

УДК 004.855.5

## **ИСПОЛЬЗОВАНИЕ МОДИФИЦИРОВАННОГО АЛГОРИТМА РОЯ ЧАСТИЦ В ЗАДАЧЕ РАЗРАБОТКИ SVM-КЛАССИФИКАТОРА**

*Статья поступила в редакцию 03.02.2016, в окончательном варианте 21.02.2016*

**Демидова Лилия Анатольевна**, доктор технических наук, профессор, Рязанский государственный радиотехнический университет, 390005, Российская Федерация, г. Рязань, ул. Гагарина, 59 / 1, e-mail: liliya.demidova@rambler.ru

**Соколова Юлия Сергеевна**, старший преподаватель, Рязанский государственный радиотехнический университет, 390005, Российская Федерация, г. Рязань, ул. Гагарина, 59 / 1, e-mail: JuliaSokolova62@yandex.ru

Для задачи построения SVM-классификатора, обеспечивающего высокое качество классификации данных, рассматривается проблема разработки модифицированного алгоритма роя частиц, реализующего одновременный поиск типа функции ядра, значений параметров функции ядра, значения параметра регуляризации, а также соответствующего им набора опорных векторов. В основу предлагаемого модифицированного алгоритма роя частиц заложена идея о «перерождении» частиц. Реализация алгоритма предусматривает, что некоторые частицы с худшими значениями показателя точности классификации данных (ПТКД) могут изменять тип своей функции ядра на тот тип, который соответствует частице с лучшим значением ПТКД. Приведены результаты сравнительного анализа традиционного и модифицированного алгоритмов роя частиц, полученные в ходе экспериментальных исследований. Они подтверждают целесообразность использования модифицированного алгоритма роя частиц с целью сокращения временных затрат на построение искомого SVM-классификатора.

**Ключевые слова:** SVM-алгоритм, разделяющая гиперплоскость, опорные векторы, классификация, оптимизация, параметры функции ядра, параметр регуляризации, алгоритм роя частиц

## **USE OF THE MODIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM IN THE SVM-CLASSIFIER DEVELOPMENT PROBLEM**

**Demidova Liliya A.**, D.Sc. (Engineering), Professor, Ryazan State Radio Engineering University, 59 / 1 Gagarin St., Ryazan, 390005, Russian Federation, e-mail: liliya.demidova@rambler.ru

**Sokolova Yulia S.**, Senior Lecturer, Ryazan State Radio Engineering University, 59 / 1 Gagarin St., Ryazan, 390005, Russian Federation, e-mail: JuliaSokolova62@yandex.ru

For the problem of development of the SVM classifier providing high quality of data classification the task of development of the modified particle swarm optimization algorithm, which carries out the simultaneous search of the kernel function type, values of the kernel function optimization parameters, value of the regularization parameter, and also the set of the support vectors corresponding to them, is considered. In a basis of the offered particle swarm optimization algorithm the idea about «regeneration» of particles is put. Realization of the algorithm assumes that some particles with the worst values of the accuracy indicator of data classification (AIDC) can change type of their kernel function to the type which corresponds to the particle with the best value of the AIDC. The results of the comparative analysis of the traditional and modified particle swarm optimization algorithms received during the experimental studies are given. These results confirm the expediency of use of the modified particle swarm optimization algorithm for the purpose of reduction of time expenditure for development of the required SVM classifier.

**Keywords:** SVM-algorithm, separating hyperplane, support vectors, classification, optimization, parameters of kernel function, regularization parameter, particle swarm algorithm

**Введение.** В настоящее время для решения широкого спектра классификационных задач в различных прикладных областях успешно применяется SVM-алгоритм (Support Vector Machines, SVM), осуществляющий обучение по прецедентам («обучение с учителем») и входящий в группу граничных алгоритмов и методов классификации [4, 13]. Классический (базовый) SVM-алгоритм реализует построение бинарного SVM-классификатора.

SVM-классификаторы применяются при решении широкого спектра прикладных задач, в частности, для анализа кредитного риска [15], в медицинской диагностике [11], для распознавания рукописных символов (почерка) [9], категоризации текстов [14], извлечения информации [7], идентификации изображений пешеходов [10], идентификации изображений лиц [3] и др.

Характерной чертой SVM-классификатора является использование специальной функции, называемой ядром. С помощью этой функции обучающая последовательность данных преобразуется из исходного пространства характеристик в пространство характеристик более высокой размерности, где происходит построение гиперплоскости, разделяющей классы. 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.

Векторы характеристик классифицируемых объектов, ближайшие к параллельным гиперплоскостям, называются опорными векторами. Пример построения разделяющей гиперплоскости в пространстве D-2 приведен на рисунке 1.

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

Существенный интерес при разработке SVM-классификатора представляет решение проблемы, связанной с выбором оптимальных значений таких параметров классификатора: типа функции ядра, значений параметров функции ядра; значения параметра регуляризации  $C$  – они заранее задаются пользователем и не изменяются в процессе обучения [1, 5]. Без адекватного качественного решения данной проблемы невозможно обеспечить получение высокоточных результатов классификации данных с применением SVM-классификатора. Значения параметров SVM-классификатора будем считать оптимальными, если достигнута высокая точность классификации: количество ошибок на обучающем и тестовом наборах минимально, причем количество ошибок обученного алгоритма на объектах тестовой выборки не сильно отличается от количества ошибок на обучающей выборке – во избежание «переобучения» классификатора.

