Selecting Minority Examples from Misclassified Data for Over-sampling

Jorge de la Calleja
Computer Science Department
I.N.A.O.E.
Tonantzintla, Puebla, 72840, Mexico

Olac Fuentes
Computer Science Department
University of Texas at El Paso
El Paso, Texas, 79968, U.S.A.

Jesús González
Computer Science Department
I.N.A.O.E.
Tonantzintla, Puebla, 72840, Mexico

Abstract

We introduce a method to deal with the problem of learning from imbalanced data sets, where examples of one class significantly outnumber examples of other classes. Our method selects minority examples from misclassified data given by an ensemble of classifiers. Then, these instances are over-sampled to create new synthetic examples using a variant of the well-known SMOTE algorithm. To build the ensemble we use the bagging method and locally weighted linear regression as the machine learning algorithm. We tested our method using several data sets from the UCI machine learning repository. Our experimental results show that our approach obtains very good results, in fact it showed better recall and precision than SMOTE.

Introduction

The class imbalance problem has received more attention in recent years, because many real-world data sets are imbalanced, i.e. some classes have a lot more examples than others. This situation makes the learning task difficult, as learning algorithms based on optimizing accuracy over all training examples will tend to classify all examples as belonging to the majority class.

Some examples of applications with imbalanced data sets include text classification (Zheng, Wu, and Srihari 2004), cancer detection (Chawla et al. 2002), searching for oil spills in radar images (Kubat and Matwin 1997), detection of fraudulent telephone calls (Fawcett and Provost 1997), astronomical object classification (de la Calleja and Fuentes 2007), and many others. In these applications we are more interested in the minority class rather than the majority class. Thus, we want accurate predictions for the positive class, perhaps at the expense of slightly higher error rates for the majority class.

In this paper we present a method to select minority examples from misclassified data given by an ensemble of classifiers. We use those examples that belong to the minority class to create synthetic examples with a variant of the well-known SMOTE method. We use bagging as the ensemble method and locally weighted linear regression as the machine learning algorithm.

The paper is organized as follows: Section 2 gives a brief description of related work. In Section 3 we describe our proposed method for dealing with imbalanced data sets. In Section 4 we show experimental results, and finally, in Section 5 conclusions are presented.

Related Work

The problem of imbalanced data sets has been addressed from two main approaches. The first one consists of sampling data, i.e. under-sampling the majority class examples or over-sampling the minority class examples, in order to create balanced data sets (Chawla et al. 2002; Japkowicz 1997; Kubat and Matwin 1997). The second is the algorithm-based approach, which focuses on creating or modifying extisting algorithms (Domingos 1999; Pazzani et al. 1994).

We now describe some methods based on on the data sampling approach. Kubat and Matwin (Kubat and Matwin 1997) presented an heuristic under-sampling method to balance the data set eliminating the noisy and redundant examples of the majority class, and keeping the original population of the minority class. Japkowicz (Japkowicz 1997) experimented with random re-sampling which consisted of re-sampling the positive class at random until it contained as many examples as the majority class; another method consisted of re-sampling only those minority examples that were located on the boundary between the minority and majority classes. Chawla et al. (Chawla et al. 2002), devised a method called Synthetic Minority Over-sampling Technique (SMOTE). This technique creates new synthetic examples from the minority class; its nearest positive neighbors are identified and new positive instances are created and placed randomly in between the instance and its neighbors. Akbani et tal. (Akbani, Kwek, and Japkowicz 2004) proposed a variant of the SMOTE algorithm combined with Veropoulos et al's different error costs algorithm, using support vector machines as the learning method. SMOTEBoost is an approach introduced by Chawla et al (Chawla et al. 2003) that combines SMOTE and the boosting ensemble. Hui Han et al. (Han, Wang, and Mao 2005) presented two new minority over-sampling methods: borderline-SMOTE1 and borderline-SMOTE2, in which only the minority examples near the borderline are over-sampled. Recently, Liu et al (Liu, An, and Huang 2006), proposed an ensemble of SVMs with an integrated sampling technique, which combines both over-sampling and under-sampling. They first re-balance the data using over-sampling and under-sampling. Then, each bootstrap sample is combined with the over-sampled positive instances to form a training set to train an SVM. Therefore, N SVMs can be obtained from N different training sets. Finally, the N SVMs are combined to make a prediction on a test example by casting a majority vote from the ensemble of SVMs.

