Neurocomputing 228 (2017) 187–197

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

PSO-based method for SVM classification on skewed data sets

Jair Cervantes[a,*], Farid Garcia-Lamont[a], Lisbeth Rodriguez-Mazahua[b], Asdrúbal López[c], José Ruiz-Castilla[a], Adrian Trueba[a]

[a] UAEMEX (Autonomous University of Mexico State), Texcoco 56259, Mexico
[b] Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Veracruz 94320, Mexico
[c] UAEMEX (Autonomous University of Mexico State), Zumpango 55600, Mexico

A R T I C L E I N F O

Keywords:
Skew data sets
SVM
Hybrid algorithms

A B S T R A C T

Over the last years, Support Vector Machines (SVMs) have become a successful approach in classification problems. However, the performance of SVMs is affected harshly by skewed data sets. An SVM learns a biased model that affects the performance of the classifier. Furthermore, SVMs are typically unsuccessful on data sets where the imbalanced ratio is very large. Lately, several techniques have been used to tackle this disadvantage by generating artificial instances. Artificial data instances attempt to add information to the minority class. However, the new instances could introduce noise and decrease the performance of the classifier. In this research, an alternative procedure is suggested, the algorithm finds systematically new instances, improving the performance of SVMs on skewed data sets. The proposed method starts obtaining the support vectors (SVs) from the skewed data set. These initial SVs are used to generate new instances and the PSO algorithm is used to evolve the artificial instances, eliminating noise instances. This research combines the best of optimization and classification techniques. To show the ability of the proposed method to improve the performance of SVMs on skewed data sets, we compare the performance of our method against some classical methods and show that our algorithm outperforms all of them on several data sets.

1. Introduction

In the past few years, Support Vector Machines (SVMs) have shown excellent generalization power in classification problems in several application fields [1–6]. In addition to their strong theoretical background and high generalization ability, SVMs have been confirmed as a robust tool for classification and regression in several noisy and complex domains. However, it has been shown that the generalization ability of SVMs drops on skewed data sets [7,8], because SVMs learn a biased model, which affects the classifier performance. Moreover, the performance of SVMs is more affected when the imbalanced ratio is very large. In order to tackle this disadvantage, many algorithms have been proposed to deal with this problem. The most basic step to process imbalanced data sets can be realized by sampling the data set. This step helps to build a better predictive model. There are two main methods that can be used to even-up the classes: under-sampling and over-sampling. Under-sampling, delete instances from the over-represented class, called under-sampling and Over-sampling add copies of instances from the under-represented class. These approaches are often very easy to implement and fast to run. They are an excellent starting point. However, there are many algorithms to improve the performance on imbalanced data sets. These algorithms are categorized into internal and external techniques. External techniques try to balance the data instances before training the classifier [9–18]. On the other hand, internal techniques redesign the architecture of the classification methods [19,8,20,21].

Although there are many classification algorithms for imbalanced data sets in the current literature, the SMOTE (Synthetic Minority Oversampling Technique) algorithm [14] is perhaps one of the most used approaches to improve the performance of classifiers on skewed data sets. The SMOTE algorithm introduces artificial instances in data sets by interpolating feature values based on neighbors. Several studies have shown that SMOTE has a better performance than under-sampling and over-sampling techniques [22–26]. Moreover, SMOTE does not cause information loss and sometimes the algorithm could potentially find hidden minority regions. In the case of skewed data sets, SMOTE could identify similar but more specific regions in the feature space as the decision region for the minority class. Despite its excellent features, SMOTE is limited to increment the density of the sets by introducing instances with limited information because the new

* Corresponding author.
E-mail addresses: jcervantesc@uaemex.mx (J. Cervantes), fglamont@gmail.com (F. Garcia-Lamont), lisbethr08@gmail.com (L. Rodriguez-Mazahua), asdrubalchau@gmail.com (A. López), jsergioruizc@gmail.com (J. Ruiz-Castilla), atruebae@hotmail.com (A. Trueba).

http://dx.doi.org/10.1016/j.neucom.2016.10.041
Received 24 December 2015; Received in revised form 30 September 2016; Accepted 15 October 2016
Available online 01 November 2016
0925-2312/

instances are obtained using a linear combination between positive examples (minority class). The best artificial examples (i.e., instances with more information of each class) are in the region between positive and negative instances. Introducing instances in this region could increment the discriminative information of positive instances, improving the performance of a classifier on imbalanced data sets. However, this external region is very sensitive to artificial instances. The generation of inadequate artificial instances leads to introduction of noise and loss of performance in the classifier. Artificial instances can cause significant differences in performance. Therefore, artificial instances must be generated carefully. Introducing optimal new instances is an important step in the proposed algorithm. However, no general studies are available to introduce the best artificial instances.

Zhang et al. [16] proposed an algorithm to expand the minority class boundary. The algorithm uses a Random Walk Over-Sampling (RWO-Sampling) approach to balance different class samples by creating synthetic samples through randomly walking in the real data. In [27] a Genetic Algorithm (GA) is used for under-sampling the majority class; the algorithm tackles the difficulties of SVM learning on large data sets because the method significantly reduces the size of the training set without losing performance. In [28] a hybrid learning model is proposed to cope with the problem of imbalanced by evolving self-organizing maps. The authors used GA to evolve the subset of the minority examples into a new stage that might discover novel knowledge from the limited and under represented minority class. In [15], a GA is used to balance skewed data sets. The authors argued that the method obtains better results than simple random sampling. García et al. [29] implemented an algorithm which performs an optimized selection of examples from data sets. The learning algorithm is based on the nested generalized exemplar method and GA to generate and select the best suitable data to enhance the classification performance over imbalanced domains. In [30], the authors proposed a classification system in order to detect the most important rules, and the rules which perturb the performance of the classifier. That system uses hierarchical fuzzy rules and a GA. All the studies mentioned before, try to improve the classifier performance by selecting subsets, balancing subsets or evolving artificial examples. However, the searching space could be huge in the most cases, this makes it difficult to find an acceptable solution. In this paper, we present a novel algorithm which uses Particle Swarm Optimization (PSO) in order to generate new examples.

The distinctive contribution of the proposed method is that the artificial instances are obtained from the most critical region for SVMs, called the margin, and evolved to eliminate bad artificial instances. The margin is the distance between the decision boundary and the closest examples with a different label. The proposed algorithm obtains artificial instances from the most important region by identifying the minority samples in the margin region and introducing new artificial examples only in this region. Techniques in the existing literature obtain artificial instances by selecting each sample in the minority class, and then introducing new artificial examples by joining any or all of the k minority class nearest neighbors. The generation of new instances in the minority class can improve the performance of SVM classification [18]. However, it is particularly difficult to introduce good instances in the margin region because this region is extremely sensitive. Introducing artificial instances in this region must be generated carefully. To find the optimal and synthetic instances is an important step in the proposed algorithm. This is the main reason for the combination of PSO and SVMs in this research. In this paper, a hybrid SMOTE-PSO algorithm is proposed in order to improve the performance of SVMs on imbalanced data sets. In SMOTE-PSO, PSO is used to guide the search process of artificial instances that improve the SVM performance. Moreover, the synthetic instances are evolved and improved by following the best particle $\mathbf{p}_{gi}$. Experimental results show that the SMOTE-PSO algorithm can get better performance than traditional models.

The rest of the paper is organized as follows. In Section 2, a brief overview of the related work on SVMs with imbalanced data sets is presented. In Section 3, the PSO algorithm is introduced. Section 4 presents the SMOTE-PSO technique. The results of the experiments are shown in Section 5. Discussion and Conclusions are given in Section 6.

2. Classification on imbalanced data sets

In this section, the problem of imbalanced data sets is introduced, and some algorithms to address this problem are described. The second subsection discusses how SVM classifiers are affected by the imbalance in data sets.

2.1. Addressing the imbalanced problem

Many real-world applications show an imbalance in data sets. The imbalance in data sets occurs when a class contains most instances of the entire data set (negative class), while the other class contains a small fraction of instances (positive class). In these problems, the goal in binary classification problems is to find a function that best generalizes the minority class, which is usually the most important. Traditionally, the performance of classical classification methods is low on imbalanced data sets, because they were not designed to address such problems. There are several techniques to deal with the challenges of imbalanced data sets. These techniques can be categorized into internal and external techniques. The most widely used external techniques are under-sampling and over-sampling. Under-sampling gets the number of instances m in the minority class $X^+ = \{x_i\}_{i=1}^m$ and randomly selects m instances in the majority class $X^- = \{x_i\}_{i=1}^p$. The over-sampling technique tries to reduce the imbalance by replicating data instances in the minority class or generating artificial instances in the minority class.

