z-SVM: An SVM for Improved Classification of Imbalanced Data

Tasadduq Imam, Kai Ming Ting, and Joarder Kamruzzaman

Gippsland School of Information Technology, Monash University, Australia
{tasadduq, kaiming, joarder}@infotech.monash.edu.au

Abstract. Recent literature has revealed that the decision boundary of a Support Vector Machine (SVM) classifier skews towards the minority class for imbalanced data, resulting in high misclassification rate for minority samples. In this paper, we present a novel strategy for SVM in class imbalanced scenario. In particular, we focus on orienting the trained decision boundary of SVM so that a good margin between the decision boundary and each of the classes is maintained, and also classification performance is improved for imbalanced data. In contrast to existing strategies that introduce additional parameters, the values of which are determined through empirical search involving multiple SVM training, our strategy corrects the skew of the learned SVM model automatically irrespective of the choice of learning parameters without multiple SVM training. We compare our strategy with SVM and SMOTE, a widely accepted strategy for imbalanced data, applied to SVM on five well known imbalanced datasets. Our strategy demonstrates improved classification performance for imbalanced data and is less sensitive to the selection of SVM learning parameters.

Keywords: class imbalance, support vector machine, SMOTE, z-SVM.

1 Introduction

Support Vector Machine (SVM) classifier [1,2] has found popularity in a wide range of classification tasks due to its improved performance in binary classification scenario [3,4,5,6]. Given a dataset, SVM aims at finding the discriminating hyperplane that maintains an optimal margin from the boundary examples called support vectors. An SVM, thus, focusses on improving generalization on training data. A number of recent works, however, have highlighted that the orientation of the decision boundary for an SVM trained with imbalanced data, is skewed towards the minority class, and as such, the prediction accuracy of minority class is low compared to that of the majority ones. Strategies like SVM ensemble trained at varying sampling rate [7,8], SVM with different cost [9] and SMOTE (Synthetic Minority Oversampling Technique) [10,11] have, therefore, been investigated to improve the minority classification accuracy for imbalanced data. A concern regarding the use of these strategies in practical applications is the necessity to pre-select a good value of the parameters that are introduced in

A. Sattar and B.H. Kang (Eds.): AI 2006, LNAI 4304, pp. 264–273, 2006.

the standard SVM learning scheme. Due to the lack of defined guideline on how to select these parameters for imbalanced data, users are required to perform empirical search for the best value of the parameters through multiple training of SVM. Also, it is not much clear as to how a particular value of these parameters affect the SVM hyperplane and the generalization capability of the learned model.

We present in this paper, a novel strategy that addresses the aforementioned issue. In particular, we focus on post adjusting the learned decision boundary of an SVM model so that it maintains a good margin from the data of both the classes, and thereby, improves classification performance in imbalanced data domain. In contrast to the existing approaches, our strategy does not require any parameter pre-selection and auto-adjusts the decision hyperplane based on training data. The rest of this paper is organized as follows. In Section 2, we briefly focus on the theory of support vector learning scheme. Then in Section 3, we discuss the effect of class imbalance and existing techniques to address class imbalance for SVM. Section 4, introduces our proposed scheme followed by some experimental results and comparative discussion in Section 5.

2 Support Vector Machine (SVM)

SVM [1,12] is a discriminant based classifier that focusses on finding the optimal separating hyperplane between class samples by finding the solution for the following quadratic optimization problem:

$$\begin{aligned} min_{\mathbf{w},b} J(\mathbf{w},b) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum \xi_i \\ s.t. \\ y_i(\mathbf{w}.\mathbf{x_i} + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{aligned} \tag{1}$$

where, $(X = \{\mathbf{x_i}\}, Y = \{y_i\})$ denotes the set of feature vectors and set of class labels, respectively for the training dataset. $\mathbf{w}$ is the weight vector for learned decision hyperplane and b is the model bias. ξ_i are the slack variables that, in geometric perspective, indicates how far a particular instance is from its correct side of the decision boundary and is non-zero for examples violating the constraint $y_i(\mathbf{w}.\mathbf{x_i} + b) \geq 1$. The parameter C is a trade-off between maximization of margin and minimization of error, with higher value of C focussing more on minimizing error. With concern that data could be linearly inseparable, SVM exploits the use of kernel functions [1,12] to compute dot product in a mapped high dimensional space and the learning task comprises finding the solution for the following dual problem of (1):