Our Method: SMMO

Ensembles of classifiers are often used to improve the accuracy of single learning algorithms (Dietterich 1997). However, we have used them in a different way, i.e. instead of identifying those examples correctly classified, we find the misclassified examples.

We adopt this strategy because those examples closer to the boundary are frequently misclassified, that is they are more difficult to identify, and then more important for classification. Therefore, these examples may contribute to train better classifiers that alow us to correctly classify more minority class examples.

The main idea of our approach is to use an ensemble of n classifiers to select those misclassified examples that belong to the minority class with the purpose of creating new examples by over-sampling.

Our proposed method performs as follows. We first train n classifiers to create an ensemble, combining their individual decisions by voting to obtain the classification of the examples. Then, we select those misclassified examples, m, that belong to the positive class to create a data set M. Then, we only use the examples in M to create new instances in order to obtain a more dense positive space. Figure 1 shows our proposed method called *SMMO* (Selecting Minority examples from Misclassified data for Over-sampling) to select misclassified minority examples.

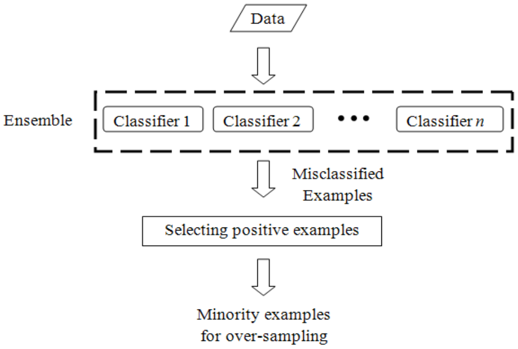

Figure 1: SMMO algorithm.

Because under-sampling the majority class leads to information loss, we decide to create new data by over-sampling the minority examples. To generate the synthetic examples we use a variant of SMOTE that we have employed in previous work (**?**). This method performs as follows: separate positive and negative examples from the original data set. Find the n closest examples to each positive example using the weighted distance. For doing this, we only consider the positive data set. Then, average these n closest instances to obtain the mean example. Take the difference between the minority example and the mean instance. After that, multiply this difference by a random number between 0 and 1, to select a random point. Finally, add the new synthetic positive instance to the original data set. In Table 1 we outline our over-sampling algorithm, called *M-SMOTE*.

Table 1: M-SMOTE algorithm.

```
$D$ is the original data set
$P$ is the set of positive examples
$M$ is the set of positive misclassified examples
for each example $m$ in $M$
    - Find the $n$ closest examples to $m$ in $P$
      using the weighted distance
    - Obtain $A$, the mean of $n$
    - $\delta$ = $m$ - $A$
    - $\eta = m + \delta * \sigma(0, 1)$
    Add $\eta$ to $D$
endfor
```

Ensemble method

An ensemble of classifiers consists of a set of classifiers whose individual decisions are combined in some way, normally by voting. Ensembles often yield better results than individual classifiers. We used *bagging* (Dietterich 1997) to create the ensemble.

The idea of bagging is to randomly generate n subsets with examples from the original training set, and then use each of these subsets to create a classifier. Each subset is obtained by random sampling, with replacement, from the original training set.

Locally Weighted Linear Regression

Locally Weighted Linear Regression (LWLR) is an instance-based learning method. This algorithm simply stores all available training examples, and when it has to classify a new example, it finds similar examples to them. In this work we use a linear model around the query point to approximate the target function.

Given a query point x_q, to predict its output parameters y_q, we assign to each example in the training set a weight given by the inverse of the distance from the training point to the query point:

$$w_i = \frac{1}{|x_q - x_i|} \tag{1}$$

Let W, the weight matrix, be a diagonal matrix with entries $w_1, \ldots, w_n$. Let X be a matrix whose rows are the vectors $x_1, \ldots, x_n$, the input parameters of the examples in the training set, with the addition of a ”1” in the last column.

Table 2: Description of Data sets.