In the past few years, the Synthetic Minority Over sampling Technique (SMOTE) algorithm [14] has been one of the most popular techniques to generate artificial instances. The SMOTE algorithm generates artificial instances by over-sampling the minority class, this is achieved taking each minority class instance and generating synthetic instances along the line segments by joining any or all the k minority class nearest neighbors. It does not cause any information loss and could potentially find hidden minority regions. However, in some cases the SMOTE algorithm could introduce noise in the data set reducing the performance of the classifier. This is because SMOTE makes the assumption that the instance between a positive class instance and its nearest neighbor is also positive [17].

Several studies have shown that SMOTE is better than under-sampling and over-sampling techniques [22–26]. Several techniques inspired in the SMOTE algorithm have been proposed [26,31,32,22,33]. In [33], authors proposed the Borderline-SMOTE method, the algorithm over-samples the instances near the class boundaries. The algorithm tries to learn the borderline between classes and introduces artificial instances in this region. However, the borderline algorithm could introduce noise by adding artificial instances because this is a very sensitive region. The introduction of bad artificial instances could damage the performance of the classifier. In [26] the authors tested ten different implementations of under-sampling and over-sampling to balance the class distribution of the training data. In their conclusions, implementations based on SMOTE have better performance than implementations based on under-sampling. In [34] Akbani et al. proposed an algorithm based on SMOTE to make the distribution of a minority class more dense, which is called SMOTE with Different Cost (SDC). SDC pushes the biased decision boundary away from the minority-class. The method introduces a scheme to penalize classification errors, SDC assigns a high penalty for the majority class, while for the minority class SDC assigns a lower penalty. Zeng and Gao [8] proposed a kernelized version of SMOTE, in this implementation the algorithm generates new instances in the feature space instead of input space. All techniques mentioned before are based on the SMOTE technique to tackle the disadvantage of imbalanced data sets. However, the principal disadvantage of these methods is that it is possible to introduce noise in the classifier when new instances are generated, in this context it is necessary to develop algorithms that cope with this problem by carefully generating artificial instances and introducing just optimal new instances in data sets.

2.2. SVM classification on imbalanced data

SVMs are one of the most effective methods for the binary classification problem [23,34]. They achieve optimal classification in linear separable case. The generalization power of SVMs is one of their mainly remarkable properties; the key advantages of SVMs are the absence of local minimal, sparseness of their solution and their capacity to generalize by optimizing the margin [35]. Formally; the training of SVMs begins with a training set X_{tr} given by

$$X_{tr} = \{(x_i, y_i)\}_{i=1}^{n} \tag{1}$$

with $x_i \in R^d$ and $y_i \in \{-1, +1\}$. The classification function is determined by

$$y_i = sign\left(\sum_{j=1}^{n} \alpha_i y_j K\langle x_i \cdot x_j\rangle + b\right) \tag{2}$$

where α_i are the Lagrange multipliers, $K\langle x_i \cdot x_j\rangle$ is the kernel matrix, and b is the bias. The optimal separating hyperplane is computed by solving the following optimization problem:

$$\min \frac{1}{2} w_i^T w_i + C \sum_{t=1}^{l} \eta_i^2 \tag{3}$$

subject to

$$y_i(w_i^T K\langle x_i \cdot x_j\rangle + b_i) \geq 1 - \eta_i \tag{4}$$

where C is the margin parameter to weight the error penalties η_i. The margin is optimal in the sense of (3).

Formally, given a data set $\{(x_i, y_i)\}_{i=1}^{n}$ and a separating hyperplane $f(x) = w_i^T x + b = 0$, the shortest distance from the separating hyperplane to the closest positive example in the non separable case is

$$\gamma_+ = \min \gamma_i, \quad \forall\, \gamma_i \in class + 1 \tag{5}$$

the shortest distance from the separating hyperplane to the closest negative example is

$$\gamma_- = \min \gamma_i, \quad \forall\, \gamma_i \in class - 1 \tag{6}$$

where γ_i is given by

$$\frac{y_i(w_i^T K\langle x_i \cdot x_j\rangle + b_i)}{\|w\|} \tag{7}$$

The margin is

$$\gamma = \gamma_+ + \gamma_- \tag{8}$$

Fig. 1 shows different margins γ, which are obtained with SVMs with a different imbalance ratio a) balanced data set, b) imbalanced ratio of 1:10, and c) imbalanced ratio of 1:30. In Fig. 1, each plus sign in blue represents a data point with positive label and each plus sign in red represents a data point with negative label. The margin is defined by the brown and blue lines. The green line represents the midpoint of the margin γ. In Fig. 1 a), the margin of the balanced data set is well defined. However, when the imbalance grows, the margin is biased towards the majority class.

Methods based on the SMOTE algorithm introduce artificial instances in the minority class in order to reduce the bias. However, SMOTE only introduces artificial instances between positive instances (one positive instance and its k nearest neighbors). Clearly, the region with more information is between support vectors with different label. Introducing artificial instances in this region could help improve the performance of SVMs.

3. Particle swarm optimization

Hybrid algorithms have been used in many fields [36–40]. PSO is a stochastic optimization technique introduced by Kennedy and Eberhart, inspired and originally designed to mimic the flocking behavior of birds and fish schooling [41]. PSO is a population-based search method that exploits the concept of social sharing of information. Each individual (called *particle*) of a given population (called *swarm*) benefits from the previous experiences and refines its position in the search space. PSO starts with an initial population of particles whose positions are randomly generated. The number of particles in a swarm is represented by m. Each particle, denoted by $P_i(t)(i = 1,\ldots,m)$, is characterized by:

1. Position $P_i \in R^d$, which represents the $i - th$ candidate solution at iteration t.
2. Velocity $V_i(t) \in R^d$.
3. Best position of the particle $\mathbf{p}_b(t) \in R^d$, which represents the best solution of the particle $P_i(t)$ up to the current iteration.

During the process, the particles adjust their positions and velocities. Also, the population memorizes the best position among all particles in the swarm up to the current iteration. This position is called the *best known position*. It is represented by $\mathbf{p}_g(t) \in R^d$. The advantages of PSO are: simplicity, ease of implementation and computational efficiency.

The velocity of the particle P_i is changed as follows:

$$V_i(t+1) = wV_i(t) + c_1 \cdot r_1(t)(\mathbf{p_b}(t) - P_i(t)) + c_2 \cdot r_2(t)(\mathbf{p_g}(t) - P_i(t)) \tag{9}$$

where w is a parameter called the *inertia weight*, c_1 is the attraction of particle $P_i(t)$ towards $\mathbf{p_b}(t)$, c_2 is the attraction of particle $P_i(t)$ towards $\mathbf{p_g}(t)$, $r_1(t)$ and $r_2(t)$ are random variables obtained from a uniform distribution in the range $[0, 1]$.

The position of particle P_i is updated as follows:

$$P_i(t+1) = P_i(t) + V_i(t) \tag{10}$$

The inertia weight w allows PSO to manipulate the grade of exploration. The parameters c_1 and c_2 define the relative attraction of the best position of a particle and the best known position. These variables determine how the particle is influenced by the cognitive rate and the social rate, respectively.

4. SMOTE-PSO

Many methods, including SMOTE, produce examples that are linear combinations of current examples for a class. These techniques obtain new instances which are generated internally. Fig. 2 a) shows a data set with two classes, this data set is imbalanced because it contains a class with many data points called majority or negative class (black squares) and a class with few data points called minority or positive class (black circles). The Fig. 2 b) shows how the new instances are generated internally. In the Figure the red circles are artificially generated by linear combination between two instances in the minority class.

The SMOTE algorithm oversamples the minority class by taking each minority class instance and introducing synthetic examples along the line segments by joining any/all of the k minority class nearest neighbors, as shown in Fig. 2 b). In Fig. 2 b), the red circles represent the generated synthetic instances. It can be seen that the density is incremented. However, it does not ensure that the separating hyperplane would be moved.