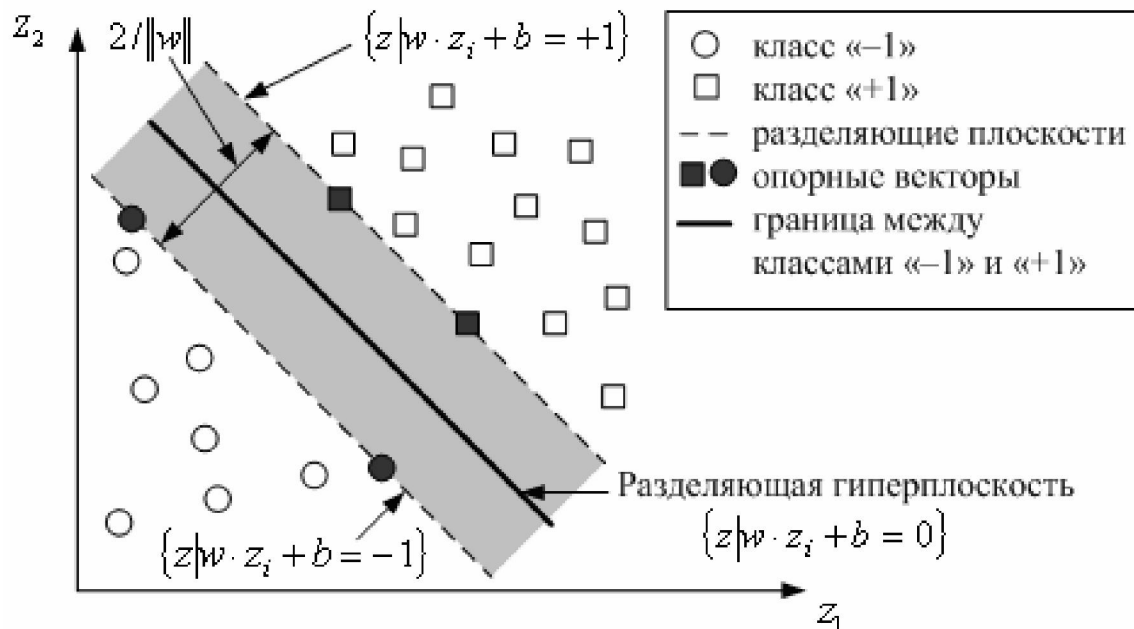


Рис. 1 Линейное разделение двух классов SVM-классификатором в пространстве D-2

В простейшем случае решение данной проблемы осуществляется посредством перебора возможных типов функции ядра, значений параметров функции ядра и значения параметра регуляризации  $C$  – это требует значительных вычислительных затрат. Для оценки качества классификации могут быть использованы показатели точности, полноты классификации и др. [1, 15].

В большинстве случаев при построении бинарных классификаторов целевая функция является сложной, многоэкстремальной, многопараметрической. Для поиска ее оптимума не применимы градиентные методы, но могут быть использованы поисковые алгоритмы стохастической оптимизации.

ции [5, 8, 15] (генетический алгоритм, иммунный алгоритм клональной селекции, алгоритм роя пчел, алгоритм колонии муравьев, алгоритм роя частиц и т.п.). В таких алгоритмах поиск оптимума производится сразу по всему пространству возможных решений.

Алгоритм роя частиц (Particle Swarm Optimization, PSO-алгоритм), основанный на идее о возможности решения задач оптимизации с помощью моделирования поведения групп животных, является наиболее простым эволюционным алгоритмом оптимизации, поскольку для его реализации необходимо уметь определять только значение оптимизируемой функции [12].

Традиционный подход к применению PSO-алгоритма заключается в его многократном применении при фиксированном типе функции ядра – с целью выбора оптимальных значений параметров функции ядра и значения параметра регуляризации  $C$ . Затем осуществляется выбор лучшего типа функции ядра, соответствующих ему значений параметров функции ядра и значения параметра регуляризации  $C$ .

Наряду с традиционным подходом к применению PSO-алгоритма предлагается применять его новый, модифицированный подход. Он реализует одновременный поиск лучшего типа функции ядра, значений параметров функции ядра и значения параметра регуляризации  $C$  [1, 5]. В связи с этим цель данной работы заключается в разработке PSO-алгоритма для решения задачи построения SVM-классификатора в соответствии с предложенным модифицированным подходом и проведении сравнения традиционного и предлагаемого PSO-алгоритмов – как по времени поиска оптимальных параметров SVM-классификатора, так и по качеству классификации данных.

В дальнейшем алгоритмы роя частиц, соответствующие традиционному и модифицированному подходам, будем называть традиционным PSO-алгоритмом и модифицированным PSO-алгоритмом соответственно.

**Теоретические основы SVM-классификации.** Пусть имеется учебный набор вида:  $\{(z_1, y_1), \dots, (z_s, y_s)\}$ , в котором каждому объекту  $z_i \in Z$  поставлено в соответствие число  $y_i \in Y = \{-1; +1\}$ , принимающее значение  $-1$  или  $+1$ , в зависимости от того, какому классу принадлежит объект  $z_i$ . При этом предполагается, что каждому объекту  $z_i$  поставлен в соответствие  $q$  – мерный вектор числовых значений характеристик  $z_i = (z_i^1, z_i^2, \dots, z_i^q)$  (обычно нормированный значениями из отрезка  $[0, 1]$ ), где  $z_i^l$  – числовое значение  $l$ -ой характеристики для  $i$ -го объекта ( $i = \overline{1, s}$ ,  $l = \overline{1, q}$ ) [2, 6]. При построении SVM-классификатора с помощью специальной функции  $\kappa(z_i, z_\tau)$ , называемой ядром, определяется классифицирующая функция (классификатор)  $F: Z \rightarrow Y$ , сопоставляющая классу  $Y = \{-1; +1\}$  произвольный объект из  $Z$ .

Для обучения SVM-классификатора необходимо определить тип функции ядра  $\kappa(z_i, z_\tau)$ , значения параметров ядра и значение параметра регуляризации  $C$ , позволяющего найти компромисс между максимизацией ширины полосы, разделяющей классы, и минимизацией суммарной ошибки. При этом в качестве функции ядра  $\kappa(z_i, z_\tau)$ , позволяющей разделить объекты разных классов, обычно используется одна из следующих функций [13, 15]:

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

где  $\langle z_i, z_\tau \rangle$  – скалярное произведение векторов  $z_i$  и  $z_\tau$ ;  $d$  [ $d \in N$  (по умолчанию  $d = 3$ )],  $\sigma$  [ $\sigma > 0$  (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;  $\text{th}$  – гиперболический тангенс.

При разработке SVM-классификатора необходимо сделать следующее.

1. Разделить учебный набор на обучающую и тестовую выборки, состоящие соответственно из  $S$  и « $S - S$ » элементов ( $S > S$ ).

2. Определить входные параметры классификатора: тип функции ядра  $\kappa(z_i, z_\tau)$ , значения параметров ядра и значение параметра регуляризации  $C$ .

3. Реализовать многократное обучение и тестирование на разных случайным образом сформированных обучающей и тестовой выборках с последующим определением лучшего SVM-классификатора. Тестовая выборка составляет от 1/10 до 1/3 от всего учебного набора. Она не участвует в настройке параметров классификатора, а используется только для проверки его точности.

В результате обучения SVM-классификатора определяется разделяющая гиперплоскость, 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  $z$  [4, 13]. Условие  $-1 < \langle w, z \rangle + b < 1$  задает полосу, которая разделяет классы. Чем шире эта полоса, тем увереннее можно классифицировать объекты. При этом объекты, ближайšie к разделяющей гиперплоскости, называются опорными векторами. Они расположены точно на границах полосы разделяющей классы и несут всю информацию об их разделении.

