Neurocomputing 101 (2013) 309–318

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data

Hualong Yu [a,*], Jun Ni [b], Jing Zhao [c]

[a] School of Computer Science and Engineering, Jiangsu University of Science and Technology, Mengxi Road No.2, Zhenjiang 212003, China
[b] Department of Radiology, Carver College of Medicine, The University of Iowa, Iowa City, IA 52242, USA
[c] College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

ARTICLE INFO

Article history:
Received 25 December 2011
Received in revised form
25 August 2012
Accepted 26 August 2012
Communicated by T. Heskes
Available online 19 September 2012

Keywords:
DNA microarray
Ant colony optimization
Class imbalance
Undersampling
Support vector machine

ABSTRACT

In DNA microarray data, class imbalance problem occurs frequently, causing poor prediction performance for minority classes. Moreover, its other features, such as high-dimension, small sample, high noise etc., intensify this damage. In this study, we propose ACOSampling that is a novel undersampling method based on the idea of ant colony optimization (ACO) to address this problem. The algorithm starts with feature selection technology to eliminate noisy genes in data. Then we randomly and repeatedly divided the original training set into two groups: training set and validation set. In each division, one modified ACO algorithm as a variant of our previous work is conducted to filter less informative majority samples and search the corresponding optimal training sample subset. At last, the statistical results from all local optimal training sample subsets are given in the form of frequence list, where each frequence indicates the importance of the corresponding majority sample. We only extracted those high frequency ones and combined them with all minority samples to construct the final balanced training set. We evaluated the method on four benchmark skewed DNA microarray datasets by support vector machine (SVM) classifier, showing that the proposed method outperforms many other sampling approaches, which indicates its superiority.

1. Introduction

In the past decade, DNA microarray has been one of the most important molecular biology technologies in the post-genomic era. By this technology, biologists and medical experts are permitted to detect the activity of thousands of genes in a cell simultaneously. At present, DNA microarray has been widely applied to predict gene functions [1], investigate gene regulatory mechanisms [2,3], provide invaluable information for drug discovery [4], classify for cancer [5,6] and mining new subtypes of a specific tumor [7–9] etc. Among these applications, cancer classification has attracted more attentions. However, it is well-known that microarray data generally has some particular features, such as high-dimension, small sample, high noise and most importantly, imbalanced class distributions. Skewed class distributions will underestimate greatly the prediction performance for minority classes and provide inaccurate evaluation for classification performance, while the other features of microarray data will further intensify this damage [10]. Therefore, it is necessary to remedy this bias by some effective strategies.

In fact, class imbalance learning has drawn a significant amount of interest since 2000 from artificial intelligence, data mining and machine learning, which can be reflected by launch of several major workshops and special issues [11], including AAAI'00 [12], ICML'03 [13] and ACM SIGKDD Explorations'04 [14] etc. There are two major methods to solve class imbalance problem: sampling-based strategy and cost sensitive learning. Sampling, which includes oversampling and undersampling, deals with class imbalance by inserting samples for minority class or discarding samples of majority class [15–20]. While cost-sensitive learning treats class imbalance by incurring different costs for different classes [21–29]. Recently, some research also focused on ensemble learning built on multiple different sampling or weighting data sets with presenting excellent performance and generalization ability [30–35]. More details about class imbalance learning methods are presented in Section 2.

In this study, we introduce a novel undersampling method based on the idea of ant colony optimization (ACO), which is named ACOSampling, to classify for skewed DNA microarray data. In fact, this method is a modified version of our previous work [36], the difference between them is this work converts the information selection from feature space to sample space. First, the original training dataset is randomly and repeatedly divided into two groups: training dataset and validation dataset. Then for

* Corresponding author. Tel.: +86 511 88690470; fax.: +86 511 88690471.
E-mail address: yuhualong@just.edu.cn (H. Yu).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.08.018

each partition, ACOSampling is conducted to find the corresponding optimal majority class sample subset. Different from the traditional ACO algorithm, ACOSampling impels ants to leave from the nest, then to pass all majority class samples one by one, by either pathway 0 or pathway 1, at last to reach the food source, where pathway 0 indicates the corresponding sample is useless and should be filtered, while pathway 1 represents it is important and should be selected. Considering the particularity of the classification tasks in this study, the overall accuracy is not an excellent measure as the fitness function, thus we construct it by three weighted indicative metrics, namely *F-measure*, *G-mean* and *AUC*, respectively. After that, many local optimal majority class sample subsets can be generated by iterative partitions, so the significance of each majority sample may be estimated according to its selection frequence, i.e., the higher the selection frequence, the more information the corresponding sample can provide. Next, a global optimum balanced sample set can be created by combining the highly ranked samples of majority class with all examples of minority class. At last, we construct a SVM classifier upon the balanced training set for recognizing future unlabeled samples.

The remainder of this paper is organized as follows. Section 2 reviews some previous work related with class imbalance problem. In Section 3, the idea and procedure of ACOSampling method is described in detail. Experimental results and discussions are presented in Section 4. At last, we conclude this paper in Section 5.

2. Previous work

As mentioned in the Section 1, the existing class imbalance learning methods could be roughly categorized into two major groups: sampling strategy and cost sensitive learning. Here, we pay special attention to sampling strategy because it is more related with our study.

The sampling is actually a re-balancing process for the given imbalanced data set. It can be distinguished into oversampling and undersampling. Oversampling, as its name indicates, increases some samples belonging to minority class, while undersampling takes away some examples of majority class. The simplest sampling methods are Random Over Sampling (ROS) and Random Under Sampling (RUS) [15]. The former will make the learner to be overfitting by simply duplicating some samples of minority class, while the latter may lose some valuable classification information due to many majority examples are randomly removed [11]. To overcome their drawbacks, some complicated sampling methods were developed. Synthetic Minority Over-sampling TEchnique (SMOTE), proposed by Chawla et al. [16], can create artificial data based on the feature space similarities between existing minority examples. Specifically, randomly select one sample x_i in minority class, find its K-nearest neighbors belonging to the same class by Euclidian distance. To create a synthetic sample, randomly select one of the K-nearest neighbors, then multiply the corresponding feature vector difference with a random number between [0,1], and finally, add this vector to x_i. Han et al. [17] observed that most misclassified samples scatter around the borderline between two categories, then presented two improved versions of SMOTE, Borderline-SmOte1 (BSO1) and Borderline-SmOte2 (BSO2), respectively. For BSO1, SMOTE only runs on those minority class samples near borderline, while for BSO2, it generates synthetic minority class samples between each frontier minority example and one of its K-nearest neighbors belonging to majority class, thus mildly enlarges decision region of minority class. One Side Selection (OSS) has very similar idea with BSO2. It shrinks the decision area of majority class by cleaning noisy samples, redundant samples and boundary examples in majority category [18]. As another improved oversampling method, Adaptive Synthetic Sampling (ADA-SYN) uses a density distribution as criterion to automatically decide the number of synthetic samples that need to be generated for each minority example by adaptively changing the weights of different minority class examples to compensate for the skewed distributions [19]. Another sampling method using density distribution is Under-sampling based on clustering (SBC), presented recently by Yen and Lee [20]. SBC may automatically decide to remove how many majority class samples in each cluster, according to the corresponding density distribution. García *et al.* [37] have simply compared two kinds of sampling strategies and found oversampling generally produces better classification performance when the dataset is highly skewed, while undersampling is more effective when imbalance ratio is very low. All in all, sampling possess many advantages, such as simple, intuitive, low time complexity and low storage cost, thus it can be more convenient to apply in real-world imbalanced classification tasks. In Section 4, we would investigate the performance of the proposed ACOSampling method compared with original data without sampling (ORI) and several benchmark sampling strategies described above, such as ROS, RUS, SMOTE, BSO1, BSO2, OSS, ADA-SYN and SBC.