$$\begin{aligned} max_{\alpha} L(\alpha) = \sum \alpha_i - \frac{1}{2}\sum \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \\ s.t. \\ \sum \alpha_i y_i = 0 \end{aligned} \tag{2}$$

$$0 \leq \alpha_i \leq C$$

where $K(.,.)$ is the kernel function, α_i are the Lagrangian multiplicative constants associated with each training data point. At the optimal point for (2), either $\alpha_i = 0$ or $0 < \alpha_i < C$ or $\alpha_i = C$. The input vectors for which $\alpha_i > 0$, are termed as support vectors. These are the only important information from the perspective of classification, as they define the decision boundary, while the rest of the inputs may be ignored. The optimal decision boundary is expressed as,

$$\mathbf{w} = \sum \alpha_i y_i \phi(\mathbf{x}_i) \tag{3}$$

where ϕ is a mapping function such that $K(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x}_i).\phi(\mathbf{x}_j)$.

For classification of a given test instance $\mathbf{x}$, SVM uses the following decision function:

$$f(\mathbf{x}) = \sum_{\mathbf{x_i} \varepsilon SV} \alpha_i y_i K(\mathbf{x}, \mathbf{x_i}) + b \tag{4}$$

where, SV is the set of support vectors. The value of $f(\mathbf{x})$ denotes the distance of the test instance from the discriminating hyperplane, while the sign indicates the class label (either positive or negative).

3 Class Imbalance and Support Vector Machine

Even though the problem of class imbalance has been well known to machine learning research community for a number of years, the study of its effect on SVM learning is relatively recent. It has been pointed out that heavily skewed class distribution causes the solution to (1) or (2), be dominated by the majority class [9,7,11]. We regard in the rest of the paper, the positive class to be the minority class for a binary classification task. As outlined in (1), SVM focuses on maximizing the margin between examples of opposite classes with a penalty for errors. For imbalanced training data, the penalty introduced in the objective function (1) for the relatively small number of positive samples is outweighed by that introduced by the large number of negative samples. As a consequence, the minimization problem in (1) focusses more on maximizing margin from the majority samples, resulting in a decision hyper-plane more skewed towards the minority class.

To cope with this skew, Veropoulos et. al. [9] suggested setting different penalties for misclassification of the classes. But there is no defined indication, as to what values should be preselected as the penalty parameters and the choice is totally empirical. The alternative to this strategy, as has been investigated in literature, is SMOTE (Synthetic Minority Oversampling Technique) [10,11]. The strategy comprises of oversampling the minority class by introducing artificial minority samples based on interpolation between a given minority sample and its nearest minority class neighbours. SMOTE has gained popularity in solving imbalance problem due to its performance. However, this strategy is also dependent on the proper determination of the user dependent parameter, which

in this case is the percentage of oversampling. Also, SMOTE generates noisy artificial data causing high rate of false alarms [8]. Some of the other interesting approaches, that have been suggested, are alignment of kernel boundary by conformal transformation (KBA) [13] and SVM ensembles [7,8]. KBA requires several iterative retraining of an SVM and therefore is excessively expensive in terms of computation time. SVM ensemble combines the output of a set of SVM trained at different sampling ratios of positive and negative examples. But performance of this strategy is also dependent on sampling parameters.

4 Proposed Approach

We present in this section a strategy, that focuses on improving classification performance for imbalanced data by auto-adjusting the hyperplane and reducing skew towards the minority class.

