Majority-Class Aware Support Vector Domain Oversampling for Imbalanced Classification Problems

Markus Kächele, Patrick Thiam, Günther Palm, and Friedhelm Schwenker

Institute of Neural Information Processing, Ulm University, James-Franck-Ring, 89081 Ulm, Germany
{markus.kaechele, patrick.thiam, guenther.palm, friedhelm.schwenker}@uni-ulm.de

Abstract. In this work, a method is presented to overcome the difficulties posed by imbalanced classification problems. The proposed algorithm fits a data description to the minority class but in contrast to many other algorithms, awareness of samples of the majority class is used to improve the estimation process. The majority samples are incorporated in the optimization procedure and the resulting domain descriptions are generally superior to those without knowledge about the majority class. Extensive experimental results support the validity of this approach.

Keywords: Imbalanced classification, One-class SVM, Kernel methods.

1 Introduction and Related Work

Real world machine learning tasks can exhibit several problems that render solving them a severe challenge. Such problems include the unreliability of labels (such as incorrect or missing ones), degraded input data (e.g. by noise or unreliable preprocessing), a very high number of feature dimensions and only a few training samples, and imbalanced training sets, where there are much more samples of one class than the other. In this work, a possible solution for the classification of imbalanced training sets is proposed. Imbalanced datasets are a common problem in machine learning because for many applications, the ease of collecting data from different classes is not equal for each class. For example in medical tasks such as segmentation of cells, the process of collecting samples from healthy patients can be much easier than from patients with a more or less common illness.

The problem with unbalanced datasets is that classifiers that are trained with them usually only learn the larger class (the majority class) because the a-priori probability of a sample belonging to it is much higher than to the other class. A reason for that is that the measure that is usually optimized is classification accuracy, which can already be relatively high by only recognizing the majority class. To overcome this problem, many different solutions have been proposed. The solutions can be grouped into sampling methods, cost-sensitive methods and one-class methods. Sampling methods rebalance the training set by either

N. El Gayar et al. (Eds.): ANNPR 2014, LNAI 8774, pp. 83–92, 2014.

oversampling the minority class [5] or subsampling the majority class [18] (or combinations of both [7]). Subsampling has the advantages that the original data is not changed and that training is faster because of the reduced training set. However the disadvantage that information is thrown away (i.e. samples from the majority class). This issue can be resolved by training more than one classifier and applying ensemble methods on the results [6,12] however with the cost of training additional classifiers.

Oversampling, on the other hand, has the advantage that no information is lost and every sample is used for training. However, in order to enlarge the minority class, synthetic samples have to be generated. This process is critical and care has to be taken when choosing a hypothesis (or data model) to generate novel samples from. If the model is too close to the original data, the possibility of overtraining arises, however if the model is too general, the underlying distribution is lost. A well known algorithm of this category is Chawla et. al's synthetic minority over-sampling technique (SMOTE) [5], in which artificial samples are generated along the connecting lines between neighbouring samples. Many extensions and modifications to the original algorithm have since been proposed. BorderlineSMOTE [9] for example focuses on oversampling of samples that are suspected to be near the decision border. KernelSMOTE (KSMOTE) [20] is an extension that works by finding neighbouring samples in kernel space and then computing new samples using the pre-images in input space.

Other methods exist, that do not alter the training set in any way and rather change the algorithmic treatment of the different classes. Class specific weights/penalty factors can for example be used to instruct the optimization procedure to compensate for classes of different sizes. One way to do this is to introduce class specific boxconstraints for the SVM [15,14,1]. Another possibility is to directly encode the imbalance of the dataset into the creation of the classifier using different loss-metrics [10].

Another possibility are one-class methods, which are used to estimate the support of the minority class and to then generate samples from the inferred model. Popular model choices are Gaussian mixture models or One-class SVMs [17,16]. In [14], this approach has been successfully employed.

A closely related methodology can be found in the field of support vector candidate selection, in which samples are found that will most likely become support vectors in a later classification task (examples that lie near the class boundary for example). This way, the dataset is reduced by discarding uninformative samples and leaving only (potentially) informative ones (for further information, the reader is referred to [13,8]).

