Sample Subset Optimization Techniques for Imbalanced and Ensemble Learning Problems in Bioinformatics Applications

Pengyi Yang, Paul D. Yoo, Juanita Fernando, Bing B. Zhou, Zili Zhang, and Albert Y. Zomaya, *Fellow, IEEE*

Souce code available from: https://code.google.com/p/sample-subset-optimization/

***Abstract*—Data sampling is a widely used technique in a broad range of machine learning problems. Traditional sampling approaches generally rely on random re-sampling from a given dataset. However, these approaches do not take into consideration additional information, such as sample quality and usefulness. We recently proposed a data sampling technique, so-called, sample subset optimization (SSO). SSO technique relies on a cross-validation procedure for identifying and selecting the most useful samples as subsets. In this study, we describe the application of SSO techniques to imbalanced and ensemble learning problems, respectively. For imbalanced learning, SSO technique is employed as an under-sampling technique for identifying a subset of highly discriminative samples in the majority class. In ensemble learning, SSO technique is utilized as a generic ensemble technique where multiple optimized subsets of samples from each class are selected for building an ensemble classifier. We demonstrate the utilities and advantages of the proposed techniques on a variety of bioinformatics applications where class imbalance, small sample size, and noisy data are prevalent.**

***Index Terms*—Sample subset optimization, under-sampling, imbalanced learning, ensemble learning, bioinformatics applications**

I. Introduction

DATA sampling is a key technique that has been utilized in a broad range of machine learning and data mining problems such as imbalanced learning [1] and ensemble learning [2]. In imbalanced learning, where samples from one class (majority class) significantly outnumber samples from the other class (minority class), a sampling technique may be employed to select a subset of samples from the majority class so as to create a more balanced class distribution. In ensemble learning, samples may be randomly re-sampled to create multiple data subsets which are subsequently used to train a group of classifiers each project the data in a different model. Many sampling techniques employed in these learning tasks rely on random re-sampling where samples are randomly selected with or without replacement. One of the main disadvantages of random re-sampling based approach is that it does not utilize additional information such as sample quality and their discriminative ability among classes, which could be useful in data classification. We recently proposed a data sampling technique called sample subset optimization (SSO) to specifically looking for the most discriminative sample subsets from all available samples [3]. In this paper, we describe the application of SSO techniques to imbalanced learning and ensemble learning, respectively. For imbalanced learning, SSO technique is employed as an under-sampling technique. Whereas for ensemble learning, SSO technique is utilized as a generic ensemble technique. Particularly, we demonstrate the utilities of these applications in addressing a variety of bioinformatics problems where class imbalance, small sample size, and noisy data are prevalent.

Class imbalance learning has recently gained considerable attention especially in bioinformatics domains that, quite often, only a limited number of positive samples are available whereas the number of negative controls is relatively much more abundant. Typical examples include the identification of genes [4], promoters [5], or splice site [6] from DNA sequences where the positive samples are inherently rare compared to their negative counterparts [7]. The classification task in such a situation is often complicated by the highly imbalanced class distribution where the negative samples from the majority class are over-represented by a learning model compared to the positive samples from the minority class, leading to an undesirable bias in the model decision boundary. For example, popular classifiers such as support vector machine (SVM) and k-nearest neighbor (kNN) are found to be very sensitive to the imbalanced class distribution [8], [9], and may perform sub-optimally when applied to the imbalanced dataset without skewness correction.

Data sampling is a popular approach to address the imbalanced class distribution [10]. In the simplest form, samples from the majority class are randomly removed to match the minority class (refer to as random under-sampling), or samples from the minority class are randomly duplicated to match the majority class (refer to as random over-sampling) [11].

Pengyi Yang is with the School of Information Technologies, University of Sydney, NSW 2006, Australia; and National Institute of Environmental Health Sciences (NIEHS), National Institute of Health (NIH), Research Triangle Park, NC 27709, USA (e-mail: pengyi.yang@nih.gov)

Paul D. Yoo is with the Department of Electrical and Computer Engineering, at Khalifa University of Science, Technology and Research, Abu Dhabi, UAE (e-mail: paul.usyd@gmail.com).

Juanita Fernando is with Faculty of Medicine, Nursing & Health Sciences, Monash University, Clayton, Australia (e-mail: juanita.fernando@med.monash.edu.au).

Bing B. Zhou and Albert Y. Zomaya are with the School of Information Technologies, University of Sydney, NSW 2006, Australia (e-mail: bing.zhou@sydney.edu.au; albert.zomaya@sydney.edu.au).

Zili Zhang is with Faculty of Computer and Information Science, Southwest University, Chongqing, China; and School of Information Technology, Deakin University, VIC 3217, Australia (e-mail: zhangzl@swu.edu.cn).

Corresponding to Pengyi Yang, phone: 1-919-5414996; fax: 1-919-5414311; e-mail: pengyi.yang@nih.gov

Manuscript received XX XX, XXXX; revised XX XX, XXXX.

However, as aforementioned these simple approaches do not distinguish which samples are more informative, and may remove representative samples or increase noise and introduce duplications [1]. A more sophisticated approach known as SMOTE is to synthesize "new" samples using original samples in the dataset [12]. Yet, for dataset with a large number of samples and highly imbalanced class distribution, a large number of synthetic samples will be introduced, which may substantially increase noise in the data. In [13], several evolutionary approaches have been proposed for creating roughly balanced dataset by using balance level and/or classification measures. In this study, we extend SSO-based data sampling along this line of research by using a cross-validation procedure and a frequency ranking procedure to detect the most useful samples from the majority class and 'intelligently' (as opposed to randomly) under-sample the given dataset.

Ensemble learning is a popular technique in bioinformatics applications [14], [15]. A key synergy between ensemble learning and imbalanced learning is the use of data sampling techniques. In ensemble learning, training samples are re-sampled multiple times to build multiple models each learns the decision boundary with a different sample subset, weights, and/or feature subset. This is particularly beneficial when the number of samples is limited which is very common in bioinformatics applications [16] due to the high expenses on sample collection and processing.

There are many ensemble methods. Among them, *bagging* [17], *boosting* [18], and their variants [19] are the most popular approaches and are frequently applied to bioinformatics applications [20], [21]. With bagging, training data are re-sampled with replacement to produce multiple training subsets, and multiple classifiers are built each using a different training subset. The boosting algorithm creates multiple classifiers iteratively. That is, an initial classifier is built with the original training data, and based on the misclassification in each iteration the algorithm adds more weights to samples that are misclassified in previous iteration and builds a new classifier for the modified training data. The ensemble is created by combining classifiers from multiple iterations. Generally, decision tree or decision stump is used to form the *base* classifier of the ensemble because (1) they can be learnt efficiently and (2) they are unstable to perturbed data which brings diversity to the ensemble [22]. While bagging algorithm relies on random sampling and therefore works in a random manner, boosting algorithm tries to classify the most "difficult" samples by greedily increasing the weight of "difficult" samples. Neither of them takes sample quality and/or sample usefulness into account and is susceptible to outliers and noisy data.

In contrast, by using SSO for ensemble learning, the training samples are used selectively according to their discriminative ability for creating base classifiers. That is, only samples that are deemed useful by SSO will be used for base classifier training. We show that SSO-based ensemble approach can improve the quality of base classifiers and increase the classification accuracy of the ensemble.

