Pattern Anal Applic
DOI 10.1007/s10044-014-0369-7

THEORETICAL ADVANCES

Meta-classifiers for high-dimensional, small sample classification for gene expression analysis

Kyung-Joong Kim · Sung-Bae Cho

Received: 12 August 2012 / Accepted: 15 April 2014

Abstract Classification using small sample size (limited number of samples) with high dimension is a challenging problem in both machine learning and medicine as there are a wide variety of possible modeling approaches. Furthermore, it is not always clear which method is optimal for a prediction task. Different modeling choices include feature selection (dimensionality reduction), classification algorithms, and ensemble selection. There are several possible combinations of these methods, and it is not always clear which is the best. In the previous works, researchers show that evolutionary computation is useful to build an ensemble from the pairs of feature selection and classification algorithms. However, there are several parameters to be determined for the evolutionary computation and it requires computational time for the optimization. In this paper, we attempt to improve the approach by adopting meta-classification with the farthest-first clustering algorithm. The effectiveness and accuracy of our method are validated by experiments on four real microarray datasets (colon, breast, prostate and lymphoma cancers) publicly available. The results confirm that the proposed method outperforms single individual classifiers and other alternatives (standard genetic algorithm, and methods from literature).

K.-J. Kim (✉)
Department of Computer Engineering, Sejong University,
Seoul 143-747, South Korea
e-mail: kimkj@sejong.ac.kr

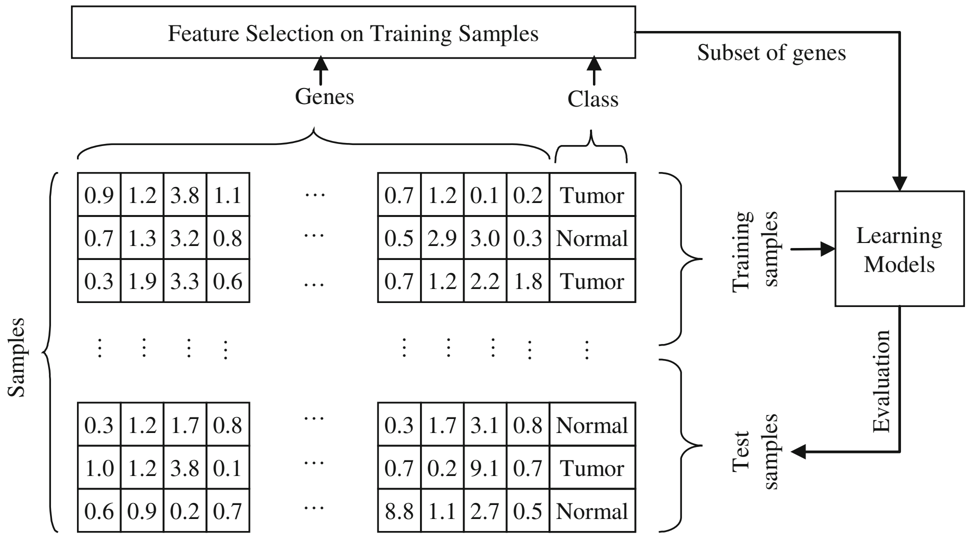
S.-B. Cho
Department of Computer Science, Yonsei University,
Seoul, South Korea
e-mail: sbcho@cs.yonsei.ac.kr

Keywords High dimension · Small sample size · Meta-classification · Ensemble classifier · Microarray data

Abbreviations

AVG	Average
CC	Cosine coefficient
CF	Classification
DCGA	Deterministic crowding genetic algorithm
DLDA	Diagonal linear discriminant analysis
ED	Euclidean distance
F1–F4	Fitness functions
FS	Feature selection
G	The number of genes
G1–G2	Global ranking feature selection methods
GA	Genetic algorithm
IG	Information gain
IV	Ideal vector
KNN	K-nearest neighbor
KNNC	KNN with cosine coefficient
KNNE	KNN with Euclidean distance
KNNP	KNN with Pearson correlation
KNNS	KNN with Spearman correlation
LOOCV	Leave-one-out cross-validation
M	The number of classification algorithms
MDL	Minimum description length
MI	Mutual information
MLP	Multi-layer perceptron
N	The number of feature selection methods
NNGE	Non-nested generalized exemplars
P	The number of training samples
PAM	Prediction analysis with microarray
PC	Pearson correlation
PCP	Pattern classification program
SNR	Signal-to-noise ratio
SP	Spearman correlation

Published online: 06 May 2014



0.9	1.2	3.8	1.1	…	0.7	1.2	0.1	0.2	Tumor
0.7	1.3	3.2	0.8	…	0.5	2.9	3.0	0.3	Normal
0.3	1.9	3.3	0.6	…	0.7	1.2	2.2	1.8	Tumor
⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮	⋮
0.3	1.2	1.7	0.8	…	0.3	1.7	3.1	0.8	Normal
1.0	1.2	3.8	0.1	…	0.7	0.2	9.1	0.7	Tumor
0.6	0.9	0.2	0.7	…	8.8	1.1	2.7	0.5	Normal

Fig. 1 An overview of classifying gene expression data

SPEGASOS	Stochastic variant of primal estimated sub-gradient solver for SVM
SVM	Support vector machine
SVML	Linear SVM
TS	Training sample

1 Introduction

Bioinformatics is one of the important application areas highly related to software development and system design [1]. The number of gene expression data is continuously increasing and we need appropriate software to mine useful knowledge from them and hardware systems to accelerate the efficient processing of data. Because the biological data have high dimensionality with small number of samples, it is challenging to design classification algorithm to handle the data. Gene activation is highly correlated with many diseases, but modeling their relationships [2, 3] is challenging. Figure 1 illustrates an example of gene expression data and their classification. Usually, gene expression data have very few samples (50–200 samples), but the number of genes is enormous (1,000–20,000 genes).

The expression value of each gene is digitized from scanned images and represented as a real value with an associated class type (“tumor” or “normal”). Our goal is to predict the type of each sample from their expression levels. Because not all genes are relevant for the classification task, it is useful to select a subset of them before learning the model. Finally, the accuracy of the model is evaluated using the test data (unseen data).

Ensemble methods are promising for tumor classification, but their accuracy is heavily dependent on the selection of the members. Like the single-classifier situation, there are a number of available pairs of feature selection and classification algorithm for ensembles. The selection of members itself is a problem to be optimized. N feature selection methods and M classification algorithms would produce $2^{N \times M}$ ensemble candidates. This is an enormous search space, and it is promising to use evolutionary computation [4] that is capable enough for a global search.

Kim et al. [5] proposed a genetic algorithm (GA) to optimize the members in the ensemble and demonstrated its performance on two gene expression datasets. In the evolutionary method, authors used only abstract-level combination method which dealt with the output of classifier as 0 or 1. They reported that the use of real-valued representation could improve the performance of classification. However, there are several parameters to be determined for the evolutionary algorithm and it requires additional time for the optimization.

In this paper, we extend the standard GA-based ensemble optimization with different levels of combination methods and types of evolutionary search algorithms. Also, instead of the selective ensemble approach, it is attempted to use all the outputs from the $M \times N$ classifiers as the input to meta-classification. We propose to use a simple clustering algorithm (farthest first) as the meta-level classifier to combine multiple classification algorithms. The meta-classifier determines the final classes of samples by combining predictions from multiple base classifiers (Fig. 2). Ho et al. [6] used simulated gene expression data to test and validate machine learning classifiers. However, in this work, the proposed method was evaluated on four real microarray datasets.

If you have M classification algorithms and N gene selection methods, it is possible to produce $M \times N$ models.

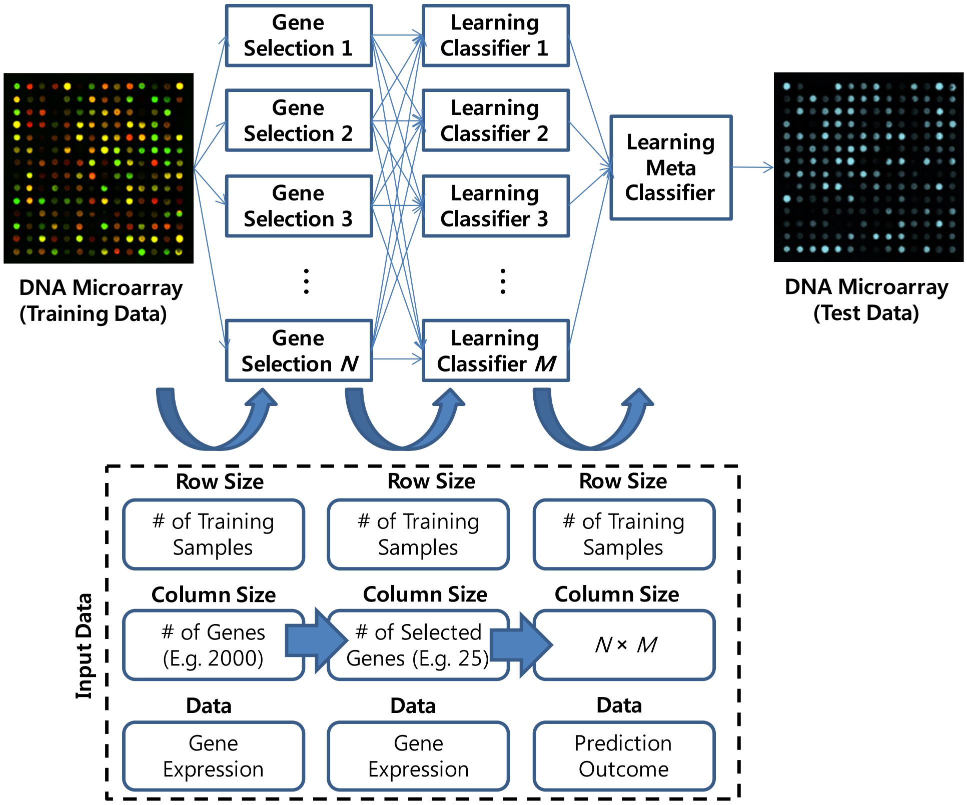


Fig. 2 An overview of meta-classifier for high-dimensional, small sample size data

Each model is trained with different classification algorithm and gene selection method. If you prefer a single classifier, you can select one model from the pool of models based on training accuracy. From our experience, the approach is not so successful because of over-fitting and unstable performance on different datasets. The solution is to combine the models as an ensemble whose member size is $M \times N$. Previous study shows that the ensemble of all members is not competitive than the single classifier [5]. The possible solution is to form an ensemble with a subset of the models but the size of ensemble search space is too big ($2^{M \times N}$). In this work, we propose to construct a meta-level classifier (inputs to the meta-classifier are the outcomes from the base classifiers). Experiment on four real-world datasets show that it is preferable to use a relatively simple meta-classifier to combine the outcomes.

