UDC 519.87

Вестник СибГАУ
2014. № 5(57). С. 128–135

САМОКОНФИГУРИРУЮЩИЙСЯ ГИБРИДНЫЙ ЭВОЛЮЦИОННЫЙ АЛГОРИТМ ФОРМИРОВАНИЯ НЕЧЕТКИХ КЛАССИФИКАТОРОВ С АКТИВНЫМ ОБУЧЕНИЕМ ДЛЯ НЕСБАЛАНСИРОВАННЫХ ДАННЫХ

В. В. Становов, О. Э. Семенкина

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: vladimirstanovov@yandex.ru

Описывается метод активного выбора обучающих примеров для самоконфигурирующегося гибридного эволюционного алгоритма формирования нечетких баз правил для задач классификации. Данный метод относится к методам отбора измерений, позволяющим не только снизить объем требуемых вычислительных ресурсов, но также улучшить качество получаемых классификаторов. Метод меняет вероятности выбора измерений для обучающей подвыборки в зависимости от того, насколько хорошо они классифицируются алгоритмом. Через некоторое число поколений выборка меняется и вероятности пересчитываются. Те измерения, которые не использовались ранее, и те, на которых алгоритм совершал ошибки, имели большую вероятность попасть в обучающую выборку. Вероятности выбора измерений рассчитывались с использованием процедуры, схожей с процедурой пропорциональной селекции в генетическом алгоритме. Описанная идея выбора обучающих примеров реализована для алгоритма построения нечетких классификаторов. Данный алгоритм использует комбинацию питсбургского и мичиганского подходов для построения баз правил с фиксированными термами, причем мичиганский подход используется вместе с оператором мутации. Размер баз правил не фиксирован и может изменяться в ходе работы алгоритма, а соответствующий номер класса и вес для каждого правила рассчитываются эвристически. Помимо этого в алгоритме применяется инициализация с использованием измерений выборки, для генерации более точных правил. В мичиганской части реализованы операторы добавления правил, удаления правил и замещения правил. При этом создание правил могло производиться как генетически, с использованием имеющихся в базе правил, так и эвристически, с использованием некорректно классифицированных объектов. Работоспособность алгоритма показана на ряде сложных задач классификации с множеством классов, в качестве мер качества классификации использовалась общая точность классификации и средняя точность по всем классам.

Ключевые слова: нечеткие системы классификации, активное обучение, эволюционный алгоритм, самоконфигурация, несбалансированные данные.

Vestnik SibGAU
2014, No. 5(57), P. 128–135

SELF-CONFIGURING HYBRID EVOLUTIONARY ALGORITHM FOR FUZZY CLASSIFIER DESIGN WITH ACTIVE LEARNING FOR UNBALANCED DATASETS

V. V. Stanovov, O. E. Semenkina

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: vladimirstanovov@yandex.ru

The paper describes an active training example selection for a self-configured hybrid evolutionary algorithm for fuzzy rule bases design for classification problems. This method is related to instance selection methods, which allow not only decreasing of required computational recourses, but also increasing the quality of the obtained classifiers. The method changes the probabilities of instances which are selected into the training subsample depending on how good they are classified by the algorithm. After several generations the sample is changed and probabilities are recalculated. Those instances which were not used before and those which were misclassified by the algorithm had higher probabilities of getting into the training sample. The probabilities of instance selection were calculated using a procedure similar to proportional selection in the genetic algorithm. The idea of training instance selection described here was implemented for the fuzzy classifiers forming. This algorithm uses the combination of Pittsburg and Michigan approach for fuzzy rule base design with fixed terms, and the Michigan approach is used together with the mutation operator. The size of the rule base is not fixed, and may change during the algorithm run, and a corresponding class number and the rule weight were calculated heuristically for every rule. Moreover, the algorithm uses an initialization procedure that

uses instances from the sample to generate more accurate rules. In the Michigan part the operators of adding rules, deleting rules and replacing rules has been implemented. The creation of new rules could be performed by genetic approach – using the existing rules, and heuristically – using those instances which were misclassified. The efficiency of the algorithm was shown on a set of complex classification problems with several classes, as an efficiency measure the overall accuracy and the average accuracy among classes was used.

Keywords: fuzzy classification system, active learning, evolutionary algorithm, self-configuration, unbalanced data.

Введение. Современные методы машинного обучения часто используют эволюционные либо бионические алгоритмы, в качестве алгоритмов формирования структуры или настройки своих компонентов, таких как искусственные нейронные сети, машины опорных векторов (support vector machines, SVM), системы на нечеткой логике, нейронечеткие системы или же их объединения в коллективы. Эволюционные и бионические алгоритмы зачастую применяются из-за сложности возникающих в процессе обучения процедур классификации, которые могут содержать большое число переменных либо переменные различных типов, множество целевых функций, множество экстремумов, плато и т. д. Для решения подобных задач лучше всего себя зарекомендовали эволюционные подходы [1; 2].