SMOTE-PSO takes a different approach. It generates new examples

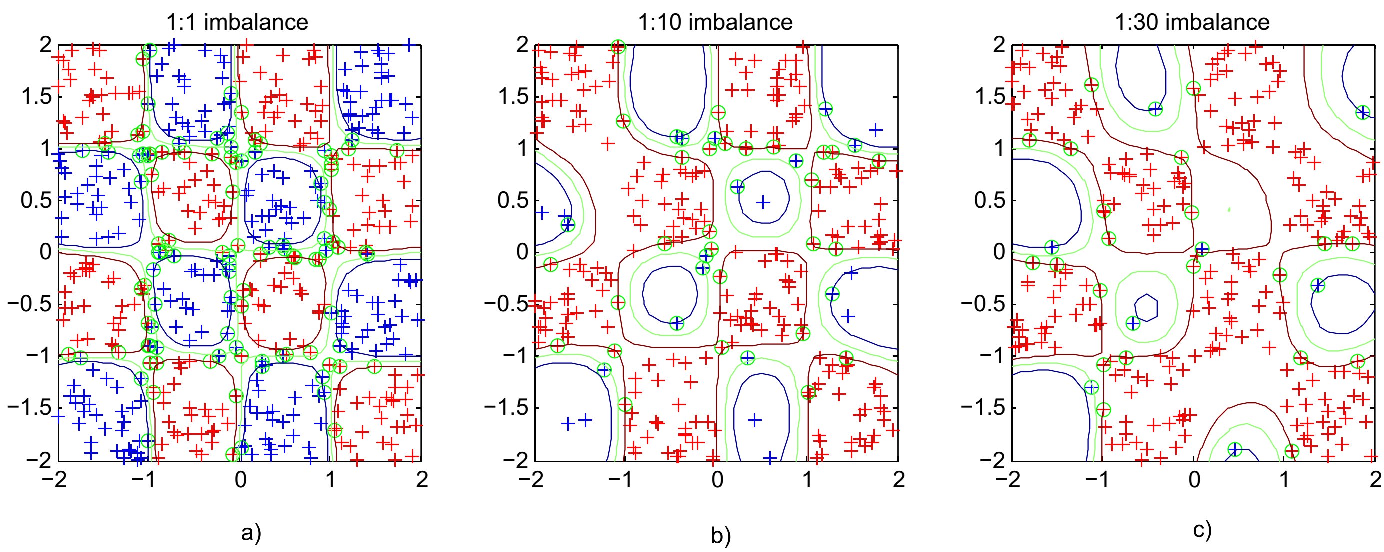

Fig. 1. Margin in data sets with different imbalance ratio. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

in the region where the density of the minority class decreases. Instead of considering only the examples of a class, examples with opposite class are used, which are located close to negative instances.

Fig. 3 a) shows the imbalanced data set with minority (black circles) and majority (black squares) classes, and Fig. 3 b) shows how the new instances are generated by a linear combination. In this case, the red circles (synthetic instances) are artificially generated by linear combination between two instances. The first instance is obtained in the minority class and the second instances computing the most closed instance in the majority class. Current algorithms can obtain one or several most closed instances to one instances in the minority class. The Fig. 3 b) shows an instance in the minority class and the three most closed instances in the majority class joined by lines.

The SMOTE-PSO algorithm oversamples the minority class by taking each minority class instance and introducing synthetic examples along the line segments by joining any/all of the k majority class nearest neighbors, as shown in Fig. 3 b). In this figure, the red circles represent the generated synthetic instances. Fig. 3 b) shows an example of this idea. In this figure, the new examples are generated ”out of the class” distribution.

In order to reduce the effect of imbalance in data sets, the SMOTE-PSO algorithm generates new examples that belong to the minority class and the PSO algorithm is used to evolve the instances that improve the performance of the SVM. If the new points are carefully located, the separating hyperplane can be shifted and the margin on the side of the minority class is increased. The idea is to move the decision boundary towards the majority class. The SMOTE-PSO algorithm is different from other methods such as SMOTE, where some internal data points are randomly generated and added to the training set.

4.1. Input data set

Obtaining a discriminative data set is an important issue in machine learning. This is because the identification of how many instances are sufficient to gain knowledge, make a good decision and validate results is really a very difficult task. In order to obtain an initial data set with discriminative abilities, we use k-fold cross-validation with k=5. To separate the input data set, each one of the subsets is separated maintaining almost equal proportion in class distribution over the data.

For instance, using k-fold cross-validation with k=5, if there are 2 class values (X^- and X^+) in a classification problem P with 1000 examples in total, and the number of examples of majority and minority classes (X^-, X^+) are respectively 800 and 200. Then, each subset will contain 200 instances with 160 negative instances and 40 positive instances.

4.2. Pre-processing

In this research, a new algorithm based on PSO is proposed in order to introduce artificial instances with high discriminative features.

Fig. 4 and Algorithm 1 describe the general process of the SMOTE-PSO method. The SMOTE-PSO algorithm starts normalizing the input data set. Each instance x_i, $x_i=(x_{i1},\ldots,x_{ir})^T \in X_T - (x_i,\ldots,x_L)$ (where X_T

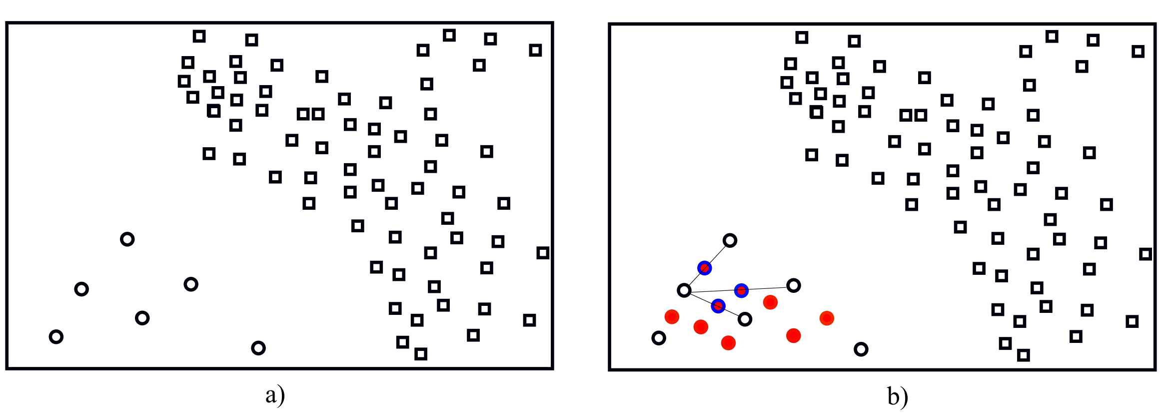

Fig. 2. Samples generated internally. a) Imbalanced data set b) Data set with synthetic instances (red circles) generated internally. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

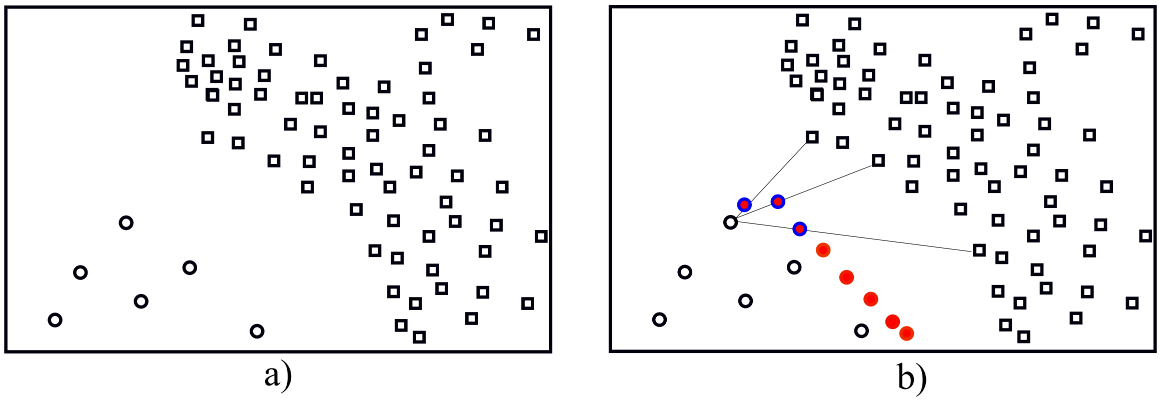

Fig. 3. Samples generated externally. a) Imbalanced data set b) Data set with synthetic instances (red circles) generated externally. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

includes the total input data set) in the input data is normalized as

$$x_i = \left(\frac{x_{i1} - \min x_{k1}}{\max{-x_{k1}} - \min x_{k1}}, \ldots, \frac{x_{ir} - \min x_{kr}}{\max x_{kr} - \min x_{kr}} \right) \tag{11}$$

where $1 \le k \le L$ and r is the dimensionality of x_i.

