PSO-Based Method for SVM Classification on Skewed Data-Sets

Jair Cervantes[1(✉)], Farid García-Lamont[1], Asdrúbal López[3], Lisbeth Rodriguez[2], José S. Ruiz Castilla[1], and Adrián Trueba[1]

[1] Posgrado e Investigación UAEMEX (Autonomous University of Mexico State), Av. Jardín Zumpango s/n, Fracc. El Tejocote, 56259 Texcoco, Mexico
jcervantesc@uaemex.mx
[2] Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba, Orizaba, Veracruz, México
[3] Zumpango University Center, University of the State of Mexico, Texcoco, Mexico

Abstract. Support Vector Machines (SVM) have shown excellent generalization power in classification problems. However, on skewed data-sets, SVM learns a biased model that affects the classifier performance, which is severely damaged when the unbalanced ratio is very large. In this paper, a new external balancing method for applying SVM on skewed data sets is developed. In the first phase of the method, the separating hyperplane is computed. Support vectors are then used to generate the initial population of PSO algorithm, which is used to improve the population of artificial instances and to eliminate noise instances. Experimental results demonstrate the ability of the proposed method to improve the performance of SVM on imbalanced data-sets.

Keywords: Support vector machines · PSO · Imbalanced data sets

1 Introduction

In the past few years, Support Vector Machines (SVM) has shown excellent generalization power in classification problems. However, it has been shown that generalization ability of SVM drops dramatically on skewed data-sets [1, 2], because SVM learns a biased model, which affects the classifier performance. Moreover, the performance of SVM is more affected when the imbalanced ratio is large. Although there are several external techniques to tackle the imbalance in data sets, SMOTE has been one of the most-used approaches among several methods. SMOTE introduces artificial instances in data sets by interpolating features values based on neighbors. In several studies have been shown that SMOTE is better than under-sampling and over-sampling techniques [3–7]. Moreover, SMOTE not cause any information loss and could potentially find hidden minority regions, because SMOTE 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 introduce instances with low information because the new instances are obtained using a linear combination between positive examples which increments the density of points. The best artificial examples

D.-S. Huang and K. Han (Eds.): ICIC 2015, Part III, LNAI 9227, pp. 79–86, 2015.
DOI: 10.1007/978-3-319-22053-6_9

(instances with more information of each class) are in the region between positive and negative instances. Introduce instances in this region could increment the discriminative information of positive instances and improve the performance of a classifier on imbalanced data sets. However, this external region is very sensible to artificial instances. Inadequate instances lead to introduce noise and loss of performance in the classifier. Different artificial instances can cause significant differences in performance. Therefore, artificial instances must be generated carefully.

In this paper, we present a novel algorithm which improves the performance of SVM classifiers on imbalanced data sets. The proposed algorithm uses a hybrid system to generate new examples. Hybrid techniques have been widely used in recent years [8, 9]. Some authors have proposed hybrid techniques to address this problem, but this problem is still a challenge today. In contrast to SMOTE, the examples generated with proposed method are derived from the most critical region for SVMs, called the margin. The margin is the distance between the decision boundary and the closest examples. Other techniques generate examples which are located randomly. The proposed algorithm obtains artificial instances from the most important region. However, although generating new instances in the minority class can improve the performance in SVM classification, this process could introduce noise in the data-set. Moreover, it is particularly difficult to introduce good instances in this region because this region is extremely sensitive. Introducing artificial instances in the data-sets must be generated carefully. To find optimal and synthetic instances is an important step in the proposed algorithm. This is the main reason to combine PSO with SVM in this research. In the proposed algorithm, 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 p_{gi}. Experimental results show that the proposed algorithm can get better performance than traditional models.

The rest of the paper is organized as follows. In Sect. 2, a brief overview to the related work on SVM with imbalanced data sets is presented. Section 3 presents the proposed method. The results of experiments are shown in Sect. 4. Conclusions are in Sect. 5.

2 SVM Classification

Formally, the training of SVM 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

$$y_i = sign\left(\sum \alpha_i y_j K\langle x_i \cdot x_j\rangle + b\right) \tag{2}$$

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:

$$\begin{aligned} &\min \frac{1}{2} w_i^T w_i + C \sum_{t=1}^{l} \eta_i^2 \\ s.t. \quad & y_i\left(w_i^T K\langle x_i \cdot x_j\rangle + b_i\right) \geq 1 - \eta_i \end{aligned} \tag{3}$$

where C is the margin parameter to weight the error penalties η_i. The margin is optimal in the sense of Eq. (3). Formally, given a data-set $\{(x_i, y_i)\}_{i=1}^{n}$ and separating hyperplane $f(x) = w_i^T x + b$ the shortest distance from separating hyperplane to the closest positive example in the non-separable case is

$$\gamma+ = \min\gamma_i, \forall\gamma_i \in class + 1 \tag{4}$$

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

$$\gamma- = \min\gamma_i, \forall\gamma_i \in class - 1 \tag{5}$$

where γ_i is given by

$$\frac{y_i\left(w_i^T K\langle x_i \cdot x_j\rangle + b_i\right)}{\|w\|} \tag{6}$$

