Boosting Support Vector Machines for Imbalanced Data Sets

Benjamin X. Wang and Nathalie Japkowicz
School of information Technology and Engineering,
University of Ottawa, 800 King Edward Ave., P.O.Box 450 Stn.A,
Ottawa, Ontario, K1N 6N5,Canada
{bxwang,nat}@site.uottawa.ca

Abstract

Real world data mining applications must address the issue of learning from imbalanced data sets. The problem occurs when the number of instances in one class greatly outnumbers the number of instances in the other class. Such data sets often cause a default classifier to be built due to skewed vector spaces or lack of information. Common approaches for dealing with the class imbalance problem involve modifying the data distribution or modifying the classifier. In this work, we choose to use a combination of both approaches. We use support vector machines with soft margins as the base classifier to solve the skewed vector spaces problem. Then we use a boosting algorithm to get an ensemble classifier that has lower error than a single classifier. We found that this ensemble of SVMs makes an impressive improvement in prediction performance, not only for the majority class, but also for the minority class.

1 Introduction

A data set is imbalanced if the number of instances in one class greatly outnumbers the number of instances in the other class. Some examples of domains presenting a class imbalance are: fraudulent telephone calls, telecommunications management, text and image classification, and disease detection. For reasons of simplicity, and with no loss in generality, only binary class data sets are considered in this paper.

Recently, the class imbalance problem has received a lot of attention in the Machine Learning community by virtue of the fact that the performance of the algorithms used degrades significantly if the data set is imbalanced (Japkowicz and Stephen, 2002). Indeed, in very imbalanced domains, most standard classifiers will tend to learn how to predict the majority class. While these classifiers can obtain higher predictive accuracies than those that also try to consider the minority class, this seemingly good performance can be argued as being meaningless.

The next section will discuss some of the approaches previously applied to deal with the class imbalance problem. Section 3 introduces the performance measures we use to evaluate our research. Section 4 discusses the motivations for our approach. Next, Section 5 describes our approach while Section 6 presents the results we obtained. Section 7 is the conclusion.

2 Previous Work

The machine learning community has addressed the issue of class imbalance in two different ways in order to solve the skewed vector spaces problem. The first method, which is classifier-independent, is to balance the original dataset. The second way involves modifying the classifiers in order to adapt them to the data sets. Here we will talk about the most effective approaches that have been proposed. We will discuss these approaches in terms of both their benefits and their limitations.

Balancing the data set. The simplest way to balance a data set is by under-sampling (randomly or selectively) the majority class while keeping the original population of the minority class (Kubat & Matwin, 1997)

Obviously this method results in information loss for the majority class. Over-sampling (Japkowicz & Stephen, 2002; Chawla et al., 2000) is the opposite of under-sampling approach. It duplicates or interpolates minority instances in the hope of reducing class imbalance. With over-sampling, the neighborhood of a positive instance is assumed to be also positive as are the instances between two positive instances. Assumptions like these, however, are data dependent and do not apply in all cases. Experimental results show that under-sampling produces better results than over-sampling in many cases. The belief is that although over-sampling does not lose any information about the majority class, it introduces an unnatural bias in favour of the minority class. Using synthetic examples to augment the minority class is believed be better than over-sampling with replacement (Chawla et al., 2000). It does not cause any information loss and could potentially find “hidden” minority regions. The disadvantage of this method is that it creates noise for the classifiers which could result in a loss of performance. Nonetheless, a method such as this one has the potential of being better than the other approaches discussed since it used a non-skewed mechanism to solve the problem of skewed data.

Modifying the classifiers. Working with classifiers to adapt data sets could be another way to deal with the imbalanced data problem. Assigning distinct costs to the training examples seems to be the best approach of this kind. Various experimental studies of this type have been performed using different kinds of classifiers (Chen et al., 2004; Guo & Viktor, 2004). In terms of SVMs, several attempts have been made to improve their class prediction accuracy (Akbani et al., 2004; Morik et al., 1999). We will discuss them in detail in Section 4. These experiments show that SVMs may be able to solve the problem of skewed vector spaces without introducing noise. However, the resulting classifiers may over-fit the data, as we will discuss later.