The rest of the paper is organized as follows: In Section II, we introduce the SSO technique. In Section III, we formulate the SSO procedure for imbalanced learning and ensemble learning. Section IV details the experimental setups, and Section V presents the experimental results. We conclude in Section VI and outline the future work.

II. Sample Subset Optimization

The key idea of sample subset optimization (SSO) is to select a subset of all available samples by minimizing the expected error using a cross-validation procedure on the training data. Let us formulate the expected k-fold cross-validation error term as follows:

$$\hat{\varepsilon}_{cvk} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n/k} \mid y_j^{(i)} - p(t_j^{(i)}|\theta^{(i)}, \boldsymbol{x}_j^{(i)})) \mid$$

where n is the number of samples, and k is the number of folds. $y_j^{(i)}$ denotes the given class label of the sample j from the test fold i. $p(t_j^{(i)}|\theta^{(i)}, \boldsymbol{x}_j^{(i)})$ denotes the prediction of the j^{th} sample from the test fold i using a feature vector $\boldsymbol{x}_j^{(i)}$ with a model $p(.)$ parameterized by $\theta^{(i)}$. In particular $\theta^{(i)}$ is derived from the training fold which could be formed by combining **some** or all samples but not those from fold i. The predicted value of the j^{th} sample is denoted as $t_j^{(i)}$.

For each fold i, SSO attempts to identify a subset of the most useful training samples denoted as $s^{(i)}$ from which a set of parameters $\theta^{(i)}(s^{(i)})$ is learnt and subsequently used for parameterizing a prediction model $p(.)$. This can be denoted as to minimize the following quantity:

$$\min_{s^{(i)}} \hat{\varepsilon}_{cvk}^{(i)} = \frac{k}{n}\sum_{j=1}^{n/k} \mid y_j^{(i)} - p(t_j^{(i)}|\theta^{(i)}(s^{(i)}), \boldsymbol{x}_j^{(i)})) \mid$$

The above can be achieved in an iterative manner by maximizing a fitness function such as $1 - \hat{\varepsilon}_{cvk}^{(i)}$. In this maximizing process, a sequence of $s_1^{(i)}, s_2^{(i)}, ..., s_g^{(i)}$ is generated. The optimization terminates when a predefined iteration g is reached and the sample subset $s_g^{(i)}$ from the last iteration is saved as an optimized subset. If a population-based optimization approach such as particle swarm optimization (PSO) [23] or genetic algorithm (GA) is applied, multiple optimized sample subsets (denoted as $\boldsymbol{s}_g^{(i)}$) can be obtained in a single optimization run. We call such a population-based approach the *batch optimization approach*.

Repeating the above procedure for each of all k folds in the cross-validation partitioning, we obtain $\boldsymbol{s}_g^{(1)}, \boldsymbol{s}_g^{(2)}, ..., \boldsymbol{s}_g^{(k)}$ each one is optimized by a different test fold and contains a population of optimized sample subsets.

III. The Proposed SSO-based Algorithms

In this section, we describe the design and the application of SSO-based algorithms for (1) imbalanced and (2) ensemble learning.

A. SSO-based Imbalance Learning

By using SSO for under-sampling, the goal is to select a subset of samples from the majority class which can evenly be combined with the samples from the minority class so that

the subset could best represent the decision boundary between the two classes.

Suppose that we use PSO for optimization, then for each sample from the majority class, a dimension in the particle space is assigned. Assuming that we have d majority samples for a training fold, a "particle" in PSO can be coded as an indicator function set $P = \{I_1,\ I_2, ...,\ I_d\}$. For each dimension, an indicator function I_j takes value "1" when the corresponding j^{th} sample is included to train a classifier. Similarly, a "0" denotes that the corresponding sample is excluded from training. The fitness function for each fold is $1 - \hat{\varepsilon}^{(i)}_{cvk}$.

GA can also be used for optimization. Here the majority samples are coded as an indicator function set $P = \{I_1,\ I_2, ...,\ I_d\}$ on a "chromosome" of GA. Similarly, an indicator function I_j takes either "1" or "0" corresponding to select or exclude the j^{th} sample from training a classifier. Standard genetic operations such as mutation and crossover can be applied to optimize the fitness function of each fold $1 - \hat{\varepsilon}^{(i)}_{cvk}$.

One approach to summarize the final result is to rank the samples from the majority class by the number of times they are included in the optimized subsets of $\boldsymbol{s}_g^{(1)},\ \boldsymbol{s}_g^{(2)}, ...,\ \boldsymbol{s}_g^{(k)}$.

Algorithm 1 summarizes this procedure in pseudo code. Assuming we provide the algorithm an imbalanced dataset D^{o} and a cross-validation fold size of k, line 4 sets the optimized subsets $\mathbf{S} = \emptyset$ and the algorithm goes into the loop by applying a stratified k-fold cross-validation to generate training and validation sets. For each pair of training and validation sets, they are subsequently used by function 'SSO(.)' to select optimized sample subsets from the majority class $\boldsymbol{s}_g^{(i)}$ which when combined with samples from the minority class minimizes the classification error on model h. The optimized subsets from all k folds are saved in $\mathbf{S}$. Once the cross-validation procedure is completed, the function 'generateBalanceData(.)' is applied to rank the samples from majority class that are most frequently included in optimized subsets $\mathbf{S}$ and the most frequently selected majority samples are selected to match the number of minority samples to generate a balanced dataset D^*. Then, a classifier C_θ is trained using this balanced dataset.

Algorithm 1 SsoSampling

1: **Input:** original dataset D^{o}; CV fold k
2: **Output:** balanced dataset D^*; classifier C_θ
3: // generate optimized sample subsets
4: $\mathbf{S} = \emptyset$;
5: **for** $i \in \{1 : k\}$ **do**
6: $train_i \leftarrow$ CV(D^{o}, i, $trainFold$=TRUE);
7: $valid_i \leftarrow$ CV(D^{o}, i, $trainFold$=FALSE);
8: // apply SSO to select samples from majority class
9: $\boldsymbol{s}_g^{(i)} \leftarrow$ SSO($train_i$, $valid_i$, h, maj=TRUE);
10: $\mathbf{S} \leftarrow \mathbf{S} \cup \boldsymbol{s}_g^{(i)}$;
11: **end for**
12: $D^* \leftarrow$ generateBalanceData(D^{o}, $\mathbf{S}$);
13: // train a single classifier
14: $C_\theta \leftarrow$ train(D^*, h);
15: **return** (D^*, C_θ);

B. SSO-based Ensemble Learning

The formulation of SSO-based ensemble learning is different from imbalanced data sampling in that all samples d from both the majority class and the minority class in the dataset are mapped to the indicator function set $P = \{I_1,\ I_2, ...,\ I_d\}$. Similarly, by using the fitness function $1 - \hat{\varepsilon}^{(i)}_{cvk}$ and an optimization technique such as PSO, a population of optimized sample subsets $\boldsymbol{s}_g$ can be generated in batch. Performing the above for each of the all k folds in cross-validation, we have $\boldsymbol{s}_g^{(1)},\ \boldsymbol{s}_g^{(2)}, ...,\ \boldsymbol{s}_g^{(k)}$. In this case each subset contains the selected samples from both classes. In contrast, in imbalanced sampling only samples from the majority class are selected while all samples from the minority class are used. Since a population of optimized sample subsets is produced each time, a population of base classifiers can be trained in a single cross-validation run and the ensemble classifier can be obtained efficiently by combining these base classifiers.