3 Proposed Method

In this Section we describe the proposed Imbalanced SVM-PSO system. The proposed Imbalanced SVM-PSO system can be divided into two parts, in the first part is trained a SVM in order to obtain the most important data points from the skewed data set, the second part describe the way of PSO try to optimize the artificial data points generated. The initial population is obtained by generating artificial instances, each instance is defined by $x_{g_i} = (x_1, x_2, \ldots, x_d)$, where d is de dimensionality of each instance. Each PSO particle is defined by $p_i = \left(x_{g1}^i, x_{g2}^i, \ldots, x_{gm}^i\right)$, where m is the number of artificial instances generated which are obtained by

$$v_k = x_{sv+}^i - x_{sv-}^j, \quad k = 1, \ldots, d \tag{7}$$

where x_{sv+}^i is the i^{th} positive support vectors (sv) of X_{tr}^+, and x_{sv-}^j represent the j^{th} sv nearest neighbors of x_{sv+}^i. 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

$$x_g = x_{sv+}^i + \varepsilon \cdot v_k \tag{8}$$

which modified only the i^{th}dimension of x_{sv+}^i.

Finally, we denote $P = [p_1, p_2, \ldots, p_{qm}]^T$ as a $(q \times m)$-dimensional vector, where q is the size of initial population. The problem is determining the artificial instances that improve the performance. The $(q \times m)$-dimensional search space Γ is defined by

$$\Gamma = \prod_{i=1}^{p \times m \times d} \left[\Gamma_{i,\min}, \Gamma_{i,\max} \right] \tag{9}$$

The search space of each individual $x = [x_1, x_2, \ldots, x_d]^T$ is defined by the minimal distance between SV's with different class, i.e.

$$x_{\min,i} = sv_{sv+}^{i} \cdot 1 \tag{10}$$

$$x_{\max,i} = \min_{1 \le k \le n} D\left(sv_i^+, sv_i^-\right) \cdot \varepsilon \tag{11}$$

When appliying a PSO to solve the optimization problem, a swarm of the candidate particles $\left\{P_i^l\right\}_{i=1}^{s}$ are moving in the search space Γ in order to find a solution $\hat{x}$ where s is the size of the swarm and $l \in \{0, 1, \ldots, L\}$ denotes the l^{th} movement of the swarm. Each particle $p(i)$ has a $(q \times m \times d)$-dimensional velocity $v = [v_1, v_2, \ldots, v_{qmd}]^T$ to direct its search, and $v \in V$ with the velocity space defined by

$$V = \prod_{i=1}^{p \times m \times d} \left[V_{i,\min}, V_{i,\max} \right] \tag{12}$$

where $V_{i,\max} = \frac{1}{2}\left(\Gamma_{i,\max} - \Gamma_{i,\min}\right)$. To start the PSO, the candidate particles $\left\{X_i^0\right\}_{i=1}^{s}$ are initialized randomly within Γ and the velocity of each candidate particle is initialized to zero, $\left\{v_j^0 = 0\right\}_{i=1}^{s}$. 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 the velocity of particle P_i wich is changed as follows:

$$V_i(t+1) = wV_i(t) + c_1 r_1(t)(\mathrm{pb(t)} - P_i(t)) + c_2 r_2(t)(\mathrm{gb(t)} - P_i(t)) \tag{13}$$

$$P_i(t+1) = P_i(t) + V_i(t) \tag{14}$$

The choice of parameters w, c_1, $r_1(t)$, $r_2(t)$ and the search space are essential to the performance of the PSO. The input space is expressed by the artificial instances generated. Each particle p_i contains m new artificial instances with dimensionality d. The general process of the algorithm proposed is described in Algorithm 1 and Algorithm 2.

Algorithm 1. General process of the proposed algorithm

Input: Skewed dataset
Output: Final hyperplane H_f

1. Divide the input data set in $X^+, x_i \in X : y = +1, i = 1,\ldots,m$ and $X^-, x_j \in X : y = -1, i = 1,\ldots,n$
2. Obtain training and testing data sets from X^+ and X^- obtain $X_{tr}^+, X_{tr}^-, X_{tf}^+, X_{tf}^-, X_{te}^+, X_{te}^-$ with 70%, 15% and 15% respectively.
3. Train the SVM with the training data set, trainSVM $\left(X_{tr}^+, X_{tr}^-\right)$
4. Obtain Support Vectors x_{svi}^- and x_{svi}^+ from H_1
5. Obtain H_f from $\left(X_{te}^+, X_{te}^-\right)$ using the PSO algorithm described in Algorithm 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 $(q \times m \times d)$ from x_{svi}^- and x_{svi}^+ with eqs. (8) and (9).
2. Set initial velocity of vectors $V_i(i = 1,2,\ldots,q \times m \times d)$ associated with the particles
3. For each position p_i of the particle $P_i(i = 1,\ldots,s)$ which contains artificial data points created from support vectors, train an SVM classifier and compute its fitness function φ.
4. Set the best position of each particle with its initial position, i.e., $\mathbf{pb}_i = P_i(i = 1,\ldots,s)$.
5. Obtain the best global particle **gb** in the swarm.
6. Update the speed of each particle using (13).
7. Update the position of each particle using (14).
8. For each candidate particle p_i, train an SVM classifier and compute its fitness function φ.
9. Update the best position **pb** 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
11. Obtain the best global particle.