In this paper, we present a system that combines the two general methods described for solving the problem of data set imbalance. The system works by modifying the classifier using cost assignation, but counters the modification by using a combination scheme, which is in effect similar to modifying the data distribution. We choose to use boosting as our combination scheme since it is works very well in terms of being able to produce very accurate prediction rules without causing over-fitting. Boosting has the added advantage of working with any type of classifier. In this paper we will focus on support vector machines, which have demonstrated remarkable success in many different applications. Our experiments show that boosting methods can be combined with SVMs very effectively in the presence of imbalanced data. Our results show that this method is not only able to solve the skewed vector spaces problem, but also the over-fitting problem caused by support vector machines.

3 Motivation for our Approach

In this section, we begin by explaining why SVM with soft margins is not sufficient for solving the class imbalance problem. We then discuss methods that have been previously devised in order to improve on this scheme and explain how our methods compares to them. Our scheme will be described in detail in Section 5.

3.1 SVMs and the skewed boundary

Support vector machines are based on the principle of Structural Risk Minimization from statistical learning theory. The idea of structural risk minimization is to find a hypothesis *h* for which we can guarantee the lowest true error. In the presence of noise, the idea of using a soft margin was introduced by Vapnik (1995).

As noted earlier, data imbalance causes a default classifier to be learned which always predicts the "negative" class. Wu and Chang (2003) observed two potential causes for the problem of a skewed boundary: (1) the imbalanced training data ratio and (2) the imbalanced support-vector ratio. For the first cause, we note that on the minority side of the boundary, the positive examples may not always reside as close to the "ideal boundary" as the negative examples. In terms of the second cause, consider the following: according to the KKT conditions, the values for α_i must satisfy $\Sigma_{i=1}^{n} \alpha_i y_i = 0$. Since the values for the minority class tend to be much larger than those for the majority class and the number of positive support vectors substantially smaller, the nearest neighborhood of a test point is likely to be dominated by negative support vectors. In other words, the decision function is more likely to classify a boundary point as negative.

3.2 Analysis of Strategies for the Imbalanced problem for SVMs

To deal with the imbalanced boundary problem, several approaches were given for adjusting the skewed boundary. We first present three approaches, then, in the next section, our strategy for handling this problem.

3.2.1 Kernel transformation method

Adaptively modifying the kernel function K based on the training data distribution is an effective method for improving SVMs. Amari and Wu (1999) propose a method of modifying a kernel function to improve the performance of a support vector machine classifier. This method is based on the structure of the Riemannian geometry induced by the kernel function. The idea is to increase the separability between classes by enlarging the space around the separating boundary surface.
Improving upon Amari and Wu's method, Wu and Chang (2003) propose a class-boundary-alignment algorithm, which also modifies the kernel matrix K based on the distribution of the training data. Instead of using an input space, they conduct the kernel transformation based on the spatial distribution of the support vectors in feature space. A new kernel function is defined as:$\tilde{K}(x, x') = D(x)D(x')K(x, x')$ Where an RBF distance function $D(x) = \sum_{k \in SV} \exp{(-\frac{|x-x_k|}{\tau_k^2})}$ is used as a positive conformal function in this equation.
This method takes advantage of the new information learned in every iteration of the SVM algorithm while leaving the input-space distance unchanged. The class boundary alignment algorithm can be applied directly to adjust the pair-wise object distance in the kernel matrix K in cases where the input space may not physically exist. Theoretical justifications and empirical studies show that kernel transformation method is effective on imbalanced classification, but this technique is not sufficiently simple to be implemented efficiently.

