УДК 004.855.5

РАЗРАБОТКА ДВУХУРОВНЕВОГО КЛАССИФИКАТОРА СЛОЖНООРГАНИЗОВАННЫХ МНОГОМЕРНЫХ ДАННЫХ БОЛЬШИХ ОБЪЕМОВ

Л. А. Демидова, д.т.н., профессор кафедры ВПМ РГРТУ, liliya.demidova@rambler.ru
Ю. С. Соколова, старший преподаватель РГРТУ, JuliaSokolova62@yandex.ru

Рассматривается задача классификации сложноорганизованных многомерных данных больших объемов, характерная для различных социально-экономических систем. ***Целью работы*** *является повышение точности классификации сложноорганизованных многомерных данных больших объемов посредством разработки двухуровневого классификатора на основе SVM-алгоритма при приемлемых вычислительных затратах. На первом уровне двухуровневого классификатора создается группа SVM-классификаторов, обучающие выборки для которых формируются на основе исходного экспериментального набора данных. На втором уровне двухуровневого классификатора разрабатывается SVM-классификатор с использованием модифицированного алгоритма роя частиц, обучающая выборка для которого формируется на основе набора опорных векторов, полученных на первом уровне двухуровневого классификатора. Приведенные результаты экспериментальных исследований подтверждают эффективность применения предлагаемого двухуровневого классификатора в задаче классификации сложноорганизованных многомерных данных больших объемов.*

Ключевые слова*: SVM-классификатор, опорные векторы, тип функции ядра, параметры функции ядра, параметр регуляризации, алгоритм роя частиц.*

DOI: 10.21667/1995-4565-2016-56-2-71-82

Введение

Ведущая роль в системах интеллектуального анализа данных отводится созданию приложений, ориентированных на обработку сложноорганизованных многомерных данных больших объемов, в том числе больших данных (Big Data), являющихся неотъемлемой частью экономики, банковской деятельности, производства, маркетинга, телекоммуникаций, веб-аналитики, медицины и пр. Стремительное увеличение количества информации, которая должна обрабатываться приложениями в различных прикладных областях, служит убедительным доводом в пользу проведения исследований и разработок, направленных на создание как аппаратных, так и программных средств, предназначенных для решения задач обработки и анализа сложноорганизованных многомерных данных больших объемов.

В системах интеллектуального анализа особое внимание уделяется вопросам классификации данных. В связи с этим необходимость в разработке адекватного инструментария, позволяющего определить класс принадлежности произвольного объекта с максимальной точностью, ощущается довольно остро. Предложенный В.Н. Вапником [1] SVM-алгоритм (Support Vector Machines, SVM) в настоящее время успешно применяется для решения широкого спектра классификационных задач в различных прикладных областях [2]. Данный алгоритм осуществляет обучение по прецедентам («обучение с учителем») и входит в группу граничных алгоритмов и методов классификации.

SVM-алгоритм реализует построение бинарного SVM-классификатора, осуществляя с помощью специальной функции, называемой ядром, перевод векторов характеристик классифицируемых объектов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве [1]. При этом по обеим сторонам разделяющей гиперплоскости строятся две параллельные гиперплоскости, задающие границы классов и находящиеся на максимально возможном расстоянии друг от друга. Предполагается, что чем больше расстояние между этими параллельными гиперплоскостями, тем меньше средняя ошибка SVM-классификатора [1, 3]. Векторы характеристик классифицируемых объектов, ближайшие к параллельным гиперплоскостям, называются опорными векторами.

SVM-классификатор предполагает выполнение процедур обучения, тестирования и классификации. При приемлемом качестве процедур обучения и тестирования SVM-классификатор может быть применен для классификации новых объектов. Для оценки качества классификации могут использоваться показатели точности и полноты классификации и др. [4].

Заданные при построении SVM-классификатора тип функции ядра, значения параметров функции ядра и значение параметра регуляризации влияют на качество классификации данных. Значения параметров SVM-классификатора будем считать оптимальными, если достигнута высокая точность классификации: количество ошибок на обучающем и тестовом наборах минимально, причем количество ошибок обученного SVM-классификатора на объектах тестовой выборки не сильно отличается от количества ошибок на обучающей выборке (во избежание «переобучения» SVM-классификатора).

Подбор оптимальных значений параметров SVM-классификатора возможен с помощью поисковых алгоритмов стохастической оптимизации, основанных на идее о возможности решения задач оптимизации посредством моделирования поведения групп животных. Наиболее простым эволюционным алгоритмом оптимизации является алгоритм роя частиц (Particle Swarm Optimization, PSO-алгоритм), поскольку для его реализации необходимо уметь определять только значение оптимизируемой функции [5], которая, в случае построения бинарных классификаторов, является сложной, многоэкстремальной, многопараметрической [4, 6, 7].

Традиционный подход к применению PSO-алгоритма в задаче разработки SVM-классификатора заключается в его применении при фиксированном типе функции ядра с целью выбора оптимальных значений параметров функции ядра и значения параметра регуляризации. При этом может быть выбрано несколько типов функции ядра, с использованием которых будет построено несколько соответствующих этим функциям ядра SVM-классификаторов с оптимальными параметрами. Тогда в качестве искомого SVM-классификатора выбирается лучший из построенных.

Наряду с традиционным подходом к применению PSO-алгоритма предлагается применять новый модифицированный подход, реализующий одновременный поиск лучшего типа функции ядра, значений параметров функции ядра и значения параметра регуляризации. В основу этого подхода заложена идея о «перерождении» частиц, предполагающая возможность того, что некоторые частицы могут изменить тип функции ядра на тип, который соответствует частице с наилучшим значением точности классификации [4, 6]. Такой модифицированный PSO-алгоритм позволяет сократить временные затраты на разработку SVM-классификатора, что очень важно для классификации сложноорганизованных многомерных данных больших объемов.

В большинстве случаев SVM-классификатор на основе модифицированного PSO-алгоритма обеспечивает высокое качество классификации данных при приемлемых временных затратах, но с увеличением количества обрабатываемых данных и описывающих их характеристик растет время поиска оптимальных параметров SVM-классификатора. В связи с этим необходимы новые технологии, позволяющие проводить качественную классификацию таких данных с приемлемыми временными затратами на разработку классификатора.

В последнее время значительное внимание уделяется вопросу повышения точностей классификационных решений, полученных на основе алгоритмов машинного обучения. Для решения данного вопроса в работах отечественных и зарубежных авторов предлагаются различные подходы, реализующие создание тех или иных ансамблей классификаторов [2, 8–13].

