Amino Acids (2010) 39:1385–1391
DOI 10.1007/s00726-010-0595-2

ORIGINAL ARTICLE

An approach for classification of highly imbalanced data using weighting and undersampling

Ashish Anand · Ganesan Pugalenthi · Gary B. Fogel · P. N. Suganthan

Received: 9 December 2009 / Accepted: 7 April 2010 / Published online: 22 April 2010

Abstract Real-world datasets commonly have issues with data imbalance. There are several approaches such as weighting, sub-sampling, and data modeling for handling these data. Learning in the presence of data imbalances presents a great challenge to machine learning. Techniques such as support-vector machines have excellent performance for balanced data, but may fail when applied to imbalanced datasets. In this paper, we propose a new undersampling technique for selecting instances from the majority class. The performance of this approach was evaluated in the context of several real biological imbalanced data. The ratios of negative to positive samples vary from ∼9:1 to ∼100:1. Useful classifiers have high sensitivity and specificity. Our results demonstrate that the proposed selection technique improves the sensitivity compared to weighted support-vector machine and available results in the literature for the same datasets.

Keywords Imbalanced datasets · SVM · Undersampling technique

Introduction

Imbalanced data sets are ubiquitous in the literature (Chawla et al. 2004; Tang et al. 2009). Some common examples can be taken from domains of text classification (Forman 2003; Mladenic and Grobelnik 1999), fraud detection (Zhang et al. 2004), bioinformatics (Robinson et al. 2007; Sales et al. 2008; Chen and Jeong 2009; Batuwita and Palade 2009a; Wu et al. 2009; Yousef et al. 2006) and medicine (Mazurowski et al. 2008). Data imbalance said to exist when all classes are not present in equal proportion. In many cases, the class of greater interest (e.g., the target class) is outnumbered by the class of lesser importance. The ratio between the two classes can be as extreme as 1:100, 1:1,000. Traditional machine learning approaches applied to these datasets tend to be biased towards prediction of the majority class while user desires for both high sensitivity and specificity. Such problems are very common in bioinformatics including prediction of miRNA genes (Yousef et al. 2006), prediction of protein-interaction sites (Chen and Jeong 2009), prediction of precursor microRNAs (Batuwita and Palade 2009a), and prediction of DNA/RNA binding residues in proteins (Shi et al. 2009; Wang et al. 2008; Wu et al. 2009).

Support-vector machines (SVMs) were introduced by Vapnik (1998) and have been successfully employed in many application areas including text classification (Joachims et al. 1998), handwriting recognition (Cortes 1995), and bioinformatics (Chen et al. 2007; Yang 2004; Sun and Huang 2006; Verma et al. 2009; Wang et al. 2005). However, when faced with imbalanced datasets, their performance may be reduced significantly (Akbani et al. 2004; Tang et al. 2009). One popular remedy for this situation is to increase the penalty associated with the minority class (Osuna et al. 1997; Veropoulos et al. 1999; Akbani et al. 2004). An easy way to achieve this is to use scale factors that are inversely proportional to the number of samples in each class to the total number of samples to

A. Anand · G. Pugalenthi · P. N. Suganthan (✉)
School of Electrical and Electronic Engineering,
Nanyang Technological University, 50 Nanyang Avenue,
Singapore 639798, Singapore
e-mail: EPNSugan@ntu.edu.sg