3.2.2 Biased penalties method

Shawe-Taylor & Cristianini (1999) show that the distance of a test point from the boundary is related to its probability of misclassification. This observation has motivated a related technique which is used in his paper. The technique is to provide a more severe penalty if an error is made on a positive example than if it is made on a negative example. By using the cost factors and adjusting the cost of false positives and false negatives, such penalties can be directly incorporated into the SVM algorithm.

Morik et al. (1999) and Shawe-Taylor & Cristianini (1999) propose an algorithm to use the L_1 norm (k=1). Two cost-factors are chosen so that the potential total cost of the false positives equals the potential total cost of the false negatives. This means that the parameters of the SVM are selected such that they obey the ratio: $C_+/C_- = n_-/n_+$. By increasing the margin on the side of the smaller class, this method provides a way to induce a decision boundary which is much more distant from the "critical" class than it is from the other. But in this model, the balance between sensitivity and specificity cannot be controlled adaptively resulting in over-fitting.
Instead of using the L_1 norm for the loss measure, Veropoulos et al. (1999) use the square of the L_2 norm (k=2). This method enables the algorithm to control the balance between sensitivity and specificity, not adding any information. Experimental results (Veropoulos et al., 1999) show that this method has the power to effectively control the sensitivity and not the specificity of the learning machine.

From this analysis we can see that what is really required is a method that is able to introduce some information to the problem in order to increase both sensitivity and specificity.

3.2.3 Boosting method (Our Approach)

Our approach seeks to improve upon Morik et al. (1999)'s method. Instead of increasing C_+ or C_- to get the balance between sensitivity and specificity, we provide another solution that modifies the training data sets x_i in order to adjust some α_i on both the positive and negative side. The respective adjustments are based on the contribution of each. We choose to use boosting, a general method which combines several simple classifiers, to modify the training data sets. The details of this technique are given in the next section.

4 Boosting SVM with Asymmetric Misclassification Cost

Boosting and other ensemble learning methods have been recently used with great success in many applications (Chawla et al., 2003; Guo & Viktor, 2004). In our algorithm, we choose to use the L_1 norm (k=1) SVM (as described in Section 4) with asymmetric misclassification cost for the component classifiers in a boosting scheme. Our method will now be presented formally: Given a set of labeled instances $\{x_i, y_i\}_{i=1}^{n}$, the class prediction function of our base classifier is formulated in terms of the kernel function K:

$$sign(f(x) = \sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + b)$$

where b is the bias and the optimal coefficients are found by maximizing the primal Lagrangian:

$$L_p = \frac{\parallel \vec{\omega} \parallel^2}{2} + C_+ \sum_{\{i|y_i=+1\}}^{n_+} \xi_i^2 + C_- \sum_{\{j|y_j=-1\}}^{n_-} \xi_j^2$$
$$- \sum_{i=1}^{n} \alpha_i [y_i(\omega \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^{n} \mu_i \xi_i$$

where $C_+ \geq \alpha_i \geq 0$ $, C_- \geq \alpha_i \geq 0$ $, \frac{C_+}{C_-} = \frac{n_-}{n_+}$ *and* $\mu_i \geq 0$. Using this component classifier, we find that the points labeled ξ_i^*, where since $\xi_i^* = \xi_i / \|\beta\|$, are said to be on the *wrong side of the margin*, as shown in figure 1. In terms of the L_1 norm margin slack vector optimization, the feasibility gap can be computed since the ξ_i are not specified when moving to the dual. The values for ξ_i can therefore be chosen in order to ensure that the primary problem is feasible. The values are calculated using the following equation:

$$\xi_i = \max(0, 1 - y_i(\sum_{j=1}^{n} y_j \alpha_j K(x_j, x_i) + b))$$

Here, the task consists of modifying the weights ω_i of the training observations x_i in the input space in order to modify the point labeled ξ_i^*. The advantage of this technique is that we are able to easily build a modified version of the training data and improve the class prediction function of the boosting procedure.

Our purpose is to sequentially apply the component classification algorithm to the modified versions of the data, thereby producing a sequence of component classifiers $G_m(x)$, m=1, 2,..., M.