The SSO-based ensemble learning algorithm is summarized in Algorithm 2. Similar to SSO-based sampling, the required input is the original dataset D^{o} and the cross-validation fold size k. The algorithm then sets the optimized subsets $\mathbf{S} = \emptyset$ and goes into the k-fold cross-validation to generate training and validation sets. For each pair of training and validation sets, the 'SSO(.)' function is applied to generate optimized sample subsets by considering samples from both classes that could minimize cross-validation errors. After obtaining all k folds of optimized sample subsets $\mathbf{S}$, the function 'generateOptimizedData(.)' is applied to create multiple optimized datasets. This is achieved by taking the original dataset D^{o} and the optimized sample subsets $\mathbf{S}$ as inputs and generates multiple datasets each contains samples indexed in an optimized sample subset $\boldsymbol{s}_g$. The optimized datasets are subsequently used for training an ensemble classifier $\mathbf{E}_\Theta$ by calling the function 'ensemble(.)' which uses the majority voting rule to combine all base classifiers each trained on a optimized dataset in $\mathbf{D}^*$.

Algorithm 2 SsoEnsemble

1: **Input:** original dataset D^{o}; CV fold k
2: **Output:** balanced datasets $\mathbf{D}^*$; ensemble classifier $\mathbf{E}_\Theta$
3: // generate optimized sample subsets
4: $\mathbf{S} = \emptyset$;
5: **for** $i \in \{1 : k\}$ **do**
6: $train_i \leftarrow$ CV(D^{o}, i, $trainFold$=TRUE);
7: $valid_i \leftarrow$ CV(D^{o}, i, $trainFold$=FALSE);
8: // apply SSO to select samples from both classes
9: $\boldsymbol{s}_g^{(i)} \leftarrow$ SSO($train_i$, $valid_i$, h, maj=FALSE);
10: $\mathbf{S} \leftarrow \mathbf{S} \cup \boldsymbol{s}_g^{(i)}$;
11: **end for**
12: // generate multiple optimized datasets
13: $\mathbf{D}^* \leftarrow$ generateOptimizedData(D^{o}, $\mathbf{S}$);
14: // train an ensemble classifier
15: $\mathbf{E}_\Theta \leftarrow$ ensemble($\mathbf{D}^*$, h);
16: **return** ($\mathbf{D}^*$, $\mathbf{E}_\Theta$);

The parameters of PSO and GA are summarized in Table I. They were found to give good empirical results in our experiments.

TABLE I
PARAMETERS USED FOR PSO AND GA OPTIMIZATIONS.

Optimizer	Parameter	Value
PSO	iteration	100
	cognitive constant	1.43
	social acceleration constant	1.43
	inertia weight	0.689
	velocity boundary	0.018 – 0.982
GA	iteration	100
	selection	Roulette wheel
	crossover probability	0.7
	mutation probability	0.1

IV. EXPERIMENT SETUPS

A. Synthetic Datasets

Synthetic datasets were generated for analyzing the behavior of SSO-based algorithms. For SSO-based under-sampling, we generated 20 samples each with two features for the majority class using a normal distribution $\mathcal{N}(5,1)$ and 10 samples each with two features for the minority class using a normal distribution $\mathcal{N}(7,1)$. Five samples as "outliers" were introduced to the majority class but were generated from the normal distribution of the minority class. This gives a class ratio of 2:5.

The behavior of SSO-based ensemble learning was also analyzed in a similar manner. Particularly, we created a synthetic dataset which contains 20 samples each with two features for each of the two classes, generated from normal distributions of $\mathcal{N}(5,1)$ and $\mathcal{N}(7,1)$, respectively. For each class, five "outlier" samples were introduced by swapping the class labels. This gives a class ratio of 1:1.

B. Bioinformatics Applications

1) Imbalanced Data Sampling: The imbalanced class distribution is common in many bioinformatics tasks. We selected five datasets which represent four different bioinformatics applications. These include miRNA identification, protein localization prediction, promoter identification from DNA sequences, kinase substrate prediction from protein phosphorylation profiling. The dataset of miRNA contains 691 positive samples and 9,248 negative samples, which has a class ratio of 0.075. Each sample is represented by 21 features [24]. One of the key application in protein functional annotation is to distinguish membrane proteins from proteins localized in other cellular compartments [25]. The protein localization dataset obtained from study [26] contains 258 membrane proteins and 1,226 proteins from other cellular compartments each one is represented by 8 protein features (a class ratio of 0.21). For promoter sequence identification, we obtained a human promoter dataset and a drosophila promoter dataset. The human promoter dataset contains 471 promoter sequences and 5,131 ccoding sequences (CDS) and intron sequences with a class ratio of 0.092. As for drosophila promoter dataset, it contains 1,936 promoter sequences and 2,722 CDS and intron sequences with a class ratio of 0.71. We encoded the samples from the two promoter datasets into 16 dinucleotide features according to Rani *et al.* [27]. The kinase substrate phosphorylation dataset is obtained from [28]. The data were curated using phosphosites database [29] and it contains 20 known substrates of protein kinase *Mek* and 1,000 negative phosphorylation sites. Each phosphorylation site is represented by the level of inhibitions and sequence motif.

2) Ensemble Learning: The datasets included in ensemble learning represent a wide range of biomedical and bioinformatics applications. The biomedical datasets included for ensemble learning were from studies conducted on diabetes of Pima Indian population [30] and heart disease [31], while the gene expression profile on leukemia [32] and liver cancer [33] were included for representing the small sample size and high feature dimension setting in microarray studies. Compared to the above imbalanced datasets, these four datasets have a relatively balanced class distribution. The diabetes dataset has 268 positive and 500 negative samples each one is represented by 8 biomedical features. The heart disease dataset consist of 120 positive and 150 negative samples, and 13 biomedical features are used to describe each sample. For the leukemia dataset, the sample size is 45 for ALL and 25 AML. For the liver cancer dataset, there are 82 tumor samples versus 75 non-tumor samples. For leukemia and liver cancer datasets, median normalization and scaling were applied to normalize the data. We calculated the between-group to within-group sum-of-square (BSS/WSS) statistics [34] for each gene, and selected 100 most differentially expressed genes.

C. Overall Experimental Procedure

We utilized a double-layered cross-validation procedure for evaluating the performance of the SSO-based algorithms (Figure 1). The original dataset was initially partitioned into training and validation sets using a 5-fold cross-validation. This is referred to as the external 5-fold cross-validation. For each external cross-validation training partition, it was further partitioned by a second cross-validation to form the internal training and validation datasets for SSO optimization and sample selection. This second cross-validation is referred to as the internal 2-fold cross-validation. The internal 2-fold cross-validation procedure maximizes the information in sample optimization and minimizes the overlap of selected samples by using each of the two folds for sample selection and sample optimization once only. The external validation sets were then used for validation on models derived from their respective external training sets. This performance evaluation procedure is described in Algorithm 3. The external 2-fold cross-validation and 10-fold cross-validation were also evaluated on protein localization prediction dataset to study the influence of k value in the cross-validation procedure on the final result.