4.3. SVM classification

In this part, an SVM is trained using X_{tr}^+ and X_{tr}^-, the idea in this step is to identify the support vectors (SVs). In SVMs the solution is given by a small subset of instances called support vectors (SVs). These SVs are the most important and successful instances in the entire data set. The success of each SV is given by its position in the feature space which defines the hyperplane and gives the solution. The hyperplane obtained in this step will be obtained by Eq. (7) and the decision function is given by Eq. (2), which is skewed due to imbalance in the input data. However, the obtained SVs will serve us to generate new instances and the initial population for the PSO. These data points are the most representative, but its effectiveness in some cases is difficult to ensure. In our case, the implemented PSO algorithm guarantees the effectiveness of the created data points.

4.4. Generation new synthetic instances

The key idea of this model is to intelligently introduce artificial instances in the region of the minority class to reduce the skew behavior of the separation margin. The artificial instances not only correct the margin, but also modify the region of the minority class. Moreover, the examples generated by the SMOTE-PSO method derive from the most critical region for SVMs, called the margin. In order to do this, SVs are used to generate artificial instances and correct the skewed hyperplane. Firstly, the SVs in the minority class are moved to the majority class. For each SV in the minority class sv^+, the algorithm finds the k nearest neighbors in the sv^- and calculates the distance between them for each dimension. The distance nu_k is given by

$$\nu_k = x^{i}_{sv^+} - x^{j}_{sv^-}, \quad k = 1, \ldots, r \tag{12}$$

where $x^{i}_{sv^+}$ is the ith *SV* of X_{tr}^+, and $x^{j}_{sv^-}$ represents the jth sv^- nearest neighbors of $x^{i}_{sv^+}$, and r defines the dimensionality of the instance. Initial vector $v_i = 0$, $k = 1, \ldots, d$, and the algorithm picks one or more random entries out of an array. In the experiments only one is selected. The artificial instance is obtained by

$$x_g = x^{i}_{sv^+} + \epsilon \cdot \nu_k \tag{13}$$

which modifies only the ith dimension of $x^{i}_{sv^+}$.

Where the step size is ϵ, in the experiments ϵ is selected between 0.001 and 0.1. According to the geometric properties of SVMs [42], the movement of the SVs in the minority class to the majority class can improve the classification accuracy, sensitivity and sensibility. In the SMOTE algorithm, the minority class is oversampled until the minority class size is equal to the majority class. In the SMOTE-PSO method, two artificial instances for each SV are generated for the positive class until the final data set has the same number of negative and positive instances.

This displacement of SV^+ moves the decision boundary towards the majority class improving the classification accuracy, sensitivity and sensibility. The initial population is conformed by $x_{svi}^- \cup x_{svj}^+ \cup x_{svg}$. Fig. 5 shows how the new instances are generated from SVs.

The choice of ϵ is very important in the generation of new instances, an $\epsilon \approx \gamma_+$ is a bad choice because we can introduce noise or outliers in the data set. In our experiments an $\epsilon < 0.1$ gives us good results.

Fig. 5 shows how the synthetic instances are generated. In the Figure, circles represent the minority class and squares represent the majority class. The instances x_1,x_2 and x_i are the support vectors in the minority class x_{svi}^+. The instances x_{j1}, x_{j2} and x_{j3} are the three support vectors in the majority class x_{svj}^- most closed to x_i joined by r_1,r_2 and r_3 respectively. In order to avoid introduce noise, the synthetic instances (data points in blue) are generated very closed to the instances in minority class.

The algorithm to generate instances is shown in Algorithm 2.

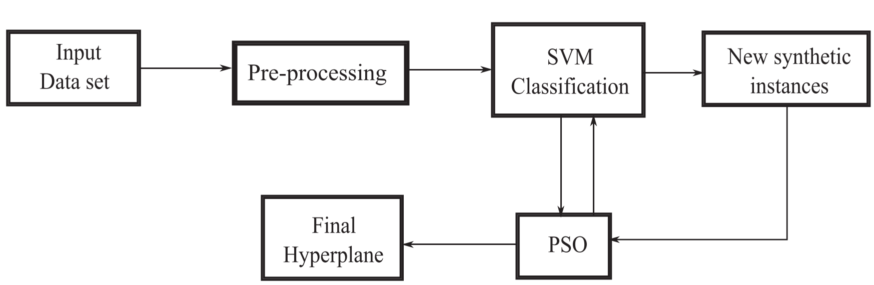

Fig. 4. SMOTE-PSO algorithm.

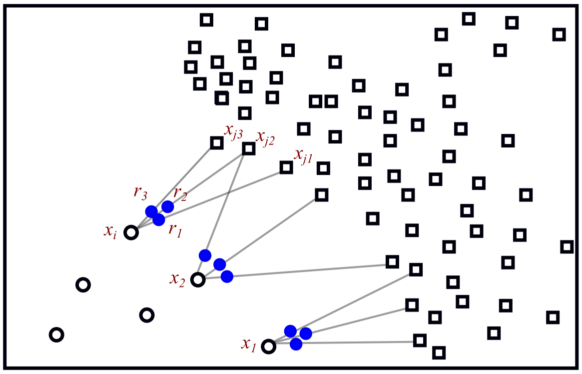

Fig. 5. Data points generated.

4.5. PSO for imbalanced data sets

In this subsection we describe the proposed Imbalanced SMOTE-PSO system. The proposed Imbalanced SMOTE-PSO system can be divided into two parts, in the first part an SVM is trained in order to obtain the most important instances from the skewed data set, the second part describes the way PSO optimizes the generated artificial instances.

4.5.1. Coding of artificial instances

The initial population is obtained by generating artificial instances, each instance is defined by $x_{g_i} = (x_1, x_2, \ldots, x_r)$, where r is the dimensionality of each instance. Each PSO particle is defined by $p_i = \left(x^i_{g_1}, x^i_{g_2}, \ldots, x^i_{g_q}\right)$, where q is the number of generated artificial instances. In each particle there are q artificial instances with dimensionality r. Fig. 6 shows how the instances in the swarm are coding and decoding.

4.5.2. Particle swarm

Particle swarm is denoted by $P = [p_1, p_2, \ldots p_m]^T$. Each particle is a vector with $(q \times r)-$ dimensionality, where q is the number of created instances, r is the dimensionality of each instance, and m is the size of the initial population. The problem is determining the artificial instances that improve the performance. The $(m \times qr)-$ dimensional search space Γ is defined by

$$\Gamma = \prod_{i=1}^{m\times q\times r} [\Gamma_{i,\min}, \Gamma_{i,\max}] \tag{14}$$

The search space of each individual $x = [x_1, x_2, \ldots x_r]^T$ is defined by the minimal distance between SVs with different class, i.e.

$$x_{\min,i} = sv_i^{+}\cdot 1 \tag{15}$$

$$x_{\max,i} = \min_{1\le k\le n} D(sv_i^{+}, sv_i^{-})\cdot \epsilon \tag{16}$$

4.5.3. Decoding

When a PSO algorithm is used to solve the optimization problem, a swarm of the candidate particles $\{P_i^l\}_{i=1}^m$ is moved in the search space Γ in order to find a solution $\wedge x$, where m is the size of the swarm, and $l \in \{0, 1, \ldots, L\}$ denotes the lth movement of the swarm.