В данной работе предлагается разработать двухуровневый классификатор, обеспечивающий повышение точности классификации сложноорганизованных многомерных данных больших объемов при приемлемых временных затратах на его разработку. При этом на первом уровне двухуровневого классификатора по результатам параллельного обучения должна быть создана группа SVM-классификаторов, обучающие выборки для которых формируются на основе исходного экспериментального набора данных. На втором уровне двухуровневого классификатора должен быть создан SVM-классификатор (с использованием модифицированного алгоритма роя частиц), обучающая выборка для которого формируется посредством ансамблирования решений группы SVM-классификаторов первого уровня, а именно посредством объединения опорных векторов, найденных SVM-классификаторами группы, в результате чего удастся существенно сократить объем данных, используемых для построения классификатора.

Аспекты разработки SVM-классификатора

При построении SVM-классификатора с помощью специальной функции $\kappa(z_i, z_\tau)$, назы-

ваемой ядром, определяется классифицирующая функция (классификатор) $F: Z \to Y$, сопоставляющая классу $Y = \{-1; +1\}$ произвольный объект $z_i \in Z$. При этом для обучения используется экспериментальный набор вида: $\{(z_1,y_1),...,(z_s,y_s)\}$, в котором каждому объекту $z_i \in Z$ поставлено в соответствие число $y_i \in Y$, принимающее значение −1 или +1, в зависимости от того, какому классу принадлежит объект z_i [1–3]. Каждый объект $z_i \in Z$ представлен q-мерным вектором числовых характеристик $z_i = (z_i^1, z_i^2, \ldots, z_i^q)$ (обычно нормированных значениями из отрезка [0, 1]), где z_i^l – числовое значение l-й характеристики для i-го объекта ($i = \overline{1,s}$, $l = \overline{1,q}$) [10, 11].

При разработке SVM-классификатора необходимо: 1) разделить экспериментальный набор на обучающую и тестовую выборки, состоящие соответственно из S и $s - S$ элементов ($s > S$); 2) определить параметры классификатора: тип функции ядра $\kappa(z_i, z_\tau)$, значения параметров ядра и значение параметра регуляризации C ($C > 0$), позволяющего найти компромисс между максимизацией ширины полосы, разделяющей классы, и минимизацией суммарной ошибки; 3) реализовать многократное обучение и тестирование на разных случайным образом сформированных обучающей и тестовой выборках с последующим определением лучшего SVM-классификатора в смысле обеспечения максимально возможной точности классификации. При этом тестовая выборка составляет от 1/10 до 1/3 от всего экспериментального набора данных, она не участвует в настройке параметров классификатора, а используется для проверки его точности [8].

В качестве функции ядра $\kappa(z_i, z_\tau)$, позволяющей разделить объекты разных классов, обычно используется одна из следующих функций [1–3]:

- линейная: $\kappa(z_i, z_\tau) = < z_i, z_\tau >$;
- полиномиальная: $\kappa(z_i, z_\tau) = (< z_i, z_\tau > + 1)^d$;
- радиальная базисная: $\kappa(z_i, z_\tau) = exp(- < z_i - z_\tau, z_i - z_\tau > /(2 \cdot \sigma^2))$;
- сигмоидная: $\kappa(z_i, z_\tau) = th(k_2 + k_1 \cdot < z_i, z_\tau >)$,

где $< z_i, z_\tau >$ – скалярное произведение векторов z_i и z_τ; d [$d \in N$ (по умолчанию $d = 3$)], σ [$\sigma > 0$ (по умолчанию $\sigma^2 = 1$)], k_2 [$k_2 < 0$ (по умолчанию $k_2 = -1$)] и k_1 [$k_1 > 0$ (по умолчанию $k_1 = 1$)] – некоторые параметры; th – гиперболический тангенс.

В случае линейной разделимости в результате обучения SVM-классификатора определяется гиперперплоскость $< w,z > + b = 0$ с вектором нормали w, разделяющая объекты из Z на два класса, где b – параметр, задающий смещение гиперплоскости относительно начала координат, $< w,z >$ – скалярное произведение векторов w и z [1–3]. Условие $-1 < < w,z > + b < 1$ задает полосу, которая разделяет классы. Чем шире эта полоса, тем увереннее можно классифицировать объекты, следовательно, необходимо максимизировать ширину полосы $2/< w,w >$ так, чтобы внутрь нее не попал ни один объект из обучающей выборки. Для этого решается задача квадратичной оптимизации [1–3]:

$$\begin{cases} < w,w > \to \min, \\ y_i \cdot (< w,z_i > + b) \geq 1, \ i = \overline{1,S}. \end{cases}$$

Если классы линейно неразделимы, задача построения разделяющей гиперплоскости (с учетом теоремы Куна-Таккера) сводится к задаче квадратичного программирования, содержащей только двойственные переменные λ_i ($i = \overline{1,S}$) [1–3]:

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

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 ($\lambda_i \neq 0$) [1–3]. Они находятся ближе всего к разделяющей гиперплоскости и несут всю информацию о разделении классов.

В результате обучения определяется классифицирующая функция, сопоставляющая объект z классу с меткой «–1» или «+1» [1–3]:

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

где $b = < w, z_i > - y_i$; $w = \sum_{i=1}^{S} \lambda_i \cdot y_i \cdot z_i$.

При этом суммирование в правиле (1)

выполняется только по опорным векторам ($\lambda_i \neq 0$).

Основная проблема, возникающая при разработке SVM-классификатора, связана с отсутствием рекомендаций по выбору типа функции ядра $\kappa(z_i, z_j)$, значений параметров функции ядра и значения параметра регуляризации C, при которых будет обеспечена высокая точность классификации объектов. Данная проблема может быть решена с применением тех или иных оптимизационных алгоритмов, в частности с использованием PSO-алгоритма, хорошо зарекомендовавшего себя при решении широкого спектра задач оптимизации.

Модифицированный PSO-алгоритм в задаче разработки SVM-классификатора

PSO-алгоритм является алгоритмом случайно-направленного поиска. Данный алгоритм работает со случайно сгенерированной популяцией решений и выполняет расчет значений целевой функции, осуществляя в процессе эволюции поиск лучшего решения [5].

