

СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ

УДК 004.855.5

АЛГОРИТМ ПОДБОРА ЗНАЧЕНИЙ ПАРАМЕТРОВ BSMOTE-АЛГОРИТМА В ЗАДАЧЕ SVM-КЛАССИФИКАЦИИ НА ОСНОВЕ НЕСБАЛАНСИРОВАННЫХ НАБОРОВ ДАННЫХ

Л. А. Демидова, д.т.н., профессор кафедры ВПМ РГРТУ; demidova.liliya@gmail.com

И. А. Ключева, аспирант РГРТУ; i.klyueva-job@yandex.ru

Рассматривается задача SVM-классификации (Support Vector Machine, SVM) на основе несбалансированных наборов данных, используемых для формирования обучающих выборок, с применением алгоритма синтетического сэмплинга – bSMOTE-алгоритма (borderline Synthetic Minority Over-sampling Technique algorithm).

Целью работы является разработка алгоритма подбора значений параметров bSMOTE-алгоритма в задаче SVM-классификации несбалансированных наборов данных, обеспечивающего сокращение временных затрат на разработку SVM-классификатора, характеризующегося высоким качеством классификации данных. Поиск значений параметров SVM-классификатора реализован с применением PSO-алгоритма (Particle Swarm Optimization algorithm). Приведены результаты экспериментальных исследований, подтверждающие целесообразность применения алгоритма подбора значений параметров bSMOTE-алгоритма в задаче SVM-классификации несбалансированных наборов данных.

Ключевые слова: несбалансированность данных, сэмплинг, bSMOTE-алгоритм, классификация, SVM-классификатор, радиальная базисная функция ядра, PSO-алгоритм.

DOI: 10.21667/1995-4565-2017-61-3-67-77

Введение

Большинство стандартных алгоритмов машинного обучения предполагают использование при разработке классификаторов сбалансированных обучающих наборов данных с равными стоимостями ошибки классификации для всех имеющихся примеров. Проблема обучения на несбалансированных наборах данных [1-6] заключается в возможности значительного снижения качества разрабатываемых классификаторов, т.к. такие наборы данных не обеспечивают требуемых характеристик распределения данных при обучении. Проблема обучения на несбалансированных наборах данных является достаточно распространенной темой для исследований последних лет [1-6].

Обучение классификаторов на несбалансированных наборах данных ставит под угрозу эффективность большинства известных алгоритмов машинного обучения, в частности, одного из

наиболее востребованных в настоящее время алгоритма обучения по прецедентам – алгоритма машины опорных векторов (SVM, Support Vector Machine) [2, 4, 7-13].

Как показывают результаты экспериментальных исследований, обучение классификаторов на несбалансированных наборах данных приводит к тому, что разработанный в итоге классификатор стремится классифицировать все объекты в качестве объектов мажоритарного класса (класса «большинства»), практически игнорируя менее представленный миноритарный класс (класса «меньшинства»), что обычно не соответствует фактической цели исследования [2, 4].

Проблема несбалансированности наборов данных встречается в различных прикладных задачах [3], например, в задаче медицинской диагностики, в которой больных пациентов, как правило, существенно меньше, чем здоровых; в задаче обнаружения мошеннических транзакций по банковским картам, в которой число мошен-

нических транзакций значительно меньше, чем обычных; кредитный скоринг, при котором число недобросовестных заемщиков является очень малым по сравнению с числом добросовестных; в задаче анализа оттока клиентов в сфере услуг (например, в телекоммуникационной отрасли), в которой число клиентов, желающих отказаться от услуг компании, существенно меньше числа остальных клиентов и т.п.

В настоящее время для решения проблемы несбалансированности наборов данных применяются различные стратегии сэмпинга [3-6]. При этом восстановление баланса классов может проходить двумя путями. В первом случае удаляют некоторое число объектов мажоритарного класса (англ. *undersampling*, андэрсэмплинг), во втором – увеличивают число объектов миноритарного (англ. *oversampling*, овэрсэмплинг).

Одним из наиболее известных алгоритмов овэрсэмплинга является SMOTE-алгоритм (*Synthetic Minority Oversampling Technique algorithm*) [2-5], для которого предложен ряд перспективных модификаций, предназначенных, в частности, для более адекватного учета характерных свойств объектов миноритарного класса. Одной из таких модификаций SMOTE-алгоритма является bSMOTE-алгоритм (*borderline-SMOTE algorithm*), реализующий синтезирование новых объектов миноритарного класса для объектов этого же класса, расположенных вблизи границы классов.

В случае применения bSMOTE-алгоритма необходимо определить оптимальные значения его параметров, использование которых обеспечит лучший вариант восстановления сбалансированности набора данных. В связи с этим необходимо рассматривать различные комбинации значений параметров bSMOTE-алгоритма с различными вариантами синтеза новых объектов. Очевидно, что реализация такого подхода к подбору значений параметров bSMOTE-алгоритма требует значительных временных затрат.

В настоящей работе предлагается алгоритм подбора значений параметров bSMOTE-алгоритма в задаче SVM-классификации несбалансированных наборов данных, обеспечивающий сокращение временных затрат (по сравнению с аналогичной реализацией на основе базового bSMOTE-алгоритма) на получение высоких значений показателей качества SVM-классификации. При этом для поиска значений параметров самого SVM-классификатора используется алгоритм роя частиц (*Particle Swarm Optimization, PSO*) [7, 8, 12-16].

Проблема несбалансированности наборов данных

При анализе набора данных на наличие проблемы несбалансированности необходимо предварительно определить, какие наборы данных следует считать несбалансированными и какие виды несбалансированности существуют.

Вообще говоря, несбалансированными можно назвать любые данные, число объектов в классах в которых не равно (даже такое соотношение числа объектов в классах, как 51:49, позволяет отнести соответствующий набор данных в категорию несбалансированных наборов). Однако обычно при констатации факта несбалансированности набора данных рассматривают соотношение числа объектов в классах, равное 10:1 и более, так как именно при подобных пропорциях наиболее ярко проявляются искажения, характерные для процесса обучения на несбалансированных наборах данных.

Несбалансированность данных может быть внутренней и внешней. Внутренняя несбалансированность зависит непосредственно от числа данных для каждого класса, то есть от их соотношения, внешняя несбалансированность возникает при неоднородности данных с точки зрения времени и способа их сбора и хранения. В настоящей работе рассматривается внутренняя несбалансированность данных.

