Computational Biology and Chemistry 36 (2012) 36–41

Contents lists available at SciVerse ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/compbiolchem

Research article

Using ensemble methods to deal with imbalanced data in predicting protein–protein interactions

Yongqing Zhang[a], Danling Zhang[a], Gang Mi[b], Daichuan Ma[c], Gongbing Li[a], Yanzhi Guo[c], Menglong Li[c,**], Min Zhu[a,*]

[a] College of Computer Science, Sichuan University, Chengdu 610065, PR China
[b] School of Life Science, Sichuan University, Chengdu 610064, PR China
[c] College of Chemistry, Sichuan University, Chengdu 610064, PR China

ARTICLE INFO

Article history:
Received 3 October 2011
Received in revised form 2 December 2011
Accepted 21 December 2011

Keywords:
Protein–protein interaction
Ensemble methods
Imbalanced data

ABSTRACT

In proteins, the number of interacting pairs is usually much smaller than the number of non-interacting ones. So the imbalanced data problem will arise in the field of protein–protein interactions (PPIs) prediction. In this article, we introduce two ensemble methods to solve the imbalanced data problem. These ensemble methods combine the based-cluster under-sampling technique and the fusion classifiers. And then we evaluate the ensemble methods using a dataset from Database of Interacting Proteins (DIP) with 10-fold cross validation. All the prediction models achieve area under the receiver operating characteristic curve (AUC) value about 95%. Our results show that the ensemble classifiers are quite effective in predicting PPIs; we also gain some valuable conclusions on the performance of ensemble methods for PPIs in imbalanced data. The prediction software and all dataset employed in the work can be obtained for free at http://cic.scu.edu.cn/bioinformatics/Ensemble_PPIs/index.html.

1. Introduction

Protein–protein interactions (PPIs) play an important role in many biological processes. In recent years, various experimental approaches have been developed for the large-scale PPIs analysis, including yeast-two-hybrid (Bader and Hogue, 2002; Ito et al., 2001; Uetz et al., 2000), mass spectroscopy methods (Gavin et al., 2006; Krogan et al., 2006) and so on. However, experimental methods cannot catch up with the growth of the newly found proteins since they are expensive, time-consuming and labor-intensive. Therefore, many computational approaches have been developed to explore the protein interactions. Some methods are based on the genomic information, such as gene neighborhood (Enright et al., 1999; Overbeek et al., 1998), gene fusion (Enright et al., 1999) and phylogenetic profiles (Han et al., 2005). Several sequence-based methods (Grantham, 1974; Guo et al., 2008; Hopp and Woods, 1981; Horton et al., 2007; Li and Godzik, 2006; Xenarios et al., 2002) have been proposed to predict PPIs based on the assumption of 'the function of a protein is derived from its structure, and its structure is specified by its amino acid sequence'. Most of these approaches have obtained satisfactory performance on dataset by comparing equal number of interacting and non-interacting protein pairs.

Generally, the proteins in the same subcellular localization are seen as positive samples whereas those in different subcellular localizations are seen as negative samples. However, the imbalanced problem will arise in PPIs because the number of positive dataset is usually much smaller than that of negative dataset. As a result, the imbalanced data tend to cause classifiers to over fit and to perform poorly on the minority class. To overcome the imbalanced problem in PPI classification, we propose two ensemble methods: integrating based-cluster under-sampling technology and the fusion classifiers. Firstly, in terms of the majority class, we could use the technology of the cluster-based under-sampling, which includes the calculation of distances from each cluster to the minority samples and remove the nearest cluster. Then we build an ensemble of support vector machine (SVM) and an ensemble of artificial neural network (ANN) for improved imbalanced classification by using the remainder majority samples and minority samples. Two ensemble methods were used to train classifiers. One method is bagging; the other is similar to Adaboost. Promising results from

Abbreviations: PPIs, protein–protein interactions; DIP, Database of Interacting Proteins; AC, auto covariance; SVM, support vector machine; ANN, artificial neural network; ROC, receiver operating characteristic; AUC, area under ROC curve.