Data set	Examples	% Minority	% Majority	Features
balance	625	7	93	4
car	1728	4	96	6
chess	3196	47	53	36
glass	214	13	87	10
ionosphere	351	35	65	34
nursery	12960	2	98	8
thyroid	215	13	87	5
tic-tac-toe	958	34	66	9
wine	178	26	74	13
yeast	484	3	97	10

Let Y be a matrix whose rows are the vectors $y_1, \ldots, y_n$, the output parameters of the examples in the training set. Then the weighted training data are given by $Z = WX$ and the weighted target function is $V = WY$. Then we use the estimator for the target function defined as $y_q = x_q^T Z^* V$, where Z^* is the pseudoinverse of Z.

Although LWLR is normally applied to regression problems, it is easy to adapt it to perform classification tasks. For an n-class classification problem, we supply as output parameter for each example a vector with n-elements, where the ith element of the vector is 1 if the example belongs to class i and 0 otherwise. When we classify a test example, we assign it to the class with the highest corresponding value in the output vector.

Experimental Results

In order to evaluate the effectiveness of the proposed method, we experimented on some different data sets from the UCI Machine Learning Repository[1]. Given that most of these data sets have more than two classes, we selected those that have the fewest examples to be the minority class, while the other instances were grouped to create the majority class (See Table 2 for details).

In all the experiments reported here we used 10-fold cross-validation. We also varied the amount of over-sampling from 100% to 1000%. In the results we show later correspond to the average of five runs of 10-fold cross-validation.

For creating the ensemble we use bagging and three classifiers of locally weighted linear regression. Also, for the experiments using M-SMOTE and SMOTE we use five nearest neighbors to create new examples.

Since accuracy is not a good metric for imbalanced data sets we evaluate our method using two metrics: *precision* and *recall*, defined as follows:

$$Recall = TP/(TP + FN) \tag{2}$$

$$Precision = TP/(TP + FP) \tag{3}$$

Where TP denotes the number of positive examples that are classified correctly, while FN and FP denote the number of misclassified positive and negative examples, respectively.

In Table 3 we show the performance of our proposed method varying the amount of over-sampling. First, we can note that in seven data sets the best result for recall is over .900, and also for precision in six results is over .900. We can also notice that when we increase the amount for over-sampling the results for recall are better than for precision. The data sets chess, glass and nursery always obtained results over .900 for both measure metrics. From these three data sets, we can remark that nursery, for example, is the data set with the highest degree of imbalance and also has more examples than the others. On the other side, chess is the most balanced data set and also has more features than the other data sets.

In Figures 2 and 3 we compare the performance of our approach SMMO with M-SMOTE and SMOTE using the recall and precision measures. We can see that our proposed method outperforms the other two methods in all data sets. In some of them the difference is significative, for example in balance, car, tic-tac-toe and yeast.

Conclusions

We have presented in this work a method for selecting minority examples from misclassified data using an ensemble of classifiers, with the purpose of over-sampling them. Our experimental results show that our approach obtains very good results, in fact it has better performance than SMOTE in all our experiments.

Future work includes testing the method in real-world applications. For example, classifying astronomical objects or biological structures, where the imbalanced class problem is very common.

Acknowledgement The first author wants to thank CONACyT for supporting this work under grant 166596.

References

Akbani, R.; Kwek, S.; and Japkowicz, N. 2004. Applying support vector machines to imbalanced datasets. In *Proceedings of ECML*, 39–50.

Chawla, N.; Bowyer, K.; Hall, L.; and Kegelmeyer, P. 2002. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence* 16:321–357.

Chawla, N.; Lazarevik, A.; Hall, L.; and Bowyer, K. 2003. Smoteboost: Improving prediction of the minority class in boosting. In *Proceedings of the seventh European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 107–119.

de la Calleja, J., and Fuentes, O. 2007. Automated star/galaxy discrimination in multispectral wide-field images. In *Proceedings of the Second International Conference on Computer VIsion and Applications.*

Dietterich, T. 1997. Ai magazine. *Machine learning research: Four current directions* 18(4):97–136.