При разработке SVM-классификатора предлагается использовать модифицированный PSO-алгоритм [4, 6, 7], в котором каждой i-й частице роя ($i = \overline{1,m}$) предлагается сопоставить вектор, описывающий ее позицию в пространстве поиска: (T_i, x_i^1, x_i^2, C_i), где T_i – номер типа функции ядра (например, 1, 2, 3 – для полиномиальной, радиальной базисной и сигмоидной функций соответственно), C_i – параметр регуляризации; x_i^1, x_i^2 – параметры функции ядра i-й частицы. При этом параметр x_i^1 полагается равным параметрам функций ядра d, σ или k_2 (в зависимости от того, какому типу функции ядра соответствует частица роя); параметр x_i^2 полагается равным параметру функций ядра k_1, если частица роя соответствует сигмоидному типу функции ядра, в противном случае значение этого параметра считается равным нулю. В процессе поиска возможно «перерождение» частицы – изменение значения ее координаты T_i (хранящей номер типа функции ядра) на номер того типа функции ядра, частицы которого показывают максимально высокое качество классификации [4, 6, 7]. При «перерождении» возможно изменение значений параметров x_i^1, x_i^2 и C_i так, чтобы они соответствовали новому типу функции ядра (с учетом диапазонов изменения их значений). Частицы, которые не подверглись «перерождению», осуществляют движение в своем собственном пространстве поиска некоторой размерности. Доля частиц, участвующих в «перерождении», определяется перед запуском алгоритма. Предлагаемая модификация PSO-алгоритма может быть представлена следующей последовательностью шагов.

Шаг 1. Определить параметры PSO-алгоритма: количество частиц в рое m, масштабирующий коэффициент для скорости K, личный и глобальный коэффициенты ускорения $\hat{\varphi}$ и $\tilde{\varphi}$, максимальное количество итераций PSO-алгоритма $N_{\max}$. Определить типы T функций ядра, участвующих в поиске ($T = 1$ – полиномиальная, $T = 2$ – радиальная базисная, $T = 3$ – сигмоидная функция ядра), и границы изменения параметров функции ядра и параметра регуляризации C для выбранных типов функций ядра T: $x_{\min}^{1T}$, $x_{\max}^{1T}$, $x_{\min}^{2T}$, $x_{\max}^{2T}$, $C_{\min}^{T}$, $C_{\max}^{T}$ ($x_{\min}^{2T} = 0$ и $x_{\max}^{2T} = 0$ для $T = 1$ и $T = 2$). Задать значение процента «перерождения» частиц p.

Шаг 2. Задать равное количество частиц для каждого ядра T, включенного в поиск. Затем для каждой i-й частицы ($i = \overline{1,m}$) инициализировать координату T_i (так, чтобы каждому используемому в процессе поиска типу функции ядра соответствовало одинаковое количество частиц). Остальные координаты i-й частицы ($i = \overline{1,m}$) сгенерировать случайным образом из соответствующих диапазонов: $x_i^1 \in [x_{\min}^{1T}, x_{\max}^{1T}]$, $x_i^2 \in [x_{\min}^{2T}, x_{\max}^{2T}]$ ($x_i^2 = 0$ при $T = 1$ и $T = 2$), $C_i \in [C_{\min}^{T}, C_{\max}^{T}]$. Инициализировать случайный вектор скорости $v_i(v_i^1, v_i^2, v_i^3)$ i-й частицы ($i = \overline{1,m}$) ($v_i^2 = 0$ при $T = 1$ и $T = 2$). Принять начальное положение i-й частицы ($i = \overline{1,m}$) за лучшее ее известное положение $(\hat{T}_i, \hat{x}_i^1, \hat{x}_i^2, \hat{C}_i)$ и определить лучшую частицу с вектором координат $(\tilde{T}, \tilde{x}^1, \tilde{x}^2, \tilde{C})$ среди всех m частиц, а также лучшую частицу для каждого типа функции ядра T, включенного в поиск, с вектором координат $(\overline{T}, \overline{x}^{1T}, \overline{x}^{2T}, \overline{C}^T)$. При этом количество выполненных итераций полагается равным 1.

Шаг 3. Пока количество выполненных итераций не превышает заданное число $N_{\max}$, выполнять такие операции:

а) «перерождение» частиц: из тех частиц, у которых координата $T_i \neq \tilde{T}$ ($i = \overline{1,m}$), выбрать p % частиц, показавших самое низкое качество

классификации, и изменить значение координаты T_i с номером типа функции ядра на значение $\widetilde{T}$; изменить значения параметров x_i^1, x_i^2, C_i «перерождаемой» частицы так, чтобы они соответствовали новому типу ядра $\widetilde{T}$ (то есть попадали в соответствующие диапазоны);

б) выполнить коррекцию вектора скорости $v_i(v_i^1, v_i^2, v_i^3)$ i-й частицы ($i = \overline{1,m}$) по формуле:

$$v_i^j = \begin{cases} \chi \cdot [v_i^j + \hat{\varphi} \cdot \hat{r} \cdot (\hat{x}_i^j - x_i^j) + \widetilde{\varphi} \cdot \widetilde{r} \cdot (\bar{x}^{jT} - x_i^j)], & j = 1,\ 2, \\ \chi \cdot [v_i^j + \hat{\varphi} \cdot \hat{r} \cdot (\hat{C}_i - C_i) + \widetilde{\varphi} \cdot \widetilde{r} \cdot (\overline{C}^T - C_i)], & j = 3, \end{cases}$$

где $\hat{r}$ и $\widetilde{r}$ – случайные числа в интервале (0, 1); $\bar{x}^{1T}, \bar{x}^{2T}, \overline{C}^T$ – координаты частицы, лучшей для типа функции ядра $T = T_i$; χ – коэффициент сжатия, рассчитанный по формуле:

$$\chi = 2 \cdot K \,/\, \left| 2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi} \right|; \quad \varphi = \hat{\varphi} + \widetilde{\varphi} \quad (\varphi > 4);$$

в) изменить положение (x_i^1, x_i^2, C_i) i-й частицы ($i = \overline{1,m}$) в соответствии с формулами:

$$x_i^j = x_i^j + v_i^j \text{ для } j = 1,\ 2,$$

$$C_i = C_i + v_i^3;$$

г) произвести расчет точности SVM-классификатора со значениями параметров (T_i, x_i^1, x_i^2, C_i) $(i = \overline{1,m})$ с целью поиска оптимальной комбинации $(\widetilde{T}, \widetilde{x}^1, \widetilde{x}^2, \widetilde{C})$, обеспечивающей высокое качество классификации;

д) увеличить количество итераций на 1.