In SSO-based under-sampling experiments, we used kNN classifier for evaluation as kNN is known to be sensitive to the imbalanced class distribution. We implemented both PSO (denoted as SSO-PSO) and GA (denoted as SSO-GA) for optimization. The optimization procedure was guided by a kNN classifier (k=3 was used for all kNN classifiers unless otherwise noted) and the internal 2-fold cross-validation. The final kNN models derived from the optimized balanced datasets were evaluated by the external 5-fold cross-validation. The area under the ROC curve (AUC) [37] value was calculated

TABLE II
SUMMARY OF DATASETS, BIOINFORMATICS APPLICATIONS, TASK TYPE, DATA AVAILABILITY, AND DATA STATISTICS.

Bioinformatics Application	Task Type	Sample Size	Minor Class	Major Class	Number of Features	Availability
Drosophila promoter identification	imbalanced learning	4658	1936	2722	16	from [35]
Protein localization prediction	imbalanced learning	1484	258	1226	8	from [36]
Human promoter identification	imbalanced learning	5602	471	5131	16	from [35]
miRNA identification	imbalanced learning	9939	691	9248	21	from [24]
Kinase substrate prediction	imbalanced learning	1020	20	1000	5	from [28]
Diabetes patient classification	ensemble learning	768	268	500	8	from [36]
Heart disease patient classification	ensemble learning	270	120	150	13	from [36]
Leukemia subtype classification	ensemble learning	70	25	45	100*	from [32]
Liver cancer classification	ensemble learning	157	75	82	100*	from [33]

*Top-100 differentially expressed genes ranked by between-group to within-group sum-of-square (BSS/WSS) statistics.

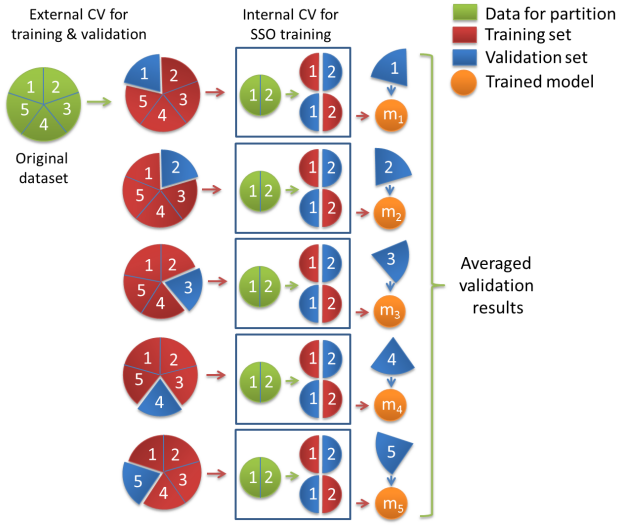


Fig. 1. A schematic illustration of the double-layered cross-validation procedure adopted in this study. The original dataset was initially partitioned by an external 5-fold cross-validation, forming the training and validation sets. The training sets from the external cross-validation were each further partitioned by a second internal cross-validation to form the partitions for SSO modeling. The validation sets from the external cross-validation were reserved for model validation.

Algorithm 3 Double-layered CV

Output: mean AUC value
$auc = \emptyset$;
for $i \in \{1:5\}$ **do**
  $exTrain_i \leftarrow$ CV(D^o, i, $trainFold$=TRUE);
  $exValid_i \leftarrow$ CV(D^o, i, $trainFold$=FALSE);
  $\mathbf{S}_i = \emptyset$;
  if $task$ = sampling **then**
    // obtain a balanced classifier
    $C_i \leftarrow$ SsoSampling($exTrain_i$, k=2)
  else
    // obtain an ensemble classifier
    $C_i \leftarrow$ SsoEnsemble($exTrain_i$, k=2)
  end if
  $auc_i \leftarrow$ evaluate($exValid_i$, C_i);
  $auc \leftarrow auc \cup auc_i$;
end for
return mean(auc);

for each fold and the mean from multiple folds was considered as a result of a test run. We performed 30 test runs to obtain the consensus performance using different random splitting point on each cross-validation.

In SSO-based ensemble learning experiments, we used decision tree (J48 implementation) as the base classifier because it is sensitive to small perturbation on datasets. We compared the performance with J48 alone and the ensembles from using bagging and boosting algorithms. PSO was used for optimization and the SSO-based ensemble is denoted as SSO$^{\mathrm{E}}$-PSO to distinguish those from imbalanced data sampling. During sample subset optimization, SSO$^{\mathrm{E}}$-PSO was guided by a decision tree and the internal 2-fold cross-validation. For each optimized subset, it was used to train a decision tree and the ensemble was formed by aggregating these decision trees with majority voting. The validation sets from the external 5-fold cross-validation were used for performance evaluation and the average result was treated as a test run. The ensemble sizes of 5, 10, 20, 30, 40 and 50 were tested. For each ensemble size, we repeated 10 test runs each with a different random splitting point on cross-validation.

V. RESULTS

A. SSO-based Under-sampling

1) On Synthetic Dataset: Figure 2a shows the decision boundary of a kNN on the original imbalanced dataset while Figure 2b shows the decision boundary of a kNN on one of the optimized subset using PSO for optimization. Three out of five outlier samples were removed after sample subset optimization. Note that this is a randomly selected subset from a population of many more subsets because both PSO and GA produced a population of optimized subsets in a single run. By counting the number of times each sample from the majority class appears in the optimized sample subsets, we could rank them from the most frequently selected ones to the least selected ones (Figure 2c). Figure 2d depicts the most frequently selected samples from the majority class after SSO procedure. All outliers were removed after the ranking and selection of the most useful majority samples. It is worth noting that while the decision boundary in Figure 2b closely resemble the final decision boundary in Figure 2d. the final decision boundary generated from ranking a population of all optimized sample subsets (performed and shown in Figure 2c) is expected to be less variant and therefore more robust for unseen data prediction.

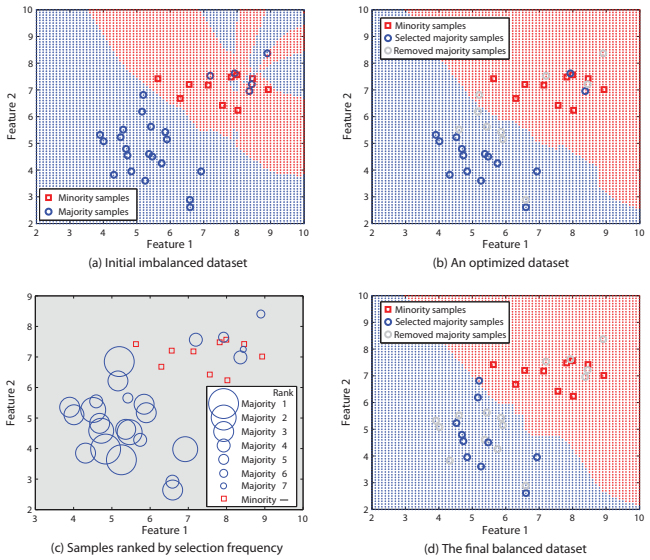


Fig. 2. SSO-based under-sampling on the synthetic dataset. The decision boundaries were created by using a kNN (k=3) classifier with Euclidean distance. (a) The decision boundary of kNN on the initial dataset that has imbalanced class distribution and five outliers. (b) The decision boundary on a dataset after SSO-based optimization. (c) Samples from the majority class were ranked by their frequency of being included in optimized sample subsets (i.e. "Majority 1", "Majority 2", etc.). The larger the circle, the more frequent a sample was included in the optimized sample subsets. (d) The decision boundary on the balanced dataset. The balanced dataset was formed by selecting top ranked samples from the majority class to match the number of samples in the minority class.