Однако использование эволюционных методов, как правило, сопряжено с необходимостью использования значительных вычислительных ресурсов. Если говорить о задачах классификации, которые являются типичными задачами анализа данных, то существует ряд подходов, позволяющих производить машинное обучение на значительных объемах данных с использованием методов сокращения данных (Data reduction). Сокращение данных может преследовать две цели: снижение числа переменных, иначе выбор информативных признаков, и снижение числа измерений. Задача отбора информативных признаков является нетривиальной задачей, для которой разработано множество подходов и методов, в том числе использующих эволюционные алгоритмы [3; 4].

Задача отбора измерений также является важной задачей, и группу методов, выбирающих некоторые измерения из их большого массива, называют отбором обучающей выборки (Training Set Selection, TSS) или же селекцией измерений (Instance Selection, IS) [5]. Суть данных методов, как правило, сводится к отбору некоторых прототипов (Prototype Selection, PS) – объектов, в определенной степени описывающих тот или иной класс.

Однако проблема отбора измерений часто связана с проблемой релевантности выбранной подвыборки, изначальной выборке данных. Удаление измерений из обучающей выборки не обязательно приводит к уменьшению информации, полезной для обучения. Более того, снижение объема выборки может помочь предотвратить переобучение в некоторых случаях, так как изначальная выборка может содержать шум. Таким образом, методы отбора измерений могут не только снизить объем требуемых вычислительных ресурсов, но и повысить качество формируемых моделей, например классификаторов. Таким образом, основной идеей данной работы является разработка таких алгоритмов выбора измерений, которые позволили бы формировать более точные и качественные классификаторы, затрачивая при этом меньше вычислительных ресурсов. Некоторые предыдущие работы в данной области использовали методы распараллеливания и разбиения выборки на подвыборки случайным образом [6].

В следующих разделах мы рассмотрим примененный алгоритм построения нечетких классификаторов, опишем идею активного выбора обучающих примеров и далее приведем результаты тестирования модифицированного алгоритма.

Гибридный нечеткий эволюционный алгоритм. За основу в использованном алгоритме формирования нечетких баз правил для задач классификации был взят алгоритм, разработанный группой Х. Ишибучи и описанный в работе [6]. Данный метод использует комбинацию питсбургского и мичиганского подходов, при этом мичиганский подход применяется наряду с оператором мутации. Ниже будет приведено краткое описание метода, так как он был запрограммирован с нуля и потому отличается от изначальной идеи в некоторых аспектах.

Основная популяция эволюционного алгоритма реализует питсбургский подход, где каждый индивид является базой правил целиком. В нашей реализации индивид представлял собой двумерную матрицу, строки которой являлись правилами. Элементы строки принимали значения в диапазоне [0, 14] соответственно 15 использованным нечетким множествам. Число правил в базе не фиксировано и может меняться в ходе работы алгоритма, однако не может превышать заранее заданный верхний лимит.

Правила выглядят следующим образом:

$$R_q : \text{если } X_1 \text{ это } A_{q,1} \text{ и } X_2 \text{ это } A_{q,2} \text{ и ... и } X_n \text{ это } A_{q,n}$$

$$\text{тогда } X \text{ это } C_q \text{ с весом } CF_q,$$

где $X_p = (X_{p1}, X_{p2}, ..., X_{pm}), p = 1...m$ – выборка измерений; n – число переменных в задаче; m – число измерений; R_q – это правило; $A_{q,i}$ – нечеткое множество, C_q – номер соответствующего класса; CF_q – вес правила.

Набор нечетких множеств представляет собой 4 разбиения на 2, 3, 4 и 5 нечетких множеств, показанных на рис. 1. Помимо этих 14 множеств используется также терм игнорирования значения переменной (“Don’t care” condition, DC). Данный метод задания нечетких множеств является достаточно гибким и позволяет решать задачи с высокой точностью, не применяя процедур настройки положений термов. Тем не менее в литературе встречаются другие подходы, к примеру, 2-парное иерархическое представление.