* Corresponding author. Tel.: +86 28 85469305; fax: +86 28 85469305.
** Corresponding author. Tel.: +86 28 89005151; fax: +86 28 85412356.
E-mail addresses: liml@scu.edu.cn (M. Li), zhumin@scu.edu.cn (M. Zhu).

1476-9271/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2011.12.003

these experiments confirmed the effectiveness of our proposed method.

2. Materials and methods

2.1. Dataset

In this paper, the dataset is composed by positive dataset and negative dataset. The former refers to interacting protein pairs, while the latter refers to non-interacting protein pairs. The positive dataset is collected from Homo sapiens core subset of DIP (Xenarios et al., 2002), version DIP_20091230 which contains 1671 interaction pairs. The protein pairs that contain a protein with less than 50 amino acids are removed and then a non-redundant subset is generated at the sequence identity level of 40% by clustering analysis using the CD-HIT program (Li and Godzik, 2006). After these screening procedures, the total positive dataset is reduced to 863.

Since the non-interacting pairs are not readily available from the database, the negative dataset is generated as described in what follows. Firstly, protein pairs are randomly generated from all the human proteins. Secondly, the protein pairs, which also appear in positive dataset, are removed. Finally, we use Wolf PSORT (Horton et al., 2007) to predict the sub-cellular location of each protein. It is practical to delete the protein pairs with the same sub cellular location, while the remaining pairs could serve as the negative dataset, because the two proteins will not interact with each other due to the different sub cellular location. At last, we get a negative dataset containing 23,676 protein pairs, and the ratio of positive data and negative data is about 1:27.

2.2. Feature extraction

To predict PPIs by using sequence information, one of the main computational challenges is to find a suitable encoding of the protein information in some vector space. In our previous study (Guo et al., 2008), we have shown that the auto covariance (AC) is an useful feature encoding for prediction. Seven physicochemical properties of amino acids were selected to translate amino acid residues of protein–protein sequence, including hydrophilic (Hopp and Woods, 1981), polarity (Grantham, 1974), hydrophobicity (Tanford, 1962), volumes of side chains of amino acids (Krigbaum and Komoriya, 1979), polarizability (Charton and Charton, 1982), solvent accessible surface area (Rose et al., 1985) and net charge index of side chains of amino acids (Zhou et al., 2006). Then each protein sequence was translated into seven vectors with each amino acid represented by the values of seven descriptors. Then the translated sequences were further processed by AC, which calculated the covariance between amino acids with a certain distance apart, and *lag* is the distance between one residue and its neighbor. The AC variables are calculated according to Eq. (1), where *j* represents one descriptor, *i* the position in the sequence *X*, *n* the length of the sequence *X* and *lag* the value of the lag. The detailed description can refer to our previous publication (Guo et al., 2008). In this work, a protein pair is converted into a 420-dimensional ($2 \times 30 \times 7$) vector by AC with *lg* (the maximum distance between an amino acid residue and its neighbor a certain number of residues away) of 30 amino acids, where 2 is the number of two protein sequence and 7 is the number of descriptors:

$$AC_{lag,j} = \frac{1}{n\text{-}lag}\sum_{i=1}^{n\text{-}lag}(X_{i,j} - \frac{1}{n}\sum_{i=1}^{n}X_{i,j}) \times (X_{(i+lag),j} - \frac{1}{n}\sum_{i=1}^{n}X_{i,j}) \quad (1)$$

2.3. Algorithms for classification

2.3.1. Support vector machine

The SVM classifier, motivated by results of statistical learning theory, is one of the most effective machine learning algorithms for many complex binary classification problems. For further details we refer to Vapnik (1998) and Vladimir and Vapnik (1995). They act as linear classifiers in a high dimensional feature space originated by a projection of the original input space. The resulting classifier is in general non-linear in the input space and it achieves good generalization performances maximizing the margin between the classes.