Cost sensitive learning methods consider the costs associated with misclassifying samples [21]. Instead of creating balanced data distributions through sampling, cost-sensitive learning assigns different costs for the samples belonging to different classes by creating a cost matrix. Based on the cost matrix, misclassifications on the minority class are more expensive than the majority class. Moreover, cost sensitive learning pursues to minimize the total cost but not error rate, thus the significance of minority class is highlighted. There are generally three kinds of cost sensitive learning methods. The first one is based on translation theorem [22]. It applies misclassification costs to the data set in terms of data-space weighting. The second class, building on metacost framework [23], uses cost-minimizing techniques to the combination schemes of ensemble methods. Some existing research has combined these two strategies, such as AdaCX series algorithms [24] and AdaCost [25]. The last class of cost sensitive learning methods directly designs appropriate cost functions for specific classifier, including cost-sensitive decision tree [26], cost-sensitive neural network [27] and cost-sensitive support vector machine [28] etc. In some application fields, it has been demonstrated that cost sensitive learning is superior to sampling approaches [29]. However, it is difficult to pre-design an appropriate cost function when class imbalance problem occurs [27].

In recent several years, ensemble learning has also become popular to be employed for solving class imbalance problems. Generally speaking, in this technology, ensemble learning framework is incorporated with sampling approach or weighting strategy to acquire better classification performance and generalization capability. Chawla et al. introduced SMOTE into Boosting ensemble learning framework to develop the SMOTEBoost learning method [30]. Unlike the base classifiers generation strategy in traditional Boosting, SMOTEBoost promotes weak classifiers through altering distributions for the samples of different classes by SMOTE. Liu et al. combined RUS and AdaBoost classifier to overcome deficiency of information loss of traditional RUS method and presented two ensemble strategies: EasyEnsemble and BalanceCascade [31]. In contrast with Boosting framework, Bagging seems to leave less room to be modified for class imbalance problem. However, there are still some improved versions about Bagging, including Asymmetric Bagging (asBagging) which has been used to retrieve image [32] and predict drug

molecules [33], Roughly Balanced Bagging (RB Bagging) based on negative binomial distributions [34] etc. In literature [35], Khoshgoftaar et al. compared the existing Boosting and Bagging technologies on noisy and imbalanced data and found Bagging series algorithms generally outperform Boosting. However, ensemble learning is more time-consuming than both former methods so that it is restricted in practical applications [35].

3. Methods

3.1. Undersampling based on ant colony optimization

Ant colony optimization (ACO) algorithm, which is developed by Colorni *et al.* [38], is one important member of swarm intelligence family. ACO simulates the behavior of foraging by real ant colony and in recent years, it has been successfully applied to solve various practical optimization problems, including TSP [39], parameter optimization [40], path planning [41], protein folding [42] etc.

In previous work, we have once designed an ACO algorithm to select tumor-related marker genes in DNA microarray data [36]. While in this study, we transform it from feature space to sample space to search an undersampling set which is regarded as the optimal subset estimated on the given validation set. However, this optimal set is not necessary absolutely balanced. In addition, in optimization process, to justly evaluate the performance for each ant, several indicative metrics have also been jointly used to construct the fitness function.

Fig. 1 describes the sample selection procedure using our ACO algorithm. As indicates in Fig. 1, the process of sample selection may be regarded as the procedure of seeking for food of one ant. Between nest and food, sites are built one by one, and each of them represents one alternative sample of majority class in original training set. In the process of moving from nest to food, ant passes each site by either pathway 0 or pathway 1, where pathway 0 denotes that the next sample will be filtered and pathway 1 represents that the next sample will be selected. At last, when the ant arrives at the food, some majority samples are extracted and combined with all minority examples to constitute the corresponding training set. A binary set {1, 0, 1, 0, 0, 1} means the 1st, 3rd and 6th majority sample have been picked out. Then the new created training set would be evaluated according to the fitness on the validation set. Ants cooperate with each other by intensity of pheromone left in every pathway to search the optimal routine.

In our ACO algorithm, many ants synchronously search pathways from nest to food. They select pathways according to the quantities of pheromone left in these pathways. The more pheromone is left, the more probability of the corresponding pathway is selected. We compute the probability of selecting a pathway by:

$$p_{ij} = \frac{\tau_{ij}}{\sum_j^k \tau_{ij}} \tag{1}$$

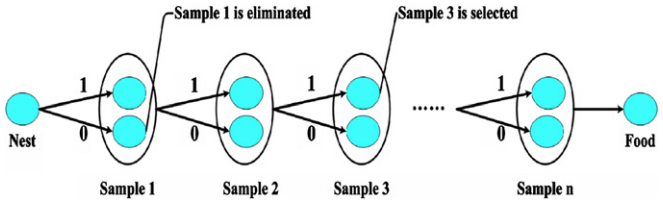


Fig. 1. Sample selection procedure based on ACO algorithm.

where i represents the ith site, i.e., the ith majority sample in original training set, j denotes pathway, which may be assigned as 1 or 0 to denote whether selecting the corresponding sample or not. τ_{ij} is pheromone intensity of the ith site in the jth pathway, p_{ij} and k are the probability of selecting the jth pathway of the ith site and possible value of pathway j (0 or 1), respectively. When an ant arrives at the food source, the corresponding sample subset will be evaluated by fitness function. It is worth noting that overall classification accuracy is not an indicative measure for imbalanced classification tasks [16]. Therefore, we use a special metric designed by Yang et al. [43] to evaluate classification performance, which is given in formula (2):

$$\begin{aligned} &\text{fitness} = \alpha \times \textit{F-measure} + \beta \times \textit{G-mean} + \gamma \times \textit{AUC} \\ &\text{s.t.}: \alpha + \beta + \gamma = 1 \end{aligned} \tag{2}$$

The fitness function is constitutive of three weighted metrics: *F-measure*, *G-mean* and *AUC*. We will introduce these metrics in Section 4.2. When one cycle finishes, the pheromone of all pathways is updated, the update function inherits from the literature [38] and is described as follows:

$$\tau_{ij}(t+1) = \rho \times \tau_{ij}(t) + \Delta\tau_{ij} \tag{3}$$

where$\rho$is the evaporation coefficient, which controls the decrement of pheromone, $\Delta\tau_{ij}$is increased pheromone of some excellent pathways. In this paper, we add pheromone in the pathways of the best 10% ants after each cycle and store these pathways in a set E. $\Delta\tau_{ij}$is defined as follows:

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{0.1 \times ant\_n} \times \text{fitness}, & \text{pathway}_{ij} \in E \\ 0, & \text{pathway}_{ij} \notin E \end{cases} \tag{4}$$

In formula (4), *ant_n* is the size of ant colony, i.e., the number of ants. When one cycle finishes, the pheromone of some pathways will be intensified and the others will be weakened, so that guaranteeing those excellent pathways are given more chances in next cycle. When convergence of ACO algorithm, all ants are inclined to select the same pathway. At last, the optimal solution returns.

In contrast with our previous work, we make a few changes in this study, such as fitness function and pheromone update

Input: Original training set: S, Validation set: V.
Process:
for i=1: number of majority samples in S
  for j=0:1
    Assign initial pheromone *ph_initial* for $pathway_{ij}$;
  end for
end for
Set the optimal solution OPS=0;
for i=1: iteration times of ant colony
  for j=1: size of ant colony *ant_n*
    Acquire sampling set SS_{ij} by formula (1);
    Train a classifier C_{ij} for SS_{ij};
    Evaluate performance of C_{ij} by V and formula (2);
  end for
  Find the optimal solution OPS_i in the *ith* iteration;
  if (OPS<OPS_i)
    OPS=OPS_i;
  end if
  Update performance for each pathway by formula (3) and (4);
end for
Output: Undersampling training set S' which corresponds to OPS

Fig. 2. Pseudo-code description of the undersampling algorithm based on ACO.

function. On the other hand, it also inherits some advantages from previous method, for example, we impose the upper and lower boundary of pheromone in each pathway to prevent the algorithm sinking into local optimization prematurely. Pseudo-code description of the algorithm is simply summarized in Fig. 2.

