев решений // Вестник Нижегородского университета им. Н.И. Лобочевского. – 2011. – № 1. – С. 193-200.
14. Boosting Algorithms: a review of mehods, theory and applications. – https://fenix.tecnico.ulisboa.pt/downloadFile/3779579716974/Boosting%20-%20Ferreira%20and%20Figueiredo%202013.pdf (дата обращения: 29.04.2018).
15. *Mayr A., Binder H., Gefeller O., Schmid M.* The evolution of boosting algorithms – From machine learning to statistical modeling // Methods iInf Med. – 2014. – Vol. 53 (6). – P. 419-427.

16. *Freund Y. and Schapire R.* Experiments with a new boosting algorithm // In Thirteenth International Conference on Machine Learning. – Bari, Italy, 1996. – P. 148-156,
17. *Freund Y. and Schapire R.* A decision-theoretic generali zation of on-line learning and an application to boosting // Journal of Computer and System Sciences. – 1997. – Vol. 55 (1). – P. 119-139.
18. *Kuncheva L.* Combining Pattern Classifiers: Methods and Algorithms. Wiley, 2004.
19. *Radha C, Rong J, Timothy C.H, Anil K.J.* Scalable Kernel Clustering: Approximate Kernel k-means. Computer Vision and Pattern Recognition, 2014.
20. *Курейчик В.М., Курейчик В.В., Родзин С.И.* Модели параллелизма эволюционных вычислений // Вестник Ростовского государственного университета путей сообщения. – 2011. – № 3 (43). – С. 93-97.
21. *Курейчик В.М., Курейчик В.В., Родзин С.И., Гладков Л.А.* Основы теории эволюционных вычислений. – Ростов-на-Дону: ЮФУ, 2010.
22. *Родзин С.И., Курейчик В.В.* Теоретические вопросы и современные проблемы развития когнитивных биоинспирированных алгоритмов оптимизации // Кибернетика и программирование. – 2017. – № 3. – С. 51-79.

REFERENCES

1. *Ka-Chun Wong.* A Short Survey on Data Clustering Algorithms, *IEEE Second International Conference on Soft Computing and Machine Intelligence, 2015*.
2. IBM Consumer products industry blog. Industry insights. Available at: https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/ (accessed 20 May .2018).
3. *Mayr A., Binder H., Gefeller O., Schmid M.* The Evolution of Boosting Algorithms – From Machine Learning to Statistical Modelling, *Methods Inf. Med*., 2014, Vol. 53, pp. 419-427.
4. *Donkuan X. Yingjie T.* A comprehensive survey of clastering algorithms, *Annals of Data Science*, 2015, Vol. 2, Issue 2, pp. 165-193.
5. *Zaycev A.A., Kureychik V.V., Polupanov A.A.* Obzor evolyucionnykh metodov optimizacii na osnove roevogo intellekta [Overview of evolutionary optimization techniques based on swarm intelligence], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12 (113), pp. 7-12.
6. *Kureichik V.V., Kravchenko Y.A.* Bioinspired algorithm applied to solve the travelling salesman problem, *World Applied Sciences Journal*, 2013, Vol. 22, No. 12, pp. 1789-1797.
7. *Gladkov L.A., Kureichik V.V., Kravchenko Y.A.* Evolutionary algorithm for extremal subsets comprehension in graphs, *World Applied Sciences Journal*, 2013, Vol. 27, No. 9, pp. 1212-1217.
8. *Kureychik V.V., Kureychik V.M., Sorokoletov P.V.* Analiz i obzor modeley evolyucii [Analysis and review of models of evolution], *Izvestiya Rossiyskoy akademii nauk. Teoriya i sistemy upravleniya* [Journal of Computer and Systems Sciences International], 2007, No. 5, pp. 114-126.
9. *Rodzin S.I., Kureychik V.V.* Sostoyanie, problemy i perspektivy razvitiya bioevristik [State, problems and prospects of bio-heuristics development], *Programmnye sistemy i vychislitel'nye metody* [Software systems and computational methods], 2016, No. 2, pp. 158-172.
10. *Kureychik V.V., Bova V.V., Kureychik Vl.Vl.* Kombinirovannyy poisk pri proektirovanii [Combined search in design], *Obrazovatel'nye resursy i tekhnologii* [Educational resources and technologies], 2014, No. 2 (5), pp. 90-94.
11. *Kureychik V.V., Kureychik Vl.Vl.* Bioispirirovannyy poisk pri proektirovanii i upravlenii [Biospherology search in the design and management], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 11 (136), pp. 178-183.
12. Busting. Osobennosti primeneniya v oblasti mashinnogo obucheniya [Boosting. Features of 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 10 June 2018).
13. *Druzhkov P.N., Zolotykh N.Yu., Polovinkin A.N.* Programmnaya realizaciya algoritma gradientnogo bustinga derev'ev resheniy [Software implementation of the algorithm is gradient boosting of decision trees], *Vestnik Nizhnegorodskogo universiteta im. N.I. Lobochevskogo* [Vestnik of Lobachevsky University of Nizhni Novgorod], 2011, No. 1, pp. 193-200.
14. Boosting Algorithms: a review of mehods, theory and applications. Available at: https://fenix.tecnico.ulisboa.pt/downloadFile/3779579716974/Boosting%20-%20Ferreira%20and%20Figueiredo%202013.pdf (accessed 29 April 2018).

