On Similarity Preserving Feature Selection

Zheng Zhao, Lei Wang, Huan Liu, and Jieping Ye, *Members, IEEE*

Abstract—In the literature of feature selection, different criteria have been developed to evaluate the goodness of features. In our investigation, we notice that a number of existing selection criteria implicitly select features that preserve sample similarity, and can be unified under a common framework. We further point out that any feature selection criteria covered by this framework cannot handle redundant features, a common drawback of these criteria. Motivated by these observations, we propose a new "Similarity Preserving Feature Selection" framework in an explicit and rigorous way. We show, through theoretical analysis, that the proposed framework not only encompasses many widely used feature selection criteria, but also naturally overcomes their common weakness in handling feature redundancy. In developing this new framework, we begin with a conventional combinatorial optimization formulation for similarity preserving feature selection, then extend it with a sparse multiple-output regression formulation to improve its efficiency and effectiveness. A set of three algorithms are devised to efficiently solve the proposed formulations, each of which has its own advantages in terms of computational complexity and selection performance. As exhibited by our extensive experimental study, the proposed framework achieves superior feature selection performance and attractive properties.

Index Terms—Feature Selection, Similarity Preserving, Redundancy Removal, Multiple Output Regression, Sparse Regularization

1 Introduction

Feature selection aims to choose a subset of the original features according to a selection criterion. It is an important technique widely used in pattern analysis. It reduces data dimensionality by removing irrelevant and redundant features, and brings about the immediate effects for applications, such as speeding up a data mining algorithm, improving predictive accuracy, and enhancing result comprehensibility [16], [23]. According to the way of utilizing label information, feature selection algorithms can be categorized as supervised algorithms [35], [41], unsupervised algorithms [11], [19] or semi-supervised algorithms [43], [49]. From the perspective of selection strategy, feature selection algorithms broadly fall into three models: filter, wrapper or embedded [16]. The filter model evaluates features without involving any learning algorithm. The wrapper model requires a learning algorithm and uses its performance to evaluate the goodness of features. Algorithms of the embedded model, e.g., C4.5 [28] and LARS [12], incorporate feature selection as a part of the learning process, and use the objective function of the learning model to guide searching for relevant features. In addition, feature selection algorithms may return either a subset of features [17], [45] or the weights of all features measuring their utility [1], [34]. Hence, they can also be categorized as subset selection algorithms or feature weighting algorithms. As an important technique for dimensionality reduction, feature selection has been applied to various areas, including computer vision [10], text mining [13], and bioinformatics [31], to name a few.

The selection criterion is a pivotal component of feature selection. It can take various forms: separability, information, dependency, consistency, learning model performance (used in the wrapper model), and so on. In our recent study, we observe that a number of existing feature selection algorithms are essentially based on assessing features' capability in preserving sample similarity, which can be inferred from either label information or predefined distance metric [48]. These feature selection algorithms include Relief and ReliefF [34], Laplacian Score [19], Fisher Score [9], SPEC [48], HSIC [35], and Trace Ratio [27]. They have demonstrated excellent performance in both supervised and unsupervised learning contexts and have been applied successfully to various real applications. These algorithms are designed to achieve different goals. For example, Fisher Score and ReliefF are designed to optimize sample separability, Laplacian Score is designed to retain sample locality, and HSIC is designed to maximize feature-class dependency. However, it can be shown that all of these algorithms select features by evaluating features' capability in preserving sample similarity in different ways. We defined a unified framework which includes the above feature selection algorithms as its special cases. And our theoretical analysis and experimental study show that in these algorithms, a crucial component is missed, which leads to their common drawback of being unable to handle feature *redundancy*. For instance, they may repeatedly select highly correlated features in the selection process. It has been known that redundant features can adversely affect the performance of classification and clustering, therefore should be removed by feature selection for improving learning performance [7], [45].