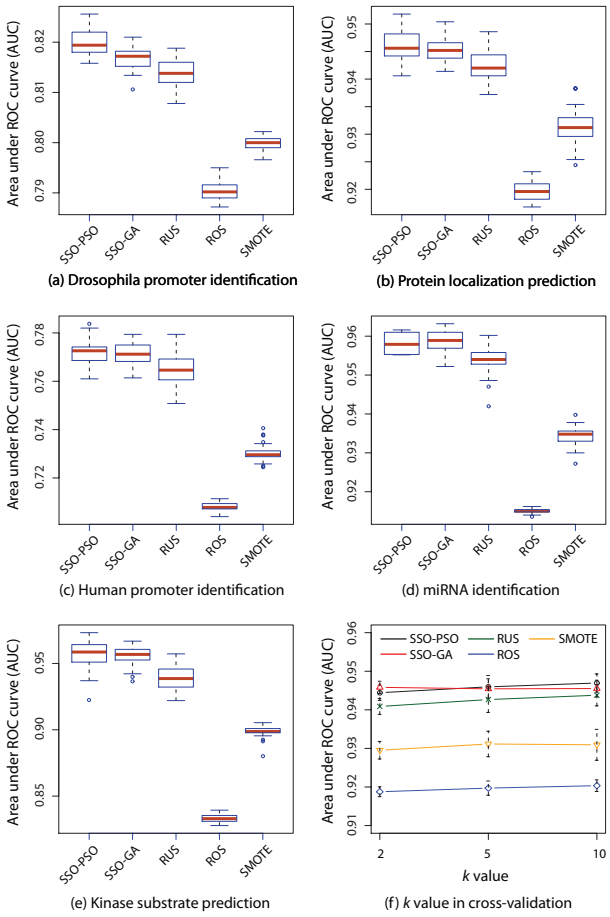


Fig. 3. Comparison of different methods for imbalanced data sampling and classification. The methods included in comparisons are SSO-PSO, SSO-GA, random under-sampling (RUS), random over-sampling (ROS), and SMOTE sampling. From (a) to (e), for each method, the classification accuracies were calculated from external 5-fold cross-validation repeated 30 times each with a different splitting point on each of the five datasets, respectively. The boxplots summarize the classification accuracies of 30 evaluation runs. (f) shows the classification results of each sampling method on protein localization dataset using the k value of 2, 5, and 10 for k-fold cross-validation.

2) Bioinformatics Applications: We evaluated SSO-based under-sampling technique in four bioinformatics applications (as described in previous section) where the learning is confounded by highly imbalanced class distribution. The performance of SSO-PSO and SSO-GA were compared with random under-sampling (RUS), random over-sampling (ROS), and SMOTE sampling techniques. Figure 3a-e show the classification results using the balanced dataset after applying each sampling method. For each sampling method, 30 test runs where conducted and the boxplots were used to summarize their performance.

It is evident that SSO-based under-sampling with kNN classification achieved, on average, the highest AUC values from all of the four datasets. We also observed that under-sampling techniques (SSO-PSO, SSO-GA, and RUS) performed significantly better than over-sampling techniques (ROS and SMOTE). This may be due to the fact that a large number of duplicated or artificial samples were introduced by over-sampling techniques for highly imbalanced datasets with a large sample size. Under-sampling techniques, however, do not cause such duplications. Within under-sampling approaches, SSO techniques helped to identify more useful samples from the majority class; thus, it performed better than random under-sampling approach. To investigate the impact of the k value on the external cross-validation procedure, we evaluated the k value of 2, 5, and 10-fold cross-validation on the protein localization dataset (Figure 3f). We found that the influence of k is minor on the final classification results. The k value of 5 was used in the rest of the experiments.

To prove that the improvements are statistically significant, we followed [38] and adopted the Wilcox signed rank test to compare the performance of SSO-PSO and SSO-GA with those of the other three methods. Table III shows the p-values for the comparisons. For those have a p-value smaller than 10^{-10}, we denoted them as $p < 1.0\times10^{-10}$. In most cases, the performance of SSO-based methods were significantly better than the other three methods with a p-value smaller than 0.05.

B. SSO-based Ensemble Learning

1) On Synthetic Dataset: The behavior of SSO on ensemble learning was similar to those on imbalanced sampling. As shown in Figure 4a, the initial dataset contains 5 outliers for each class. Figure 4a also shows the classification boundary created by a decision tree using the initial dataset whereas Figure 4b-d each shows a decision tree classification boundary after using SSO optimization procedure. Figure 4b-d were selected as examples to represent the typical decision boundaries from the population of all SSO optimized sample subsets.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT SAMPLING METHODS ON IMBALANCED LEARNING DATASETS USING WILCOX SIGNED RANK TEST.

Algorithms	Drosophila promoter	Protein localization	Human promoter	miRNA identification	Substrate prediction
SSO-PSO vs. RUS	$p = 1.8 \times 10^{-9}$	$p = 2.3 \times 10^{-4}$	$p = 1.5 \times 10^{-4}$	$p = 1.2 \times 10^{-2}$	$p = 3.9 \times 10^{-7}$
SSO-GA vs. RUS	$p = 4.4 \times 10^{-5}$	$p = 7.9 \times 10^{-4}$	$p = 4.6 \times 10^{-4}$	$p = 4.0 \times 10^{-5}$	$p = 5.4 \times 10^{-8}$
SSO-PSO vs. ROS	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$
SSO-GA vs. ROS	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$
SSO-PSO vs. SMOTE	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$
SSO-GA vs. SMOTE	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$	$p < 1.0 \times 10^{-10}$

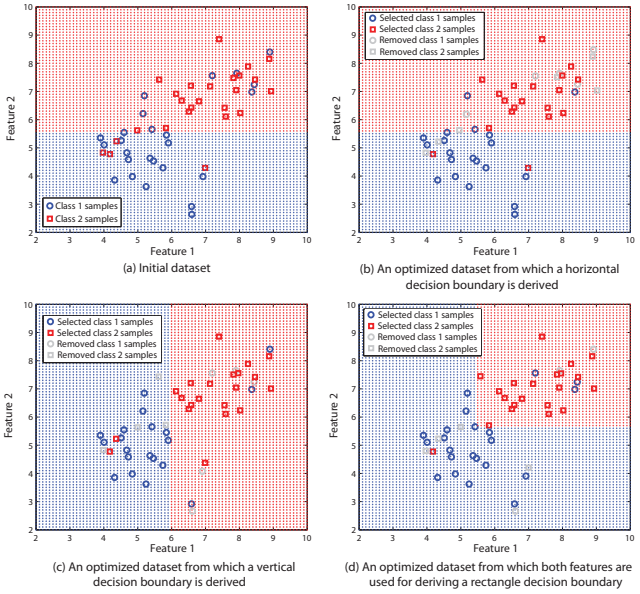


Fig. 4. SSO-based ensemble learning on the synthetic dataset. The decision boundaries were created by using a decision tree (J48) classifier. (a) The decision boundary of a decision tree on the initial dataset that has five outliers in each of the two classes. (b) An example of a horizontal decision boundary on a dataset after SSO-based sample subset selection. (c) An example of a vertical decision boundary on a dataset after SSO-based sample subset selection. (d) An example of a rectangle decision boundary on a dataset after SSO-based sample subset selection.