Также несбалансированность данных может быть относительной и абсолютной [6]. Относительная несбалансированность определяется спецификой задачи, например, в задаче диагностики редких заболеваний зачастую набор данных содержит значительное число примеров, относящихся к нормальным случаям (случаям здоровья), и малое число примеров, относящихся к целевым случаям (случаям заболеваний). Абсолютная несбалансированность относится непосредственно к аспектам организации самих данных: она возникает при недостатке имеющихся данных для корректного описания целевой области.

Зачастую говорят о межклассовой и внутриклассовой несбалансированности данных. Обычно под несбалансированностью понимается именно межклассовая скошенность (англ. *skewed*) данных, то есть сильное различие между числом примеров для каждого класса в обучающей выборке. Под несбалансированностью кластеров, составляющих класс, понимают внутриклассовую скошенность [6].

В качестве алгоритмов, используемых для установления принадлежности объекта к тому или иному классу, чаще всего используют алго-

ритмы машинного обучения с учителем (англ. supervised learning), основная идея которых заключается в обучении классификатора на данных с известными классами с целью обеспечения возможности правильного определения классов новых объектов [2, 4, 7-13].

Если объекты в наборе данных не сбалансированы, существует большая вероятность того, что решение задачи классификации приведет к ошибочным результатам. Так, в частности, можно разработать классификатор, который обеспечит высокую точность классификации, равную, например, 99% в случае, когда несбалансированность в наборе данных наблюдается в соотношении 1:99, при игнорировании 1% объектов класса меньшинства. Однако такой классификатор не способен обеспечить корректную классификацию любых объектов из набора данных (в частности, классификацию объектов, относящихся к классу меньшинства).

Успешность применения алгоритма машинного обучения с учителем во многом зависит от закона организации выборки объектов, на основе которой выполняется «обучение». Несмотря на то, что большинство подобных алгоритмов требует использования сопоставимого числа примеров для каждого из классов, зачастую произвести балансировку данных не представляется возможным по нескольким причинам. Ключевыми из этих причин являются специфика целевой области (балансировка данных может понизить показатель репрезентативности набора данных) и разная цена ошибок первого и второго рода при классификации. В связи с этим возникает задача обучения классификатора на несбалансированных наборах данных (англ. skewed data). При этом предполагается, что несбалансированные наборы характеризуются асимметрией в распределении данных.

Сущность несбалансированности наборов данных, в частности, при бинарной классификации, заключается в следующем: большее число объектов исходного набора данных принадлежат одному классу («мажоритарному» классу) и гораздо меньшее число объектов относятся к другому классу («миноритарному» классу).

В соответствии с базовыми предположениями, заключенными в большинстве алгоритмов, целью обучения является максимизация доли правильных решений по отношению ко всем принятым решениям, а данные для обучения и генеральная совокупность подчиняются одному и тому же распределению [13]. Однако желание учета этих предположений при наличии несбалансированного набора данных приводит к тому, что классификатор оказывается не способен

классифицировать данные согласно алгоритму лучше, чем тривиальный классификатор, полностью игнорирующий менее представленный класс и маркирующий все объекты как принадлежащие к мажоритарному классу.

Следует отметить, что издержки ошибочной классификации различны. Так, стоимость неверной классификации примеров миноритарного класса зачастую обходится в разы дороже, чем ошибочная классификация примеров мажоритарного класса, т.к. в реально используемых наборах данных объекты миноритарного класса представляют собой редкие, но наиболее важные экземпляры.

При использовании классических подходов к задаче классификации несбалансированных наборов данных разработанный классификатор может обеспечить очень высокую точность классификации, но при этом он будет стремиться классифицировать все объекты в качестве объектов мажоритарного класса, что обычно не соответствует фактической цели исследования [6].

Очевидно, что для обеспечения высокого качества классификации с применением тех или иных классификаторов необходимо привлечение алгоритмов анализа и адекватной обработки несбалансированных наборов данных.

Принципы SVM-классификации

Для решения широкого спектра классификационных задач в различных прикладных областях успешно применяется SVM-алгоритм (Support Vector Machines, SVM) [2, 4, 7-13], являющийся одним из наиболее популярных алгоритмов машинного обучения с учителем.

SVM-алгоритм предполагает выполнение обучения, тестирования и классификации. При удовлетворительном качестве обучения и тестирования разработанный SVM-классификатор может быть применен для классификации новых объектов.

В случае задачи бинарной классификации объекты исходного набора данных разделены на два класса с метками из множества $Y = \{-1, +1\}$. При этом предполагается, что каждому объекту z_i соответствует вектор $z_i = (z_i^1, z_i^2, \dots, z_i^n)$ числовых значений в n -мерном пространстве характеристик. Тогда набор исходных данных может быть представлен множеством $\{(z_1, y_1), \dots, (z_s, y_s)\}$, в котором каждому объекту $z_i \in Z$ ($i = \overline{1, s}$; s – число объектов в исходном наборе данных) соответствует число $y_i \in Y = \{-1, +1\}$, принимающее значение «-1» или «+1», в зависимости от

того, к какому классу принадлежит объект z_i ($i = \overline{1, S}$) [12].

При разработке SVM-классификатора определяется классифицирующая функция $F: Z \rightarrow Y$, устанавливающая для объекта $z_i \in Z$ его класс принадлежности $y_i \in Y = \{-1; +1\}$.

В случае линейной разделимости классов в результате обучения SVM-классификатора определяется разделяющая гиперплоскость [12], которая может быть задана уравнением $\langle w, z \rangle + b = 0$, где w – вектор-перпендикуляр к разделяющей гиперплоскости; b – параметр, соответствующий кратчайшему расстоянию от начала координат до гиперплоскости; $\langle w, z \rangle$ – скалярное произведение векторов w и z .

Условие $-1 < \langle w, z \rangle + b < +1$ задает полосу, которая разделяет классы. Чем шире эта полоса, тем увереннее можно классифицировать объекты.

Для максимизации ширины полосы $2/\|w\|$ таким образом, чтобы внутри нее не попал ни один объект из обучающей выборки, решается задача квадратичной оптимизации [1-3]:

$$\begin{cases} \|w\|^2 = \langle w, w \rangle \rightarrow \min, \\ y_i \cdot (\langle w, z_i \rangle + b) \geq 1, \quad i = \overline{1, S}, \end{cases} \quad (1)$$

где S ($s > S$) – число объектов в обучающей выборке; $i = \overline{1, S}$.