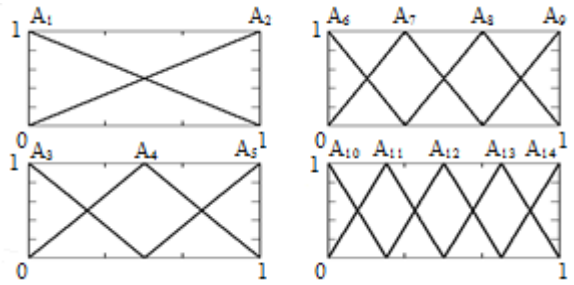

Рис. 1. Четыре нечетких разбиения, использованных в алгоритме

В качестве операторов селекции использовались ранговая и турнирная селекция с размером турнира 2, а также три типа мутации – слабая, средняя, сильная. Применялся один оператор скрещивания, специфический для данного алгоритма. Вероятность мутации зависела от размера базы правил, т. е. от числа правил следующим образом: для слабой мутации – $1/(3\cdot|S|\cdot n)$, для средней – $1/(|S|\cdot n)$, для сильной – $3/(|S|\cdot n)$, где $|S|$ – число правил в базе, n – число переменных. При скрещивании новая база правил состояла из правил, содержащихся у родителей, при этом число правил выбиралось как случайное число в диапазоне $[1, |S_1| + |S_2|]$. Правила для заполнения нового индивида также выбирались равномерно и случайно из общего пула правил родителей. Если конечное число правил превышало верхний лимит, то несколько правил удалялись случайным образом.

Инициализация популяции производилась с использованием измерений, содержащихся в выборке. Данный шаг необходим, потому что при генерации случайных правил вероятность того, что сгенерированное правило будет описывать хотя бы часть выборки, очень мала. Поэтому для формирования базы правил случайным образом выбиралось измерение. Далее подбиралось правило, которое в наилучшей степени описывает данное измерение, при этом, если под это правило подходят несколько термов из различных разбиений, то они имеют шанс быть выбранными в правило с определенной вероятностью, зависящей от степеней принадлежности. После формирования правила половина термов в нем случайным образом изменяется на терм игнорирования значения переменной. Этот шаг необходим для получения более общих правил.

После генерации правил рассчитывается наиболее подходящий номер класса и вес нечеткого правила. Эти величины рассчитываются с использованием меры поддержки (Confidence, *Conf*) правила по выборке, позаимствованной из области оценки ассоциативных правил:

$$Conf\left(A_q \to Class\, k\right) = \frac{\sum_{x_p \in Class\, k} \mu_{Aq}\left(x_p\right)}{\sum_{p=1}^{m} \mu_{Aq}\left(x_p\right)},$$

где $\mu_{Aq}\left(x_p\right)$ – это степень принадлежности измерения x_p к нечеткому множеству A_q. Класс, которому соответствует наибольшее значение поддержки, устанавливается в качестве наиболее подходящего класса. Вес правила рассчитывается также с использованием поддержки [7]:

$$CF_q = Conf\left(A_q \to Class\, q\right) - \\ - \sum_{k=1, k\neq C_q}^{M} Conf\left(A_q \to Class\, k\right).$$

При нечетком выводе объект классифицируется правилом-победителем, т. е. правилом, имеющим наибольшее произведение степени принадлежности на вес правила. Степень принадлежности при этом рассчитывается как произведение степеней принадлежности по всем переменным. Стоит отметить, что эвристика назначения номера класса назначает правилу тот класс, который имеет наибольшее значение поддержки, однако для несбалансированных данных значения поддержки по каждому из классов может быть смещено, так как поддержка зависит от числа измерений каждого класса. Это значит, что для классов, имеющих меньшее число измерений, значения поддержки могут быть меньше не потому, что данное правило хуже описывает этот класс, а потому что меньше измерений этого класса описывается правилом. Это приводит к тому, что значительная часть сгенерированных правил связывается с классами, имеющими большее число измерений, в то время как данные правила могут оказаться важными для описания миноритарного класса.

Чтобы избежать данного смещения, здесь мы используем модифицированную процедуру определения наиболее подходящего номера класса, использующую число измерений каждого класса в качестве веса:

$$Class\, q = \arg\max_k \left(\frac{Conf\left(A_q \to Class\, k\right)}{m_k} \cdot m\right),$$

где m_k – это число измерений класса k.

Мичиганская часть могла быть выполнена тремя способами: добавление правил, удаление правил и замещение правил. При этом каждое правило рассматривалось как индивид, а база правил – как популяция. Пригодности правилам назначались в соответствии с числом измерений, которые были корректно классифицированы правилом. При этом удалялись правила, имеющие наименьшую пригодность. Добавление новых правил производилось двумя типами: генетическим подходом и эвристическим подходом. Генетический подход подразумевал получение новых

правил из имеющихся в базе посредством применения генетических операторов, таких как селекция, скрещивание и мутация. Эвристический подход генерировал новые правила из некорректно классифицированных измерений. Замещение правил подразумевало сначала удаление нескольких правил, а затем добавление равного числа правил. При этом число правил для удаления / добавления зависело от текущего числа правил в базе.

Пригодность базы правил рассчитывалась как сумма точности на обучающей подвыборке в процентах с весом 100, числа правил в базе с весом 1 и общей длины всех правил с весом 1. На сегодняшний день множество алгоритмов использует многокритериальные подходы [8].

