Trace Ratio Criterion for Feature Selection

Feiping Nie[1], **Shiming Xiang**[1], **Yangqing Jia**[1], **Changshui Zhang**[1] **and Shuicheng Yan**[2]
[1] State Key Laboratory on Intelligent Technology and Systems,
[1] Tsinghua National Laboratory for Information Science and Technology (TNList),
[1] Department of Automation, Tsinghua University, Beijing 100084, China
[2] Department of Electrical and Computer Engineering, National University of Singapore, Singapore
{feipingnie, jiayq84}@gmail.com; {xsm, zcs}@mail.tsinghua.edu.cn; eleyans@nus.edu.sg

Abstract

Fisher score and Laplacian score are two popular feature selection algorithms, both of which belong to the general graph-based feature selection framework. In this framework, a feature subset is selected based on the corresponding score (subset-level score), which is calculated in a trace ratio form. Since the number of all possible feature subsets is very huge, it is often prohibitively expensive in computational cost to search in a brute force manner for the feature subset with the maximum subset-level score. Instead of calculating the scores of all the feature subsets, traditional methods calculate the score for each feature, and then select the leading features based on the rank of these feature-level scores. However, selecting the feature subset based on the feature-level score cannot guarantee the optimum of the subset-level score. In this paper, we directly optimize the subset-level score, and propose a novel algorithm to efficiently find the global optimal feature subset such that the subset-level score is maximized. Extensive experiments demonstrate the effectiveness of our proposed algorithm in comparison with the traditional methods for feature selection.

Introduction

Many classification tasks often need to deal with high-dimensional data. Data with a large number of features will result in higher computational cost, and the irrelevant and redundant features may also deteriorate the classification performance. Feature selection is one of the most important approaches for dealing with high-dimensional data (Guyon & Elisseeff 2003). According to the strategy of utilizing class label information, feature selection algorithms can be roughly divided into three categories, namely unsupervised feature selection (Dy & Brodley 2004), semi-supervised feature selection (Zhao & Liu 2007a), and supervised feature selection (Robnik-Sikonja & Kononenko 2003). These feature selection algorithms can also be categorized into wrappers and filters (Kohavi & John 1997; Das 2001). Wrappers are classifier-specific and the feature subset is selected directly based on the performance of a specific classifier. Filters are classifier-independent and the feature subset is selected based on a well-defined criterion. Usually, wrappers could obtain better results than filters because wrappers are directly related to the algorithmic performance of a specific classifier. However, wrappers are computationally more expensive compared with filters and lack of good generalization capability over classifiers.

Fisher score (Bishop 1995) and Laplacian score (He, Cai, & Niyogi 2005) are two popular filter-type methods for feature selection, and both belong to the general graph-based feature selection framework. In this framework, the feature subset is selected based on the score of the entire feature subset, and the score is calculated in a trace ratio form.

The trace ratio form has been successfully used as a general criterion for feature extraction previously (Nie, Xiang, & Zhang 2007; Wang *et al.* 2007). However, when the trace ratio criterion is applied for feature selection, since the number of possible subsets of features is very huge, it is often prohibitively expensive in computational cost to search in a brute force manner for the feature subset with the maximum subset-level score. Therefore, instead of calculating the subset-level score for all the feature subsets, traditional methods calculate the score of each feature (feature-level score), and then select the leading features based on the rank of these feature-level scores.

The selected subset of features based on the feature-level score is suboptimal, and cannot guarantee the optimum of the subset-level score. In this paper, we directly optimize the subset-level score, and propose a novel iterative algorithm to efficiently find the globally optimal feature subset such that the subset-level score is maximized. Experimental results on UCI datasets and two face datasets demonstrate the effectiveness of the proposed algorithm in comparison with the traditional methods for feature selection.

Feature Selection $\subset$ Subspace Learning

Suppose the original high-dimensional data $\boldsymbol{x} \in \mathbb{R}^d$, that is, the number of features (dimensions) of the data is d. The task of subspace learning is to find the optimal projection matrix $\mathbf{W} \in \mathbb{R}^{d \times m}$ (usually $m \ll d$) under an appropriate criterion, and then the d-dimensional data $\boldsymbol{x}$ is transformed to the m-dimensional data $\boldsymbol{y}$ by

$$\boldsymbol{y} = \mathbf{W}^T \boldsymbol{x}, \tag{1}$$

where $\mathbf{W}$ is a column-full-rank projection matrix.

When turning to feature selection, the task is simplified to find the optimal feature subset such that an appropriate criterion is optimized. Suppose m features are selected, then the data $\boldsymbol{x}$ with d-features is reduced to the data $\boldsymbol{y}$ with m features. If we use the matrix form, the feature selection procedure can be expressed as

$$\boldsymbol{y} = \mathbf{W}^T \boldsymbol{x}, \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{d \times m}$ is a selection matrix. Denote a column vector by $\boldsymbol{w}_i \in \mathbb{R}^d$ that has the form

$$\boldsymbol{w}_i = [\underbrace{0, \cdots, 0}_{i-1}, 1, \underbrace{0, \cdots, 0}_{d-i}]^T. \tag{3}$$

Then $\mathbf{W}$ in Equation (2) can be written as

$$\mathbf{W} = [\boldsymbol{w}_{I(1)}, \boldsymbol{w}_{I(2)}, ..., \boldsymbol{w}_{I(m)}], \tag{4}$$

where the vector I is a permutation of $\{1, 2, ..., d\}$.