Based on the above observations, we explicitly propose the concept of "Similarity Preserving Feature Selection", and develop corresponding feature selection

• *Z. Zhao is with SAS Institute Inc. Cary, NC 27513, USA. E-mail: zheng.zhao@sas.com; L. Wang is with the School of Computer Science and Software Engineering, University of Wollongong, NSW, Australia. E-mail: leiw@uow.edu.au; H. Liu and J. Ye are with Arizona State University, Tempe, AZ 85281, USA. E-mail: {huan.liu,jieping.ye}@asu.edu*

framework in a direct and rigorous way. We provide a thorough analysis for the proposed feature selection framework in dealing with redundant features, and explain how the weakness of the existing algorithms can be overcome. Beginning with a conventional combinatorial optimization formulation, we show how the feature selection in our framework can be reformulated as a multiple-output regression problem [18] with an $L_{2,1}$-norm constraint. Based on these formulations, we devise three different algorithms to identify the optimal feature set. The first one takes the traditional sequential forward selection approach to solve the combinatorial optimization formulation. By using the Nesterov's method in constrained optimization [24], [25], [26], the second one globally solves the constrained optimization problem and selects features accordingly. However, this method requires cross-validation for parameter tuning to obtain good performance, which is time-consuming. Inspired by the group LAR [46], we develop the third algorithm that generates an approximate solution path for the constrained optimization problem. It does not require parameter tuning, and is more efficient. Each of the three algorithms has its own advantages in terms of time complexity and selection performance.

The contributions of this paper include: (1) We propose a general framework that unifies a number of existing feature selection algorithms and allows their joint study. (2) We explicitly propose the concept of "Similarity Preserving Feature Selection", and develop a new selection framework in a direct and rigorous way. (3) We formulate our framework as a regularized multiple-output regression problem and propose different ways to solve it efficiently. (4) We theoretically show that the proposed framework is able to avoid selecting redundant features, overcoming a common drawback of the existing algorithms. And (5) we conduct extensive experiments to evaluate the proposed framework with both supervised and unsupervised learning. The results demonstrate its generality and superior performance.

1.1 Notation

In the paper, we use $\mathbf{X} \in \mathbb{R}^{n\times m}$ to denote the data matrix, where n is the number of instances and m is the number of features. For each data set, we use $f_1, \ldots, f_m$ to denote the m features, and $\mathbf{f}_1, \ldots, \mathbf{f}_m$ are the corresponding feature vectors, where $\mathbf{f}_i \in \mathbb{R}^n$ and $\mathbf{X} = (\mathbf{f}_1, \ldots, \mathbf{f}_m)$. We also use $\mathbf{x}_1, \ldots, \mathbf{x}_n$ to denote the n instances, $\mathbf{x}_i \in \mathbb{R}^m$ and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$. Given n instances, the pairwise similarity among them can be presented as a symmetric matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$. Given $\mathbf{K}$, we use $\mathbb{G}(V, E)$ to denote the undirected graph constructed from $\mathbf{K}$, where V is the vertex set, and E is the edge set. The i-th vertex of $\mathbb{G}$ corresponds to $\mathbf{x}_i$, and there is an edge between each vertex pair $(\mathbf{x}_i, \mathbf{x}_j)$, where the weight k_{ij} is the (i, j) the entry of $\mathbf{K}$. In this case $\mathbf{K}$ is called the affinity matrix of $\mathbb{G}$. Given $\mathbb{G}$ and its affinity matrix $\mathbf{K}$, let $\mathbf{d}$ denote the vector: $\mathbf{d} = (d_1, d_2, \ldots, d_n)$, where $d_i = \sum_{j=1}^n k_{ij}$. The degree matrix $\mathbf{D}$ of the graph $\mathbb{G}$ is defined by: $\mathbf{D}_{ij} = d_i$ if $i = j$, and 0 otherwise. Here d_i can be interpreted as an estimation of the density around $\mathbf{x}_i$, since the more data points that are close to $\mathbf{x}_i$, the larger the value of d_i. Given $\mathbf{K}$ and $\mathbf{D}$, the Laplacian matrix $\mathbf{L}$ and the normalized Laplacian matrix $\mathcal{L}$ are defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{K}; \quad \mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}.$$

In multiple-output regression analysis [18], assuming that $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_k) \in \mathbb{R}^{n\times k}$ denotes the response matrix, and $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_k) \in \mathbb{R}^{m\times k}$ denotes the weight matrix, the goal is to learn a weight matrix $\mathbf{W}$ to minimize the following objective function:

$$\|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 = \text{Trace}\left((\mathbf{Y} - \mathbf{X}\mathbf{W})(\mathbf{Y} - \mathbf{X}\mathbf{W})^\top\right),$$

where $\|\cdot\|_F$ denotes the Frobenius norm [14]. In the paper, we use boldface characters in uppercase, such as $\mathbf{X}$ and $\mathbf{K}$, to denote matrices, use boldface characters in lowercase, such as $\mathbf{x}$ and $\mathbf{d}$, to denote vectors, and use normal characters in lowercase to denote scalars. We use $\mathbf{1}$ to denote the vector with all its elements being 1, and $\mathbf{I}$ to denote the identity matrix with all its diagonal elements being 1 and all other elements being 0.