Domingos, P. 1999. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th In-*

[1] http://www.ics.uci.edu/∼ mlearn/MLRepository.html

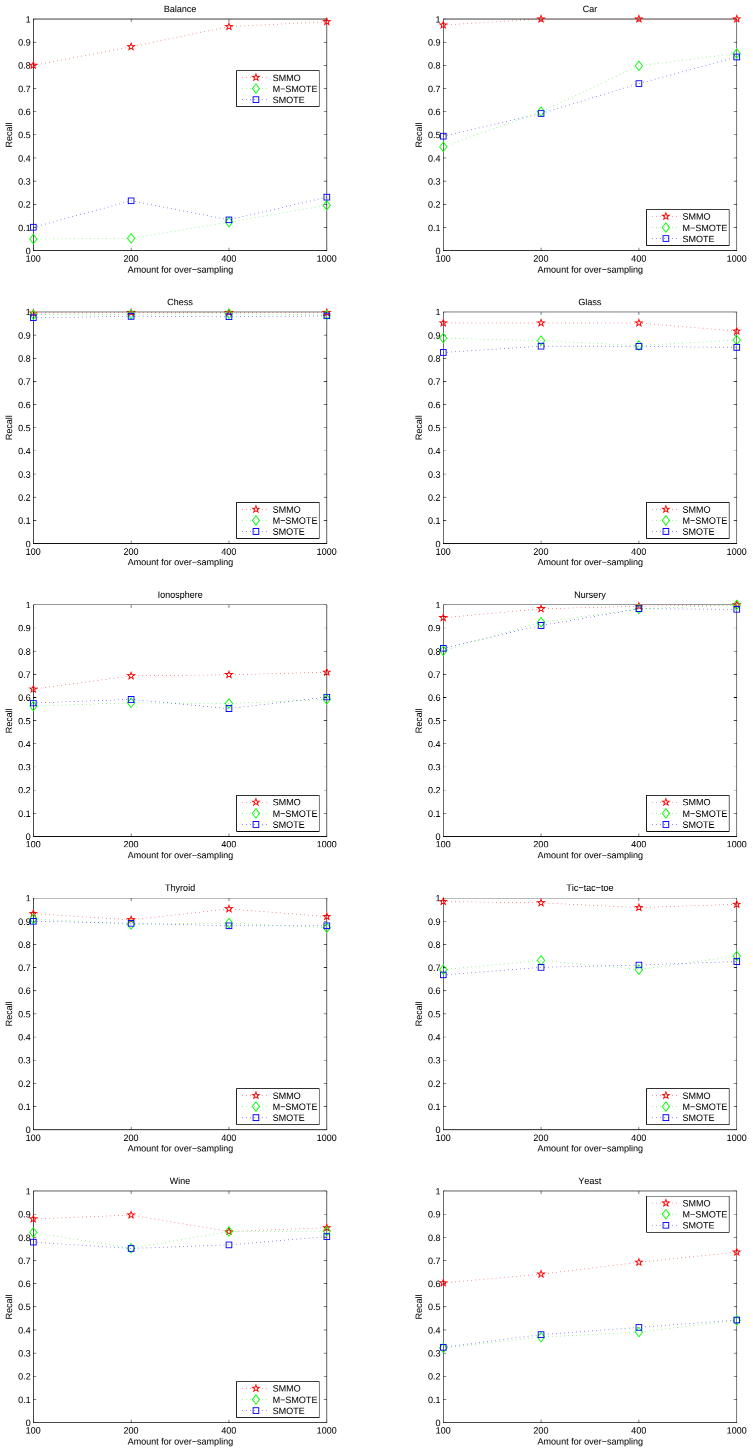

Figure 2: Comparison Recall graphs for the datasets.