From this point of view, feature selection can be seen as a special subspace learning task, where the projection matrix is constrained to be selection matrix. However, feature selection has its advantages over subspace learning: 1) owing to the special structure of $\mathbf{W}$, feature selection algorithm is often faster than the corresponding subspace learning algorithm; 2) the result of feature selection is explainable; and 3) after performing feature selection, we only need to produce a small subset of features for further data processing.

A General Graph-based Feature Selection Framework Under Trace Ratio Criterion

Let the data matrix be $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$, where each data $\boldsymbol{x}_i$ has d features denoted by $\{F_1, F_2, ..., F_d\}$. A feature subset $\{F_{I(1)}, F_{I(2)}, ..., F_{I(m)}\}$ is denoted as $\Phi(I)$, where I is a permutation of $\{1, 2, ..., d\}$. Similarly, we set $\mathbf{W}_I = [\boldsymbol{w}_{I(1)}, \boldsymbol{w}_{I(2)}, ..., \boldsymbol{w}_{I(m)}]$, where $\boldsymbol{w}_i$ is defined as the same as in Equation (3). Suppose the feature subset $\Phi(I)$ is selected, then the data $\boldsymbol{x}$ is transformed to $\boldsymbol{y}$ by $\boldsymbol{y} = \mathbf{W}_I^T \boldsymbol{x}$.

A graph is a natural and effective way to encode the relationship among data, and has been applied in many machine learning tasks, such as clustering (Shi & Malik 2000), manifold learning (Belkin & Niyogi 2003), semi-supervised learning (Zhu, Ghahramani, & Lafferty 2003), and subspace learning (He *et al.* 2005).

For the task of feature selection, we construct two weighted undirected graphs $\mathcal{G}_w$ and $\mathcal{G}_b$ on given data. Graph $\mathcal{G}_w$ reflects the within-class or local affinity relationship, and graph $\mathcal{G}_b$ reflects the between-class or global affinity relationship. Graphs $\mathcal{G}_w$ and $\mathcal{G}_b$ are characterized by the weight matrices $\mathbf{A}_w$ and $\mathbf{A}_b$, respectively.

In general, to reflect the within-class or local affinity relationship in data, $(\mathbf{A}_w)_{ij}$ is a relatively larger value if data $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same class or are close to each other, and a relatively smaller value otherwise. Therefore, we should select the feature subset such that $\sum_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 (\mathbf{A}_w)_{ij}$ is as small as possible.

Similarly, to reflect the between-class or global affinity relationship in data, $(\mathbf{A}_b)_{ij}$ is a relatively larger value if data $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the different classes or are distant from each other, and is a relatively smaller value otherwise. Therefore, we should select the feature subset such that $\sum_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 (\mathbf{A}_b)_{ij}$ is as large as possible.

To achieve the above two objectives, an appropriate criterion could be

$$\mathcal{J}(\mathbf{W}_I) = \frac{\sum_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 (\mathbf{A}_b)_{ij}}{\sum_{ij} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 (\mathbf{A}_w)_{ij}}, \tag{5}$$

namely,

$$\mathcal{J}(\mathbf{W}_I) = \frac{tr(\mathbf{W}_I^T \mathbf{X} \mathbf{L_b} \mathbf{X}^T \mathbf{W}_I)}{tr(\mathbf{W}_I^T \mathbf{X} \mathbf{L_w} \mathbf{X}^T \mathbf{W}_I)}, \tag{6}$$

where $\mathbf{L}_w$ and $\mathbf{L}_b$ are the Laplacian matrices (Chung 1997). They are defined as $\mathbf{L}_w = \mathbf{D}_w - \mathbf{A}_w$, where $\mathbf{D}_w$ is a diagonal matrix with $(\mathbf{D}_w)_{ii} = \sum_j (\mathbf{A}_w)_{ij}$, and $\mathbf{L}_w = \mathbf{D}_b - \mathbf{A}_b$, where $\mathbf{D}_b$ is a diagonal matrix with $(\mathbf{D}_b)_{ii} = \sum_j (\mathbf{A}_b)_{ij}$.

For the sake of simplicity, we denote hereafter the matrices $\mathbf{B} = \mathbf{X} \mathbf{L}_b \mathbf{X}^T \in \mathbb{R}^{d \times d}$ and $\mathbf{E} = \mathbf{X} \mathbf{L}_w \mathbf{X}^T \in \mathbb{R}^{d \times d}$. Then the criterion in (6) is rewritten as

$$\mathcal{J}(\mathbf{W}_I) = \frac{tr(\mathbf{W}_I^T \mathbf{B} \mathbf{W}_I)}{tr(\mathbf{W}_I^T \mathbf{E} \mathbf{W}_I)}. \tag{7}$$

Obviously, both $\mathbf{B}$ and $\mathbf{E}$ are positive semidefinite.