2 Existing Feature Selection Criteria Based on Similarity Preserving

In this part, we review the existing feature selection criteria that select features via evaluating their capability in preserving sample similarity. These algorithms include Laplacian Score [19], SPEC [48], Fisher Score [9], Trace Ratio criterion [27], ReliefF [34], and HSIC [35].

2.1 Feature Selection with Laplacian Score

Laplacian Score is proposed in [19] to select features that retain sample locality specified by an affinity matrix $\mathbf{K}$. Given $\mathbf{K}$, its corresponding degree matrix $\mathbf{D}$ and Laplacian matrix $\mathbf{L}$, the Laplacian Score of a feature $\mathbf{f}$ is calculated in the following way:

$$\text{SC}_L(\mathbf{f}) = \frac{\widetilde{\mathbf{f}}^\top \mathbf{L}\widetilde{\mathbf{f}}}{\widetilde{\mathbf{f}}^\top \mathbf{D}\widetilde{\mathbf{f}}}, \quad \text{where} \quad \widetilde{\mathbf{f}} = \mathbf{f} - \frac{\mathbf{f}^\top \mathbf{D}\mathbf{1}}{\mathbf{1}^\top \mathbf{D}\mathbf{1}}\mathbf{1}.$$

Since features are evaluated independently in Laplacian Score, selecting k features with Laplacian Score can be achieved by greedily picking the top k features which have the minimal SC_L values.

2.2 Feature Selection with SPEC

Proposed in [48], SPEC is an extension of Laplacian Score. In SPEC, given the affinity matrix $\mathbf{K}$, the degree matrix $\mathbf{D}$, and the normalized Laplacian matrix $\mathcal{L}$, three evaluation criteria are proposed for measuring feature relevance in the following ways:

$$\text{SC}_{S,1}(\mathbf{f}_i) = \hat{\mathbf{f}}_i^\top \gamma(\mathcal{L})\ \hat{\mathbf{f}}_i = \sum_{j=1}^n \alpha_j^2 \gamma(\lambda_j),$$

$$\mathrm{SC}_{S,2}(\mathbf{f}_i) = \frac{\hat{\mathbf{f}}_i^{\top}\,\gamma(\mathcal{L})\,\hat{\mathbf{f}}_i}{1-\left(\hat{\mathbf{f}}_i^{\top}\xi_1\right)^2} = \frac{\sum_{j=2}^{n}\alpha_j^2\gamma(\lambda_j)}{\sum_{j=2}^{n}\alpha_j^2},$$

$$\mathrm{SC}_{S,3}(\mathbf{f}_i) = \sum_{j=1}^{k}(\gamma(2)-\gamma(\lambda_j))\alpha_j^2.$$

In the above equations, $\hat{\mathbf{f}}_i = (\mathbf{D}^{\frac{1}{2}}\mathbf{f}_i)\cdot||(\mathbf{D}^{\frac{1}{2}}\mathbf{f}_i)||^{-1}$; (λ_j,ξ_j) is the j-th Eigen-pair of $\mathcal{L}$; $\alpha_j = \cos\theta_j$, where θ_j is the angle between $\hat{\mathbf{f}}_i$ and ξ_j; and $\gamma(\cdot)$ is an increasing function which is used to rescale the eigenvalues of $\mathcal{L}$ for denoising. The top eigenvectors of $\mathcal{L}$ are the optimal soft cluster indicators of the data [38]. By comparing with these eigenvectors, SPEC selects features that assign similar values to instances that are similar according to $\mathbf{K}$. In [48] it is shown that Laplacian Score is a special case of the second criterion, $\mathrm{SC}_{S,2}(\cdot)$, defined in SPEC. Note that SPEC also evaluates features independently.

2.3 Feature Selection with Fisher Score

Given class labels $\mathbf{y} = \{y_1,\ldots,y_n\}$, Fisher Score [9] selects features that assign similar values to the samples from the same class and different values to samples from different classes. The evaluation criterion used in Fisher Score can be formulated as

$$\mathrm{SC}_F(\mathbf{f}_i) = \frac{\sum_{j=1}^{c} n_j\left(\mu_{i,j}-\mu_i\right)^2}{\sum_{j=1}^{c} n_j\sigma_{i,j}^2},$$

where μ_i is the mean of the feature $\mathbf{f}_i$, n_j is the number of samples in the jth class, and $\mu_{i,j}$ and $\sigma_{i,j}^2$ are the mean and the variance of $\mathbf{f}_i$ on class j, respectively. In [19], it is shown that Fisher Score is a special case of Laplacian Score, when the similarity matrix is defined as

$$\mathbf{K}_{ij}^{FIS} = \begin{cases} \frac{1}{n_l}, & y_i = y_j = l \\ 0, & otherwise \end{cases}, \tag{2.1}$$

where n_l is the number of instances in the l-th class.