In this work, an oversampling method is presented that is based on estimating the support of the minority class using support vector domain description (SVDD). However, the original formulation is modified such that the model is aware of nearby majority class samples by incorporating them in the optimization function using negative weights. In the resulting domain description, regions with a large of number majority class samples but also isolated samples that lie near minority class samples are avoided by adapting the hyperplane to position

them outside of the estimated domain. The domain description is then used to generate new samples to balance the classification problem.

The remainder of this work is organized as follows. In the next section the modified SVDD description is introduced. The sampling algorithm is explained in Section 3. Experimental results are presented in Section 4 together with a discussion, before the work is concluded in the last section.

2 Majority Class Aware Support Vector Domain Description

As in Tax and Duin's original SVDD formulation [17], the task is to find the minimum enclosing ball of radius R of the training samples $x_i \in \mathbb{R}^d$ to an unknown center a. In order to be insensitive to outliers, analogously to the definition of the SVM by Vapnik [19], so called *slack variables* ξ_i are introduced. The parameter C controls the trade-off between accuracy of the model (amount of samples inside the sphere) and generalization (tight fit of underlying distribution; outliers should be identified as such). The original objective was to minimize

$$F(R, a, \xi_i) = R^2 + C \sum_i \xi_i \tag{1}$$

under the constraints $(x - a)^T (x - a) \leq R^2 + \xi_i$ and $\forall i, \xi_i \geq 0$. Since the task here does not only consist of learning a data distribution but also the generation of new samples of a given class with a later classification experiment in mind, the material at hand (the samples of the minority class) is extended by *negative* examples (samples of the majority class), that should be avoided in the model learning task. In order to prevent problems that arise when the much larger majority class is included, an individual weight w_i for each sample is introduced. This way, they can either be switched on or off when needed, or weighted down to prevent domination of the minimization process. The constraints therefore change to:

$$w_i(R^2 - (x_i - a)^T (x_i - a)) + \xi_i \geq 0 \qquad \forall i, \xi_i \geq 0 \tag{2}$$

where $w_i \in \mathbb{R}$ are the sample weights. The constraint is built such that a weight $w_i < 0$ indicates that a sample should be outside the sphere and analogously $w_i > 0$ enforces the placement of the sample inside the sphere. Combining eq. 1 with the constraints and Lagrange multipliers α_i and γ_i leads to

$$\begin{aligned} L(R, a, \alpha_i, \xi_i) &= R^2 + C \sum_i \xi_i - \sum_i \gamma_i \xi_i \\ &- \sum_i \alpha_i \left[w_i (R^2 - \left\{ x_i^2 - 2 \langle a, x_i \rangle + a^2 \right\}) + \xi_i \right] \end{aligned} \tag{3}$$

Determining the partial derivatives with respect to R, a and ξ_i and setting them to 0 yields:

$$\frac{\partial L}{\partial R} = 2R - \sum_i \alpha_i w_i 2R \stackrel{!}{=} 0 \qquad \Rightarrow \sum_i \alpha_i w_i = 1 \tag{4}$$

and

$$\begin{aligned} \frac{\partial L}{\partial a} &= -\left[\sum_i 2\alpha_i w_i x_i - 2\alpha_i w_i a\right] \stackrel{!}{=} 0 \\ \Rightarrow a &= \frac{\sum_i \alpha_i w_i x_i}{\sum_i \alpha_i w_i} \stackrel{(4)}{\Rightarrow} a = \sum_i \alpha_i w_i x_i \end{aligned} \tag{5}$$

and

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \gamma_i \stackrel{!}{=} 0 \qquad \Rightarrow 0 \leq \alpha_i \leq C \tag{6}$$