Base on the criterion (5), the score of a feature subset $\Phi(I)$ is calculated as

$$score(\Phi(I)) = \frac{tr(\mathbf{W}_I^T \mathbf{B} \mathbf{W}_I)}{tr(\mathbf{W}_I^T \mathbf{E} \mathbf{W}_I)}. \tag{8}$$

The task of feature selection is to seek the feature subset with the maximum score by solving the following optimization problem:

$$\Phi(I) = arg \max_{\Phi(I)} \frac{tr(\mathbf{W}_I^T \mathbf{B} \mathbf{W}_I)}{tr(\mathbf{W}_I^T \mathbf{E} \mathbf{W}_I)}. \tag{9}$$

It is important to note that the criterion (5) provides a general graph framework for feature selection. Different ways of constructing the weight matrices $\mathbf{A}_w$ and $\mathbf{A}_b$ will lead to different unsupervised, semi-supervised or supervised feature selection algorithm. Fisher score (Bishop 1995) and Laplacian score (He, Cai, & Niyogi 2005) are two representative instances.

In Fisher score, the weight matrices $\mathbf{A}_w$ and $\mathbf{A}_b$ are defined by

$$(\mathbf{A}_w)_{ij} = \begin{cases} \frac{1}{n_{c(i)}}, & \text{if } c(i) = c(j); \\ 0, & \text{if } c(i) \neq c(j). \end{cases} \tag{10}$$

$$(\mathbf{A}_b)_{ij} = \begin{cases} \frac{1}{n} - \frac{1}{n_{c(i)}}, & \text{if } c(i) = c(j); \\ \frac{1}{n}, & \text{if } c(i) \neq c(j). \end{cases} \tag{11}$$

where $c(i)$ denotes the class label of data point $\boldsymbol{x}_i$, and n_i denotes the number of data in class i.

In Laplacian score, the weight matrices are defined by [1]

$$(\mathbf{A}_w)_{ij} = \begin{cases} e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}}, & \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are neighbors}; \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

[1] In order to be consistent with Fisher score, the Laplacian score here is the reciprocal of the one in (He, Cai, & Niyogi 2005).

$$\mathbf{A}_b = \frac{1}{\mathbf{1}^T\mathbf{D}_w\mathbf{1}}\mathbf{D}_w\mathbf{1}\mathbf{1}^T\mathbf{D}_w. \quad (13)$$

Fisher score is a supervised method and makes use of the label information for constructing the weight matrices $\mathbf{A}_w$ and $\mathbf{A}_b$, while Laplacian score is an unsupervised method and no label information is applied for constructing the two weight matrices.

Traditional Solution: Feature-Level Score

The number of possible subsets of features increases greatly with respect to the number of features d, and hence the computational cost is very high to search in a brute force manner for the optimal subset of features based on the score defined in (8). Instead of directly calculating the score of a subset of features, traditional methods calculate the score of each feature, and then select the leading features based on the rank of the scores (Bishop 1995; He, Cai, & Niyogi 2005; Zhao & Liu 2007b).

Under the criterion (5), the score of the i-th feature is

$$score_1(F_i) = \frac{\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{w}_i}{\boldsymbol{w}_i^T\mathbf{E}\boldsymbol{w}_i}. \quad (14)$$

The traditional algorithm for feature selection is summarized in Table 1. It is obvious that the selected subset of features based from the algorithm in Table 1 cannot guarantee the global optimum of the subset-level score in (8).

Table 1: *Algorithm for feature selection based on the feature-level score*.

Input:

The selected feature number m, the matrices $\mathbf{B} \in \mathbb{R}^{d\times d}$ and $\mathbf{E} \in \mathbb{R}^{d\times d}$.

Output:

The selected feature subset $\Phi(I^*) = \{F_{I^*(1)}, F_{I^*(2)}, ..., F_{I^*(m)}\}$.

Algorithm:

1. Calculate the score of each feature F_i defined in Equation (14).
2. Rank the features according to the scores in descending order.
3. Select the leading m features to form $\Phi(I^*)$.

Globally Optimal Solution: Subset-Level Score

In this section, we propose a novel iterative algorithm to efficiently find the optimal subset of features of which the subset-level score is maximized.