3.2. ACOSampling strategy

By ACO algorithm mentioned above, an excellent undersampling subset may be extracted as the final training set to construct a classifier and recognize future testing samples. However, to guide optimization procedure, we have to divide the original training set into two parts: training set and validation set, before ACO algorithm works. Generally, it can cause two severe problems for constructed classifier: information loss and overfitting due to the employment of validation set. In particular, when classification tasks are based on small sample set, these problems become more serious. To solve this problem, we present a novel strategy named as ACOSampling to produce more robust classifier by the combination of reduplicative partition of original sample set and ACO algorithm. The frame diagram of ACOSampling strategy presents in Fig. 3.

As can be observed from Fig. 3, in our design, ACOSampling applies 3-fold cross validation to evaluate classification performance, i.e., two-thirds samples are extracted into original training set and the rest ones are used for testing each time. In practical applications, it may be easily modified to fit various validation methods. Meanwhile, to impartially estimate the amount of information of each majority example and avoid overfitting for final generated classifier, the original training set is randomly and repeatedly divided into two groups: training set (2/3) and validation set (1/3) for 100 times. It is clear that in these 100 repeated partitions, each sample has equal chance to be picked into training set. Then we conduct ACO algorithm in each partition to find the corresponding optimal undersampling set and figure out majority class samples ranking frequence list (take Fig. 4 as an example, the more times one sample emerges, the more information the sample can provide for classification) based on these local optimal sets. Next, a balanced dataset is created by combining the highly ranked samples of majority class with all examples of minority class. At last, we train a classifier upon the balanced sample set and evaluate its performance by testing set. According to the descriptions above, main loop of ACOSampling method can be summarized by pseudo-code in Fig. 5.

3.3. Support vector machine

Support vector machine (SVM) introduced by Vapnik [44], is a valuable tool for solving pattern classification problem. In contrast with traditional classification methods, SVM possesses several prominent advantages as follows: (1) high generalization capability; (2) absence of local minima; (3) be suitable for small-sample

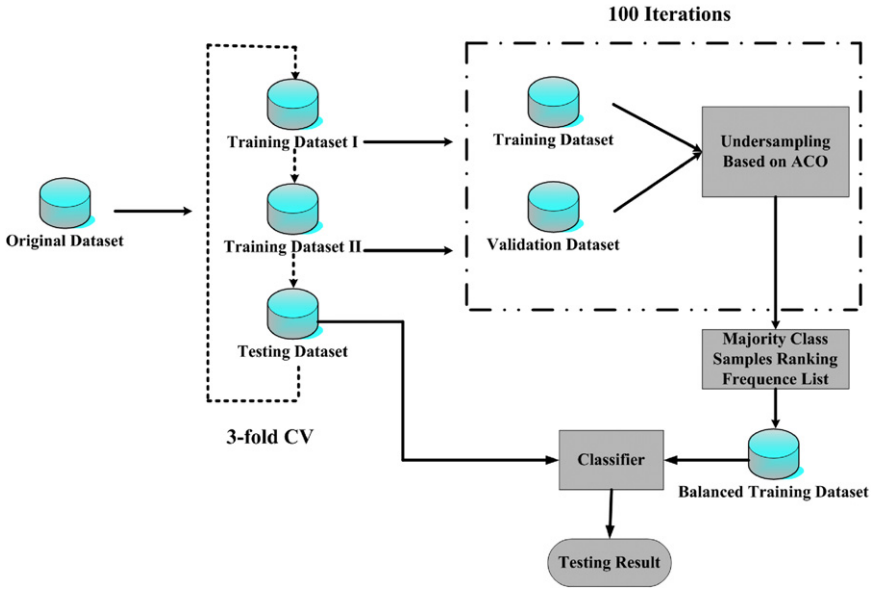
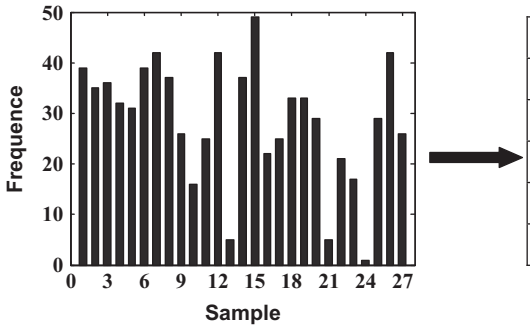


Fig. 3. The frame diagram of ACOSampling strategy.



Rank	Index	Frequence
1	15	49
2	7	42
…	…	…
26	21	4
27	24	1

Fig. 4. Ranking frequence list for samples of majority class.

Input: Initial training set: *IS*
Process:
for *i*=1:100 (iteration times)
Divide randomly *IS* into two sets: training set *S* and validation set *V*;
Run undersampling algorithm based on ACO (refers to Figure 2) to acquire *S'*;
Record majority class sample index of *S'* into REC_i;
end for
Compute emerging times for each majority example based on all records REC_1-REC_{100} and give the corresponding frequence list;
Rank all samples in descending order according to the frequence list;
Combine some highly ranked majority samples and all minority samples to construct a balanced training set: *BS*.
Output: Final training set: *BS*

Fig. 5. Pseudo-code description of ACOSampling strategy.

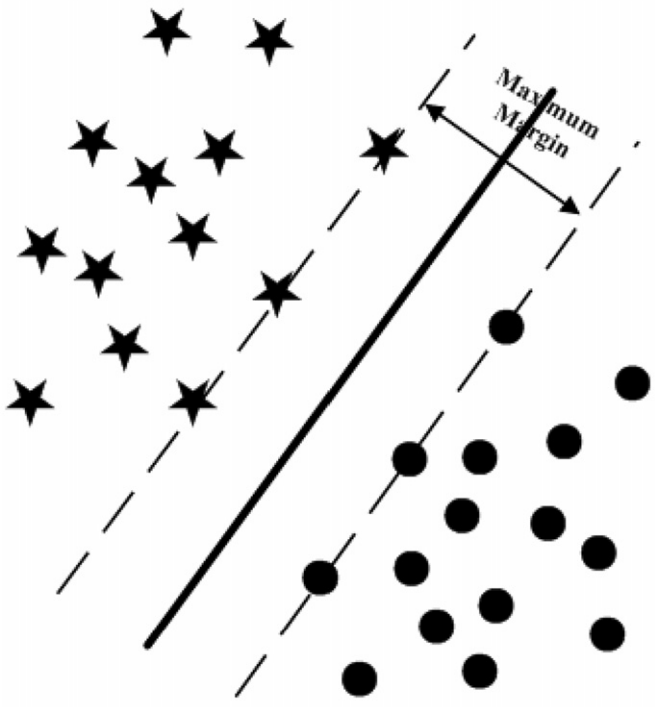


Fig. 6. SVM constructs a hyperplane (bold line) to maximize the margin between two classes (circle and pentagram). The samples emerging on the dashed lines are called as support vectors. New instances will be classified to the side of the hyperplane they fall into.

dataset. Its main idea is to implicitly map data to a higher dimensional space via a kernel function and then solve an optimization problem to identify the maximum-margin hyperplane that separates two class training instances. New instances are classified according to the side of the hyperplane they fall into (see Fig. 6).

