Annual Review & Research in Biology
3(2): 92-106, 2013

SCIENCEDOMAIN *international*
www.sciencedomain.org

Improving the Prediction of Protein-Protein Interaction Sites Using a Novel Over-Sampling Approach and Predicted Shape Strings

Lan Anh T. Nguyen[1*], Osamu Hirose[2], Xuan Tho Dang[1], Tu Kien T. Le[1], Thammakorn Saethang[1], Vu Anh Tran[1], Mamoru Kubo[2], Yoichi Yamada[2] and Kenji Satou[2]

[1]Graduate School of Natural Science and Technology, Kanazawa University, Japan.
[2]Institute of Science and Engineering, Kanazawa University, Japan.

Authors' contributions

This work was carried out in collaboration between all authors. Authors LATN and KS defined the research question, designed, performed the experiments. Author LATN wrote the first draft of the manuscript and author OH substantially revised the draft. Authors YY and MK provided helpful comments. All authors read and approved the final manuscript.

Research Article

Received 31st December2012
Accepted 6th March 2013
Published 24th March 2013

ABSTRACT

Identification of protein-protein interaction (PPI) sites is one of the most challenging tasks in bioinformatics and many computational methods based on support vector machines have been developed. However, current methods often fail to predict PPI sites mainly because of the severe imbalance between the numbers of interface and non-interface residues. In this study, we propose a novel over-sampling method that relaxes the class-imbalance problem based on local density distributions. We applied the proposed method to a PPI dataset that includes 2,829 interface and 24,616 non-interface residues. The experimental result showed a significant improvement in predictive performance comparing with the other state-of-the-art methods according to the six evaluation measures.

Keywords: Protein-protein interaction sites; shape strings; class imbalance; over-sampling.

**Corresponding author: Email: lananh257@gmail.com;*

1. INTRODUCTION

Protein-protein interactions, known as physical contacts among proteins, are essential molecular processes for living organisms to maintain their lives. They play a central role in various biological functions such as regulation of metabolic and signaling pathways, DNA replication, protein synthesis, immunological recognition, and so forth. Especially, physical interface between two interacting proteins is a key to understand enzymatic activities of proteins. Therefore, one important task in bioinformatics is to develop computational methods to find binding interfaces between two interacting proteins accurately.

However, a naive approach based on support vector machines, one of the most standard classifiers, often fail to predict binding interfaces among interacting proteins with high specificity since the number of non-interaction residues is much larger than the number of interaction residues. This is so-called the class-imbalance problem. A dataset is imbalanced if the number of samples in some classes is significantly larger than in other classes. In the serious cases, the ratio of minority class to majority class can be as large as 1:100,000 [1]. Use of traditional machine learning techniques for these datasets often lead to undesirable results that only majority class is correctly predicted. This is a common problem in bioinformatics such as prediction and classification for miRNAs [2], beta-turns [3,4], microRNAs [5,6], breast cancer, lung cancer [7] and so on.

Many methods to deal with the class-imbalance problem have been developed. One important class of such methods is resampling-based techniques such as over-sampling and under-sampling methods, which have been reported to improve classification accuracy significantly [1].

In this study, we propose a novel oversampling approach in order to relax class-imbalance for the dataset of PPI sites. Instead of dealing with all minority class samples equivalently, we intentionally increase the number of minority samples according to their local distribution. Furthermore, predicted shape strings, which have been utilized in many researches in recent years [3,8,9,10], are used to enrich the feature groups. We present numerical experiments compared with state-of-the-art methods such as Anand et al. [6].

2. MATERIALS AND METHODS

2.1 Dataset

In this study, we used two datasets. The first one (that was named D1050) was the same with Chen and Jeong [11]. For predicting interface residues and non-interface residues, Chen and Jeong used the information of physicochemical features, evolutionary conservation score, amino acid distances, and position specific score matrix (PSSM) to extract features for 99 polypeptide chains of 54 hetero complexes [11]. By using a sliding window with size 21, the central residue of a partial peptide was assigned as interface residue if its relative solvent accessible surface area (RASA) was greater than 25% and the difference of accessible surface areas (ASAs) between its unbound state and bound state was greater than 1Å^2. As a result, each residue was represented as a 1,050 features. The dataset contained 2,829 interface residues (positive class) and 24,616 non-interface residues (negative class). The ratio of positive class samples to negative class samples was 1:8.7. That is, this dataset was highly imbalanced.

The second dataset (was named D1239) was prepared by adding information of predicted shape strings to the original dataset. The shape string of a protein is a sequence of symbols categorized according to the phi-psi torsion angles. There are eight shape symbols representing for eight categories (S,R,U,V,K,A,T,G). DSP program [8] was used to predict the shape strings. Each residue was predicted as one of these eight states or state N as the undefined phi-psi angle pair. Each sample in this dataset includes 1, 239 features.