2.4 Feature Selection with Trace Ratio Criterion

The trace ratio criterion for subset-level feature selection is proposed in [27]. It defines two weight matrices $\mathbf{K}_w$ and $\mathbf{K}_b$. $\mathbf{K}_w$ represents the within-class or local affinity relationship of instances, whereas $\mathbf{K}_b$ represents the between-class or global counterpart. Let $\mathbf{L}_w = \mathbf{D}_w - \mathbf{K}_w$ and $\mathbf{L}_b = \mathbf{D}_b - \mathbf{K}_b$, where $\mathbf{D}_w$ and $\mathbf{D}_b$ are the degree matrices of $\mathbf{K}_w$ and $\mathbf{K}_b$, respectively. Let k be the number of features to be selected. An $m\times k$ selection matrix $\mathbf{W}$ is expressed as $[\mathbf{w}_{i_1},\mathbf{w}_{i_2},\cdots,\mathbf{w}_{i_k}]$, where the column vector $\mathbf{w}_{i_j}$ has one and only one "1" at its i_j-th element. The set $\{i_1,i_2,\cdots,i_k\}$ is a subset of $\{1,2,\cdots,m\}$. The trace ratio criterion seeks the best selection matrix $\mathbf{W}$ by maximizing the following criterion:

$$\mathbf{W}^{\star} = \arg\max_{\mathbf{W}} = \frac{\mathrm{Trace}(\mathbf{W}^{\top}\mathbf{X}^{\top}\mathbf{L}_b\mathbf{X}\mathbf{W})}{\mathrm{Trace}(\mathbf{W}^{\top}\mathbf{X}^{\top}\mathbf{L}_w\mathbf{X}\mathbf{W})}. \tag{2.2}$$

It is shown in [27] that this optimization problem can be solved by iteratively solving the following subproblems:

$$\begin{cases} \mathbf{W}_{i+1} &= \arg\max\limits_{\mathbf{W}} \mathrm{Trace}\left(\mathbf{W}^{\top}\mathbf{X}^{\top}(\mathbf{L}_b-\lambda_i\mathbf{L}_w)\mathbf{X}\mathbf{W}\right) \\ \lambda_{i+1} &= \frac{\mathrm{Trace}(\mathbf{W}_{i+1}^{\top}\mathbf{X}^{\top}\mathbf{L}_b\mathbf{X}\mathbf{W}_{i+1})}{\mathrm{Trace}(\mathbf{W}_{i+1}^{\top}\mathbf{X}^{\top}\mathbf{L}_w\mathbf{X}\mathbf{W}_{i+1})} \end{cases}. \tag{2.3}$$

Recognizing that the optimal λ is the root of the following piecewise linear function:

$$f(\lambda) = \max_{\mathbf{W}} \mathrm{Trace}\left(\mathbf{W}^{\top}\mathbf{X}^{\top}(\mathbf{L}_b-\lambda_i\mathbf{L}_w)\mathbf{X}\mathbf{W}\right),$$

we can efficiently solve the above optimization problem. Although the trace ratio criterion evaluates a set of features jointly, we will show that it does not take feature redundancy into account and is prone to selecting redundant or even duplicated features.

2.5 Feature Selection with ReliefF

Relief [21] and its multiclass extension ReliefF [22] are supervised feature weighting algorithms of the filter model. Assuming that p instances are randomly sampled from data, the evaluation criterion of Relief is defined as

$$\mathrm{SC}_R(\mathbf{f}_i) = \frac{1}{2}\sum_{t=1}^{p} d\left(f_{t,i}-f_{NM(\mathbf{x}_t),i}\right) - d\left(f_{t,i}-f_{NH(\mathbf{x}_t),i}\right),$$

where $f_{t,i}$ denotes the value of instance $\mathbf{x}_t$ on feature $\mathbf{f}_i$, $f_{NH(\mathbf{x}_t),i}$ and $f_{NM(\mathbf{x}_t),i}$ denote the values on the ith feature of the nearest points to $\mathbf{x}_t$ with the same and different class label respectively, and $d(\cdot)$ is a distance measurement. To handle multiclass problems, the above criterion is extended to the following formulation:

$$\mathrm{SC}_R(\mathbf{f}_i) = \frac{1}{p}\cdot\sum_{t=1}^{p}\left\{-\frac{1}{m_{\mathbf{x}_t}}\sum_{\mathbf{x}_j\in NH(\mathbf{x}_t)} d\left(f_{t,i}-f_{j,i}\right)\right.$$

$$\left. + \sum_{y\neq y_{\mathbf{x}_t}}\frac{1}{m_{\mathbf{x}_t,y}}\frac{P(y)}{1-P\left(y_{\mathbf{x}_t}\right)}\sum_{\mathbf{x}_j\in NM(\mathbf{x}_t,y)} d\left(f_{t,i}-f_{j,i}\right)\right\},$$

where $y_{\mathbf{x}_t}$ is the class label of the instance $\mathbf{x}_t$ and $P(y)$ is the probability of an instance being from the class y. $NH(\mathbf{x})$ or $NM(\mathbf{x},y)$ denotes a set of nearest points to $\mathbf{x}$ with the same class of $\mathbf{x}$, or a different class (the class y), respectively. $m_{\mathbf{x}_t}$ and $m_{\mathbf{x}_t,y}$ are the sizes of the sets $NH(\mathbf{x}_t)$ and $NM(\mathbf{x}_t,y)$, respectively. Usually, the size of both $NH(\mathbf{x})$ and $NM(\mathbf{x},y)$, $\forall\ y\neq y_{\mathbf{x}_t}$, is set to a prespecified constant k. The evaluation criteria of Relief and ReliefF suggest that the two algorithms select features contributing to the separation of samples from different classes. In Section 3, we show that this is equivalent to selecting features that preserve a special form of sample similarity derived from the class label.

2.6 Feature Selection with HSIC

HSIC is first proposed in [15] for measuring the dependence between two kernels. In [35], HSIC is extended and applied to feature selection. The basic idea is to select a subset of features, such that the obtained kernel maximizes the HSIC criterion with respect to a given kernel matrix $\mathbf{K}$. In [35] an unbiased estimator of HSIC is given as

$$SC_H(\mathbf{F}) = \frac{1}{n(n-3)}\left[\text{Trace}(\mathbf{K_F K}) + \frac{\mathbf{1}^\top \mathbf{K_F} \mathbf{1}\mathbf{1}^\top \mathbf{K1}}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1}^\top \mathbf{K_F K1}\right]. \quad (2.4)$$

In the equation, $\mathbf{F}$ is a subset of the original features and $\mathbf{K_F}$ is the kernel obtained from $\mathbf{F}$. To achieve unbiased estimation, HSIC criterion requires that the diagonal elements of $\mathbf{K}$ and $\mathbf{K_F}$ be set to 0. Based on the HSIC criterion, features can be selected via either backward elimination or forward selection.

3 A Unified Framework for Existing Feature Selection Criteria

In this part, we show that all the criteria reviewed in Section 2 can be unified under a common framework, which has the following formulation:

$$SC(\mathbf{f}) = \hat{\mathbf{f}}^\top \hat{\mathbf{K}} \hat{\mathbf{f}}, \quad (3.1)$$

where $\hat{\mathbf{f}}$ is the normalized feature vector obtained from $\mathbf{f}$, and $\hat{\mathbf{K}}$ is the refined similarity matrix derived from $\mathbf{K}$. The difference between different criteria is that they use different rules to generate $\hat{\mathbf{f}}$ and $\hat{\mathbf{K}}$.

We can show that using Eq. (3.1) to select k features can be formulated as

$$\max_{\mathbb{F}_{sub}} \sum\nolimits_{\mathbf{f}\in\mathbb{F}_{sub}} SC(\mathbf{f}) = \max_{\mathbb{F}_{sub}} \sum\nolimits_{\mathbf{f}\in\mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \hat{\mathbf{K}} \hat{\mathbf{f}}, \quad (3.2)$$

where $\mathbb{F}_{sub}$ is the set of k selected features. It has been shown in [33], that solving the following problem:

$$\max\nolimits_{\mathbf{S}\succeq 0} \text{Trace}\left(\mathbf{S}\hat{\mathbf{K}}\right) \quad st. \quad \text{Trace}(\mathbf{S}) \le 1,$$

will result in a kernel matrix $\mathbf{S}$, which well preserves the sample similarity specified in $\hat{\mathbf{K}}$. Also, we have