После выполнения данного алгоритма будет определена частица с оптимальной комбинацией значений параметров $(\widetilde{T}, \widetilde{x}^1, \widetilde{x}^2, \widetilde{C})$, обеспечивающая высшее качество классификации на включенных в поиск типах функций ядра.

В модифицированном PSO-алгоритме у частиц происходит изменение координаты (значения), отвечающей за номер функции ядра. Поэтому после выполнения данного алгоритма может оказаться, что все частицы будут сосредоточены в пространстве поиска, которое соответствует функции ядра с максимально высоким качеством классификации. При этом остальные пространства поиска окажутся пустыми, так как у всех частиц, принадлежавших этим пространствам изначально, произойдет «перерождение» координаты с номером (значения) типа функции ядра. В ряде случаев (при небольших значениях количества итераций $N_{\max}$ или процента «перерождения» частиц p) также возможно, что у некоторых частиц «перерождение» ядер не произойдет и они останутся в своем пространстве поиска [6, 7].

Подход к разработке двухуровневого классификатора

Основное ограничение, связанное с применением PSO-алгоритма к задаче классификации сложноорганизованных многомерных данных больших объемов, связано с существенными временными затратами на поиск оптимальных параметров классификатора (типа функции ядра, значений параметров ядра и значения параметра регуляризации). Несколько сократить время поиска можно, используя небольшое количество частиц в рое и небольшое количество итераций модифицированного PSO-алгоритма. Но в этом случае будет сокращено количество формируемых и сравниваемых SVM-классификаторов, что, вероятно, приведет к построению менее хорошего SVM-классификатора.

Один из подходов к уменьшению времени разработки SVM-классификатора предполагает сокращение объема набора данных, используемых для создания обучающей и тестовой выборок данных. При этом из рассмотрения следует исключить те из имеющихся объектов, которые никаким образом не повлияют на результаты классификации. В этом случае может быть использован следующем теоретическом факт: при обучении SVM-классификатора определяется классифицирующая функция (1), в которой суммирование идет не по всей выборке объектов, а только по опорным векторам, для которых $\lambda_i \neq 0$ [1–3]. Именно опорные векторы содержат всю информацию о разделении классов и по ним происходит построение гиперплоскостей, разделяющих классы.

С учетом данного теоретического факта возможна разработка двухуровневого классификатора, обеспечивающего повышение точности классификации сложноорганизованных многомерных данных больших объемов при приемлемых временных затратах. При этом на первом уровне двухуровневого классификатора предлагается создать группу SVM-классификаторов, обучающие выборки для которых формируются на основе исходного экспериментального набора данных, а на втором уровне двухуровневого классификатора – создать SVM-классификатор с использованием модифицированного алгоритма роя частиц, обучающая выборка для которого формируется на основе набора опорных векторов, полученных на первом уровне двухуровневого классификатора.

Предлагаемый подход к разработке двухуровневого классификатора может быть описан следующей последовательностью действий.

1. Обучить на экспериментальном наборе данных t частных SVM-классификаторов с использованием различных обучающих выборок данных TR_1, TR_2, …, TR_t с целью получения t частных SVM-классификаторов, составляющих группу. При обучении частных SVM-классификаторов следует использовать различные типы функции ядра, различные значения параметров функции ядра и различные значения параметра регуляризации.

2. Получить по результатам обучения всех частных классификаторов наборы опорных векторов SV_1, SV_2, …, SV_t, в результате объединения которых сформировать набор SV из ℓ объектов ($\ell \le s$, где s – количество объектов экспериментального набора данных).

3. Выделить из набора SV поднабор SV^+, состоящий из L объектов, ($L \le \ell$), которые были определены теми или иными частными классификаторами как опорные векторы и реальный класс которых совпал с классом, в который этот объект был отнесен тем или иным частным классификатором. Это необходимо для того, чтобы заведомо ложные данные не участвовали в обучении итогового SVM-классификатора. Остальные объекты набора SV составят поднабор SV^- из $\ell - L$ объектов. Поднабор SV^+ будет использоваться для обучения, а поднабор SV^- – для тестирования итогового SVM-классификатора.

4. Определить с помощью PSO-алгоритма оптимальные параметры SVM-классификатора, используя в качестве обучающей выборки поднабор SV^+, а в качестве тестовой поднабор SV^-.

5. Построить итоговый SVM-классификатор на основе PSO-алгоритма.

6. Выполнить доклассификацию объектов исходного экспериментального набора данных, не вошедших в поднаборы SV^+ и SV^-.

7. Оценить точность построенного двухуровневого классификатора, а также время, затраченное на его разработку.

Экспериментальные исследования

Целесообразность использования предлагаемого подхода к разработке двухуровневого классификатора была подтверждена на тестовых и реальных данных.

Реальные данные для проведения экспериментальных исследований были взяты из проекта Statlog и библиотеки машинного обучения UCI. В частности, были использованы два набора данных из медицинской диагностики, два набора данных для кредитного скоринга и один набор данных для создания прогнозной модели распознавания спама на основе данных электронных писем:

– выборка данных хирургического отделения университета штата Висконсин о диагностике рака молочной железы 569 пациентов, содержащая информацию о клеточных ядрах, описываемых 30 характеристиками ($q = 30$; в таблице 1 набор WDBC, источник http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/);

– выборка данных о диагностике болезни сердца у 270 пациентов, описываемых 13 характеристиками ($q = 13$; в таблице 1 набор Heart, источник http://archive.ics.uci.edu/ml/machine-learning- databases/statlog/heart/);

– выборка данных о заявках на потребительские кредиты в Австралии, содержащая информацию о 690 заявках, описываемых 14 характеристиками ($q = 14$; в таблице 1 набор Australian, источник http://archive.ics.uci.edu/ml/machine-learning- databases/statlog/australian/);

– выборка данных заемщиков немецкого банка, содержащая информацию о 1000 кредиторах, описываемых 24 характеристиками ($q = 24$; в таблице 1 набор German, источник http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german/);

– выборка электронных писем, состоящая из 4601 сообщений, описываемых 57 характеристиками ($q = 57$; в таблице 1 набор Spam, источник https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/).

Кроме того, для экспериментальных исследований была использована тестовая выборка данных МОТП12 (в таблице 1 набор МОТП12; источник http:// machinelearning.ru/wiki/images/b/b2/MOTP12_svm_example.rar) [6].

На всех наборах данных имеет место случай бинарной классификации.

Экспериментальные вычисления производились на ПЭВМ, работающей под 64-разрядной версией Windows 7, с оперативной памятью 3 Гб и четырехядерным процессором Intel® Core™ i3 с тактовой частотой каждого ядра 2,53 ГГц. При моделировании использовался SVM-алгоритм из пакета прикладных программ для выполнения технических вычислений MATLAB 7.12.0.635.

