Exploiting Local Coherent Patterns for Unsupervised Feature Ranking

Qinghua Huang, Dacheng Tao, *Member, IEEE*, Xuelong Li, *Senior Member, IEEE*, Lianwen Jin, and Gang Wei

***Abstract*—Prior to pattern recognition, feature selection is often used to identify relevant features and discard irrelevant ones for obtaining improved analysis results. In this paper, we aim to develop an unsupervised feature ranking algorithm that evaluates features using discovered local coherent patterns, which are known as biclusters. The biclusters (viewed as submatrices) are discovered from a data matrix. These submatrices are used for scoring relevant features from two aspects, i.e., the interdependence of features and the separability of instances. The features are thereby ranked with respect to their accumulated scores from the total discovered biclusters before the pattern classification. Experimental results show that this proposed method can yield comparable or even better performance in comparison with the well-known Fisher score, Laplacian score, and variance score using three UCI data sets, well improve the results of gene expression data analysis using gene ontology annotation, and finally demonstrate its advantage of unsupervised feature ranking for high-dimensional data.**

***Index Terms*—Bicluster score, feature selection, unsupervised learning.**

I. Introduction

FEATURE selection is an important preprocessing step before recognizing meaningful patterns from a data set with a large number of features. Many studies have shown that features (dimensionality) can be reduced without degrading classification/clustering performance [1]–[6]. Selecting an appropriate subset of more representative features (or dimensions) can even improve the identification performance for patterns. In contrast to the area of dimension reduction [3], [6], [32], [33], the objective of feature selection is to find optimal or suboptimal subsets from the original feature sets for irrelevant features removal, intrinsic class information preservation, and improvement of supervised and unsupervised classification performance of classifiers. Many studies have shown that a good feature selection method is very effective in improving pattern mining performance and learning accuracy in numerous real-world applications [4]–[9]. Feature selection is therefore regarded as an important preprocessing step for analyzing various sorts of data analysis.

The methods of feature selection can be grouped into two categories, i.e., the filter [10] and wrapper [11] methods. The filter methods evaluate the relevance of each feature (or feature subset) and select the features that can maximize some preset performance measures. They are independent of the subsequent learning algorithms (e.g., some classifiers). In contrast, the second category (i.e., wrapper methods) makes use of predetermined learning algorithms to evaluate the feature subsets. Hence, the goal of feature selection is to find the subset of features that minimizes the classification error using a specified classifier. Wrappers usually yield good classification accuracy for a particular classifier at the cost of less generalization of the selected features on other classifiers. In addition, although the wrapper methods often outperform filter methods in practice, they are intractable to large data sets and, hence, more computationally intensive.

In both filter and wrapper methods, the optimal feature subset needs to be found. Accordingly, a number of methods, including exhaustive search [4], sequential forward (backward) selection [12], sequential forward (backward) floating search [13], evolutionary search [14], etc., are performed to examine combinations of feature subsets. Because the computational complexity quickly increases with the number of features, it is always impractical to evaluate a large number of feature subsets. To overcome this problem, a number of filter methods adopt the ranking method [18], [19], [25], [26] in which the original d features are individually assessed and the m $(< d)$ best features can be selected for subsequent pattern analysis. Although the ranking method is much faster than that of exhaustively (or heuristically) searching for the optimal (or suboptimal) feature subset, it has been recognized that the subset of individually "good" features may not collectively provide good classification performance [15], mainly due to the lack of information about feature correlations.

On the other hand, wrapper approaches evaluate the classification quality of each feature subset. The features with

Manuscript received July 2, 2010; revised November 23, 2010; accepted April 8, 2011. Date of publication June 16, 2011; date of current version November 18, 2011. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2011CB707000, by the National Natural Science Funds of China under Grants 61001181, U0735004, 60902087, and 61072093, by the Specialized Research Funds for the Doctoral Program of Higher Education of China under Grant 20100172120010, by the Fundamental Research Funds for the Central Universities, South China University of Technology, under Grants 2009ZM0059, 2009ZZ0014, and 2009ZM0036, and by the Open Project of State Key Laboratory of Industrial Control Technology of Zhejiang University under Grant ICT1105. This paper was recommended by Associate Editor S. Sarkar.

Q. Huang, L. Jin, and G. Wei are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: qhhuang@scut.edu.cn; eelwjin@scut.edu.cn; ecgwei@scut.edu.cn).

D. Tao is with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW 2007, Australia (e-mail: dacheng.tao@uts.edu.au).

X. Li is with the Center for Optical Imagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2011.2151256

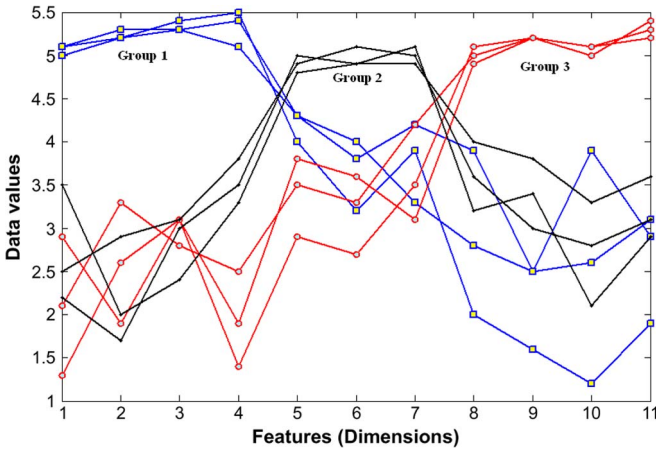

Fig. 1. Illustration of the effect of classification performance under feature subsets. The instances in group 1 can be well identified with the feature subset {1, 2, 3, 4}, those in group 2 can be well identified with the feature set {5, 6, 7}, and those in group 3 can well be identified with the feature set {8, 9, 10, 11}. There does not exist an optimal feature subset that can well identify all of the three groups.

2	7	8	4	17	26	3	9	5	1
4	18	**3**	13	**6**	12	**8**	**14**	9	10
24	3	9	11	4	16	3	11	20	4
11	6	**3**	10	**6**	17	**8**	**14**	13	5
5	8	**3**	7	**6**	2	**8**	**14**	6	10
10	22	8	2	10	16	9	7	1	13
21	13	8	5	15	27	14	2	3	7
14	18	2	6	9	4	12	11	23	1
1	4	**3**	8	**6**	19	**8**	**14**	11	5
7	3	11	24	3	2	8	16	9	6

Fig. 2. Example of a bicluster. A bicluster with constant columns is formed by the highlighted elements, which are actually a submatrix with a local coherent pattern.

strong interactions can be detected and grouped into a subset during the search for possible combinations of features. Under a specific feature subset, the classifier associated with a wrapper method is globally applied to all instances for evaluating its separability. However, in some cases, a group of samples can be best clustered under a feature subset a, but another group of samples can be best clustered under another feature subset b, as illustrated in Fig. 1. In this situation, it is difficult to determine which feature subset is optimal, and selecting an "optimal" subset of features makes it impractical for discovering many significant clusters that are best recognized under other feature subsets. In other words, there are many meaningful patterns, each of which has a subgroup of instances only under a certain subset of features in many data.

Another example of such local pattern can also be seen in Fig. 2, where the aggregated rows (instances) or columns (features) are not necessarily consecutive. The local coherent pattern containing a subset of instances and a subset of features is first termed as a *bicluster* [16] for gene expression profiling in microarray data. Applying conventional clustering algorithm to rows or columns thus results in significant difficulty in finding biclusters. In recent years, more and more attention has been paid to discovering biclusters in microarray data [17]. Because a bicluster contains a subset of experimental conditions and a subset of genes, the interrelationships among the conditions and those among the genes can be revealed. In other words, due to the intrinsic idea of bidimensional clustering, the discovered biclusters are able to provide important clues for extracting feature interdependencies and clusters of instances and are therefore potentially useful for evaluating features by simultaneously considering both feature interdependencies and instance separability. Liu *et al.* [23] made use of a spectral biclustering algorithm specifically for semi-unsupervised gene ranking and combination.

So far, most of the filter and wrapper methods on feature selection can be regarded as supervised algorithms since the class labels are used. The supervised methods evaluate feature subsets with respect to the relevance between features and class labels. If the class labels are sufficient to categorize the data set, supervised methods often outperform unsupervised methods. Even with the presence of class labels, it is a challenging problem. We discuss unsupervised learning, which is more challenging. Recently, more and more attention has been paid to developing feature selection algorithms for unlabeled data. Some unsupervised methods [18], [19] find good features according to the separability of instances. Dy *et al.* [12] described an unsupervised wrapper method using an expectation–maximization (EM) algorithm. The quality of clusters obtained from different feature subsets are used for measuring cluster separability. In more recent work [20], [21], feature similarity was measured for detecting redundant features. Law *et al.* [22] proposed a concept of feature saliency estimated using an EM algorithm for simultaneously selecting features and clustering instances.