$$\begin{aligned}\max_{\mathbb{F}_{sub}} \sum_{\mathbf{f}\in\mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \hat{\mathbf{K}} \hat{\mathbf{f}} &= \max_{\mathbb{F}_{sub}} \sum_{\mathbf{f}\in\mathbb{F}_{sub}} \text{Trace}\left(\hat{\mathbf{f}}\,\hat{\mathbf{f}}^\top \hat{\mathbf{K}}\right) \\ &= \max_{\mathbb{F}_{sub}} \text{Trace}\left\{\hat{\mathbf{K}} \sum_{\mathbf{f}\in\mathbb{F}_{sub}} \hat{\mathbf{f}}\,\hat{\mathbf{f}}^\top\right\} \\ &= \max_{\mathbb{F}_{sub}} \text{Trace}\left\{\hat{\mathbf{K}}\left(\mathbf{X}_{\mathbb{F}_{sub}}^\top \mathbf{X}_{\mathbb{F}_{sub}}\right)\right\}.\end{aligned}$$

Here $\mathbf{X}_{\mathbb{F}_{sub}}$ is the data containing only the features in $\mathbb{F}_{sub}$. Thus $\max_{\mathbb{F}_{sub}} \sum_{\mathbf{f}\in\mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \hat{\mathbf{K}} \hat{\mathbf{f}}$ equals to select a set of features $\mathbb{F}_{sub}$, such that the linear kernel on $\mathbf{X}_{\mathbb{F}_{sub}}$ can preserve the pairwise sample similarity specified in $\hat{\mathbf{K}}$ well. Therefore the features that maximize the value of Eq. (3.2) should have strong capability in preserving the pairwise sample similarity specified in $\hat{\mathbf{K}}$. Below we show each criterion introduced in Section 2 can be reformulated in the form of Eq. (3.2).

3.1 Laplacian Score

Given a set of features, $\mathbf{f}_1, \ldots, \mathbf{f}_m$, the Laplacian score for the ith feature can be expressed as [19]

$$SC_L(\mathbf{f}_i) = \frac{\widetilde{\mathbf{f}}_i^\top \mathbf{L} \widetilde{\mathbf{f}}_i}{\widetilde{\mathbf{f}}_i^\top \mathbf{D} \widetilde{\mathbf{f}}_i}, \quad \text{with} \quad \widetilde{\mathbf{f}}_i = \mathbf{f}_i - \frac{\mathbf{f}_i^\top \mathbf{D1}}{\mathbf{1}^\top \mathbf{D1}}\mathbf{1} = \mathbf{f}_i - \mu_i \mathbf{1},$$

where μ_i is the density-weighted mean of $\mathbf{f}_i$. Since $\mathbf{L} = \mathbf{D} - \mathbf{K}$, where $\mathbf{K}$ is the similarity matrix and $\mathbf{D}$ is the degree matrix, we have

$$SC_L(\mathbf{f}_i) = 1 - \frac{\widetilde{\mathbf{f}}_i^\top \mathbf{K} \widetilde{\mathbf{f}}_i}{\widetilde{\mathbf{f}}_i^\top \mathbf{D} \widetilde{\mathbf{f}}_i} = 1 - \left(\frac{\widetilde{\mathbf{f}}_i}{\|\mathbf{D}^{\frac{1}{2}} \widetilde{\mathbf{f}}_i\|}\right)^\top \mathbf{K} \left(\frac{\widetilde{\mathbf{f}}_i}{\|\mathbf{D}^{\frac{1}{2}} \widetilde{\mathbf{f}}_i\|}\right).$$

Note that $\widetilde{\mathbf{f}}_i^\top \mathbf{D} \widetilde{\mathbf{f}}_i$ is the density weighted variance of $\mathbf{f}_i$, denoted as σ_i^2 [19]. Since $\widetilde{\mathbf{f}}_i^\top \mathbf{D} \widetilde{\mathbf{f}}_i = \|\mathbf{D}^{\frac{1}{2}} \widetilde{\mathbf{f}}_i\|_2^2$, we can interpret $\|\mathbf{D}^{\frac{1}{2}} \widetilde{\mathbf{f}}_i\|$ as the standard deviation of $\mathbf{f}_i$. Therefore, we can write $SC_L(\mathbf{f}_i)$ as

$$SC_L(\mathbf{f}_i) = 1 - \hat{\mathbf{f}}_i^\top \mathbf{K} \hat{\mathbf{f}}_i. \quad (3.3)$$

where $\hat{\mathbf{f}}_i \triangleq (\mathbf{f}_i - \mu_i \mathbf{1})/\sigma_i$, which is a normalized $\mathbf{f}_i$. Therefore the problem of using Laplacian score to select k features can be formulated as

$$\max_{\mathbb{F}_{sub}} \sum\nolimits_{\mathbf{f}\in\mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \mathbf{K} \hat{\mathbf{f}}, \quad \hat{\mathbf{f}} \triangleq (\mathbf{f} - \mu \mathbf{1})/\sigma. \quad (3.4)$$