При построении SVM-классификатора для выбора оптимальных значений его параметров были использованы традиционный и модифицированный PSO-алгоритмы. При этом в поиск были включены ядра с полиномиальной (№1), радиальной базисной (№2) и сигмоидной функциями (№3). Для них были установлены одинаковые значения параметров PSO-алгоритма (в

частности, было задано количество частиц в рое, равное 600, а количество итераций – 20) и одинаковые диапазоны изменения значений искомых параметров SVM-классификатора.

Таблица 1 – Результаты поиска с применением традиционного и модифицированного PSO-алгоритмов

Набор данных	Количество объектов	Количество характеристик	Тип PSO-алгоритма	Найденные параметры			Ошибки		Число опорных векторов	Точность (%)	Время поиска (с)
				номер ядра	параметр регуляризации	параметр ядра	при обучении	при тестировании			
Heart	270	13	традиционный	2	9,6	3,34	7 из 230	7 из 40	106	94,81	2276
			модифицированный	2	6,01	2,99	6 из 230	3 из 40	131	96,67	876
WDBC	569	30	традиционный	2	9,36	2,89	0 из 427	3 из 142	113	99,47	3919
			модифицированный	2	9,84	3,99	0 из 427	2 из 142	79	99,65	1464
Australian	690	14	традиционный	2	9,28	2,51	23 из 518	25 из 172	248	93,04	9086
			модифицированный	2	4,45	2,22	26 из 518	20 из 172	258	93,33	2745
German	1000	24	традиционный	1	1,58	3	0 из 850	42 из 150	438	95,8	14779
			модифицированный	1	5,53	4	0 из 850	42 из 150	546	95,8	5766
МОТП12	400	2	традиционный	2	6,31	0,19	7 из 340	8 из 60	156	96,25	15632
			модифицированный	2	5,69	0,22	11 из 340	4 из 60	146	96,25	7146
Spam	4601	57	традиционный	2	7,82	2,47	40 из 3681	56 из 920	1634	97,91	92645
			модифицированный	2	8,57	2,45	36 из 3681	60 из 920	1659	97,91	44933

В таблице 1 приведено краткое описание характеристик каждого набора данных и представлены результаты поиска оптимальных значений параметров SVM-классификатора с применением традиционного и модифицированного PSO-алгоритмов (в одинаковых диапазонах изменения параметров и при одинаковых параметрах PSO-алгоритма); количество ошибок, допущенных при обучении и тестировании SVM-классификатора; время поиска.

Так, например, для набора WDBC с применением традиционного и модифицированного PSO-алгоритмов в качестве оптимального ядра было определено ядро с радиальной базисной функцией (№2). При этом для традиционного PSO-алгоритма оптимальными значениями параметра ядра и параметра регуляризации оказались соответственно значения $\sigma = 2{,}89$ и $C = 9{,}36$, а для модифицированного PSO-алгоритма – соответственно значения $\sigma = 3{,}99$ и $C = 9{,}84$. Точность классификации, достигнутая с применением традиционного PSO-алгоритма, составила 99,47%, а с применением модифицированного PSO-алгоритма – 99,65%. Время поиска составило соответственно 3919 и 1464 секунды.

Из таблицы 1 видно, что в результате поиска для рассматриваемых наборов данных оба алгоритма определили в качестве оптимальных одинаковый тип функции ядра, близкие значения параметра функции ядра и параметра регуляризации, а также близкие значения точности обучения и тестирования SVM-классификатора, построенного с учетом найденных значений параметров. Однако по времени поиска модифицированный PSO-алгоритм оказался более эффективным – затраченное им время на поиск искомого решения оказалось меньше (в 2 – 3 раза), чем время поиска с применением традиционного PSO-алгоритма.

Как показывают экспериментальные исследования, время поиска определяется как входными параметрами PSO-алгоритма (коэффициентами скорости, количеством и типом функции искомых ядер, диапазонами поиска и т.п.), так и самим набором данных, используемым при обучении и тестировании SVM-классификаторов (их количеством, числом характеристик). Меньшее время поиска модифицированного алгоритма в сравнении с временем поиска традиционным алгоритмом связано с тем, что частицы «перерождаются» из ядра, в котором на построение классификатора уходит больше времени (в частности, самым затратным по времени является ядро с полиномиальной функцией), чем на его построение в новом ядре, куда частицы попадают.

При определении оптимальных параметров SVM-классификатора с использованием традиционного или модифицированного алгоритма в заданном пространстве поиска происходит построение большого числа классификаторов с последующим определением того из них, который показывает максимальную точность классификации при минимальном числе опорных векторов. Поэтому при включении в поиск 600 частиц на 20 итераций PSO-алгоритма приходится строить и сравнивать 12000 классификаторов, а если за среднее время обучения и

тестирования одного классификатора принять 3 секунды, то временные затраты на поиск оптимальных параметров SVM-классификатора составят 36000 секунд ($600 \times 20 \times 3$) или примерно 10 часов.

Анализируя данные таблицы 1, можно заключить, что увеличение объема входных данных приводит к проблеме увеличения времени поиска оптимальных параметров SVM-классификатора с помощью PSO-алгоритма. Сократить время поиска оптимальных параметров предлагается посредством сокращения количества исходных данных, используя предложенный подход к разработке двухуровневого классификатора. Хорошим демонстрационным примером для визуального представления предложенного подхода служит набор данных MOTП12.

Из таблицы 1 видно, что, несмотря на небольшие объем ($s = 400$) и количество характеристик ($q = 2$), запущенные для этого набора PSO-алгоритмы находят оптимальные параметры для SVM-классификатора за довольно продолжительное время (дольше, чем, например, для набора WDBC из 569 объектов с 30 характеристиками). Это связано с тем, что данные сложноорганизованы. На рисунке 1 представлены расположение этих данных в пространстве 2-D и их разбиение на два класса, объекты которых помечены маркерами «крестик» и «звездочка».