2.2 Methods

2.2.1 Resampling techniques

As presented in [1], resampling techniques such as over-sampling methods, under-sampling methods, and under-over-sampling combination methods effectively improve classification accuracy for imbalanced datasets. Under-sampling methods balance the imbalanced dataset by removing samples in the majority class until the dataset becomes balanced. An important disadvantage of under-sampling methods is that this removal of majority samples leads to a significant information loss for the majority group. On the contrary, over-sampling methods increase the number of samples in the minority class. The synthetic samples are generated by various methods. The most naive technique is random over-sampling, which arbitrarily chooses some minority samples and replicates them (one or many times). One of the other common methods is SMOTE [12], which synthesizes the new samples locating between each minority class sample and its randomly chosen nearest neighbors. While random over-sampling techniques often lead to the over-fitting, SMOTE may result in the overlapping between classes [1]. Especially when the number of minority samples is small and they are distributed sparsely among the majority samples, the problem becomes more serious because most of the synthetic samples will be located among the majority class samples. Prati et al. [13] showed that the decrease in classification performance is caused by not only class-imbalance but also data-overlapping. Borderline-SMOTE [14] addresses this drawback by generating new samples for minority samples if they are located near the borderline, while the samples, which are surrounded by majority samples or have enough minority nearest neighbors are not considered. Though Borderline-SMOTE successfully improved predictive accuracy for imbalanced datasets, the overlapping problem is not carefully avoided.

In order to alleviate the problem of overlapping and over-fitting simultaneously, we propose a novel over-sampling algorithm, which we call Over-Sampling based on local Density (OSD). Instead of generating the same number of synthetic samples for each minority sample as SMOTE, OSD algorithm focuses on only minority samples located where the local density of minority samples is small in comparison with that of majority samples. As the local minority density is smaller, OSD increases the number of minority samples more strongly by synthesizing artificial minority samples. Here we define local density for each sample as follows:

Definition 1. Suppose m and n are the numbers of samples with the same and different class labels for sample x, respectively. Local density of x with radius r is the proportion m/(m+n).

2.2.2 OSD- a novel over-sampling approach

A key idea of the OSD algorithm is to increase the number of minority samples located where the local density of minority samples is small in comparison with majority samples.

For each minority class sample *x*, first of all, OSD finds neighbors of *x* and divides into two groups, majority and minority neighbors, according to their class labels (line 2). Note that the terms "majority" and "minority" are used in the global context. Here, neighbors of *x* are defined as samples in hyper-sphere with radius *r*. The number of synthetic samples for each *x* depends on its local distribution with parameter *d* (lines 6-9):

- If *x* doesn't have neighbor (i.e. $m + n = 0$), or local density of *x* is 0 (i.e. $m = 0$), *x* locates far from the other minority samples and OSD generates the maximum number of synthetic samples with the same class labels as *x* in order to avoid the class imbalance problem and diminish boundary variance derived from local sparsity, simultaneously. Hence, *d* new samples will be synthesized.
- If local density of *x* is greater than 0, $d*(1-m/(m+n))$ new synthetic samples are created.
- If sample *x* has no different class label neighbor, OSD does not adjust the local density of *x*.

Then, OSD generates the samples by function **New_sample_generation** (line 10). The synthesized samples are generated so that their distances to *x* are always less than *r_min* and they tend to be located closer to x as follows: (1) OSD randomly generates a number r' which follow the density $p(r) = cr^{-1/2}$(0<r<1) where $c=r\_min\ /k^{1/2}$ with k is the number of features. (2) adds it to the element of feature vector (lines 14-15). The pseudo-code for OSD algorithm is as follows:

OSD algorithm

Input: Minority dataset *M*; Majority dataset *N*; ratio of generation *d*; radius *r*.
Output: Set of synthetic samples.
Begin
1. For each $x \in M$
2. calculate the local minority neighbors *m* & local majority neighbors *n* for *x*;
3. calculate the distance *r_min* from *x* to its local majority nearest neighbor;
4. if (*r_min*>*r*)
5. *r_min* = *r*;
6. if (*m*+*n* = 0)
7. *number_of_new_samples* = *d*;
8. else
9. *number_of_new_samples* = $d*(1-m/(m+n))$;
10. **New_samples_generation**(*x*,*r_min*, *number_of_new_samples*);
11. End_for

End

Function New_sample_generation(x, r_min, d)
Input: Sample $x = (x_1,x_2,\ldots,x_k,class\_label)$; number of new samples *d*; radius *r_min.*
Output: Set of synthetic samples *new_samples_array* of *x*.