В случае линейной разделимости классов можно выбрать гиперплоскости таким образом, чтобы между ними не лежал ни один объект из обучающей выборки. Затем максимизировать расстояние между гиперплоскостями (ширину полосы)  $2/\langle w, w \rangle$ , решив задачу квадратичной оптимизации [13]:

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

При решении задачи классификации объектов приходится выполнять построение SVM-классификатора как в случае линейной разделимости классов, так и в случае линейной неразделимости классов. Поэтому, с учетом теоремы Куна – Таккера, в общем случае задача построения разделяющей гиперплоскости может быть переформулирована как двойственная задача поиска седловой точки функции Лагранжа. Она сводится к задаче квадратичного программирования, содержащей только двойственные переменные [13]:

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

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

Если задача (1) решена, то классификация некоторого объекта  $z$  может быть выполнена с использованием следующего правила:

$$F(z) = \text{sign} \left( \sum_{i=1}^S \lambda_i \cdot y_i \cdot \kappa(z_i, z) + b \right). \quad (2)$$

Реально суммирование в правиле (2) выполняется лишь по опорным векторам (то есть по объектам из обучающей выборки, для которых  $\lambda_i \neq 0$ ).

Основная проблема, возникающая при обучении SVM-классификатора, связана с отсутствием рекомендаций по выбору значения параметра регуляризации  $C$ ; функции, описывающей ядро  $\kappa(z_i, z_\tau)$ ; значений параметров самой функции ядра, при которых будет обеспечена высокая точность классификации данных. Эта проблема может быть решена с применением оптимизационных алгоритмов, в частности, с использованием PSO-алгоритма.

**PSO-алгоритм оптимизации.** При реализации традиционного PSO-алгоритма  $n$ -мерное пространство поиска ( $n$  – количество параметров, подлежащих оптимизации) населяется роем из  $m$  агентов-частиц (элементарных решений). Положение (позиция)  $i$ -ой частицы задается вектором  $x_i = (x_i^1, x_i^2, \dots, x_i^n)$ , который определяет некоторый набор значений параметров оптимизации. При этом в аналитической записи целевой функции  $f(x) = f(x^1, x^2, \dots, x^n)$  алгоритма оптимизации (оптимум – это, например, минимум которой необходимо найти) такие параметры могут как присутствовать в явном виде, так и отсутствовать.

В процессе инициализации роя частицы случайным образом располагаются по всей области поиска. При этом каждая  $i$ -ая частица ( $i = \overline{1, m}$ ) имеет свой собственный вектор скорости  $v_i \in R^n$ , который в каждый конкретный момент времени, соответствующей некоторой итерации PSO-алгоритма, определенным образом влияет на значения координат позиции  $i$ -ой частицы ( $i = \overline{1, m}$ ). Эти координаты в  $n$ -мерном пространстве поиска однозначно определяют значение целевой функции  $f(x_i) = f(x_i^1, x_i^2, \dots, x_i^n)$ , которое является некоторым решением задачи оптимизации [12].

Для каждой позиции  $n$ -мерного пространства поиска, в которой побывала  $i$ -ая частица ( $i = \overline{1, m}$ ), выполняется вычисление значения целевой функции  $f(x_i)$ . При этом каждая  $i$ -ая частица запоминает, какое лучшее значение целевой функции лично она нашла, а также координаты позиции в  $n$ -мерном пространстве, соответствующие этому значению целевой функции.

Кроме того, каждая  $i$ -ая частица ( $i = \overline{1, m}$ ) «знает», где расположена позиция, являющаяся лучшей (с точки зрения достижения оптимума целевой функции) среди всех позиций, которые «разведали» (опробовали) частицы – благодаря этому имитируется мгновенный обмен информацией между всеми частицами роя.

На каждой итерации частицы корректируют свою скорость, так чтобы, с одной стороны, быть поближе к лучшей позиции, которую частица нашла сама, и в то же время приблизиться к позиции, которая в данный момент является глобально лучшей (среди совокупности позиций, найденных всеми частицами). Через некоторое количество итераций частицы должны собраться вблизи наиболее хорошей позиции (глобально лучшей по результатам всех итераций). Однако возможно, что часть частиц роя останется где-то в относительно неплохом локальном оптимуме или нескольких таких оптимумах.

Сходимость PSO-алгоритма зависит от того, каким образом выполняется коррекция векторов скоростей частиц. Известны различные подходы к выполнению коррекции вектора скорости  $V_i$  для  $i$ -ой частицы ( $i = \overline{1, m}$ ) [12].

В традиционном PSO-алгоритме коррекция  $j$ -ой координаты вектора скорости ( $j = \overline{1, n}$ ) для  $i$ -ой частицы ( $i = \overline{1, m}$ ) производится по формуле [12]:

$$v_i^j = v_i^j + \hat{\phi} \cdot \hat{r} \cdot (\hat{x}_i^j - x_i^j) + \tilde{\phi} \cdot \tilde{r} \cdot (\tilde{x}^j - x_i^j), \quad (3)$$

где  $v_i^j$  –  $j$ -ая координата вектора скорости  $i$ -ой частицы;  $x_i^j$  –  $j$ -ая координата вектора  $x_i$ , задающего позицию  $i$ -ой частицы;  $\hat{x}_i^j$  –  $j$ -ая координата вектора лучшей позиции, найденного  $i$ -ой частицей за все время ее существования;  $\tilde{x}^j$  –  $j$ -ая координата глобально лучшей позиции всего роя частиц, в которой целевая функция имеет оптимальное значение;  $\hat{r}$  и  $\tilde{r}$  – случайные числа в интервале (0, 1), которые вносят элемент стохастичности в процесс поиска;  $\hat{\phi}$  и  $\tilde{\phi}$  – личный и глобальный коэффициенты ускорения частиц – они являются константами и определяют поведение и эффективность PSO-алгоритма в целом.

С помощью личного и глобального коэффициентов ускорения частиц в (3) масштабируются случайные числа  $\hat{r}$  и  $\tilde{r}$ . При этом глобальный коэффициент ускорения  $\tilde{\phi}$  управляет воздействием глобальной лучшей позиции на скорости всех частиц (групповое управление), а личный коэффициент ускорения  $\hat{\phi}$  – воздействием личной лучшей позиции на скорость этой частицы.