G. B. Fogel
Natural Selection, Inc, 9330 Scranton Road,
Suite 150, San Diego, CA 92121, USA

establish a cost parameter *C*. This approach is referred to as weighted-SVM. Another approach is to use resampling techniques to create a balanced dataset. Resampling can be accomplished either by oversampling the minority class or undersampling the majority class. Both resampling techniques have their advantages and disadvantages. While oversampling (Akbani et al. 2004) introduces new artificial data, undersampling leads to information loss. We believe that generating artificial data in the context of real biological data is to be avoided as it has the potential to introduce new error into the system that is being modeled. For example, while considering the case of prediction of DNA binding residues in proteins, if one generates artificial data by resembling the given known DNA binding residue, it becomes difficult to know if such residues exist or not exist in real protein sequences. Several studies have compared these approaches for SVM classifiers (Akbani et al. 2004; Tang et al. 2009; Liu et al. 2009). Tang et al. (2009) proposed granular SVM where repetitive undersampling was performed. Starting with the set of all training samples, the SVM was trained. Negative samples which were support vectors (SVs) were removed for the next training samples. This process was iterated until all positive samples and an aggregation of negative samples (based on removed negative SVs at each repetition) were combined to make a final training sample. This approach exploited the ability of SVs in determining useful hyperplanes. In another study, Liu et al. (2009) proposed two approaches for an ensemble of random sampling. Underutilization of large proportion of negative samples was the basic motivation behind the methods proposed by Tang et al. (2009) and Liu et al. (2009). Both methods have been shown to perform better than random sampling over the datasets used for evaluation. Tang et al. (2009) have also shown that weighted-SVM is more stable and very effective over other approaches such as SVM-SMOTE (Akbani et al. 2004) and SVM-random undersampling.

Here, we propose a new undersampling technique targeting the boundary samples which are always difficult to deal with for all classifiers. Instances far from decision boundaries are relatively easy to classify. The proposed technique selects instances from the majority class which are more likely to be near a decision boundary. The proposed undersampling technique is then combined with the weighted-SVM. The new approach is compared to weighted-SVM. In classical SVM, an equal penalty is used for both classes whereas for weighted-SVM, different weights are used for misclassification of the two classes. We further compare the performances of the proposed undersampling technique combined with the weighted-SVM to available results in the literature on the considered datasets.

Materials and methods

Datasets

Four datasets were used in our simulation studies. The ratio of positive to negative samples varied across these datasets from 1:9 to 1:100. There were also significant variations in total number of samples and number of features over all datasets. The datasets are described briefly below.

Active-site dataset

An active-site dataset was obtained from Pugalenthi et al. (2008). 606 non-redundant protein chains containing 2,096 catalytic residues and 205,811 non catalytic residues were selected from the Catalytic Site Atlas database (Porter et al. 2004). Each residue in the dataset was represented by a vector of 100 features. The details of each feature are provided below.

Amino acid type and groups We classified 20 amino acids into 10 groups based on the presence of side chain chemical group such as phenyl (FWY), carboxyl (DE), imidazole (H), primary amine (K), guanidino (R), thiol (C), sulfur (M), amido (QN), hydroxyl (ST) and nonpolar (AGILVP). The frequency of amino acid groups were computed for each catalytic residue and its spatial neighbors (Pugalenthi et al. 2008).

Structural features Structural features such as solvent accessibility, secondary structures, hydrogen bonds and Ooi number (a count of the number of other C-α atoms within a radius of 14 Å of the given residue's own C-α atom. This gives a good impression of which parts of the structure are buried and which are exposed on the surface (Nishikawa and Ooi, 1986) were computed for each residue and its spatial neighbors from the protein structure using JOY package (Mizuguchi et al. 1998).

Spatial neighbors Residues considered as spatial neighbors are likely to have more influence on the selection of catalytic residues. Spatial neighbors were defined as residues within 5 Å distance from catalytic residues in the 3D protein structure. Average sequence conservation, amino acid composition, and composition of 10 amino acid groups were computed from all spatial neighbors. The other structural features computed include secondary structure, hydrogen bond, Ooi number and solvent accessibility.

Physicochemical properties A total of 15 physicochemical properties were calculated from each catalytic residue and its spatial neighbors. The computed properties included molecular weight, hydrophobicity, hydrophilicity, hydration potential, refractivity, average accessible surface area, free energy transfer, flexibility, residue volume, mutability, melting point, optical activity, side chain volume, polarity, and isoelectric points following Kawashima et al. (2008).

Disulfide dataset

The dataset used for the prediction was obtained from the protein data bank (PDB) database (Berman et al. 2000). The protein chain containing at least one disulfide bonds were selected from PDB database. To make the dataset completely non-redundant, proteins chains having higher than 40% sequence identity were removed using CD-HIT (Li and Godzik 2006). After careful manual examination, 1,317 non-redundant protein chains containing 2,341 disulfide bonds were selected for prediction work. The negative datasets (20,678 cysteine pairs) were generated from all possible combinations of cysteines that do not form disulfide bond. Each cysteine pair was represented by a feature vector of 151 features. The details of the features are.