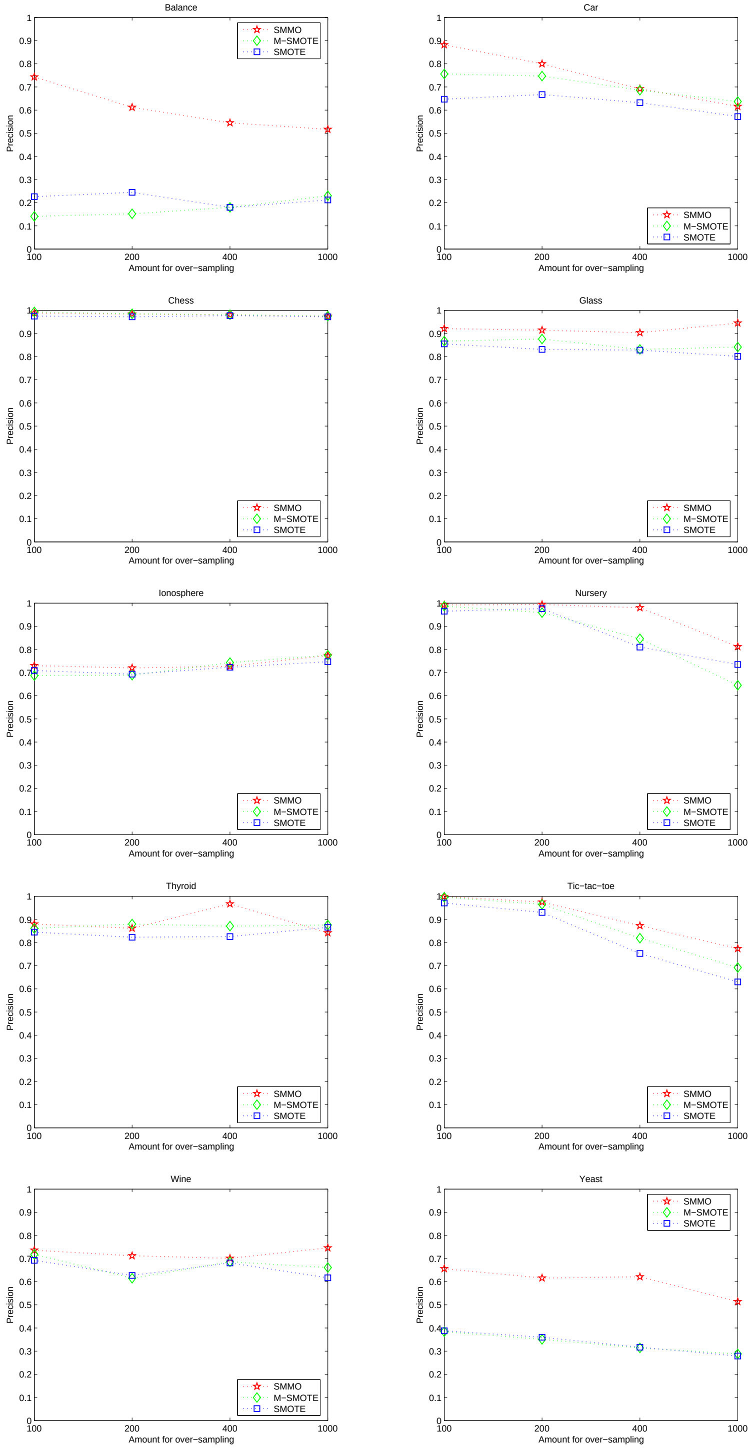

Figure 3: Comparison Precision graphs for the datasets.

Table 3: The table below shows the performance of our proposed method using different amount of over-sampling.

	100%		200%		400%		1000%	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
balance	.800	.743	.880	.612	.967	.545	.988	.517
car	.974	.882	1	.800	1	.692	1	.616
chess	.995	.988	.997	.984	.997	.980	.997	.974
glass	.952	.921	.952	.914	.952	.903	.917	.945
ionosphere	.636	.730	.693	.720	.698	.727	.709	.774
nursery	.944	.992	.983	.994	.994	.980	1	.812
thyroid	.933	.880	.906	.862	.953	.967	.920	.842
tic-tac-toe	.985	.999	.979	.975	.959	.873	.973	.774
wine	.879	.736	.896	.712	.825	.701	.841	.746
yeast	.603	.656	.641	.616	.692	.621	.736	.514

ternational Conference on Knowledge Discovery and Data Mining, 155–164.

Fawcett, T., and Provost, F. 1997. Combining data mining and machine learning for effective user profile. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, 8–13.

Han, H.; Wang, W.; and Mao, B. 2005. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Proceedings of ICIC*, 878–887.

Japkowicz, N. 1997. The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning*, 111–117.

Kubat, M., and Matwin, S. 1997. Addressing the curse of imbalanced training sets: One sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 179–186.

Liu, Y.; An, A.; and Huang, X. 2006. Boosting predicion accuracy on imbalanced datasets with svm ensembles. In *Proceedings of PAKDD, LNAI*, number 3918, 107–118.

Pazzani, M.; Merz, C.; Murphy, P.; Ali, K.; Hume, T.; and Brunk, C. 1994. Reducing misclassification costs. In *Proceedings of the Eleventh International Conference on Machine Learning*, 217–225.

Zheng, Z.; Wu, X.; and Srihari, R. 2004. Feature selection for text categorization on imbalanced data. In *Proceedings of the SIGKDD Explorations*, 80–89.