A Hybrid Algorithm to Improve the Accuracy of Support Vector Machines on Skewed Data-Sets

Jair Cervantes[1], De-Shuang Huang[2], Farid García-Lamont[1], and Asdrúbal López Chau[1]

[1] Posgrado e Investigación UAEMEX (Autonomous University of Mexico State)
Av. Jardín Zumpango s/n, Fracc, El Tejocote, Texcoco, 56259, Mexico
[2] Department of Control Science & Engineering,
Tongji University Cao'an Road 4800, Shanghai, 201804 China

Abstract. Over the past few years, has been shown that generalization power of Support Vector Machines (SVM) falls dramatically on imbalanced data-sets. In this paper, we propose a new method to improve accuracy of SVM on imbalanced data-sets. To get this outcome, firstly, we used undersampling and SVM to obtain the initial SVs and a sketch of the hyperplane. These support vectors help to generate new artificial instances, which will take part as the initial population of a genetic algorithm. The genetic algorithm improves the population in artificial instances from one generation to another and eliminates instances that produce noise in the hyperplane. Finally, the generated and evolved data were included in the original data-set for minimizing the imbalance and improving the generalization ability of the SVM on skewed data-sets.

Keywords: Support Vector Machines, Hybrid, Imbalanced.

1 Introduction

Many real-world applications show imbalance in data-sets. In these problems the goal in classification problems is to find a function that best generalizes the minority class, usually it is the most significant one. Traditionally, classical classification methods do not perform well on imbalanced data-sets, because they were not designed to address such problems. Support Vector Machines (SVM) have shown excellent generalization power in classification problems. However, it has been shown that this generalization ability of SVM drops dramatically on skewed data-sets [7] [10]. The most widely used techniques to tackle this kind of problems are under-sampling, over-sampling and Synthetic Minority Over-sampling Technique (SMOTE) [2]. Under-sampling, gets the number of instances m in the minority class and selects randomly m instances in the majority class. Over-sampling technique eliminates the imbalance by replicating data instances in the minority class or generating artificial instances from the minority class. SMOTE Generates artificial instances over-sampling the minority class by taking each minority class instance and generating synthetic instances along the line segments, joining any or all of the k minority class nearest neighbors. It does not cause any information loss and could potentially find hidden minority regions.

D.-S. Huang et al. (Eds.): ICIC 2014, LNCS 8588, pp. 782–788, 2014.

However, it could introduce noise in the classifiers which could result in a loss of performance because the algorithm makes the assumption that the instance between a positive class instances and its nearest neighbors is also positive [8]. Several techniques inspired in SMOTE algorithm have been proposed [12][13][1][10][9]. Methods based on Evolve Algorithms (EA) have also proposed in the literature. [11] proposed an algorithm to expands the minority class boundary. The algorithm uses a Random Walk Over-Sampling approach (RWO-Sampling) to balancing different class samples by creating synthetic samples through randomly walking from the real data. In [14] is proposed a hybrid learning model to cope with the problem of imbalanced by evolving self-organizing maps. The authors use GA to evolve the subset of the minority examples into new stage that might discover novel knowledge from the limited and underrepresented minority class. In [3], the authors proposed a classification system in order to detect the most important rules, and the rules which perturb the performance of classifier. That system uses hierarchical fuzzy rules and a GA. Garcia et al. [4] implemented an algorithm which performs an optimized selection of examples from data sets. The learning algorithm 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 this paper, we present a new algorithm to generate artificial instances in order to improve the performance of SVM on imbalanced data-sets. In the proposed algorithm, GA is used to guide the search process of new artificial instances. The artificial instances are evolved using a GA, each generation the instances that best contribute to the performance of the data-set are improved by selection, cross and mutation operators, reducing the likelihood of add noise instances to the classifier.

2 SVM Classification via Genetic Algorithm

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

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

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

$$\gamma- = \min \gamma_i, \forall \gamma_i \in class - 1 \tag{2}$$

where γ_i is given by

$$\frac{y_i\left(w_i^T K\left\langle x_i \cdot x_j \right\rangle + b_i\right)}{\|w\|} \tag{3}$$

The margin is

$$\gamma = \gamma_+ + \gamma_- \tag{4}$$

Methods based on the SMOTE algorithm introduce artificial instances in 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). Furthermore, the region with more information is between support vectors with different label. Introducing artificial instances in this region could help improve the performance of SVM. In order to avoid off noise by adding artificial instances and introduce new artificial data points with high discriminative features, we propose a novel algorithm based on a genetic algorithm. Figure 1 describes the general process of the proposed method.