In this paper, we propose to use a meta-level learning to combine a number of classification algorithms for the small-sized gene expression data. For the small-sized problem, it is difficult to select the best combination of the feature selection and the classification algorithm. Based on our proposal, it is possible to build a meta-learner from a set of several classification algorithms and different feature subset selection algorithms. The rest of this paper is organized as follows. Section 2 describes the related works. Section 3 applies the clustering meta-classifiers and the extension of the evolutionary ensembles to the $M \times N$ classifiers learned. Section 4 describes the experimental results and analysis.

2 Related works

The number of genes is usually quite substantial, but only subsets of them are useful for classification. The problem of gene selection is to choose subsets of them from all the genes. There are a wide variety of gene selection methods, which can be categorized into three classes: filter, wrapper and embedded approaches [7]. In the filter approach, the selection of the genes is independent of the choice of classification algorithms. This requires less computational cost than the other two methods. The wrapper approach selects genes based on the interaction between gene subsets and specific classification algorithms [8]. The values of gene subsets are evaluated according to the performance of the classifier trained with them. This is known as a classifier dependent and computationally expensive method.

Inza et al. [9] compared the filter and wrapper approaches in terms of accuracy and computational cost. In the embedded method, the gene selection is built into the classifier learning algorithm [10]. RANKGENE is an open-source program to select genes with the filter and wrapper approaches [11]. Recently, many efforts have been made on ensemble technique for gene selection [12]. A recent work discovered biologically significant genes from multiple gene expression data sources [11]. They used a gene-disease database to evaluate the values of the subsets found.

Buturovic implemented an open-source pattern classification program (PCP) for gene expression analysis [13]. The program contains six classification algorithms and three gene selection methods with six gene selection criteria. However, it does not support the ensemble approaches. Diaz-Uriarte et al. [14] used the random forest (an ensemble of trees) to classify gene expression datasets. This showed comparable performance to other classification methods (DLDA, KNN, and SVM). However, it supports only tree-based ensembles. Dettling [15] proposed a new type of ensemble method called bagboosting for gene expression dataset analysis. This combines two representative ensemble methods, bagging and boosting, to generate classifiers for the datasets.

Aitken et al. [16] used an evolutionary algorithm to choose relevant genes, but they did not apply it to the classification part. Li et al. [17] used the genetic algorithm to choose relevant genes combined with K-nearest neighbor method. In the paper, the classification algorithm was fixed to K-nearest neighbor method. The performance of the classification system depends on the feature selection method and classification algorithm used. There are several works reporting the performance of feature selection methods and classification algorithms for different datasets [18–21]. It is known that the ensemble of classifiers can perform better than single classifiers if they are combined properly [22]. In gene expression classification, the ensemble of heterogeneous or homogenous members (a pair of feature selection and classification algorithm) is proposed to exploit synergism of multiple models [15, 23–25].

Reduction of dimensionality has been one of the important problems in the domain of image classification, bioinformatics and biometrics. Recently, two-dimensional LDA (2D linear discriminant analysis) is successful for face recognition and Tao et al. [56] propose a preprocessing step for the 2DLDA for gait recognition problem. In [57], they report that Fisher's LDA has a tendency to merge together nearby classes if the dimension of the projected subspace is strictly lower than c-1 for the c-class classification task. Zhang et al. [58] unify spectral analysis-based dimensionality reduction algorithms with a framework, named "patch alignment". There are novel applications with the dimensional reduction algorithms in cartoon animations [59, 60].

Semi-supervised learning combines the labeled and unlabeled training samples to increase the generalization ability of models. It shows that the semi-supervised learning performs well on the multi-labeled image classification problems [46, 47]. In the gene expression data, there are some works on the use of prior knowledge with the semi-supervised learning for cancer outcome prediction problems [48, 49]. In the approach, the authors used the prior knowledge, protein–protein interaction network to guide the semi-supervised learning. In the hypergraph approach, the labeled and unlabeled samples are used to build a graph based on their similarities. Using an iterative algorithm, the labels of the unlabeled data and the weights have been optimized [50–52]. The approach has been used for the image classification [53] and cartoon animation [54, 55].

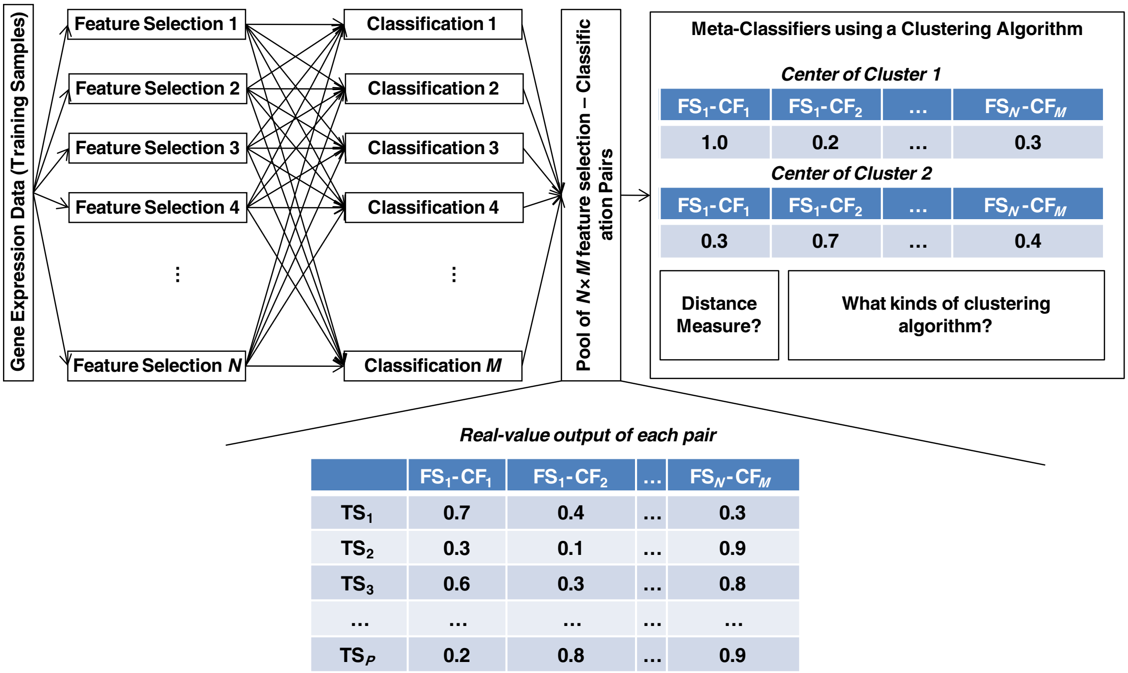
Semi-supervised learning has been an important tool to improve the performance on the combined sets of the labeled and unlabeled samples. In this paper, we used four gene expression datasets (colon, breast, prostate, and lymphoma). They are all labeled but the number of samples is very small. If there are a large number of unlabeled samples, then that could be very useful for the semi-supervised learning. However, the public datasets do not provide enough unlabeled samples and the design of new experiments for the semi-supervised learning needs additional data or prior knowledge. In this work, we focus on the development of new algorithm with the small number of labeled samples.

Clustering algorithms discover the hidden structure of the data by grouping samples based on similarity. If the number of clusters is known, the most popular technique is K-means algorithm. It is possible to use kernel instead of standard distance measure and the technique is named as "kernel K-means" [61]. Recently, non-negative matrix factorization (NMF) has been widely used for document mining, image understanding and audio analysis [62]. To improve the convergence speed of NMF, gradient-descent approach is proposed [63, 64].

3 Proposed method

The proposed method is composed of two steps: training and meta-classifier learning. In the training phase, N feature selection methods rank all genes based on different criteria. Only top-ranked genes are used to generate training datasets for each selection method. M learning methods are repeated for N different training sets, and finally there are $M \times N$ base classifiers learned. The next step is to form a meta-classifier with the classifiers.

The meta-classification includes ensemble searching and learning meta-level classifier. For the ensemble, it is not possible to enumerate all the ensembles to find the best one because of the large search space. It is necessary to use an optimization to find the best subset of the base classifiers for an ensemble. There are several parameter choices for



Center of Cluster 1

FS_1-CF_1	FS_1-CF_2	…	FS_N-CF_M
1.0	0.2	…	0.3

Center of Cluster 2

FS_1-CF_1	FS_1-CF_2	…	FS_N-CF_M
0.3	0.7	…	0.4

Real-value output of each pair

	FS_1-CF_1	FS_1-CF_2	…	FS_N-CF_M
TS_1	0.7	0.4	…	0.3
TS_2	0.3	0.1	…	0.9
TS_3	0.6	0.3	…	0.8
…	…	…	…	…
TS_P	0.2	0.8	…	0.9

Fig. 3 Overview of meta-classification using the $M \times N$ base classifiers

the optimization procedures such as encoding of solution and optimization algorithms. The representative approach is to use the evolutionary computation strong for the global search.