Given training set $S=\{(x_i,y_i)|x_i\in R^d, y_i\in\{-1,+1\}, i=1,\cdots,N\}$, where$x_i$is a *d*-dimension sample, $y_i$is the corresponding class label, *N* is the number of samples. The discriminant function of SVM can be described as follows:

$$g(x)=\mathrm{sgn}\left(\sum_{i=1}^{sv}\alpha_i y_i K(x,x_i)+b\right) \tag{5}$$

In formula (5), *sv* represents the number of support vectors, α_i is lagrange multiplier, *b* is the bias of optimum classification hyperplane, while $K(x, x_i)$ denotes the kernel function. In this work, we conduct the experiments with radial basis kernel function (RBF) because it generally produces more excellent generalization performance and lower computational cost compared with other kernel functions in practical applications [45]. RBF kernel function is described as follows:

$$K(x_i,x_j)=\exp\left\{-\frac{|x_i-x_j|^2}{2\sigma^2}\right\} \tag{6}$$

The detailed description about the theory of SVM please refers to literature [44]. We choose SVM as baseline classifier because it generally provides better classification performance for high-dimensional and small sample data, e.g., DNA microarray data, than some typical classifiers.

3.4. Preprocessing and feature selection of DNA microarray data

Generally, genes exist with different value range in microarray datasets. To impartial evaluate the significance for each gene, it is necessary to conduct a data preprocessing procedure. In this paper, we normalized expression levels of each gene to be mean 0 and variance 1 [46]. The computational formula is given as:

$$g_{ij}'=\frac{g_{ij}-\mu_i}{s_i} \tag{7}$$

where g_{ij} and g_{ij}' represent original and normalized expression value of the *j*th sample on the *i*th gene, respectively. While μ_i and s_i are mean and variance for the *i*th gene in original dataset, respectively.

Moreover, in microarray datasets, there are lots of genes with noise, redundancy or irrelevant information for classification task. It is thus important for achieving excellent performance to select a few feature genes which are strongly related with classification task [47].

In recent years, a lot of feature gene selection strategies were proposed [7,36,46–49] and most of them have been proved helpful for improving classification accuracy. These strategies may be grouped into two classes: filter and wrapper. The former evaluates individually each gene and assigns a score reflecting its correlation with the class label according to certain criteria. Genes are then ranked based on their scores and some top-ranked ones are selected. While wrapper searches the optimal solution in gene space according to the returned accuracy percentage of a specific classifier. Generally speaking, wrapper could obtain better classification performance but expend more computational cost than filter [48]. For imbalanced microarray data classification task, researchers have designed some special and complicated feature gene extraction approaches [50–53]. However, the target of this study is to develop one more effective undersampling method, thus a simple and efficient strategy named as SNR (Signal-Noise Ratio) [7] is used. It is described as follows:

$$\mathrm{SNR}(i)=|\mu_{i1}-\mu_{i2}|/(s_{i1}+s_{i2}) \tag{8}$$

where μ_{i*} and s_{i*} stand for mean and standard deviation calculated by all samples belong to ∗class. Take colon dataset [5] as an example, we compute SNR value for all 2000 genes and rank them in ascending sequence (see Fig. 7).

Fig. 7 shows that only quite a few genes have high SNR values and they could be regarded as feature genes related closely with classification task. In this study, we select top-100 ranked genes to conduct experiments.

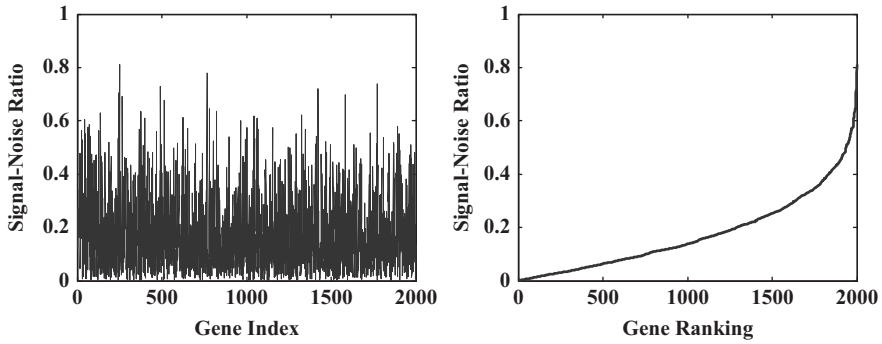


Fig. 7. SNR value distribution for gene index (left) and gene ranking (right).

Table 1
Datasets used in this study.

Dataset	Size	Genes	Maj:Min	Imbalance ratio
Colon [5]	62	2000	40:22	1.82
CNS [53]	60	7129	39:21	1.86
Lung [8]	39	2880	24:15	1.60
Glioma [9]	50	10367	36:14	2.57

4. Experiments

4.1. Datasets

Four benchmark imbalanced microarray datasets are used in our experiments, including Colon dataset [5], CNS (Central Neural System) dataset [54], Lung cancer dataset [8] and Glioma dataset [9]. The first three datasets are composed of binary class samples and Glioma dataset consists of four subtypes: cancer glioblastomas (CG), non-cancer glioblastomas (NG), cancer oligodendrogliomas (CO) and non-cancer oligodendrogliomas (NO). For Glioma dataset, CG is used as the positive class with 14 examples and the others are integrated into one class with 36 examples. In these four datasets, the size of samples is 39–62, the number of genes is from 2000 to 10367 and the imbalance ratio is 1.60–2.57. Information about these datasets is summarized in Table 1 and they are available at http://datam.i2r.a-star.edu.sg/datasets/krbd/.

4.2. Evaluation criteria and parameters settings

It is well-known that in skewed recognition tasks, overall accuracy (*Acc*) generally gives bias evaluation, thus some other specific evaluation metrics, such as *F-measure*, *G-mean* and area under the receiver operating characteristic curve (*AUC*), are needed to estimate classification performance of a learner. *F-measure* and *G-mean* may be regarded as functions of the confusion matrix as shown in Table 2. They are calculated as follows:

$$F\text{-}measure = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

$$G\text{-}mean = \sqrt{TPR \times TNR} \tag{10}$$

where Precision, Recall, *TPR* and *TNR* are further defined as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{11}$$

$$\text{Recall} = TPR = \frac{TP}{TP+FN} \tag{12}$$

Table 2
Confusion matrix.

	Predicted positive class	Predicted negative class
Actual positive class	*TP* (True positive)	*FN* (False negative)
Actual negative class	*FP* (False positive)	*TN* (True negative)

Table 3
Initial parameters settings.

Common parameters for ACOSampling	Value
ant_n: population size	50
ITA: iteration times of ant colony	50
ITP: iteration times of partition for original training set	100
dispose: evaporation coefficient	0.8
ph_initial: the initial pheromone in each pathway	1.0
ph_{min}: the lower boundary of pheromone	0.5
ph_{max}: the upper boundary of pheromone	2.0
α, β, γ: weight for three metrics	1/3
Common parameters for SVM	Value
σ: the parameter of RBF kernel function	5
C: the penalty factor	500

$$TNR = \frac{TN}{TN+FP} \tag{13}$$

and the overall accuracy *Acc* is computed as:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{14}$$

AUC is the area below the ROC curve which depicts the performance of a method using the (*FPR*, *TPR*) pairs. It has been proved to be a reliable performance measure for class imbalance problem [11].

In the study, some initial parameters used in ACOSampling and SVM have been given empirically according to our previous work [36] (see Table 3). As for several parameters such as ph_{min} and ph_{max}, we have made a little adjustment based on extensive experimental results.

4.3. Results and discussions

First, we conduct experiments on four imbalanced microarray datasets (refers to Section 4.1) with top-100 feature genes extracted by SNR strategy. To present superiority of the proposed method, the performance of some typical sampling approaches, such as original data without sampling (ORI), ROS, RUS, SMOTE, BSO1, BSO2, OSS, ADA-SYN and SBC etc., are tested synchronously.

Table 4
Performance comparison for various sampling methods on four datasets.