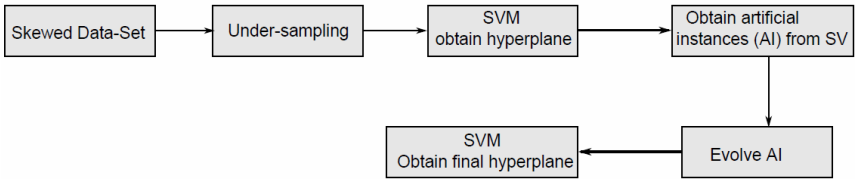

Fig. 1. Proposed method

2.1 Generating Artificial Instances

The proposed algorithm starts using under-sampling and obtaining an initial decision boundary by SVM learning. SVM is trained by X_{tr}^{+} and X_{tr}^{-} to obtain support vectors. The hyperplane and the decision function obtained is skew due to the imbalanced data $\left(X_{tr}^{+}, X_{tr}^{-}\right)$. SV obtained in this first stage of SVM are used to generate the new population using a distance between positive SVs and negative SVs. For each SV in the minority class sv^{+}, the algorithm finds the k nearest neighbors in the

sv^{-} and calculate the distance between them for each dimension. The distance v_i is given by

$$v_i = x_{sv+}^{i} - x_{sv-}^{j}, i = 1,...,d \tag{5}$$

where x_{sv+}^{i} is the i-esime SV of X_{tr}^{+}, and x_{sv-}^{j} represent the j-esime sv^{-} nearestneighbors of x_{sv+}^{i}. Initial vector $v_i = 0, i = 1,...,d.$ and the algorithm pick one or more random entries out of an array. In our experiments we select only one. The artificial instance is obtained by

$$x_g = x_{sv+}^{i} + \varepsilon \cdot v_i \tag{6}$$

which is modified in just the i-esime dimension of x_{sv+}^{i}. Where the step size is ε; we select it between 0.1 and 0.001.

2.2 Genetic Algorithm

SV obtained are used to generate a population for a genetic algorithm. The artificial instances added to data-set could potentially find hidden regions in minority class because these instances were obtained with the most discriminative features of the entire data set. However, the principal disadvantage is that it could introduce noise for the classifier which provokes a loss of performance. On the other hand, classical methods cannot decide which artificial instances can improve the SVM performance in imbalanced data sets because the search space is often huge, complex or poorly understood. The GA objective is to find those evolved instances that improve the SVM's performance. Figure 2 illustrates how it is encoded and decoded each individual in the population. The proposed algorithm has two main parts, the first describes how the artificial data are generated and the second describes how are encoded, decoded and evaluated each individual to converge to a solution.

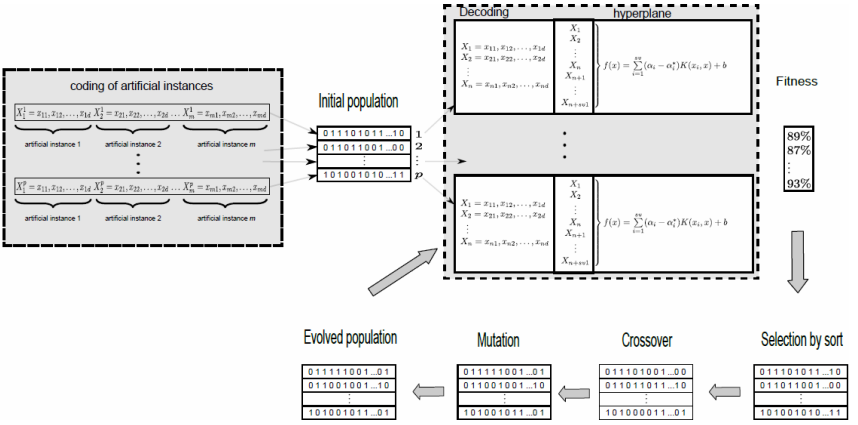

Fig. 2. Encoding and decoding of the proposed method

The initial population is made up by $X_{AI} = \left(x_{AI1}, x_{AI2}, ..., x_{AIp}\right)$ where p is the number of individual in the population and, $x_{AI1} = \left\{x_{g1}, x_{g2}, ..., x_{gm}\right\}$ where $x_g \in R^d$ *i.e.* each individual in a population is a set of artificial instances in the search space. Figure 2 shows the proposed coded and decoded process. To locate the best artificial instances GA needs to encode $x_{AI1} = \left\{x_{g1}, x_{g2}, ..., x_{gm}\right\}$ as binary strings, which are called chromosomes of GA, here all individual artificial instances x_{gi} are converted into binary numbers (we used gray code) and then the binary numbers are concatenated into one string of bits. Each individual in population is