4.1 Mathematical Formulation

The learned weight vector equation of (3) can be re-written as:

$$\mathbf{w} = \sum \alpha_p y_p \phi(\mathbf{x}_p) + \alpha_n y_n \phi(\mathbf{x}_n) \tag{5}$$

and the classification decision equation of (4) can be re-written as:

$$f(\mathbf{x}) = \sum_{\mathbf{x_p} \varepsilon SV; y_p > 0} \alpha_p y_p K(\mathbf{x}, \mathbf{x_p}) + \sum_{\mathbf{x_n} \varepsilon SV; y_n < 0} \alpha_n y_n K(\mathbf{x}, \mathbf{x_n}) + b \tag{6}$$

where, $\mathbf{x_p}$ and $\mathbf{x_n}$ are positive and negative support vectors in a learned SVM model. α_p, y_p and α_n, y_n are the associated Lagrange multipliers and corresponding labels respectively. As equation (6) illustrates, classification depends on a number of factors: the Lagrange multiplication constants associated with positive and negative support vectors, the kernel function and also the number of positive and negative support vectors. For imbalanced training data, these factors are more biased toward negative (majority) class, causing the tendency to classify an unknown instance as negative.

To reduce the bias of a trained SVM to majority class for imbalanced data, we introduce a multiplicative weight, z, associated with each of the positive class support vectors. We thus reformulate equation (6) as:

$$f(\mathbf{x}, z) = z \sum_{\mathbf{x_p} \varepsilon SV; y_p > 0} \alpha_p y_p K(\mathbf{x}, \mathbf{x_p}) + \sum_{\mathbf{x_n} \varepsilon SV; y_n < 0} \alpha_n y_n K(\mathbf{x}, \mathbf{x_n}) + b \tag{7}$$

This may be viewed as weighting the α_p-s such that minority classification is improved. Also, as equation (5) illustrates, the weight vector of a learned SVM model may be expressed as a linear combination of the positive and negative support vectors. Weighting the α_p-s shifts the weight vector from learned position to another position that is further away from the positive class, thereby reducing

the skew towards minority class. Thus the introduced multiplicative weight z is, in effect, a correction of the originally learned boundary of SVM to cope with class imbalance. Technique to determine the appropriate value of z automatically, is presented in next section.

4.2 The Gmean Measure and Determination of z

We focus on setting z to the value that shifts the hyperplane to a position such that the geometric mean (gmean) of the accuracy of positive and negative samples, is maximized for the training data.

The gmean measure [14] is defined as: $gmean = \sqrt{acc_+ * acc_-}$, where acc_+ and acc_- are the classification rate for positive class (sensitivity) and negative class samples respectively. The gmean measure has already been used in a number of research works [11,13] as a performance measure for imbalanced data. Our use of gmean in this context, however, is focussed on shifting trained decision boundary to a well-balanced position to improve classification in imbalanced data scenario. As gmean penalizes heavily the misclassification of small class (eg.,failing to recognize a few minority examples reduces acc_+ drastically) , adjusting the SVM learning according to this measure, in effect, auto-incorporates the class imbalance information for training an SVM.

Returning to our discussion on setting a proper value of z, if we gradually increase the value of z from 0 to some positive value, M, in (7), and use the new model to classify data, the new model will classify everything as negative for $z = 0$ and classify everything as positive for $z = M$. Since gmean is a function of accuracy of both classes, it's value will increase from 0 (at some point $z = z^l$) to some maximum value (at $z = z^*$) and drop to 0 again (at $z = z^h$). The problem is an unconstrained optimization problem of single variable z and is stated in (8).

$$max_z J(z) = \sqrt{\frac{\sum_{\mathbf{x_u} \epsilon X; y_u > 0} I(y_u f(\mathbf{x_u}, z))}{P} . \frac{\sum_{\mathbf{x_v} \epsilon X; y_v < 0} I(y_v f(\mathbf{x_v}, z))}{N}} \quad (8)$$

where, (X, Y) denotes the set of of training vectors. P and N are the total number of positive and negative training vectors respectively. $I(yf(\mathbf{x}, z))$ is a function such that it has value 1 if $yf(\mathbf{x}, z) \geq 0$ and value 0, otherwise. $(\mathbf{x_u}, y_u)$ and $(\mathbf{x_v}, y_v)$ are the set of positive and negative training vectors respectively. The term $\sum_{\mathbf{x_u} \epsilon X; y_u > 0} I(y_u f(\mathbf{x_u}, z))$ calculates the number of positive class training samples correctly classified. Similarly the term $\sum_{\mathbf{x_v} \epsilon X; y_v < 0} I(y_v f(\mathbf{x_v}, z))$ indicates the same of negative class training samples. Overall, the function $J(z)$ formulates the gmean measure on the training set.