Performance (%)	Sampling method									
	ORI	*ROS*	*RUS*	*SMOTE*	*BSO1*	*BSO2*	*OSS*	*ADA-SYN*	*SBC*	*ACOSampling*
Colon dataset										
Acc	83.23 ± 2.72	84.19 ± 2.48	84.52 ± 3.47	85.48 ± 1.61	83.07 ± 1.94	84.03 ± 2.10	**85.65 ± 2.84**	**85.65 ± 2.10**	83.23 ± 2.41	85.63 ± 1.83
F-measure	75.24 ± 4.49	76.78 ± 4.78	79.31 ± 4.07	79.37 ± 2.85	74.99 ± 3.29	76.91 ± 3.08	**81.13 ± 2.96**	79.76 ± 3.29	78.95 ± 2.84	**81.13 ± 2.63**
G-mean	80.23 ± 3.76	81.54 ± 4.15	84.17 ± 3.22	83.83 ± 2.54	80.01 ± 2.78	81.68 ± 2.45	85.76 ± 2.29	84.21 ± 2.80	84.25 ± 2.47	**85.92 ± 2.41**
AUC	87.23 ± 2.32	87.76 ± 3.20	89.16 ± 1.88	89.13 ± 1.69	88.20 ± 1.51	88.61 ± 2.72	91.33 ± 1.89	88.82 ± 1.50	90.19 ± 2.19	**94.18 ± 1.56**
CNS dataset										
Acc	82.83 ± 1.98	83.33 ± 2.36	82.00 ± 1.00	84.33 ± 2.49	84.50 ± 2.48	84.33 ± 2.49	83.17 ± 2.29	**84.67 ± 2.21**	79.50 ± 1.98	83.83 ± 3.42
F-measure	75.50 ± 2.99	76.08 ± 3.29	77.45 ± 1.44	77.56 ± 3.66	78.13 ± 3.23	78.55 ± 3.75	77.58 ± 2.29	78.44 ± 3.32	76.10 ± 2.37	**79.75 ± 3.83**
G-mean	80.96 ± 2.44	81.32 ± 2.55	83.21 ± 1.31	82.51 ± 3.09	83.08 ± 2.56	83.79 ± 3.24	83.03 ± 1.69	83.43 ± 2.86	81.94 ± 2.12	**85.17 ± 3.23**
AUC	92.21 ± 1.94	92.26 ± 0.89	92.91 ± 1.47	93.05 ± 1.09	**93.36 ± 1.44**	93.22 ± 1.74	92.94 ± 1.24	92.81 ± 1.22	92.43 ± 1.08	93.33 ± 1.47
Lung dataset										
Acc	65.38 ± 3.29	64.62 ± 2.99	67.44 ± 3.25	65.90 ± 2.82	65.13 ± 2.86	67.44 ± 2.00	68.21 ± 3.08	65.38 ± 2.63	67.18 ± 2.99	**71.79 ± 4.59**
F-measure	56.10 ± 5.31	53.30 ± 4.10	60.56 ± 4.66	55.58 ± 5.35	55.40 ± 4.38	59.39 ± 2.30	62.50 ± 5.52	54.79 ± 3.53	60.43 ± 4.14	**67.86 ± 4.50**
G-mean	63.46 ± 4.24	61.48 ± 3.37	66.89 ± 3.82	63.35 ± 4.20	62.93 ± 3.52	66.16 ± 1.90	68.06 ± 4.18	62.67 ± 2.90	66.74 ± 3.45	**72.32 ± 4.19**
AUC	67.75 ± 2.52	67.36 ± 2.49	71.22 ± 2.33	67.92 ± 2.94	68.14 ± 2.90	69.78 ± 2.74	73.53 ± 3.56	68.00 ± 3.34	73.22 ± 2.06	**77.42 ± 4.16**
Glioma dataset										
Acc	92.80 ± 1.83	94.00 ± 0.89	92.20 ± 3.03	93.60 ± 1.20	94.00 ± 1.79	93.40 ± 1.28	93.20 ± 1.60	94.20 ± 1.40	93.40 ± 1.80	**94.40 ± 1.96**
F-measure	87.08 ± 3.54	89.35 ± 1.63	87.54 ± 4.01	88.56 ± 2.16	89.32 ± 3.08	88.80 ± 1.59	88.57 ± 2.39	89.77 ± 2.42	88.87 ± 2.73	**90.54 ± 3.00**
G-mean	90.94 ± 2.96	92.71 ± 1.53	93.16 ± 1.96	91.97 ± 1.76	92.47 ± 2.03	93.19 ± 0.78	93.26 ± 1.50	93.08 ± 1.69	93.44 ± 1.66	**94.32 ± 1.30**
AUC	98.71 ± 0.34	98.93 ± 0.18	98.75 ± 0.56	98.87 ± 0.47	**99.15 ± 0.27**	98.73 ± 0.37	98.75 ± 0.90	98.97 ± 0.36	98.77 ± 0.47	99.13 ± 0.16

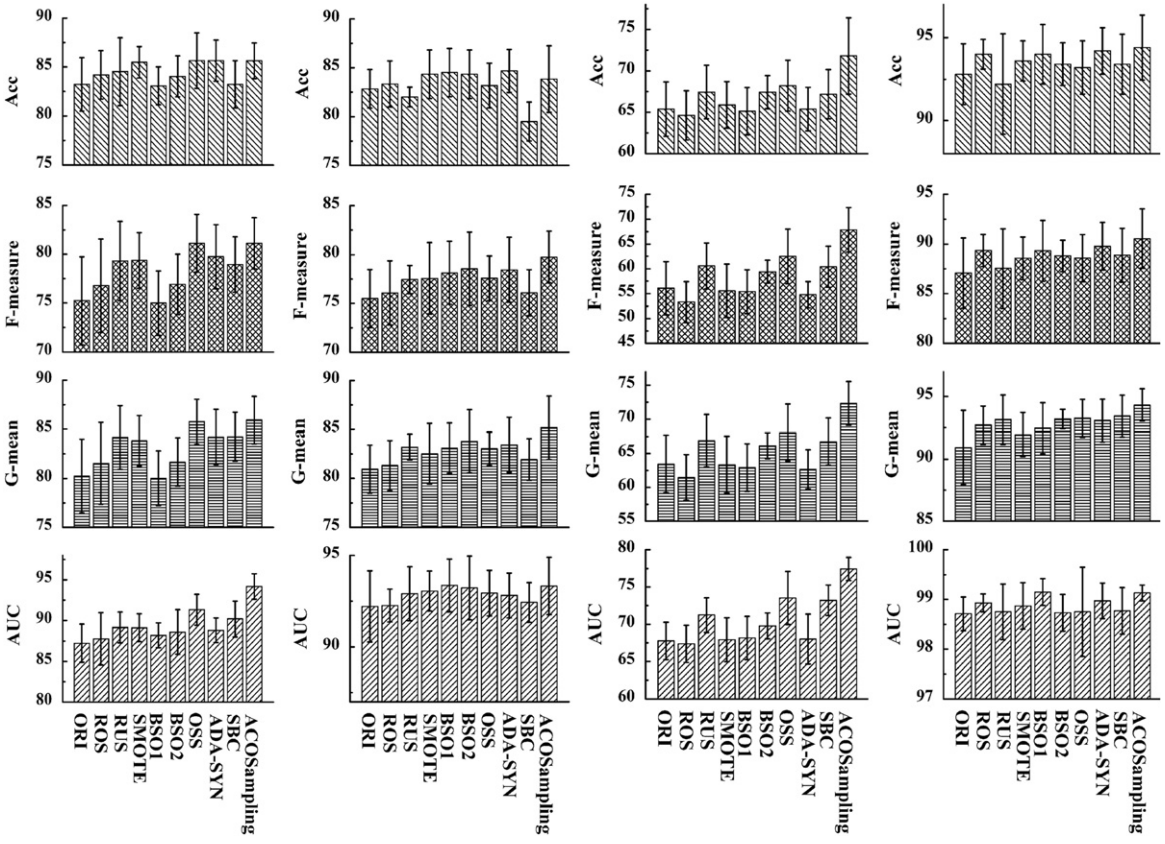


Fig. 8. Performance comparison for various sampling methods on four datasets. 1st column: Colon dataset; 2nd column: CNS dataset; 3rd column: Lung dataset; 4th column: Glioma dataset.