Secondary structure For each sequence secondary structure information was assigned using PSIPRED method (McGuffin et al. 2000). The type of secondary structural element (Helix-H, Strand-E and Coil-C) at two cysteines (CYS1 and CYS2) was computed. Further, the frequency of secondary structural elements was calculated from 11 residues that surround cysteine residues (5 upstream residues and 5 downstream residues from cysteine). The frequencies of secondary structural elements between two cysteines (CYS1 and CYS2) are also included.

Amino acid groups We classified 20 amino acids into 10 groups based on the presence of side chain chemical group such as phenyl (FWY), carboxyl (DE), imidazole (H), primary amine (K), guanidino (R), thiol (C), sulfur (M), amido (QN), hydroxyl (ST) and nonpolar (AGILVP) (Pugalenthi et al. 2008). The frequency of 10 amino acid groups and frequency of hydrophobic, hydrophilic and neutral amino acids were computed from 11 residues that surround cysteine residues. Similarly, the frequency of amino acid groups between two cysteines (CYS1 and CYS2) was computed.

Physicochemical properties Matrices containing quantitative values for 13 amino acid physicochemical properties scaled between 0 and 1 were obtained from the UMBC AAIndex database (Kawashima et al. 2008). The computed properties include molecular weight, hydrophobicity, hydrophilicity, refractivity, average accessible surface area, flexibility, melting point, side chain volume, polarity, isoelectric points, and average frequency of helix and beta sheets. The average physicochemical property from 10 residues that surround cysteine and between two cysteine residues was calculated.

Xwchen-data (Chen and Jeong 2009)

This dataset was downloaded from the website http://ittc.ku.edu/~xwchen/bindingsite/prediction.htm and contained 2,829 interface residues and 24,616 non-interface residues. Each residue was represented by sequence-based feature vector of length 1,050. These features were grouped into the three categories: physic-chemical features and evolutionary conservation score, amino-acid distance, and PSSM. A more detailed description of each feature can be found in Chen and Jeong (2009).

Micropred-data (Batuwita and Palade 2009a)

This dataset was downloaded from the website http://web.comlab.ox.ac.uk/people/ManoharaRukshan.Batuwita/microPred.htm. This dataset contained 691 non-redundant human pre-miRNAs, 8,494 non-redundant human pseudo hairpins (8,494) and 754 non-redundant human other non-coding RNAs. The objective was to classify human precursor microRNA (pre-miRNAs) from both genome pseudo hairpins and other non-coding RNAs. Thus, numbers of positive and negative samples were 691 and 9,248. Each sample was represented by a feature vector of length 48 and each of these features was described in detail in (Batuwita and Palade 2009a).

SVM classifier

SVMs are from the family of margin based classifiers. SVMs can be used to identify optimal hyperplanes with maximal distance between the hyperplane and the nearest samples from each of the two classes. The decision function of SVM is given as $f(x) = w^T\varnothing(x) + b$ where $\boldsymbol{w} = [w_1, w_2,\ldots,w_d]^T$ is the weight vector and b is bias. Given a training set x_i, $i = 1,\ldots,n$, $x_i \in R^d$ and corresponding labels $y_i \in \{-1,1\}$, SVM solves the following optimization problem:

$$\begin{aligned}&\text{Minimize}\,\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{i=1}^{n}\xi_i\\&\text{Subject to}\,y_i(\mathbf{w}.\varnothing(\mathbf{x_i}) + b) \geq 1 - \xi_i\\&\xi_i \geq 0,\ i = 1, 2, \ldots, n\end{aligned} \tag{1}$$

Here C is the penalty or cost parameter which balances the trade-off between training accuracy and generalization. However, the performance of SVM drops significantly