В случае линейной неразделимости классов с учетом теоремы Куна-Таккера задача построения разделяющей гиперплоскости сводится к задаче квадратичного программирования, содержащей только двойственные переменные λ_i ($i = \overline{1, S}$) [12]:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^S \lambda_i + \\ + \frac{1}{2} \cdot \sum_{i=1}^S \sum_{\tau=1}^S \lambda_i \cdot \lambda_\tau \cdot y_i \cdot y_\tau \cdot \kappa(z_i, z_\tau) \rightarrow \min_{\lambda}, \\ \sum_{i=1}^S \lambda_i \cdot y_i = 0, \\ 0 \leq \lambda_i \leq C, \quad i = \overline{1, S}, \end{cases} \quad (2)$$

где λ_i – двойственная переменная; z_i – объект из обучающей выборки; y_i – число (-1 или +1), характеризующее классовую принадлежность объекта z_i из обучающей выборки; $\kappa(z_i, z_\tau)$ – спрямляющая функция ядра (kernel function); C – параметр регуляризации ($C > 0$); S – количество объектов в обучающей выборке; $i = \overline{1, S}$.

В результате обучения SVM-классификатора определяются опорные векторы, являющиеся векторами характеристик тех объектов z_i из обучающей выборки, для которых значения соответствующих им двойственных переменных λ_i отличны от нуля ($\lambda_i \neq 0$). Опорные векторы находятся ближе всего к разделяющей гиперплоскости и несут всю информацию о разделении классов.

В случае линейной неразделимости классов используется некоторая функция ядра и проводится классификация объектов в пространстве более высокой размерности. В качестве функций ядра могут быть использованы, в частности, полиномиальная, радиальная базисная, сигмоидная функции [9, 12]. При этом решение задачи выбора типа функции ядра и подбора оптимальных значений ее параметров с целью разработки искомого SVM-классификатора связано с существенными временными затратами. В настоящей работе при разработке SVM-классификатора используется радиальная базисная функция ядра $\kappa(z_i, z_\tau) = \exp(-\langle z_i - z_\tau, z_i - z_\tau \rangle / (2 \cdot \sigma^2))$, где σ ($\sigma > 0$) – параметр, значение которого наряду со значением параметра регуляризации C определяются таким образом, чтобы обеспечить высокое качество классификационных решений.

В результате обучения SVM-классификатора определяется классифицирующая функция, устанавливающая для произвольного объекта z его класс принадлежности с меткой «-1» или «+1» [9, 12]:

$$F(z) = \text{sign} \left(\sum_{i=1}^S \lambda_i \cdot y_i \cdot \kappa(z_i, z) + b \right). \quad (3)$$

Алгоритм подбора значений параметров bSMOTE-алгоритма

В настоящее время для решения проблемы несбалансированности набора данных применяются различные стратегии сэмпинга [1-6], реализующие принципы андэрсэмпинга (англ. undersampling) и овэрсэмпинга (англ. oversampling). При использовании андэрсэмпинга из обучающей выборки, формируемой на основе исходного набора данных, удаляют некоторое число примеров мажоритарного класса, а при использовании овэрсэмпинга в обучающую выборку, формируемую на основе исходного набора данных, добавляют некоторое число примеров миноритарного класса.

Стратегии сэмпинга можно разделить на случайные и специальные.

Стратегии случайного сэмпинга добавляют в процессе овэрсэмпинга в обучающую выборку

ку новые примеры посредством копирования случайно выбранных примеров миноритарного класса или в процессе андерсэмплинга из обучающей выборки случайно выбранные примеры мажоритарного класса. В результате общее число примеров в миноритарном/мажоритарном классе увеличивается/уменьшается на величину, равную числу скопированных/удалённых примеров, при этом меняется и распределение примеров между классами.

Стратегии специального сэмпинга противостоят стратегиям случайного сэмпинга, реализуя некоторые алгоритмы восстановления баланса классов, основанные на учете тех или иных закономерностей, наблюдаемых в рамках используемых обучающих выборок. Одним из перспективных специальных алгоритмов сэмпинга является алгоритм синтетического сэмпинга с генерацией данных – SMOTE-алгоритм (Synthetic Minority Oversampling Technique) [2-5], реализующий увеличение числа объектов миноритарного класса. Данный алгоритм создаёт синтетические объекты миноритарного класса с учетом степени сходства в пространстве характеристик между уже существующими объектами, используя принципы алгоритма k ближайших соседей (k NN-алгоритма; k Nearest Neighbors Algorithm) [12]. При этом синтетические объекты «похожи» на объекты, уже имеющиеся в миноритарном классе, но при этом не дублируют их.

Пусть Z^1 – набор объектов обучающей выборки, а Z_{min}^1 – набор объектов миноритарного класса ($Z_{min}^1 \in Z^1$, $Z = Z^1 \cup Z^2$, $Z^1 \cap Z^2 = \emptyset$, где Z^2 – набор объектов тестовой выборки).

Пусть n – размерность пространства характеристик, S_{min} – число объектов в миноритарном классе. Тогда каждому объекту z_i миноритарного класса в n -мерном пространстве характеристик можно поставить в соответствие вектор $z_i = (z_i^1, z_i^2, \dots, z_i^n)$ ($i = \overline{1, S_{min}}$).

Реализация SMOTE-алгоритма в простейшем случае может быть описана следующей последовательностью шагов.

Шаг 1. Для каждого объекта z_i миноритарного класса найти k объектов-ближайших соседей из миноритарного класса (то есть принадлежащих набору Z_{min}^1) с использованием k NN-алгоритма, при этом в качестве метрики расстояния может использоваться евклидова метрика.

Шаг 2. Для каждого объекта z_i миноритарного класса случайным образом выбрать одного

из k ближайших соседей, найденных на шаге 1: $z_K = (z_K^1, z_K^2, \dots, z_K^n)$, ($1 \leq K \leq k$) и синтезировать новый объект миноритарного класса: $t_i = z_i + (z_i - z_K) \cdot rand$, где $rand$ – случайное число из отрезка $[0, 1]$.