Так как в алгоритме использовалось несколько типов селекции, мутации, а также несколько операторов в мичиганской части и типов добавления правил, и для всех них была применена процедура самонастройки. Самонастройка необходима по той причине, что различные типы операторов имеют разные свойства и меняют поведение алгоритма в процессе поиска. Заранее, до решения поисковой задачи, невозможно определить, какие из операторов будут лучшими для конкретной задачи. Применение самонастройки меняет вероятности применения операторов в процессе работы алгоритма так, что более успешные операторы получают большую вероятность быть выбранными.

В данной работе был использован подход, ранее описанный в [9; 10] и показавший свою эффективность в [11]. Суть метода заключается в поощрении тех операторов, которые позволили получить более высокую пригодность в среднем на каждом поколении. Пусть z – число операторов определенного типа. Начальные вероятности устанавливаются равными: $p_i = 1/z$. Успешность операторов определяется с использованием усредненных пригодностей:

$$AvgFit_i = \frac{1}{n_i}\sum_{j=1}^{n_i} f_{ij}, i = 1,\ 2,\ ...,\ z,$$

где n_i – количество потомков, в формировании которых принял участие i-й тип оператора; f_{ij} – пригодность j-го потомка, построенного с помощью i-го оператора; $AvgFit_i$ – средняя пригодность решений, построенных при помощи i-го оператора.

После этого вероятность применения оператора, чье значение $AvgFit_i$ является наибольшим среди всех операторов такого типа, увеличивается на $(zK\text{-}K)/(zN)$, а вероятности применения остальных операторов уменьшаются на $K/(zN)$, где N – число поколений генетического алгоритма, K – константа, её значение устанавливалось равным 0,5.

Активный выбор обучающих примеров из выборки. Как мы отмечали во введении, сокращение данных может в некоторых случаях привести к более низкой точности на тестовой выборке. Таким образом, следовало бы разработать метод селекции измерений, который бы позволял получать как минимум столь же точные результаты, как и классический подход. Подобные подходы ранее рассматривались в [12; 13]. Метод выбора измерений, предлагаемый в данной работе, использует информацию о корректной классификации объектов выборки для выбора тех измерений, которые более сложны для классификации. Так как мы используем эволюционный подход, то существует возможность менять обучающую выборку в ходе обучения каждые несколько поколений.

На начальном этапе мы выбираем 33 % начальной обучающей выборки и формируем подвыборку, при этом все измерения имеют равную вероятность быть выбранными. После процедуры инициализации алгоритм обучается в течение 200 поколений. На каждом поколении лучший текущий индивид проверяется на всей выборке, и если он представляет собой лучшее известное решение для всей выборки, то запоминается. Лучшее известное решение всегда включается в популяцию наряду с лучшим решением для всей выборки. Стоит отметить, что, так как мы работаем с подвыборкой, то лучший текущий индивид может проявлять признаки переобучения на подвыборку, вследствие чего он может некорректно классифицировать всю имеющуюся выборку. По этой причине необходимо запоминать наилучшее найденное на текущий момент решение.

После периода адаптации длительностью в 200 поколений в наших вычислительных экспериментах мы рассчитываем точности каждого индивида на всей выборке. Это шаг является важным, так как при том, что лучшее решение для подвыборки может показывать средние результаты для всей выборки, в популяции могут присутствовать другие индивиды, которые, хотя и являются не лучшими решениями для подвыборки, представляют собой очень хорошие решения для описания всей выборки.

Для того чтобы направить процесс обучения на следующем периоде адаптации и сконцентрировать процесс обучения на «проблематичных» областях пространства переменных, необходимо изменить вероятности выбора измерений. Во-первых, необходимо выбирать те измерения, которые не были использованы на предыдущих этапах. Во-вторых, следует также выбирать измерения, которые были неверно классифицированы. Первый принцип соответствует исследованию новых имеющихся измерений, в то время как второй означает поиск и концентрацию на наиболее интересных областях пространства. После каждого периода адаптации длиной в 200 поколений необходимо заново выбрать новые измерения для формирования подвыборки.

Для реализации данной идеи каждому измерению ставится в соответствие число U_i, $i = 1, \ldots, m$. В начале все U_i устанавливаются равными 1. После выбора этих измерений и обучения классификатора значения U_i для тех измерений, которые были выбраны, изменяются в соответствии с лучшим индивидом для подвыборки. Если измерение j было выбрано и классифицировано корректно, то $U_j = U_j + 1$, если же оно было классифицировано неверно, то $U_j = 1$.