Meanwhile, the meta-classifier uses the decision from all the base classifiers as inputs and produces the final outcome. It is known that the number of classes for the problem is two for the cancer problem. Based on the fact, it is possible to cluster the training samples into two groups based on the arrays of predictions from the multiple classifiers. After the clustering, each cluster is mapped into one of the two classes. To classify a new instance, it is necessary to get the predictions from the multiple classifiers and assign the vector of predictions into one of the two groups using the clusters. Figure 3 illustrates these two steps in detail.

3.1 $M \times N$ base classifiers feature selection

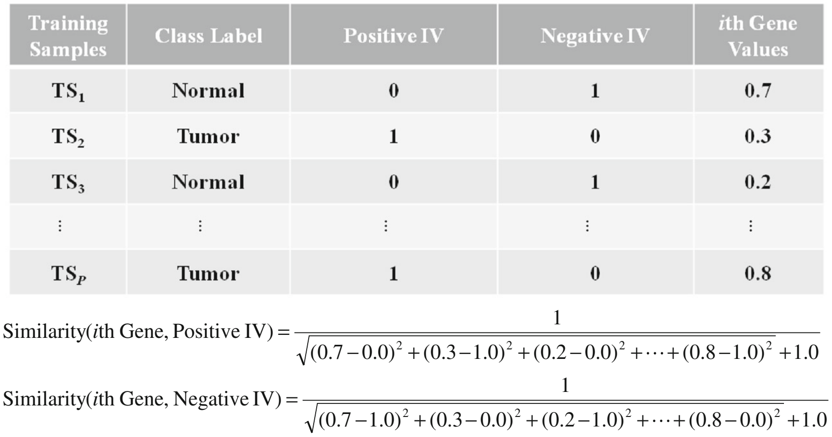
3.1.1 Feature selection

In this step, we use nine gene selection methods based on different criteria. They are categorized into similarity-based, information-theoretic and global ranking. In similarity-based methods, the value of each gene is evaluated based on the similarity to ideal vectors (Fig. 4). If there is a gene that shows the same characteristics with the ideal vectors, this means that we can classify the training samples correctly with only the single gene. Because it is not common to classify samples correctly using only single gene, this vector is called as "ideal" one. The length of the ideal vector is equal to the number of samples.

- Positive ideal vector: if the *i*th sample is "Tumor", the *i*th entry of the vector is 1; otherwise, the value is 0.
- Negative ideal vector: it is opposite to the positive ideal vector. If the sample is "normal", the *i*th value of the vector is 1.

The ideal vector is used as a reference to measure each gene's discriminative capability [25]. If there are ten training samples (five from tumor and five from normal samples), the gene's ideal expression values might be all one for the tumor and zero for normal. The closeness between the ideal vector and the each gene's expression values can be used in the gene selection algorithms.

We can sort the genes in accordance with the similarity between the gene's values for training samples and ideal vectors [26]. Because we have the two ideal vectors, there are two different rankings based on positive and negative ideal vectors. Finally, half of the genes are chosen from the rankings by the positive ideal vector, and others are from the one by the negative ideal vector. For example, if we decide to select 20 genes, 10 genes are very close to the positive ideal vectors and 10 genes are very close to the negative ones. There are four different similarity measures used: inverse of Euclidean distance measure, Pearson correlation, cosine coefficient and Spearman correlation.



Training Samples	Class Label	Positive IV	Negative IV	*i*th Gene Values
TS_1	Normal	0	1	0.7
TS_2	Tumor	1	0	0.3
TS_3	Normal	0	1	0.2
⋮	⋮	⋮	⋮	⋮
TS_P	Tumor	1	0	0.8

$$\text{Similarity}(i\text{th Gene, Positive IV}) = \frac{1}{\sqrt{(0.7-0.0)^2+(0.3-1.0)^2+(0.2-0.0)^2+\cdots+(0.8-1.0)^2}+1.0}$$

$$\text{Similarity}(i\text{th Gene, Negative IV}) = \frac{1}{\sqrt{(0.7-1.0)^2+(0.3-0.0)^2+(0.2-1.0)^2+\cdots+(0.8-0.0)^2}+1.0}$$

Fig. 4 An example of similarity-based gene selection [IV means the ideal vector. There are two ideal vectors (positive and negative). In the positive IV, it interprets the normal sample as one and tumor sample as zero. If there is a gene expression like the positive IV, that is the ideal gene to classify the samples into tumor and normal. However, in real word, the genes are not expressed in that way. So, the name of the vector is "ideal." The negative IV is just inversion of the positive IV. For each gene, it calculates the similarity between expression values of the genes and the ideal one]

In information-theoretic methods, they use the information theory to rank the genes. They are information gain and mutual information. Because the gene expression values are continuous, it is necessary to convert them into discrete value. For each gene, a threshold value for the conversion is determined to maximize information gain [11]. In the following formula, k is the total number of classes, n_l is the number of values in the left partition, n_r is the number of values in the right partition, l_i is the number of values that belong to class i in the left partition, and r_i is the number of values that belong to class i in the right partition. The information gain of a gene is defined as follows:

$$\mathrm{IG}(g_i) = \sum_{i=1}^{k}\left(\frac{l_i}{P}\log\frac{l_i}{n_l} + \frac{r_i}{P}\log\frac{r_i}{n_r}\right) - \sum_{i=1}^{k}\left(\frac{l_i+r_i}{P}\right)\log\left(\frac{l_i+r_i}{P}\right) \tag{1}$$

For each gene, it is possible to separate the training samples into two categories (right and left partitions) with a threshold value. If the gene value for the sample is higher than the threshold, the sample is categorized into right partition and vice versa. Based on the "impurity" of samples in the category, the information gain calculates the discrimination ability of the gene. If the partitioning separates samples perfectly, the gene has the maximum information gain. The equation has been used in famous decision tree learning algorithm C4.5 and other open-source gene selection program, RANKGENE.

Mutual information provides information on the dependency relationship between two probabilistic variables of events. If two events are completely independent, the mutual information is 0. The more they are related, the higher the mutual information is. It is based on the ratio between Pr(A)Pr(B) and Pr(A,B) = Pr(A)Pr(B|A) or Pr(B)Pr(A|B). If the A and B are independent, the Pr(B|A) and Pr(A|B) are equal to Pr(B) and Pr(A), respectively. In that case, the logarithm term becomes zero. Mutual information has been used in several bioinformatics papers [24–26] (T: Tumor, N: Normal).

$$\mathrm{MI}(g_i) = \mathrm{MI}(g_i \geq \mu(\overline{g}), N) + \mathrm{MI}(g_i \geq \overline{g_i}, T) + \mathrm{MI}(g_i < \overline{g_i}, N) + \mathrm{MI}(g_i < \overline{g_i}, T)$$

$$\mathrm{MI}(g_i > \overline{g_i}, N) = P(g_i > \overline{g_i}, N)\log_{10}\frac{P(g_i > \overline{g_i}, N)}{P(g_i > \overline{g_i}) \times P(N)} \tag{2}$$

If we calculate the mean μ and standard deviation σ from the distribution of gene expressions within their classes, the signal-to-noise ratio (SNR) of gene g_i is defined as follows. It is a simple measure to rank the genes based on mean and standard deviation of samples from a homogenous class [38].

$$\mathrm{SNR}(g_i) = \frac{|\mu_N(g_i) - \mu_T(g_i)|}{\sigma_N(g_i) + \sigma_T(g_i)} \tag{3}$$

In global ranking methods, the ranking of genes is determined based on the results of the seven gene selection methods (four similarity-based methods, IG, MI and SNR). There are two variants for the ranking methods. In the first method, it is determined based on the number of times that each gene is selected from the seven approaches (ED, PC, CC, SP, IG, MI, and SNR). If the gene is chosen by IG and MI, the score of the gene is two. For the second method, the ranking of each gene is summed over the seven methods (sum of the scored values). This method is a kind of hybrid gene selection algorithms and proposed in this paper.

3.1.2 Classification algorithms

We consider the following six classification methods (KNN variants, SVML, and MLP) for building ensembles. We choose them because they have been widely used for the gene expression classification research. Instead of the six algorithms, we can replace some of them with other classifiers.

K-nearest neighbor is one of the most common methods for instance-based induction. Given an input vector, KNN extracts the k closest vectors in the reference set based on similarity measures, and makes a decision for the label of the input vector using the labels of the k-nearest neighbors. In this paper, many similarity measures were used such as the inverse of Euclidean distance (KNNE), Pearson correlation (KNNP), cosine coefficients (KNNC) and Spearman correlation (KNNS) [24]. They are KNN variants. If the k is not 1, the final outcome is based on the majority voting of the k-nearest neighbors. In the binary mode, it outputs 0 or 1 as the final outcome. In the real-valued mode, it outputs real values ranging from 0 to 1. For example, if k is 3 and two of the nearest neighbors agree that the sample is tumor (1), the final outcome is 0.66.

A feed-forward multi-layer perception is an error backpropagation neural network that can be applied to pattern recognition problems. It requires engineering regarding the architecture of the model (the number of hidden layers, hidden nodes, and so on). In this classification problem, the number of output nodes is two (normal and tumor nodes). If the output from the normal node is larger than that from the tumor node, the sample is classified as normal. In the binary mode, 0 (normal) and 1 (tumor) are the outputs. In the real-value mode, real values ranging from 0 to 1 are the outputs.