В результате реализации данного алгоритма к S_{min} объектам миноритарного класса будет добавлено еще S_{min} синтезированных объектов, описываемых в n -мерном пространстве характеристик векторами $t_i = (t_i^1, t_i^2, \dots, t_i^n)$ ($i = \overline{1, S_{min}}$).



Рисунок 1 – Схема генерации синтетических объектов в SMOTE-алгоритме

В зависимости от степени несбалансированности исходного набора данных может быть синтезировано необходимое число новых объектов миноритарного класса.

На рисунке 1 представлена схема генерации синтетических объектов в SMOTE-алгоритме в пространстве D-2.

В настоящее время известен ряд модификаций SMOTE-алгоритма, к которым относится и bSMOTE-алгоритм. В [5] рассматриваются две версии bSMOTE-алгоритма: *borderline-SMOTE1* и *borderline-SMOTE2*-алгоритмы, реализующие увеличение числа объектов миноритарного класса таким образом, что для синтеза новых объектов используются только пограничные объекты миноритарного класса, то есть объекты, лежащие вблизи границы классов. При этом, если в версии *borderline-SMOTE1*-алгоритма синтезирование новых объектов для пограничных объектов миноритарного класса реализуется на основе ближайших соседей из миноритарного класса обучающей выборки, то в версии *borderline-SMOTE2*-алгоритма синтезирование новых объектов для пограничных объектов миноритарного класса реализуется на основе ближайших соседей из всей обучающей выборки [5].

В настоящей работе был использован *borderline-SMOTE1*-алгоритм, далее именуемый как bSMOTE-алгоритм.

В bSMOTE-алгоритме перед созданием синтетических объектов осуществляется оценка типа каждого объекта миноритарного класса, при этом проверяется, является ли объект надежным объектом (safe), шумовым объектом (выбросом) (noise) или пограничным (danger). Для этого для каждого объекта миноритарного класса выполняется выявление m объектов – ближайших соседей – из всей обучающей выборки исходного набора данных.

Надежные объекты – объекты, наиболее близкие к эталонным объектам или же сами эталонные объекты, под которыми понимают объекты, плотно окруженные объектами своего класса и являющиеся наиболее типичными его представителями [5].

Шумовые объекты (выбросы) – объекты, плотно окруженные объектами другого класса, которые обычно классифицируются неверно. Они могут возникать из-за грубых ошибок или пропусков в исходных данных, а также по причине отсутствия важной информации, которая позволила бы отнести эти объекты к правильно-му классу. Число таких объектов невелико [5].

Пограничные объекты – объекты, лежащие вблизи границы классов. Классификация таких объектов неустойчива по сравнению с объектами, располагающимися далеко от границы классов, в том смысле, что малые изменения состава обучающей выборки могут изменять их классификацию [5].

В bSMOTE-алгоритме выполняется генерация новых синтетических объектов вблизи пограничных объектов миноритарного класса с целью уменьшения вероятности их ошибочной классификации.

Таким образом, в bSMOTE-алгоритме производится увеличение числа только пограничных объектов миноритарного класса, в то время как SMOTE-алгоритм предполагает увеличение числа всех объектов миноритарного класса. При этом результаты экспериментальных исследований подтверждают явное преимущество bSMOTE-алгоритма по сравнению со SMOTE-алгоритмом в смысле в контексте обеспечения более высокой точности классификации пограничных объектов миноритарного класса.

Реализация bSMOTE-алгоритма может быть описана следующей последовательностью шагов.

Шаг 1. Для каждого объекта z_i миноритарного класса найти m объектов-ближайших соседей из всей обучающей выборки (то есть принадлежащих набору Z^1) с использованием k NN-алгоритма. Определить число объектов m' ($0 \leq m' \leq m$) мажоритарного класса из числа m

ближайших соседей объекта z_i миноритарного класса.

Шаг 2. Если $m' = m$, то есть все m объектов-ближайших соседей являются объектами мажоритарного класса, признать объект z_i миноритарного класса шумовым объектом (выбросом) и не использовать его на следующих шагах алгоритма.

Если $m/2 \leq m' < m$, то есть для объекта z_i миноритарного класса число объектов-ближайших соседей из мажоритарного класса больше, чем число объектов-ближайших соседей из миноритарного класса, считать, что классификация объекта z_i неустойчива и включить объект z_i в набор пограничных объектов $DANGER \subseteq Z_{min}^1$.

Пусть S_{min}^D – число объектов в наборе $DANGER$ ($0 \leq S_{min}^D \leq S_{min}$). Тогда каждому пограничному объекту z_i миноритарного класса в n -мерном пространстве характеристик можно поставить в соответствие вектор $z'_i = (z_i^1, z_i^2, \dots, z_i^n)$ ($i = \overline{1, S_{min}^D}$).

Если $0 \leq m' < m/2$, считать объект z_i миноритарного класса надежным, то есть находящимся далеко от границы классов и не использовать его на следующих шагах алгоритма.

Шаг 3. Определить для каждого пограничного объекта z'_i миноритарного класса k ближайших соседей из Z_{min}^1 .

Шаг 4. Для каждого пограничного объекта z'_i миноритарного класса случайным образом выбрать K' ($1 \leq K' \leq k$) ближайших соседей из числа k объектов, найденных на шаге 3: $q_1 = (z_1^1, z_1^2, \dots, z_1^n), \dots, q_j = (z_j^1, z_j^2, \dots, z_j^n), \dots, q_{K'} = (z_{K'}^1, z_{K'}^2, \dots, z_{K'}^n)$ ($j = \overline{1, K'}$).

Синтезировать для каждого пограничного объекта z'_i миноритарного класса K' новых объектов миноритарного класса. $t'_{ij} = z'_i + (z'_i - q_j) \cdot rand_j$, где $rand_j$ ($j = \overline{1, K'}$) – случайное число из отрезка $[0, 1]$.

В результате реализации данного алгоритма к S_{min} объектам миноритарного класса будет добавлено еще $K' \times S_{min}^D$ синтезированных объектов, описываемых в n -мерном пространстве характеристик векторами $t'_{ij} = (t_{ij}^1, t_{ij}^2, \dots, t_{ij}^n)$ ($i = \overline{1, S_{min}^D}$, $j = \overline{1, K'}$).