Each particle $p(i)$ has a $(q \times r)-$ dimensional velocity $\mathbf{v} = [v_1, v_2, \ldots v_{qr}]^T$ to direct its search, and $\mathbf{v} \in \mathbf{V}$ with the velocity space defined by

$$V = \prod_{i=1}^{q\times r} [V_{i,\min}, V_{i,\max}] \tag{17}$$

where $V_{i,\max} = \frac{1}{2}(\Gamma_{i,\max} - \Gamma_{i,\min})$. To start PSO, the candidate particles $\{X_i^0\}_{i=1}^m$ are randomly initialized within Γ, and the velocity of each candidate particle is initialized to zero, $\{v_i^0 = 0\}_{i=1}^m$. The cognitive information $\mathbf{pb}_i^l$ and the social information $\mathbf{gb}^l$ record the best position visited by the particle i and the best position visited by the entire swarm, respectively, during l movements. The cognitive information $\mathbf{pb}_i^l$ and the social information $\mathbf{gb}^l$ are used to update the velocities according to eq. (9). Each iteration the synthetic instances are evolved improving the classification accuracy.

Algorithm 1. General process of the SMOTE-PSO algorithm.

Input: Skewed dataset
Output: Optimized final hyperplane (H_f)
1: Divide the input data set in $X_{tr} = \{X_{tr}^+, X_{tr}^-$ where $X_{tr}^+ = \{x_i \in X: y=+1\}, i = 1,\ldots,m$ and $X_{tr}^- = \{x_j \in X: y = -1\}, j = 1,\ldots,n$
2: Train the SVM with the training data set, train SVM X_{tr}^+, X_{tr}^-
3: Obtain support vectors x_{svi}^- and x_{svi}^+ from hyperplane in i^{th} iteration (H_i)
4: Generate new instances using equations (12) and (13)
5: Obtain H_f from X_{te}^+, X_{te}^- using the PSO algorithm described in Algorithm 2

Meta-optimizers can be used to tune the PSO parameters. In order to maximize the search speed on each specific problem, the optimization parameters are used. [43] present a interesting work to obtain optimum parameters for PSO on several optimization scenarios. The parameters were tuned on several benchmark problems with several dimensionalities. The choice of parameters w, c_1, c_2 and the search space are essential to the performance of PSO and therefore have been the focus of prior research. Variables c_1 and c_2 represent the cognitive and social learning parameters that pull each particle towards the global best positions and r_1, r_2 are uniform random values in the range [0,1]. These variables determine together the space searching ability and control the behavior of the algorithm. Apparently w, c_1, c_2 are the parameters that determine the accuracy and convergence characteristics of the algorithm. Low values of these variables provoke that particles roam far from the target regions. On the other hand, high values result in an abrupt movement towards the target region [43,44].

The performance of PSO is also highly influenced by the acceleration constants and inertia weight. A grid search method was implemented for different c_1 and c_2 in the range [1 4] and w in the range [0.7 1.6]. The parameter w was decreased by 5% in each iteration while c_1 and c_2 was constant. According to the experiments described here, the PSO algorithm obtains the highest performance when w=1 and $c_1 = c_2 = 2$. Kennedy has studied the effect of the random variables c_1 and c_2 on the particle trajectories and asserted that if $c_1 + c_2 > 4$, velocities and positions explode towards infinity. In the proposed research based on experiments, the acceleration constants have been set to 2.0 and w=1 according to empirical experiences [45].

The input space is expressed by the generated artificial instances. Each particle p_i contains q new artificial instances with dimensionality r. The general process of the SMOTE-PSO algorithm is described in Algorithm 1 and 2.

Algorithm 2. PSO algorithm.

Input: Support vectors x_{svi}^- and x_{svi}^+, number of iterations ρ
Output: Global best particle
1: Generate an initial swarm of size $m \times q \times r$ from x_{svi}^- and x_{svi}^+ with Eqs. (12) and (13).
2: Set initial velocity vectors $V_i(i = 1,\ldots,q \times r)$ associated with the particles.
3: For each position p_i of the particle $P_i(i = 1,\ldots,m)$ which contains artificial instances created from SVs, train an SVM

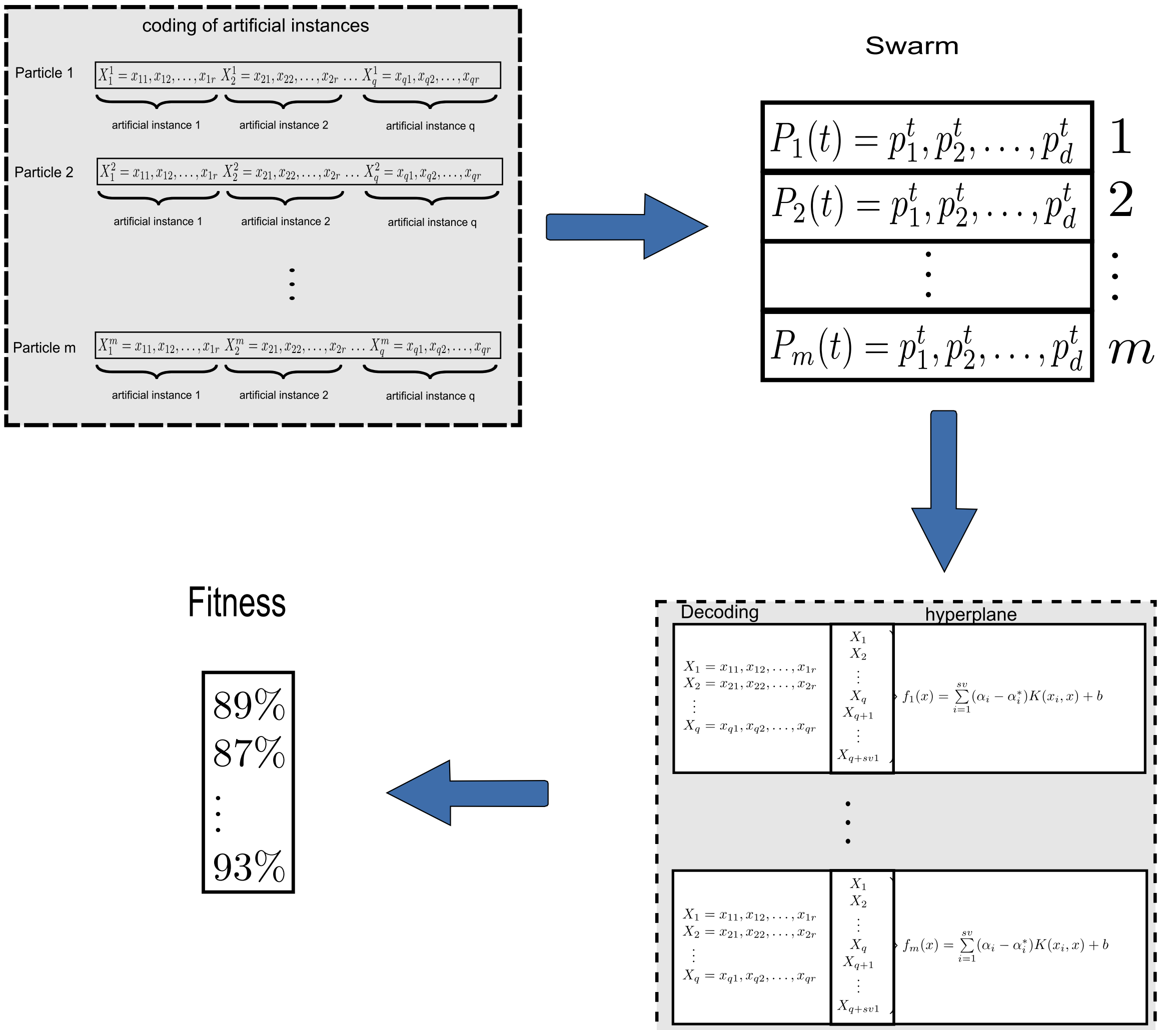

Fig. 6. Optimizing instances by PSO algorithm.

classifier and compute its fitness function φ.

4: Set the best position of each particle with its initial position, i.e., $\mathbf{p}_{bi} = p_i$, $(i = 1, \ldots, m)$.

5: Obtain the best global particle $\mathbf{p}_g$ in the swarm.

6 Update the speed of each particle using (9).

7 Update the position of each particle using (10).

8 For each candidate particle p_i, train an SVM classifier and compute its fitness function φ.

9 Update the best position $\mathbf{p}_\mathbf{b}$ of each particle if its current position has a smaller fitness function.