Begin
12. For *i* = 1:*d*
13. *new_sample_class_label* = *class_label*;
14. for *j* = 1:*k*
15. $new\_sample_j = x_j + r'$;

16. end_for
17. push(*new_samples_array*,*new_sample*);
18. end_for

End

2.2.3 KSVM-THR

We note that OSD generally does not balance imbalanced datasets entirely. To address this issue, we combine OSD and KSVM-THR, SVM with adjustment of the decision parameter, proposed by Lin and Chen [7]. The decision threshold θ of KSVM-THR is defined as

$$\theta = -1 + 2 * (p + \alpha)/(p + n + 2 * \alpha)$$

where p and n are the numbers of minority and majority class samples, respectively. The constant α is the tuning parameter and in the experiments below, it was optimized by grid search. If a data set is balanced, θ becomes zero. In this study, we utilize this technique to compose OSD-THR and RU-OSD-THR that combine KSVM-THR with OSD and RUS-OSD (Random Under-Sampling –OSD).

2.2.4 Experimental design

SVM with Gaussian RBF kernel was utilized to create a basic classifier. We conducted 10-fold cross validation. All the features of the datasets were normalized. Noise samples in the datasets were filtered out before over-sampling, where we defined samples that have the same feature vector and that belongs to different classes as noise samples. The overall predicting process is shown in Fig. 1. To determine the radius *r* for algorithm OSD, we calculated the distance between each pair of samples in the training set, sorted them in ascending order, saved in array *D*, set *k* = dim(*D*)*0.1% (*k* = dim(*D*)*0.01% for the D1239) where dim(*D*) was the size of *D* and assigned *r* as value of element k^{th} of *D*.

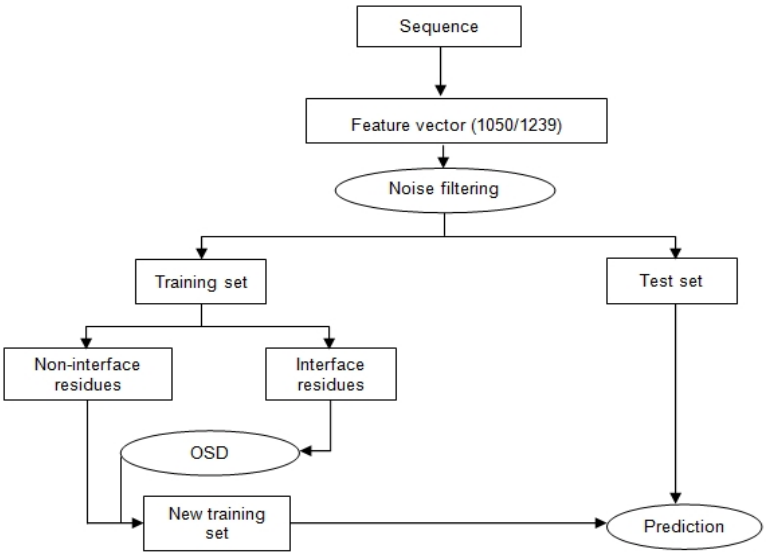

Fig. 1. Schematic representation of our method

Since the ratio of positive class to negative class of this dataset is 1:8.7, overall accuracy is not suitable for evaluating the performance of classifier. If the class-imbalance problem is severe, a naive approach that assigns all samples to the majority class makes overall accuracy high though no sample was assigned to the minority class [1]. Thus, as measures of performance evaluation, we use overall accuracy, sensitivity, specificity, G-mean and Matthews correlation coefficient, which are defined as follows:

Overall accuracy = $(TP + TN)/(TP + FN + TN + FP)$

Sensitivity =$TP/(TP + FN)$

Specificity =$TN/(TN + FP)$

G-mean (Balanced accuracy) = $(Sensitivity \times Specificity)^{1/2}$

Matthews correlation coefficient (*MCC*) = $\frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{((\mathrm{TP}+\mathrm{FN})\times(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN}))^{1/2}}$

Where TP and TN are the numbers of interface residues and non-interface residues that are correctly predicted; FP and FN are the numbers of non-interface residues and interface residues that are predicted as different from what they really are. Sensitivity and specificity have been commonly used in medical community [6]. G-mean is the combination of both sensitivity and specificity [15]. Matthews correlation coefficient measures how good the correlation of the predicted class labels and the actual class labels is. It lies in [-1,+1], where -1,1, and 0 represents the worst, the best and the random predictor, respectively.

3. RESULTS AND DISCUSSIONS

3.1 Evaluation on the D1050 Dataset

Using D1050 dataset, we evaluated the performance of OSD algorithm. It was compared with pure KSVM (KSVM without resampling), Random Under-Sampling (RUS), KSVM-THR, weighted SVM, SMOTE, the method of Chen and Jeong, and the under-sampling method introduced by Anand et al. [6]. The results of all these methods are shown in Table 1. In addition, Table 2 shows the results of experiments with the different decision thresholds of the methods.