На рисунке 2 продемонстрирована работа SMOTE- и bSMOTE-алгоритмов в пространстве D-2 на тестовом примере при решении задачи

восстановления баланса классов (число объектов в исходном наборе данных – 3000, число характеристик – 2, число объектов мажоритарного класса – 2700; число объектов миноритарного класса – 300; число объектов мажоритарного класса после восстановления баланса – 2700; число объектов миноритарного класса после восстановления баланса – 2230; общее число объектов после восстановления – 4930).

Следует отметить, что при восстановлении баланса классов алгоритм балансировки может стремиться к получению как точного, так и приближенного совпадения числа объектов в классах.

Существуют различные подходы к определению числа генерируемых новых синтетических объектов. В настоящей работе число синтетических объектов в SMOTE – и bSMOTE-алгоритмах определялось значением разности числа объектов мажоритарного класса S_{max} и числа объектов миноритарного класса S_{min} : $t_i = (t_i^1, t_i^2, \dots, t_i^n)$ ($i=1, S_{max} - S_{min}$).

Очевидно, что целесообразно подбирать такие значения параметров bSMOTE-алгоритма, использование которых обеспечит лучший вариант восстановления сбалансированности набора данных. В связи с этим необходимо рассматривать различные комбинации значений параметров bSMOTE-алгоритма с различными вариантами синтеза новых объектов. Очевидно, что

реализация такого подхода к подбору значений параметров bSMOTE-алгоритма требует значительных временных затрат.

В настоящей работе реализован подбор оптимальных значений двух параметров bSMOTE-алгоритма: k – число ближайших соседей, использованных для создания синтетических образцов; m – число ближайших соседей, которое используется для определения находится ли объект миноритарного класса на границе классов.

Предлагаемый алгоритм подбора значений параметров bSMOTE-алгоритма может быть описан следующей последовательностью шагов.

Шаг 1. Сгенерировать пары (k_i, m_j) на основе целочисленных значений параметров в диапазонах $[k_{min}, k_{max}]$ и $[m_{min}, m_{max}]$ ($i=1, k_{max}-k_{min}+1$; $j=1, m_{max}-m_{min}+1$).

Шаг 2. Разработать для каждой пары (k_i, m_j) r SVM-классификаторов, используя bSMOTE-алгоритм, в предположении о равновероятной пригодности каждой пары (k_i, m_j) .

Шаг 3. Оценить качество классификации разработанных SVM-классификаторов и сохранить полученные модели SVM-классификаторов. Если достигнуто максимальное число итераций, определить «лучший» SVM-классификатор и завершить работу алгоритма. В противном случае перейти к шагу 4.

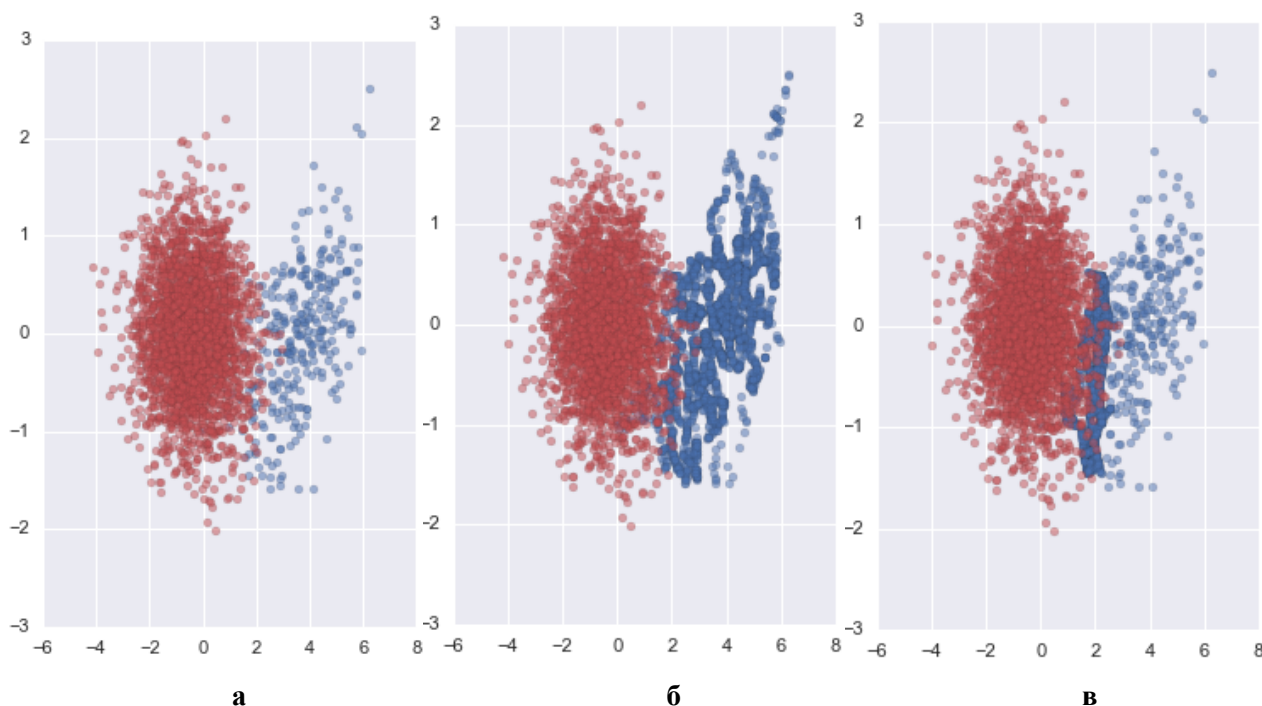


Рисунок 2 – Восстановление баланса классов:
а – исходный набор данных, **б** – набор данных после применения SMOTE-алгоритма;
в – набор данных после применения bSMOTE-алгоритма

Шаг 4. Оценить среднее качество классификации SVM-классификаторов, используя, например, показатель F -меры [8, 12] для каждой пары (k_i, m_j) . Изменить вероятности применения bSMOTE – алгоритма для каждой пары (k_i, m_j) : увеличить вероятность для лучшей пары (k_i, m_j) (с максимальным значением среднего качества классификации), то есть увеличить количество запусков bSMOTE-алгоритма для лучшей пары, и уменьшить вероятности для других пар (k_i, m_j) , то есть сократить количество запусков bSMOTE-алгоритма для других пар. Разработать для каждой пары (k_i, m_j) SVM-классификаторы, используя bSMOTE-алгоритма в соответствии с новыми вероятностями. Перейти к шагу 3.