The SVM can find a separating hyperplane in the feature space and categorize points in that space without representing the feature space explicitly: it can be revealed that under some conditions, the SVM can work in the feature space using only data in the input space, avoiding the computational cost of explicitly representing the feature vectors in an high dimensional feature space. Given a training set of *n* data points $\{x_i, y_i\}$, $i = 1, 2, \ldots, n$, *n* is the total number of samples, $y_i = \{1, -1\}$, where x_i is the *i*th input pattern and y_i is the *i*th output pattern, the support vector method approach aims at constructing a classifier of the form:

$$F(x) = sign(\sum_{i=1}^{n} a_i y_i K(x_i, x) + b) \quad (2)$$

where the (x_i, y_i) are pairs of labeled data of the training set, $K(x_i,x)$ is a kernel function, a_i is the Lagrangian parameter, and *b* is the threshold both learned by the SVM algorithm. The radial basis function (RBF) was selected as the kernel function and defined as follows: $K(x_i, x) = \exp(-(x - x_i)^2/\sigma^2)$.

2.3.2. Artificial neural network

The ANN ensemble was raised by Hansen and Salamon (1990). The purpose is to enhance the generalization capability by training multiple neural networks and fusing them. In this work, Back-Propagation network (BP) is used for classification. The BP is a feed forward network with one or more layer nodes and it is viewed as a generalization of least mean square algorithm. It is an iterative gradient search technique designed to minimize the mean square error between the actual and desired net outputs. A three-layer network was proven to be capable of emerging arbitrarily complex decision regions and computing any continuous likelihood function required in a classifier (Lippmann, 1987). In this section, the BP network with three-layer is used to train based classifier.

2.3.3. Ensemble method

In this work, we take into account ensemble methods to improve the performance of the overall system. The effectiveness of ensemble methods is highly reliant on the independence of the error committed by the base learner. The performance of ensemble methods strongly depends on the accuracy and the diversity of the base learner. The easiest approach to generate diverse base classifier is by manipulating the training data. In this part, we investigated bagging ensemble techniques, and the way to combine different learning algorithms is by majority voting.

Bagging (bootstrap aggregating) was introduced by Breiman (1996) and voted classifiers generated by different bootstrap samples. A bootstrap sample is generated by random sampling replacement from the training dataset. The final classifier $H(x)$ is constructed by aggregating $H_i(x)$. The bagging algorithm is shown:

Input: Training sets $T=(x_i, y_i)$, $i=1$ to n; Integer n (iteration number);
Output: Classifier $H(x)$;

1. For each iteration $i=1$ to n {
2. Select a subset T_i of size N form the original training examples T, the size of T_i is the same with the T where some instances may not appear in T_i while other appear more than ones;
3. Generate a classifier $H_i(x)$ from the T_i;
   }
4. The final classifier $H(x)$ is formed by aggregating the n classifiers to classify an instance x, a vote for class y is recorded by every classifier $H_i(x)=y$;
5. $H(x)$ is the class with the majority voting.

2.3.4. The imbalanced data problem

Imbalanced data problem often appears in many practical classification events (Das and Sengur, 2010; Li et al., 2010; Liu et al., 1999). A number of solutions for the imbalanced problem were proposed both at the data and algorithmic levels (Cohen et al., 2006). At the data level, these solutions include over-sampling and under-sampling. Over-sampling method increases the number of minority class samples to decrease the degree of imbalanced distribution. The over-sampling technique includes random technique (Wu and Chang, 2003), synthetic minority over-sampling technique (SMOTE) (Chawla et al., 2002) and cluster-based method (Jo and Japkowicz, 2004). The under-sampling method includes random under-sampling, condensed nearest neighbor rule (Hart, 1968), Tomek links (Tomek, 1976), One Sided Selection (Kubat and Matwin, 1997), cluster-based under-sampling (Yen and Lee, 2009), etc. At the algorithmic level, solutions include decision threshold tuning (Huang et al., 2004), cost-sensitive learning (Elkan, 2001), one-class classifiers (Raskutti and Kowalczyk, 2004), hyperplane re-adjustment (Chen et al., 2005), classifier ensembles (Breiman, 1996; Kang and Cho, 2006; Shen and Chou, 2007), etc. So, in this article, we propose a new hybrid sampling algorithm, which integrates both under-sampling techniques and ensemble classifier, to overcome the imbalanced problem in PPIs classification.