Since non-interface residues approximately nine times outnumbered interface residues, KSVM could not perform well, whereas weighted-SVM, which assigns different costs of misclassification to minority and majority classes, could predict more positive samples than KSVM. Also, KSVM-THR achieved better performance by decreasing the decision threshold.

RUS removed many negative samples to balance the dataset (the new ratio of negative: positive samples was 1.1:1) so it improved the prediction results in comparison with KSVM and weighted-SVM but the best previous method (Anand et al.). However, RUS-THR was worse than RUS: since RUS itself balanced the dataset, the decrease in decision threshold resulted in a more high sensitivity and low specificity. Meanwhile, RUS-OSD achieved better sensitivity, specificity, and G-mean than the corresponding results of Anand et al. by eliminating a part of majority class samples and then using OSD to increase the minority class samples.

Two of our over-sampling methods, OSD and OSD-THR, outperformed the method of Anand et al. (Table 1). For example, over all accuracy, specificity, and G-mean of OSD were 10.70%, 12.30%, and 3.36% higher than the competing method while sensitivity was 3.18% lower. The latter approach, OSD-THR, was better than the best previous method at all evaluation metrics.

Since MCC was not reported in [6], we could not directly compare with their method, under various conditions. However, at least under the condition that sensitivity equals to 70%, the MCC values of the method in [11] and our method were 0.32 and 0.48, respectively. Fig. 2 describes the correspondence between MCC and sensitivity of KSVM and OSD.

Fig. 3 demonstrates the ROC curves of OSD and the other methods. ROC curve of Cheng and Jeong was taken from [11]. It shows that while RUS decreased the performance of KSVM, the combination of RUS and OSD achieved a better result.

Table 1. Performance measures comparison of different methods on the dataset D1050 in terms of best G-mean

Method	Overall accuracy (%)	Sensitivity (%)	Specificity (%)	G-mean
KSVM	90.11	4.66	99.93	21.59
OSD	88.23	67.86	90.57	78.40
RUS (1.1:1)	76.17	70.59	76.81	73.63
RUS-OSD	75.31	80.73	74.69	77.65
KSVM-THR	90.66	11.48	99.76	33.85
OSD-THR	83.36	77.73	84.01	80.80
RUS-THR(1.1:1)	65.71	82.11	83.82	72.39
RUS-OSD-THR	64.94	88.51	62.24	74.22
Weighted-SVM*	91.57	55.87	95.56	73.08
SMOTE*	92.96	51.74	97.69	71.07
Chen and Jeong (2009)*	71.90	71.20	71.98	71.59
Anand et al. (2010)*	77.53	71.04	78.27	74.54

**: Result was taken from the paper of Anand et al.*

Table 2. Performance of KSVM-THR, OSD-THR, RUS-THR and RUS-OSD-THR with different decision threshold values on the dataset D1050

Method	KSVM-THR				OSD-THR				RUS-THR				RUS-OSD-THR			
Thr	ACC	SN	SP	G	ACC	SN	SP	G	ACC	SN	SP	G	ACC	SN	SP	G
0.96	89.69	0.07	1.00	2.65	91.93	31.07	98.93	55.44	90.34	21.63	98.24	46.10	90.66	31.28	97.49	55.22
1.73	89.69	0.00	1.00	0.00	89.68	0.00	99.99	0.00	89.71	0.42	99.97	6.51	89.72	0.42	99.98	6.51
8.52	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00
-2.92	10.30	1.00	0.00	0.00	10.31	1.00	0.00	0.00	10.31	1.00	0.00	0.00	10.30	1.00	0.00	0.00
-1.24	17.60	99.71	8.16	28.54	35.80	98.23	28.63	53.03	20.02	99.46	10.89	32.91	21.19	99.78	12.16	34.83
-0.85	90.23	58.71	93.85	74.23	62.94	91.69	59.64	73.94	38.16	96.50	31.45	55.09	39.08	97.84	32.32	56.24
-0.79	91.52	49.80	96.32	69.26	65.90	90.70	63.05	75.62	41.35	95.72	35.10	57.97	41.97	97.13	35.63	58.83
-0.73	92.03	43.51	97.61	65.17	68.36	89.74	65.90	76.90	43.90	94.76	38.06	60.05	44.43	96.42	38.45	60.89
-0.58	91.91	29.26	99.11	53.85	74.74	86.63	73.37	79.73	51.83	91.19	47.31	65.68	51.82	94.23	46.94	66.51
-0.45	91.34	20.25	99.51	44.89	78.72	83.28	78.20	80.70	57.81	87.84	54.35	69.10	57.28	92.08	53.28	70.04
-0.37	91.01	15.69	99.67	39.55	81.14	80.98	81.16	81.07	61.57	85.25	58.85	70.83	60.83	90.77	57.39	72.18
-0.32	90.81	13.22	99.73	36.31	82.47	79.28	82.84	81.04	63.92	83.42	61.67	71.73	63.14	89.46	60.11	73.33
-0.28	90.66	11.48	99.76	33.85	83.36	77.73	84.01	80.80	65.71	82.11	63.82	72.39	64.94	88.51	62.24	74.22