The problem can be solved by greedily picking the top k features which have the maximal $\hat{\mathbf{f}}^\top \mathbf{K} \hat{\mathbf{f}}$ values.

3.2 SPEC

SPEC is an extension for Laplacian score to make it more robust to noise. First, let us assume that the data is noise free, for which, we can set $\gamma(x) = x$ and $k = n$. In this case, we can show that the $\text{SC}_{S,2}(\cdot)$ of SPEC is equivalent to Laplacian score [48]. Similar as in Section 3.1, we can also rewrite the $\text{SC}_{S,1}(\cdot)$ and $\text{SC}_{S,3}(\cdot)$ of SPEC in the following forms, respectively:

$$\text{SC}_{S,1}(\mathbf{f}) = 1 - \hat{\mathbf{f}}^\top \mathbf{K} \hat{\mathbf{f}}, \quad \text{SC}_{S,3}(\mathbf{f}) = 1 + \hat{\mathbf{f}}^\top \mathbf{K} \hat{\mathbf{f}}, \quad (3.5)$$

where $\hat{\mathbf{f}}_i \triangleq \mathbf{f}_i/\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}_i\|$. Since $\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}_i\|$ is the density weighted norm of $\mathbf{f}_i$, $\hat{\mathbf{f}}_i$ forms the density weighted normalized $\mathbf{f}_i$. The analysis shows that when noise is not considered, the problem of using SPEC to select k features can be formulated as

$$\max_{\mathbb{F}_{sub}} \sum\nolimits_{\mathbf{f}\in\mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \mathbf{K} \hat{\mathbf{f}}, \quad \hat{\mathbf{f}} \triangleq \mathbf{f}/\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|. \quad (3.6)$$

When noise reduction mechanisms are applied, selecting k features with SPEC can be formulated as

$$\max_{\mathbb{F}_{sub}} \sum\nolimits_{\mathbf{f}\in\mathbb{F}_{sub}} \hat{\mathbf{f}}_{(i)}^\top \hat{\mathbf{K}}_{(i)} \hat{\mathbf{f}}_{(i)}, \text{ where}$$

$$\hat{\mathbf{f}}_{(1)} \triangleq \mathbf{f}/\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|, \quad \hat{\mathbf{K}}_{(1)} = \mathbf{D}^{\frac{1}{2}} \mathbf{U} \left(\mathbf{I} - \gamma(\mathbf{\Sigma})\right) \mathbf{U}^\top \mathbf{D}^{\frac{1}{2}},$$

$$\hat{\mathbf{f}}_{(2)} \triangleq (\mathbf{f} - \mu \mathbf{1})/\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|, \hat{\mathbf{K}}_{(2)} = \mathbf{D}^{\frac{1}{2}} \mathbf{U} \left(\mathbf{I} - \gamma(\mathbf{\Sigma})\right) \mathbf{U}^\top \mathbf{D}^{\frac{1}{2}},$$

$$\hat{\mathbf{f}}_{(3)} \triangleq \mathbf{f}/\|\mathbf{D}^{\frac{1}{2}} \mathbf{f}\|, \quad \hat{\mathbf{K}}_{(3)} = \mathbf{D}^{\frac{1}{2}} \mathbf{U}_k \left(\gamma(2\mathbf{I}) - \gamma(\mathbf{\Sigma}_k)\right) \mathbf{U}_k^\top \mathbf{D}^{\frac{1}{2}},$$

where (i) denotes the i-th criterion defined in SPEC. $\mathbf{U}$ and $\mathbf{\Sigma}$ contain the singular vectors and singular values of the normalized Laplacian matrix, $\mathcal{L} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$. In this case, the original similarity matrix $\mathbf{K}$ is mapped to $\hat{\mathbf{K}}$ by the operation $\gamma(\cdot)$ on $\mathbf{\Sigma}$ for noise reduction.