Suppose the subset-level score in (8) reaches the global maximum λ^* if $\mathbf{W}_I = \mathbf{W}_{I^*}$, that is to say,

$$\frac{tr(\mathbf{W}_{I^*}^T\mathbf{B}\mathbf{W}_{I^*})}{tr(\mathbf{W}_{I^*}^T\mathbf{E}\mathbf{W}_{I^*})} = \lambda^*, \quad (15)$$

and

$$\frac{tr(\mathbf{W}_I^T\mathbf{B}\mathbf{W}_I)}{tr(\mathbf{W}_I^T\mathbf{E}\mathbf{W}_I)} \leq \lambda^*, \ \forall\ \Phi(I). \quad (16)$$

From Equation (16), we can derive that

$$\begin{aligned} & \frac{tr(\mathbf{W}_I^T\mathbf{B}\mathbf{W}_I)}{tr(\mathbf{W}_I^T\mathbf{E}\mathbf{W}_I)} \leq \lambda^*, \ \forall\ \Phi(I) \\ \Rightarrow \quad & tr(\mathbf{W}_I^T(\mathbf{B} - \lambda^*\mathbf{E})\mathbf{W}_I) \leq 0, \ \forall\ \Phi(I) \\ \Rightarrow \quad & \max_{\Phi(I)} tr(\mathbf{W}_I^T(\mathbf{B} - \lambda^*\mathbf{E})\mathbf{W}_I) \leq 0. \end{aligned} \quad (17)$$

Note that $tr(\mathbf{W}_{I^*}^T(\mathbf{B} - \lambda^*\mathbf{E})\mathbf{W}_{I^*}) = 0$, and from Equation (17), we have

$$\max_{\Phi(I)}\ tr(\mathbf{W}_I^T(\mathbf{B} - \lambda^*\mathbf{E})\mathbf{W}_I) = 0. \quad (18)$$

Let the function

$$f(\lambda) = \max_{\Phi(I)}\ tr(\mathbf{W}_I^T(\mathbf{B} - \lambda\mathbf{E})\mathbf{W}_I), \quad (19)$$

then we have $f(\lambda^*) = 0$.

Note that $\mathbf{B}$ and $\mathbf{E}$ are positive semidefinite, We will see from Equation (24) that $f(\lambda)$ is a monotonically decreasing function. Therefore, finding the global optimal λ^* can be converted to finding the root of equation $f(\lambda) = 0$.

Here, we define another score of the i-th feature as

$$score_2(F_i) = \boldsymbol{w}_i^T(\mathbf{B} - \lambda\mathbf{E})\boldsymbol{w}_i. \quad (20)$$

Note that $f(\lambda)$ can be rewritten as

$$f(\lambda) = \max_{\Phi(I)}\ \sum_{i=1}^{m} \boldsymbol{w}_{I(i)}^T(\mathbf{B} - \lambda\mathbf{E})\boldsymbol{w}_{I(i)}. \quad (21)$$

Thus $f(\lambda)$ equals to the sum of the first m largest scores.

Suppose for a $\phi(I_n)$, λ_n is calculated by

$$\lambda_n = \frac{tr(\mathbf{W}_{I_n}^T\mathbf{B}\mathbf{W}_{I_n})}{tr(\mathbf{W}_{I_n}^T\mathbf{E}\mathbf{W}_{I_n})}. \quad (22)$$

Denote $f(\lambda_n)$ by

$$f(\lambda_n) = tr(\mathbf{W}_{I_{n+1}}^T(\mathbf{B} - \lambda_n\mathbf{E})\mathbf{W}_{I_{n+1}}), \quad (23)$$

where $\mathbf{W}_{I_{n+1}}$ can be efficiently calculated according to the rank of scores defined in Equation (20).

Note that in Equation (19), $\mathbf{W}_I$ is not fixed w.r.t λ, so $f(\lambda)$ is piecewise linear. The slope of $f(\lambda)$ at point λ_n is

$$f'(\lambda_n) = -tr(\mathbf{W}_{I_{n+1}}^T\mathbf{E}\mathbf{W}_{I_{n+1}}) \leq 0. \quad (24)$$

We use a linear function $g(\lambda)$ to approximate the piecewise linear function $f(\lambda)$ at point λ_n such that

$$\begin{aligned} g(\lambda) &= f'(\lambda_n)(\lambda - \lambda_n) + f(\lambda_n) \\ &= tr(\mathbf{W}_{I_{n+1}}^T(\mathbf{B} - \lambda\mathbf{E})\mathbf{W}_{I_{n+1}}). \end{aligned} \quad (25)$$

Let $g(\lambda_{n+1}) = 0$, we have

$$\lambda_{n+1} = \frac{tr(\mathbf{W}_{I_{n+1}}^T\mathbf{B}\mathbf{W}_{I_{n+1}})}{tr(\mathbf{W}_{I_{n+1}}^T\mathbf{E}\mathbf{W}_{I_{n+1}})}. \quad (26)$$

Since $g(\lambda)$ approximates $f(\lambda)$, λ_{n+1} in (26) is an approximation to the root of equation $f(\lambda) = 0$. Update λ_n by λ_{n+1}, we can obtain an iterative procedure to find the root of equation $f(\lambda) = 0$ and thus the optimal solution in (9).