In this paper, we propose a new unsupervised feature ranking algorithm based on the discovery of biclusters. To the best of our knowledge, it is the first attempt to making use of biclustering analysis for selecting and ranking features. Because this method incorporates a biclustering algorithm to discover biclusters and ranks the features, it has some characteristics of both feature ranking and wrapper methods and can therefore be viewed as a hybrid model. We make use of the discovered biclusters to evaluate features from two aspects, i.e., the interdependencies among features and the separability of instances. By considering both the feature correlations and instance separability in evaluating the features, we propose a scoring scheme to rank each of the features and test its performance using several often-used UCI data sets [24], a real yeast gene expression data set [28], and a high-dimensional data set [30].

This paper is organized as follows. Section II introduces the proposed algorithm in detail. Section III presents the experimental results and the last section draws conclusions.

II. Unsupervised Feature Ranking Algorithm

A. Basic Idea

As illustrated in Fig. 2, a bicluster defined by a subset of rows (instances) and a subset of columns (features) indicates a submatrix, which can be viewed as a local coherent pattern. In such a pattern, the set of features associated with the submatrix have the same contribution to the identification of the clustered instances, indicating that there exist correlations among these features. Similarly, the correlations among its instances can also be revealed, and the instances can be represented as a cluster discovered under the feature subset, indicating a successful separation from the other instances. Thus, it is observed that a bicluster well exploited from the data matrix can provide useful information about both the intercorrelations among its features and the separability of the instance subset from the others under its features. In this paper, we make use of the biclusters found in a data matrix to score the features, and this new scoring scheme is named as bicluster score.

To use the intrinsic information contained in a bicluster for evaluating features, we first propose an effective biclustering algorithm that converts the problem of searching for biclusters into two easy-to-apply procedures: 1) conventional hierarchical clustering (HC) of instances for each feature and 2) heuristic search for the biclusters (submatrices) associated with the clustered instances exploited in the first procedure. More specifically, a cluster under a single feature in the first procedure can be regarded as part of a potential bicluster. Its instances may be the rows of the bicluster. Consequently, the second procedure is to search for the features under which the same instances can also be well clustered. The instances associated with the cluster found in the first procedure and the features found in the second procedure form a submatrix which is the bicluster to be exploited. We may find a number of clusters in the first procedure and the same number of biclusters after the second procedure. From the discovered biclusters, two factors (i.e., the feature interdependencies and the instance separabilities) are thereafter considered and incorporated into the computation of the bicluster score for each feature. Finally, the features are ranked according to their bicluster scores.

B. Biclustering Discovery of Locally Coherent Patterns

As shown in Fig. 3, biclusters have several different models, including the constant, multiplicative, and coherent evolutionary models. Details about biclustering algorithms can be found in [9] and [17]. However, most of those algorithms are specifically designed for analyzing gene expression profiles, where the genes may be coregulated in a scaling, shifting, or even hybrid manner [25] and, hence, cannot be directly used to solve a generalized classification/clustering problem.

According to the Euclidean distance, the models with constant columns in Fig. 3 can be regarded as a group of points forming a compact cluster in a multidimensional space, which can be well recognized using some conventional clustering algorithms, such as HC [4]. Consequently, we focus on the biclusters with constant columns in this paper.

1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0

Constant

2.0	2.0	2.0	2.0
1.5	1.5	1.5	1.5
2.5	2.5	2.5	2.5
0.5	0.5	0.5	0.5

Constant rows

1.2	1.8	1.6	2.0
1.2	1.8	1.6	2.0
1.2	1.8	1.6	2.0
1.2	1.8	1.6	2.0

Constant columns

1.5	2.5	0.5	5.5
3.5	4.5	2.5	7.5
2.5	3.5	1.5	6.5
4.5	5.5	3.5	8.5

Additive

1.0	2.0	3.0	4.0
0.5	1.0	1.5	2.0
2.0	4.0	6.0	8.0
1.5	3.0	4.5	6.0

Multiplicative

1.0	4.0	5.0	2.0
2.0	3.0	7.0	2.5
4.0	6.0	8.0	5.0
3.0	8.0	9.0	4.0

Coherent evolution

Fig. 3. Various bicluster patterns including constant, constant rows, constant columns, coherent values with an additive model, coherent values with a multiplicative model, and coherent evolution values in columns.

Although the rows of a bicluster can be simply extracted using a conventional clustering method when the feature subset is determined, it is not easy to find the feature subset for a specific bicluster. For instance, there are 2^L possible feature subsets when the full size of features is L. If L is relatively large, exhaustively searching for these feature subsets will take a very large amount of computation time. Instead of searching for all possible feature subsets, we propose a new biclustering algorithm involving three rapid procedures, i.e., 1) discovery of bicluster seeds by applying the clustering to each feature; 2) heuristic formation of biclusters; and 3) removal of redundant biclusters.

In the first procedure, we detect the clusters of the elements in each of the columns. As demonstrated in Figs. 2 and 3, the elements under a single column in the submatrix of a bicluster are approximately the same with a small variance and, hence, can be found by a directly clustering method. In this paper, the clustered elements under a single column can be thought of as being potentially associated with a single or multiple biclusters. A cluster detected in a single column is called a bicluster seed in this paper. Thus, given a data matrix M with n_r rows and n_c columns, we first apply an agglomerative HC method using the average linkage for clustering all of the elements under every column and then obtain a preliminary set of bicluster seeds, as formulated by

$$[C_s(i,j), N_{cl}(j)] = HC(j, T_d), \qquad j = 1 \cdots n_c \tag{1}$$

$$BS\_set = \{C_s(i,j | i = 1 \cdots N_{cl}(j), j = 1 \cdots n_c\} \tag{2}$$

where $HC(j, T_d)$ is the HC algorithm applied to the elements under the jth colomn with a preset distance threshold T_d, $N_{cl}(j)$ denotes the number of clusters for the jth column, $C_s(i,j)$ is the ith bicluster seed under the jth column, and BS_set is the aggregation of the bicluster seeds detected from all columns. The time complexity of this procedure on each single column is $O(Ln_r^2)$.

As aforementioned, each of the detected bicluster seeds in BS_set is regarded as a potential part of some unknown biclusters. In the second procedure, we need to form larger biclusters from these small bicluster seeds and refine these large biclusters according to a predefined criterion. An algorithm including three steps is proposed as follows.

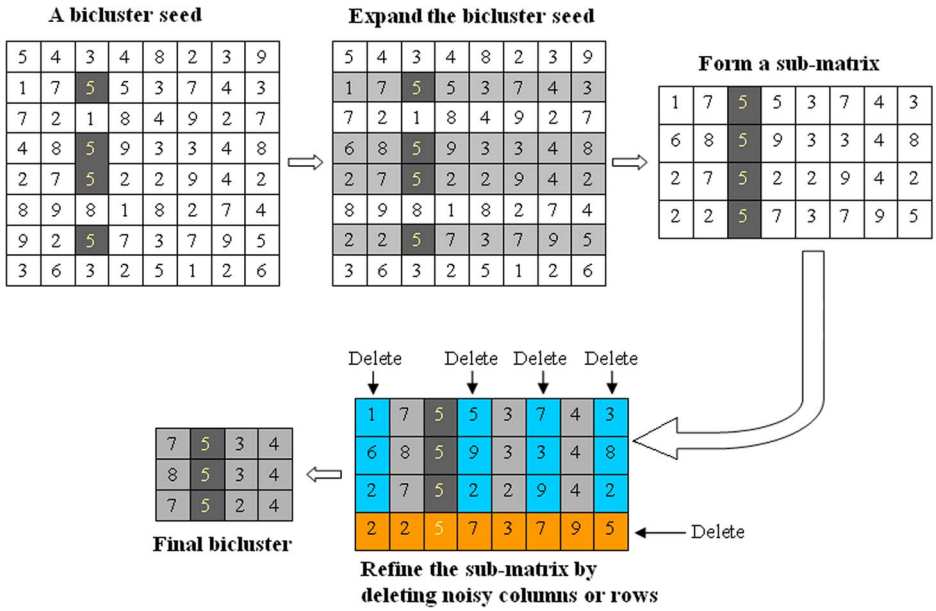

Fig. 4. Example for illustrating the procedure of expanding a bicluster seed and refining the expanded submatrix into a real bicluster.