The average classification results based on 10 times' 3-fold cross validation are presented in Table 4 and Fig. 8.

As shown in Table 4 and Fig. 8, almost all sampling methods outperform the method only using original data (i.e., ORI), indicating that sampling is effective to improve classification performance for imbalanced high-dimensional and small sample classification tasks. This improvement not only reflects in some particularly designed assessment criteria for imbalanced classification, such as *F-measure*, *G-mean* and *AUC*, but also is embodied in the overall accuracy *Acc*.

Compared with those traditional sampling strategies, we are more interested in the performance of ACOSampling method. From Table 4 and Fig. 8, we observe that ACOSampling acquires the highest *F-measure* and *G-mean* in all datasets. For *AUC* metric,

ACOSampling attains the highest value on Colon dataset and Lung dataset, and it ranks only second to BSO1 on two other datasets. The results indicate that the proposed ACOSampling strategy is more effective and can extract some majority examples with more classification information than those typical sampling approaches. At the same time, we find an interesting phenomenon that the proposed method could obviously improve classification performance on Lung dataset, but only a slight improvement on the other datasets. This can be explained by the viewpoint of Ref. [31] which partitions class imbalance tasks into two groups: *harmful* and *unharmful*, according to the judgment of whether the classification performance suffers from serious degeneration or not. Undoubtedly, Lung dataset may be regarded as a clear *harmful* class imbalance task, and thus ACOSampling performs better on this dataset. However, we have to admit that ACOSampling is more time-consuming because it runs iteratively for estimating the significance of each majority sample. This could be well explained by "No Free Lunch Theorems" of Wolpert et al. [55], which demonstrates that there is no optimization method that outperforms all others in all circumstances.

Moreover, Fig. 8 shows that undersampling generally produces better results than oversampling on our low imbalance ratio datasets, which is similar with the finding of Ref. [37]. This

Table 5
Performance comparison for ACOSampling method based on different number of feature genes on four datasets.

Performance (%)	Number of feature genes						
	10	20	50	100	200	500	1000
Colon dataset							
Acc	82.74 ± 3.95	83.71 ± 2.10	84.84 ± 2.41	**85.63 ± 1.83**	84.20 ± 2.68	82.10 ± 3.91	75.32 ± 6.08
F-measure	77.27 ± 4.92	78.74 ± 2.66	79.50 ± 3.54	**81.13 ± 2.63**	78.61 ± 3.47	76.08 ± 4.15	69.42 ± 6.68
G-mean	82.59 ± 4.24	83.94 ± 2.16	84.35 ± 3.05	**85.92 ± 2.41**	83.58 ± 2.81	81.43 ± 3.30	75.80 ± 5.72
AUC	90.01 ± 3.30	90.31 ± 2.03	91.78 ± 2.15	**94.18 ± 1.56**	87.70 ± 1.41	87.35 ± 1.45	83.05 ± 3.52
CNS dataset							
Acc	69.33 ± 3.18	78.17 ± 4.44	81.67 ± 4.01	**83.83 ± 3.42**	79.50 ± 3.42	78.67 ± 2.96	73.83 ± 2.59
F-measure	63.47 ± 3.72	74.36 ± 4.06	77.23 ± 4.18	**79.75 ± 3.83**	73.17 ± 3.70	70.61 ± 2.53	61.43 ± 2.90
G-mean	70.65 ± 3.33	80.16 ± 3.77	82.91 ± 3.49	**85.17 ± 3.23**	79.35 ± 2.93	77.08 ± 1.99	69.55 ± 2.31
AUC	78.18 ± 3.93	89.27 ± 3.38	92.47 ± 1.66	**93.33 ± 1.47**	90.31 ± 1.69	84.58 ± 4.08	76.04 ± 1.58
Lung dataset							
Acc	68.21 ± 3.28	71.79 ± 3.80	68.20 ± 2.85	71.79 ± 4.59	**74.10 ± 1.80**	70.26 ± 4.47	68.20 ± 4.89
F-measure	63.10 ± 2.94	66.62 ± 4.13	65.26 ± 3.23	67.86 ± 4.50	**69.72 ± 3.14**	65.99 ± 5.42	63.85 ± 6.82
G-mean	68.43 ± 2.91	71.77 ± 3.53	69.29 ± 2.75	72.32 ± 4.19	**74.59 ± 2.43**	70.93 ± 4.71	68.70 ± 5.69
AUC	75.22 ± 4.63	79.78 ± 2.44	74.89 ± 4.32	77.42 ± 4.16	**80.22 ± 3.31**	73.56 ± 4.18	72.55 ± 3.74
Glioma dataset							
Acc	93.40 ± 2.01	**96.80 ± 1.83**	96.60 ± 2.69	94.40 ± 1.96	89.60 ± 2.50	88.40 ± 2.15	82.40 ± 1.96
F-measure	89.19 ± 3.06	**94.68 ± 2.93**	94.22 ± 4.45	90.54 ± 3.00	83.69 ± 3.28	82.80 ± 2.85	75.85 ± 2.41
G-mean	94.26 ± 1.83	**97.74 ± 1.30**	96.95 ± 3.04	94.32 ± 1.30	90.92 ± 1.69	91.40 ± 1.89	86.60 ± 2.01
AUC	99.20 ± 0.53	**99.86 ± 0.25**	99.68 ± 0.43	99.13 ± 0.16	98.08 ± 1.17	98.06 ± 1.27	91.03 ± 3.50