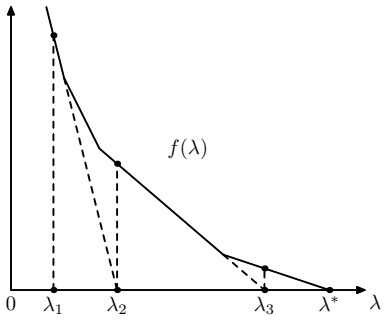


Figure 1: *Since the function $f(\lambda)$ is piecewise linear, the algorithm can iteratively find the root of equation $f(\lambda) = 0$ in a few steps. Suppose λ_1 is an initial value in the algorithm, then the updated value is λ_2 in the first step and λ_3 in the second step. Finally, the optimal value λ^* is achieved in the third step.*

Theorem 1 *The λ in the iterative procedure increases monotonically.*

Proof.

$$\lambda_n = \frac{tr(\mathbf{W}_{I_n}^T \mathbf{B} \mathbf{W}_{I_n})}{tr(\mathbf{W}_{I_n}^T \mathbf{E} \mathbf{W}_{I_n})} \leq \max_{\Phi(I)} \frac{tr(\mathbf{W}_I^T \mathbf{B} \mathbf{W}_I)}{tr(\mathbf{W}_I^T \mathbf{E} \mathbf{W}_I)} = \lambda^*. \tag{27}$$

Since $f(\lambda)$ is monotonically decreasing, we know $f(\lambda_n) \geq 0$. According to Equation (23), we have

$$\frac{tr(\mathbf{W}_{I_{n+1}}^T \mathbf{B} \mathbf{W}_{I_{n+1}})}{tr(\mathbf{W}_{I_{n+1}}^T \mathbf{E} \mathbf{W}_{I_{n+1}})} \geq \lambda_n. \tag{28}$$

That is, $\lambda_{n+1} \geq \lambda_n$. Therefore, the λ in the iterative procedure increases monotonically. □

Note that $f(\lambda)$ is piecewise linear, only a few steps are needed to achieve the optimum. We illustrate the iterative procedure in Figure 1 and summarize the algorithm in Table 2. Suppose r is the number of zero diagonal elements of $\mathbf{E}$, the algorithm in Table 2 can be performed if $m > r$, while in Table 1, r should be 0. One interesting property of the objective function for feature selection is stated as below:

Theorem 2 *The optimal subset-level score in (8) is monotonically decreased with respect to the selected feature number m. That is to say, if $m_1 < m_2$, then*

$$\max_{\Phi(I)} \frac{\sum_{i=1}^{m_1} \boldsymbol{w}_{I(i)}^T \mathbf{B} \boldsymbol{w}_{I(i)}}{\sum_{i=1}^{m_1} \boldsymbol{w}_{I(i)}^T \mathbf{E} \boldsymbol{w}_{I(i)}} \geq \max_{\Phi(I)} \frac{\sum_{i=1}^{m_2} \boldsymbol{w}_{I(i)}^T \mathbf{B} \boldsymbol{w}_{I(i)}}{\sum_{i=1}^{m_2} \boldsymbol{w}_{I(i)}^T \mathbf{E} \boldsymbol{w}_{I(i)}} \tag{29}$$

The proof is provided in appendix. From Theorem 2 we know, when the selected feature number m increases, the optimal subset-level score in (8) will be decreased. We will verify this property in the experiments.

Experiments

In this section, we empirically compare the performance of the subset-level score with the feature-level score, when the trace ratio criterion is used for feature selection.

Table 2: *Algorithm for feature selection based on the subset-level score*.

Input:

The selected feature number m, the matrices $\mathbf{B} \in \mathbb{R}^{d \times d}$ and $\mathbf{E} \in \mathbb{R}^{d \times d}$.

Output:

The selected feature subset $\Phi(I^*) = \{F_{I^*(1)}, F_{I^*(2)}, ..., F_{I^*(m)}\}$.

Algorithm:

1. Initialize $\Phi(I)$, and let $\lambda = \frac{tr(\mathbf{W}_I^T \mathbf{B} \mathbf{W}_I)}{tr(\mathbf{W}_I^T \mathbf{E} \mathbf{W}_I)}$.
2. Calculate the score of each feature F_i defined in Equation (20).
3. Rank the features according to the scores in descending order.
4. Select the leading m features to update $\Phi(I)$, and let $\lambda = \frac{tr(\mathbf{W}_I^T \mathbf{B} \mathbf{W}_I)}{tr(\mathbf{W}_I^T \mathbf{E} \mathbf{W}_I)}$.
5. Iteratively perform step 2-4 until convergence.

Two typical trace ratio based feature selection algorithms are performed in the experiments: Fisher score and Laplacian score. In the Fisher score, we denote the traditional method (feature-level score) by F-FS, and our method (subset-level score) by S-FS. In the Laplacian score, we denote the traditional method (feature-level score) by F-LS, and our method (subset-level score) by S-LS.

Two sets of datasets are used in the experiments, the first one are taken from the UCI Machine Learning Repository (Asuncion & Newman 2007), and the second one are taken from the real-world face image databases, including AT&T (Samaria & Harter 1994) and UMIST (Graham & Allinson 1998). A brief description of these datasets is summarized in Table 3.