2.3.5. Proposed method

In this study, our method is to combine ensemble learning algorithm with under-sampling techniques. This combination will allow us to lessen the imbalance ratio and, therefore, make the learning task more tractable. In this method, the majority class is first under-sampled by K-means clustering algorithm. After the process of decomposition, we calculate the distances from the disjunctive clusters to the minority class, then remove the cluster with nearest distances. Then the remaining clusters majority class is resampled by bootstrap technique. It performs random sampling with replacement only on the majority class so that its size is equal to the number of minority class, and keeps the whole minority samples in all subsets. After the sampling procedure, we get N new dataset from the majority class. Each of the new dataset and minority class are combined into N new training sets. With the newly generated dataset, we train N classifiers with one classifier corresponding to one training set. Each of these classifiers is a SVM and ANN. In this section, we use two ensemble methods to train dataset. The first one is Bagging, named Ensemble1. In this method, each classifier trained separately. The second one is similar to Adaboost, named Ensemble2. In this method, the dataset of latter classifier depends on the training result of front classifier. If the training of a sample goes wrong, it will be picked out and be trained again in the latter classifier. Finally, a simple majority voting method is used in the fusion unit, and the final result is determined by majority votes among the outputs of the N classifiers for further processing with 10-fold cross-validation. Fig. 1 displays the overview of Ensembe1.

2.3.6. Evaluation measures

The performance of the proposed ensemble methods is measured using 10-fold cross-validation. Each dataset is randomly divided into ten subsets with an approximately equal number. Each classifier is trained and tested ten times with one dataset.

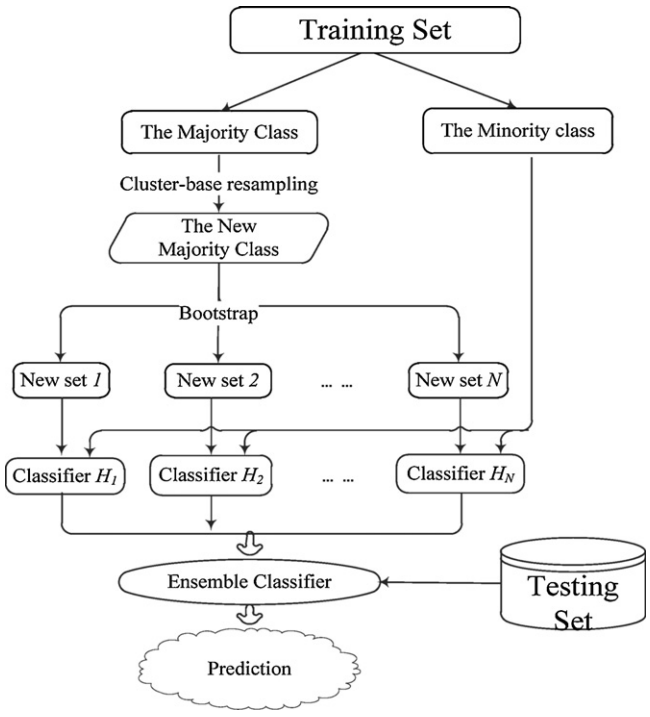

Fig. 1. The overview of the Ensemble1.

Some widely used measures in bioinformatics are employed in this study, such as *Accuracy* (Acc), *Sensitivity* (Sen), *Specificity* (Spec), *F-measure* (Fm) and *AUC* (area under ROC curve) score. These measures are defined as follows in Table 1. The definitions of the abbreviations used: *TP* is the number of true positive; *FP* is the number of false positive; *TN* is the number of true negative; and *FN* is the number of false negative. The *AUC* score is the normalized area under the ROC curve, the larger, and the better. The ROC curve shows the trade-off between *sensitivity* and *specificity* and gives a complete evaluation of the computational method.