Once we have obtained the final hyperplane, it gives us a decision function (Eq. (2)). From this decision function we obtain the performance by testing data set X_{te}^+ and X_{te}^-. In the proposed algorithm, the population size and number of iterations or stop criterion serve like a mechanism to avoid over-learning in training data. Our empirical

studies have determined that size of populations with 10 particles and iterations minor to 100 works appropriately. The fitness function value φ associated with the i^{th} particle P_i is essentially the objective function for the problem. In our case we use the G-mean measure as fitness function which is given by $G - mean = \sqrt{S_n^f \cdot S_n^t}$, where S_n^t and S_n^f represent the Sensitivity and Specificity respectively.

4 Experimental Results

In this section is showed the improvement achieved in SVM by the proposed algorithm. The usefulness of the proposed methodology is checked by means of comparisons using classical implementations to imbalanced data sets. In this study, we have selected a wide benchmark of 18 data-sets selected 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 the experiments all data sets were normalized and the 10 fold cross validation method was applied for the measurements.

The approach is implemented in Matlab. In all the experiments presented were used k fold cross validation. The results of the experiments on skewed data sets are reported in Table 1. In this table the first column indicates the data set, and the other columns report the corresponding AUC, G-mean measure. In the Table, σ represents standard deviations of the proposed method.

The experimental results show that the performance of the proposed method is better than classical implementations when imbalance ratio is large. In all data-sets, the

Table 1. Detailed results table for the algorithm proposed

	Under-sampling		Over-sampling		SMOTE		PM		
Dataset	AUC	*G*	AUC	*G*	AUC	*G*	AUC	*G*	σ
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
Glass0	0.805	0.761	0.801	0.768	0.765	0.725	0.839	0.817	0.023
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	0.871	0.806	0.877	0.886	0.877	0.944	0.936	0.021
ecoli3	0.809	0.787	0.798	0.780	0.741	0.817	0.869	0.836	0.071
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
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
German	0.753	0.728	0.735	0.641	0.785	0.710	0.806	0.74	0.004
Haberman	0.683	0.632	0.652	0.600	0.689	0.634	0.742	0.683	0.089
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
pima	0.696	0.725	0.647	0.718	0.714	0.735	0.742	0.785	0.042
glass2	0.607	0.639	0.624	0.652	0.674	0.725	0.738	0.742	0.020
shuttle	0.950	0.871	0.921	0.853	0.950	0.877	0.961	0.891	0.082

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

5 Conclusions

In this paper a novel method that enhances the performance of SVM for skewed data sets was presented. The method reduces the effect of imbalance ratio by exciting and evolving SV and moving separating hyper plane toward majority class. The method is different from other state of the art methods for two reasons, the new instances are added close to optimal separating hyperplane, and they are evolved to improve the classifier's performance. According to the experiments, the proposed method produces the most noticeable results when the imbalance ratio is big. The principal advantage of the proposed method is the performance improvement on imbalanced data-sets by adding artificial examples. However, the first disadvantage is the computational cost. The proposed method can be used only on small data sets. The computational complexity of the proposed method on medium and large data-sets is prohibitive. In comparison with under-sampling, over-sampling and SMOTE the proposed algorithm is computationally very expensive.

Acknowledgements. This research is supported in part by the UAEM Grant No. 3771/2014/CIB.

References

1. Koknar-tezel, S., Latecki, L.J.: Improving SVM classification on imbalanced data sets in distance spaces. In: IEEE International Conference on Data Mining, pp. 259–267 (2009)
2. Cai, Q., He, H., Man, H.: Imbalanced evolving self-organizing learning. Neurocomput. **133**, 258–270 (2014)
3. Batuwita, R., Palade, V.: Class imbalance learning methods for support vector machines. In: He, H., Ma, Y. (eds.) Imbalanced Learning: Foundations Algorithms and Applications. Wiley, New York (2013)
4. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM Explor. Newsl. **6**(1), 20–29 (2004)
5. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: improving prediction of the minority class in boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003)
6. Wu, G., Chang, E.: KBA: Kernel boundary alignment considering imbalanced data distribution. IEEE Trans. Knowl. Data Eng. **17**(6), 786–795 (2005)

7. Nguyen, H.M., Cooper, E.W., Kamei, K.: Borderline over-sampling for imbalanced data classification. Int. J. Knowl. Eng. Soft Data Paradig. **3**(1), 4–21 (2011)
8. Du, J.-X., Huang, D.S., Wang, X.-F., Gu, X.: Shape recognition based on neural networks trained by differential evolution algorithm. Neurocomput. **70**, 896–903 (2007)
9. Huang, D.S., Du, J.-X.: A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks. IEEE Trans. Neural Netw. **19**(12), 2099–2115 (2008)