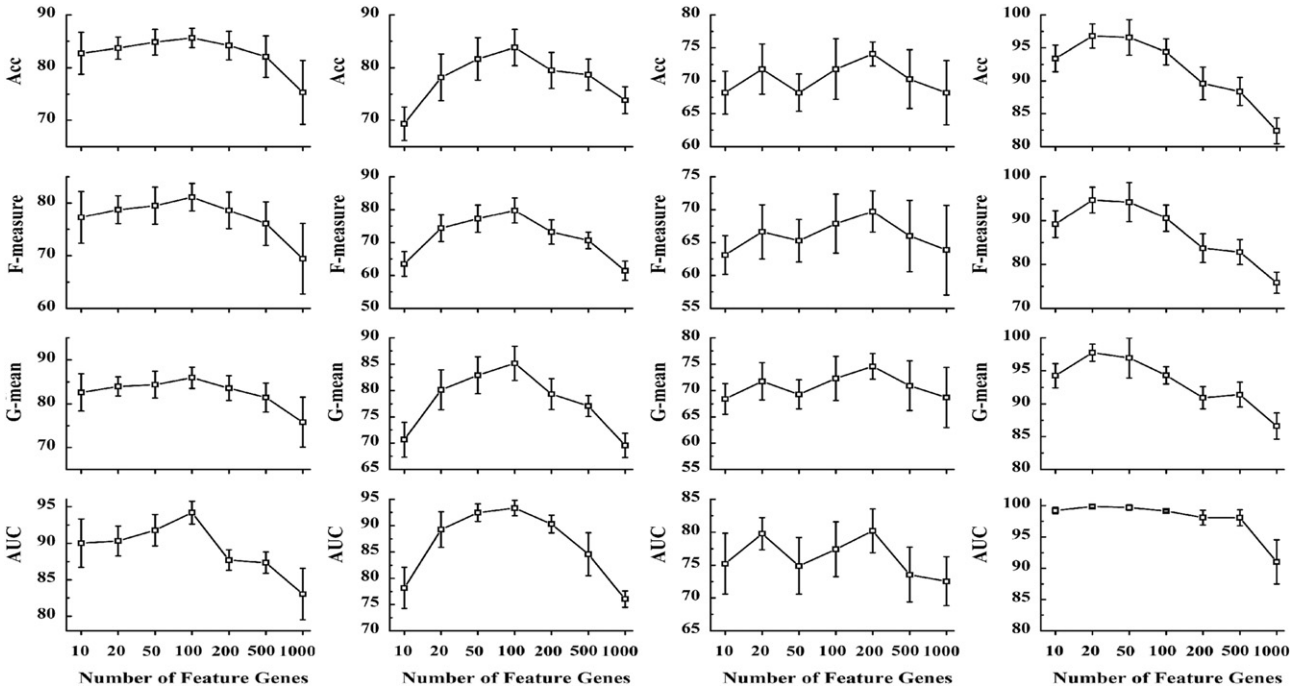


Fig. 9. Performance comparison for ACOSampling method based on different number of feature genes on four datasets. 1st column: Colon dataset; 2nd column: CNS dataset; 3rd column: Lung dataset; 4th column: Glioma dataset.

implies that using all majority class samples is not necessary when the dataset is a little skewed. In five oversampling methods, ROS performs worst in most cases, while the other four strategies show comparative performance with each other. In those undersampling methods exclusive of ACOSampling, OSS performs best in most datasets, while compared with RUS, SBC does not reveal enough competitive power. Therefore, in practical applications, if the time-complexity is limited strictly, OSS should be one considerable alternative.

Then we investigate the relationship between number of selected feature genes and classification performance for ACOSampling method. The number of feature genes is assigned as 10, 20, 50, 100, 200, 500 and 1000, respectively. We conduct 10 times' 3-fold cross validation in each group, then present the results in Table 5 and Fig. 9.

Though there are some fluctuations, a trend can be still observed from the curves in Fig. 9, i.e., the middle high and low on both sides. That means both of selecting too few or too many feature genes would degenerate classification performance. We believe the reason is the former causes the deficiency of useful information and the latter adds in much noise and redundant information. Table 5 shows that for Colon dataset and CNS dataset, the best performance can be obtained with 100 feature genes, while for Lung dataset and Glioma dataset, the optimal number are 200 and 20, respectively. Therefore, for high-dimensional and small sample classification tasks, for example, DNA microarray data, it is necessary to extract a few class-related features previously, which is also verified by Wasikowski et al. [56].

In contrast with the previous work by Yang et al. [43] which is similar with this study, our ACOSampling owns one specific merit: stronger generalization ability derived from 100 times' random partitions. Using these random partitions, we could give more righteous evaluation for the significance of each majority class sample. While Ref. [43] tries to avoid overfitting by integrating the results from multiple different kinds of classifiers, which would cause more bias for final classification results than our method. However, we have to admit that the proposed method in this study is more time-consuming than their work. Therefore, we declare that the proposed ACOSampling method is more suitable to deal with imbalanced classification tasks with the characteristic of small sample simultaneously.

5. Conclusions

In this paper, we present a novel heuristic undersampling method named as ACOSampling to address imbalanced DNA microarray data classification problem. By extensive experiments, it has demonstrated that the proposed method is effective and can automatically extract those so-called "information samples" from majority class. However, since its procedure of sampling is more time-consuming than those typical sampling approaches, it will be more efficient on small sample classification tasks.

Considering the excessive computational and storage cost of the proposed algorithm, we intend to improve its efficiency by modifying its formation rule in future work. We also expect that our ACOSampling can be applied to other real-world data mining applications, where we suffer from class imbalance. Moreover, considering ubiquitous multiclass imbalanced classification tasks in practical applications, we will investigate the possibility of extending current ACOSampling to multiclass tasks in the future work, too.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China under grant No.60873036, the Ph.D Programs Foundation for young researchers of Ministry of Education of China under Grant No.20070217051 and Ph.D Foundation of Jiangsu University of Science and Technology under Grant No.35301002.

References

[1] X. Zhou, M.C. Kao, W.H. Wong, From the cover: transitive functional annotation by shortest-path analysis of gene expression data, Proc. Nat. Acad. Sci. U.S.A. 99 (20) (2002) 12783–12788.
[2] D. Husmeier, Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks, Bioinformatics 19 (17) (2003) 2271–2282.
[3] E. Segal, M. Shapira, A. Regev, et al., Module networks: discovering regulatory modules and their condition specific regulators from gene expression data, Nat. Genet. 34 (2) (2003) 166–176.
[4] W.E. Evans, R.K. Guy, Gene expression as a drug discovery tool, Nat. Genet. 36 (3) (2004) 214–215.
[5] U. Alon, N. Barkai, D.A. Notterman, et al., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by Oligonucleotide array, Proc. Nat. Acad. Sci. U.S.A. 96 (12) (1999) 6745–6750.
[6] T.P. Conrads, M. Zhou, E.F. Petricoin, et al., Cancer diagnosis using proteomic patterns, Expert Rev. Mol. Diagn. 3 (4) (2003) 411–420.
[7] T.R. Golub, D.K. Slonim, P. Tamayo, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.
[8] D.A. Wigle, I. Jurisica, N. Radulovich, et al., Molecular profiling of non-small cell lung cancer and correlation with disease-free survival, Cancer Res. 62 (11) (2002) 3005–3008.
[9] C.L. Nutt, D.R. Mani, R.A. Betensky, et al., Gene expression-based classification of malignant gliomas correlates better with survival than histological classification, Cancer Res. 63 (7) (2003) 1602–1607.
[10] R. Blagus, L. Lusa, Class prediction for high-dimensional class-imbalanced data, BMC Bioinf. 11 (523) (2010).
[11] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[12] N. Japkowicz, Workshop on learning from imbalanced data sets, in: Proceedings of the 17th American Association for Artificial Intelligence, Austin, Texas, USA, 2000.
[13] N.V. Chawla, N. Japkowicz, A. Kolcz, Workshop on learning from imbalanced data sets II, in: Proceedings of the 20th International Conference of Machine Learning, Washington, USA, 2003.
[14] N.V. Chawla, N. Japkowicz, A. Kolcz, Editorial: special issue on learning from imbalanced data sets, ACM SIGKDD Explor. Newsl 6 (1) (2004) 1–6.
[15] C. Ling, C. Li, Data mining for direct marketing problems and solutions, in: Proceedings of the 4th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining, New York, USA, 1998, pp.73–79.
[16] N.V. Chawla, K.W. Bowyer, L.O. Hall, et al., SMOTE: synthetic minority oversampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357.
[17] H. Han, W.Y. Wang, B.H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: Proceedings of the 2005 International Conference of Intelligent Computing, Hefei, China, 2005, pp.878–887.
[18] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: Proceedings of the 14th International Conference of Machine Learning, Nashville, Tennessee, USA, 1997, pp.179–186.
[19] H. He, Y. Bai, E.A. Garcia, et al., ADASYN: adaptive synthetic sampling approach for imbalanced learning, in: Proceedings of the 2008 International Joint Conference of Neural Networks, Hong Kong, China, 2008, pp.1322–1328.
[20] S.J. Yen, Y.S. Lee, Cluster-based under-sampling approaches for imbalanced data distributions, Expert. Syst. Appl. 36 (3) (2009) 5718–5727.
[21] C. Elkan, The foundations of cost-sensitive learning, in: Proceedings of the 17th International Joint Conference of Artificial Intelligence, Seattle, Washington, USA, 2001, pp.973–978.
[22] B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by cost-proportionate example weighting, in: Proceedings of the 3rd International Conference of Data Mining, Melbourne, Florida, USA, 2003, pp.435–442.
[23] P. Domingos, MetaCost: a general method for making classifiers cost-sensitive, in: Proceedings of the 5th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining, San Diego, CA, USA 1999, PP.155–164.
[24] Y. Sun, M.S. Kamel, A.K.C. Wong, et al., Cost-sensitive boosting for classification of imbalanced data, Pattern Recognit. 40 (12) (2007) 3358–3378.
[25] W. Fan, S.J. Stolfo, J. Zhang, et al., AdaCost: misclassification cost-sensitive boosting, in: Proceedings of the 16th International Conference of Machine Learning, Bled, Slovenia, 1999, pp.97–105.
[26] C. Drummond, R.C. Holte, Exploiting the cost (In)sensitivity of decision tree splitting criteria, in: Proceedings of the 17th International Conference of Machine Learning, Stanford, CA, USA, 2000, pp.239–246.
[27] Z.H. Zhou, X.Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Trans. Knowl. Data Eng. 18 (1) (2006) 63–77.