3. Results and discussion

3.1. Comparison between ensemble classifier and single classifier

In our experiments, SVM and ANN are used for base classifiers. Here the Libsvm package 2.91 (http://www.csie.ntu.edu.tw/~cjlin/libsvm/) is used with radial basis function (RBF) as the kernel function. The regularization parameter c and the kernel width parameter g are 2.0 and 0.03125, respectively. In this section, the BP neural network is implemented with MATLAB. By testing, we find that the neuron number of the hidden layer is 25, so the network is of 420-25-2 structure. Besides, the S-tangent function *transig* and the linear function *purelin* are respectively adopted as the neuron transmitting function of the hidden layer and output layer. The number of training epochs is 50. The ensemble size is

Table 1
Evaluation measurements.

Measurement	Abbreviation	Equation
Accuracy	Acc	$(TP+TN)/(TP+TN+FP+FN)$
Sensitivity	Sen	$TP/(TP+FN)$
Specificity	Spec	$TN/(TN+FP)$
F-measure	Fm	$2\,TP/(2\,TP+FP+FN)$

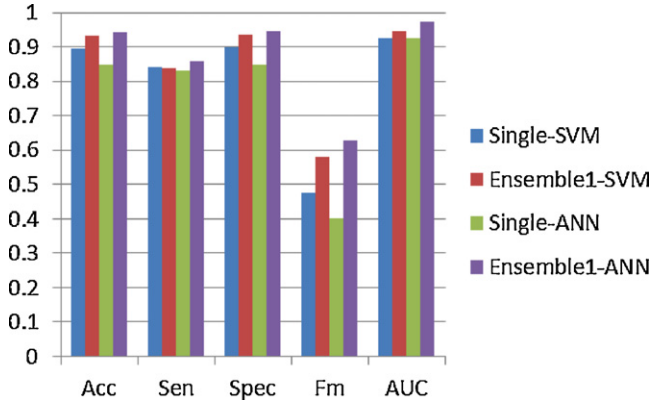

Fig. 2. The comparison between performances of the single based classifier and the Ensemble1 classifier.

taken as 30. It is worth pointing out that all of the classifiers use the same parameters.

To demonstrate the performance, we compare the performance of the Ensemble1 classifier with the base classifiers predicting PPIs for imbalanced data. Fig. 2 reports the predicted results of Ensemble1 classifier and the based classifier. In Fig. 2 'Single' means an individual base classifier. The result of Ensemble1-SVM is reported to achieve 93.32% *Accuracy*, 83.95% *Sensitivity*, 93.69% *Specificity*, 58.21% *F-measure* and 0.9447 *AUC*. While the result of Ensemble1-ANN is reported to achieve 94.37% *Accuracy*, 85.7% *Sensitivity*, 94.68% *Specificity*, 62.71% *F-measure* and 0.9733 *AUC*. From the chart, we obtain the following conclusions about the performance of the ensemble methods. On the one hand, the ensemble method generates more successful results than single classifiers. It can be seen from Fig. 2 that the performances of ensemble method are better than any other based classifier, especially the *F-measure* value which increase remarkably. The comparison results confirm that Ensemble1 with cluster-based under-sampling technique can effectively deal with imbalanced data and obviously improve prediction performance. On the other hand, with the unstable based classifier of the ANN, the ensemble method brings significant improvements compared with ensemble of SVM.

Moreover, we analyzed the ROC curves in Fig. 3. And then we found out that the ROC curve of Ensemble1 method is way better than the ROC curve of Single classifier. This clearly proves that the ensemble methods are significantly better than base classifier.

Table 2
The results of the Ensemble1 and the Ensemble2.

	Acc (%)	Sen (%)	Spec (%)	Fm (%)	AUC
Ensemble1-SVM	93.32	83.95	93.69	58.21	0.9447
Ensemble2-SVM	**93.90**	**83.95**	**94.51**	**60.99**	**0.9465**
Ensemble1-ANN	94.37	85.70	94.68	62.71	0.9733
Ensemble2-ANN	**96.90**	**79.53**	**97.94**	**74.44**	**0.9702**

The bold values mean the difference between two lines and highlight the Ensemble2.