(Wu and Chang 2003; Akbani et al. 2004), when samples from one class (mostly negative or non-target class) outnumber samples from the other class by a large ratio. In this paper, we consider the negative or "non-target" class as the majority class and positive or target class as the minority class. A popular approach is to bias the classifier by increasing the penalty associated with misclassifying the positive class relative to the negative class. For example, Osuna et al. (1997) modified Eq. 1 as follows

$$\begin{aligned}&\text{Minimize}\,\frac{1}{2}\mathbf{w}.\mathbf{w} + C_+\sum_{y_i=1}\xi_i + C_-\sum_{y_i=1}\xi_i\\&\text{Subject to}\,y_i(\mathbf{w}.\varnothing(\mathbf{x_i}) + b) \geq 1 - \xi_i\\&\xi_i \geq 0,\ i = 1, 2, \ldots, n\end{aligned} \tag{2}$$

Our proposed undersampling approach

Samples situated far from the decision boundaries are more likely to be classified correctly. The samples from the two classes lying close to each other are more likely to be wrongly classified by any classifier and the decision boundary is also likely to be in the vicinity of these "boundary samples." The proposed approach has used this property to select negative samples. On other hand, random sampling chooses samples irrespective of their location. If the chosen samples are far from the decision boundary and if the samples located near the true decision boundary are not chosen by the random selection process, the optimal separating hyperplane may be wrongly positioned resulting in increased misclassifications. At the same time, given that random sampling is a non-deterministic approach, several random undersamplings may be required to determine a better estimate of the true predictive performance of the SVM trained by using the random sampling approach. The instability of random sampling was also discussed in Tang et al. (2009). As we use SVM as the basic classifier, we exploited the above discussed property in the undersampling approach summarized as follows:

- Calculate the weighted Euclidean distance of each negative sample from each of the positive samples. All features are weighted by its Fisher's score (Eq. 3). The Fisher ratio $\mathrm{FR}(x_i)$ for each feature x_i is calculated as

  $$\mathrm{FR}(x_i) = \frac{\left(\bar{x}_{i,p} - \bar{x}_{i,n}\right)^2}{\hat{\sigma}_{i,p}^2 + \hat{\sigma}_{i,n}^2} \tag{3}$$

  where, $\bar{x}_{i,p}$ is the mean value of the feature x_i for positive samples, $\bar{x}_{i,n}$ is the mean value for negative samples, $\hat{\sigma}_{i,p}^2$ and $\hat{\sigma}_{i,n}^2$ are the variances of the feature x_i for positive and negative samples, respectively. Thus, for any two samples $X_1 = (x_{11},\ldots,x_{1i},\ldots,\mathrm{x}_{1n})$ and $X_2 = (x_{21},\ldots,x_{2i},\ldots,x_{2n})$, weighted Euclidean distance can be given by

  $$D(X_1, X_2) = \sqrt{\sum \mathrm{FR}(x_i) \times (x_{1i} - y_{1i})^2}$$

- For each positive sample, sort negative samples in ascending order of distance from the positive sample.
- For each positive sample, select user-defined number of negative samples. The user-defined number indicates the desired ratio of negative samples to positive samples. At this stage, special care is taken to avoid repetitive selection of negative samples. If a particular negative sample has been already selected, the next available negative sample is selected.

Experimental design

We employed a two-loop cross-validation strategy (Statnikov et al. 2005) to obtain reliable estimates of performance and to avoid overfitting. The first or inner loop was used for model selection (i.e., to determine the best parameter values for the classifier). The second or outer loop was used to estimate the performance of the classifier constructed in the inner loop. We employed a stratified tenfold cross-validation in the outer loop and a stratified fivefold cross-validation in the inner loop. For the weighted-SVM, the costs of misclassification for two classes C_+ and C_- were defined as $C_+ = C{*}w_+$ and $C_- = C{*}w_-$, where $w_+ = n_+ + n_-/n_+$ and $w_- = n_+ + n_-/n_-$. n_+ and n_- represent the number of training samples in the positive and negative classes, respectively. LibSVM (Chang and Lin 2001) tool was used to generate a basic SVM classifier using an RBF kernel with two model parameters: cost C and kernel parameter γ. We have used the following four measures to evaluate the performance of different methods:

$$\begin{aligned}&\text{Overall Accuracy} := \frac{\text{TP} + \text{TN}}{\text{TP + FP + TN + FN}}\\&\text{Sensitivity (TPR)} := \frac{\text{TP}}{\text{TP + FN}}\\&\text{Specifity (TNR)} := \frac{\text{TN}}{\text{TN + FP}}\\&\text{Gmean (balanced accuracy)} := \sqrt{\text{Sensitivity} \times \text{Specifity}}\end{aligned}$$