После обновления значений U_i формируется обучающая выборка с использованием процедуры, подобной пропорциональной селекции в классическом генетическом алгоритме. Вероятность измерения быть выбранным p_i рассчитывается как

$$p_i = \frac{1/U_i}{\sum_{j=1...m} 1/U_j}.$$

Те измерения, которые были классифицированы корректно, увеличивают свой счетчик успешных использований в обучающей выборке. В случае, если после периода адаптации лучший индивид для подвыборки неверно классифицирует какое-либо измерение, которое было неоднократно классифицировано корректно, его счетчик устанавливается равным 1, что означает большую вероятность быть выбранным в будущем. Таким же образом те измерения, которые всегда некорректно классифицируются, всегда сохраняют большую вероятность быть выбранными. После нескольких периодов адаптации те измерения, которые неоднократно были классифицированы корректно, получают меньшие вероятности выбора. Это приводит к ситуации, когда те измерения, которые легко поддаются классификации, реже выбираются в подвыборку, так как они представляют меньший интерес для процесса обучения. Метод стремится использовать эти измерения равномерно, чтобы покрыть всю информацию, имеющуюся в обучающей выборке. После значительного числа поколений алгоритм находит проблематичные области и фокусируется на них, чтобы найти лучшее разделение классов.

Рис. 2. Изменение вероятностей выбора примеров

Чтобы лучше понять эффекты, которые возникают в результате подобной процедуры, далее приводится два рисунка (рис. 2, 3), объясняющих изменение вероятностей выбора измерений и изменение того, какие из измерений выбираются. На рис. 2 представлен пример изменения вероятностей для случая 8 измерений. В начале все измерения имеют равные вероятности и все U_i равны 1. Выбранные измерения показаны серым цветом. После первого периода адаптации измерение 3 было классифицировано некорректно, так что значение U_3 было установлено равным 1, а измерение 6 было верно классифицировано, поэтому U_6 увеличилось на 1. После нескольких периодов адаптации значения U были обновлены в соответствии с формулой. Таким образом, измерения 1 и 7 имеют более высокую вероятность быть выбранными, чем другие.

На рис. 3 показан графический пример того, какие измерения выбираются алгоритмом и как процедура активного выбора примеров приводит к повышению качества классификации. В примере 2 класса и 31 измерение, различные классы показаны крестиками и квадратами. Слева показано начало работы алгоритма, при котором все измерения имеют равные вероятности быть выбранными. Большие либо меньше вероятности показаны размерами знаков. Знаки, обведенные кружками, означают выбранные для данного периода адаптации. Разделяющая поверхность, полученная к концу первого периода адаптации, некорректно классифицирует измерения 16 и 18, которые имеются в подвыборке. Справа показан следующий период адаптации, где измерения 16 и 18 не изменили размер ($U_{16} = U_{18} = 1$) и их размер равен размеру неиспользованных измерений, в то время как верно классифицированные примеры (1, 3, 7, 12, 14, 19, 20, 28, 29) получили меньший размер и соответственно меньшую вероятность.

Нижний график (рис. 3) демонстрирует ситуацию после нескольких периодов адаптации, где те измерения, которые ближе к разделяющей поверхности, получили большие вероятности быть выбранным (и они действительно выбраны на данной итерации), в то время как далекие от разделяющей поверхности измерения получили меньшие вероятности.

Меры *Acc* и *Ave* для определения качества классификации. При решении сложных задач классификации методы машинного обучения могут быть неэффективны вследствие различных свойств имеющихся данных. Одним из таких свойств является сбалансированность выборки по классам. Если один из классов представлен большим числом измерений, чем другой, то у алгоритма обучения могут быть проблемы с корректным выделением миноритарного класса. Задачи классификации со значительным дисбалансом, как правило, более сложны при решении, чем сбалансированные задачи. Несмотря на то, что на них может быть получена высокая точность классификации, это может означать, что верно классифицирован только мажоритарный класс, в то время как миноритарный класс, который часто представляет больший интерес, классифицируется неверно. Это означает, что, несмотря на высокую точность, задача остается нерешенной.

Проблема несбалансированных данных ранее рассматривалась в [14]. Наряду с общей точностью *Acc*, как мера классификации для несбалансированных данных часто применяется средняя точность по классам *Ave*. Точности на всех классах суммируются и затем делятся на число классов, что в итоге дает значение меры *Ave*. Следующие два выражения формально описывают данные меры:

$$Acc = \frac{\sum_{i=0}^{k} E_i}{\sum_{i=0}^{k} m_i}, \quad Ave = \sum_{i=0}^{k} \frac{E_i}{m_i \cdot k},$$

где E_i – это число ошибок классификатора на i-м истинном классе; m_k – это число измерений класса k. Далее будет показано, как активный подход к обучению может повысить не только точность классификации, но и сбалансированность по классам даже при использовании меры *Acc* в функции пригодности.