(i) According to the number of rows, the bicluster seeds in BS_set are sorted in an ascending order.
(ii) Beginning with the bicluster seed with the lowest row number, each of the bicluster seeds in BS_set is then expanded along the column dimension. Given a bicluster seed with R_j rows, a new submatrix Ms can be formed with R_j rows and all of the n_c columns.
(iii) An optimization algorithm is finally required to find the largest bicluster that meets a certain homogeneity criterion in Ms.

In the step (iii), the mean-square-residue (MSR) score [16], which has been widely used as a metric for measuring the homogeneity of a bicluster, is employed as the homogeneity criterion. Given a submatrix with R rows and C columns, its MSR score is defined by

$$h(R,C) = \frac{1}{|R|\cdot|C|}\sum_{i\in R, j\in C}(e_{ij} - e_{iC} - e_{Rj} + e_{RC})^2$$

$$e_{iC} = \frac{1}{|C|}\sum_{j\in C} e_{ij}, \quad e_{Gj} = \frac{1}{|R|}\sum_{i\in R} e_{ij}$$

$$e_{RC} = \frac{1}{|R|\cdot|C|}\sum_{i\in R, j\in C} e_{ij}$$

$$\text{If } h(R,C) \le \delta, \text{ accept it as a valid bicluster} \qquad (3)$$

where e_{ij} denotes the element value at the ith row and jth column in the bicluster, δ is a homogeneity threshold defining the maximum allowable dissimilarity within the elements of the bicluster, and $h(R,C)$ is the value of the MSR score for the bicluster. The homogeneity threshold is set by users according to their respective applications.

The task of step (iii) is fulfilled based on the MSR score. A local search algorithm is then designed to find the largest bicluster in Ms. For a submatrix Ms, defining an array of nodes denoting its rows and columns, the search is performed by iteratively deleting the nodes that mostly contributes to the MSR score of Ms until the MSR score of the shrunken submatrix is no larger than a predefined homogeneity threshold T_m. More specifically, it starts with every Ms associated with the clusters in BS_set and consists of the following steps.

(a) Input a submatrix M.
(b) Set an array of the nodes denoting all of the rows and columns of M.
(c) For node i, calculate the MSR score for a new submatrix M^i_{new} in which the node i has been deleted from M. After each node in M has been considered, a set of new submatrices, Set_M_{new} and the corresponding MSR scores are recorded.
(d) Delete the node j corresponding to the new submatrix M^j_{new} that has the smallest MSR score in Set_M_{new}, and set the M^j_{new} as M.
(e) If the MSR score for M is larger than a predefined value T_m, repeat step (ii). Otherwise, output M as the largest bicluster.

The algorithm proposed for the second procedure is applied to each of the bicluster seeds in BS_set and the output biclusters are put into a new bicluster set, BC_set. This procedure is summarized and demonstrated in Fig. 4. The complexity of this local search algorithm is $O(dn^2)$, where d is the number of clusters in BS_set, and n is the number of both rows and columns.

The third procedure gets rid of all redundant biclusters that are fully covered by larger ones. This procedure is needed because a redundant bicluster that is part of a larger bicluster would lead to repetitive measures of the instance seperability and feature correlations of its features. First, the biclusters are ranked in BC_set with respect to their column numbers in ascending order. Second, the sorted biclusters are put into a new bicluster set s_BC. Starting from the bicluster ranking at the second place in s_BC, a bicluster is deleted if it is actually a

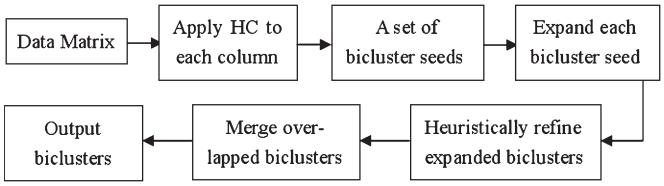

Fig. 5. Diagram for illustrating the biclustering algorithm.

submatrix of another bicluster ranked at a lower place. Third, till there is no bicluster that can be deleted in s_BC, the remaining is the final output. The complexity of this procedure is $O(n^2)$. This biclustering algorithm is summarized in Fig. 5.

Instead of exhaustively or heuristically searching for feature subsets in conventional wrapper methods, the procedure of searching for the column combination of a bicluster in our method is converted into a heuristic refining a submatrix. Considering all of the three procedures, the computational complexity of the proposed biclustering algorithm is much decreased. Thus, our algorithm is much more efficient without the need to repeatedly perform a clustering algorithm to evaluate every new column combination.

It is noteworthy that the values of an instance may greatly vary under different features. An HC method with a fixed T_d may be able to detect a cluster with a large number of elements under a specific feature but unable to detect a cluster under another feature if T_d is much smaller than the feature's variance. Therefore, we use the following method to normalize each column to ensure that most of the values in each column fall into a limited range:

$$e_n(i,j) = \frac{e(i,j) - mean\,(e(\cdot,j))}{2std\,(e(\cdot,j))}, \qquad i = 1 \ldots n_r; \quad j = 1 \ldots n_c \quad (4)$$

where $e(i,j)$ is the element value at the ith row and jth column, $mean(e(\cdot,j))$ denotes the mean of the elements under the jth column, $std(e(\cdot,j))$ is the standard deviation of the jth column, and $e_n(i,j)$ is the normalized element value. After the normalization of data values, the distance threshold T_d and the homogeneity threshold T_m are fixedly set to 0.01 and 0.02, respectively, in this paper.

C. Feature Ranking Scheme

Once the biclusters have been found from the data matrix, we need to extract information from them that can be used to evaluate each of the features. As motivated by the two factors (i.e., feature correlation and instance separability) mentioned earlier, a scoring scheme (called bicluster score) is first proposed by considering both factors in this paper. We define two subsidiary scores that stand for the two factors, respectively, i.e., the *correlation score*, which measures the correlations among features in a feature subset, and the *separability score*, which measures the separability of a feature. For the kth feature, suppose that it appears in any one of the biclusters from a bicluster subset Z_k, the two scores (denoted as Cor_Score and Sep_Score, respectively) are defined as follows:

$$Cor\_Score(k) = \sum_{i=1}^{n_{b,k}} \frac{n_{f,k}(i)}{n_c} \quad (5)$$

$$Sep\_Score(k) = \frac{n_{s,k}}{n_r} \sqrt{\sum_{i=1}^{n_{b,k}} (\mu_{i,k} - \mu_{a,k})^2 / n_{b,k}} \quad (6)$$

where $n_{b,k}$ denotes the number of biclusters in Z_k, $n_{f,k}(i)$ is the number of features for the ith bicluster in Z_k, $n_{s,k}$ is the number of the rows enumerated from all of the biclusters in Z_k, $\mu_{i,k}$ is the element average for the ith bicluster in Z_k under the kth feature, and $\mu_{a,k}$ is the average of $\mu_{i,k}$, $i = 1, \ldots, n_{b,k}$.

It is observed that in Cor_Score, $n_{f,k}(i)/n_c$ is the ratio of the number of columns for the ith bicluster in Z_k to the full length of columns. The Cor_Score actually equals the summation of the ratios. If a feature is associated with a larger number of biclusters, and/or the column dimensions of these biclusters cover a larger portion of the full size of the dimension, the corresponding Cor_Score is larger and vice versa. In Sep_Score, $n_{s,k}/n_r$ denotes the ratio of the instances that can be clustered by the biclustering algorithm to the full number of instances, and $\sqrt{\sum_{i=1}^{n_{b,k}} (\mu_{i,k} - \mu_{a,k})^2 / n_{b,k}}$ is the squared variance of the cluster centers for the kth feature. The larger the ratio and/or the variance are, the larger the Sep_Score for the feature is.

Finally, the bicluster score (denoted as Bic_Score) for the kth feature is obtained by considering both of the two subsidiary scores and is expressed as

$$Bic\_Score(k) = \alpha \cdot \overline{Cor\_Score(k)} + \overline{Sep\_Score(k)} \quad (7)$$

where $\overline{Cor\_Score(k)}$ and $\overline{Sep\_Score(k)}$ denote the normalized values for $Cor\_Score(k)$ and $Sep\_Score(k)$, $k = 1, \ldots, n_c$, respectively, and α is a regulation coefficient for balancing the contributions of the Cor_Score and the Sep_Score to the final Bic_Score. The features with higher Bic_Score are viewed as being better at characterizing the data clusters and linking with other features.