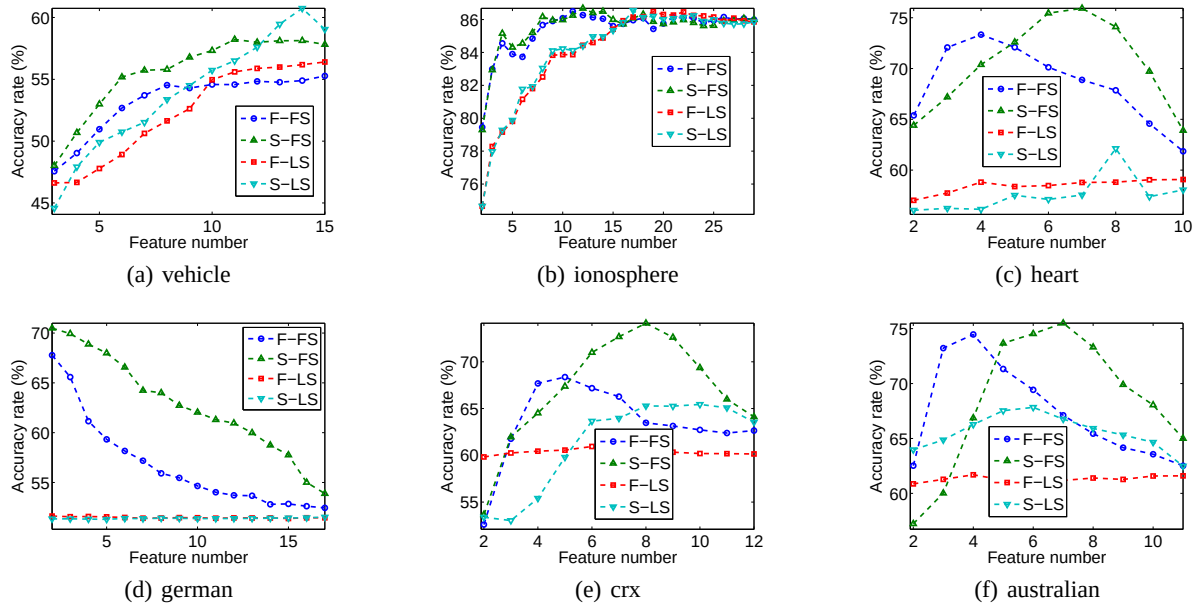
The performances of the algorithms are measured by the classification accuracy rate with selected features on testing data. The classification is based on the conventional 1-nearest neighbor classifier with Euclidean distance metric. In each experiment, we randomly select several samples per class for training and the remaining samples for testing. The average accuracy rates versus selected feature number are recorded over 20 random splits.

In most cases, our method converges in only three to five steps. As more than 95% computation time is spent on the calculation of the diagonal elements of the matrices $\mathbf{B}$ and $\mathbf{E}$, our method nearly does not increase the computation complexity in comparison with the traditional method.

Results on UCI Datasets

Six datasets from the UCI machine learning repository are used in this experiment. In each dataset, the training number per class is 30.

The results of accuracy rate versus selected feature num-



(a) vehicle (b) ionosphere (c) heart

(d) german (e) crx (f) australian

Figure 2: Accuracy rate vs. dimension.

Table 3: *A brief description of the datasets in the experiments, including the class number, total data number, training sample number and data dimension.*

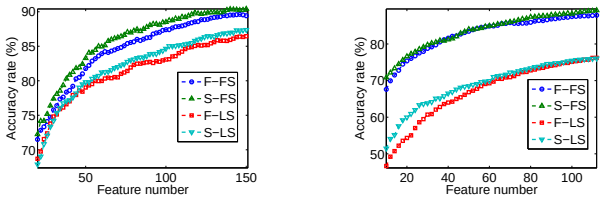
	class	total num.	train. num.	dimension
vehicle	4	846	120	18
ionosphere	2	351	60	34
heart	2	270	60	13
german	2	1000	60	20
crx	2	690	60	15
australian	2	690	60	14
AT&T	40	400	200	644
UMIST	20	575	100	644

ber are shown in Figure 2. In most cases, our method (S-FS or S-LS) obtains a better result than the corresponding traditional method (F-FS or F-LS). We also notice that in a few cases, our method does not outperform the corresponding traditional method. The reason is that, although a larger subset-level score is expected to perform better, this score is not directly related to the accuracy rate, which is the usual case in filter-type methods for feature selection. Therefore, although our method theoretically guarantees to find the feature subset with the optimal subset-level score, it is not always guaranteed to obtain the optimal accuracy rate. But generally the consistency between the subset score and the accuracy rate can be expected if the objective function is well defined.

Results on Face Datasets

In this experiment, we used two face datasets, including AT&T dataset and UMIST dataset. In each dataset, the training sample number per class is 5.