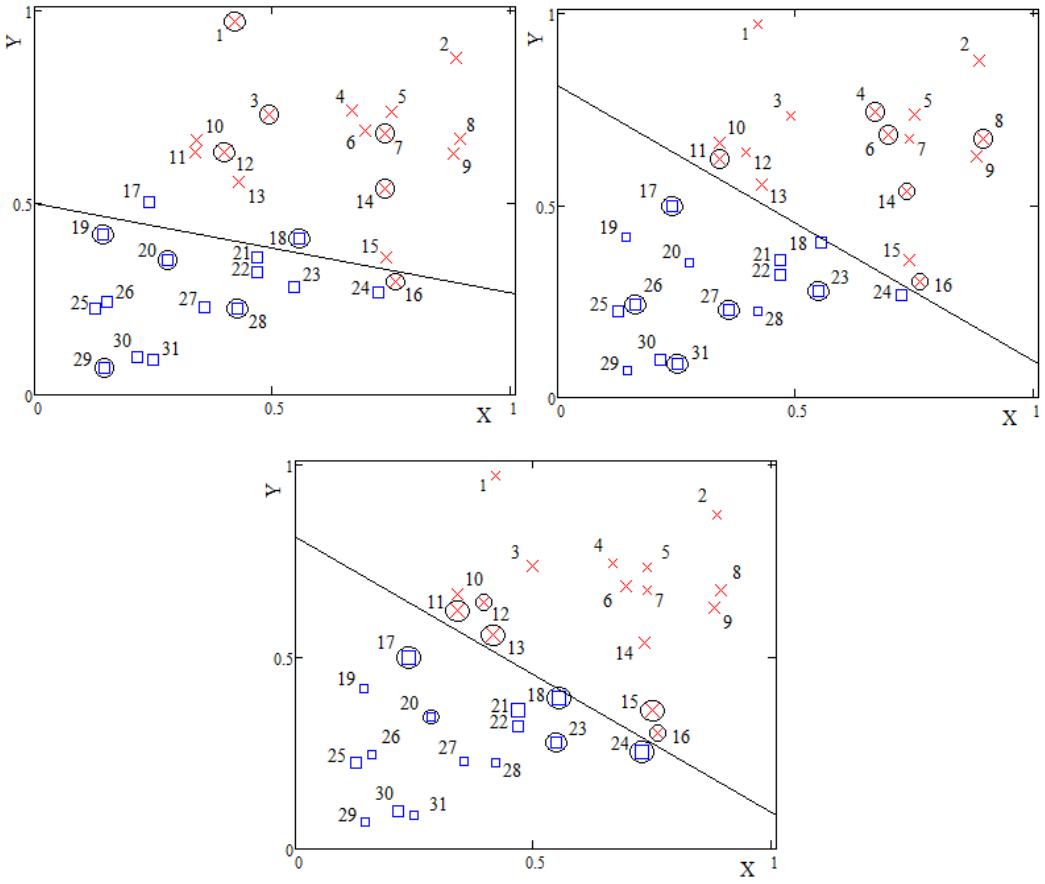

Рис. 3. Активный выбор обучающих примеров

Тестирование алгоритма и результаты. Для тестирования алгоритма было выбрано 4 задачи классификации с репозиториев KEEL и UCI [15; 16]. Данные задачи могут представлять трудности для некоторых алгоритмов, так как содержат большое число переменных, множество классов, а также несколько тысяч измерений. Задачи, на которых мы протестировали алгоритмы, следующие:

1) Page-blocks, 5472 измерения, 10 переменных, 5 классов;

2) Phoneme, 5404 измерения, 5 переменных, 2 класса;

3) Segment, 2310 измерений, 19 переменных, 7 классов;

4) Satimage, 6435 измерений, 36 переменных, 6 классов.

Табл. 1 показывает число измерений для каждого класса этих задач.

При тестировании алгоритма мы использовали следующие параметры: число индивидов – 100, число поколений – 10 000, число правил – 40, размер обучающей подвыборки – 33 % от всей выборки, длительность периода адаптации – 200 поколений. Для каждой задачи тестировалось 8 вариантов алгоритма: с активным обучением и без, с несмещенной процедурой назначения номеров классов и оригинальной, с мерой *Acc* и мерой *Ave* в функции пригодности. Для получения адекватных результатов мы использовали процедуру 10-частной кроссвалидации со стратифицированным разбиением, 2 раза для каждой задачи. Стоит отметить, что время работы алгоритма с активным обучением сокращалось в 3 раза.

В табл. 2 показаны результаты для задачи Page-blocks. Следует отметить, что для данной задачи активный выбор примеров из обучающей выборки незначительно меняет поведение алгоритма обучения и получаемые в результате классификаторы. На обучающей выборке точность обоих методов практически одинакова, также как и точность на тестовой. При этом мера *Ave* для случая с активным обучением оказывается ниже практически во всех случаях. Следует отметить, что данная задача является наименее сбалансированной из всех.

Результаты для задачи Phoneme демонстрируют преимущество активного обучения перед классическим подходом: точность при использовании активного обучения увеличивается как для обучающей, так и для тестовой выборки, хотя в случае использования меры *Acc* в функции пригодности это приводит к получению менее сбалансированных классификаторов (табл. 3). При использовании меры *Ave* в качестве меры качества классификации, алгоритм получает в равной мере сбалансированные классификаторы, но с более высокой точностью.