The predictions from all of the component classifiers are then combined by a weighted majority vote to produce the final prediction: $G(x) = sign(\Sigma_{m=1}^{M} \alpha_m G_(m)(x))$. Here the values for α_1,α_2,...α_M are computed by the boosting algorithm and are used to weight the contribution of each respective $G_m(x)$. The resulting effect is to give greater influence to the more accurate classifiers in the sequence.

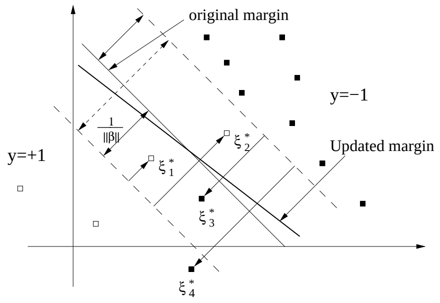

Figure 1: Support vector machines with Asymmetric Misclassification Cost in the imbalanced non-separable case. The points labeled ξ_i^* are on the wrong side of the margin by an amount $\xi_i^* = \xi_i/\|\beta\|$; points on the correct side have $\xi_i^* = 0$. The margin shown results after the update to the points labeled ξ_i^*.

Figure 2 shows the details of our boosting-SVM algorithm. In this algorithm, the classifier S is induced from the current weight observation. The resulting weighted error rate ε is computed as shown at line (c). The weight α_m is then found by calculating $\alpha_m = \lambda \log(1-\varepsilon)/\varepsilon$. Here the λ is an empirical parameter used to tune the magnitude of the penalization for each iteration. We use G-mean instead of prediction accuracy to evaluate the classifier since it combines the values of both sensitivity and specificity. We apply our algorithm on the training data set X_{train} until the G-mean value on the test set $X_{validation}$ cannot be improved.

```
Algorithm Boosting-SVM:
**Given:** Sequence of N examples $X_{Train}$, $X_{Validation}$
M; /* the maximum running iterations */
**Output:** G; /* output ensemble classifier */
**Variables:**
$\omega_i$; /* weights to training observations $(x_i, y_i)$, i=1,2,...,N */
```

```
T; /* the selected running iterations */
$\rho$ /* G-mean value */
Function Calls:
S; /* single classifier */
SVMTrain(X); /* training the single classifier S using SVMs with Asymmetric Cost */
SVMClassify(X,S); /* classify X by the classifier S */
Gmean(G); /* obtain the G-mean value from G */
Begin
Initialize
$\omega_i = 1$, i=1,2,...N
$\rho = 0$; $\rho_{best} = 0$
T=1.
Do for m=1, 2,, M
(a) $X_{train}(x) \leftarrow X_{train}(x)$ using weights $\omega_i$.
(b) $S_m \leftarrow SVMTrain(X_{train})$.
(c) Compute $\varepsilon_m = \frac{\Sigma_{i=1}^{N} \omega_i I(y_i \neq SVMClassify(X_{train}, S_m))}{\Sigma_{i=1}^{N} \omega_i}$
(d) Compute $\alpha_m = \lambda \log(1 - \varepsilon_m)\varepsilon_m \ (0 < \lambda \leq 1)$
(e) Set $\omega_i \leftarrow \omega_i \cdot \exp[\alpha_m \cdot I(y_i \neq SVMClassify(X_{train}, S_m))]$, i=1,2,...,N.
(f) $G_m = sign[\Sigma_{j=1}^{m} \alpha_j S_j]$
(g) $\rho_m = Gmean[G_m(X_{validation})]$
(g) if $\rho_m > \rho_{best}$, then T=m and $\rho_{best} = \rho_m$.
Return $G_t$.
End
```

Algorithm 1. Boosting-SVM with Asymmetric Cost algorithm

The final classification is given following a vote by the sequence of component classifiers. Figure 3 provides an example of a final classifier built from three component classifiers. The ensemble classifier will have lower training error on the full training set than any other single component classifier. The ensemble classifier will also have lower error than a single linear classifier trained on the entire data set.