Substitution of Equations 4, 5 and 6 into Equation 3 and rearrangement yields

$$L(R, a, \alpha_i, \xi_i) = \sum_i \alpha_i w_i x_i^2 - 2\sum_i \alpha_i w_i x_i (\sum_j \alpha_j w_j x_j) + (\sum_j \alpha_j w_j x_j)^2 \tag{7}$$

which leads to the dual form of the original problem:

$$L(R, a, \alpha_i, \xi_i) = \sum_i \alpha_i w_i \left\langle x_i, x_i \right\rangle - \sum_{ij} \alpha_i \alpha_j w_i w_j \left\langle x_i, x_j \right\rangle \tag{8}$$

The dual form has to be maximized under the constraints $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i w_i = 1$. This is a convex function and can be optimized using quadratic programming.
By incorporating a mapping function $\phi : \mathcal{S} \rightarrow \mathcal{F}$ from the domain of the samples to a high dimensional feature space $\mathcal{F}$, the dot products in Equation 8 can be replaced by $\langle \phi(x_i), \phi(x_j) \rangle$, which in turn can be substituted for a *kernel function* $K(x_i, x_j)$ using the kernel trick [3] to achieve non-linear models.

In Figure 1 the effect of the weights on the hyperplane is illustrated. One sample is selected and its weight is gradually decreased until it becomes negative so that the hyperplane starts to bend around the sample to exclude it.

3 Oversampling Using Modified Support Vector Domain Description

In contrast to SMOTE or KernelSMOTE, the proposed algorithm consists of the two phases (1) model building and (2) sample generation:

Phase 1: Model Building

The sample distribution is estimated using support vector domain description. The difference to the original one is that samples of the majority class are weighted negatively in order to keep them outside of the hypersphere. This is done to prevent that samples are generated near negative examples and therefore rendering them contradicting to the original training set. The weights should be determined based on the task at hand either manually or using cross validation.

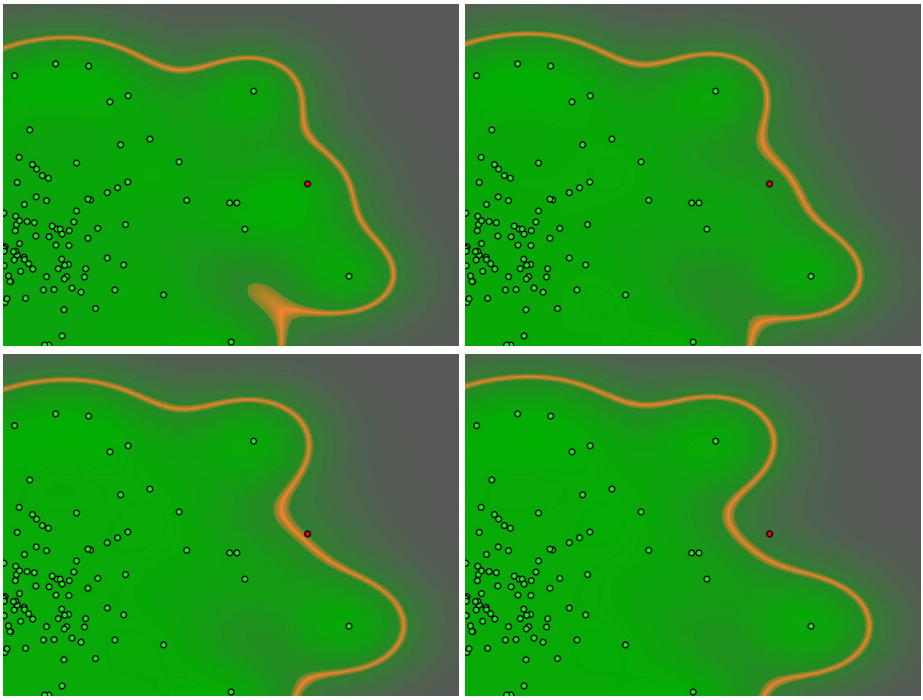

Fig. 1. Effect of sample weights on the hyperplane. By decreasing the weight for the red sample the hyperplane is bent such that the sample begins to traverse the border and resides outside the hypersphere in the end. In this manner, samples can be *forced* to be on either side of the hyperplane with a distance dependent on the magnitude of the weight.