3.2. Comparison with ensemble methods

In addition, we investigated and compared the effectiveness of our proposed methods, Ensemble1 and Ensemble2. It can be seen from Table 2 that both Ensemble1-SVM and Esemble2-SVM achieved good results, though the improvement in the performance of Esemble2-SVM was slight. It is shown that data disturbance has little influence on the performance of ensemble SVM method; we conclude that the SVM is a stable machine learning method. While on ensemble ANN method, Ensemble1-ANN and Ensemble2-ANN have some distinction, especially for *F-measure* value. The performance of the Ensemble2-ANN outperforms the Ensemble1-ANN, more than 20% increase in *F-measure* value. The results imply that the second method is better for the overall prediction performance of classifier, also demonstrate the ANN is more sensitive for data slightly disturbance. Table 2 shows that ensemble SVM and ensemble ANN have some divergence for the same data classification, so we should consider this problem when we choose the classifier.

3.3. Web server

The interaction prediction server, Ensemble_PPIs, can be freely available to any researcher. A user can visit Ensemble_PPIs at http://cic.scu.edu.cn/bioinformatics/Ensemble_PPIs/index.html. A prediction page of Ensemble_PPIs is shown in Fig. 4. On the Ensemble_PPIs web page, users should paste the protein sequence in FASTA format and choose the prediction method. The web server returns the predicted interacting pairs and non-interacting pairs along the input sequence.

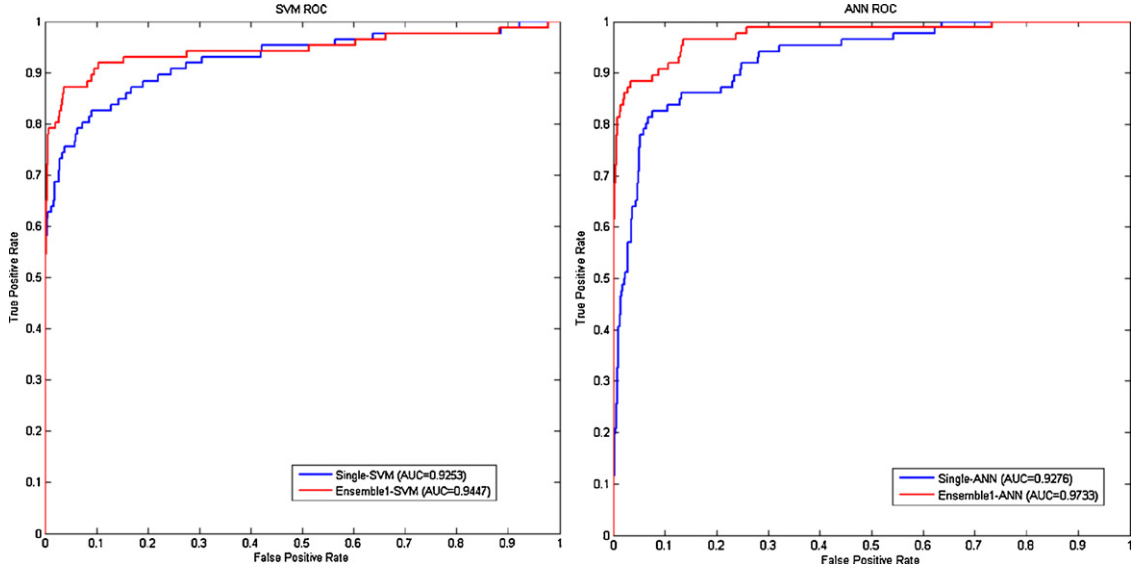

Fig. 3. The left figure is the roc analysis of single-SVM and Ensemble1-SVM. The right figure is the roc analysis of single-ANN and Ensemble1-ANN.