Specifically, Figure 4b shows a decision tree model in which a horizontal decision boundary is obtained from an SSO optimized sample subset. Compared to the initial dataset, 7 out of 10 outliers, which were introduced to the two classes artificially, have been removed. Figure 4c shows a different example that a decision tree model with a vertical decision boundary was obtained from the SSO optimized sample subset. Similarly, 6 out 10 outliers have been removed after SSO optimization. Figure 4d shows another typical decision tree model where both feature 1 and feature 2 were used as the splitting points in the model to from a rectangle decision boundary. These three examples illustrated that the diversity and the improved quality of the base classifiers of which the ensemble classifier is comprised.

2) Bioinformatics Applications: The classification results from SSO-based ensemble with PSO (denoted as SSO$^{\mathrm{E}}$-PSO), Bagging, and Boosting are shown in Figure 5. The error bar represents the standard deviation of the classification over 10 independent test runs, except the results of the Boosting algorithm, because it is deterministic in terms of model construction. For each ensemble algorithm, the ensemble size of 5, 10, 20, 30, 40, and 50 were evaluated. Table IV shows the AUC values of a single decision tree on each dataset and the mean AUC values of each ensemble algorithm obtained by averaging across different ensemble sizes.

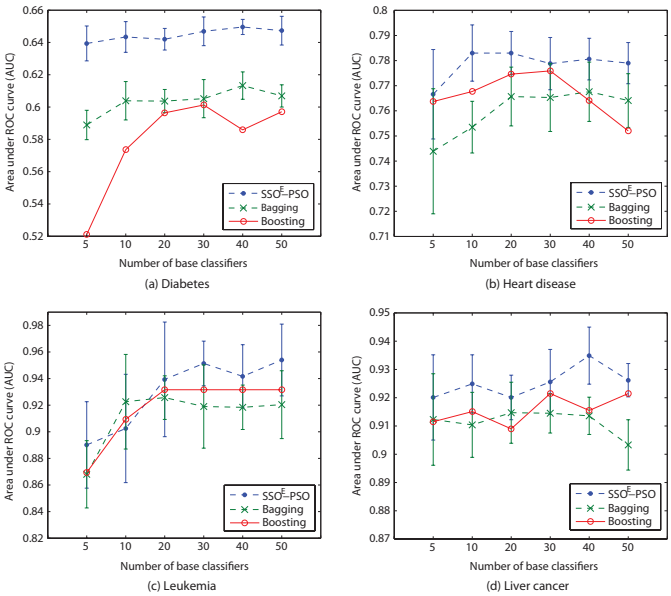


Fig. 5. Sample classification comparison. The y-axis indicates the classification accuracy in terms of AUC and the x-axis indicates the number of base classifiers used to form an ensemble. For each method, the classification accuracies were calculated from a 5-fold cross-validation repeated 10 times each with a different data partition. The middle points of the performance curves are the averages of the classification accuracies from these 10 evaluation runs and the error bars are the standard deviations.

We observed that ensemble algorithms significantly improved upon the single classifier of decision tree. In addition, as shown in Figure 5 and Table IV, SSO$^{\mathrm{E}}$-PSO is more accurate than Bagging and Boosting in all cases, indicating that sample subsets optimizaiton is indeed a viable procedure to improve ensemble accuracy. Bagging seems to be less successful, and this may attribute to its random procedure in subset generation. From this perspective, Boosting is more successful because the importance of the samples are updated progressively in each iteration. However, in the training process of Boosting, all training samples are considered and the highest classification weights are always given to these most "difficult" samples which could be the outliers of the dataset. Instead of trying to classify the most "difficult" samples aggressively, the proposed SSO-based ensemble technique attempts to utilize the most representative samples from the training set which may have a better generalization property

on unseen data classification.

TABLE IV
PERFORMANCE OF DIFFERENT ENSEMBLE METHODS AND A SINGLE TREE IN TERMS OF AUC VALUE. THE AUC VALUE FOR THE ENSEMBLE METHODS ARE THE AVERAGE ACROSS DIFFERENT ENSEMBLE SIZES.

Algorithms	Diabetes	Heart	Leukemia	Liver
J48 (single tree)	0.576	0.711	0.806	0.883
Mean of Bagging	0.604	0.760	0.912	0.912
Mean of Boosting	0.579	0.766	0.918	0.916
Mean of SSO^{E}-PSO	0.645	0.779	0.930	0.925

The improvement of ensemble classifier compared to single classifier is often attributed to the diversity of individual classifiers [22]. To investigate whether SSO^{E}-PSO produced base classifiers are identical, we decomposed the ensemble classifiers and evaluate the performance of each base classifier. Figure 6 shows the AUC values of individual base classifiers (J48) and their ensemble create from SSO^{E}-PSO. The results are divided with respect to the dataset and then further divided with respect to the external 5-fold cross-validation partition. The diversity of the base classifiers are indicated by their varied performance and the ensemble classifiers are generally more accurate compared to most of the individual base classifiers.

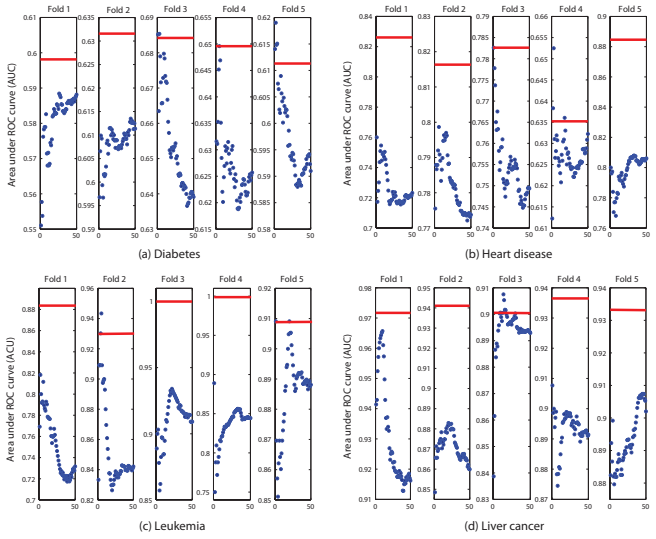


Fig. 6. Ensemble decomposition. Comparison of AUC values of each base classifiers in SSO^{E}-PSO (denoted as blue points) with the AUC value of the majority voting of base classifiers (denoted as a red line). For each dataset, the result is divided with respect to the fold of an external 5-fold cross validation.

The Friedman rank-sum test was applied to measure the performance statistically. Since the performance of the ensemble models with different ensemble sizes may not follow any symmetric distribution, we used the Friedman rank-sum test as it transfers the AUC values into ranks for each ensemble method and perform the comparison based on the ranks only. Table V summarizes the performance of SSO^{E}-PSO compared to Bagging and Boosting, respectively. For diabetes, heart disease, and liver cancer datasets, the improvement was significant at α=0.05, the commonly used significance level. The performance on the Leukemia dataset has a marginal significance (α=0.1). Overall, the results indicate that SSO-based ensemble performs statistically better than Bagging and Boosting ensembles.