Support vector machine classifies the data into two classes. SVM builds up a hyperplane as the decision surface in such a way as to maximize the margin of separation between normal and tumor samples. In this paper, linear kernel (SVML) is used. In the real-value output mode, the output of SVM is normalized as a real value between 0 (normal) and 1 (tumor).

3.2 Meta-classifiers with evolutionary ensembles

From the training phase, we get $M \times N$ classifiers. For each training sample, there are $M \times N$ predictions from the base classifiers but it is necessary to combine them to produce final outcome. A simple straightforward approach is to find the best subset of base classifiers and use them to predict the final results as a committee. It is a combinatorial optimization problem with large search space and there are several papers on applying evolutionary computation to the problem [5, 30–34].

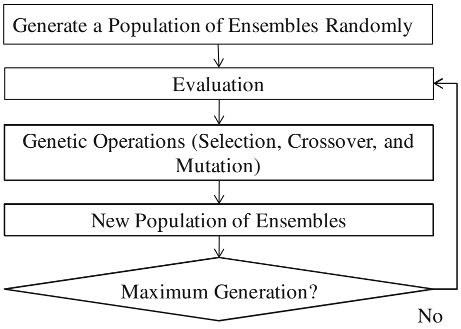

Generate a Population of Ensembles Randomly
Evaluation
Genetic Operations (Selection, Crossover, and Mutation)
New Population of Ensembles
Maximum Generation?
No

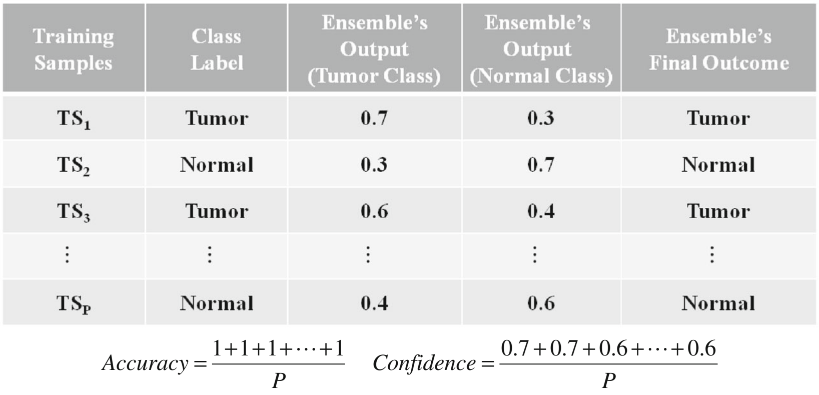
Fig. 5 Searching for ensembles by evolutionary algorithms

The next step is to form an ensemble of them automatically. The possible number of ensembles is $2^{M \times N}$. From the base classifier learning, it produces $M \times N$ different models. If all the models participate in the ensemble, size of the ensemble is $M \times N$. For each base classifier, there are two options (include or not include the classifier). In total, there are $2^{M \times N}$ ensembles. The size of ensembles is ranging from 0 to $M \times N$. An evolutionary algorithm is a machine learning method for optimization. Initially, it randomly generates a population of solutions encoded as binary or real-valued vectors. Each solution is evaluated by a predefined objective measure (fitness function). Like the natural evolutionary mechanism, this algorithm adopts the survival of the fittest, crossover and mutations of solutions. From the genetic operations, a new population of solutions is generated. This is repeated until the maximum number of generation is reached (Fig. 5).

3.2.1 Encodings of ensembles and combination methods

An encoding is the representation of a solution (in this paper, the solution is an ensemble) in the evolutionary algorithm. In the binary encoding of ensembles, if the bit is “1”, the classifier participates in the ensemble and “0” indicates non-participation. With abstract-level output of classifiers, majority voting is used as a combination method. In the real-value encoding of ensembles, it encodes the weights of each classifier in the ensemble.

Based on [35], the output information that various classification algorithms produce can be divided into two levels. In the abstract level, a classifier only outputs a unique label (cancer or normal). In the measurement level, a classifier attributes each label a measurement value to address the degree that the sample has the label. In this



Training Samples	Class Label	Ensemble's Output (Tumor Class)	Ensemble's Output (Normal Class)	Ensemble's Final Outcome
TS_1	Tumor	0.7	0.3	Tumor
TS_2	Normal	0.3	0.7	Normal
TS_3	Tumor	0.6	0.4	Tumor
⋮	⋮	⋮	⋮	⋮
TS_P	Normal	0.4	0.6	Normal

$$Accuracy = \frac{1+1+1+\cdots+1}{P} \qquad Confidence = \frac{0.7+0.7+0.6+\cdots+0.6}{P}$$

Fig. 6 An example of fitness functions (it shows calculation of F1 and F2. F3 is defined as the multiplication of F1 and F2. In F4, the size of ensemble is included in the fitness function)

study, we use only the two levels. If we use the two different levels of information from the base classifiers and the two encoding schemes for the evolutionary algorithms, there are four combinations.

- Binary classifier's output + binary ensemble encoding: the final outcome is calculated by the majority voting of members participated in the ensemble.
- Real-valued classifier's output + binary ensemble encoding: the final outcome is based on the sum of outputs of members participated in the ensemble divided by the number of members. If the outcome is <0.5, it is classified as normal.
- Binary classifier's output (*op*) + real-valued ensemble encoding: This compares the sum of weights in the ensemble (*t* and *n*, each represents "Tumor (*T*)" and "Normal (*N*)"). If *n* is larger than *t*, the sample is classified as normal. This equation has been used in [5].

$$t = \sum_{op=T} w_i \qquad n = \sum_{op=N} w_i \tag{4}$$

- Real-valued classifier's output + real-valued ensemble encoding: the final outcome is the weighted sum of classifier's outputs (*op*) divided by the sum of weights in the ensemble. If the outcome (o_f) is <0.5, it is classified as normal. This equation is newly proposed in this paper.

$$o_\mathrm{f} = \frac{\sum w_i \times op_i}{\sum w_i} \tag{5}$$

3.2.2 Fitness functions

It is important to measure the value of ensembles in the evolutionary algorithms. Figure 6 illustrates an example of fitness calculation.

- Accuracy (F1): a simple fitness function is to use accuracy of the ensemble on the training samples. This simply considers the number of correctly classified samples divided by the total number of training samples [32–34].
- Confidence (F2): this is sum of confidence of ensembles for the true class label of the training samples. In the accuracy measure, this gives the same scores regardless of the confidence of classification.
- Accuracy × confidence (F3)
- Minimum description length (MDL) principle (F4): in this measure, the final fitness value is the F1-C × (ensemble size). This prefers the ensemble with higher accuracy and the least number of members. *C* (in this paper, 0.01) is a constant value [5, 31].

3.2.3 Types of evolutionary algorithms

The genetic algorithm has been widely used as a representative method of evolutionary algorithms. Recently, there are new types of evolutionary algorithms to increase the diversity of population and avoid premature convergence. This kind of method is referred to as speciation algorithms [18]. The deterministic crowding genetic algorithm (DCGA) is one of the most successful algorithms for the speciation.

In DCGA, the diversity of the population is maintained with a special selection mechanism. At first, two ensembles are chosen as parents and they produce two children with genetic operators (crossover and mutation). The following step is to calculate distance between children and parents. For each parent, one child is assigned to maximize overall similarity, and only the fittest one between the parent and child survives to the next generation. In this way, similar individuals with less fitness are culled from the population. In this paper, we also define a hybrid method based on other evolutionary algorithms. This method chooses the best ensemble from the final populations of binary GA, binary DCGA, real-valued GA, and real-valued DCGA. Among the best ensembles from the four evolutionary runs, it chooses the one with the highest fitness.

3.3 Meta-classifiers with clustering

Evolutionary computation is promising for the optimization of the ensemble, but we need to determine the type of evolutionary algorithms, representations and operators. Also, because it is based on population-based search, we should evaluate multiple candidates to guide the search.

In this paper, we propose to use clustering algorithm to group training samples based on the predictions of the $M \times N$ classifiers. The number of clusters is fixed to the number of classes known from the training samples. In the clustering, the label of the training samples is ignored and only features of each sample (predictions of classifiers) are used. After the clustering, it is possible to identify the label of each cluster from the samples assigned. If there is a new instance classified from the $M \times N$ classifiers, the distance between the outcome vector for the sample and two central vectors is used to assign it to one of them. In sum, the first step is to get the classification results from the multiple classifiers and they are used to group samples into clusters. For the clustering, a simple farthest first is used [27, 28].

In the meta-classifier learning, it removes instances with missing class. Because the next step is unsupervised learning, it removes class attributes for the clustering algorithm. After running the clustering algorithm defined, it is necessary to find the minimum error mapping of classes to clusters considering all possible classes to cluster assignments. The farthest-first clustering is an approximated solution to maximize the radius of clusters which is known as NP-hard. Unlike other clustering algorithm, the cost function for the optimization is "maximum cluster radius." Initially, it picks a data point randomly. The next choice should be the point farthest from it. Then, it chooses the point farthest from the first two. It is repeated until the algorithm obtains k points (the same with the number of clusters). The first k points are cluster centers and remaining points are assigned to the closest center. Gonzalez used a farthest-first traversal as an approximation algorithm for the below cost function [29].