Для задачи Segment, для которой все классы сбалансированы, активное обучение позволяет достичь значительного улучшения качества классификации (табл. 4). Так как эта задача сбалансирована, то значения меры *Ave* и *Acc* совпадают.

Таблица 1

Число объектов классов для всех задач

Задача/класс	1	2	3	4	5	6	7
Page-blocks	4913	329	28	87	115	–	–
Phoneme	3818	1586	–	–	–	–	–
Segment	330	330	330	330	330	330	330
Satimage	1533	703	1358	626	707	1508	–

Таблица 2

Результаты для задачи Page-blocks

Результаты без активного обучения				
Конфигурация	*Acc* на обуч.	*Acc* на тест.	*Ave* на обуч.	*Ave* на тест.
Acc+смещ.	**0,965**	**0,960**	0,647	0,623
Acc+несмещ.	0,946	0,944	0,582	0,566
Ave+смещ.	0,925	0,922	0,919	0,867
Ave+несмещ.	0,917	0,917	**0,927**	**0,895**
Результаты с активным обучением				
Acc+смещ.	**0,964**	**0,959**	0,635	0,586
Acc+несмещ.	**0,964**	**0,960**	0,660	0,663
Ave+смещ.	0,909	0,900	0,877	0,849
Ave+несмещ.	0,898	0,892	0,902	0,848

Таблица 3

Результаты для задачи Phoneme

Результаты без активного обучения				
Конфигурация	*Acc* на обуч.	*Acc* на тест.	*Ave* на обуч.	*Ave* на тест.
Acc+смещ.	0,827	0,818	0,798	0,788
Acc+несмещ.	0,819	0,812	0,783	0,774
Ave+смещ.	0,816	0,807	**0,836**	**0,825**
Ave+несмещ.	0,805	0,794	0,820	0,808
Результаты с активным обучением				
Acc+смещ.	**0,837**	**0,826**	0,738	0,767
Acc+несмещ.	**0,839**	**0,829**	0,785	0,773
Ave+смещ.	0,820	0,815	0,831	**0,823**
Ave+несмещ.	0,820	0,810	**0,836**	**0,824**

Таблица 4

Результаты для задачи Segment

Результаты без активного обучения				
Конфигурация	*Acc* на обуч.	*Acc* на тест.	*Ave* на обуч.	*Ave* на тест.
Acc+смещ.	0,934	0,919	0,934	0,919
Acc+несмещ.	0,933	0,918	0,933	0,918
Ave+смещ.	0,928	0,910	0,928	0,910
Ave+несмещ.	0,930	0,917	0,930	0,917
Результаты с активным обучением				
Acc+смещ.	**0,967**	0,942	**0,967**	0,942
Acc+несмещ.	**0,966**	0,939	**0,966**	0,939
Ave+смещ.	**0,965**	0,943	**0,965**	0,943
Ave+несмещ.	**0,967**	**0,949**	**0,967**	**0,949**

Таблица 5

Результаты для задачи Satimage

Результаты без активного обучения				
Конфигурация	*Acc* на обуч.	*Acc* на тест.	*Ave* на обуч.	*Ave* на тест.
Acc+смещ.	0,842	0,831	0,760	0,748
Acc+несмещ.	0,843	0,834	0,761	0,752
Ave+смещ.	0,845	0,835	0,770	0,759
Ave+несмещ.	0,843	0,836	0,769	0,760
Результаты с активным обучением				
Acc+смещ.	**0,886**	**0,873**	0,843	0,830
Acc+несмещ.	**0,884**	0,865	0,844	0,822
Ave+смещ.	0,878	0,863	**0,860**	**0,845**
Ave+несмещ.	0,874	0,857	0,856	0,835

Для задачи Satimage активное обучение также демонстрирует лучше результаты, чем классический подход (табл. 5). При этом значительное улучшение наблюдается как на обучающей, так и на тестовой выборке, как при использовании меры *Acc*, так и меры *Ave* в функции пригодности. Более того, получаемые классификаторы во всех случаях значительно более сбалансированы.

Заключение. В данной работе был рассмотрен подход к сокращению данных, который позволяет не только сократить объем требуемых вычислительных ресурсов, но и повысить качество получаемых классификаторов. Представленный алгоритм активного обучения классификаторов посредством выбора обучающих примеров показал свою эффективность на ряде задач классификации, при этом формируя не только более точные в целом базы правил, но также и одинаково точные на всех классах. Дальнейшая разработка данного метода может включать модификации назначения счетчиков для измерений, изучение зависимости результатов от объема подвыборки и длины периода адаптации. Также интересным представляется применение данного подхода для других алгоритмов и методов классификации, например, для нейронных сетей [17; 18], машин опорных векторов [19], генетического программирования [20] и других методов.