Overall accuracy is not a preferred performance measure for imbalanced datasets. When working with a high imbalance, a naive classifier classifying everything as a majority class sample will result in a high predictive accuracy. Sensitivity and specificity are two very common measures used in medical community and increasingly in machine learning. Kubat et al. (1997) combined these two measures and suggested Gmean or balanced accuracy. Recently, a new performance measure, called the "adjusted geometric mean" (AGm) was also proposed (Batuwita and Palade 2009b). In this paper, Gmean (Kubat et al. 1997) was used for model selection to identify the best model parameters. The whole approach can be summarized as follows:

- Step 1 Split the data in a stratified manner into *K* parts. These *K* partitions are for the outer loop
- Step 2 For each fold *i*,
    - Step 2.1 Consider the *i*th part as test data and the remaining (*K-1*) parts collectively as training data
    - Step 2.2 Calculate the weighted Euclidean distance of each negative sample from all positive samples in the training data
    - Step 2.3 Select negative samples as presented in "Our Proposed Undersampling Approach."
    - Step 2.4 Combine selected negative samples and positive samples in the training data to obtain the modified training data
    - Step 2.5 Break the modified training data into *L* parts. These *L* partitions are for the inner loop
    - Step 2.6 Get the best model parameters giving the best average Gmean based on the *L*-fold experiment
- Step 3 Build classifier model using modified training data obtained from (*K-1*) parts and the selected best model parameters
- Step 4 Get the performance measure on the held out *i*th test data
- Step 5 Repeat the steps from 2 to 4 for all partitions
- Step 6 Calculate average performance measure over *K*-folds

Results and discussion

We analyzed the performance of the proposed method on four datasets. As discussed in the previous section, we have evaluated the performance in terms of average overall accuracies, average true positive rates (sensitivity), average true negative rates (specificity) and average Gmean. The average in all cases was calculated using all 10 cross-validations. It is important to note that the test data has a similar proportion of positive and negative samples as the original data and is not balanced as the training data by undersampling. In real-world scenarios, we expect that there will be more negative class samples than samples from the positive class in such imbalanced data. We did not attempt to make the test dataset balanced for the purpose of realizing utility of the proposed approach.

The percentage of samples removed by the proposed undersampling varied for different datasets. It also varied within different folds of the same dataset. For micropred, the ratio of negative and positive samples used for making final model was 2/3 (ratio in complete dataset ∼14). Similarly, the imbalance ratio for Xwchen, active-site and cysteine datasets were 2 (∼9), 7/8 (∼97) and 2/5 (∼9), respectively. Here, in the case of micropred and active-site datasets, the two numbers were found to be best during model selection via inner CV (as described in the experimental design) while a single optimal imbalance ratio was found for other two datasets.

We have compared our results to the best available results for publicly obtained datasets. However, it is noteworthy to mention that for some of these methods there were strong differences in experimental design.

Table 1 shows the four performance measures over all datasets. We can observe that the proposed undersampling approach with weighted-SVM has a very consistent performance compared to the weighted-SVM approach using all of the data. Weighted-SVM using all of the data was affected by the imbalanced nature of data. This can be inferred when looking at the difference in sensitivity and specificity. The proposed undersampling approach led to the balanced sensitivity and specificity in most of the cases considered. We also observe that even Gmean is affected by the comparatively high specificity with respect to sensitivity. The results for the datasets Active-site and Xwchen are perfect examples showing the failure of the weighted-SVM without our proposed undersampling procedure. Hence, it is necessary to compare the results of different methods by looking at the three measures namely, Gmean, sensitivity, and specificity.