The farthest-first clustering is a very simple method compared to other clustering algorithms. For the two class problem, the algorithm selects a training sample randomly and assigns it as the center of the first cluster. It selects the farthest training sample from the center and assigns the sample as the center of the second cluster. Meta-learning usually trains a classifier to produce the final predictions using the outputs (predictions) of the base classifiers. In the gene expression data problem, the number of sample is relatively small. It is important to avoid over-fitting on the training samples. In the context of meta-learning, the simple farthest-first clustering is appropriate to avoid the over-training problems.

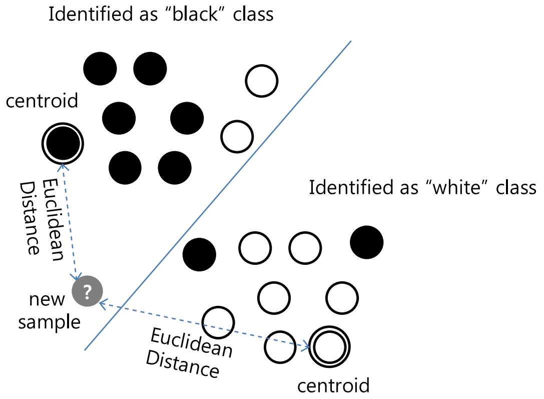


Fig. 7 The classification of the new sample (The new sample is classified as "black" category because it is close to the centroid of the cluster identified as "black."). It is necessary to apply the dimensionality reduction methods and classification on the new sample. As a result, we can get the meta-level data (the dimension is $M \times N$) to be used as an input to the meta-level classifier

Because all the training samples are assigned into one of the cluster, it is possible to check the most popular labels for each centroid. If "normal" samples are prevalent in one cluster, it is possible to label the cluster as "normal." In this way, the unsupervised clustering can be used for supervised classification. A new sample can be classified into one of classes based on the distances to the centroids of each cluster (Fig. 7). Because the clustering is running on meta-level (making the final decision based on other classifier's initial decisions), it is necessary to make a pool of classifiers that produce 1st stage decisions. We need to assume the majority decision making if we use the clustering in the context of the classification. If the numbers are balanced, it is possible to run the clustering algorithm again. Because the clustering algorithms are based on the random initialization, it is possible to get slightly different statistics if they are balanced.

In the first stage, different feature selection methods are used to select small number of relevant features. The next step is to learn multiple machine learning models from the different sets of features. If the number of feature selection is N and M classifier are used, there are total $M \times N$ pair of them. The number of features for the meta-level learning is $M \times N$ and each feature stores normalized output from the pair (Table 1). The farthest-first clustering algorithm is applied to the meta-level training samples, and finally unseen test sample is classified (Fig. 8).

4 Results

4.1 Experimental settings

The expression level of each gene is normalized to 0–1. For each gene, we found the maximum (max) and minimum

Table 1 An example of training samples for the meta-classifier (each row represents one training sample and the column is corresponding to the outputs from each pair of classifier + feature selection. For example, KNNE classifier with Euclidean distance outputs 0.666 on $(P - 1)$th training samples)

Sample ID	KNNE -ED	KNNP -ED	KNNC-ED	KNNS-ED	MLP-ED	SVML-ED	KNNE-PC	…	SVML-G2	Class
0	1.0	1.0	1.0	1.0	0.997	0.622	0.666	…	1	Tumor
1	0.0	0.0	0.0	0.0	0.136	0.446	0.0	…	0	Normal
2	1.0	1.0	1.0	1.0	0.477	0.642	1.0	…	1	Tumor
…	…	…	…	…	…	…	…	…	…	…
$P - 1$	0.666	1.0	1.0	0.0	0.013	0.516	0.333	…	0.333	Tumor
P	1.0	1.0	1.0	0.667	0.987	0.637	1.0	…	1.000	Tumor

In the figure, the shaded two samples are centroids of clusters

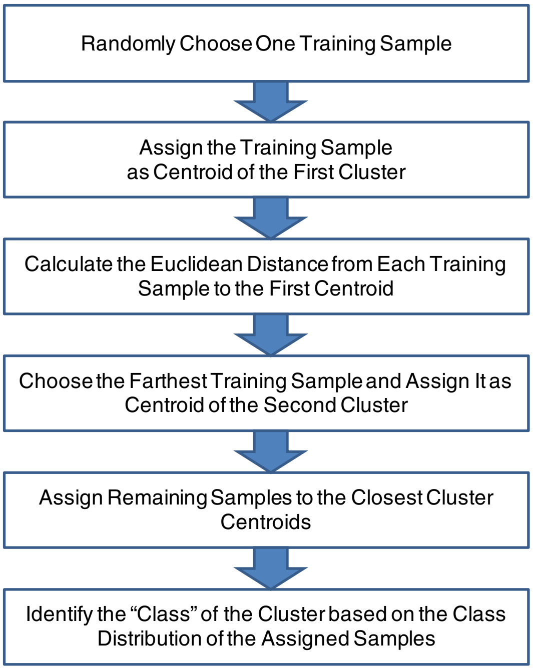

Randomly Choose One Training Sample
Assign the Training Sample as Centroid of the First Cluster
Calculate the Euclidean Distance from Each Training Sample to the First Centroid
Choose the Farthest Training Sample and Assign It as Centroid of the Second Cluster
Assign Remaining Samples to the Closest Cluster Centroids
Identify the "Class" of the Cluster based on the Class Distribution of the Assigned Samples

Fig. 8 Farthest-first clustering algorithm for binary classification problems

Table 2 Summary of datasets used

	# of genes	# of samples
Colon [38]	2,000	62
Prostate [39]	1,2600	102
Breast [40]	2,4481	97
Lymphoma [41]	4,026	45

Colon: http://microarray.princeton.edu/oncology/affydata/index.html

Prostate: http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=75

Breast: http://www.rii.com/publications/2002/vantveer.html

Lymphoma: http://llmpp.nih.gov/lymphoma/

Table 3 Parameters of classification algorithms

Classifier	Parameter	Value
MLP	# of input nodes	25
	# of hidden nodes	8
	# of output nodes	2
	Learning rate	0.05
	Momentum	0.7
	Learning algorithm	Back propagation
KNN	k	3
SVM	Kernel function	Linear

expression values (min) from the samples. The gene expression value (g) is adjusted to $(g - \min)/(\max - \min)$. In the gene selection, the number of top-ranked genes is 25 for all datasets. There is no report on the optimal number of genes, but a previous study suggests that 25 be reasonable [24]. For information gain gene selection, we implemented it based on the RANKGENE source code and our IG method showed the same results with the RANKGENE [11, 36]. We used LIBSVM for the SVM classification [37]. The datasets and parameters of classification algorithms are summarized in Tables 2 and 3.

In the evolutionary algorithms, the population size is 20, the maximum number of generation is 100, crossover rate is 0.9 and mutation rate is 0.01. C for fitness function F4 is 0.001. The type of crossover operator is one-point crossover. In the binary encoding, the mutation operator converts 0–1 or 1–0. In the real-value encoding, the weight is replaced with a randomly generated one. The final results are an average of 100 runs. For each of 10 runs, the gene expression data are randomly separated to the training dataset (2/3) and test dataset (1/3).

4.2 Classification accuracy

Table 4 summarizes the accuracy of 54 feature selection–classification algorithm pairs ($N = 9$ and $M = 6$). For each dataset, the best pair, feature selection method and classification algorithm are different. For example, KNNP-CC is

Table 4 Accuracy of feature selection–classification algorithm pairs (on test samples) (average of 10 runs) (bolded number means the best accuracy)

	ED	PC	CC	SP	IG	MI	SNR	G1	G2	AVG
(a) Colon										
KNNE	84.3	83.3	84.8	82.4	81.0	82.9	82.9	83.8	83.3	83.2
KNNP	87.1	86.2	88.1	84.3	84.8	83.8	85.2	85.2	86.7	85.7
KNNC	87.1	87.1	88.1	84.8	84.8	85.2	85.7	85.2	85.7	86.0
KNNS	86.7	87.1	85.2	86.7	85.7	85.2	86.2	85.2	84.3	85.8
MLP	80.0	82.9	78.5	79.0	80.0	83.8	83.3	81.0	80.5	81.1
SVML	85.7	85.2	85.7	83.3	83.3	84.8	85.2	84.3	84.8	84.7
AVG	85.2	85.3	85.2	83.4	83.3	84.3	84.8	84.1	84.2	84.4
(b) Prostate										
KNNE	90.0	90.9	90.0	89.4	91.2	91.2	88.8	89.1	85.0	89.5
KNNP	90.0	92.4	92.1	89.1	91.8	93.8	90.3	90.3	82.4	90.2
KNNC	91.5	92.1	91.5	89.1	92.4	92.9	90.9	92.1	87.9	91.1
KNNS	89.7	92.1	90.6	90.0	92.1	93.5	91.8	90.3	80.3	90.0
MLP	85.9	89.1	90.3	90.6	92.6	92.1	88.8	89.7	85.0	89.3
SVML	90.0	93.2	93.2	91.5	93.8	**94.1**	92.1	93.8	91.2	92.5
AVG	89.5	91.6	91.3	90.0	92.3	92.9	90.4	90.9	85.3	90.5
(c) Breast										
KNNE	79.1	**79.7**	76.6	78.8	75.9	76.6	75.9	76.3	76.3	77.2
KNNP	73.8	74.4	70.3	77.2	71.3	74.7	74.7	73.4	72.2	73.5
KNNC	78.8	79.1	77.2	77.2	74.4	77.5	74.7	74.7	75.9	76.6
KNNS	72.2	76.9	70.6	76.9	72.8	77.8	77.2	71.9	67.8	73.8
MLP	75.9	79.1	77.2	74.7	74.4	76.6	78.1	77.2	73.4	76.3
SVML	74.1	76.6	73.8	70.9	74.7	75.3	75.3	74.4	74.4	74.4
AVG	75.6	77.6	74.3	75.9	73.9	76.4	76.0	74.6	73.3	75.3
(d) Lymphoma										
KNNE	93.3	93.3	90.0	90.7	92.0	92.7	94.7	95.3	94.7	93.0
KNNP	90.0	92.7	91.3	91.3	94.0	91.3	94.7	90.7	86.7	91.4
KNNC	92.7	93.3	92.7	90.0	91.3	91.3	**96.0**	94.7	86.7	92.1
KNNS	90.7	94.7	94.7	90.0	93.3	92.0	94.0	90.7	86.0	91.8
MLP	94.0	93.3	93.3	88.7	92.7	92.0	95.3	95.3	95.3	93.3
SVML	92.0	90.7	90.0	90.7	90.0	90.0	92.7	91.3	94.0	91.3
AVG	92.1	93.0	92.0	90.2	92.2	91.6	94.6	93.0	90.6	92.1

the best one for the colon cancer dataset, but it is not the best for other three datasets. MLP is the best only for the lymphoma, while MI is the best only for the prostate dataset.