III. Experiments

To evaluate the performance of the proposed feature ranking algorithm, we conduct experiments using several standard data sets and make the comparison with three often-used feature selection algorithms: variance score [18], Laplacian score [19], and Fisher score [18]. The former two methods are unsupervised, while the Fisher score is supervised.

The method of variance uses the variance of instances for each of the features as a measure to evaluate the separability. For a given feature f and the instance values $e(i,f)$, $i = 1, \ldots, n_r$, $f = 1, \ldots, n_c$, the variance score is defined by

$$VS_f = \frac{1}{n_r} \sum_{i=1}^{n_r} (e(i,f) - \mu_f)^2, \qquad \mu_f = \frac{1}{n_r} \sum_{i=1}^{n_r} e(i,f). \quad (8)$$

The Laplacian score is based on Laplacian eigenmaps and locality-preserving projection. It prefers to selecting features with strong locality-preserving power. The computation of Laplacian score is expressed as follows:

$$LS_f = \frac{\sum_{i,j} \left(e(i,f) - e(j,f)\right)^2 S_{ij}}{\sum_i \left(e(i,f) - \mu_f\right)^2 D_{ii}},$$

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, & \text{if } x_i \text{ and } x_j \text{ are neighbors, } D_{ii} = \sum_j S_{ij} \\ 0, & \text{Otherwise} \end{cases} \tag{9}$$

where μ_f is the mean of the instances under the feature f, t is a constant parameter, and "x_i and x_j are neighbors" means that either x_i belongs to the k-nearest neighbors of x_j or vice versa.

The Fisher score can be grouped into the category of supervised feature selection algorithms as it uses class labels and ranks features according to their discriminant ability. For the fth feature, let $n_l(i)$ denote the number of instances for the ith class, $i = 1, \ldots, c$, and μ_f^i and σ_f^i be the mean and variance of the ith class, respectively. The Fisher score for the fth feature is calculated by

$$FS_f = \frac{\sum_{i=1}^{c} n_l(i) \left(\mu_f^i - \mu_f\right)^2}{\sum_{i=1}^{c} n_l(i) \left(\sigma_f^i\right)^2}. \tag{10}$$

A. UCI Data Sets and the Classifier

In this paper, we use three real-world data sets downloaded from the UCI database [24]. They are the wine data, the Wisconsin diagnostic breast cancer (wdbc) data and the congressional voting records (House-Votes-84) data. The wine data set has 13 features and 178 instances categorized into three groups. The instances are wines, and the features are chemical components. The wdbc data has 569 instances and 30 features. It contains two groups, i.e., benign and malignant breast tumors. The House-Votes-84 data has 435 instances, which are congressmen, and grouped into two parties, i.e., republican and democrat. The features are 16 votes for different topics. An affirmative vote is denoted as 1, a negative vote is denoted as -1, and an abstaining vote is denoted as 0.

In the experiments, we can generate a pair of training and testing sets by randomly selecting half of instances from all classes as the training set and setting the remaining half as the testing set. For each UCI data set, 20 pairs of training and testing sets are generated. Different feature selection algorithms are then applied to the testing sets. The features are ranked according to their scores computed by each algorithm. The feature number can be preset by users. With a predetermined feature number, the nearest neighborhood (1-NN) method with Euclidean distance is used as a classifier to obtain the classification accuracy on the testing data under the corresponding feature subset. Following the experimental methods used in [26] and [27], we evaluate the bicluster score by comparing the classification accuracy obtained by different feature selection algorithms. For the bicluster score, we set the parameter α in (7) to 1.0 when comparing with the other three algorithms.

TABLE I
AVERAGED ACCURACY (IN PERCENTAGE) OF DIFFERENT ALGORITHMS USING THE WINE DATA

Data	Bicluster	Fisher	Variance	Laplacian
Set 1	80.71±5.77	70.6±2.48	69.76±2.37	69.29±2.36
Set 2	87.36±8.59	71.91±1.92	71.53±1.94	71.35±1.76
Set 3	86.89±7.24	73.78±1.54	74.53±2.26	74.44±2.25

TABLE II
AVERAGED ACCURACY (IN PERCENTAGE) OF DIFFERENT ALGORITHMS USING THE WDBC DATA

Data	Bicluster	Fisher	Variance	Laplacian
Set 1	80.88±8.74	88.73±3.98	75.02±2.21	74.99±2.21
Set 2	80.33±9.44	88.63±4.15	73.83±2.93	73.79±2.93
Set 3	79.11±5.44	89.16±4.11	75.86±3.01	75.79±3.03

TABLE III
AVERAGED ACCURACY (IN PERCENTAGE) OF DIFFERENT ALGORITHMS USING THE HOUSE-VOTES-84 DATA

Data	Bicluster	Fisher	Variance	Laplacian
Set 1	95.17±0.88	93.92±1.57	87.16±15.22	92.97±3.51
Set 2	95.38±1.00	93.85±2.52	89.05±10.72	92.99±3.32
Set 3	93.0±1.25	91.32±1.49	86.09±14.0	89.73±3.59

B. Results for UCI Data Sets

Because of the limited length for this paper, we select three pairs of training and testing data sets for each of the three UCI data and present in detail the simulation results using the selected data in Tables I–III and Figs. 6–8. Thereafter, we summarize the feature selection performance of different algorithms and present the overall comparison results using all of the 20 pairs of data sets for each UCI data in Fig. 9.

Table I shows the averaged accuracy values of the four algorithms using three pairs of training and testing wine data. The accuracy values versus the number of removed features for the three data sets can also be seen in Fig. 6. It is obvious that the bicluster score significantly outperforms the others. The reason can be explained by the intrinsic properties of the features in wine data. Because the features are chemical components contained in the wines, the density of one component can influence those of the other components. Thus, it is concluded that there are strong interdependencies among the features. As stated earlier, our algorithm is good at discovering feature interdependencies and, hence, can achieve the best results for the wine data. From Fig. 6(d), the performance of bicluster score is approximately improved as the balancing parameter α is increasing. It implies that considering