На шаге 4 предлагается использовать следующий подход для оценки среднего качества классификации SVM-классификаторов для каждой пары (k_i, m_j) :

– для каждой пары (k_i, m_j) найти общее число SVM-классификаторов N_{ij}^g , полученных за текущее число итераций g предлагаемого алгоритма;

– для каждой пары (k_i, m_j) найти общую сумму значений используемого показателя качества классификации SVM-классификаторов S_{ij}^g (в данном исследовании применялся показатель F -меры), полученных за текущее число итераций g предложенного алгоритма;

– для каждой пары (k_i, m_j) найти отношение S_{ij}^g / N_{ij}^g и использовать его как среднее качество классификации SVM-классификаторов для пары (k_i, m_j) .

Необходимо отметить, что в процессе реализации предлагаемого алгоритма генерируются разные сбалансированные обучающие выборки ввиду использования генератора случайных чисел для каждой пары (k_i, m_j) , поэтому разрабо-

танные SVM-классификаторы будут отличаться друг от друга.

Предлагаемый алгоритм одновременно реализует подбор значений параметров bSMOTE-алгоритма для последующего восстановления сбалансированности классов и разработку SVM-классификатора, характеризующегося высоким качеством классификации данных.

Экспериментальные исследования

Для апробации предложенного алгоритма подбора значений параметров bSMOTE-алгоритма при решении задачи SVM-классификации были использованы реальные медицинские данные из репозитория машинного обучения UCI [7-9]. В частности, рассматривались набор данных «Heart» ([https://archive.ics.uci.edu/ml/datasets/Statlog+\(Heart\)/](https://archive.ics.uci.edu/ml/datasets/Statlog+(Heart)/)), набор данных «Hepatitis» (<https://archive.ics.uci.edu/ml/datasets/Hepatitis/>), набор данных «WDBC» (Breast Cancer Wisconsin (Original) Data Set) ([https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Original\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original))). В этих наборах к классу с меткой «+1» (то есть к положительному классу) относятся данные, соответствующие нормальному (здоровым) исходам, а к классу с меткой «-1» (то есть к отрицательному классу) относятся данные, соответствующие аномальному (нездоровым) исходам. Подробные характеристики представленных наборов данных приведены в таблице 1.

В таблице 1 значения показателя несбалансированности классов $Ratio$ рассчитаны по формуле:

$$Ratio = 1 - \frac{S_{min}}{S_{max}}, \quad (4)$$

где S_{min} – число объектов миноритарного класса; S_{max} – число объектов мажоритарного класса.

Как видно из таблицы 1, рассматриваемые наборы данных имеют разные значения показателя несбалансированности классов $Ratio$. При этом набор данных «Hepatitis» является самым несбалансированным ($Ratio = 0.74$). Два других набора данных характеризуются меньшим уровнем несбалансированности (при этом набор данных «Heart» является практически сбалансированным).

Таблица 1 – Характеристика наборов данных

| Набор данных | Число объектов в наборе данных | Число характеристик | Число объектов положительно-го класса (метка «+1») | Число объектов отрицательного класса (метка «-1») | Показатель несбалансированности |
|--------------|--------------------------------|---------------------|--|---|---------------------------------|
| «Heart» | 270 | 13 | 120 | 150 | 0.2 |
| «Hepatitis» | 155 | 19 | 123 | 32 | 0.74 |
| «WDBC» | 699 | 10 | 458 | 241 | 0.47 |

Программная реализация алгоритма подбора значений параметров bSMOTE-алгоритма при решении задачи SVM-классификации выполнена с использованием языка программирования Python 3.5. При этом при разработке SVM-классификаторов использовалась радиальная базисная функция ядра [7-9, 12].

В результате применения bSMOTE-алгоритма для восстановления баланса классов число объектов миноритарного класса увеличивалось таким образом, что для набора данных «Heart» общее число объектов после балансировки стало равным 304 (154 объектов класса с меткой «+1» и 150 объектов класса с меткой «-1»); для набора данных «Hepatitis» – 223 (123 объекта класса с меткой «+1» и 100 объекта класса с меткой «-1»); для набора данных «WDBC» – 872 (458 объектов класса с меткой «+1» и 414 объектов класса с меткой «-1»).

В таблице 2 представлены сравнительные результаты экспериментальных расчетов, полученные при разработке SVM-классификаторов на основе несбалансированных наборов данных из таблицы 1 без применения сэмпинга (тип алгоритма: 1) и с применением сэмпинга на основе предложенного в данной работе алгоритма (тип алгоритма: 2). При этом поиск значений параметров SVM-классификаторов (значения параметра регуляризации и значения параметра радиальной базисной функции ядра) во всех экспериментах осуществлялся с применением PSO-алгоритма [7, 8, 12-16]. Первый столбец в таблице 2 определяет наименование набора данных и тип используемого алгоритма (разделены знаком «/»).

Для оценки качества разработанных SVM-классификаторов применялись такие показатели качества классификации [8], как: показатель об-

щей точности (*Accuracy, Acc*), называемый также показателем общей доли правильных ответов (*Overall Success Rate, OSR*); показатель чувствительности (*Sensitivity, Se*), называемый также показателем полноты (*Recall, Re*); показатель специфичности (*Specificity, Sp*); а также показатель сбалансированной *F*-меры (*F-measure, F₁*), рассчитываемые соответственно по формулам:

$$OSR = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5)$$

$$Se = \frac{TP}{TP + FN}, \quad (6)$$

$$Sp = \frac{TN}{TN + FP}, \quad (7)$$

$$F_1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re}. \quad (8)$$

где *TP* – число истинно положительных наблюдений; *TN* – число истинно отрицательных наблюдений; *FP* – число ложноположительных наблюдений («ложных обнаружений», ошибка II рода); *FN* – число ложноотрицательных наблюдений («ложных пропусков», ошибка I рода);

$$Pr = \frac{TP}{TP + FP}; Re = Se.$$

Результаты расчетов, приведенные в таблице 2, получены при применении следующих значений параметров алгоритма типа 2: число пар (k_i, m_j) равно 4, число запусков алгоритма равно 3, начальное число классификаторов для каждой пары (k_i, m_j) равно 5. Диапазоны поиска значений параметров k_i и m_j определяются отрезком [1, 30]. При поиске значений параметров SVM-классификаторов число запусков PSO-алгоритма было установлено равным 5.