Благодарности. Работа выполнена при финансовой поддержке Министерства образования и науки Российской Федерации, проект 140/14.

Acknowledgements. This work was financially supported by the Ministry of Education and Science of the Russian Federation, project 140/14.

References

1. L. B. Booker, D. E. Goldberg, and J. H. Holland, Classifier systems and genetic algorithms. *Artif. Intell*. 1989, vol. 40, no. 1–3, p. 235–282.

2. Bodenhofer U., Herrera F. Ten Lectures on Genetic Fuzzy Systems. Preprints of the International Summer School: Advanced Control–Fuzzy, Neural, Genetic. 1997. Slovak Technical University, Bratislava. P. 1–69.

3. Brester C., Semenkin E. Development of adaptive genetic algorithms for neural network models multicriteria design. *Vestnik SibGAU*. 2013, no. 4 (50), p. 99–103 (In Russ.).

4. Brester Ch. Yu., Semenkin E. S., Sidorov M. Yu., Automatic informative feature extraction system for emotion recognition in speech. *Program products and systems*. 2014, no. 4 (108), p. 127–131.

5. J. R. Cano, F. Herrera, M. Lozano, Evolutionary Stratified Training Set Selection for Extracting Classification Rules with trade off Precision-Interpretability. *Data & Knowledge Engineering archive*. 2007, no. 60, Iss. 1, p. 90–108.

6. Ishibuchi H., Mihara S., Nojima Y. Parallel Distributed Hybrid Fuzzy GBML Models With Rule Set Migration and Training Data Rotation. *IEEE Transactions on fuzzy systems*. 2013, vol. 21, no. 2.

7. Ishibuchi H., T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems. *IEEE Trans. Fuzzy Systems*. 2005, no. 13, p. 428–435.

8. M. Fazzolari, R. Alcalá, Y. Nojima, H. Ishibuchi, F. Herrera, A Review of the Application of Multi-Objective Evolutionary Fuzzy Systems: Current Status and Further Directions. *IEEE Transactions on Fuzzy Systems*. 2013, vol. 21, no. 1, p. 45–65.

9. E. Semenkin, M. Semenkina, Self-configuring genetic algorithm with modified uniform crossover operator. in Y. Tan, Y. Shi, Z. Ji (Eds.), *Advances in Swarm Intelligence*. 2012. PT1, LNCS 7331, p. 414–421.

10. E. Semenkin, M. Semenkina, Self-Configuring Genetic Programming Algorithm with Modified Uniform Crossover. *Proc. of the IEEE Congress on Evolutionary Computation*, (CEC), Brisane (Australia). 2012.

11. M. Semenkina, E. Semenkin, Hybrid self-configuring evolutionary algorithm for automated design of fuzzy classifier. in Y. Tan, Y. Shi, C.A.C. Coello (Eds.), *Advances in Swarm Intelligence*. 2014. PT1, LNCS 8794, p. 310–317.

12. J. R. Cano, F. Herrera, M. Lozano, Stratification for scaling up evolutionary prototype selection. *Pattern Recognition Letters*. 2005, vol. 26, Iss. 7, p. 953–963.

13. J. R. Cano, F. Herrera, M. Lozano, A Study on the Combination of Evolutionary Algorithms and Stratified Strategies for Training Set Selection in Data Mining. *Advances in Soft Computing*. 2005, no. 32, p. 271–284.

14. Bhowan U., Genetic Programming for Classification with Unbalanced Data. Victoria University of Wellington. 2012.

15. J. Alcalá-Fdez, L. Sánchez, S. Garcia, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, KEEL: A software tool to assess evolutionary algorithms for data mining problems. *Soft Comput*. 2009, vol. 13, no. 3, p. 307–318.

16. Asuncion A., Newman D. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. 2007. Available at: http:// www.ics.uci.edu/~mlearn/MLRepository.html.

17. Akhmedova S. A., Semenkin E. S., Co-Operation of Biology Related Algorithms Meta-Heuristic in ANN-Based Classifiers Design. *Proceedings of the World Congress on Computational Intelligence* (WCCI'14). 2014. P. 867–872.

18. Khritonenko D. I., Semenkin E. S. Distributed Self-Configuring Evolutionary Algorithms for Artificial Neural Networks Design. *Vestnik SibGAU*. 2013, no. 4 (50), p. 112–116 (In Russ.).

19. Akhmedova S. A., Semenkin E. S., Gasanova T., Minker. W., Co-Operation of Biology Related Algorithms for Support Vector Machine Automated Design. *Engineering and Applied Sciences Optimization* (OPT-i'14). 2014. P. 1831–1837.

20. Semenkin E., Semenkina M. Classifier Ensembles Integration with Self-Configuring Genetic Programming Algorithm Adaptive and Natural Computing Algorithms. *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, 2013, vol. 7824, p. 60–69.