**Thr = Decision threshold; *ACC = accuracy (%); *SN = sensitivity (%); *SP = specificity (%); *G = G-mean (%)*

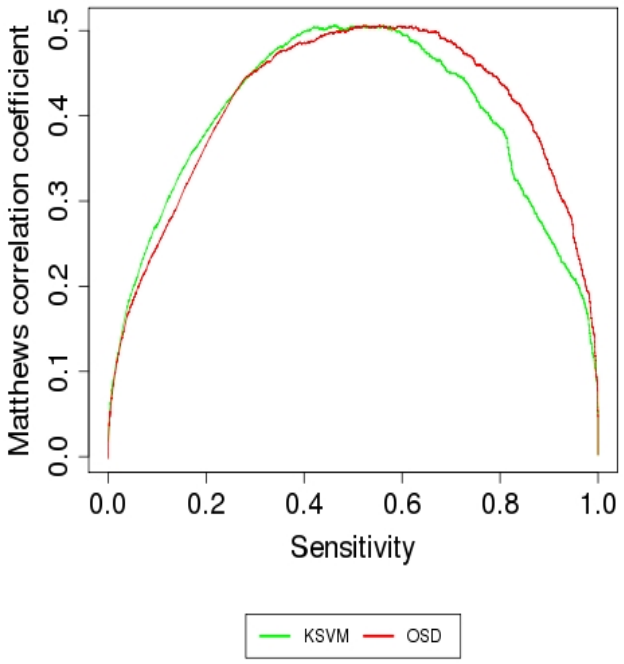

Fig. 2. MCC vs. sensitivity of the two methods KSVM and OSD on the D1050 dataset

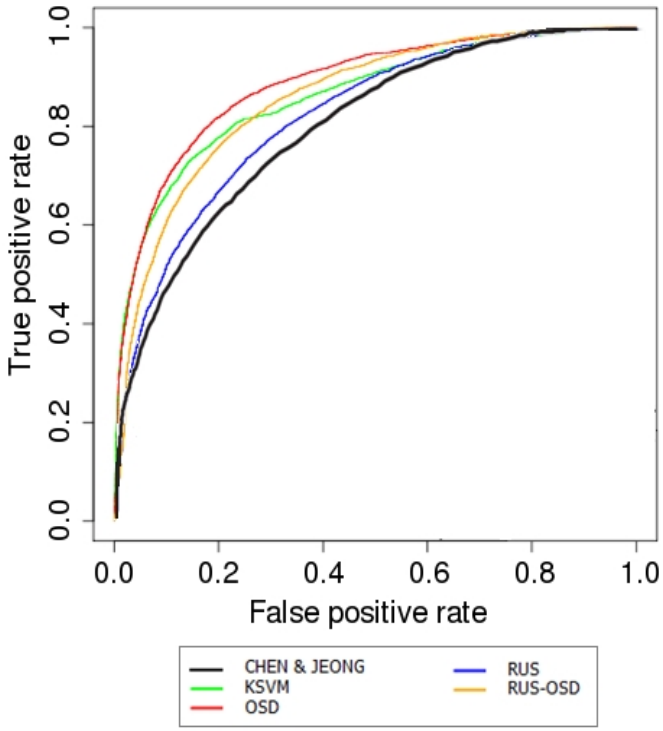

Fig. 3. ROC curves of the competing methods on the D1050 dataset

3.2 Evaluation on the D1239 Dataset

We conducted experiments on the D1239 dataset and compared with the results of the D1050 to evaluate the effect of shape strings and the new over-sampling algorithm on the PPI sites prediction problem.

In addition to the evaluation criteria above, F-measure and Area Under Precision/Recall Curve(AUC-PR) [16] were used. F-measure is defined as follows:

F-measure $= (2 * precision * recall)/(precision + recall)$

where:

precision $= TP/(TP + FP)$

recall$= TP/(TP + FN)$

These metrics show the ability of classifier for detecting rare positive samples in the imbalanced dataset. Table 3 shows the results of experiments on the dataset D1239 with the different decision thresholds of the methods. Table 4 shows the improvements using our algorithm and new decision threshold in the comparison of the naïve classifier. In Table 4, OSD and OSD-THR outperformed the others and the best previous result in G-mean. It indicates that our over-sampling algorithm based on the local density can relieve the class-imbalance problem in this dataset. On the other hand, KSVM and KSVM-THR on the dataset D1239 achieved higher accuracy, sensitivity, G-mean than on the D1050. It demonstrated that shape string is an informative feature for discriminating interface and non-interface residues. Fig. 4 and 5 show that performance curves on D1239 are similar to the ones on D1050.