evaluated using a SVM learning adding the SVs found in the initial hyperplane *i.e.* the fitness function is obtained by training a SVM with $x_{svi}^{-} \cup x_{svj}^{+} \cup x_{AIk}$. Furthermore, each individual x_{AI1} is evaluated with the fitness function and manipulated using several genetic operators in order to evolve the population and optimize the solution of the problem. The stronger individuals among the population replace the weaker ones by competition. The survival of the fittest individuals among the population over consecutive generations. In order to prove the performance of the proposed algorithm we used G-mean measure as fitness function in our experiments. The final hyperplane is obtained until a stop criterion has been met. The proposed algorithm stops if there is no improvement in the fitness value for the best string during an interval of the three last generations or if the fitness value is 1.

3 Complexity of the Algorithm

The complexity of the proposed algorithm mainly depends on the fitness evaluation. Other factors that directly influence the complexity are data structures, population, genetic operators, and the implementation of the genetic operators. In the first stage of SVM, the complexity of SVM is O(n2), in our case, it is given for the under-sampled data-set m, then the algorithmic complexity here is O(m)2. In the stage of the GA we added 2 artificial instances for each SV. The simplest case is the roulette wheel selection, point mutation, and two point crossovers with both individuals. The populations represented by fixed length vectors has time complexity $O(gens \times q \times 3SV)$ where gens is the number of generations, $(q \times 3SV)$ is the complexity of point mutation and the time complexity of crossover. Moreover, a SVM is used in order to obtain the fitness function. The complexity of the proposed method does not depend on the entire input data, it just depends on the SVs obtained in the first stage. In the most cases, it is only a small part of the entire data set. The total complexity is given by

$$O(m)^2 + O(gens \times q \times 3m) + O\left(gens \times (3SV)^2\right)$$

Is clear that, the algorithmic complexity is the main disadvantage of the proposed method. However, to use some other learning algorithm to obtain fitness could drastically reduce the computational complexity.

4 Experimental Results

In this section, we show the experimental results obtained with the proposed algorithm. For the case of skewed data sets, is necessary to use a different performance measure to avoid wrong conclusions. In our experiments, we use the sensitivity $S_n^T = (T_p / T_p + F_N)$, specificity $S_n^F = (T_N / T_N + F_P)$, and

G-mean $= \sqrt{S_n^T + S_n^F}$ to evaluate the performance, where T_P , T_N , F_P and F_N are true positives, true negatives, false positives and false negatives respectively.

4.1 Selection Model and Data-Sets

In the experiments all data sets were normalized and 30 runs were executed in each experiment to obtain the results. In this paper, we use two point crossover and flip bit mutation. In the experiments, we used a crossover probability of p_c = 0.9, and a mutation probability of $p_m = 1/n$, where n is the string length for Gray coded. We use the radial basis function (RBF) as kernel, In order to select the optimal parameters of SVM, the hyper-parameter space is explored with some values of γ, and the regularization parameter C.

We use several benchmark data sets from the KEEL data-set repository (http:// sci2s.ugr.es/ keel/datasets.php). Table 1 shows the data sets used in the experiments. The approach was implemented in Matlab. We use 70% of data sets to training, 15% to obtain fitness from artificial data points and, 15% to testing in order to perform the classification of unseen examples. The results are reported in Table 1. In this Table we show results obtained with Under-sampling, Over-sampling, Synthetic Minority Over-sampling Technique (SMOTE) and Proposed Method (PM).