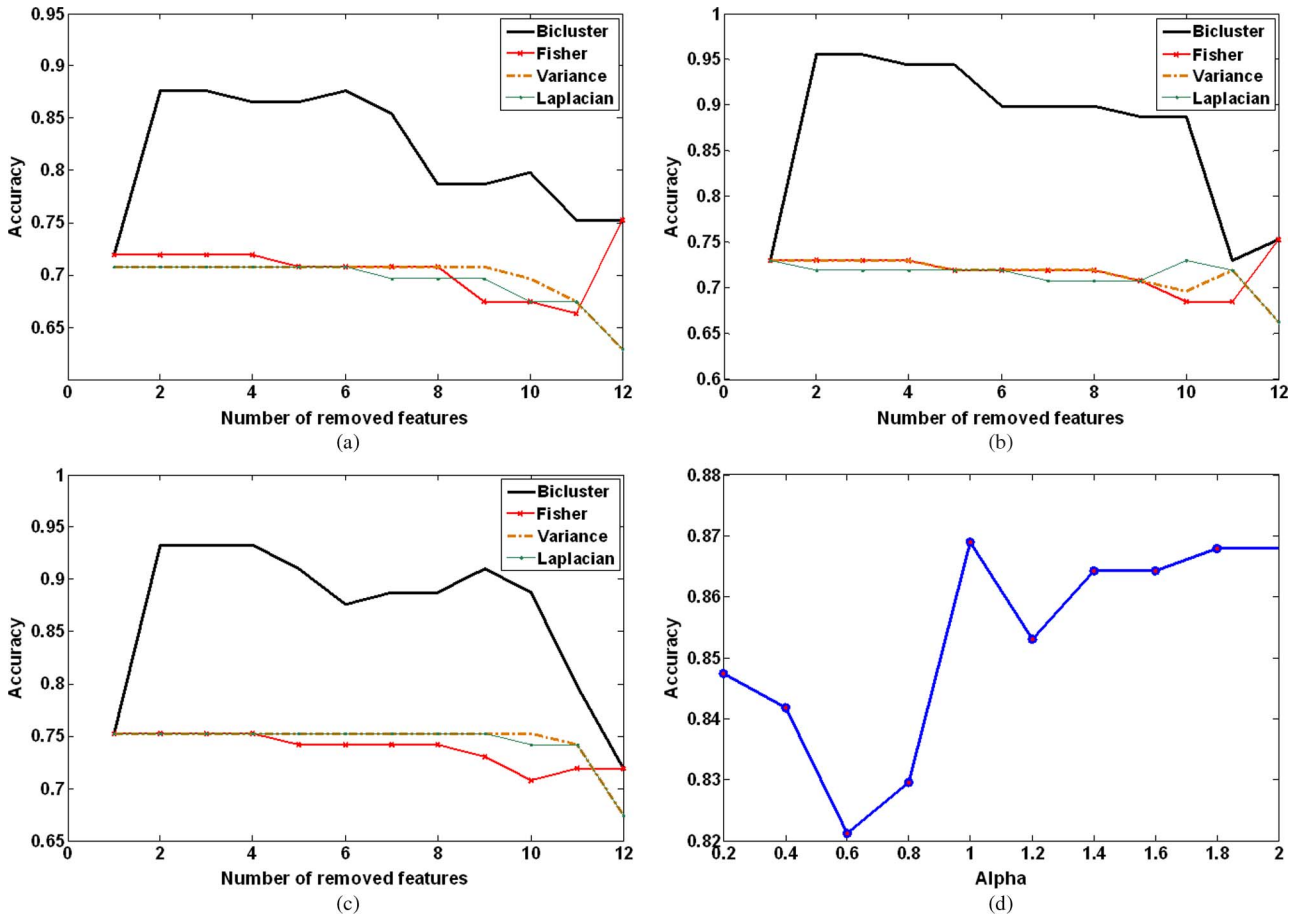

Fig. 6. Feature selection performance on the wine data sets. (a) Data set 1. (b) Data set 2. (c) Data set 3. (d) Accuracy against α for the bicluster score on data set 3.

interdependencies of features is able to improve the feature selection performance.

The simulation results for the wdbc data are summarized in Table II and can be seen in Fig. 7. According to the results, the Fisher Score significantly outperforms the others. Nevertheless, our algorithm performs better than the variance and Laplacian scores, indicating an improved performance of feature selection for the data without class labels. According to the description for wdbc data, the features are some quantities (such as area, smoothness, and dimensions) measured from breast tumor regions. Because these measures are performed from very different aspects, it can be concluded that the interdependencies among the 30 features are relatively weak; thus, the Cor_Score cannot effectively influence the feature orders. This explanation can be further proved as shown in Fig. 7(d). It is noted that the varying values for the balancing parameter α cannot significantly change the classification accuracy, indicating a relatively weak effect on the feature selection.

Table III summarizes the averaged accuracy obtained by different algorithms on the House-Votes-84 data. For these data, our algorithm outperforms the others. As illustrated in Fig. 8, the bicluster score achieves comparable (even better) classification accuracy to that of the Fisher score and much better results than the other two unsupervised methods. We can conclude that there exist interrelationships among the features, which are actually proposals in various economic and political fields. Different proposals are likely to have overlapped parts focused by the public. In addition, a group of congressmen from a specific party may have positive views on a subgroup of proposals and negative views on another subgroup. Hence, there exist many subsets of features that internally influence each other. Our algorithm is able to discover these feature subsets; meanwhile, the interactions among these proposals to be voted can accordingly help in differentiating the congressmen's political stands. The influence of different values of α is shown in Fig. 8(d) and provides further support to this point. It can be seen that the total classification accuracy keeps increasing as α is increased.

Fig. 9 illustrates the overall comparison of feature selection performance for the four different algorithms using the UCI data sets. The bicluster score is compared individually with the other three methods. In each subfigure, the number corresponding to the bicluster score means the number of simulation data pairs in which our method outperforms its counterpart in terms of the overall classification accuracy. It is observed that our algorithm outperforms the others for the data of wine and House-Vote-84 and achieves comparable results with the variance and Laplacian scores for the data of wdbc.

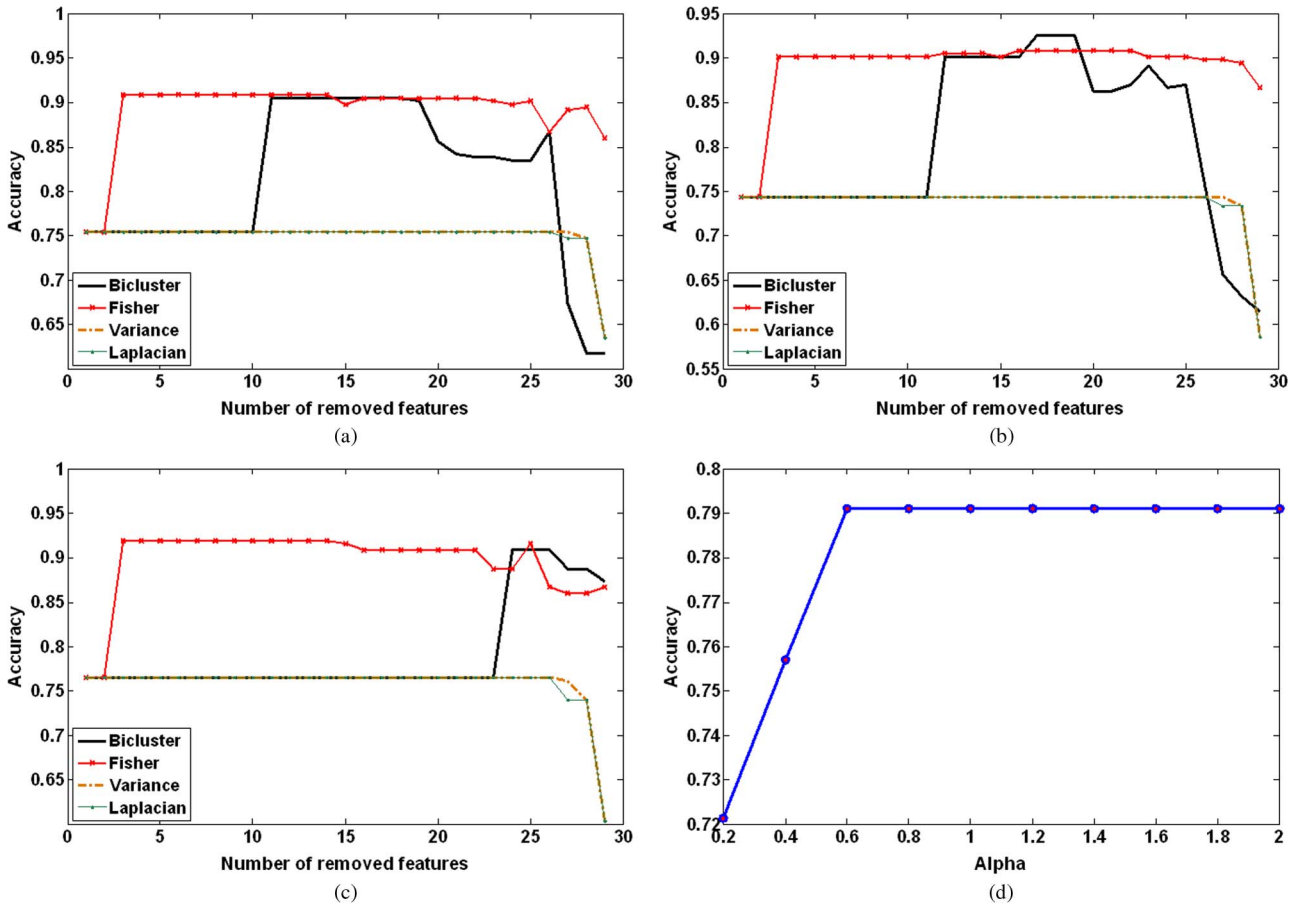

Fig. 7. Feature selection performance on the wdbc data sets. (a) Data set 1. (b) Data set 2. (c) Data set 3. (d) Accuracy against α for the bicluster score on data set 3.

To unveil the influence of the bicluster mergence procedure introduced in Section II-B, we compare the feature ranking performance of the proposed merging method with different overlap degrees. We define the overlap degree between two overlapping biclusters as the ratio of the overlapping part to the full size of the smaller bicluster. If the overlap degree for any two of biclusters is no less than a preset threshold, the two biclusters should be merged into a larger bicluster. The new bicluster should be then heuristically refined using the local search algorithm introduced in Section II-B. By selecting the data sets shown in Figs. 6–8, the bicluster scores are recomputed by setting the threshold of overlap degree between any two of remaining biclusters to be 100%, 65%, 30%, and 0%, respectively. Table IV shows the comparisons of the classification accuracy using different overlap degrees. The bicluster score with larger thresholds of overlap degree in the procedure of bicluster mergence achieves better classification accuracy. It indicates that the mergence of biclusters with a high overlap degree can be relatively more helpful in improving the performance of feature ranking.