The AT&T face database includes 40 distinct individuals



(a) AT&T (b) UMIST

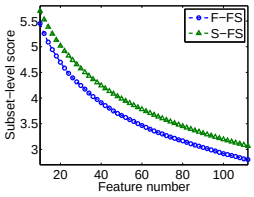
Figure 3: Accuracy rate vs. dimension.

and each individual has 10 different images. The UMIST repository is a multiview database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views. Images are down-sampled to the size of 28×23.

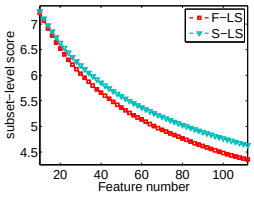
The results of accuracy rate versus selected feature number are shown in Figure 3. From the figure we can see, our method (S-FS or S-LS) obtains a better result than the corresponding traditional method (F-FS or F-LS) in most cases.

Comparison on Subset-level Scores

We have proved in the previous section that our method can find the feature subset such that the subset-level score calculated by Equation (8) is maximized. In contrast, traditional methods, which are based on the feature-level score calculated by Equation (14), cannot guarantee that the subset-level score of the selected feature subset reaches the global maximum. Figure 4 shows the subset-level scores of the selected feature subset by traditional methods and our method in the UMIST dataset. We can observe that the subset-level scores of the feature subset found by traditional methods



(a) Fisher Scores



(b) Laplacian Scores

Figure 4: Comparison between the subset-level scores of the feature subset selected by traditional method(F-FS, F-LS) and our method(S-FS, S-LS) .

are consistently lower than those of our method. We can also observe that the optimal subset-level score found by our method monotonically decreases with respect to the selected feature number, which is consistent with Theorem 2.

Conclusion

In this paper, we proposed a novel algorithm to solve the general graph-based feature selection problem. Unlike traditional methods which treat each feature individually and hence are suboptimal, our proposed algorithm directly optimizes the score of the entire selected feature subset. The theoretical analysis guarantees the algorithmic convergency and global optimum of the solution. Our proposed algorithm is general, and can be used to extend any graph-based subspace learning algorithm to its feature selection version. In addition, we are planning to further study the technique applied in this paper for solving the kernel selection problem encountered by traditional kernel based subspace learning.

Appendix

In order to prove Theorem 2, we first prove the following two lemmas.

Lemma 1 **If** $\forall i, a_i \geq 0, b_i > 0$ *and* $\frac{a_1}{b_1} \geq \frac{a_2}{b_2} \geq \cdots \geq \frac{a_k}{b_k}$, *then* $\frac{a_1}{b_1} \geq \frac{a_1+a_2+\cdots+a_k}{b_1+b_2+\cdots+b_k} \geq \frac{a_k}{b_k}$.

Proof. Let $\frac{a_1}{b_1} = p$. So $\forall\ i, a_i \geq 0, b_i > 0$, we have $a_i \leq pb_i$. Therefore $\frac{a_1+a_2+\cdots+a_k}{b_1+b_2+\cdots+b_k} \leq \frac{p(b_1+b_2+\cdots+b_k)}{b_1+b_2+\cdots+b_k} = \frac{a_1}{b_1}$.

Let $\frac{a_k}{b_k} = q$. So $\forall\ i, a_i \geq 0, b_i > 0$, we have $a_i \geq qb_i$. Therefore $\frac{a_1+a_2+\cdots+a_k}{b_1+b_2+\cdots+b_k} \geq \frac{q(b_1+b_2+\cdots+b_k)}{b_1+b_2+\cdots+b_k} = \frac{a_k}{b_k}$. □

Lemma 2 *If* $\forall\ i, a_i \geq 0, b_i > 0$, $m_1 < m_2$ *and* $\frac{a_1}{b_1} \geq \frac{a_2}{b_2} \geq \cdots \geq \frac{a_{m_1}}{b_{m_1}} \geq \frac{a_{m_1+1}}{b_{m_1+1}} \geq \cdots \geq \frac{a_{m_2}}{b_{m_2}}$, *then we have* $\frac{a_1+a_2+\cdots+a_{m_1}}{b_1+b_2+\cdots+b_{m_1}} \geq \frac{a_1+a_2+\cdots+a_{m_2}}{b_1+b_2+\cdots+b_{m_2}}$.

Proof. According to Lemma 1, we know $\frac{a_1+a_2+\cdots+a_{m_1}}{b_1+b_2+\cdots+b_{m_1}} \geq \frac{a_{m_1}}{b_{m_1}} \geq \frac{a_{m_1+1}}{b_{m_1+1}} \geq \frac{a_{m_1+1}+a_{m_1+2}+\cdots+a_{m_2}}{b_{m_1+1}+b_{m_1+2}+\cdots+b_{m_2}}$. Thus we have $\frac{a_1+a_2+\cdots+a_{m_1}}{b_1+b_2+\cdots+b_{m_1}} \geq \frac{a_{m_1+1}+a_{m_1+2}+\cdots+a_{m_2}}{b_{m_1+1}+b_{m_1+2}+\cdots+b_{m_2}}$. According to Lemma 1 again, we have $\frac{a_1+a_2+\cdots+a_{m_1}}{b_1+b_2+\cdots+b_{m_1}} \geq \frac{a_1+a_2+\cdots+a_{m_2}}{b_1+b_2+\cdots+b_{m_2}}$.