5 Experiments and Discussion

In our experiments, we compare the performance of our classifier with eight other popular methods: (I)Adacost (Fan et al, 1999), (II)SMOTEboost (Chawla et al., 2003), (III)WRF (Chen et al., 2004), (IV)Databoost-IM (Guo & Viktor, 2004), (V)Under-sampling with SVMs, (VI)SMOTE (Chawla et al., 2000) with SVMs, (VII)SVMs with Asymmetric Cost (Morik et al., 1999), (VIII)SMOTE combined with VII (Akbani et al., 2004). For under-sampling we used a random sampling method. For both under-sampling and SMOTE, the minority class was over-sampled at 200%, 300%, 400% and 500%. We use the same component classifier for all methods. The results obtained were then averaged. For our method and the SVMs with Asymmetric Cost and L_1

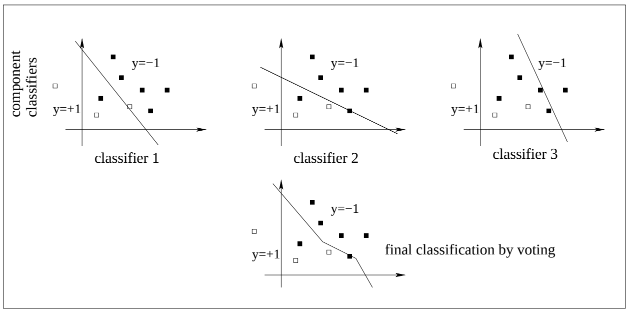

Figure 2: The final classification is given by the voting of the component classifiers and yields a nonlinear decision boundary. The three component classifiers trained by SVM are shown on the top and the resulting classifier is given on the bottom. The ensemble classifier has lower error than a single classifier trained on the entire data set.

norm, we set the cost ratio by: $\frac{C_+}{C_-} = \frac{n_-}{n_+}$. In our experiment we use 10-fold cross-validation to train our classifier since it provides more realistic results than the holdout method. In the boosting schemes we use 70% of the data set for training, 20% to set the threshold for each boosting iteration. The remaining 10% of the data is used as normal in the 10-fold cross validation testing. All training, validation, and test subsets were sampled in a stratified manner that ensured each of them had the same ratio of negative to positive examples (Morik et al., 1999). For all SVM classifiers, we used a linear kernel function to prevent the choice of kernel function from affecting our results.
We chose to experiment on 12 different imbalanced data sets. Abalone19, B-cancer, Car3, Glass7, Heart-disease1, Letter4, Segment and Yeast are from UCI datasets. Lupus-I, Lupus-II, Stroke-I, Stroke-II, are health related data sets.

The next table lists Kubat's G-mean (as a percentage) measure (Kubat & Matwin, 1997) obtained for each algorithm. This measure is more representative of an algorithm's performance.

As a result, when comparing the four approaches we can see that Boosting-SVM with Asymmetric Cost C_+ and C_- yields the best average performance. The result demonstrates that our approach has the power to effectively control both sensitivity and specificity without adding noise. Our approach is always better than SVMs with Asymmetric Cost and L_1 norm since we use it as the component classifier. The improvement in terms of both sensitivity and specificity means that this method is able to avoid over-fitting the data.

6 Conclusion

We have proposed the boosting-SVMs with Asymmetric Cost algorithm for tackling the problem associated with imbalanced data sets. Through theoretical justifications and empirical studies, we demonstrated this method to be effective. We find that our

Table 1: Kubat's G-mean for each algorithm from 10-fold cross validation

DATASET	I	II	III	IV
ABALONE	56.14	56.95	57.39	61.09
B-CANCER	55.64	58.74	58.03	60.01
CAR	91.95	89.13	90.93	91.89
GLASS	**94.41**	91.07	90.54	92.34
H-DISEASE	47.34	47.09	46.60	48.76
LETTER	86.03	87.23	86.22	87.99
LUPUS-I	74.68	74.61	74.56	77.54
LUPUS-II	64.07	63.13	66.00	67.41
SEGMENT	96.10	96.23	95.76	**97.29**
STROKE-I	63.10	63.14	62.88	65.25
STROKE-II	62.04	61.42	62.05	63.30
YEAST	66.00	67.57	69.52	66.94
MEAN	71.46	71.36	71.70	73.32

boosted SVM classifiers are robust in two ways: (1) they improve the performance of the SVM classifier trained with training set; and (2) they are sufficiently simple to be immediately applicable.