How do we choose a pair from the 54 feature selection–classification algorithms before we see the results on test samples (depicted in Table 4)? Choosing a pair based on test accuracy is not realistic because the test cases are unseen in the training phase. The easiest way is to randomly choose a pair without any information. An alternative method is to choose a pair based on training accuracy. Table 5 summarizes the performance of the strategies. The random strategy showed the lowest accuracy on test samples. The second strategy (choose one according to training accuracy) showed unstable accuracy on test samples. For the breast and lymphoma, it showed better accuracy than random choice, but it did not for the colon and prostate. This is because of over-fitting on the training samples.

Table 6 summarizes the test accuracy of ensembles found by evolutionary algorithms. It shows the results for all combinations of the control parameters. Table 7 summarizes the average accuracy of individual classifiers and ensembles found by evolutionary algorithms. It shows that the ensembles outperform the individual classifiers for all datasets. Table 8 shows the effect of real/binary outputs of member classifiers. For the colon and breast, it improved the performance but did not work for the prostate and lymphoma. Overall, setting the classifier's output as a real value is more effective than a binary one. Also it is important to use proper representation of

Table 5 Choice of the pair from 54 candidates (average of 10 runs)

	Choose a pair based on test accuracy (IDEAL)[a]	Choose a pair randomly (RANDOM)	Choose a pair based on training accuracy (T_ACCURACY)
Colon	88.1 ± 0.38	84.4 ± 0.21	79.0 ± 0.00
Prostate	94.1 ± 0.32	90.5 ± 0.27	89.0 ± 0.31
Breast	79.7 ± 0.93	75.3 ± 0.25	79.1 ± 0.00
Lymphoma	96.0 ± 0.33	92.1 ± 0.23	92.5 ± 0.20

Statistical significance test	IDEAL	RANDOM	T_ACCURACY
IDEAL			
Colon		$p = 0.001$	$p = 0.001$
Prostate		$p = 0.001$	$p = 0.001$
Breast		$p = 0.001$	$p = 0.1$
Lymphoma		$p = 0.001$	$p = 0.001$
RANDOM			
Colon			$p = 0.001$
Prostate			$p = 0.001$
Breast			$p = 0.001$
Lymphoma			$p = 0.001$

[a] Choosing a pair based on test accuracy is not a realistic method

evolutionary algorithm together with the proper combination scheme.

Figure 9 shows the performance of classification algorithms for colon, prostate, breast and lymphoma cancers. The algorithms are used to combine the results from the 54 base classifiers. Except the evolutionary ensembles, all classifiers are used for the meta-classification (learning the classifier using the outputs from other classifiers). Dagging creates a number of disjoint, stratified folds out of the data and feeds each chunk of data to a copy of the supplied base classifier (in this paper, artificial neural network) [42, 43]. Decorate exploits specially constructed artificial training examples for building diverse ensembles of classifiers. NNGE is similar to nearest neighbor algorithm using non-nested generalized exemplars, hyperpipe contains all points of each category (records attribute bounds), SPEGASOS stands for stochastic variant of primal estimated subgradient solver for SVM and Bayesian network represents joint distribution of variables using graphical models.

For colon cancer, the best one is dagging with multilayer perceptron (MLP) with 88.1 % accuracy. The proposed method is 87.62 % and the evolutionary ensemble is 86.8 %. It is clear that the dagging and clustering are good at classifying the colon cancer dataset than other alternatives. For prostate, the best classification algorithms are NNGE and SPEGASOS. However, there is small difference between the proposed clustering-based meta-classifier and the best one. In the breast cancer data, the best classifier is decorate with 80.91 %. The proposed method records 80.3 % slightly lower than the best one while the evolutionary ensembles show 77.9 % accuracy. Finally, there is no difference between the proposed method and evolutionary ensembles for the lymphoma data. It is interesting that Bayesian network is the best for the dataset. Although the structure of network was learnt from data, it is similar to naive Bayesian classifier. Although the proposed clustering-based method is not always the best, it shows higher accuracy for the four datasets. For example, the dagging (MLP) is very good at the colon cancer but a bit lower than the proposed method in the breast cancer.

Figure 10 shows the distribution of predictions (each cross represents a training sample) over SVML classifiers with different feature selection methods. Although they are highly correlated, there are overlapped areas making classification difficult. The figure shows the behavior of the farthest-first clustering algorithm. They are located to maximize the radius of clusters. Figure 11 shows a comparison of clustering algorithms used for the meta-classifier learning.

Table 9 summarizes the ranks of 40 ensembles and 54 individual classifiers sorted by average accuracy over the four gene expression datasets. The ensemble outperformed the best individual pair of feature selection and classification. From the 1st rank to 38th rank, there are only two individual classifiers. It is clear that the classifier's output as real values is beneficial to the performance of ensembles. The best ensemble was found by real-valued GA with F4 fitness function and real-valued classifier's outputs. The proposed meta-classifier with clustering outperforms the evolutionary ensembles regardless of their representations and types. Table 10 shows the performance comparison with other works published in the field of bioinformatics. It shows that the proposed method works very well compared to other methods. In the prostate cancer dataset, it performs the best. Researchers use different evaluation methods to test their work (for example, n-fold cross-validation, LOOCV, and random partitions). In this work, we adopt random partitioning (2/3 training, 1/3 test) followed from [15].

5 Conclusion and future works

In this paper, we proposed a meta-classifier with clustering algorithm to combine the outputs from multiple

Table 6 Accuracy of ensembles found by evolutionary algorithms (on test samples, average of 100 runs)

Classifier's output	Fitness function	Evolutionary algorithms					AVG
		Binary GA	Real-value GA	Binary DCGA	Real-value DCGA	Hybrid	
(a) Colon							
Real	F1	87.0	87.4	87.0	87.0	86.6	87.0
Real	F2	87.2	87.5	86.7	87.4	86.7	87.1
Real	F3	87.1	87.4	86.8	87.1	86.8	87.0
Real	F4	86.9	**87.5**	86.7	87.2	86.7	87.0
Binary	F1	87.0	87.0	86.6	87.0	87.0	86.9
Binary	F2	86.7	86.8	85.7	86.2	85.7	86.2
Binary	F3	86.3	87.1	85.3	86.6	85.3	86.1
Binary	F4	86.7	87.1	86.6	86.9	86.6	86.8
AVG		86.9	87.2	86.4	86.9	86.4	86.8
(b) Prostate							
Real	F1	93.1	92.9	93.0	92.8	93.1	93.0
Real	F2	92.8	93.0	92.9	92.8	92.9	92.9
Real	F3	92.8	93.1	92.8	92.9	92.8	92.9
Real	F4	93.1	93.0	92.7	93.0	92.7	92.9
Binary	F1	93.0	93.0	93.0	92.9	93.0	93.0
Binary	F2	93.1	93.1	93.0	93.2	93.0	93.1
Binary	F3	93.3	93.0	**93.5**	93.2	**93.5**	93.3
Binary	F4	93.2	93.0	92.6	93.0	92.6	92.9
AVG		93.0	93.0	92.9	93.0	92.9	93.0
(c) Breast							
Real	F1	78.5	78.3	78.3	78.6	78.5	78.4
Real	F2	78.1	78.4	78.0	78.3	78.0	78.2
Real	F3	78.3	78.1	78.1	78.4	78.1	78.2
Real	F4	**78.8**	78.5	**78.8**	78.7	**78.8**	78.7
Binary	F1	77.8	77.2	77.5	77.0	77.8	77.5
Binary	F2	77.1	77.1	77.6	77.6	77.6	77.4
Binary	F3	77.2	77.0	77.9	76.9	77.9	77.4
Binary	F4	77.8	77.2	78.6	77.0	78.6	77.8
AVG		77.9	77.7	78.1	77.8	78.2	77.9
(d) Lymphoma							
Real	F1	93.2	93.5	93.5	93.4	93.2	93.4
Real	F2	93.4	93.5	93.5	93.3	93.5	93.5
Real	F3	93.5	93.4	93.3	93.5	93.3	93.4
Real	F4	93.1	93.4	93.6	93.3	93.6	93.4
Binary	F1	93.1	93.7	93.4	93.6	93.1	93.4
Binary	F2	93.2	93.7	93.7	**93.8**	93.6	93.6
Binary	F3	93.6	**93.8**	93.7	93.7	93.7	93.7
Binary	F4	93.5	93.3	93.5	93.5	93.5	93.5
AVG		93.3	93.5	93.5	93.5	93.5	93.5

Bold values indicate the best accuracy for the dataset

feature-classification algorithm pairs. It shows that the proposed method is simple with small number of parameters to be determined and highly accurate compared to other alternatives on four real microarray data. Although the clustering algorithm is simple to be implemented, it works well on the combination problems. Although the evolutionary computation approach is promising, there are several parameters to be determined and sometimes it suffers from the over-fitting to the training samples.