Proof of Theorem 2. Without loss of generality, suppose $\frac{\boldsymbol{w}_1^T\mathbf{B}\boldsymbol{w}_1}{\boldsymbol{w}_1^T\mathbf{E}\boldsymbol{w}_1} \geq \frac{\boldsymbol{w}_2^T\mathbf{B}\boldsymbol{w}_2}{\boldsymbol{w}_2^T\mathbf{E}\boldsymbol{w}_2} \geq \cdots \geq \frac{\boldsymbol{w}_{m_2}^T\mathbf{B}\boldsymbol{w}_{m_2}}{\boldsymbol{w}_{m_2}^T\mathbf{E}\boldsymbol{w}_{m_2}}$ and $\frac{\sum_{i=1}^{m_2}\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{w}_i}{\sum_{i=1}^{m_2}\boldsymbol{w}_i^T\mathbf{E}\boldsymbol{w}_i} = \max_{\Phi(I)} \frac{\sum_{i=1}^{m_2}\boldsymbol{w}_{I(i)}^T\mathbf{B}\boldsymbol{w}_{I(i)}}{\sum_{i=1}^{m_2}\boldsymbol{w}_{I(i)}^T\mathbf{E}\boldsymbol{w}_{I(i)}}$. Note that $m_1 < m_2$, therefore, according to Lemma 2, we have

$$\max_{\Phi(I)} \frac{\sum_{i=1}^{m_1}\boldsymbol{w}_{I(i)}^T\mathbf{B}\boldsymbol{w}_{I(i)}}{\sum_{i=1}^{m_1}\boldsymbol{w}_{I(i)}^T\mathbf{E}\boldsymbol{w}_{I(i)}} \geq \frac{\sum_{i=1}^{m_1}\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{w}_i}{\sum_{i=1}^{m_1}\boldsymbol{w}_i^T\mathbf{E}\boldsymbol{w}_i} \geq \frac{\sum_{i=1}^{m_2}\boldsymbol{w}_i^T\mathbf{B}\boldsymbol{w}_i}{\sum_{i=1}^{m_2}\boldsymbol{w}_i^T\mathbf{E}\boldsymbol{w}_i} = \max_{\Phi(I)} \frac{\sum_{i=1}^{m_2}\boldsymbol{w}_{I(i)}^T\mathbf{B}\boldsymbol{w}_{I(i)}}{\sum_{i=1}^{m_2}\boldsymbol{w}_{I(i)}^T\mathbf{E}\boldsymbol{w}_{I(i)}}.$$

□

Acknowledgments

The work was supported by NSFC (Grant No. 60721003, 60675009), P. R. China, and in part supported by AcRF Tier-1 Grant of R-263-000-464-112, Singapore.

References

Asuncion, A., and Newman, D. 2007. *UCI Machine Learning Repository*.

Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15(6):1373–1396.

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press.

Chung, F. R. K. 1997. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, No. 92, American Mathematical Society.

Das, S. 2001. Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML*, 74–81.

Dy, J. G., and Brodley, C. E. 2004. Feature selection for unsupervised learning. *JMLR* 5:845–889.

Graham, D. B., and Allinson, N. M. 1998. Characterizing virtual eigensignatures for general purpose face recognition. in face recognition: From theory to applications. *NATO ASI Series F, Computer and Systems Sciences* 163:446–456.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *JMLR* 3:1157–1182.

He, X. F.; Yan, S. C.; Hu, Y. X.; Niyogi, P.; and Zhang, H. J. 2005. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(3):328–340.

He, X.; Cai, D.; and Niyogi, P. 2005. Laplacian score for feature selection. In *NIPS*.

Kohavi, R., and John, G. H. 1997. Wrappers for feature subset selection. *Artif. Intell.* 97(1-2):273–324.

Nie, F.; Xiang, S.; and Zhang, C. 2007. Neighborhood minmax projections. In *IJCAI*, 993–998.

Robnik-Sikonja, M., and Kononenko, I. 2003. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning* 53:23–69.

Samaria, F. S., and Harter, A. C. 1994. Parameterisation of a stochastic model for human face identification. In *2nd IEEE Workshop on Applications of Computer Vision*, 138–142.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22(8):888–905.

Wang, H.; Yan, S.; Xu, D.; Tang, X.; and Huang, T. S. 2007. Trace ratio vs. ratio trace for dimensionality reduction. In *CVPR*.

Zhao, Z., and Liu, H. 2007a. Semi-supervised feature selection via spectral analysis. In *SDM*.

Zhao, Z., and Liu, H. 2007b. Spectral feature selection for supervised and unsupervised learning. In *ICML*, 1151–1157.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 912–919.