Таблица 2 – Результаты экспериментальных расчетов

| Набор данных/ Тип алгоритма | Размер обучающей/ тестовой выборки | Ошибки | | | | | | Общая точность (%) | Чувствительность (%) | Специфичность (%) | <i>F</i> -мера |
|--------------------------------|---------------------------------------|----------------------------------|----------------------------------|--------------------|----------------------------|----------------------------------|--------------------|--------------------|----------------------|-------------------|----------------|
| | | На обучающей выборке | | | На тестовой выборке | | | | | | |
| | | Положительный класс (метка «+1») | Отрицательный класс (метка «-1») | Общее число ошибок | Положительный класс («+1») | Отрицательный класс (метка «-1») | Общее число ошибок | | | | |
| «Heart»/1 | 216/54 | 7 | 2 | 9 | 2 | 6 | 8 | 93.70 | 92.50 | 94.67 | 0.93 |
| «Heart»/2 | 240/60 | 2 | 2 | 4 | 5 | 4 | 9 | 95.72 | 95.45 | 96.00 | 0.96 |
| «Hepatitis»/1 | 124/31 | 0 | 0 | 0 | 3 | 8 | 11 | 92.90 | 97.56 | 75.00 | 0.96 |
| «Hepatitis»/2 | 196/50 | 0 | 0 | 0 | 0 | 2 | 2 | 99.10 | 100.00 | 98.00 | 0.99 |
| «WDBC»/1 | 559/140 | 0 | 0 | 0 | 4 | 1 | 5 | 99.28 | 99.13 | 99.59 | 0.99 |
| «WDBC»/2 | 732/184 | 1 | 0 | 1 | 3 | 0 | 3 | 99.54 | 99.13 | 100.00 | 0.99 |

В PSO-алгоритме число частиц в рое было установлено равным 300. В дальнейшем с целью улучшения поисковых характеристик PSO-алгоритма целесообразно использовать гибридную версию PSO-алгоритма, реализующую применение алгоритма поиска по сетке – DOE-алгоритма (Design Of Experiment algorithm) [7, 8, 12].

Разделение исходного набора данных на обучающую и тестовую выборки производилось так, что размер тестовой выборки составлял 20% от размера исходного набора данных.

Время поиска оптимальных значений параметров bSMOTE-алгоритма составило: 2395.28 секунды для набора данных «Heart», 1789.73 секунды для набора данных «Hepatitis» и 10651.82 секунды для набора данных «WDBC». При этом оптимальные значения для пар параметров bSMOTE-алгоритма составили: (15, 9) для набора данных «Heart», (20, 23) для набора данных «Hepatitis» и (2, 8) для набора данных «WDBC».

Результаты экспериментальных исследований, приведенные в таблице 2, показывают, что предложенный алгоритм подбора значений параметров bSMOTE-алгоритма при решении задачи SVM-классификации обеспечивает повышение значений показателей качества классификации. В частности, для набора данных «Hepatitis», имеющего самое большое значение показателя несбалансированности (таблица 1) значение показателя общей точности увеличилось с 92.90% до 99.10%, значение показателя чувствительности – с 97.56% до 100.00%, значение показателя специфичности – с 75.00% до 98.00%, а показателя F -меры – с 0.96 до 0.99).

Заключение

Полученные результаты экспериментальных исследований демонстрируют повышение качества SVM-классификации несбалансированных наборов данных с применением предложенного алгоритма подбора значений параметров bSMOTE-алгоритма при решении задачи SVM-классификации. При этом предложенный алгоритм позволяет существенно сократить временные затраты на подбор оптимальных значений параметров bSMOTE-алгоритма, который при стандартном применении bSMOTE-алгоритма предполагал бы многократный запуск bSMOTE-алгоритма для различных пар значений искомых параметров с целью выбора комбинации значений параметров, обеспечивающей разработку SVM-классификатора, характеризующегося высоким качеством классификации данных.

В ходе дальнейших исследований предполагается реализовать многоцелевую версию алгоритма подбора значений параметров bSMOTE-

алгоритма при решении задачи SVM-классификации.

Библиографический список

1. **Садов М. А.** Исследование методов классификации текстов для несбалансированных данных // *Полиматис*. 2016. № 2. С. 28-41.
2. **Клюева И. А.** Исследование аспектов применимости стратегий сэмпинга для решения проблемы несбалансированности структур данных // *Новые информационные технологии в научных исследованиях*. Рязань: Рязанский государственный радиотехнический университет, 2016. С. 198-199.
3. **Chawla N., Bowyer K., Hall L., Kegelmeyer W.** SMOTE: Synthetic Minority Over-sampling Technique // *Journal of Artificial Intelligence Research*. 2002. Vol. 16. pp. 321-357.
4. **Demidova L., Klyueva I.** Improving the Classification Quality of the SVM Classifier for the Imbalanced Datasets on the Base of Ideas the SMOTE Algorithm // *ITM Web of Conferences*. 2017. Vol. 10. 02002.
5. **Han H., Wen-Yuan W., Bing-Huan M.** Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning // *Advances in intelligent computing*. 2005. Vol. 2. No. 5. pp. 878-887.
6. **He H., Ma Y.** Imbalanced Learning: Foundations, Algorithms, and Applications. Wiley-IEEE Press, 2013. 216 p.
7. **Демидова Л. А., Клюева И. А.** Разработка и исследование гибридных версий алгоритма роя частиц на основе алгоритмов поиска по сетке // *Вестник Рязанского государственного радиотехнического университета*. 2016. № 57. С. 105-116.
8. **Демидова Л. А., Клюева И. А.** Разработка SVM-классификатора с применением гибридных версий алгоритма роя частиц на основе поиска по сетке // *Cloud of science*. 2017. Т. 3. № 4. С. 528-547.
9. **Демидова Л. А., Соколова Ю. С.** Аспекты применения алгоритма роя частиц в задаче разработки SVM-классификатора // *Вестник Рязанского государственного радиотехнического университета*. 2015. № 53. С. 84-92.
10. **Batuwita R., Palade V.** FSVM-CIL: fuzzy support vector machines for class imbalance learning // *IEEE Transactions on Fuzzy Systems*. 2010. Vol. 18. No. 3. pp. 558-571.
11. *Bio-Inspired Credit Risk Analysis. Computational Intelligence with Support Vector Machines / Ed. by Lean Yu, Shouyang Wang, Kin Keung Lai, Ligang Zhou.* Springer-Verlag Berlin Heidelberg, 2008. 244 p.
12. **Demidova L., Klyueva I., Sokolova Y., Stepanov N., Tyart N.** Intellectual Approaches to Improvement Of the Classification Decisions Quality On the Base Of the SVM Classifier // *Procedia Computer Science*. 2017. Vol. 103. pp. 222-230.
13. **Vapnik V.** *Statistical Learning Theory*. New York: John Wiley & Sons, 1998. 732 p.
14. **Карпенко А. П.** *Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой*. М.: Изд-во МГТУ им. Н. Э. Баумана, 2014. 446 с.