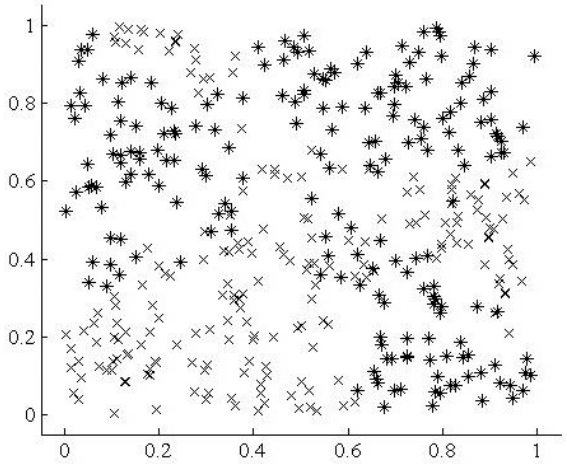

Рисунок 1 – Представление данных набора MOTП12 в пространстве D-2

Таблица 2 – Параметры и характеристики итогового и частных классификаторов (набор MOTП12)

Номер частного классификатора	1	2	3	4	5	6	7	8	9	**Итог**
Тип функции ядра	1	1	1	2	2	2	3	3	3	**2**
Параметр регуляризации	1	3,2	0,2	1,2	1,7	4,5	8,4	9	6,6	**8,5**
Параметры ядра	3	5	3	0,50	0,6	0,6	0,80; -3,00	0,50; -2,20	0,90; -2,50	**0,25**
Точность на всей выборке (%)	90,5	91	88,25	90,75	91,25	91,25	85	87,25	86,5	**97**
Количество опорных векторов	85	59	111	127	113	83	146	145	103	**104**
Размер обучающей выборки	300	280	320	340	320	300	340	320	300	**203**
Количество ошибок при обучении	28	22	31	34	26	19	49	38	41	**12**
Размер тестовой выборки	100	120	80	60	80	100	60	80	100	**13**
Количество ошибок при тестировании	10	14	16	3	9	16	11	13	13	**0**
Количество классифицируемых объектов	–	–	–	–	–	–	–	–	–	**184**
Количество ошибок при классификации	–	–	–	–	–	–	–	–	–	**0**
Чувствительность классификатора (%)	85,85	89,27	81,95	88,78	87,32	89,76	78,05	81,46	80	**97,56**
Специфичность классификатора (%)	95,38	92,82	94,87	92,82	95,38	92,82	92,31	93,33	93,33	**96,41**
Количество ошибок I рода	9	14	10	14	9	14	15	13	13	**7**
Количество ошибок II рода	29	22	37	23	26	21	45	38	41	**5**
Время построения, с	1	1	1	<1	<1	<1	<1	<1	<1	**3267**

Из рисунка 1 видно, что провести кривую, разделяющую классы, довольно затруднительно.

Для этого набора данных на первом уровне двухуровневого классификатора было обучено 9 частных SVM-классификаторов с использованием различных параметров (таблица 2).

В поиск были включены по 3 ядра с полиномиальной (№1), радиальной базисной (№2) и сигмоидной (№3) функциями. В таблице 2 для сигмоидной функции ядра первое значение параметра ядра – k_1, второе – k_2. Время, использованное на построение одного частного классификатора, составило в среднем менее 1 секунды.

На рисунке 2 показаны исходный экспериментальный набор данных и разделяющая эти данные кривая, полученная в результате построения SVM-классификатора при различных типах функции ядра для трех случаев. На

рисунке 2 объекты, определенные частными классификаторами в качестве опорных векторов, помечены кружочком, а объекты, вошедшие в тестовую выборку, – точкой. Из исходных 400 объектов было отобрано 216, определенных каким-либо частным классификатором в качестве опорных векторов (рисунок 3). Причем 203 из них были классифицированы верно и составили обучающую выборку (SV^+), а 13 были классифицированы неверно и вошли в тестовую выборку (SV^-).

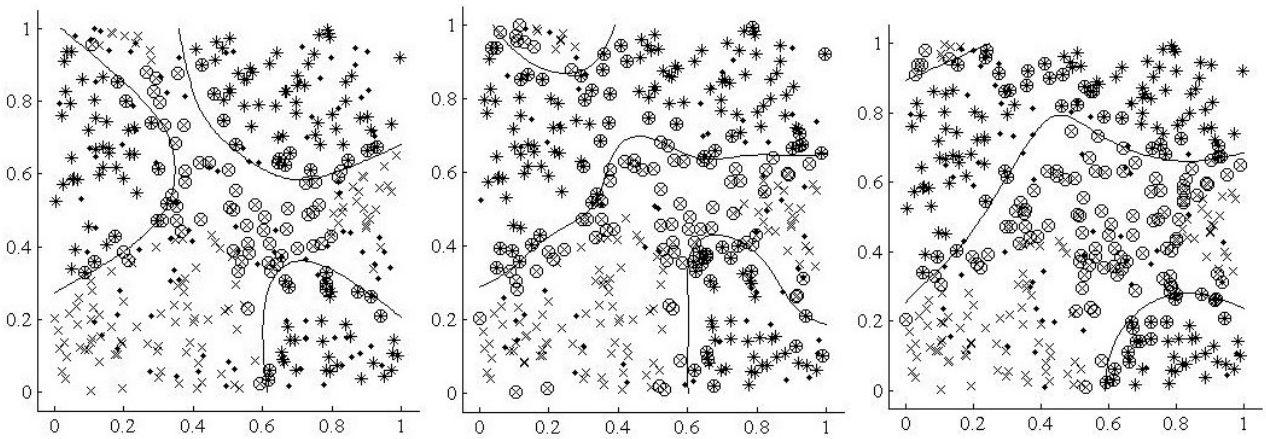

Рисунок 2 – Результат разделения набора МОТП12 частными SVM-классификаторами: слева – полиномиальной, в центре – радиальной базисной, справа – сигмоидной функциями ядра

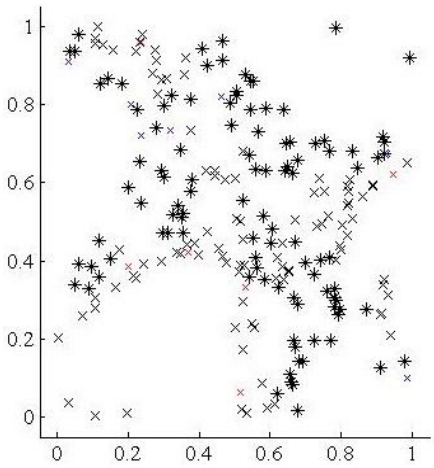

Рисунок 3 – Набор объектов, используемый для построения SVM-классификатора второго уровня