15. *Mayr A., Binder H., Gefeller O., Schmid M.* The evolution of boosting algorithms – From machine learning to statistical modeling, *Methods iInf Med*., 2014, Vol. 53 (6), pp. 419-427.
16. *Freund Y. and Schapire R*. Experiments with a new boosting algorithm, *In Thirteenth International Conference on Machine Learning*. Bari, Italy, 1996, pp. 148-156,
17. *Freund Y. and Schapire R.* A decision-theoretic generali zation of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, 1997, Vol. 55 (1), pp. 119-139.
18. *Kuncheva L*. Combining Pattern Classifiers: Methods and Algorithms. Wiley, 2004.
19. *Radha C, Rong J, Timothy C.H, Anil K.J*. Scalable Kernel Clustering: Approximate Kernel k-means. Computer Vision and Pattern Recognition, 2014.
20. *Kureychik V.M., Kureychik V.V., Rodzin S.I.* Modeli parallelizma evolyucionnykh vychisleniy [Models of parallelism of evolutionary calculations], *Vestnik Rostovskogo gosudarstvennogo universiteta putey soobshcheniya* [Vestnik RGUPS], 2011, No. 3 (43), pp. 93-97.
21. *Kureychik V.M., Kureychik V.V., Rodzin S.I., Gladkov L.A.* Osnovy teorii evolyucionnykh vychisleniy [Fundamentals of the theory of evolutionary computation]. Rostov-on-Don: YuFU, 2010.
22. *Rodzin S.I., Kureychik V.V.* Teoreticheskie voprosy i sovremennye problemy razvitiya kognitivnykh bioinspirirovannykh algoritmov optimizacii [Theoretical questions and contemporary problems of the development of cognitive bio-inspired algorithms for optimization], *Kibernetika i programmirovanie* [Cybernetics and programming], 2017, No. 3, pp. 51-79.

Статью рекомендовал к опубликованию д.т.н., профессор В.И. Финаев.

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.

Курситыс Илона Олеговна – e-mail: i.kursitys@mail.ru; 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.

Kursitys Ilona Olegovna – e-mail: i.kursitys@mail.ru; the department of computer aided design; graduate student.

УДК 519.113: 681.3 DOI 10.23683/2311-3103-2018-5-131-142

А.И. Долгий, С.М. Ковалев

ИНТЕРПРЕТИРУЕМОСТЬ НЕЧЕТКИХ ТЕМПОРАЛЬНЫХ МОДЕЛЕЙ*

Рассматривается проблема оценки интерпритационной пригодности математических моделей, основанных на нечеткой логики. Показывается, что интерпретируемость является одной из основных причин популярности нечеткой логики и широкого распространения технологий нечеткого моделирования. Разрабатывается подход к оценке интерпретируемости нечетких темпоральных моделей, описывающих динамику процессов. Нечеткие темпоральные модели представлены в виде продукционных правил, антецеденты которых заданы с использованием нечеткого временного отношения предшествования. Идея предлагаемого подхода базируется на предположении, что интерпретируемость

* Работа выполнена при поддержке РФФИ (проекты: №№ 16-07-00032-а, 16-07-00086-а).