C. Results for Real Microarray Data

In the field of bioinformatics, the availability of gene expression profiles under various experimental conditions corresponding to different biological processes has led to fruitful applications of well-established pattern classification algorithms. Lots of attention has been paid to offering more accurate and automatic pattern analysis of gene profiles for revealing real biological pathways [17]. However, the microarray data arranged as a data matrix can often be viewed as a high-dimensional data set due to the large amount of genes and conditions contained, suffering the effect of the curse of dimensionality, which leads to a challenge for mining meaningful biological patterns.

In addition to UCI data, we evaluate our algorithm using real microarray data in this paper. The gene expression data set of *S.cerevisiae* provided by Gasch *et al.* [28] is used. The data contains 2993 genes and 173 experimental conditions. We perform different feature selection algorithms to rank the experimental conditions, and follow the feature selection assessment method proposed by Zhu *et al.* [14]. They performed a one-versus-all strategy, where a group of genes associated with a selected gene ontology (GO) annotation is viewed as a relevant class (positive class), and the other genes are viewed as belonging to an irrelevant class (negative class) unassociated with the given GO annotation. In our experiments, an online GO analysis tool, i.e., MIPS [29], is used for assigning the genes to corresponding biological GO function categories. In the experiment, we generate two data

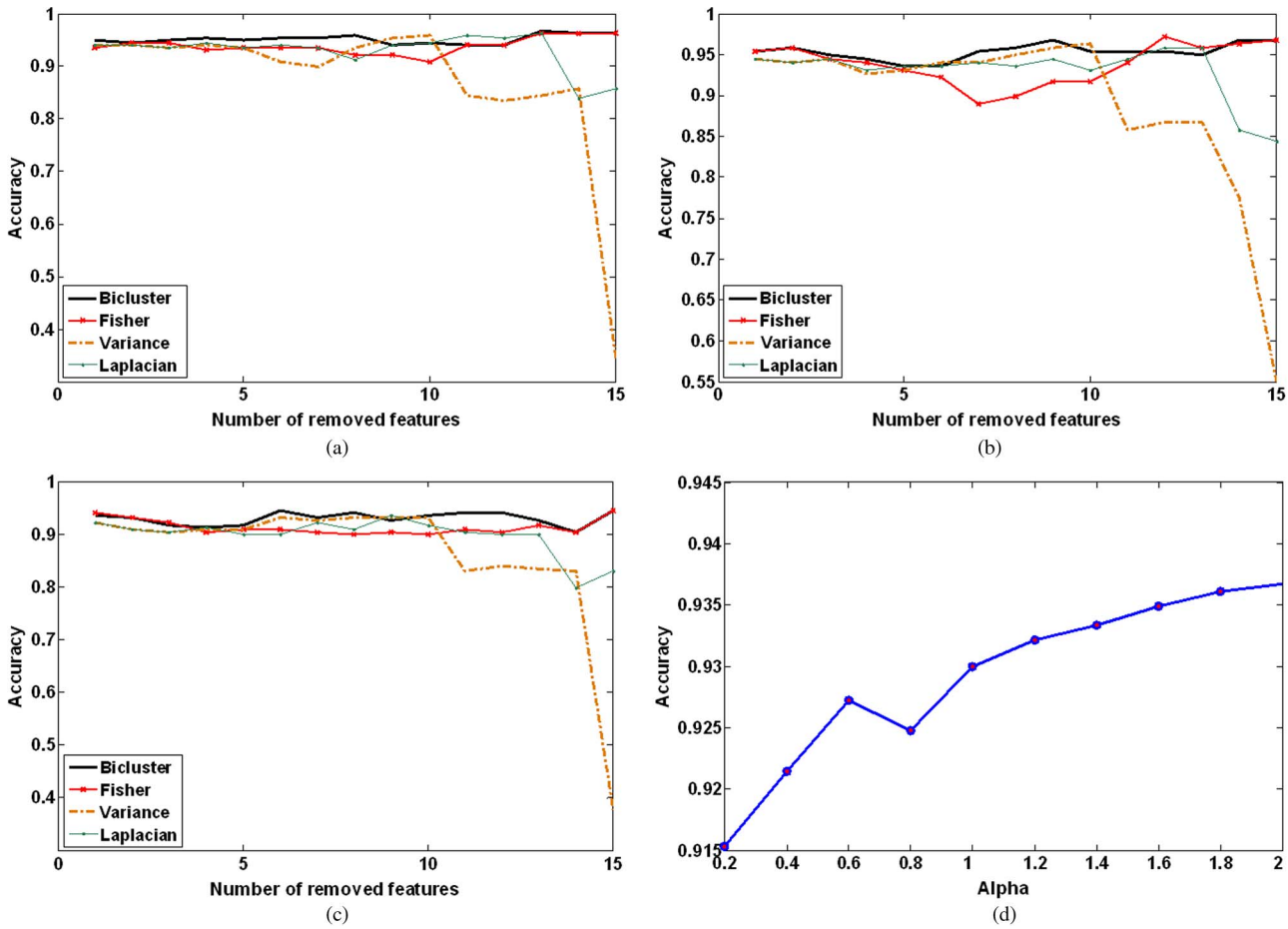

Fig. 8. Feature selection performance on the House-Votes-84 data sets. (a) Data set 1. (b) Data set 2. (c) Data set 3. d) Accuracy against α for the bicluster score on data set 3.

sets, each of which contains 1000 randomly chosen genes with all conditions. The genes in each data set are then input into the MIPS. Table V presents the specified gene function and the number of genes in association with the function for each data set.

Using the 1-NN classifier, we evaluate the classification accuracy and error rates for different feature selection methods when the condition number decreases. The error rate is defined as the ratio of the number of negative genes incorrectly classified into the positive class to the full number of negative genes.

Figs. 10 and 11 illustrate the results using the two real data sets. Table VI summarizes the averaged accuracy and error rates obtained using the four different algorithms with the two data sets. It can be observed that the bicluster score obtains the best classification accuracy and relatively lower error rates, indicating good capability of selecting significant experimental conditions for classification of genes in a real microarray data. In comparison with the Fisher score, which is a supervised method, our algorithm, which is an unsupervised method, achieves comparable results and further validates its merit in selecting experimental conditions of microarray data. The results also imply that there exist interrelationships among the conditions (i.e., experimental stresses) in terms of the gene expression responses since the bicluster score has improved the performance of condition selection in comparison with the variance and Laplacian scores.

D. Results for High-Dimensional Data

To further illustrate the feature ranking performance of the proposed bicluster score, we apply the bicluster score to a publicly available high-dimensional data set, i.e., Gina, which can be downloaded from [30]. The Gina data set has 970 features and 3153 labeled instances. We randomly select 1000 instances to construct a training set and another 1000 instances as a testing set. The proposed bicluster score is applied on the training set to rank the features, and the 1-NN classifier is carried out to achieve the feature selection performance based on the testing set. Similarly, the Fisher, Laplacian, and variance scores are compared with the bicluster score using the same data sets. Fig. 12 illustrates the comparison results. It is observed that the Fisher score (a supervised method) achieves the best average classification accuracy, which is 85.82%. For the other three unsupervised methods, our bicluster score achieves the average accuracy of 78.34%, which is better than the variance and Laplacian scores, which achieve the average accuracy values of 75.61% and 75.20%, respectively. Thus, the performance improvement for unsupervised feature selection using the bicluster score can be proved with the high-dimensional data.

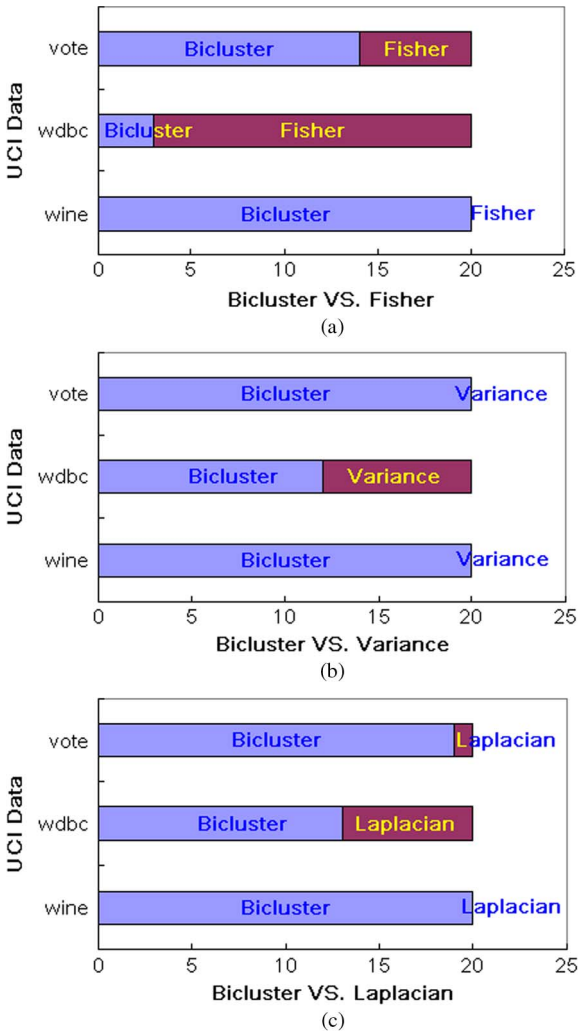

Fig. 9. Overall comparisons among the four algorithms using all of the simulation results in terms of the classification accuracy. (a) Bicluster score vs. Fisher score. (b) Bicluster score vs. variance score. (c) Bicluster score vs. Laplacian score.