15. **Demidova L., Klyueva I., Pylkin A.** The Study of Characteristics of the Hybrid Particle Swarm Algorithm in Solution of the Global Optimization Problem // 5th Mediterranean Conference on Embedded Computing

(MECO). 2016. pp. 322–325.

16. **Jun Sun, Choi-Hong Lai, Xiao-Jun Wu.** Particle Swarm Optimisation: Classical and Quantum Perspectives. CRC Press, 2011. 419 p.

UDC 004.855.5

SEARCH ALGORITHM OF THE PARAMETERS VALUES OF THE BSMOTE-ALGORITHM IN THE PROBLEM OF THE SVM-CLASSIFICATION BASED ON THE IMBALANCED DATASETS

L. A. Demidova, PhD (technical sciences), full professor, RSREU, Ryazan; liliya.demidova@rambler.ru
I. A. Klyueva, post-graduate student, RSREU, Ryazan; i.aleschenko@yandex.ru.

The problem of SVM (Support Vector Machine) classification based on the unbalanced data sets applied to generate train sets, using the synthetic algorithm sampling – bSMOTE algorithm (borderline Synthetic Minority Oversampling Technique algorithm) has been considered. The aim of this work is the development of the search algorithm of bSMOTE-algorithm parameters values in the problem of SVM classification of unbalanced datasets, providing the reducing of time expenditures for the development of SVM classifier, characterized by high quality of data classification. The search of SVM classifier parameters values has been implemented with the use of PSO algorithm (Particle Swarm Optimization algorithm). The results of experimental studies confirming the feasibility of the search algorithm of bSMOTE-algorithm parameters values in the problem of SVM classification of unbalanced datasets have been given.

Key words: *imbalanced data, sampling, bSMOTE-algorithm, classification, SVM classifier, radial basis kernel function, PSO algorithm.*

DOI: 10.21667/1995-4565-2017-61-3-67-77

References

1. **Sadov M. A.** Issledovanie metodov klassifikacii tekstov dlja nesbalansirovannyh dannyh. Polimatis. 2016, no. 2, pp. 28–41 (in Russian).

2. **Klyueva I. A.** Issledovanie aspektov primenimosti strategij sjemplinga dlja reshenija problemy nesbalansirovannosti struktur dannyh. Novye informacionnye tehnologii v nauchnyh issledovanijah. Rjazan': Rjazanskij gosudarstvennyj radiotekhnicheskij universitet, 2016, pp. 198–199 (in Russian).

3. **Chawla N., Bowyer K., Hall L., Kegelmeyer W.** SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research. 2002, vol. 16, pp. 321–357.

4. **Demidova L., Klyueva I.** Improving the Classification Quality of the SVM Classifier for the Imbalanced Datasets on the Base of Ideas the SMOTE Algorithm. ITM Web of Conferences. 2017, vol. 10, 02002.

5. **Han H., Wen-Yuan W., Bing-Huan M.** Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Advances in intelligent computing. 2005, vol. 2, no. 5, pp. 878–887.

6. **He H., Ma Y.** Imbalanced Learning: Foundations, Algorithms, and Applications, Wiley-IEEE Press, 2013, 216 p.

7. **Demidova L. A., Klyueva I. A.** Razrabotka i isledovanie gibridnyh versij algoritma roja chastic na osnove algoritmov poiska po setke. Vestnik Rjazanskogo gosudarstvennogo radiotekhnicheskogo universiteta. 2016, no. 57, pp. 105–116 (in Russian).

8. **Demidova L. A., Klyueva I. A.** Razrabotka SVM-klassifikatora s primeneniem gibridnyh versij algoritma roja chastic na osnove poiska po setke. Cloud of

science. 2017, vol. 3, no. 4, pp. 528–547 (in Russian).

9. **Demidova L. A., Sokolova Yu. S.** Aspekty primenenija algoritma roja chastic v zadache razrabotki SVM-klassifikatora. Vestnik Rjazanskogo gosudarstvennogo radiotekhnicheskogo universiteta. 2015, no. 53, pp. 84–92 (in Russian).

10. **Batuwita R., Palade V.** FSVM-CIL: fuzzy support vector machines for class imbalance learning. IEEE Transactions on Fuzzy Systems. 2010, vol. 18, no. 3, pp. 558–571.

11. Bio-Inspired Credit Risk Analysis. Computational Intelligence with Support Vector Machines / Ed. by **Lean Yu, Shouyang Wang, Kin Keung Lai, Ligang Zhou**, Springer-Verlag Berlin Heidelberg, 2008, 244 p.

12. **Demidova L., Klyueva I., Sokolova Y., Stepanov N., Tyart N.** Intellectual Approaches to Improvement Of the Classification Decisions Quality On the Base Of the SVM Classifier. Procedia Computer Science. 2017, vol. 103, pp. 222–230.

13. **Vapnik V.** Statistical Learning Theory, New York, John Wiley & Sons, 1998, 732 p.

14. **Karpenko A. P.** Sovremennye algoritmy poiskovoj optimizacii. Algoritmy, vdohnovlennye prirodoj (The modern algorithms of the search optimization. The algorithms inspired by nature), Moscow, Izd-vo MGTU im. N. Je. Baumana, 2014, 446 p. (in Russian).

15. **Demidova L., Klyueva I., Pylkin A.** The Study of Characteristics of the Hybrid Particle Swarm Algorithm in Solution of the Global Optimization Problem. 5th Mediterranean Conference on Embedded Computing (MECO). 2016, pp. 322–325.

16. **Jun Sun, Choi-Hong Lai, Xiao-Jun Wu.** Particle Swarm Optimisation: Classical and Quantum Perspectives, CRC Press, 2011, 419 p.