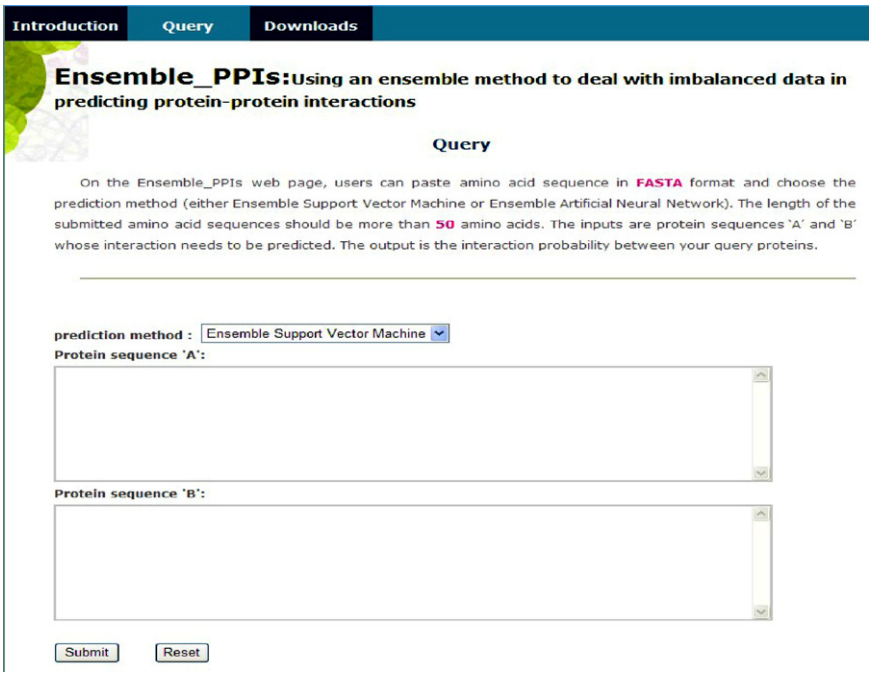

Fig. 4. The prediction page of Ensemble_PPIs.

3.4. Conclusions

In this work, we have shown two ensemble methods using cluster-based under-sampling technique to handle the imbalanced problem and fusion classifiers to increase the accuracy of classification on protein–protein interaction. We evaluate the ensemble classifiers and compare them on the dataset of DIP with 10-fold cross-validation. The prediction results clearly show that our methods are quite effective in PPIs. What is more, the ensemble method with different types of based classifiers is discussed to improve the accuracy in our work. The PPIs predictor is available on a public server (http://cic.scu.edu.cn/bioinformatics/Ensemble_PPIs/index.html).

Acknowledgments

The authors would like to express their appreciation of Dr. Yizhou Li's assistance and to thank the anonymous reviewers for their constructive suggestions to improve the work.

This work was supported by the National Natural Science Foundation of China (Nos. 20905054, 20972103) and the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20090181120058).