We can observe that for *Micropred* dataset, the proposed approach led to both better Gmean and sensitivity. It has improved the Gmean and sensitivity by ∼6 and ∼10%, respectively compared to the prior best result (Batuwita and Palade 2009a). It is noteworthy to mention that the best result in Batuwita and Palade (2009a) was obtained using fewer features. When using all 48 features, SE = 80.32%, SP = 98.71% and GM = 89.04% were reported. For the Xwchen-data (Chen and Jeong 2009), weighted-SVM using all of the data provided the best specificity but at the cost of low sensitivity and Gmean. The proposed approach provided higher Gmean and sensitivity. For the Active-site and Cysteine data, the proposed approach led to improved performance in terms of sensitivity and Gmean.

Conclusion

In this paper, we have presented a new deterministic undersampling technique, and evaluated its performance

Table 1 Comparison of different methods based on the four performance measures

Dataset	Method	Overall Acc.	Gmean	Sensitivity	Specificity
Micropred (Batuwita and Palade 2009a)	Wt-SVM	99.06	93.43	87.39	99.93
	Proposed method	99.83	**99.03**	**98.12**	99.96
	Previous best result	–	93.58	90.02	97.28
	SMOTE	99.34	98.36	97.25	99.49
Xwchen (Chen and Jeong 2009)	Wt-SVM	91.57	73.08	55.87	95.56
	Proposed method	77.53	**74.54**	71.04	78.27
	Previous best result	71.90	71.59	**71.20**	71.98
	SMOTE	92.96	71.07	51.74	97.69
Active-site	Wt-SVM	91.82	76.65	63.78	92.11
	Proposed method	80.33	**78.72**	**76.45**	81.06
	SMOTE	94.77	75.74	60.34	95.13
Cysteine	Wt-SVM	50.04	59.37	75.09	47.21
	Proposed method	70.74	**72.96**	**75.91**	70.15
	SMOTE	85.56	60.56	40.51	90.66

Best results for each dataset are shown in bold

over four datasets. We have compared the proposed algorithm with the weighted-SVM on all datasets and also with previously available best results for two of the four datasets where comparisons were appropriate to methods used in the public domain. We observed that the proposed technique either led to better or at least competitive Gmean and sensitivity with respect to weighted-SVM. Among the two SVM-based approaches, the proposed method was more stable in terms of sensitivity and specificity as for some datasets, the weighted-SVM approach led to higher specificity compared to low sensitivity and improved Gmean value. Our results indicate that using a weighted-SVM is a useful option for unbalanced datasets. An undersampling approach with weighted-SVM is a better option for classification using imbalanced data as the training a model based on all of the data given computational requirements for the latter approach.

Acknowledgments The authors acknowledge financial support offered by the Agency for Science, Technology, and Research, Singapore (A*Star) under grant #052 101 0020.

References

Akbani R, Kwek S, Japkowicz N (2004) Applying support vector machines to imbalanced datasets. Lect Notes Comput Sci 3201:39–50

Batuwita R, Palade V (2009a) microPred: effective classification of pre-miRNAs for human miRNA gene prediction. Bioinformatics 25:989–995

Batuwita R, Palade V (2009b) AGm: a new performance measure for class imbalance learning. Application to bioinformatics problems. In: Proceedings of 8th international conference on machine learning and applications, ICMLA 2009, 13–15 December 2009, Miami Beach, USA

Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The protein data bank. Nucl Acids Res 28:235–242

Chang CC, Lin CJ (2001) LIBSVM: a library for support vector machines, 2001, Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm

Chawla NV, Japkowicz N, Kotcz A (2004) Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explor Newsl 6:1–6

Chen X, Jeong JC (2009) Sequence-based prediction of protein interaction sites with an integrative method. Bioinformatics 25:585–591

Chen J, Liu H, Yang J, Chou KC (2007) Prediction of linear B-cell epitopes using amino acid pair antigenicity scale. Amino Acids 33(3):423–428

Cortes C (1995) Prediction of generalization ability in learning machines. University of Rochester, Rochester

Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3:1289–1305

Joachims T, Nedellec C, Rouveirol C (1998) Text categorization with support vector machines: learning with many relevant features. In: Machine learning: ECML-98. Springer, Berlin

Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M (2008) AAindex: amino acid index database, progress report 2008. Nucleic Acids Res 36:D202–D205

Kubat M, Holte R, Matwin S (1997) Learning when negative examples abound. In: Proceedings of the 9th European conference on Machine Learning. LNCS, vol 1224. Springer, London, pp 146–153