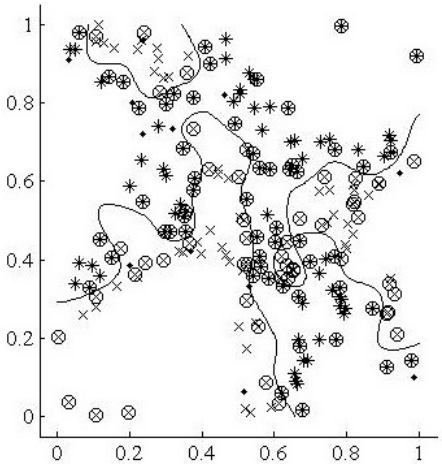

Рисунок 4 – Кривая, разделяющая классы для SVM-классификатора второго уровня

С использованием полученных таким образом обучающей и тестовой выборок на втором уровне двухуровневого классификатора на основе модифицированного PSO-алгоритма были определены оптимальные параметры SVM-классификатора. Параметры PSO-алгоритма и диапазоны изменения параметров функций ядер, участвующих в поиске, были определены аналогично предыдущим случаям (чтобы можно было проводить сравнение). Время поиска оптимальных параметров составило 3267 секунд, что более чем в 2 раза меньше времени, потраченного на поиск оптимальных параметров, при использовании всего набора экспериментальных данных (7146 секунд).

На рисунке 4 показаны результаты обучения и тестирования SVM-классификатора на основе модифицированного PSO-алгоритма и кривая, разделяющая классы.

Объекты, участвующие в обучении SVM-классификатора, на рисунке 4 помечены маркерами «звездочка» и «крестик», а объекты тестовой выборки помечены маркером «точка». Объекты, определенные в этом наборе в качестве опорных векторов, помечены маркером круглой формы.

Остальные 184 объекта (46% от исходного набора) не использовались при построении итогового SVM-классификатора и составили классифицируемый набор данных. После построения итогового SVM-классификатора все 184 объекта были правильно классифицированы.

В ходе эксперимента было установлено, что

частные классификаторы показывают точность классификационных решений от 85% до 91,25%. Точность итогового классификатора составила 97%. Таким образом, использование предложенного подхода позволило повысить точность классификации более чем на 5,5%, по сравнению с максимальной точностью одного из частных SVM-классификаторов, входящего в группу. При этом количество объектов, используемых при обучении и тестировании SVM-классификатора, было сокращено с 400 до 216.

Предложенный подход был применен при разработке двухуровневого классификатора для набора данных Spam, который может рассматриваться как набор многомерный данных большого объема. На первом уровне двухуровневого классификатора с использованием различных параметров было обучено 10 частных SVM-классификаторов (таблица 3), обучающие выборки данных для которых формировались на основе исходного экспериментального набора данных.

Из обученных частных классификаторов были отобраны SVM-классификаторы, показывающие неплохую точность классификации (от 80%) при небольшом количестве опорных векторов (до 1000), на основе которых формировалась выборка данных для обучения SVM-классификатора на основе модифицированного PSO-алгоритма на втором уровне двухуровневого классификатора.

Таблица 3 – Параметры и характеристики итогового и частных классификаторов (набор Spam)

Номер частного классификатора	1	2	3	4	5	6	7	8	9	10	**Итог**
Тип функции ядра	1	1	1	2	2	2	2	3	3	3	**1**
Параметр регуляризации	1	2	2,5	10	12	10	10	0,5	0,5	3,4	**0,1**
Параметры ядра	3	4	3	15	10	15	8,00	0,15; -1,00	0,50; -1,50	0,30; -0,80	**3**
Точность на всей выборке (%)	89,15	86,66	84,7	94,09	94,39	93,96	91,28	86,09	84,24	81,85	**97,26**
Количество опорных векторов	579	561	576	752	760	859	723	515	584	638	**717**
Размер обучающей выборки	3681	3911	3451	3221	3681	3681	3681	3451	3911	3451	**1834**
Количество ошибок при обучении	364	497	505	178	189	214	315	484	624	625	**27**
Размер тестовой выборки	920	690	1150	1380	920	920	920	1150	690	1150	**221**
Количество ошибок при тестировании	135	117	199	94	69	64	86	156	101	210	**26**
Количество классифицируемых объектов	–	–	–	–	–	–	–	–	–	–	**2546**
Количество ошибок при классификации	–	–	–	–	–	–	–	–	–	–	**73**
Чувствительность классификатора (%)	73,97	69,11	62,82	92,11	89,74	91,89	80,14	76,78	77,88	81,96	**95,37**
Специфичность классификатора (%)	99,03	98,06	98,92	95,37	97,42	95,3	98,53	92,14	88,38	81,78	**98,49**
Количество ошибок I рода	27	54	30	129	72	131	41	219	324	508	**42**
Количество ошибок II рода	472	560	674	143	186	147	360	421	401	327	**84**
Время построения, с	5	16	5	3	3	3	3	2	2	2	**19599**

В качестве оптимальной была определена полиномиальная функция ядра, для которой оптимальное значение параметра d оказалось равно 3 ($d = 3$), а оптимальное значение параметра регуляризации оказалось равно 0,1 ($C = 0{,}1$). При этом время поиска составило 19566 секунд, что более чем в два раза меньше, чем при поиске параметров с помощью модифицированного PSO-алгоритма, примененного к исходному экспериментальному набору из 4601 письма (44933 секунды).

Точность итогового SVM-классификатора составила 97,26%, при этом точность классификационных решений частных классификаторов составляла от 81,85% до 94,39%. Таким образом, использование предлагаемого двухуровневого классификатора позволило повысить точность классификации почти на 3%, по сравнению с максимальной точностью одного из частных SVM-классификаторов, входящего в группу. При этом количество объектов, используемых при обучении и тестировании SVM-классификатора, было сокращено с 4601 до 2055, т.е. более чем в два раза.

Заключение

Предлагаемый двухуровневый классификатор позволяет принимать высокоточные решения по классификации сложноорганизованных многомерных данных больших объемов. При этом существенное сокращение временных затрат на разработку искомого классификатора

обеспечено благодаря значительному сокращению объема набора данных, используемых при формировании обучающей и тестовой выборок для SVM-классификатора на основе модифицированного PSO-алгоритма (на втором уровне двухуровневого классификатора). Данное сокращение объема набора данных достигнуто в результате исключения из исходного экспериментального набора данных объектов, не являющихся опорными векторами по результатам построения группы SVM-классификаторов (на первом уровне двухуровневого классификатора).

Применение принципов, предложенных к использованию при разработке двухуровневого классификатора, будет особенно востребовано в задачах, связанных с классификацией больших данных (Big Data), являющихся неотъемлемой частью современных информационных потоков.