The use of a suitable kernel function might be beneficial to create more complex domain boundaries and thus to allow a better fit of the underlying data distribution. In order to decrease training time, a preprocessing step can be applied to filter majority class samples that are nowhere near minority class samples.

Phase 2: Sample Generation

After the domain description is fit, it is used to infer novel samples. This step can be done using various methods. One possibility is to use rejection sampling to generate new samples by repeatedly drawing random numbers from the respective range of each feature dimension and then checking if the new sample is inside the hypersphere or not. This method has the advantage that it is simple to implement and can also be used to generate samples in regions inside the hypersphere where no minority sample directly resides (i.e. regions that are not directly connected to those regions that hold the original samples). A disadvantage is that the cost of producing new samples is tightly linked to the

dimensionality of the features and also the complexity of the learned model. Possibly a large amount of random numbers will be used to generate the desired number of samples.

Another possibility is to perform a random walk starting from the samples of the minority class and to terminate randomly or when the path leaves the sphere. This algorithm has the advantage that it starts inside the boundary and therefore avoids sampling the empty space around the domain description. The disadvantage is that the area is not sampled uniformly and depending on the termination criterion of the random walk, densely populated regions will attract more samples than sparsely populated ones.

Using one of those methods, the minority class is resampled to approximately the same size as the majority class.

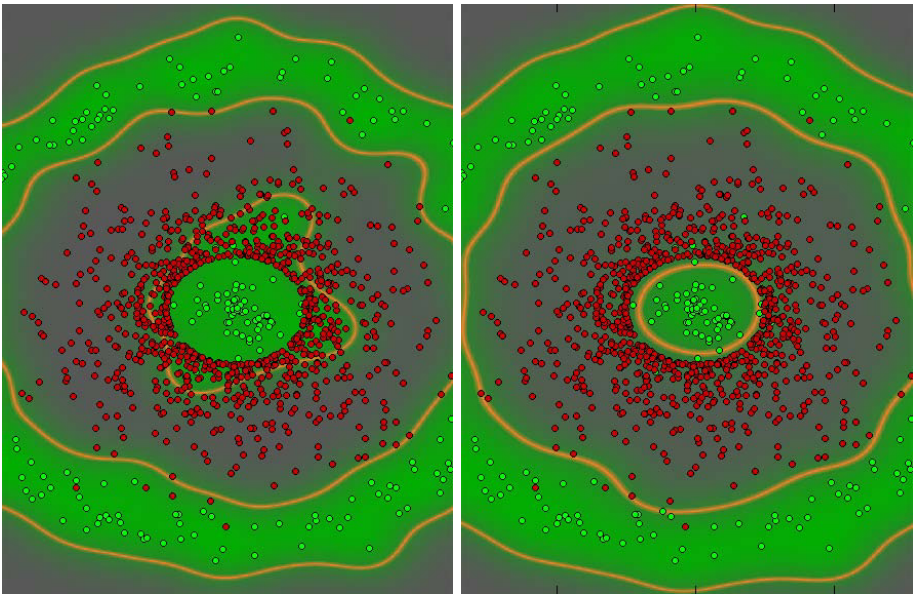

Fig. 2. Estimated sample distribution without and with negatively weighted majority class. In the centers of the figures, the differences are most dominant. On the left side, the model overlaps with the majority class to a great extent, while the center on the right side is constrained to the circle where only the positive samples reside. If the weight is continually increased, the optimization will try to exclude bordering points even more, narrowing the corridor on the outside and the circle in the inside.

4 Experimental Results

Experimental validation of the proposed algorithm has been carried out on a number of freely available, imbalanced datasets and is compared with four different algorithms. The algorithms were selected so, that they cover a broad

spectrum of varieties from simple subsampling over cost-sensitive learning to oversampling. The experiments consisted of classifying imbalanced datasets of different size and with different imbalance ratios. For an overview, the reader is referred to Table 1. The comparison algorithms were:

- Support vector machine with randomly subsampled training sets
- Support vector machine with class specific boxconstraint
- SMOTE
- KernelSMOTE

Table 1. Overview of datasets with their characteristics. The datasets were selected so that a wide range of input dimensionalities, number of instances and imbalance ratios can be found. The datasets *diabetes*, *ecoli* and *glass* are part of the UCI machine learning repository (`http://archive.ics.uci.edu/ml/`).