[28] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced data sets, in: Proceedings of the 15th European Conference on Machine Learning, Pisa, Italy, 2004, pp.39–50.
[29] X.Y. Liu, Z.H. Zhou, The influence of class imbalance on cost-sensitive learning: an empirical study, in: Proceedings of the 6th IEEE International Conference on Data Mining, Hong Kong, China, 2006, pp.970–974.
[30] N.V. Chawla, A. Lazarevic, L.O. Hall, et al., SMOTEBoost: improving prediction of the minority class in boosting, in: Proceedings of the 7th European Conference on Principles of Data Mining and Knowledge Discovery, Cavtat–Dubrovnik, Croatia, 2003, pp.107–119.
[31] X.Y. Liu, J. Wu, Z.H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern. Part B Cybern. 39 (2) (2009) 539–550.
[32] D. Tao, X. Tang, X. Li, et al., Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 28 (7) (2006) 1088–1099.
[33] G.Z. Li, H.H. Meng, W.C. Lu, et al., Asymmetric bagging and feature selection for activities prediction of drug molecules, BMC Bioinf. 9 (S6) (2008) S7.
[34] S. Hido, H. Kashima, Y. Takahashi, Roughly balanced bagging for imbalanced data, Stat. Anal. Data Min. 2 (5-6) (2009) 412–426.
[35] T.M. Khoshgoftaar, J.V. Hulse, A. Napolitano, Comparing boosting and bagging techniques with noisy and imbalanced data, IEEE Trans. Syst. Man Cybern. Part B Cybern. 41 (3) (2011) 552–568.
[36] H.L. Yu, G.C. Gu, H.B. Liu, et al., A modified ant colony optimization algorithm for tumor marker gene selection, Genomics Proteomics Bioinf. 7 (4) (2009) 200–208.
[37] V. García, J.S. Sánchez, R.A. Mollineda, et al., The Class Imbalance Problem in Pattern Classification and Learning, in: II Congreso Español de Informática, 2007, pp. 283–291.
[38] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies. in: Proceedings of the 1st European Conference on Artificial Life, Paris, France, 1991, pp.134–142.
[39] A. Uğur, D. Aydin, An interactive simulation and analysis software for solving TSP using ant colony optimization algorithms, Adv. Eng. Software 40 (5) (2009) 341–349.
[40] X. Zhang, X. Chen, Z. He, An ACO-based algorithm for parameter optimization of support vector machines, Expert. Syst. Appl. 37 (9) (2010) 6618–6628.
[41] H. Duan, Y. Yu, X. Zhang, et al., Three-dimension path planning for UCAV using hybrid meta-heuristic ACO-DE algorithm, Simul. Modell. Pract. Theory 18 (8) (2010) 1104–1115.
[42] A. Shmygelska, H.H. Hoos, An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem, BMC Bioinf. 6 (30) (2005).
[43] P. Yang, L. Xu, B. Zhou, et al., A particle swarm based hybrid system for imbalanced medical data sampling, BMC Genomics 10 (S3) (2009) S34.
[44] V. Vapnik, Statistical Learning Theory, Wiley Publishers, New York, USA, 1998.
[45] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[46] Y.H. Wang, F.S. Makedon, J.C. Ford, et al., HykGene: a hybrid approach for selecting feature genes for phenotype classification using microarray gene expression data, Bioinformatics 21 (8) (2005) 1530–1537.
[47] E. Xing, M. Jordan, R. Karp, Feature selection for high-dimensional genomic microarray data, in: Proceedings of the 18th International Conference of Machine Learning, Williamstown, MA, USA, 2001, pp.601–608.
[48] I. Inza, P. Larranaga, R. Blanco, Filter versus wrapper gene selection approaches in DNA microarray domains, Artif. Intell. Med. 31 (2) (2004) 91–104.
[49] J.H. Chiang, S.H. Ho, A combination of rough-based feature selection and RBF neural network for classification using gene expression data, IEEE Trans. Nanobiosci. 7 (1) (2008) 91–99.
[50] K. Yang, Z. Cai, J. Li, et al., A stable gene selection in microarray data analysis, BMC Bioinf. 7 (228) (2006).
[51] G.Z. Li, H.H. Meng, J. Ni, Embedded gene selection for imbalanced microarray data analysis, in: Proceedings of the 3rd International Multi-symposiums on Computer and Computational Sciences, Shanghai, China, 2008, pp.17–24.
[52] A.H.M. Kamal, X.Q. Zhu, R. Narayanan, Gene selection for microarray expression data with imbalanced sample distributions, in: Proceedings of the 2009 International Conference of Bioinformatics, Systems Biology and Intelligent Computing, Shanghai, China, 2009, pp.3–9.
[53] Q. Shen, Z. Mei, B.X. Ye, Simultaneous genes and training samples selection by modified particle swarm optimization for gene expression data classification, Comput. Biol. Med. 39 (7) (2009) 646–649.
[54] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, et al., Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (6870) (2002) 436–442.
[55] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[56] M. Wasikowski, X.W. Chen, Combating the small sample class imbalance problem using feature selection, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1388–1400.

Hualong Yu received the B.S. degree from Heilongjiang University, Harbin, China, in 2005. Then he received M.S. and Ph.D degree from the college of computer science and technology, Harbin Engineering University, Harbin, China, in 2008 and 2010, respectively. Since 2010, he has been one lecturer and master supervisor in Jiangsu University of Science and Technology, Zhenjiang, China. He is author or co-author for over 20 research papers, 3 books and the program committee member for ICICSE2012. His research interests mainly include pattern recognition, machine learning and Bioinformatics, etc.

Jun Ni received the B.S. degree from Harbin Engineering University, Harbin, China, the M.S. degree from Shanghai Jiaotong University, Shanghai, China and the Ph.D degree from the University of Iowa, IA, USA. He is currently an associate professor and director of Medical Imaging HPC and Informatics Lab, Department of Radiology, Carver College of Medicine, the University of Iowa, Iowa City, IA, USA. He is also visiting professor in Harbin Engineering University and Shanghai University, China, since 2006 and 2009, respectively. He edited or co-edited 34 books or proceedings and authored or co-authored 115 peer-reviewed journal and conference papers. In addition, he is editor-in-chief of International Journal of Computational Medicine and Healthcare, associate editor of IEEE Systems Journal and editorial board member for 15 other professional journals. Since 2003, he has also been General/Program Chairs for over 50 International conferences. Currently, his research interests include distributed computation, parallel computing, medical imaging informatics, computational biology and Bioinformatics, etc.

Jing Zhao received the Ph.D degree in Harbin Institute of Technology, Harbin, China, in 2005. She is currently a professor and Ph.D supervisor in college of computer science and technology, Harbin Engineering University, Harbin, China and a senior visiting scholar in Duke University, USA. She is author or co-author for over 20 research papers. Her research interests include software reliability, mobile computing and image processing.