В настоящее время известны различные версии традиционного PSO-алгоритма. В одной из самых распространенных (канонической версии) предлагается выполнять нормировку коэффициентов ускорения  $\hat{\varphi}$  и  $\tilde{\varphi}$  – для того, чтобы сходимость алгоритма не так сильно зависела от выбора их значений [12]. При этом коррекция каждой  $j$ -ой координаты вектора скорости ( $j = \overline{1, n}$ )  $i$ -ой частицы ( $i = \overline{1, m}$ ) производится в соответствии с формулой:

$$v_i^j = \chi \cdot [v_i^j + \hat{\varphi} \cdot \hat{r} \cdot (\hat{x}_i^j - x_i^j) + \tilde{\varphi} \cdot \tilde{r} \cdot (\tilde{x}^j - x_i^j)], \quad (4)$$

где  $\chi$  – коэффициента сжатия;

$$\chi = 2 \cdot K / |2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi}|; \quad (5)$$

$$\varphi = \hat{\varphi} + \tilde{\varphi} \quad (\varphi > 4);$$

$K$  – некоторый масштабирующий коэффициент, принимающий значения из интервала (0, 1).

При использовании формулы (4) для коррекции вектора скорости гарантируется сходимость PSO-алгоритма и нет необходимости в явном контроле скорости частиц [12].

Пусть коррекция вектора скорости  $i$ -ой частицы ( $i = \overline{1, m}$ ) выполнена в соответствии с формулой (3) или (4). Тогда коррекция  $j$ -ой координаты позиции  $i$ -ой частицы ( $i = \overline{1, m}$ ) выполняется по формуле:

$$x_i^j = x_i^j + v_i^j.$$

Далее для каждой  $i$ -ой частицы ( $i = \overline{1, m}$ ) рассчитывается новое значение целевой функции  $f(x_i)$  и выполняется проверка: не стала ли новая позиция с вектором координат  $x_i$  лучшей среди всех позиций, в которых  $i$ -ая частица ранее побывала. Если новая позиция  $i$ -ой частицы признается лучшей для нее на текущий момент времени, то информация об этой позиции сохраняется в векторе  $\hat{x}_i$  ( $i = \overline{1, m}$ ) – с «запоминанием» значения целевой функции  $f(x_i)$  в этой позиции. Затем среди всех новых позиций частиц роя осуществляется проверка на наличие глобально лучшей позиции. Если некоторая новая позиция, соответствующая одной из частиц роя, признается глобально лучшей на текущий момент времени, то информация о ней сохраняется в векторе  $\tilde{x}$  – с «запоминанием» значения целевой функции в этой позиции.

**Традиционный и модифицированный PSO-алгоритмы в задаче разработки SVM-классификатора.** В случае использования PSO-алгоритма при разработке SVM-классификатора частицам роя могут быть сопоставлены векторы, описывающие их позиции в пространстве поиска и закодированные параметрами функции ядра и параметром регуляризации:  $(x_i^1, x_i^2, C_i)$ , где  $i$  – номер частицы ( $i = \overline{1, m}$ );  $C_i$  – параметр регуляризации;  $x_i^1, x_i^2$  – параметры функции ядра  $i$ -ой частицы. При этом параметр  $x_i^1$  полагается равным параметрам функций ядра  $d$ ,  $\sigma$  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.

Тогда традиционный подход к применению PSO-алгоритма при разработке SVM-классификатора заключается в многократном применении PSO-алгоритма при фиксированном типе функции ядра – с целью выбора оптимальных значений параметров функции ядра и значения параметра регуляризации. Такой подход предполагает выполнение следующей последовательности шагов [5].

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, участвующие 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  $T = 2$ ).

Шаг 2. Для каждого выбранного на шаге 1 типа функции ядра  $T$  сгенерировать начальное положение  $i$ -ой частицы ( $i = \overline{1, m}$ ) с помощью случайного вектора  $(x_i^1, x_i^2, C_i)$ , где  $x_i^1 \in [x_{\min}^{1T}, x_{\max}^{1T}]$ ,  $x_i^2 \in [x_{\min}^{2T}, x_{\max}^{2T}]$  ( $x_i^2 = 0$  при  $T = 1$  и  $T = 2$ ),  $C_i \in [C_{\min}^T, C_{\max}^T]$ ; инициализировать случайный вектор скорости  $v_i(v_i^1, v_i^2, v_i^3)$   $i$ -ой частицы ( $i = \overline{1, m}$ ) ( $v_i^2 = 0$  при  $T = 1$  и  $T = 2$ ). Принять начальное положение  $i$ -ой частицы ( $i = \overline{1, m}$ ) за лучшее ее известное положение  $(\hat{x}_i^1, \hat{x}_i^2, \hat{C}_i)$  и определить лучшую частицу среди всех частиц  $(\tilde{x}^1, \tilde{x}^2, \tilde{C})$  для рассматриваемого типа функции ядра  $T$ . После чего  $N_{\max}$  раз для каждой  $i$ -ой частицы ( $i = \overline{1, m}$ ) выполнить:

- коррекцию вектора скорости  $v_i(v_i^1, v_i^2, v_i^3)$   $i$ -ой частицы и ее положения  $(x_i^1, x_i^2, C_i)$  по формулам:

$$v_i^j = \begin{cases} \chi \cdot [v_i^j + \hat{p} \cdot \hat{r} \cdot (\hat{x}_i^j - x_i^j) + \tilde{p} \cdot \tilde{r} \cdot (\tilde{x}^j - x_i^j)], & j = 1, 2, \\ \chi \cdot [v_i^j + \hat{p} \cdot \hat{r} \cdot (\hat{C}_i - C_i) + \tilde{p} \cdot \tilde{r} \cdot (\tilde{C} - C_i)], & j = 3, \end{cases} \quad (6)$$

$$x_i^j = x_i^j + v_i^j \text{ для } j = 1, 2 \quad (7)$$

$$C_i = C_i + v_i^3, \quad (8)$$

где  $\hat{r}$  и  $\tilde{r}$  – случайные числа в интервале  $(0, 1)$ ,  $\chi$  – коэффициент сжатия, рассчитанный по формуле (5);

- расчет точности SVM-классификатора с параметрами  $(x_i^1, x_i^2, C_i)$  для рассматриваемого типа функции ядра  $T$  с целью поиска оптимальной комбинации  $(\tilde{x}^1, \tilde{x}^2, \tilde{C})$ , обеспечивающей высокое качество классификации.