10 Return to 5 if the pre-specified stopping condition is not yet satisfied **return** Obtain the best global particle.

The final obtained hyperplane gives us a decision function Eq. (2). From the final decision function, we can obtain the performance by testing data set X_{te}^- and X_{te}^+. In the SMOTE-PSO algorithm, the population size and the number of iterations or stop criterion are used like mechanisms to avoid over-learning in the training data.

4.5.4. *Fitness function*

The fitness function value φ associated with the *ith* particle P_i is essentially the objective function of the problem. Fitness function provides a way to find the best solutions, and also controls the update process. The choice of the fitness function is important because helps to PSO algorithm to evaluate the goodness of each candidate solution P_i. Each particle in the proposed method is created by modifying the SV found in the first stage of the SVM. The fitness function value of each particle can be evaluated using the SVM and obtaining the performance of each particle in the data set Te_f. For classification, we can consider factors such as prediction accuracy, error rate, in this research, the area under the curve (AUC) and g-mean are used as fitness functions.

5. Experimental results

In this section, our aim is to show the improvement achieved in SVMs by the combination of the generated data points and a PSO algorithm to evolve the synthetic instances. The usefulness of the SMOTE-PSO technique is checked by means of comparisons using classical implementations to imbalanced data sets. In order to select the best hyper-parameters and validate the obtained results, the model selection and the metrics used are described in this section.

5.1. *Model selection*

Training an SVM involves the choice of some parameters. Such parameters have an important effect on the performance of the classifier. In all the experiments we use the radial basis function (RBF) as the kernel, this function is defined in (18).

$$K(x_i - x_j) = e^{(\gamma\|x_i - x_j\|)}, \gamma > 0 \qquad (18)$$

Cross-validation was used to find parameters in (18) and also for computing the regularization parameter of SVMs. We use model

Table 1
Imbalanced data sets.

Data set	*mc* (+)	*Mc*(−)	Features (f)	Imbalance ratio
Shuttle	1706	2175	9	1:01.275
Liver_disorders	145	200	6	1:01.379
Glass1	76	138	9	1:01.816
Pima	268	500	8	1:01.866
Glass0	70	144	9	1:02.057
German	300	700	20	1:02.333
Haberman	81	225	3	1:02.777
Vehicle2	218	628	18	1:02.881
Vehicle3	212	634	18	1:02.991
Ecoli1	77	259	7	1:03.364
New-thyroid1	35	180	5	1:05.143
New-thyroid2	35	180	5	1:05.143
Ecoli3	35	301	7	1:08.600
Ecoli-0147vs2356	145	1535	7	1:10.586
Glass2	17	197	9	1:11.588
Ecoli-0147vs56	125	1535	6	1:12.280
Abalone	42	689	7	1:16.405
Letter	789	19211	16	1:24.349
Yeast4	51	1433	8	1:28.098
Yeast6	35	1449	8	1:41.400
Page-blocks	115	5358	10	1:46.591

selection to get the optimal parameters. The hyper-parameter space is explored with the kernel parameter γ=[10^{-2}, 10^{-1}, 10^{0}, 10^{1}] and the regularization parameter C = [10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}].

In the experiments all data sets were normalized and the 5-fold cross-validation method was applied for the measurements. Several authors recommend using $k > 10$ for cross-validation. However, in many imbalanced data sets, the use of k=10 is prohibitive because the minority group could remain without instances or with very few instances.

5.2. Metrics for testing classifiers on skewed data sets

Most times, accuracy is the measurement used to evaluate and compare a classifier method against others. On skewed data sets, using accuracy as a metric to evaluate a classifier can lead to wrong conclusions, because the minority class has a small impact on accuracy compared with the majority class. Consider, for example a data set with an imbalance ratio of 99 to 1. A classifier that achieves 99% of accuracy is considered good for balanced data sets. However, on skewed data sets, this performance measure is not useful. In order to evaluate and assess the improvement of a classifier on skewed data sets, it is necessary to use a different performance measure. Medical and machine learning communities use more and more the sensitivity and specificity to evaluate the performance.

Sensitivity is computed with (19) and defines the proportion of positive examples that are correctly identified, whereas the specificity is the proportion of negative examples that are correctly identified.

$$S_n^{true} = \frac{T_P}{T_P + F_N} \tag{19}$$

and specificity is computed with (20).

$$S_n^{false} = \frac{T_N}{T_N + F_P} \tag{20}$$

where T_P is the number of objects (true class +1) that have been predicted as class +1. T_N is the number of objects (true class −1) that have been predicted as −1. F_P is the number of objects (true class −1) that have been predicted as class +1. F_N is the number of objects (true class +1) that have been predicted as class −1. G-mean is used in this research, which is a combination of sensitivity and specificity

$$G - mean = \sqrt{S_n^{false^{true}_{n}}} \tag{21}$$

In addition to the numeric performance metrics mentioned above, the area under the ROC curve (AUC) is also used in this paper. The Receiver Operating Characteristic (ROC) analysis is a widely used method for analyzing the performance of binary classifiers. The area under the ROC curve represents how separable two objects are. AUC and G-mean metrics are commonly utilized by many researchers for evaluating classifiers on imbalanced data sets [34,23,17,47]. However, AUC describes the ranking ability or the quality of the classification and G-mean describes the performance of the classifier on both classes (minority and majority).

A ROC curve can be generated using the labels of the input data set and the classifier output. A detailed description on how to plot a ROC curve also can be found in [46]. The most important advantage of ROC analysis is that it is not necessary to specify the misclassification costs. The visual and numeric metrics associated with this method allow for great flexibility in performance analysis. In the experiments, G-mean and AUC-ROC measures are used as fitness functions.

5.3. Data sets and results

We conducted several experiments with different classical classification algorithms for imbalanced data sets. In the experiments carried out, we use undersampling oversampling and SMOTE algorithms and the results obtained with these algorithms were compared with the proposed method. In all algorithms grid search technique was used to optimize parameters.

In this study, we have selected a wide benchmark of 21 data sets from the KEEL data set repository. Keel Data sets are imbalanced ones (Public available at http://sci2s.ugr.es/keel/datasets.php). Table 1 shows the data sets used in the experiments. In order to measure the performance of the SMOTE-PSO method in different scenarios, the chosen data sets have an imbalance ratio from 1 to 1.248 up to 1–41.4. Table 1 summarizes the properties of the selected data sets. This table shows the number of examples in the minority class (mc), the number of examples in the majority class (Mc), the number of features (f) and the imbalance ratio for each data set. In the case of missing values we have removed those instances from the data set.

The approach was implemented in Matlab. The results for all the algorithms used in this study is reported in Table 2. The first column indicates the data set, and the other columns report the corresponding *AUC* and G-mean measure, σ represents standard deviations of the SMOTE-PSO method.

In Table 2, the full test results obtained can be observed. The standard deviations for the G-mean measure is included. The best results for each data set are highlighted in bold font. As shown in Table 2, the SMOTE-PSO method achieves the best results compared with undersampling, oversampling and SMOTE techniques on almost all data sets. The average performance of AUC and g-mean on data sets with small imbalances (<10: 1) is 0.8584 and 0.8462, respectively, and the average performance on data sets with large imbalances (>10: 1) is 0.8723 and 0.8611, respectively. The experimental results show that the performance of the SMOTE-PSO method is better than classical implementations when imbalance ratio is large.

We summarize the strongest points of our SMOTE-PSO system as follows:

In almost all data sets, the SMOTE-PSO method achieves better measure performance than classical competent methods. These results allow us to highlight the goodness of the SMOTE-PSO model to evolve synthetic instances. The improvement provided by the SMOTE-PSO methodology proves that a right management of the PSO algorithm associated with SVMs has a positive synergy with the tuning of artificial instances, leading to an improvement in the global behavior of the system.

On the other hand, the standard deviations in results obtained confirm that the SMOTE-PSO method can effectively deal with imbalanced data and improve prediction performance.

Table 2
Detailed results for the SMOTE-PSO algorithm.