A univariate unconstrained optimization technique is used to determine the optimal $z = z^*$. For our strategy, we have applied Golden section search algorithm [15] for the optimization process. The derived z^* value is used to make classification decision according to (7). We call our approach z-SVM.

4.3 Runtime Complexity

The runtime complexity of our approach is computed as follows. The runtime complexity of an SVM algorithm depends on the the number of training points and the kernel computation involved in quadratic optimization. Hence for training set of size N, the complexity of SVM learning is $O(N^3)$, with some practical software approximating it close to $O(N^2)$ [16]. For the extra processing involved in our approach, the overhead is only due to the number of gmean evaluations. For each of these evaluations, the time complexity involved is $O(N^2)$ due to kernel evaluation for each of the data points. For the search strategy, the total number of gmean evaluations with respect to training size is $O(1)$. Hence the total runtime complexity of z-SVM, derived from an SVM trained using $O(N^2)$ complexity algorithm, is $O(N^2)$.

5 Experiments and Results

5.1 Experimental Settings

We investigated the effect of incorporating the z-SVM strategy on a learned SVM model using five well known UCI datasets, as presented in Table 1. The original dataset files have been processed as in [13]. A 10-fold cross validation was used for each of the datasets.

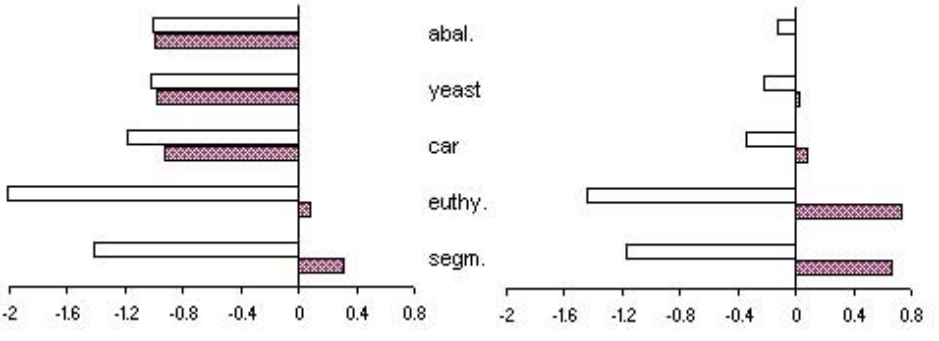
As performance measure, we have used gmean and sensitivity (minority class recognition accuracy). We have not considered Area under ROC (AUC) [17]. Our experiments show that for highly imbalanced datasets, even when the sensitivity is 0, AUC value is considerably high. For instance, an SVM trained on highly imbalanced dataset as abalone, yeast and car, results in AUC values 0.6765, 0.8573 and 0.9889 respectively, while the corresponding sensitivity values are 0. This is because AUC is more focussed on how a classifier relatively ranks the positive and negative class, rather than what label is predicted for a particular test data. Hence for deterministic classifier like SVM, AUC calculated using the decision score of SVM model [17] can be high even when the actual prediction performance is low, especially for minority class. As such we focussed on the use of gmean and sensitivity as performance measure.

5.2 Effect of z-SVM on Decision Boundary

To investigate the effect our algorithm on decision boundary, we trained SVM model for each of the five datasets.We then applied z-SVM on these models. Fig. 1 indicates average distance of the positive (filled up bars) and negative (blank bars) training samples from the learned boundary. It is evident that, for highly imbalanced data (abalone, car, yeast), the trained positive examples fall on average on the wrong side of the hyper-plane, implying that in test environment, that model is more likely to misclassify positive samples. Also, the average distance of decision hyperplane from positive examples is relatively much less than that of negative examples, implying the skew of hyperplane towards the

Table 1. Five UCI datasets with extreme imbalance to moderate imbalance

Data set	Total	Positive (%Pos.)	Negative (%Neg.)
abalone	4177	32 (0.77%)	4145 (99.23%)
yeast	1484	51 (3.44%)	1433 (96.56%)
car	1728	69 (3.99%)	1659 (96.01%)
euthyroid	2000	238 (11.90%)	1762 (88.10%)
segmentation	210	30 (14.29%)	180 (85.71%)