В результате для каждого типа  $T$  функции ядра, участвующего в поиске, будет определена частица с оптимальной комбинацией значений параметров  $(\tilde{x}^1, \tilde{x}^2, \tilde{C})$ , обеспечивающая высокое качество классификации при использовании соответствующего типа функции ядра  $T$ .

Шаг 3. Выбрать из полученных для каждого типа функции ядра  $T$ , включенного в поиск, те значения параметров  $(\tilde{x}^1, \tilde{x}^2, \tilde{C})$  SVM-классификатора и соответствующий тип функции ядра  $\tilde{T}$ , при которых качество классификации оказалось максимальным (наилучшим).

Наряду с традиционным подходом к применению PSO-алгоритма при разработке SVM-классификатора предлагается применять новый подход [1, 5], реализующий одновременный поиск лучшего типа функции ядра  $\tilde{T}$ , значений параметров  $\tilde{x}^1$  и  $\tilde{x}^2$  функции ядра, значения параметра регуляризации  $\tilde{C}$ . При таком подходе каждой  $i$ -ой частице роя ( $i = \overline{1, m}$ ) соответствует 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); параметры  $x_i^1, x_i^2, C_i$  определяются аналогично предыдущему случаю. При этом 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. 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. Предложенная модификация PSO-алгоритма может быть представлена следующей последовательностью шагов [5].

Шаг 1. Аналогичен шагу 1 в описанном выше алгоритме. При этом дополнительно надо определить процент «перерождения» частиц  $p$ .

Шаг 2. Задать равное количество частиц для каждого ядра  $T$ , включенного в поиск. Затем для каждой  $i$ -ой 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  $i$ -ой 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  $v_i(v_i^1, v_i^2, v_i^3)$   $i$ -ой частицы ( $i = \overline{1, m}$ ) ( $v_i^2 = 0$  при  $T = 1$  и  $T = 2$ ). Принять начальное положение  $i$ -ой 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  $(\bar{T}, \bar{x}^{1T}, \bar{x}^{2T}, \bar{C}^T)$ . При этом количество выполненных итераций полагается равным 1.

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

- «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  $\tilde{T}$ ; изменить значения параметров  $x_i^1, x_i^2, C_i$  «перерождаемой» частицы так, чтобы они соответствовали новому типу ядра  $\tilde{T}$  (то есть попадали в соответствующие диапазоны);
- выполнить коррекцию вектора скорости  $v_i(v_i^1, v_i^2, v_i^3)$   $i$ -ой частицы ( $i = \overline{1, m}$ ) по формуле:

$$v_i^j = \begin{cases} \chi \cdot [v_i^j + \hat{\phi} \cdot \hat{r} \cdot (\hat{x}_i^j - x_i^j) + \tilde{\phi} \cdot \tilde{r} \cdot (\bar{x}^{jT} - x_i^j)], & j=1, 2, \\ \chi \cdot [v_i^j + \hat{\phi} \cdot \hat{r} \cdot (\hat{N}_i - \tilde{N}_i) + \tilde{\phi} \cdot \tilde{r} \cdot (\bar{C}^T - \tilde{N}_i)], & j=3, \end{cases} \quad (9)$$

где  $\hat{r}$  и  $\tilde{r}$  – случайные числа в интервале (0, 1),  $\chi$  – коэффициент сжатия, рассчитанный по формуле (5),  $\bar{x}^{1T}, \bar{x}^{2T}, \bar{C}^T$  – координаты частицы, лучшей для типа функции ядра  $T = T_i$ ; при этом формула (9) является модификацией формулы (6): вместо значений координат глобально-лучшей частицы  $\tilde{x}^1, \tilde{x}^2, \tilde{C}$  используются значения  $\bar{x}^{1T}, \bar{x}^{2T}, \bar{C}^T$ ;

- изменить положение  $(x_i^1, x_i^2, C_i)$   $i$ -ой частицы в соответствии с формулами (7)–(8);
- 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  $(\tilde{T}, \tilde{x}^1, \tilde{x}^2, \tilde{C})$ , обеспечивающей высокое качество классификации;
- увеличить количество итераций на «1».

После выполнения данного алгоритма будет определена частица с оптимальной комбинацией значений параметров  $(\tilde{T}, \tilde{x}^1, \tilde{x}^2, \tilde{C})$ , 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 воз-



можно, что у некоторых частиц «перерождение» ядер не произойдет, и они останутся в своем пространстве поиска.

Использование такого подхода к применению PSO-алгоритма в задаче разработки SVM-классификатора позволяет снизить временные затраты на построение искомого SVM-классификатора.

Оценка качества SVM-классификатора может быть выполнена с применением различных показателей качества классификации [1]. Это могут быть показатель, учитывающий данные кросс-проверки; показатели точности и полноты классификации; показатель, основанный на анализе ROC-кривой и др.

**Сравнительный анализ эффективности SVM-классификаторов, построенных с применением традиционного и модифицированного PSO-алгоритмов.** Целесообразность использования модифицированного PSO-алгоритма при разработке SVM-классификатора была подтверждена на тестовых и реальных данных. При проведении эксперимента для конкретного набора данных производился запуск традиционного PSO-алгоритма и предложенного модифицированного PSO-алгоритма – с последующим сравнением найденных ими оптимальных значений параметров SVM-классификатора; точности классификации, проводимой с учетом найденных значений; времени, затраченного на поиск.

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 и набор данных для кредитного скоринга.

1. Выборка данных хирургического отделения университета штата Висконсин о диагностике рака молочной железы 569 пациентов, содержащая информацию о клеточных ядрах, описываемых 30-ю характеристиками ( $q = 30$ ). Она была получена на основе оцифрованных изображений (в таблице набор WDBC, источник <http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/>). При этом у 212 пациентов (класс 1) была правильно диагностирована злокачественная опухоль, а у 357 пациентов (класс 2) диагноз не был подтвержден.

2. Выборка данных о диагностике болезни сердца у 270 пациентов, описываемых 13 характеристиками ( $q = 13$ , в таблице набор Heart, источник <http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/heart/>). При этом у 150 пациентов (класс 1) заболевание было диагностировано правильно, а у 120 пациентов (класс 2) диагноз не был подтвержден.

3. Выборка данных о заявках на потребительские кредиты в Австралии, содержащая информацию о 690 заявках, описываемых 14 характеристиками ( $q = 14$ , в таблице набор Australian, источник <http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/australian/>). При этом 382-м заявкам (класс 1) было отказано в выдаче кредита, а для 308 заявок (класс 2) кредит был одобрен.