	Under-sampling		Over-sampling		SMOTE		SMOTE-PSO		
Data set	AUC	G	AUC	G	AUC	G	AUC	G	σ
Shuttle	0.950	0.871	0.921	0.853	0.950	0.877	**0.961**	**0.891**	0.082
Liver_disorders	0.786	0.737	0.754	0.691	0.837	0.792	**0.871**	**0.856**	0.005
Glass1	0.765	0.624	0.741	0.673	0.746	0.636	**0.802**	**0.779**	0.047
Pima	0.696	0.725	0.647	0.718	0.714	0.735	**0.742**	**0.785**	0.042
Glass0	0.805	0.761	0.801	0.768	0.765	0.725	**0.839**	**0.817**	0.023
German	0.753	0.728	0.735	0.641	0.785	0.710	**0.806**	**0.74**	0.004
Haberman	0.502	0.537	0.520	0.600	0.609	0.634	**0.622**	**0.683**	0.089
Vehicle2	0.944	0.939	0.945	0.898	0.953	0.945	**0.993**	**0.971**	0.054
Vehicle3	0.593	0.675	0.635	0.678	0.658	0.706	**0.734**	**0.715**	0.002
Ecoli1	0.852	0.417	0.806	0.877	0.886	0.877	**0.944**	**0.936**	0.021
New-thyroid1	0.989	0.981	0.983	0.964	0.977	0.959	**0.995**	**0.991**	0.014
New-thyroid2	0.978	0.963	0.917	0.973	0.972	0.969	**0.986**	**0.977**	0.009
Ecoli3	0.809	0.787	0.798	0.780	0.741	0.817	**0.869**	**0.836**	0.071
Ecoli-0147vs2356	0.673	0.850	0.744	0.761	0.837	0.820	**0.853**	**0.870**	0.182
Glass2	0.607	0.639	0.624	0.271	0.674	0.725	**0.738**	**0.742**	0.020
Ecoli-0147vs56	0.612	0.825	0.592	0.850	0.742	0.787	**0.877**	**0.901**	0.015
Abalone	0.835	0.776	0.821	0.781	0.845	0.783	**0.872**	**0.814**	0.001
Letter	0.996	0.952	0.954	0.842	**0.998**	**0.993**	0.997	0.954	0.017
Yeast4	0.793	0.781	0.786	0.729	0.791	0.761	**0.847**	**0.824**	0.062
Yeast6	0.845	0.817	0.841	0.816	0.837	0.812	**0.848**	**0.826**	0.085
Page-blocks	0.867	0.927	0.901	0.931	0.913	0.917	**0.927**	**0.967**	0.104
Average	0.792	0.776	0.784	0.766	0.820	0.808	0.863	0.851	

5.4. Discussion

Some algorithms like SMOTE generate artificial instances to improve the performance of SVMs. However, finding the best instances that maximize the classification hyperplane is not an easy issue. A small change in a data feature can improve or affect the SVM performance. Consequently, the search space of each problem is often huge, complex or poorly understood. The finding of data points that improve the SVM performance in imbalanced data sets cannot be realized by classical methods. PSO has the ability to exploit good regions and searching by exploring new areas. PSO can be applied to the search problem of finding artificial instances with discriminative abilities. The use of the PSO algorithm allows to obtain artificial instances with an excellent ability to improve the SVM performance. To accomplish this, we take into account two important issues: the representation of the solution (synthetic examples of each particle) and the definition of the fitness function (measure performance).

Due to the nature of SVMs, the decision surface relies on the positive/negative support vectors. In this way, the creation of new data points between positive and negative support vectors can be unfavorable, because in some extreme cases, a single positive misclassified example could introduce a significant drop in the performance of the classifier. In order to face this disadvantage, the SMOTE-PSO method evolves the best artificial instances improving always the performance of the initial instances.

The principal disadvantage of the SMOTE-PSO method is its algorithmic complexity. The complexity of SMOTE-PSO depends mainly on the cost function of PSO, and the cost function of the proposed method is the evaluation of the SVM which has a complexity on the order of n^3 for each iteration (where n is the number of examples in the training set) [48].

The SMOTE-PSO method works well in small data sets, but on large data sets its time complexity is prohibitive.

6. Conclusions

Current classification methods produce good results when they are applied to data sets that are balanced, however for the specific case of skewed data sets most classifiers cannot obtain acceptable results because decision boundaries are computed regardless minority and majority classes.

The generation of artificial instances has been a new successful technique to tackle the imbalance in data sets. However, these popular techniques are based on generating internal examples. The generation of external instances is very difficult for two reasons; i) It could introduce noise in the data set if the instances are not generated carefully, ii) There is not a method to generate good data.

The SMOTE-PSO method generates new instances in the region where the density of the minority class decreases. Instead of considering only the positive examples of a class, SMOTE-PSO uses the instances with different label. If the artificial instances are carefully located, the separating hyperplane can be shifted and the margin on the side of the minority class is increased.

The principal advantage of SMOTE-PSO is the performance improvement on imbalanced data sets by adding artificial examples. However, the first disadvantage is the computational cost. SMOTE-PSO can be used only on small data sets. The computational complexity of SMOTE-PSO on medium and large data sets is prohibitive. In comparison with under-sampling, over-sampling and SMOTE, the SMOTE-PSO algorithm is computationally very expensive.

In this paper a novel method that enhances the performance of SVM for skewed data sets was presented. The method reduces the effect of the imbalance ratio by exciting SVs and moving the separating hyperplane towards the majority class. The method is different from other state-of-the-art methods for two reasons: the new instances are added close to the optimal separating hyperplane, and they are evolved to improve the performance of the classifier. The average performance of AUC and G-mean on data sets with small imbalances (<10: 1) is 0.8584 and 0.8462, respectively, and the average performance on data sets with large imbalances (>10: 1) is 0.8723 and 0.8611, respectively. According to the experiments, SMOTE-PSO produces the most noticeable results in average when the imbalance ratio is bigger than 10:1.

Acknowledgements

This research was supported in part by the UAEM Grant No. 3771/2014/CID.