TABLE IV
AVERAGED ACCURACY (IN PERCENTAGE) OF THE BICLUSTER SCORE WITH DIFFERENT OVERLAP DEGREES IN THE PROCEDURE OF BICLUSTER MERGENCE USING THE SELECTED UCI DATA SETS

Data		Overlap degree			
		100%	65%	30%	0%
wine	Set 1	80.71±5.77	81.24±6.03	76.07±6.83	74.72±6.51
	Set 2	87.36±8.59	85.62±7.12	77.49±8.13	76.87±7.70
	Set 3	86.89±7.24	86.30±7.71	75.73±7.01	73.88±5.78
wdbc	Set 1	80.88±8.74	80.06±8.85	77.12±6.89	77.05±6.63
	Set 2	80.33±9.44	79.54±7.94	77.44±5.70	76.58±6.82
	Set 3	79.11±5.44	79.67±7.31	78.35±7.52	77.12±7.06
house-vote-84	Set 1	95.17±0.88	95.15±0.94	92.73±1.40	90.34±1.03
	Set 2	95.38±1.00	94.78±0.86	92.55±1.52	90.98±1.24
	Set 3	93.0±1.25	93.0±1.25	89.36±1.29	88.34±1.43

TABLE V
NUMBER OF GENES IN THE POSITIVE CLASS AND THE GO CATEGORY DETERMINED FOR THE TWO YEAST MICROARRAY DATA SETS

Data	GO functional category	No. of genes in positive class
Set 1	10.03 cell cycle	152
Set 2	10.03 cell cycle	156

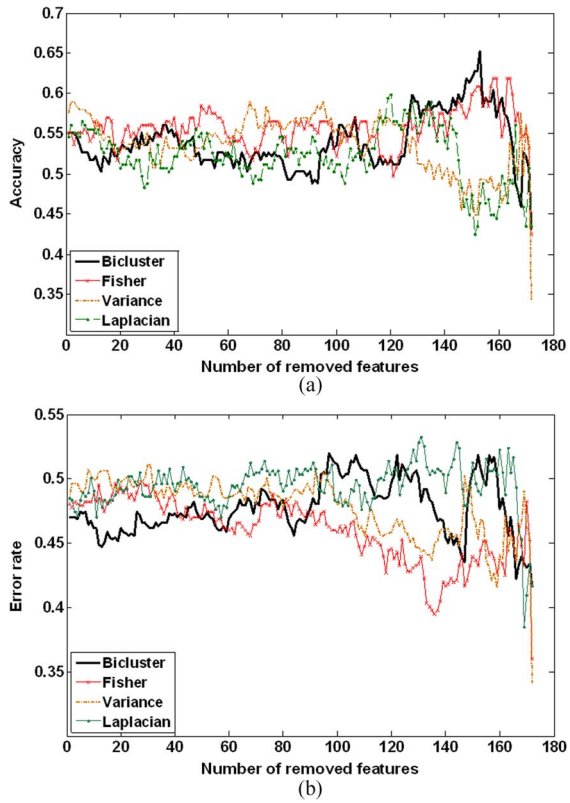

Fig. 10. Comparison of feature selection performance for four algorithms using real yeast microarray data set 1. (a) Accuracy versus the number of removed features. (b) Error rate versus the number of removed features.

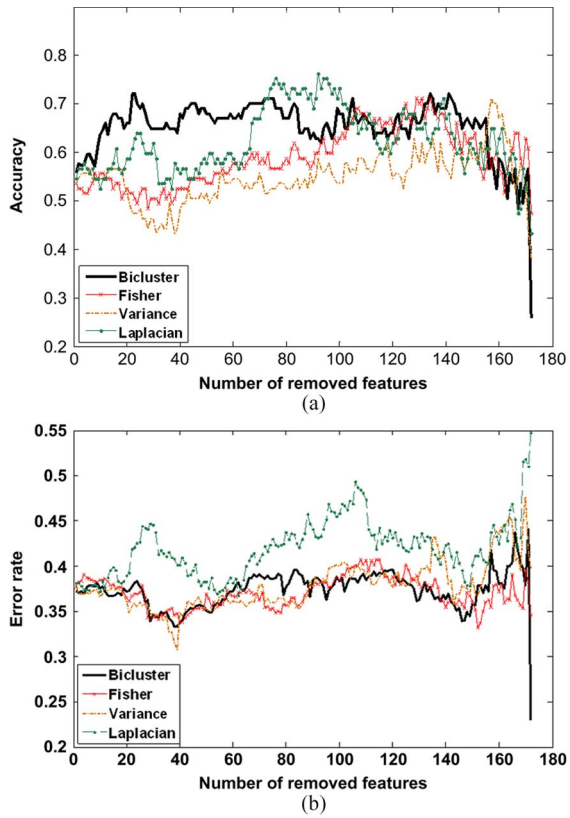

Fig. 11. Comparison of feature selection performance for four algorithms using real yeast microarray data set 2. (a) Accuracy versus the number of removed features. (b) Error rate versus the number of removed features.

TABLE VI
AVERAGED ACCURACY AND ERROR RATES (IN PERCENTAGE) OF DIFFERENT ALGORITHMS ON THE TWO MICROARRAY DATA SETS

Data	Bicluster		Fisher		Variance		Laplacian	
	Accuracy	Error rate	Accuracy	Error rate	Accuracy	Error rate	Accuracy	Error rate
Set 1	65.25±4.19	35.31±3.59	61.47±4.7	38.21±2.55	62.4±7.27	38.19±2.99	65.08±5.14	37.9±3.57
Set 2	65.81±5.32	37.37±2.12	58.94±5.85	37.04±1.7	54.87±5.29	37.81±2.67	62.47±6.62	41.97±3.23

IV. DISCUSSIONS AND CONCLUSION

In this paper, a novel unsupervised feature selection algorithm is proposed. This algorithm is based on an unsupervised biclustering algorithm that can discover local coherent patterns in a data matrix. The discovered local patterns including a subset of instances and a subset of features simultaneously reveal both the separability of instances and interdependencies among features. We propose a new scoring scheme, which is called bicluster score. Like a wrapper method, the bicluster score first discovers biclusters in a data matrix, then calculates two subsidiary scores by considering the clustered instances and features for each bicluster and finally computes the bicluster score by summing the two subsidiary scores.

The experimental results using three UCI data sets demonstrate that the bicluster score can outperform two often-used unsupervised feature ranking algorithm and produce comparable or even better results than the Fisher score, which is a supervised method. In particular, our algorithm significantly outperforms the other three algorithms using the wine data set, indicating that the features (chemical components) have relatively strong correlations. In contrast, for the wdbc data, the Fisher score demonstrates the best performance on feature selection, showing that the features extracted from breast tumor images are relatively independent of each other. As a result, our algorithm is unable to improve the classification results by considering the correlations among the features. For the House-Votes-84 data set, our algorithm generates the best results, illustrating the features (proposals to be voted) are interrelated to each other, and some of them have similar influences on a subgroup of congressmen from a specific political organization. In addition, the results using the real microarray data illustrates that the bicluster score can yield comparable or even better feature selection performance in comparison with the other three algorithms, presenting good merit in practice. Finally, we evaluate the feature ranking performance of bicluster score using a high-dimensional data set, i.e., Gina. By comparing the four methods, the bicluster score outperforms the other two unsupervised methods. Although the Fisher score achieves the best results, the bicluster score has demonstrated its advantage in unsupervised applications.