References

Bader, G.D., Hogue, C.W.V., 2002. Analyzing yeast protein–protein interaction data obtained from different sources. Nature Biotechnology 20, 991–997.
Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.
Charton, M., Charton, B.I., 1982. The structural dependence of amino acid hydrophobicity parameters. Journal of Theoretical Biology 99, 629–644.
Chawla, N.V., et al., 2002. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357.
Chen, X., Gerlach, B., Casasent, D., 2005. Pruning support vectors for imbalanced data classification. Proceedings of the International Joint Conference on Neural Networks, 1883–1888.
Cohen, G., et al., 2006. Learning from imbalanced data in surveillance of nosocomial infection. Artificial Intelligence in Medicine 37, 7–18.
Das, R., Sengur, A., 2010. Evaluation of ensemble methods for diagnosing of valvular heart disease. Expert Systems with Applications 37, 5110–5115.
Elkan, C., 2001. The foundations of cost-sensitive learning. Citeseer, 973–978.
Enright, A.J., et al., 1999. Protein interaction maps for complete genomes based on gene fusion events. Nature 402, 86–90.
Gavin, A.C., et al., 2006. Proteome survey reveals modularity of the yeast cell machinery. Nature 440, 631–636.
Grantham, R., 1974. Amino acid difference formula to help explain protein evolution. Science 185, 862.
Guo, Y., et al., 2008. Using support vector machine combined with auto covariance to predict protein Cprotein interactions from protein sequences. Nucleic Acids Research 36, 3025.
Han, J.D.J., et al., 2005. Effect of sampling on topology predictions of protein–protein interaction networks. Nature Biotechnology 23, 839–844.
Hansen, L.K., Salamon, P., 1990. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 993–1001.
Hart, P., 1968. The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14, 515–516.
Hopp, T.P., Woods, K.R., 1981. Prediction of protein antigenic determinants from amino acid sequences. Proceedings of the National Academy of Sciences of the United States of America 78, 3824.
Horton, P., et al., 2007. WoLF PSORT: protein localization predictor. Nucleic Acids Research 35, W585.
Huang, K., et al., 2004. Learning classifiers from imbalanced data based on biased minimax probability machine. Computer Vision and Pattern Recognition 2, 558–563.
Ito, T., et al., 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proceedings of the National Academy of Sciences of the United States of America 98, 4569.
Jo, T., Japkowicz, N., 2004. Class imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter 6, 40–49.
Kang, P., Cho, S., 2006. EUS SVMs: Ensemble of Under-sampled SVMs for Data Imbalance Problems. Springer, pp. 837–846.
Krigbaum, W., Komoriya, A., 1979. Local interactions as a structure determinant for protein molecules: II. Biochimica et Biophysica Acta (BBA): Protein Structure 576, 204–228.
Krogan, N.J., et al., 2006. Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. Nature 440, 637–643.
Kubat, M., Matwin, S., 1997. Addressing the Curse of Imbalanced Training Sets: One-sided Selection. Citeseer, pp. 179–186.
Li, W., Godzik, A., 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22, 1658.
Li, W., Miao, D., Wang, W., 2010. Two-level hierarchical combination method for text classification. Expert Systems with Applications 38, 2030–2039.
Lippmann, R.P., 1987. An introduction to computing with neural nets. Artificial Neural Network: Theoretical Concepts 209, 36–54.
Liu, J.J., et al., 1999. Imbalanced expression of functionally different WT1 isoforms may contribute to sporadic unilateral Wilms' tumor. Biochemical and Biophysical Research Communications 254, 197–199.
Overbeek, R., et al., 1998. Use of contiguity on the chromosome to predict functional coupling. In silico Biology 1, 93–108.
Raskutti, B., Kowalczyk, A., 2004. Extreme re-balancing for SVMs: a case study. ACM SIGKDD Explorations Newsletter 6, 60–69.

Rose, G.D., et al., 1985. Hydrophobicity of amino acid residues in globular proteins. Science 229, 834.
Shen, H.B., Chou, K.C., 2007. Hum-mPLoc: an ensemble classifier for large-scale human protein subcellular location prediction by incorporating samples with multiple sites. Biochemical and Biophysical Research Communications 355, 1006–1011.
Tanford, C., 1962. Contribution of hydrophobic interactions to the stability of the globular conformation of proteins. Journal of the American Chemical Society 84, 4240–4247.
Tomek, I., 1976. Two modifications of CNN. IEEE Transactions on Systems, Man and Cybernetics 6, 769–772.
Uetz, P., et al., 2000. A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. Nature 403, 623–627.
Vapnik, V., 1998. The support vector method of function estimation. In: Nonlinear Modeling: Advanced Black-Box Techniques, pp. 55–86.
Vladimir, V.N., Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer.
Wu, G., Chang, E.Y., 2003. Class-boundary Alignment for Imbalanced Dataset Learning. Citeseer, pp. 49-56.
Xenarios, I., et al., 2002. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Research 30, 303.
Yen, S.J., Lee, Y.S., 2009. Cluster-based under-sampling approaches for imbalanced data distributions. Expert Systems with Applications 36, 5718–5727.
Zhou, P., et al., 2006. Genetic algorithm-based virtual screening of combinative mode for peptide/protein. Huaxue Xuebao 64, 691–697.