Библиографический список

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

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

3. **Вьюгин В. В**. Элементы математической теории машинного обучения: учеб. пособие. М.: МФТИ, 2010. 252 с.

4. **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.

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

6. **Demidova L., Nikulchev E., Sokolova Yu**. The SVM Classifier Based on the Modified Particle Swarm Optimization // International Journal of Advanced Computer Science and Applications (IJACSA). 2016. Vol. 7. No. 2. P. 16-24.

7. **Demidova L., Sokolova Yu**. Modification Of Particle Swarm Algorithm For The Problem Of The SVM Classifier Development // В сборнике: 2015 International Conference «Stability and Control Processes» in Memory of V.I. Zubov (SCP). 2015. P. 623–627.

8. **Демидова Л. А., Соколова Ю. С.** Разработка ансамбля SVM-классификаторов с использованием декорреляционного алгоритма максимизации // Информатика и системы управления. 2016. № 1 (47). С. 95-105.

9. **Demidova L., Sokolova Yu.** Development of the SVM Classifier Ensemble for the Classification Accuracy Increase // 6th Seminar on Industrial Control Systems: Analysis, Modeling and Computation (ITM Web of Conferences). 2016. Vol. 6.

10. **Демидова Л. А., Соколова Ю. С.** Использование SVM-алгоритма для уточнения решения задачи классификации объектов с применением алгоритмов кластеризации // Вестник Рязанского государственного радиотехнического университета. 2015. № 51. С. 103-113.

11. **Demidova L., Nikulchev E., Sokolova Yu**. Use of Fuzzy Clustering Algorithms' Ensemble for SVM Classifier Development // International Review on Modelling and Simulations (IREMOS). 2015. Vol. 8. No. 4. P. 446-457.

12. **Demidova L., Sokolova Yu**. SVM-Classifier Development With Use Of Fuzzy Clustering Algorithms' Ensemble On The Base Of Clusters' Tags' Vectors' Similarity Matrixes // В сборнике: 16th International Symposium on Advanced Intelligent Systems 2015. P. 889–906.

13. **Demidova L., Sokolova Yu**. Training Set Forming For SVM Algorithm With Use Of The Fuzzy Clustering Algorithms Ensemble On Base Of Cluster Tags Vectors Similarity Matrices // В сборнике: 2015 International Conference «Stability and Control Processes» in Memory of V.I. Zubov (SCP) 2015. P. 619-622.

UDC 004.855.5

DEVELOPMENT OF TWO-LEVEL CLASSIFIER FOR ELABORATE MULTIDIMENSIONAL DATA OF LARGE CAPACITIES

L. A. Demidova, PhD (technical sciences), full professor, RSREU, Ryazan; liliya.demidova@rambler.ru
Yu. S. Sokolova, lecturer, RSREU, JuliaSokolova62@yandex.ru

The problem of classification of elaborate multidimensional data of large capacities which is inherent in various social and economic systems has been considered. ***The purpose of work*** *is the increase of classification accuracy of elaborate multidimensional data of large capacities by means of development of two-level classifier on the base of SVM algorithm at the acceptable time expenditures. At the first level of two-level classifier the group of SVM classifiers for which the training sets are formed on the base of initial data set is created. At the second level of two-level classifier SVM classifier with the use of the modified particle swarm algorithm for which the training set is formed on the base of the set of support vectors received at the first level of two-level classifier is developed. The given results of experimental studies confirm application efficiency of the offered two-level classifier for the problem of classification of elaborate multidimensional data of large capacities.*

Keywords*: SVM-classifier, support vectors, kernel function type, kernel function parameters, regularization parameter, particle swarm algorithm.*

DOI: 10.21667/1995-4565-2016-56-2-71-82

References

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

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

3. **Vyugin V. V.** *Elementy matematicheskoj teorii mashinnogo obucheniya: ucheb. posobie* (Elements of mathematical machine learning theory: a tutorial). Moscow, MFTI, 2010, 252 p. (in Russian).

4. **Demidova L. A., Sokolova Yu. S.** Aspekty primeneniya algoritma roya chastic v zadache razrabotki SVM-klassifikatora. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2015, no. 53. pp. 84-92 (in Russian).

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

6. **Demidova L., Nikulchev E., Sokolova Yu**. The SVM Classifier Based on the Modified Particle Swarm Optimization. *International Journal of Advanced Computer Science and Applications (IJACSA)*. 2016, vol. 7, no. 2, pp. 16-24.

7. **Demidova L., Sokolova Yu**. Modification Of Particle Swarm Algorithm For The Problem Of The SVM Classifier Development. *V sbornike: 2015 International Conference «Stability and Control Processes» in Memory of V.I. Zubov (SCP)* 2015, pp. 623–627.

8. **Demidova L. A., Sokolova Yu. S**. Razrabotka ansamblya SVM-klassifikatorov s ispolzovaniem dekorrelyacionnogo algoritma maksimizacii. *Informatika i sistemy upravleniya*. 2016, no 1 (47), pp. 95-105 (in Russian).

9. **Demidova L., Sokolova Yu.** Development of the SVM Classifier Ensemble for the Classification Accuracy Increase. *6th Seminar on Industrial Control Systems: Analysis, Modeling and Computation (ITM Web of Conferences)*. 2016, vol. 6.

10. **Demidova L. A., Sokolova Yu. S**. Ispolzovanie SVM-algoritma dlya utochneniya resheniya zadachi klassifikacii obektov s primeneniem algoritmov klasterizacii. *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2015, no. 51, pp. 103-113 (in Russian).

11. **Demidova L., Nikulchev E., Sokolova Yu**. Use of Fuzzy Clustering Algorithms' Ensemble for SVM Classifier Development. *International Review on Modelling and Simulations (IREMOS)*. 2015, vol. 8, no. 4, pp. 446-457.

12. **Demidova L., Sokolova Yu**. SVM-Classifier Development With Use Of Fuzzy Clustering Algorithms' Ensemble On The Base Of Clusters' Tags' Vectors' Similarity Matrixes. *V sbornike: 16th International Symposium on Advanced Intelligent Systems 2015*, pp. 889–906.

13. **Demidova L., Sokolova Yu**. Training Set Forming For SVM Algorithm With Use Of The Fuzzy Clustering Algorithms Ensemble On Base Of Cluster Tags Vectors Similarity Matrices. *V sbornike: 2015 International Conference «Stability and Control Processes» in Memory of V.I. Zubov (SCP)* 2015, pp. 619-622.