Table 1. Detailed results table for the algorithm proposed

Data-set	Over-sampling			Under-sampling			SMOTE			PM		
	S_n^T	S_n^F	G	S_n^T	S_n^F	G	S_n^T	S_n^F	G	S_n^T	S_n^F	G
Liver_disorders	0.68	0.69	0.68	0.64	0.75	0.69	0.89	0.28	0.49	0.92	0.80	0.85
fourclass	0.78	0.78	0.78	0.81	0.8	0.80	0.91	0.71	0.80	1.00	1.00	1.00
pima	0.67	0.82	0.74	0.72	0.79	0.75	0.74	0.82	0.77	0.77	0.94	0.85
glass0	1.00	0.44	0.66	0.99	0.47	0.68	1.00	0.47	0.68	0.92	0.72	0.81
glass1	0.84	0.46	0.62	0.8	0.57	0.67	0.91	0.31	0.53	0.84	0.97	0.90
cleveland0vs4	0.95	0.70	0.81	0.20	0.99	0.44	0.60	0.99	0.77	1.00	0.92	0.95
segment	0.65	0.84	0.73	0.70	0.83	0.76	0.72	0.8	0.78	0.72	0.90	0.80
diabetes	0.74	.071	0.72	0.69	0.76	0.72	0.83	0.62	0.71	0.87	0.79	0.82
ecoli1	0.92	0.84	0.87	0.89	0.86	0.87	0.91	0.84	0.87	0.98	0.89	0.93
ecoli3	0.93	0.83	0.87	0.89	0.88	0.88	0.83	0.92	0.87	1.00	0.96	0.97
pagebloks13vs4	0.98	0.90	0.93	0.90	0.98	0.93	0.94	1.00	0.96	1.00	1.00	1.00
yeast4	0.75	0.87	0.80	0.71	0.88	0.79	0.54	0.97	0.72	1.00	0.89	0.94
yeast6	0.86	0.88	0.87	0.81	0.92	0.86	0.70	0.98	0.82	0.92	0.94	0.92
vehicle2	0.97	0.91	0.93	0.82	0.98	0.89	0.97	0.93	0.94	1.00	0.96	0.97
vehicle3	0.76	0.67	0.71	0.57	0.82	0.68	0.88	0.66	0.76	0.87	0.85	0.85
shuttle	0.81	0.88	0.84	0.78	0.89	0.83	0.84	0.87	0.85	0.87	0.92	0.89

In the results, we can observe that the proposed method obtains better classification results than the other techniques for almost all the imbalanced data-sets. SVM with GA has a positive synergy with the lateral searching of artificial instances, and leads to good global behavior, in the obtained results, when imbalance ratio is large, the performance achieved by the proposed method is better than the traditional methods.

5 Conclusions

In this paper, we proposed a novel method that enhances the performance of SVM on skewed data sets. The proposed method introduces artificial instances which are created modifying the SVs found in a first stage of SVM training. These artificial instances are evolved by using a genetic algorithm. According to the experiments, the proposed method produces noticeable results in comparison with actual implementations.

References

1. Akbani, R., Kwek, S.S., Japkowicz, N.: Applying Support Vector Machines to Imbalanced Datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
2. Chawla, N.V., Bowyer, K.W., Kegelmeyer, W.P.: Smote: Synthetic minority oversampling technique. J. of Artificial Intelligence Research 16, 321–357 (2002)
3. Fernández, A., de Jesus, M.J., Herrera, F.: Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets. International Journal of Approximate Reasoning 50(3), 561–577 (2009)
4. García, S., Derrac, J., Triguero, I., Carmona, C.J., Herrera, F.: Evolutionary-based selection of generalized instances for imbalanced classification. Knowledge-Based Systems 25(1), 3–12 (2012)
5. Koknar-tezel, S., Latecki, L.J.: Improving SVM Classification on Imbalanced Data Sets in Distance Spaces. In: IEEE Int. Conference on Data Mining, pp. 259–267 (2009)
6. Wang, B.X., Japkowicz, N.: Boosting support vector machines for imbalanced data sets. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) ISMIS 2008. LNCS (LNAI), vol. 4994, pp. 38–47. Springer, Heidelberg (2008)
7. Wu, G., Chang, E.: KBA: Kernel boundary alignment considering imbalanced data distribution. IEEE Trans. Knowl. Data Eng. 17(6), 786–795 (2005)
8. Zeng, Z.-Q., Gao, J.: Improving SVM classification with imbalance data set. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009, Part I. LNCS, vol. 5863, pp. 389–398. Springer, Heidelberg (2009)
9. Zhang, H., Li, M.: RWO-Sampling: A random walk over-sampling approach to imbalanced data classification. To appear in Information Fusion (2014)
10. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005)
11. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newsl. 6(1), 20–29 (2004)