Table 7 Comparison of averaged performance between single classifiers and ensembles found by evolutionary algorithms

	AVG of 54 feature selection–classification algorithm pairs	AVG of 40 ensembles found by evolutionary algorithms
Colon	84.4	86.8
Prostate	90.5	93.0
Breast	75.3	77.9
Lymphoma	92.1	93.5

Table 8 Effect of classifier's outputs of feature selection–classification algorithm pairs

	Real-value classifier's output	Binary classifier's output
Colon	87.0	86.5
Prostate	93.0	93.1
Breast	78.4	77.5
Lymphoma	93.4	93.5

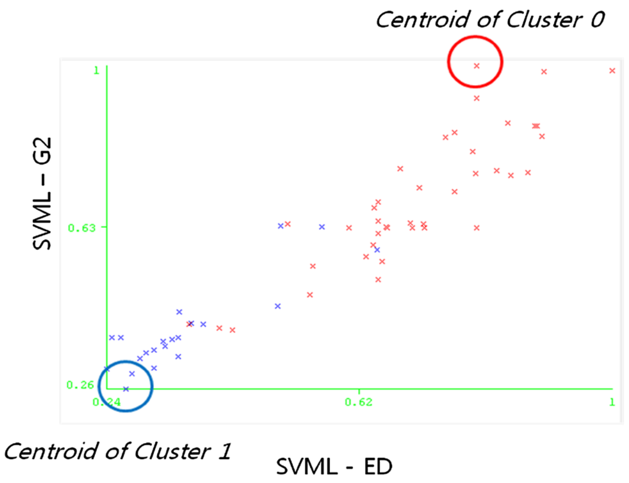


Fig. 10 Visualization of prediction results from two classifiers on colon cancer dataset

It showed that the proposed method outperformed other evolutionary alternatives and results published in the literatures. The proposed method showed robust

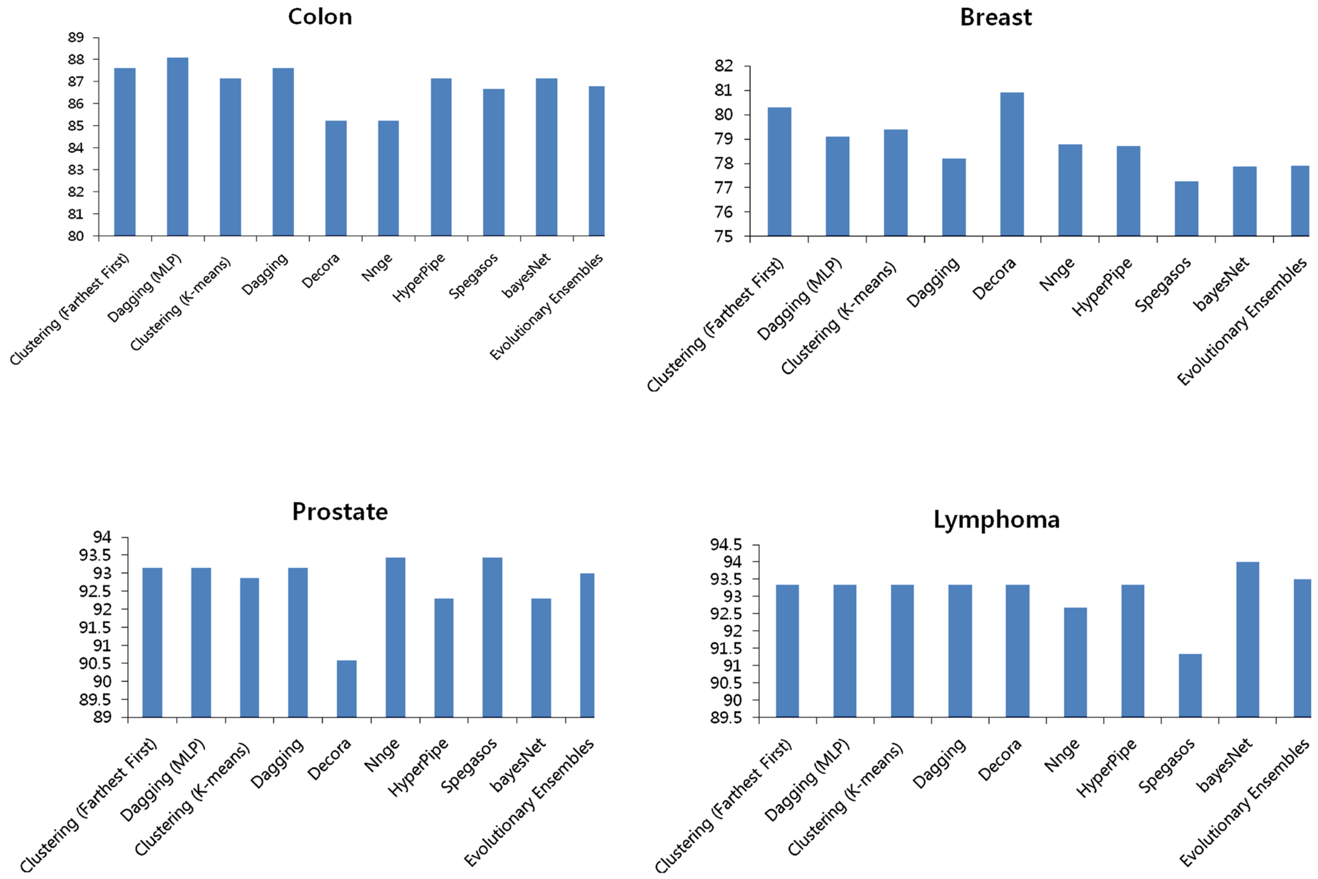
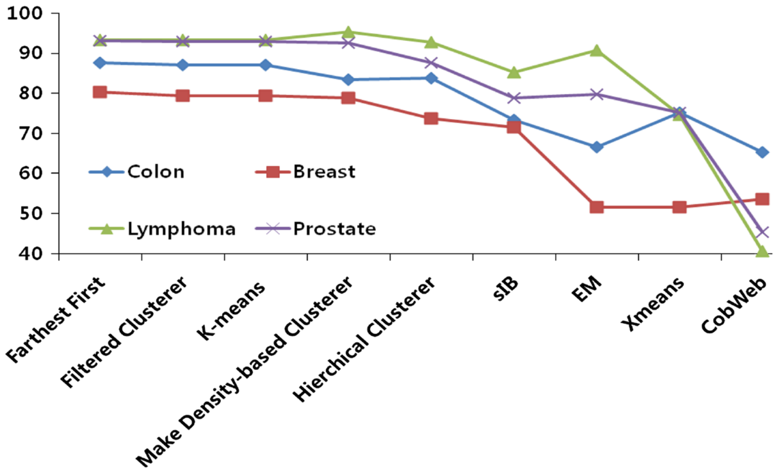


Fig. 9 Comparison with other meta-classifiers and evolutionary ensembles (average of 10 runs) (The purpose of this figure is to show that the clustering-based approach shows relatively stable results compared to other candidates. For the four datasets, the proposed method is always in the top three among the ten methods.)

Fig. 11 Comparison of different clustering algorithms with statistical significance test results [It shows that FF, FC, KM, DC and HC are significantly better than other four clustering algorithms (IB, EM, XM and CW). For FF, FC, KM, DC, and HC, there is no statistically significant difference on their performance.]



		FF	FC	KM	DC	HC	IB	EM	XM	CW
FF	Colon						p=0.001	p=0.002	p=0.01	p=0.001
	Prostate						p=0.001	p=0.5	p=0.001	p=0.001
	Breast						p=0.01	p=0.001	p=0.001	p=0.001
	Lymphoma						p=0.05		p=0.02	p=0.001

Table 9 Average accuracy over the four gene expression datasets

Rank	Combination methods	Fitness function	Evolutionary algorithms	Accuracy on test samples
1	Clustering (farthest First)			88.6
2	Dagging (MLP)			88.4
3	Clustering (simple K-means)			88.2
4	Measurement	F4	Real GA	88.1
5	Measurement	F2	Real GA	88.1
6	Measurement	F4	Real DCGA	88.0
7	Measurement	F1	Real GA	88.0
8	Measurement	F3	Real GA	88.0
9	Measurement	F3	Real DCGA	88.0
10	Measurement	F1	GA	88.0

Table 10 Comparison with other works [15] (2/3 training and 1/3 test datasets) (average over multiple runs)

Method	Test accuracy
Colon	
Bagboosting	83.9
Boosting	80.9
Random forest	85.1
SVM	85.0
PAM	**88.1**
DLDA	87.1
kNN	83.6
Evolutionary ensembles	86.8
Proposed method	87.6
Prostate	
Bagboosting	92.5
Boosting	91.3
Random forest	91.0
SVM	92.1
PAM	83.5
DLDA	85.8
kNN	89.4
Evolutionary ensembles	93.0
Proposed method	**93.1**

Bold values indicate the best accuracy for the dataset

classification accuracy over four widely used microarray datasets. Those results indicate the careful choice of meta-classifiers in the multiple classifier system is key points to get the high accuracy. To show the effectiveness and accuracy of the proposed method, we compared several alternatives by changing the fitness function, representation (binary and real), combination methods (abstract and measurement levels), and evolutionary algorithms (GA and DCGA).