References

[1] Z. Liu, R. Wang, M. Tao, X. Cai, A class-oriented feature selection approach for multi-class imbalanced network traffic datasets based on local and global metrics fusion, Neurocomputing 168 (30) (2015) 365–381.
[2] D. Wu, Z. Wang, Y. Chen, H. Zhao, Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset, Neurocomputing 190 (2016) 35–49.
[3] J. Sartakhti, M. Zangooei, K. Mozafari, Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA), Comput. Methods Prog. Biomed. 108 (2) (2012) 570–579.
[4] F. Melgani, Y. Bazi, Classification of Electrocardiogram Signals With Support Vector Machines and Particle Swarm Optimization, IEEE Trans. Inf. Technol. Biomed. 12 (5) (2008) 667–677.
[5] F. Kuang, W. Xu, S. Zhang, A novel hybrid KPCA and SVM with GA model for intrusion detection, Appl. Soft Comput. 18 (2014) 178–184.
[6] W.S. Chen, P.C. Yuen, J. Huang, D.Q. Dai, Kernel machine-based one-parameter regularized fisher discriminant method for face recognition, IEEE Trans. Syst. Man Cyber. 35 (4) (2005) 659–669.
[7] S. Koknar, L. Jan L, Improving SVM Classification on Imbalanced Data Sets in Distance Spaces, IEEE International Conference on Data Mining, 2009, pp. 259–267.
[8] Z.Q. Zeng, J.Gao, Improving svm classification with imbalance data set, in: Proceedings of the 16th International Conference on Neural Information Processing: Part I, (ICONIP–09). Berlin, Heidelberg: Springer-Verlag, 2009, pp. 389–398.
[9] Ch. Yang, J. Yang, J. Wang, Margin calibration in SVM class-imbalanced learning, Neurocomputing 73 (1–3) (2009) 397–411.
[10] M. Antonelli, P. Ducange, F. Marcelloni, An experimental study on evolutionary fuzzy classifiers designed for managing imbalanced datasets, Neurocomputing 146 (2014) 125–136.
[11] C.H. Jian, J. Gao, Y. Ao, A new sampling method for classifying imbalanced data based on support vector machine ensemble, Neurocomputing 193 (2016) 115–122.
[12] S. Cateni, V. Colla, M. Vannucci, A method for resampling imbalanced datasets in binary classification tasks for real-world problems, Neurocomputing 135 (2014) 32–41.
[13] J. Fu, S. Lee, Certainty-based active learning for sampling imbalanced datasets, Neurocomputing 119 (2013) 350–358.
[14] N.V. Chawla, K.W. Bowyer, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[15] S. Zou, Y. Huang, Y. Wang, J. Wang, C.Zhou, Svm learning from imbalanced data by ga sampling for protein domain prediction, in: Proceedings of the 2008, the 9th International Conference for Young Computer Scientists, (ICYCS 08). Washington, DC, USA: IEEE Computer Society, 2008, pp. 982–987.
[16] H. Zhang, M. Li, RWO-Sampling: a random walk over-sampling approach to imbalanced data classification, Inf. Fusion 20 (2014) 99–116.
[17] B.X. Wang, N. Japkowicz, Boosting support vector machines for imbalanced data sets, in: Proceedings of the 17th international conference on Foundations of intelligent systems (ISMIS'08). Springer-Verlag, Berlin, Heidelberg, 2008, pp. 38–47.
[18] M.A.H. Farquad, I. Bose, Preprocessing unbalanced data using support vector machine, Decis. Support Syst. 53 (1) (2012) 226–233.
[19] G. Wu, E. Chang, KBA: kernel boundary alignment considering imbalanced data distribution, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 786–795.
[20] S. Tan, Neighbor-weighted k-nearest neighbor for unbalanced text corpus, Expert Syst. Appl. 28 (4) (2005) 667–671.
[21] G. Wu, E. Chang, Class-boundary alignment for imbalanced data set learning, presented at the International Conference Data Mining, Workshop Learning Imbalanced Data Sets II, 2003, pp. 49–56.
[22] N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, Smoteboost: improving prediction of the minority class in boosting, Proc. Princ. Knowl. Discov. Databases (2003) 107–119.
[23] R. Batuwita, V. Palade, Class Imbalance Learning Methods for Support Vector Machines, In Imbalanced Learning: Foundations, Algorithms and Applications, Haibo He and Yunqian Ma Ma (Eds.), Wiley, 2013.
[24] H. Guo, H.L. Viktor, Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach SIGKDD Explor. Newsl., ACM 6, 2004, pp. 30–39.
[25] H.M. Nguyen, E.W. Cooper, K. Kamei, Borderline over-sampling for imbalanced data classification, Int. J. Knowl. Eng. Soft Data Paradig. 3 (1) (2011) 4–21.
[26] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 20–29.
[27] J. Choi, A Selective Sampling Method for Imbalanced Data Learning on Support Vector Machines, PhD Dissertation, 2010, Iowa State University.
[28] Q. Cai, H. He, H. Man, Imbalanced evolving self-organizing learning, Neurocomputing 133 (2014) 258–270.
[29] S. García, J. Derrac, I. Triguero, C.J. Carmona, F. Herrera, Evolutionary-based selection of generalized instances for imbalanced classification, Knowl. Based Syst. 25 (1) (2012) 3–12.
[30] A. Fernandez, M. Jesus, F. Herrera, Improving the performance of fuzzy rule based classification systems for highly imbalanced datasets using an evolutionary adaptive inference system, in Bio-Inspired Systems: Computational and Ambient Intelligence. Berlin, Germany. Springer-Verlag, 2009, pp. 294–301.
[31] S. Hu, Y. Liang, L. Ma, Y. He, MSMOTE: improving classification performance when training data is imbalanced, in: Proceedings of the 2nd International Workshop Computer Science Engineering 2, 2009, pp. 13–17.
[32] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, IEEE Joint Conference on Neural Networks, 2008, pp. 1322–1328.
[33] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, Lect. Notes Comput. Sci. 3644 (2005) 878–887.
[34] R. Akbani, S. Kwek, N. Japkowicz, Applying Support Vector Machines to Imbalanced Datasets, in: Proceedings ECML, 2004, pp. 39–50.
[35] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition Data Min. Knowl. Discov., Kluwer Academic Publishers 2, 1998, pp. 121–167.
[36] J.X. Du, D.S. Huang, X.F. Wang, X. Gu, Shape recognition based on neural networks trained by differential evolution algorithm, Neurocomputing 70 (4–6) (2007) 896–903.
[37] Ji-Xiang Du, D.S. Huang, Guo-Jun Zhang, Zeng-Fu Wang, A novel full structure optimization algorithm for radial basis probabilistic neural networks, Neurocomputing 70 (1–3) (2006) 592–596.
[38] D.S. Huang, Ji-Xiang. Du, A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, IEEE Trans. Neural Netw. 19 (12) (2008) 2099–2115.
[39] P. Prosser, Hybrid algorithms for the constraint satisfaction problem, Comput. Intell. 9 (3) (1993) 268–299.
[40] D.S. Huang, W. Jiang, A general CPL-AdS methodology for fixing dynamic parameters in dual environments, IEEE Trans. Syst. Man Cybern. Part B 42 (5) (2012) 1489–1500.
[41] J. Kennedy, R. Eberhart, Particle swarm optimization. in: Proceedings of the International Conference on Neural Networks.IV, 199, 5, 1942, –1948.
[42] K.P. Bennett, E.J. Bredensteiner, Duality and Geometry in SVM Classifiers, in: Proceedings of the 17th International Conference on Machine Learning, San Francisco, CA, 2000, pp. 57–64.
[43] M. Erik, H. Pedersen, M.E.H. Pedersen, Good Parameters for Particle Swarm Optimization, Technical Report HL1001, Hvass Laboratories, vol. HL1001, 2010, pp. 1–12.
[44] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, Inf. Sci. 177 (22) (2007) 5033–5049.
[45] J. kennedy, The behavior of particle swarm, in: V.W. Nsaravan, D. Waagen, (eds), in: Proceedings of the 7th international conference on evolutionary programming, 1998, pp. 581–589.
[46] T. Fawcett, An introduction to roc analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.
[47] J. Cervantes, X. Li, W. Yu, Splice site detection in dna sequences using a fast classification algorithm, in: Proceedings of the 2009 IEEE international conference on Systems, Man and Cybernetics, (SMC'09). Piscataway, NJ, USA: IEEE Press, 2009, pp. 2683–2688.
[48] J. Cervantes, F. Garca-Lamont, A. Lpez Chau, L. Rodrguez-Mazahua, J. Ruiz, Data selection based on decision tree for SVM classification on large data sets, Appl. Soft Comput. 37 (2015) 787–798.

Jair Cervantes received the B.S. degree in Mechanical Engineering from Orizaba Technologic Institute, Veracruz, Mexico, in 2001 and the M.S degree in Automatic Control from CINVESTAV-IPN, México, in 2005. In 2009 he got a Ph.D. in Computer Science at Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN). His research interests include support vector machines, pattern classification, neural networks, fuzzy logic, clustering and genetic algorithms.

Farid García-Lamont received a B.Sc. degree in Robotics from ESIMEIPN, México in 2000; in 2004 obtained a M.Sc. in Automatic Control from the Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV-IPN), México. In 2010 received a PhD degree in Computer Science from CINVESTAV-IPN. His research interests are pattern recognition, applications of artificial intelligence and robotics.

Lisbeth Rodríguez-Mazahua received the B.S. degree in informatic and the M.S. degree in computer science from the Instituto Tecnológico de Orizaba, Veracruz, Mexico, in 2004 and 2007, respectively. In 2012 she got a Ph.D. in Computer Science at Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAVIPN). From 2012–2014, she was a Professor of computer science at Universidad Autónoma del Estado de México, Centro Universitario UAEM Texcoco. Since February 2014 she is doing a postdoctoral research at Instituto Tecnológico de Orizaba. Her current research interests include distribution design of databases, database theory, autonomic database systems, multimedia databases, data mining and Big Data.

Asdrúbal López-Chau, received his B.S. degree in Communications and electronics from ESIME-IPN, Mexico, in 1997; in 2000, received his M.S. the degree in Computer Engineering from CIC-IPN, Mexico; in 2013, his Ph.D. degree in Computer Science from CINVESTAV-IPN, Mexico. Since 2009, he is a computer science researcher at Universidad Autonoma del Estado de Mexico, CU UAEM Zumpango, Mexico.

José Sergio Ruiz Castilla received the M.S. degree in computer Sciences from Post graduate college in Texcoco Mexico in 2005. His reseach interests include geographic information systems, pattern recognition, neural networks.

Adrian Trueba received the B.S. degree in agronomy in 1980 from Agriculture College in Tropical, Tabasco, Mexico and the MS and PHD degrees in Applied Computing and Edaphology in Graduate School, in 1990 and 2004. His research interests include geographic information systems, pattern recognition, neural networks.