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT ENSEMBLE METHODS IN TERMS OF FRIEDMAN RANK-SUM TEST

Algorithms	Diabetes	Heart	Leukemia	Liver
SSO^{E}-PSO vs. Bagging	$p\approx$0.01	$p\approx$0.01	$p\approx$0.1	$p\approx$0.01
SSO^{E}-PSO vs. Boosting	$p\approx$0.01	$p\approx$0.01	$p\approx$0.1	$p\approx$0.01

For further assessment of SSO on sample selection, we extracted the 50% most frequently selected samples from Leukemia dataset while employing SSO^{E}-PSO for building an ensemble classifier of 100 base classifiers. Figure 7a shows the hierarchical clustering of the 50% most frequently selected samples using the top 100 genes filtered by BSS/WSS. By cutting the tree dendrogram into three clusters, we observe two heterogynous acute myeloid leukemia (AML) clusters and one acute lymphoblastic leukemia (ALL) cluster. The Leukemia dataset contains 25 AML samples in which 24 were frequently selected by SSO and were clustered correctly by allowing them to be grouped into two separated clusters. There are 45 ALL samples in the initial dataset from which only 10 were the most frequently selected ones (at the 50% cutoff) and 9 of them were clustered into a single cluster. Compared to the clustering of the entire dataset (Figure 7b), only 17 AML samples were clustered correctly under the same conditions. The other 8 AML samples (in grey color) were clustered with ALL samples and are therefore indistinguishable from ALL samples.

This result indicates that (1) almost all AML samples are important according to SSO procedure whereas only a small subset of ALL samples is sufficient to define the class profile; (2) there are two types of AML samples and they are best distinguished with ALL samples by including a subset of most representative ALL samples; and (3) not all ALL samples are informative and several of them may contain high level of data noise.

VI. CONCLUSION AND FUTURE WORK

In this paper, we extended the SSO-based technique for imbalanced data sampling and ensemble learning. We evaluated the proposed methods on simulation studies and several key bioinformatics applications. The improvement over many traditional methods were confirmed statistically. We also demonstrated that SSO-based techniques are able to select the most useful samples. In bioinformatics applications, these are often the most representative samples for a type of disease or phenotype under study. Therefore, the interpretation of these most frequently selected samples could have important biological implications. This has been illustrated in our analysis of Leukemia dataset where most of AML samples were very frequently selected by SSO procedure and were clustered into two unique clusters.

Future work include (1) the analysis of ensemble diversity; (2) the application of SSO to simultaneous feature and sample selection; (3) learning in multiclass imbalanced data; and (4)

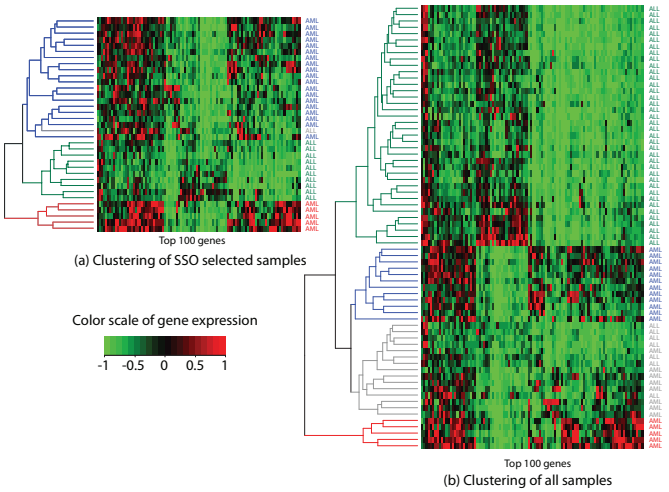


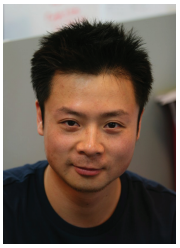
Fig. 7. Hierarchical clustering of (a) 50% most frequently selected samples in SSO procedure and (b) all samples from Leukemia dataset, using the top 100 genes filtered by BSS/WSS. The tree dendrograms are cut at the level of three branches to form three separated clusters marked by blue (first cluster of AML), red (second cluster of AML), and green (ALL cluster). Samples that are clustered incorrectly (or indistinguishable) at the level of three branches are marked in gray.

integrating SSO for ensemble-based approach in imbalanced learning. First, it is known that GA and PSO may produce highly correlated solutions which in our ensemble application will result in base classifiers with identical predictions. Therefore, in would be useful to explicitly measure ensemble diversity in fitness function. Second, current applications of SSO are limited to identify important samples only. By modifying the algorithm to incorporate feature selections into the optimization procedure, we may identify the most important samples conditioned on a unique subset of features. This could be useful in classifying high-dimensional datasets such as those generated from microarray and proteomics studies, where the identification of molecular disease markers are important for defining disease phenotypes. Third, ensemble learning has been demonstrated to be effective in learning from multiclass imbalanced data [39]. While only binary-class datasets were considered in current experiment, the minimization of cross-validation error and the application of SSO could be extended to multiclass datasets. Lastly, recent research shows that ensemble learning is an effective approach for class imbalanced problem [40]. In current study, we treated them as separate applications for SSO. Nevertheless, it is interesting to incorporate SSO for simultaneous ensemble and imbalanced learning. These potential applications will be pursued in our future work.