Li W, Godzik A (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22:1658–1659

Liu XY, Wu J, Zhou ZH (2009) Exploratory Undersampling for Class-Imbalance Learning. IEEE Trans Syst Man Cybern B 39:539–550

Mazurowski MA, Habas PA, Zurada JM, Lo JY, Baker JA, Tourassi GD (2008) Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. Neural Netw 21:427–436

McGuffin LJ, Bryson K, Jones DT (2000) The PSIPRED protein structure prediction server. Bioinformatics 16:404–405

Mizuguchi K, Deane CM, Blundell TL, Johnson MS, Overington JP (1998) JOY: protein sequence-structure representation and analysis. Bioinformatics 14:617–623

Mladenic D, Grobelnik M (1999) Feature selection for unbalanced class distribution and naive bayes. In: Proceedings of the Sixteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, pp 258–267

Nishikawa K, Ooi T (1986) Radial locations of amino acid residues in a globular protein: correlation with the sequence. J Biochem 100:1043–1047

Osuna E, Freund R, Girosit F (1997) Training support vector machines: an application to face detection. In: 1997 IEEE computer society conference on computer vision and pattern recognition, 1997, pp 130–136

Porter CT, Bartlett GJ, Thornton JM (2004) The catalytic site atlas: a resource of catalytic sites and residues identified in enzymes using structural data. Nucleic Acids Res 32:D129

Pugalenthi G, Kumar KK, Suganthan PN, Gangal R (2008) Identification of catalytic residues from protein structure using support vector machine with sequence and structural features. Biochem Biophys Res Commun 367:630–634

Robinson M, Sharabi O, Sun Y, Adams R, Boekhorst R, Rust AG, Davey N (2007) Using real-valued meta classifiers to integrate and contextualize binding site predictions. Lect Notes Comput Sci 4431:822–829

Sales AP, Tomaras GD, Kepler TB (2008) Improving peptide-MHC class I binding prediction for unbalanced datasets. BMC Bioinform 9:385

Shi MG, Xia JF, Li XL, Huang DS (2009) Predicting protein–protein interactions from sequence using correlation coefficient and high-quality interaction dataset. Amino Acids

Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, Levy S (2005) A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics 21:631–643

Sun XD, Huang RB (2006) Prediction of protein structural classes using support vector machines. Amino Acids 30:469–475

Tang Y, Zhang YQ, Chawla NV, Krasser S (2009) SVMs modeling for highly imbalanced classification. IEEE Trans Syst Man Cybern B 39:281–288

Vapnik V (1998) Statistical learning theory. Wiley, New York

Verma R, Varshney GC, Raghava GP (2009) Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile. Amino Acids

Veropoulos K, Campbell C, Cristianini N (1999) Controlling the sensitivity of support vector machines. In: Proceedings of the sixteenth international joint conference on artificial intelligence (IJCAI99)

Wang M, Yang J, Chou KC (2005) Using string kernel to predict signal peptide cleavage site based on subsite coupling model. Amino Acids 28(4):395–402

Wang Y, Xue Z, Shen G, Xu J (2008) PRINTR: prediction of RNA binding sites in proteins using SVM and profiles. Amino Acids 35(2):295–302

Wu G, Chang EY (2003) Class-boundary alignment for imbalanced dataset learning. In: ICML 2003 workshop on learning from imbalanced data sets II. Washington, DC

Wu J, Liu H, Duan X, Ding Y, Wu H, Bai Y, Sun X (2009) Prediction of DNA-binding residues in proteins from amino acid sequences using a random forest model with a hybrid feature. Bioinformatics 25:30–35

Yang ZR (2004) Biological applications of support vector machines. Briefings Bioinform 5:328–338

Yousef M, Nebozhyn M, Shatkay H, Kanterakis S, Showe LC, Showe MK (2006) Combining multi-species genomic data for microRNA identification using a Naive Bayes classifier. Bioinformatics 22:1325–1334

Zhang J, Bloedorn E, Rosen L, Venese D, Inc AOL, Dulles VA (2004) Learning rules from highly unbalanced data sets. In: Fourth IEEE international conference on data mining, 2004. ICDM'04, pp 571–574