In this paper, we applied the proposed method to binary classification problems but it can be extended to multi-class problems. In the case, the base classifier with binary classification has to be modified to handle the multi-class cases. The correlation of classifiers provides with useful information to combine multiple classifiers [44] and it is promising to selectively choose classifiers highly correlated. In this work, the topology of artificial neural network is fixed but it is possible to evolve the architecture with weight parameters simultaneously.

Acknowledgement This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2013 R1A2A2A01016589, 2010-0018950, 2010-0018948).

References

1. Psomopoulos FE, Mitkas PA (2010) Bioinformatics algorithm development for grid environments. J Syst Softw 83: 1249–1257
2. Slonim DK (2002) From patterns to pathways: gene expression data analysis comes of age. Nat Genet 32:502–508
3. Braga-Neto U (2007) Fads and fallacies in the name of small-sample microarray classification. IEEE Signal Process Mag 24:91–99
4. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Boston
5. Kim KJ, Cho SB (2008) An evolutionary algorithm approach to optimal ensemble classifiers for DNA microarray data analysis. IEEE Trans Evol Comput 12:377–388
6. Xie X, Ho JWK, Murhpy C, Kaiser G, Xu B, Chen TY (2011) Testing and validating machine learning classifiers by metamorphic testing. J Syst Softw 84:544–558
7. Saeys Y, Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23:2507–2517
8. Blanco R, Larranaga P, Inza I, Sierra B (2004) Gene selection for cancer classification using wrapper approaches. Int J Pattern Recognit Artif Intell 18:1373–1390
9. Inza I, Larranaga P, Blanco R, Cerrolaza AJ (2004) Filter versus wrapper gene selection approaches in DNA microarray domains. Artif Intell Med 31:91–103
10. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Mach Learn 46:389–422
11. Su Y, Murali TM, Pavlovic V, Schaffer M, Kasif S (2003) RankGene: identification of diagnostic genes based on expression data. Bioinformatics 19:1578–1579
12. Liu H, Liu L, Zhang H (2010) Ensemble gene selection by grouping for microarray data classification. J Biomed Inform 43:81–87
13. Buturovic LJ (2006) PCP: a program for supervised classification of gene expression profiles. Bioinformatics 22:245–247
14. Diaz-Uriarte R, de Andres SA (2006) Gene selection and classification of microarray data using random forest. BMC Bioinform 7:3
15. Dettling M (2004) Bagboosting for tumor classification with gene expression data. Bioinformatics 20:3583–3593
16. Jirapech-Umpai T, Aitken S (2005) Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes. BMC Bioinform 6:148
17. Li L, Weinberg CR, Darden TA, Pedersen LG (2001) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17:1131–1142
18. Dudoit S, Fridlyand J, Speed TP (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. J Am Stat Assoc 97:77–87
19. Cho SB, Won HH (2003) Data mining for gene expression profiles from DNA microarray. Int J Softw Eng Knowl Eng 13:593–608
20. Pochet N, Smet FD, Suykens JAK, Moor BLRD (2004) Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. Bioinformatics 20:3185–3195
21. Lee JW, Lee JB, Park M, Song SH (2005) An extensive comparison of recent classification tools applied to microarray data. Comput Stat Data Anal 48:869–885
22. Kuncheva LI (2004) Combining pattern classifiers: methods and algorithms. Wiley, New York
23. Tan AC, Gilbert D (2003) Ensemble machine learning on gene expression data for cancer classification. Appl Bioinform 2:S75–S83
24. Cho SB, Ryu JW (2002) Classifying gene expression data of cancer using classifier ensemble with mutually exclusive features. Proc IEEE 90:1744–1753
25. Cho SB, Won HH (2007) Cancer classification using ensemble of neural networks with multiple significant gene subsets. Appl Intell 26:243–250
26. Won HH, Cho SB (2003) Neural network ensemble with negatively correlated features for cancer classification. Lect Notes Comput Sci 2714:1143–1150
27. Hochbaum D, Shmoys DB (1985) A best possible heuristic for the k-center problem. Math Oper Res 10:180–184
28. Dasgupta S (2010) Hierarchical clustering with performance guarantees. In: Classification as a tool for research, studies in classification, data analysis, and knowledge organization, pp. 3–14. doi:10.1007/978-3-642-10745-0_1
29. Gonzalez TF (1985) Clustering to minimize the maximum intercluster distance. Theoret Comput Sci 38:293–306
30. Cho SB, Park CH (2004) Speciated GA for optimal ensemble classifiers in DNA microarray classification. IEEE Congr Evolut Comput 590–597
31. Kim KJ, Cho SB (2005) DNA gene expression classification with ensemble classifiers optimized by speciated genetic algorithm. In: First international conference on pattern recognition and machine intelligence, pp 649–653
32. Park CH, Cho SB (2003) Evolutionary ensemble classifier for lymphoma and colon cancer classification. IEEE Congr Evolut Comput 2378–2385
33. Park CH, Cho SB (2003) Evolutionary computation for optimal ensemble classifier in lymphoma cancer. In: 14th international symposium on methodologies for intelligent systems, pp 521–530
34. Kim KJ, Cho SB (2010) Exploring features and classifiers to classify microRNA expression profiles of human cancer. In: 17th international conference on neural information processing, pp 234–241
35. Xu L, Krzyzak A, Suen CY (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans Syst Man Cybern 22:418–435
36. RANKGENE. http://genomics10.bu.edu/yangsu/rankgene/
37. LIBSVM. http://www.csie.ntu.edu.tw/~cjlin/libsvm/
38. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D et al (1999) Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci USA 96:6745–6750
39. Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C et al (2002) Gene expression correlates of clinical prostate cancer behaviour. Cancer Cell 1:203–209
40. van't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AAM, Mao M et al (2002) Gene expression profiling predicts clinical outcome of breast cancer. Nature 415:530–536
41. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A et al (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403:503–511
42. Witten IH, Frank E, Hall MA (2011) Data mining: practical machine learning tools and techniques, 3rd edn. Morgan Kaufmann, London
43. WEKA Toolkit. www.cs.waikato.ac.nz/ml/weka/
44. Kim KJ, Cho SB (2006) Ensemble classifiers based on correlation analysis for DNA microarray classification. Neurocomputing 70:187–199

45. Dehuri S, Roy R, Cho SB, Ghosh A (2012) An improved swarm optimized functional link artificial neural network (ISO-FLANN) for classification. J Syst Softw 85:1333–1345
46. Luo Y, Tao D, Geng Bo, Xu C, Maybank SJ (2013) Manifold regularized multitask learning for semi-supervised multilabel image classification. IEEE Trans Image Process 22:523–536
47. Luo Y, Tao D, Xu C, Xu C, Liu H, Wen Y (2013) Multiview vector-valued manifold regularization for multilabel image classification. IEEE Trans Neural Netw Learn Syst 24:709–722
48. Hwang TH, Tian Z, Kuang R, Kocher JP (2008) Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. In: IEEE international conference on data mining, pp 293–302
49. Tian Z, Hwang TH, Kuang R (2009) A hypergraph-based learning algorithm for classifying gene expression and array CGH data with prior knowledge. Bioinformatics 25:2831–2838
50. Zhou D, Huang J, Scholkopf (2005) Learning from labeled and unlabeled data on a directed graph. In: Proceedings of the 22nd international conference on machine learning, pp 1036–1043
51. Zhu X, Ghahramani Z, Lafferty J (2003) Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the international conference on machine learning, pp 912–919
52. Wu M, Scholkopf B (2007) Transductive classification via local learning regularization. J Mach Learn Res-Proc Track 2:628–635
53. Yu J, Tao D, Wang M (2012) Adaptive hypergraph learning and its application in image classification. IEEE Trans Image Process 21:3262–3272
54. Yu J, Wang M, Tao D (2012) Semisupervised multiview distance metric learning for cartoon synthesis. IEEE Trans Image Process 21:4636–4648
55. Yu J, Liu D, Tao D, Seah HS (2011) Complex object correspondence construction in two-dimensional animation. IEEE Trans Image Process 20:3257–3269
56. Tao D, Li X, Wu X, Maybank SJ (2007) General tensor discriminant analysis and Gabor features for gait recognition. IEEE Trans Pattern Anal Mach Intell 29:1700–1715
57. Tao D, Li X, Wu X, Maybank SJ (2009) Geometric mean for subspace selection. IEEE Trans Pattern Anal Mach Intell 31:260–274
58. Zhang T, Tao D, Li X, Yang J (2009) Patch alignment for dimensionality reduction. IEEE Trans Knowl Data Eng 21:1299–1313
59. Yu J, Liu D, Tao D, Seah HS (2012) On combining multiple features for cartoon character retrieval and clip synthesis. IEEE Trans Syst Man Cybern—Part B: Cybern 42:1413–1427
60. Yu J, Tao D (2013) Modern machine learning techniques and their applications in cartoon animation research, Wiley-IEEE Press, Piscataway
61. Dhillon IS, Guan Y, Kulis B (2004) Kernel k-menas: Spectral clustering and normalized cuts. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, pp 551–556
62. Pauca VP, Shahnaz F, Berry MW, Plemmons RJ (2004) Text mining using non-negative matrix factorizations. In: Proceedings of the fourth SIAM international conference on data mining, pp 452–456
63. Guan N, Tao D, Luo Z, Yuan B (2011) Non-negative patch alignment framework. IEEE Trans Neural Netw 22:1218–1230
64. Guan N, Tao D, Luo Z, Yuan B (2012) NeNMF: an optimal gradient method for nonnegative matrix factorization. IEEE Trans Signal Process 60:2882–2898