(a) SVM **(b)** z-SVM

Fig. 1. Average distance of examples from the trained hyperplane for (a) SVM and (b) z-SVM derived from SVM. White bar and shaded bar indicate the distance for negative and positive classes respectively, with positive class being the minority.

positive (minority) class. The z-SVM algorithm attempts to reduce the skew in learned decision boundary by keeping a relatively balanced margin between the decision hyperplane and class points. Comparison of Fig. 1(a) and (b) illustrates that z-SVM has been successful in overcoming the error in training by shifting the decision boundary away from positive class and thus increasing the margin of hyperplane from positive class data. The better positioning of the hyperplane in z-SVM means a test sample is more likely to be classified correctly than that in SVM.

5.3 Performance of z-SVM

In Table 2, we present the effect of our strategy on the performance of SVM trained with with Radial Basis kernel and $\gamma = 1$ ($K(x_1, x_2) = exp^{(-\gamma||x_1 - x_2||^2)}$).

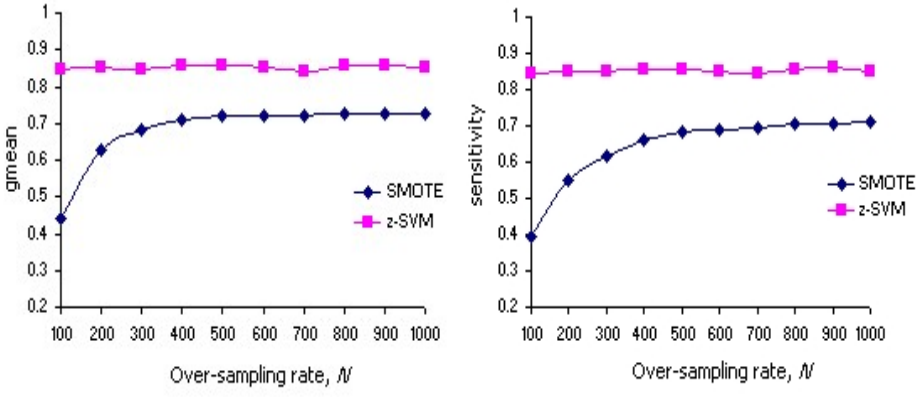
Table 2 indicates that z-SVM improves the prediction capability of a standard SVM in terms of gmean and sensitivity, indicating a good classification performance on imbalanced data. Incorporation of z-SVM allows the model to recognize minority samples, even when the original model fails to recognize any minority sample (as is the case for abalone, yeast and car dataset). A comparison with SMOTE employing 100% oversampling rate (Table 2) shows that z-SVM, derived from standard SVM, performs much better than SMOTE in terms of gmean and sensitivity when the dataset is highly imbalanced, while showing slight improvement or comparable results on moderately imbalanced data. The results imply that, for any degree of imbalance in dataset, z-SVM performs better or at least comparable to SMOTE.

Table 2. Performance comparison for different SVM based techniques with Radial Basis kernel and γ=1. Oversampling rate,N=100% for SMOTE.

Data	Scheme	Gmean	Sens.
abalone	SVM	0.0000	0.0000
	SMOTE	0.0000	0.0000
	z-SVM	0.6202	0.6267
yeast	SVM	0.0000	0.0000
	SMOTE	0.0000	0.0000
	z-SVM	0.7281	0.6667
car	SVM	0.0000	0.0000
	SMOTE	0.3486	0.1833
	z-SVM	0.9361	0.9167
euthyroid	SVM	0.7259	0.5417
	SMOTE	0.8930	0.8306
	z-SVM	0.9042	0.8686
segmentation	SVM	0.9266	0.8667
	SMOTE	0.9759	0.9667
	z-SVM	0.9759	0.9667