However, a drawback of bicluster score should be the increased computational expense. In our experiments, the computational time for the bicluster score is usually five to ten times as much as those of its counterparts. The acceleration of the computation of bicluster score is worth being a future research topic. Another drawback is the difficulty of choosing an appropriate α in the bicluster score for a data set without any prior knowledge. According to the motivation of the bicluster score, we aim to rank the features by taking into account not only the separability but also the feature correlations. A well-selected α should be able to perfectly balance the Cor_Score and Sep_Score. It is reasonable that prior knowledge of the feature correlations in a data set would be useful for the determination of α. In our future study, various data sets with some prior knowledge for feature correlations will be carefully tested to find out the methods for the optimal selection of α.

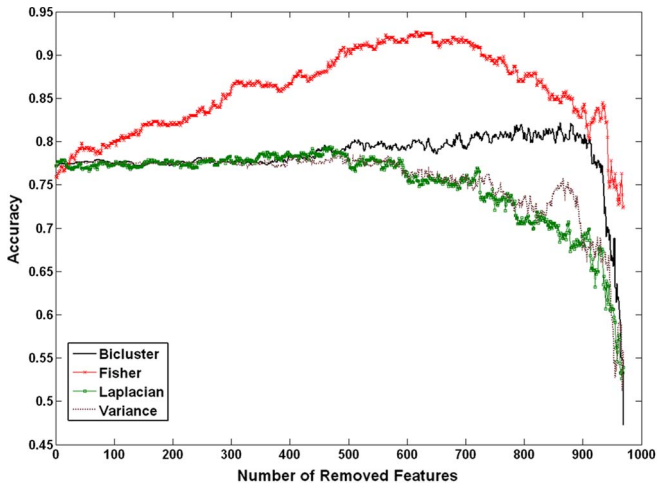

Fig. 12. Comparison of feature selection performance for the four algorithms using a high-dimensional data set, i.e., Gina.

It is also worth noting that the bicluster score is actually making use of a support-based pruning strategy and needs several supporting parameters, e.g., T_m and T_d. Different support levels for the parameters would significantly affect the final results. Recently, Xiong *et al.* [31] have proposed a framework for mining highly correlated association patterns called hyperclique patterns, which can overcome the problem of support-based pruning. Hence, incorporating the hyperclique patterns into the bicluster score will be part of our future research.

In summary, the results have demonstrated that the bicluster score is able to conduct feature selection on several UCI and real microarray data sets with good performance. By ranking the features according to the discovered biclusters, it can be expected to be suitable for various data sets, particularly the ones with strong interdependencies among the features.

REFERENCES

[1] T. W. S. Chow, P. Y. Wang, and E. W. M. Ma, "A new feature selection scheme using a data distribution factor for unsupervised nominal data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 499–509, Apr. 2008.

[2] Q. H. Hu, W. Pedrycz, D. R. Yu, and J. Lang, "Selecting discrete and continuous features based on neighborhood decision error minimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 1, pp. 137–150, Feb. 2010.

[3] X. L. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 342–352, Apr. 2008.

[4] A. K. Jain, R. P. W. Duin, and J. C. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[5] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[6] X. L. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2D PCA," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 4, pp. 1170–1175, Aug. 2010.

[7] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.: Int. J.*, vol. 1, no. 3, pp. 131–156, 1997.
[8] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.
[9] M. H. Asyali, D. Colak, O. Demirkaya, and M. S. Inan, "Gene expression profile classification: A review," *Curr. Bioinform.*, vol. 1, no. 1, pp. 55–73, Jan. 2006.
[10] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Boston, MA: Kluwer, 2001.
[11] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1/2, pp. 273–324, Dec. 1997.
[12] J. G. Dy, C. E. Brodley, A. Kak, L. S. Broderick, and A. M. Aisen, "Unsupervised feature selection applied to content-based retrieval of lung images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 3, pp. 373–378, Mar. 2003.
[13] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, Nov. 1994.
[14] Z. X. Zhu, Y. S. Ong, K. W. Wong, and K. T. Seow, "Experimental condition selection in whole-genome functional classification," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Singapore, 2004, pp. 295–300.
[15] H. C. Peng, F. H. Long, and C. Ding, "Feature selection based on mutual information: Criteria of Max-dependency, Max-relevance and Min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[16] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proc. 8th Int. Conf. ISMB*, 2000, pp. 93–103.
[17] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 1, no. 1, pp. 24–45, Jan.–Mar. 2004.
[18] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
[19] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in Neural Information Processing Systems, 17*. Cambridge, MA: MIT Press, 2005.
[20] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002.
[21] H. L. Wei and S. A. Billings, "Feature subset selection and ranking for data dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 162–166, Jan. 2007.
[22] M. Law, M. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1154–1166, Sep. 2004.
[23] B. Liu, C. Wan, and L. Wang, "An efficient semi-supervised gene selection via spectral biclustering," *IEEE Trans. Nanobiosci.*, vol. 5, no. 2, pp. 110–114, Jun. 2006.
[24] [Online]. Available: http://www.ics.uci.edu/~mlearn/MLSummary.html
[25] X. Xu, Y. Lu, A. K. H. Tung, and W. Wang, "Mining shifting-and-scaling co-regulation patterns on gene expression profiles," in *Proc. 22nd Int. Conf. Data Eng.*, 2006, p. 89.
[26] D. Q. Zhang, S. C. Chen, and Z. H. Zhou, "Constraint Score: A new filter method for feature selection with pairwise constraints," *Pattern Recognit.*, vol. 41, no. 5, pp. 1440–1451, May 2008.
[27] J. N. Liang, S. Yang, and A. Winstanley, "Invariant optimal feature selection: A distance discriminant and feature ranking based solution," *Pattern Recognit.*, vol. 41, no. 5, pp. 1429–1439, May 2008.
[28] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown, "Genomic expression programs in the response of Yeast cells to environmental changes," *Mol. Biol. Cell*, vol. 11, no. 12, pp. 4241–4257, Dec. 2000.
[29] [Online]. Available: http://mips.gsf.de/proj/yeast/catalogues/funcat
[30] [Online]. Available: http://clopinet.com/isabelle/Projects/modelselect/
[31] H. Xiong, P. N. Tan, and V. Kumar, "Hyperclique pattern discovery," *Data Mining Knowl. Discov.*, vol. 13, no. 2, pp. 219–242, Sep. 2006.
[32] X. L. Li and Y. Pang, "Deterministic column-based matrix decomposition," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 1, pp. 145–149, Jan. 2010.
[33] Y. Yuan, X. L. Li, Y. Pang, X. Lu, and D. Tao, "Binary sparse nonnegative matrix factorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 772–777, May 2009.

Qinghua Huang received the B.E. and M.E. degrees in automatic control and pattern recognition from the University of Science and Technology of China, Hefei, China, in 1999 and 2002, respectively, and the Ph.D. degree in biomedical engineering from the Hong Kong Polytechnic University in 2007.

He is currently an Associate Professor with the School of Electronic and Information Engineering, South China University of Technology. His research interests include ultrasonic imaging, medical image analysis, bioinformatics, and intelligent computation and its applications.

Dacheng Tao (M'07) is currently a Professor with the Centre for Quantum Computation and Information Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW, Australia. He mainly applies statistics and mathematics for data analysis problems in data mining, computer vision, machine learning, multimedia, and video surveillance. He is the author or a coauthor of more than 100 scientific articles at top venues, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON ON IMAGE PROCESSING, *Advances in Neural Information Processing Systems*, International conference on Artificial Intelligence and Statistics (AISTATS), Conference on Artificial Intelligence (AAAI), EEE Conference on Computer Vision and Pattern Recognition (CVPR), European Conference on Computer Vision (ECCV), IEEE International Conference on Data Mining ICDM; *ACM Transactions on Knowledge Discovery from Data*, and ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), with best paper awards.

Xuelong Li (M'02–SM'07) is a Researcher (Full Professor) with the Center for Optical Imagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

Lianwen Jin received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 1991 and the Ph.D. degree from the South China University of Technology, Guangzhou, in 1996.

He is currently a Professor with the School of Electronic and Information Engineering, South China University of Technology. His research interests include character recognition, pattern analysis and recognition, image processing, machine learning, and intelligent systems. He is the author of more than 100 scientific papers.

Gang Wei was born in January 1963. He received the B.S. degree from Tsinghua University, Beijing, China, in 1984 and the M.S. and Ph.D. degrees from South China University of Technology (SCUT), Guangzhou, China, in 1987, and 1990, respectively.

He was a Visiting Scholar with the University of Southern California, Los Angeles, from June 1997 to June 1998. He is currently a Professor with the School of Electronic and Information Engineering, SCUT. His research interests are digital signal processing and communications.