Name	Dimensionality	Number of instances	Imbalance ratio
diabetes	8	768	1.9
ecoli	7	336	3.4
glass	9	214	4.6
ring	2	1170	3.3

As classification algorithm for SMOTE, KSMOTE and the algorithm proposed here, SVMs were chosen. The experimental setup consisted of stratified k-fold cross validation to obtain different imbalanced subsets (an exception to this was the random subsampling for the first comparison algorithm). Based on those subsets, the methods were trained on $k-1$ subsets by oversampling the minority class to the same size as the majority class and then validated using the remaining subset. Parameter selection involved the *boxconstraint* of the SVM and the kernel parameter γ of the RBF kernel and was carried out using a grid search with a cross validation on a randomly selected subset of the whole dataset of approximately half of the original size. Each experiment was repeated 10 times. To evaluate the performances, the gmean measure as defined in Equation 9 was used

$$gmean = \sqrt{acc^+ * acc^-} \tag{9}$$

where acc^+ and acc^- stand for the rates of true positives and true negatives, respectively. In Table 2 the results are summarized.

As can be seen, the proposed algorithm achieves competitive results and ranks first, together with KSMOTE. The experiments were conducted once with negatively weighted majority samples and once without. The variant with the weights clearly outperforms the one without. Only for the ring dataset the results without weights were better (and only slightly worse than KSMOTE in this case). To see the effects of the weights on the generated hyperplanes, the reader is referred to Figure 2.

Table 2. Summary of experimental results. The values denote the averaged gmean of the classification results from a 3-fold cross-validation for every dataset. To minimize statistical outliers, each experiment was repeated 10 times. As can be seen, the proposed algorithm exhibits superior performance over the remaining algorithms, except for KSMOTE, which performs approximately equally well. The weighted version of the SVDD sampling outperforms the unweighted version in almost every case.

Name	diabetes	ecoli	glass	ring
SVM (bagging)	0.705	0.791	0.913	0.940
SVM (cost)	0.641	0.823	0.918	0.978
SMOTE	0.691	0.882	0.850	0.940
KSMOTE	**0.731**	0.891	0.913	**0.957**
Proposed (w/o weights)	0.696	0.851	0.841	0.956
Proposed (w/ weights)	0.723	**0.905**	**0.945**	0.922

5 Discussion

In comparison to other algorithms such as SMOTE or KSMOTE, the proposed algorithm has the advantage that not only a local neighbourhood is used to infer the new samples. As with other one-class mechanisms, the underlying distribution is estimated and as a result, hypotheses are constructed that can be used to generate new samples. A major advantage of the algorithm as proposed here is that samples of the majority class are also considered in the model fitting phase. Overgeneral minority distributions can be avoided as well as regions where positive and negative samples overlap. If (a subset of) the negative samples are weighted strongly enough, the optimization procedure seeks to put such regions outside the sphere (thereby potentially cutting holes into the domain). The argument that SVM classifiers are able to deal with overlapping regions on their own (even in the same way, since both techniques are very similar) can be extenuated because samples can also be generated for other classification algorithms such as random forests [4] or multi-layer perceptrons. Another advantage is that the samples can effortlessly be generated with accompanying confidence values that indicate how certain a sample belongs to the class. This can be achieved using the distance to the hyperplane. A drawback of the approach is that the support vector domain description is parametrized by (usually at least) two parameters, namely a kernel parameter such as γ and the boxconstraint C. However this poses only a minor problem that can be solved using a grid search in the parameter space. In the experimentation process, the search for proper weights was not critical (i.e. there was no need for an extra cross validation for the weights). It mostly made a difference whether weights were used or not, but their exact value was less important (the reader is again referred to Figure 2).