Кроме того, для экспериментальных исследований были использованы две тестовые выборки данных: Test [2, 6] и MOTP12 (источник [http://machinelearning.ru/wiki/images/b/b2/MOTP12\\_svm\\_example.rar](http://machinelearning.ru/wiki/images/b/b2/MOTP12_svm_example.rar)).

На всех наборах данных имеет место случай бинарной классификации.

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, радиальной базисной и сигмоидной функциями. Для них были установлены одинаковые значения параметров PSO-алгоритма и одинаковые диапазоны изменения значений искоемых параметров SVM-классификатора.

Экспериментальные вычисления производились на ПЭВМ, работающей под 64-разрядной версией Windows 7, с оперативной памятью 3 Гб и одноядерным процессором Intel® Core™ i3 с тактовой частотой ядра 2,53 ГГц. При использовании многоядерных процессоров очевидна возможность реализации распараллеливания вычислений, что позволит увеличить скорость выполнения расчетов.

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 парапараметрах 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 (значение номер 2). При этом для традиционного PSO-алгоритма оптимальными значениями параметра ядра и параметра регуляризации оказались соответственно значения  $\sigma = 6,81$  и  $C = 4,93$ , а для модифицированного PSO-алгоритма – соответственно значения  $\sigma = 4,01$  и  $C = 9,83$ .

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 (сек.)
				номер ядра	$C$	$x_1$	$x_2$	при обучении	при тесте			
WDBC	569	30	традиц.	2	4,93	6,81	–	5 из 427	0 из 142	66	99,12	10108
			модиф.	2	9,83	4,01	–	1 из 427	1 из 142	80	99,65	3250
Heart	270	13	традиц.	2	8,51	3,18	–	7 из 192	11 из 78	106	93,33	6558
			модиф.	2	5,92	2,89	–	4 из 192	10 из 78	99	94,81	2733
Australian	690	14	традиц.	2	6,79	2,31	–	21 из 492	28 из 198	237	92,9	16018
			модиф.	2	7,64	2,38	–	21 из 492	26 из 198	225	93,19	8013
MOTPH12	400	2	традиц.	2	8,10	0,18	–	9 из 340	9 из 60	166	95,50	15697
			модиф.	2	6,37	0,26	–	10 из 340	8 из 60	107	95,50	9145
Test	300	2	традиц.	1	8,85	3	–	0 из 240	0 из 60	7	100	3433
			модиф.	1	9,24	3	–	0 из 240	0 из 60	6	100	648

Точность классификации, достигнутая с применением традиционного PSO-алгоритма, составила 99,12 %, а с применением модифицированного PSO-алгоритма – 99,65 %. При этом время поиска составило соответственно 10108 и 3250 секунды.

Для набора Heart на рисунках 2–4 показаны примеры расположения роя частиц в пространствах поиска D-2 и D-3 в момент инициализации, а также на 3-ей и 12-ой итерациях соответственно – при поиске с применением модифицированного PSO-алгоритма. В поиск были включены ядра с полиномиальной, радиальной базисной и сигмоидной функциями, для которых были установлены следующие диапазоны изменения значений параметров:  $3 \leq d \leq 8$ ,  $d \in N$  (для полиномиальной функции);  $0,1 \leq \sigma \leq 10$  (для радиальной базисной функции);  $-10 \leq k_2 \leq -0,1$  и  $0,1 \leq k_1 \leq 10$  (для сигмоидной функции). Для параметра регуляризации  $C$  диапазон изменения был определен как:  $0,1 \leq C \leq 10$ . Кроме того, были заданы следующие значения параметров PSO-алгоритма: количество частиц  $m$  в рое равное 600 (по 200 на каждый тип функции ядра); количество итераций  $N_{\max} = 20$ ; коэффициенты личного и глобального ускорения, равные соответственно  $\hat{\varphi} = 2$  и  $\tilde{\varphi} = 5$ ; масштабирующий коэффициент  $K = 0,3$ ; показатель «перерождения» частиц  $p = 20\%$ .

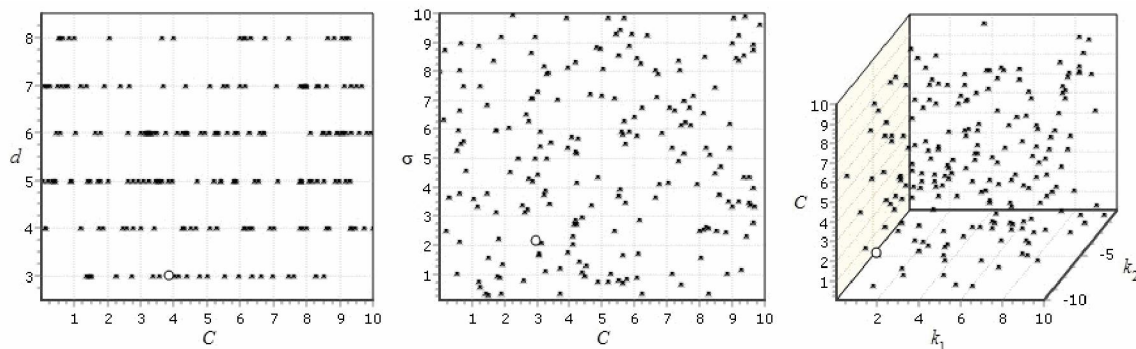


Рис. 2. Расположение частиц в рое в момент инициализации (слева – полиномиальная, в центре – радиальная базисная, справа – сигмоидная функция ядра)

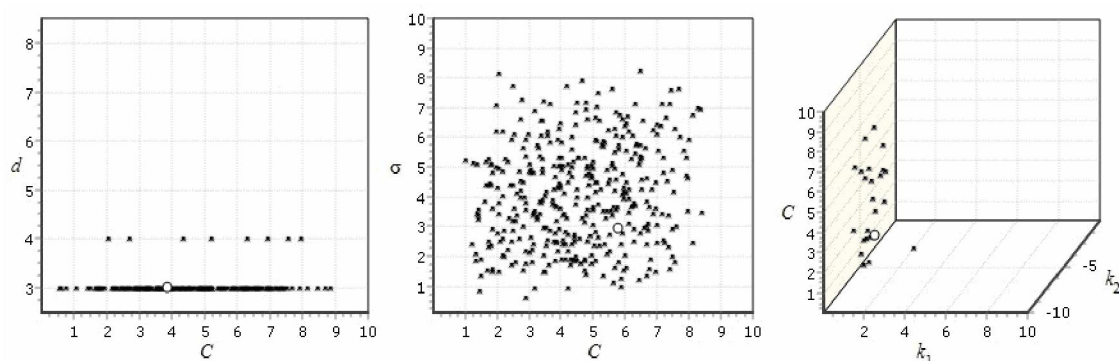


Рис. 3. Расположение частиц в рое на 3-ей итерации  
 (слева – полиномиальная, в центре – радиальная базисная, справа – сигмоидная функция ядра)