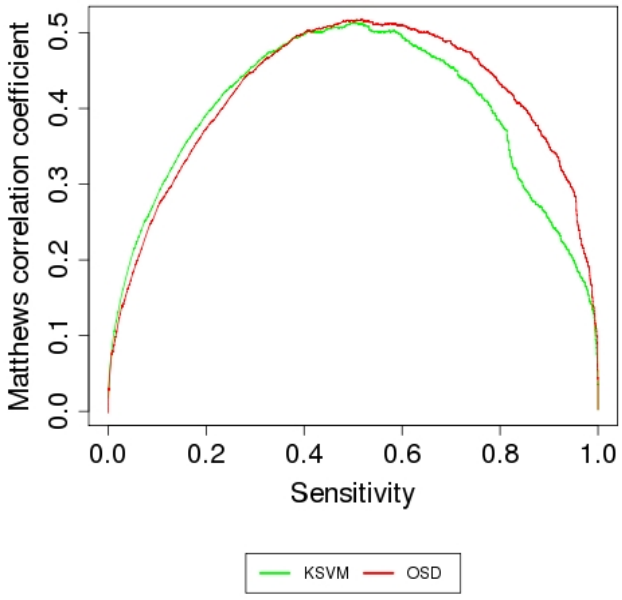

Fig. 4. MCC vs. sensitivity of KSVM and OSD on the D1239 dataset

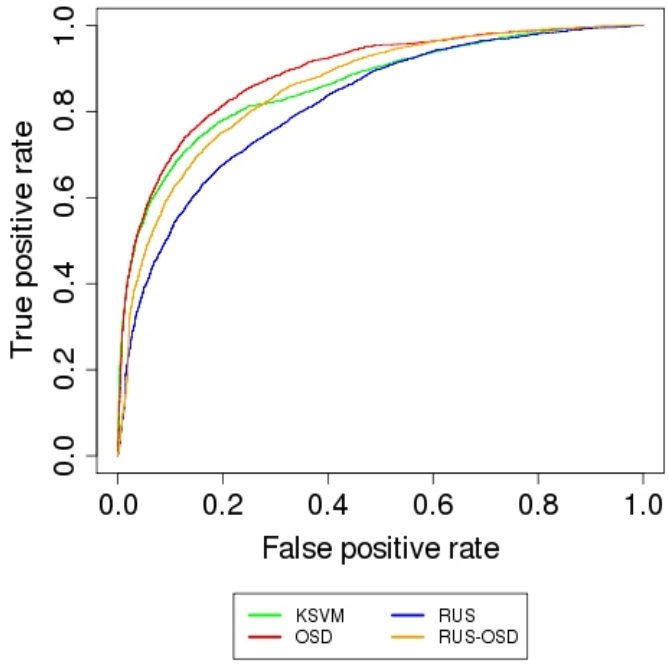

Fig. 5. ROC curves of the competing methods on the D1239 dataset

Table 5 displays the comparative results on the datasets D1050 and D1239. Though sensitivity of OSD and OSD-THR decreased 4.73% and 3.29% (from 67.86% to 63.13% and from 77.73% to 74.44%), respectively, precision increased 4.45% and 3.42%. All the experiments on D1239 achieved higher F-measure than the corresponding one on the D1050. In addition, F-measure of OSD and OSD-THR on the both datasets are higher than that one of Chen and Jeong (49%) [17]. Furthermore, AUC-PR of KSVM and OSD on D1050 and D1239 were 0.56, 0.55, 0.58, and 0.57, respectively. In Fig. 6, it can be seen that the performance of KSMV on D1239 is apparently better than the one on D1050 in the area of recall lower than 0.3 and precision higher than 0.8. It means that shape string is effective for performance improvement in this area.