In future work, we hope to test more effective boosting methods on our algorithm. We will test this framework on different kernel functions and we will use more efficient measures to evaluate performance in our experiments.

References

[1] Akbani, R., Kwek, S., and Japkowicz, N. (2004), Applying Support Vector Machines to Imbalanced Datasets,in the Proceedings of the 2004 European Conference on Machine Learning (ECML'2004).

[2] Amari,S.,& Wu,S.(1999). Improving support vector machine classifiers by modifying kernel functions. Neural Networks,12,783-789.

[3] N. Chawla, K. Bowyer, L. Hall, & W. P. Kegelmeyer, (2000). SMOTE: synthetic minority over-sampling technique. International Conference on Knowledge Based Computer Systems.

[4] N. Chawla, A. Lazarevic, L. Hall, K. Bowyer (2003). SMOTEBoost: Improving Prediction of the Minority Class in Boosting, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia , 107-119.

[5] Chen C., Liaw, A., and Breiman, L. (2004). Using random forest to learn unbalanced data. Technical Report 666, Statistics Department, University of California at Berkeley.

Table 2: Kubat's G-mean for each algorithm from 10-fold cross validation

Dataset	V	VI	VII	VIII	B-SVM
Abalone	56.27	61.53	78.39	76.92	**79.52**
B-cancer	58.05	60.99	58.63	59.83	**61.89**
Car	76.15	79.50	91.67	91.60	**92.49**
Glass	85.81	89.27	88.22	82.50	91.33
H-disease	47.21	47.86	38.74	47.85	**54.36**
Letter	53.54	70.73	88.90	87.99	**89.66**
Lupus-I	46.80	74.18	75.38	77.17	**84.54**
Lupus-II	55.14	57.41	68.19	67.02	**71.05**
Segment	94.05	95.49	94.31	94.78	96.36
Stroke-I	64.58	64.58	64.23	63.69	**66.70**
Stroke-II	62.29	62.10	62.10	61.86	**64.74**
Yeast	67.08	69.61	66.31	66.87	**71.42**
Mean	63.91	69.44	72.92	73.17	**77.01**

[6] W. Fan, S. Stolfo, J.Zhang, P. Chan (1999). AdaCost: Misclassification Cost-Sensitive Boosting, Proceedings of 16th International Conference on Machine Learning, Slovenia.

[7] H. Guo and HL Viktor (2004). Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach, ACM SIGKDD Explorations, 6(1), 30-39.

[8] N. Japkowicz and S. Stephen (2002). The Class Imbalance Problem: A Systematic Study: *Intelligent Data Analysis, Volume 6, Number 5, pp. 429-450*

[9] Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. Proceddings of the Fourteenth International Conference on Machine Learning, 179-186.

[10] Katharina Morik, Peter Brockhausen, Thorsten Joachims (1999). Combining Statistical Learning with a Knowledge-Based Approach - A Case Study in Intensive Care Monitoring. ICML: 268-277

[11] Shawe-Taylor, J. and Cristianini, N. (1999) Further results on the margin distribution. *In Proceedings of the 12th Conference on Computational Learning Theory*.

[12] Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

[13] Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of support vector machines. *Proceedings of the International Joint Conference on Artificial Intelligence*, 55-60.

[14] Wu, G., & Chang, E. (2003). Adaptive feature-space conformal transformation for imbalanced data learning. *Proceedings of the 20th International Conference on Machine Learning*.