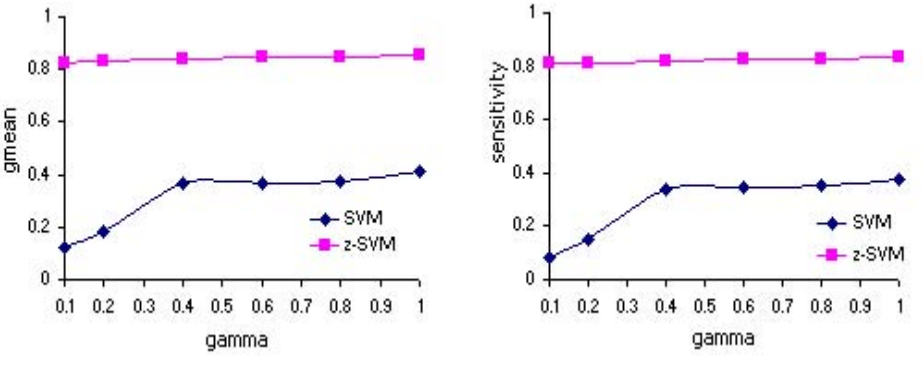
We also investigated the effect of varying oversampling rate (N=100% to 1000%) on SMOTE and z-SVM derived from SMOTE's model. Fig.2 presents the values of gmean and sensitivity, averaged over all five datasets, corresponding to each oversampling rate. Results show that z-SVM outperforms SMOTE over the whole range of oversampling rate.

We also investigated the effect of varying kernel parameters on the performance of SVM. Fig. 3 shows the performance of SVM with kernel parameters at γ = 0.1, 0.2, 0.4, 0.6, 0.8 and 1.0, averaged over all five datasets, along with z-SVM applied to SVM's models. As illustrated, while there is considerable variation in the performance of SVM, the performance of z-SVM is less drastic and remains better over the full range. This shows that z-SVM is relatively less sensitive to user specified kernel parameters.



(a) gmean **(b)** sensitivity

Fig. 2. Average (a) gmean and (b) sensitivity, for change of N for SMOTE and z-SVM over five UCI datasets



(a) gmean **(b)** sensitivity

Fig. 3. Average (a) gmean and (b) sensitivity, for change of γ for SVM over five UCI datasets

6 Conclusion

In this paper we presented a new strategy to improve prediction accuracy of SVM for imbalanced data. The proposed strategy reduces the skewness of the decision boundary towards the minority class and automatically orients the boundary so that a good margin is maintained for each class, which yields better recognition performance for imbalanced data. In contrast to existing schemes like SMOTE which requires multiple training of SVM to select appropriate parameters for good performance, our scheme eliminates such need and is less sensitive to the selection of learning parameters, no matter the base models are derived from

SVM or SMOTE. Future research will be directed towards further alignment of SVM boundary considering individual support vector and spatial distribution of its neighborhood.

References

1. Vapnik, N.V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York. (2000)
2. Schlkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press (2001)
3. Begg, R., Palaniswami, M., Owen, B.: Support vector machines for automated gait classification. IEEE Trans. Biomedical Engineering **52**(5) (2005) 828– 838
4. Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection using neural networks and support vector machines. In: International Joint Conference on Neural Networks. Volume 2. (2002) 1702–1707
5. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: ECML. (1998) 137–142
6. Drucker, H., Wu, D., Vapnik, N.V.: Support vector machines for spam categorization. IEEE Trans. Neural Networks **10**(5) (1999) 1048–1054
7. Yan, R., Liu, Y., Jin, R., Hauptmann, A.: On predicting rare classes with svm ensembles in scene classification. In: Proc. 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03). Volume 3. (2003) III–21–4
8. Liu, Y., An, A., Huang, X.: Boosting prediction accuracy on imbalanced datasets with svm ensembles. In: PAKDD. (2006) 107–118
9. Veropoulos, K., Campbell, C., Cristianini, N.: Controlling the sensitivity of support vector machines. In: International Joint Conference on Artificial Intelligence(IJCAI99). (1999) 55–60
10. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR) **16** (2002) 321–357
11. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: ECML. (2004) 39–50
12. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press (2000)
13. Wu, G., Chang, E.Y.: Kba: Kernel boundary alignment considering imbalanced data distribution. IEEE Trans. Knowl. Data Eng. **17**(6) (2005) 786–795
14. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: ICML. (1997) 179–186
15. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press (1981)
16. Collobert, R., Bengio, S., Bengio, Y.: A parallel mixture of svms for very large scale problems. Neural Computation, **14**(5) (2002) 1105–1114
17. Fawcett, T.: Roc graphs: Notes and practical considerations for researchers (2004)