В пространствах поиска сами частицы помечены маркерами-звездочками, а лучшая позиция в пространстве поиска – белым круглым маркером. Как видно из этих рисунков, в ходе реализации модифицированного PSO-алгоритма частицы роя в собственном пространстве поиска движутся к некоторой лучшей (оптимальной) для текущей итерации позиции, демонстрируя коллективный поиск лучшей позиции в ареале обитания. При этом корректируются скорость и направление движения каждой частицы. Помимо этого происходит «перерождение» частиц – у некоторых из них изменяется пространство поиска на то пространство, в котором частицы показывают максимально высокое качество классификации.

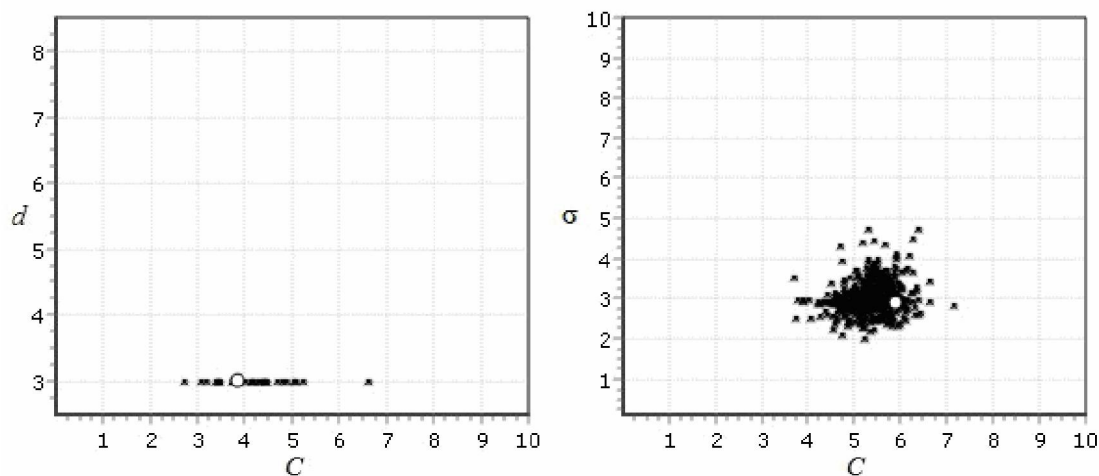


Рис. 4. Расположение частиц в рое на 12-ой итерации  
 (слева – радиальная базисная, справа – сигмоидная функция ядра)

Кроме того, в ходе реализации модифицированного PSO-алгоритма происходит не только изменение координат частиц, отвечающих за параметры функции ядра  $K(z_i, z_t)$  и параметр регуляризации  $C$ , но изменяется и тип функции ядра. В результате этого все частицы устремляются в единое пространство поиска (в данном случае – пространство, соответствующее радиальной функции ядра). При этом они удаляются (убираются) из того пространства, в котором были изначально инициализированы.

В рассматриваемом примере после 20 итерации только у 7 частиц не изменился тип функции ядра, остальные же частицы расположились около лучшей позиции в пространстве поиска, отвечающей за найденное оптимальное решение (рис. 5).

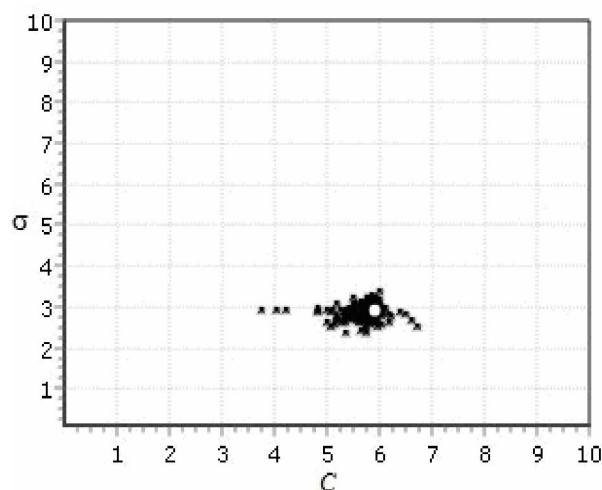


Рис. 5 Расположение частиц после 20-ой итерации

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.

С целью проведения последующей классификации новых объектов для каждого из рассмотренных наборов данных реализовано сохранение найденных с помощью PSO-алгоритма (традиционного и модифицированного) типа функции ядра, параметров ядра, параметра регуляризации и координат опорных векторов, которые однозначно определяют гиперплоскость, разделяющую объекты на классы.

**Выводы.** Результаты экспериментальных исследований, полученные на основе тестовых данных, традиционно используемых для оценки качества классификации, подтверждают перспективность использования модифицированного PSO-алгоритма при разработке SVM-классификатора для решения задачи выбора оптимальных типа функции ядра, значений параметров функции ядра, а также значения параметра регуляризации, обеспечивающих высокое качество классификации объектов при приемлемых временных затратах. Предложенный подход к решению задачи разработки SVM-классификатора может быть использован в различных прикладных областях, где возникает необходимость проведения декомпозиции объектов или выявления некоторого явления по представленным характеристикам объектов наблюдений.

#### Список литературы

1. Демидова Л. А. Аспекты применения алгоритма роя частиц в задаче разработки SVM-классификатора / Л. А. Демидова, Ю. С. Соколова // Вестник Рязанского государственного радиотехнического университета. – 2015. – № 53. – С. 84–92.
2. Демидова Л. А. Использование SVM-алгоритма для уточнения решения задачи классификации объектов с применением алгоритмов кластеризации / Л. А. Демидова, Ю. С. Соколова // Вестник Рязанского государственного радиотехнического университета. – 2015. – № 51. – С. 103–113.
3. Bouzalmat A. Face Recognition Using SVM Based on LDA / A. Bouzalmat, J. Kharroubi, A. Zarghili // IJCSI International Journal of Computer Science Issues. – 2013. – Vol. 10, issue 4, № 1. – P. 171–179.
4. Chapelle O. Choosing Multiple Parameters for Support Vector Machine / O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee // Machine Learning. – 2002. – № 46 (1–3). – P. 131–159.
5. Demidova L. Modification of Particle Swarm Algorithm for the Problem of the SVM Classifier Development / L. Demidova, Yu. Sokolova // Stability and Control Processes: International Conference in Memory of V.I. Zubov. – 2015. – P. 619–622.
6. Demidova L. Use of Fuzzy Clustering Algorithms' Ensemble for SVM Classifier Development / L. Demidova, Yu. Sokolova, E. Nikulchev // International Review on Modelling and Simulations (IREMOS). – 2015. – Vol. 8, № 4. – P. 446–457.