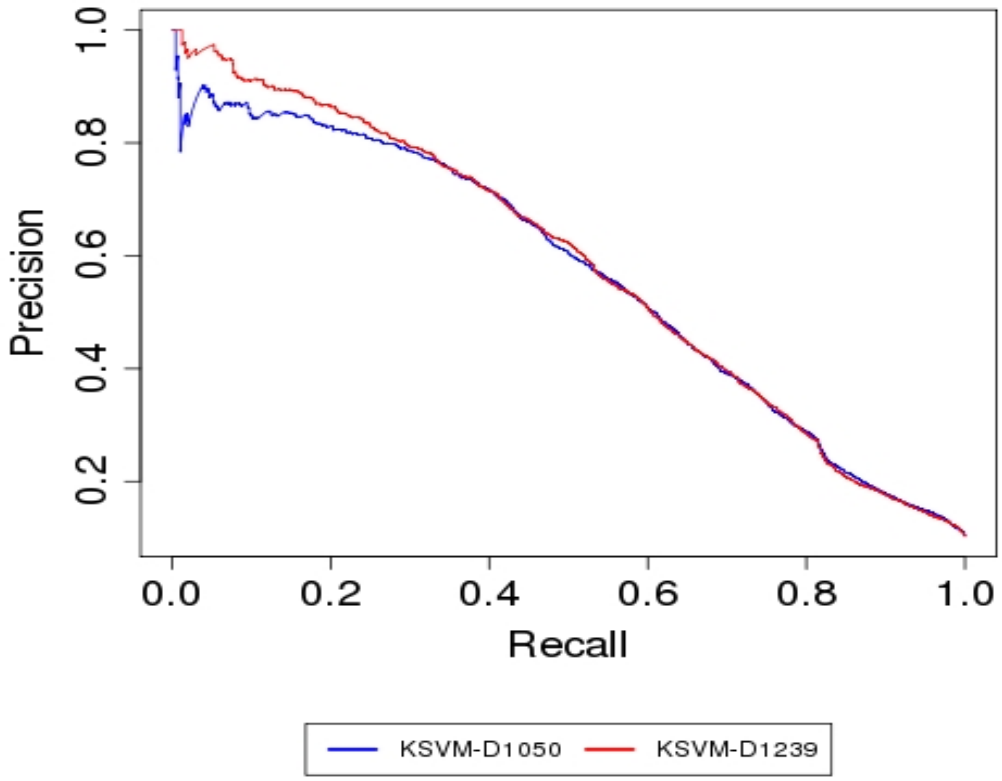

Fig. 6. PR curves for the datasets with shape string (D1239) and without shape string (D1050) prediction with KSVM as basic classifier

Table 3. Performance of KSVM-THR, OSD-THR, RUS-THR and RUS-OSD-THR with different decision threshold values on the dataset D1239

Method	KSVM-THR				OSD-THR				RUS-THR				RUS-OSD-THR			
Thr	ACC	SN	SP	G	ACC	SN	SP	G	ACC	SN	SP	G	ACC	SN	SP	G
0.96	89.70	0.14	1.00	3.76	90.66	12.44	99.65	35.21	90.30	20.29	98.35	44.67	91.00	32.91	97.68	56.70
1.73	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.67	0.18	99.96	4.20	89.69	0.49	99.94	7.03
8.52	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00	89.69	0.00	1.00	0.00
-2.92	10.30	1.00	0.00	0.00	10.30	1.00	0.00	0.00	10.31	1.00	0.00	0.00	10.31	1.00	0.00	0.00
-1.24	17.85	99.61	8.46	29.03	34.74	98.30	27.44	51.93	19.73	99.26	10.59	32.43	20.99	99.61	11.96	34.51
-0.85	90.11	59.31	93.65	74.53	64.88	92.08	61.75	75.41	37.76	96.36	31.02	54.67	39.13	97.63	32.40	56.24
-0.79	91.60	50.83	96.29	69.96	68.02	90.27	65.46	76.88	41.01	95.40	34.76	57.59	42.02	97.24	35.68	58.90
-0.73	91.99	44.50	97.45	65.85	70.54	89.14	68.40	78.09	43.69	94.56	37.85	59.82	44.65	96.50	38.69	61.10
-0.58	91.97	29.97	99.10	54.50	76.93	84.72	76.04	80.26	51.66	91.34	47.10	65.59	52.28	94.49	47.43	66.95
-0.45	91.57	22.23	99.54	47.04	80.92	80.38	80.98	80.68	57.61	87.45	54.18	68.83	58.00	92.19	54.07	70.60
-0.37	91.32	18.55	99.68	43.01	83.21	77.80	83.83	80.76	61.33	84.66	58.65	70.46	61.50	90.31	58.19	72.49
-0.32	91.17	16.54	99.74	40.62	84.56	75.82	85.57	80.54	63.76	82.40	61.62	71.26	63.95	88.55	61.12	73.57
-0.28	91.07	15.30	99.78	39.08	85.49	74.44	86.76	80.36	65.54	80.88	63.78	71.82	65.72	87.56	63.21	74.39

**Thr = Decision threshold; *ACC = accuracy (%); *SN = sensitivity (%); *SP = specificity (%); *G = G-mean (%)*

Table 4. Performance measures comparison of different methods on the dataset D1239