REFERENCES

[1] H. He and E. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.
[2] T. Dietterich, "Ensemble methods in machine learning," in *Proceedings of the First International Workshop on Multiple Classifier Systems*. Springer-Verlag, 2000, pp. 1–15.
[3] P. Yang, Z. Zhang, B. Zhou, and A. Zomaya, "Sample subset optimization for classifying imbalanced biological data," in *Proceedings of the 15th PAKDD*, 2011, pp. 333–344.
[4] I. Meyer, "A practical guide to the art of rna gene prediction," *Briefings in Bioinformatics*, vol. 8, no. 6, pp. 396–414, 2007.
[5] J. Zeng, S. Zhu, and H. Yan, "Towards accurate human promoter recognition: a review of currently used sequence features and classification methods," *Briefings in Bioinformatics*, vol. 10, no. 5, pp. 498–508, 2009.
[6] S. Sonnenburg, G. Schweikert, P. Philips, J. Behr, and G. Rätsch, "Accurate splice site prediction using support vector machines," *BMC Bioinformatics*, vol. 8, no. Suppl 10, p. S7, 2007.
[7] R. Blagus and L. Lusa, "Class prediction for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 11, no. 1, p. 523, 2010.
[8] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proceedings of the 15th ECML*, 2004, pp. 39–50.
[9] W. Liu and S. Chawla, "Class confidence weighted knn algorithms for imbalanced data sets," in *Proceedings of the 15th PAKDD*, 2011, pp. 345–356.
[10] N. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.
[11] X. Liu, J. Wu, and Z. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst. Man Cybern. Part B-Cybern.*, vol. 39, no. 2, pp. 539–550, 2009.
[12] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[13] S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," *Evolutionary Computation*, vol. 17, no. 3, pp. 275–306, 2009.
[14] K. Kim and S. Cho, "An evolutionary algorithm approach to optimal ensemble classifiers for dna microarray data analysis," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 377–388, 2008.
[15] D. Che, Q. Liu, K. Rasheed, and X. Tao, "Decision tree and ensemble learning algorithms with their applications in bioinformatics," *Software Tools and Algorithms for Biological Systems*, pp. 191–199, 2011.
[16] U. Braga-Neto and E. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.
[17] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
[18] R. Schapire, Y. Freund, P. Bartlett, and W. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.
[19] M. Islam, X. Yao, S. Shahriar Nirjon, M. Islam, and K. Murase, "Bagging and boosting negatively correlated neural networks," *IEEE Trans. Syst. Man Cybern. Part B-Cybern.*, vol. 38, no. 3, pp. 771–784, 2008.
[20] D. Nam, S. Yoon, and J. Kim, "Ensemble learning of genetic networks from time-series expression data," *Bioinformatics*, vol. 23, no. 23, pp. 3225–3231, 2007.
[21] Y. Peng, "A novel ensemble machine learning for robust microarray data classification," *Computers in Biology and Medicine*, vol. 36, no. 6, pp. 553–573, 2006.
[22] L. Kuncheva and C. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.
[23] Y. del Valle, G. Venayagamoorthy, S. Mohagheghi, J. Hernandez, and R. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.
[24] R. Batuwita and V. Palade, "micropred: effective classification of pre-mirnas for human mirna gene prediction," *Bioinformatics*, vol. 25, no. 8, pp. 989–995, 2009.
[25] M. Remmert, D. Linke, A. Lupas, and J. Söding, "Hhompprediction and classification of outer membrane proteins," *Nucleic Acids Research*, vol. 37, no. suppl 2, pp. W446–W451, 2009.
[26] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins," in *Proceedings of the 4th ISMB*. AAAI Press, 1996, pp. 109–115.
[27] T. Rani, S. Bhavani, and R. Bapi, "Analysis of e. coli promoter recognition problem in dinucleotide feature space," *Bioinformatics*, vol. 23, no. 5, pp. 582–588, 2007.
[28] C. Pan, J. Olsen, H. Daub, and M. Mann, "Global effects of kinase inhibitors on signaling networks revealed by quantitative phosphoproteomics," *Molecular & Cellular Proteomics*, vol. 8, no. 12, pp. 2796–2808, 2009.
[29] P. Hornbeck, I. Chabra, J. Kornhauser, E. Skrzypek, and B. Zhang, "Phosphosite: A bioinformatics resource dedicated to physiological protein phosphorylation," *Proteomics*, vol. 4, no. 6, pp. 1551–1561, 2004.

[30] J. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 1988, pp. 261–265.
[31] R. Detrano, A. Janosi *et al.*, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *The American Journal of Cardiology*, vol. 64, no. 5, pp. 304–310, 1989.
[32] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri *et al.*, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
[33] X. Chen, S. Cheung, S. So, S. Fan, C. Barry, J. Higgins, K. Lai, J. Ji, S. Dudoit, I. Ng *et al.*, "Gene expression patterns in human liver cancers," *Molecular Biology of the Cell*, vol. 13, no. 6, pp. 1929–1939, 2002.
[34] S. Dudoit, J. Fridlyand, and T. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.
[35] U. Ohler, G. Liao, H. Niemann, G. Rubin *et al.*, "Computational analysis of core promoters in the drosophila genome," *Genome Biology*, vol. 3, no. 12, pp. 81–87, 2002.
[36] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml
[37] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
[38] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.
[39] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 4, pp. 1119–1130, 2012.
[40] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 4, pp. 463–484, 2012.



Pengyi Yang received his PhD degree in Computer Science from The University of Sydney, Australia in 2012. He is currently a research follow in Systems Biology Group, Biostatistics Branch, National Institute of Environmental Health Sciences (NIEHS), National Institute of Health (NIH), USA. His research interests include the applications of machine learning algorithms and statistical modelings in computational and systems biology. This work was carried out mainly during his PhD candidature while he was funded by the NICTA International Postgraduate Award and the NICTA Research Project Award.

Paul D. Yoo is an Assistant Professor in the Department of Electrical and Computer Engineering, at Khalifa University of Science, Technology and Research (KUSTAR). He was a Research Scientist in the Centre for Distributed and High Performance Computing, at the University of Sydney from 2008 to 2009, and PHD Researcher (Quantitative Research) at the Capital Markets CRC initiated and administered by the Australia Federal Department for Education, Science and Training, from 2004 to 2008. His research is centered on the application of various computer science and mathematical methods to the discovery of the syntactic and semantic patterns in nucleic acid and amino acid sequences. This includes the development of new sequence pattern extraction tools and structural classification/prediction algorithms in 3D. In addition, he has carried out research in the application of many such methods ranging from the analysis of HIV gp120 glycosylation sites and the modeling of various post-translational modifications of a protein.

Juanita Fernando is the Academic Convenor BMedSc(Hons) with the Faculty of Medicine, Nursing and Health Sciences at Monash University. Chair of the Health Sub-Committee with the Australian Privacy Foundation and former Councillor with the Australasian College of Health Informatics, Dr Fernando's research concerns biomedical informatics, data exchange standards and information privacy and security. Dr Fernando has developed a particular emphasis on e-health and m-health tools and their contribution to workflow methodologies in the health sector.

Bing B. Zhou received the BS degree from Nanjing Institute of Technology, China and the PhD degree in Computer Science from Australian National University. He is currently an associate professor at the University of Sydney. His research interests include parallel/distributed computing, Grid and cloud computing, peer-to-peer systems, parallel algorithms, and bioinformatics. He has a number of publications in leading international jounals and conference proceedings. His research has been funded by the Australian Research Council through several Discovery Project grants.

Zili Zhang is a professor at Southwest University, Chongqing, China, and a senior lecturer at Deakin University, Australia. He received his BSc from Sichuan University, MEng from Harbin Institute of Technology, and PhD from Deakin University, all in computing. He authored or co-authored more than 100 refereed papers in international journals or conference proceedings, 1 monograph, and 4 textbooks. His research interests include agent-based computing, big data analysis, and agent-data mining interaction and integration. Contact him at zzhang@deakin.edu.au or zhangzl@swu.edu.cn.

Albert Y. Zomaya is currently the Chair Professor of High Performance Computing & Networking and Australian Research Council Professorial Fellow in the School of Information Technologies, The University of Sydney. He is also the Director of the Centre for Distributed and High Performance Computing which was established in late 2009. He is the author/coauthor of seven books, more than 400 publications in technical journals and conferences, and the editor of 14 books and 17 conference volumes. He is currently the Editor in Chief of the IEEE Trans. on Computers and serves as an associate editor for 19 journals including some of the leading journals in the field. Professor Zomaya was the Chair the IEEE Technical Committee on Parallel Processing (19992003) and currently serves on its executive committee. He also serves on the advisory board of the IEEE Technical Committee on Scalable Computing, the advisory board of the Machine Intelligence Research Labs. Professor Zomaya served as General and Program Chair for more than 60 events and served on the committees of more than 600 ACM and IEEE conferences. Professor Zomaya is a Fellow of the IEEE, AAAS, the Institution of Engineering and Technology (U.K.), a Distinguished Engineer of the ACM and a Chartered Engineer (CEng). He received the 1997 Edgeworth David Medal from the Royal Society of New South Wales for outstanding contributions to Australian Science. He is also the recipient of the IEEE Computer Societys Meritorious Service Award and Golden Core Recognition in 2000 and 2006, respectively. Also, he received the IEEE TCPP Outstanding Service Award and the IEEE TCSC Medal for Excellence in Scalable Computing, both in 2011. His research interests are in the areas of parallel and distributed computing and complex systems.