7. Kuspriyanto. Performance Evaluation of SVM-Based Information Extraction using  $\tau$  Margin Values / Kuspriyanto, O. S. Santoso, D. H. Widyantoro, H. S. Sastramihardja, K. Muludi, S. Maimunah // *International Journal on Electrical Engineering and Informatics*. – 2010. – Vol. 2, № 4 – P. 256–265.
8. Lessmann S. Genetic Algorithms for Support Vector Machine Model Selection / S. Lessmann, R. Stahlbock, S. F. Crone // 2006 International Joint Conference on Neural Networks Sheraton Vancouver Wall Centre Hotel. – Vancouver, BC, Canada, July 16–21, 2006. – P. 3063–3069.
9. Neves R. F. P. An Efficient Way Combining SVMs for Handwritten Digit Recognition / R. F. P. Neves, C. Zanchettin, A. N. G. Lopes Filho // *Artificial Neural Networks and Machine Learning – ICANN 2012*. – 2012. – Vol. 7553. – P. 229–237.
10. Nisha A. S. PART – based pedestrian detection and tracking using HOG-SVM classification / A. S. Nisha, K. Thulasimani // *International Journal of Computer Science and Mobile Applications*. – 2014. – Vol. 2, № 1. – P. 142–155.
11. Raikwal J. S. Performance Evaluation of SVM and K-Nearest Neighbor Algorithm over Medical Data set / J. S. Raikwal, K. Saxena // *International Journal of Computer Applications (0975–8887)*. – 2010. – Vol. 50, № 14. – P. 35–39.
12. Sun Jun. Particle Swarm Optimisation: Classical and Quantum Perspectives / Jun Sun, Choi-Hong Lai, Xiao-Jun Wu. – CRC Pres, 2011. – P. 419.
13. Vapnik V. Statistical Learning Theory / V. Vapnik. – John Wiley & Sons, inc., New York, 1998. – P. 740.
14. Wei L. Text Classification Using Support Vector Machine with Mixture of Kernel / L. Wei, B. Wei, B. Wang // *A Journal of Software Engineering and Applications*. – 2012. – № 5 – P. 55–58.
15. Yu L. Bio-Inspired Credit Risk Analysis. Computational Intelligence with Support Vector Machines / L. Yu, Sh. Wang, K. K. Lai, L. Zhou. – Springer-Verlag Berlin Heidelberg, 2008. – P. 244.

#### References

1. Demidova L. A., Sokolova Yu. S. Aspekty primeneniya algoritma roya chastits v zadache razrabotki SVM-klassifikatora [Application's Aspects of Particle Swarm Algorithm in a problem of the SVM-classifier's development]. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta* [Bulletin of Ryazan State Radio Engineering University], 2015, no. 53, pp. 84–92.
2. Demidova L.A., Sokolova Y.S. Ispolzovanie SVM-algoritma dlya utochneniya pesheniya zadachi klassifikatsii objektov s primeneniem algoritmov klasterizatsii [SVM-algorithm for qualification of objects classification using clustering algorithms]. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta* [Bulletin of Ryazan State Radio Engineering University], 2015, no. 51, pp. 103–113.
3. Bouzalmat A., Kharroubi J., Zarghili A. Face Recognition Using SVM Based on LDA, *IJCSI International Journal of Computer Science Issues*, 2013, vol. 10, issue 4, no. 1, pp. 171–179.
4. Chapelle O., Vapnik V., Bousquet O., Mukherjee S. Choosing Multiple Parameters for Support Vector Machine, *Machine Learning*, 2002, no. 46 (1–3), pp. 131–159.
5. Demidova L., Sokolova Yu. Modification of Particle Swarm Algorithm for the Problem of the SVM-classifier Development, *International Conference «Stability and Control Processes» in Memory of V.I. Zubov*, 2015, pp. 619–622.
6. 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.
7. Kuspriyanto, Santoso O. S., Widyantoro D. H., Sastramihardja H. S., Muludi K., M. Siti. Performance Evaluation of SVM-Based Information Extraction using  $\tau$  Margin Values, *International Journal on Electrical Engineering and Informatics*, 2010, vol. 2, no. 4, pp. 256–265.
8. Lessmann S., Stahlbock R., Crone S.F. Genetic Algorithms for Support Vector Machine Model Selection, *2006 International Joint Conference on Neural Networks Sheraton Vancouver Wall Centre Hotel. Vancouver. BC. Canada*, July 16–21, 2006, pp. 3063–3069.
9. Neves R. F.P., Zanchettin C., Filho A. N. G. Lopes. An Efficient Way Combining SVMs for Handwritten Digit Recognition, *Artificial Neural Networks and Machine Learning – ICANN 2012*, 2012, vol. 7553, pp. 229–237.
10. Nisha A. S., Thulasimani K. PART – based pedestrian detection and tracking using HOG-SVM classification, *International Journal of Computer Science and Mobile Applications*, 2014, vol. 2, no. 1, pp. 142–155.
11. Raikwal J. S., Saxena K. Performance Evaluation of SVM and K-Nearest Neighbor Algorithm over Medical Data set, *International Journal of Computer Applications (0975–8887)*, 2010, vol. 50, no. 14, pp. 35–39.
12. Sun J., Lai Choi-Hong, Wu Xiao-Jun. *Particle Swarm Optimization: Classical and Quantum Perspectives*, CRC Pres, 2011, pp. 419.
13. Vapnik V. *Statistical Learning Theory*, John Wiley & Sons, inc., New York, 1998, pp. 740.
14. Wei L., Wei B., Wang B. Text Classification Using Support Vector Machine with Mixture of Kernel. *A Journal of Software Engineering and Applications*, 2012, no. 5, pp. 55–58.
15. Yu L., Wang S., Lai K. K., Zhou L. *Bio-Inspired Credit Risk Analysis. Computational Intelligence with Support Vector Machines*, Springer-Verlag Berlin Heidelberg, 2008, pp. 244.