Method	Overall accuracy (%)	Sensitivity (%)	Specificity (%)	G-mean
KSVM	90.45	8.02	99.92	28.31
OSD	89.61	63.13	92.66	76.48
KSVM-THR	91.07	15.30	99.78	34.79
OSD-THR	85.49	74.44	86.76	80.36

Table 5. Performance measures comparison on the datasets D1239 and D1050

Data set	Method	Precision (%)	Recall (%)	F-measure (%)
D1050	KSVM	89.18	4.66	8.86
	OSD	45.27	67.86	54.31
	KSVM-THR	85.07	11.48	20.24
	OSD-THR	35.84	77.73	49.06
D1239	KSVM	92.65	8.02	14.76
	OSD	49.72	63.13	55.63
	KSVM-THR	89.09	15.30	26.12
	OSD-THR	39.26	74.44	51.40

4. CONCLUSION

In this study, we aimed at the identification of protein-protein interaction sites. The PPI datasets used in this study were highly class-imbalanced, which often decrease classification performance of SVMs. To avoid this issue, we proposed a novel over-sampling technique that effectively utilizes local density of minority samples. We also proposed several methods combined with KSVM-THR and random under-sampling methods to reinforce the tolerance for the class imbalance problem. Experimental results showed that the combination of our OSD algorithm and new feature group led to higher sensitivity, G-mean, precision, MCC, F-measure, and AUC-PR, at least comparable performance with the state-of-the-art methods. In addition, we found that the information of predicted shape strings increase the performance for predicting whether interface or non-interface residues. Further extensions can be considered, for example, combining our algorithm with other heuristic under-sampling method, or feature selection methods.

ACKNOWLEDGEMENTS

The first author has been supported by Vietnamese Government Scholarship to study in Japan. The authors wish to thank Xue-wen Chen and Jong Cheol Jeong for discussions and providing us the datasets.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. He H, Garcia EA. Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering. 2009;21(9):1263–1284.
2. Yousef M, Nebozhyn M, Shatkay H, Kanterakis S, Showe LC, Showe MK. Combining multi-species genomic data for microRNA identification using a Naive Bayes classifier. Bioinformatics (Oxford, England). 2006;22(11):1325–34.
3. Tang Z, Li T, Liu R, Xiong W, Sun J, Zhu Y, Chen G. Improving the performance of β-turn prediction using predicted shape strings and a two-layer support vector machine model. BMC bioinformatics. 2011;12(1):283.
4. Kountouris P, Hirst JD. Predicting beta-turns and their types using predicted backbone dihedral angles and secondary structures. BMC bioinformatics. 2010;11(1):407.
5. Batuwita R, Palade V. microPred: effective classification of pre-miRNAs for human miRNA gene prediction. Bioinformatics (Oxford, England). 2009;25(8):989–995.
6. Anand A, Pugalenthi G, Fogel GB, Suganthan PN. An approach for classification of highly imbalanced data using weighting and undersampling. Amino acids. 2010;39(5):1385–91.
7. Lin W-J, Chen JJ. Class-imbalanced classifiers for high-dimensional data. Briefings in bioinformatics. 2012.
8. Sun J, Tang S, Xiong W, Cong P, Li T. DSP: a protein shape string and its profile prediction server. Nucleic acids research. 2012;40(Web Server):W298–302.
9. Zhu Y, Li T, Li D, Zhang Y, Xiong W, Sun J, et al. Using predicted shape string to enhance the accuracy of γ-turn prediction. Amino acids. 2012;42(5):1749–55.
10. Wang D, Li T, Sun J, Li D, Xiong W, Wang W, et al. Shape string: a new feature for prediction of DNA-binding residues.Biochimie. 2013;95(2):354-358.
11. Chen X, Jeong JC. Sequence-based prediction of protein interaction sites with an integrative method. Bioinformatics (Oxford, England). 2009;25(5):585–91.
12. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 2002;16(1):321–357.
13. Prati RC, Batista GEAPA, Monard MC. Class Imbalances versus Class Overlapping: An Analysis of a Learning System Behavior. in Proc. MICAI; 2004
14. Hui Han, Wenyuan Wang BM. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. ICIC. 2005;878–887.
15. Kubat M, Holte R, Matwin S. Learning When Negative Examples Abound. Lecture Notes in Computer Science. 1997;1224:146–153.
16. Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning - ICML '06. New York, New York, USA: ACM Press. 2006;233–240.
17. Chen P, Li J. Sequence-based identification of interface residues by an integrative profile combining hydrophobic and evolutionary information. BMC bioinformatics. 2010;11(1):402.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history.php?iid=196&id=9&aid=1162