6 Conclusion and Future Work

In this paper, a method was presented to solve the classification of imbalanced data by sampling the minority class using majority class aware support vector

domain description. First, the data distribution of the class samples is estimated by the algorithm. Then, novel samples are generated using the proposed random sampling techniques. Sampling of new data points in overlapping or bordering regions can be avoided using individual weights that can for example be set negatively for majority class samples. Contrary, important regions can also be highlighted by giving them higher positive weights. Experimental validation was presented to emphasize the feasibility of the proposed mechanism. In future experiments, the applicability of the weighted SVDD for uncertainly labeled data, such as affect in human-computer interaction scenarios [11] will be investigated. The idea is that by incorporating uncertainty in the form of confidence values or fuzzy labels, more reliable models will emerge from the training process. Another idea could be to use the modified SVDD in Co-training like scenarios [2] to iteratively learn different classes and then reweight them using the gained knowledge to extract compact class descriptions.

Acknowledgements. This paper is based on work done within the Transregional Collaborative Research Centre SFB/TRR 62 *Companion-Technology for Cognitive Technical Systems* funded by the German Research Foundation (DFG). Markus Kächele is supported by a scholarship of the Landesgraduiertenförderung Baden-Württemberg at Ulm University.

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. Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-training. In: COLT: Proceedings of the Workshop on Computational Learning Theory. Morgan Kaufmann Publishers (1998)
3. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory, COLT 1992, pp. 144–152. ACM (1992)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
6. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving prediction of the minority class in boosting. Proceedings of the Principles of Knowledge Discovery in Databases (PKDD), 107–119 (2003)
7. Cohen, G., Hilario, M., Sax, H., Hugonnet, S.: Data imbalance in surveillance of nosocomial infections. In: Perner, P., Brause, R., Holzhütter, H.-G. (eds.) ISMDA 2003. LNCS, vol. 2868, pp. 109–117. Springer, Heidelberg (2003)
8. Guo, L., Boukir, S., Chehata, N.: Support vectors selection for supervised learning using an ensemble approach. In: Proceedings of the International Conference on Pattern Recognition (ICPR), pp. 37–40 (August 2010)
9. 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)

10. Hong, X., Chen, S., Harris, C.: A kernel-based two-class classifier for imbalanced data sets. IEEE Transactions on Neural Networks 18(1), 28–41 (2007)
11. Kächele, M., Glodek, M., Zharkov, D., Meudt, S., Schwenker, F.: Fusion of audio-visual features using hierarchical classifier systems for the recognition of affective states and the state of depression. In: De Marsico, M., Tabbone, A., Fred, A. (eds.) Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM), pp. 671–678. SciTePress (2014)
12. Kächele, M., Schwenker, F.: Cascaded fusion of dynamic, spatial, and textural feature sets for person-independent facial emotion recognition. In: Proceedings of the International Conference on Pattern Recognition (ICPR) (to appear, 2014)
13. Li, M., Chen, F., Kou, J.: Candidate vectors selection for training support vector machines. In: Third International Conference on Natural Computation, ICNC 2007, vol. 1, pp. 538–542 (August 2007)
14. Raskutti, B., Kowalczyk, A.: Extreme re-balancing for svms: A case study. SIGKDD Explor. Newsl. 6(1), 60–69 (2004)
15. Schels, M., Scherer, S., Glodek, M., Kestler, H., Palm, G., Schwenker, F.: On the discovery of events in EEG data utilizing information fusion. Computational Statistics 28(1), 5–18 (2013)
16. Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation 13(7), 1443–1471 (2001)
17. Tax, D.M.J., Duin, R.P.W.: Support vector domain description. Pattern Recognition Letters 20, 1191–1199 (1999)
18. Van Hulse, J., Khoshgoftaar, T.M., Napolitano, A.: Experimental perspectives on learning from imbalanced data. In: Proceedings of the International Conference on Machine Learning, ICML 2007, pp. 935–942. ACM, New York (2007)
19. Vapnik, V.N.: Statistical Learning Theory, vol. 2. Wiley (1998)
20. 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)