3.3 Fisher Score

As shown in [19], when $\mathbf{K}$ is defined as

$$\mathbf{K}_{ij} = \begin{cases} \frac{1}{n_l}, & y_i = y_j = l \\ 0, & \text{otherwise} \end{cases}. \tag{3.7}$$

Laplacian Score and Fisher Score are equivalent and have the following relationship:

$$\text{SC}_L = \frac{1}{1 + \text{SC}_F}. \tag{3.8}$$

Therefore the analysis in Section 3.1 also applies to the case of Fisher Score.

3.4 Trace Ratio Criterion

Trace-ratio criterion [27] selects k features via maximizing the following criterion:

$$SC_{TR}(\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \ldots, \mathbf{f}_{i_k}) = \frac{\text{Trace}(\hat{\mathbf{X}}^\top \mathbf{L}_b \hat{\mathbf{X}})}{\text{Trace}(\hat{\mathbf{X}}^\top \mathbf{L}_w \hat{\mathbf{X}})},$$

where $\hat{\mathbf{X}} = (\mathbf{f}_{i_1}, \ldots, \mathbf{f}_{i_k})$. Without loss of generality, we assume that each feature $\mathbf{f}_i$ $(i = 1, \cdots, m)$ has been centered to zero density weighted mean, i.e. $\mathbf{f}^\top \mathbf{D}\mathbf{1} = 0$. In [27], with different definitions of $\mathbf{L}_b$ and $\mathbf{L}_w$, the trace-ratio criterion generalizes Fisher Score and Lapalacian Score to evaluate a feature subset as a whole. For the Fish Score case, we can show that,

$$SC_{TR}(\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \ldots, \mathbf{f}_{i_k}) = \frac{\sum_{t=1}^k \mathbf{f}_{i_t}^\top \mathbf{K} \mathbf{f}_{i_t}}{\sum_{t=1}^k \mathbf{f}_{i_t}^\top \mathbf{L} \mathbf{f}_{i_t}}, \tag{3.9}$$

where $\mathbf{K}$ is defined in Eq. (3.7). With this definition of $\mathbf{K}$, it can be shown that $\mathbf{D} = \mathbf{I}$ and $\mathbf{L} = \mathbf{I} - \mathbf{K}$. Therefore, maximizing Eq. (3.9) is equivalent to maximizing

$$\frac{\sum_{t=1}^k \mathbf{f}_{i_t}^\top \mathbf{K} \mathbf{f}_{i_t}}{\sum_{t=1}^k \mathbf{f}_{i_t}^\top \mathbf{D} \mathbf{f}_{i_t}} = \frac{\sum_{t=1}^k \mathbf{f}_{i_t}^\top \mathbf{K} \mathbf{f}_{i_t}}{\sum_{t=1}^k \mathbf{f}_{i_t}^\top \mathbf{f}_{i_t}} = \frac{\text{Trace}(\hat{\mathbf{X}}^\top \mathbf{K} \hat{\mathbf{X}})}{\text{Trace}(\hat{\mathbf{X}}^\top \hat{\mathbf{X}})}. \tag{3.10}$$

Note that $\text{Trace}(\hat{\mathbf{X}}^\top \hat{\mathbf{X}})$ is just the sum of the weighted variances of $\mathbf{f}_{i_1}, \cdots, \mathbf{f}_{i_k}$. Assuming that features are normalized to have unit norm, this term is a constant and the above problem can be formulated as

$$\max_{\mathbb{F}_{sub}} \sum\nolimits_{\mathbf{f} \in \mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \mathbf{K} \hat{\mathbf{f}}, \quad \hat{\mathbf{f}} \triangleq \mathbf{f}/\|\mathbf{f}\|. \tag{3.11}$$

When $\mathbf{K}_w = \mathbf{K}$ and $\mathbf{K}_b = \left(\mathbf{1}^\top \mathbf{D}\mathbf{1}\right)^{-1} \mathbf{D}\mathbf{1}\mathbf{1}^\top \mathbf{D}$, the trace ratio criterion generalizes the Laplacian Score as

$$SC_{TR}(\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \ldots, \mathbf{f}_{i_k}) = \frac{\sum_{t=1}^k \mathbf{f}_{i_t}^\top \mathbf{D} \mathbf{f}_{i_t}}{\sum_{t=1}^k \mathbf{f}_{i_t}^\top \mathbf{L} \mathbf{f}_{i_t}}, \tag{3.12}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{K}$. In this case, assuming that $\hat{\mathbf{f}}$ is defined as $\hat{\mathbf{f}} \triangleq \mathbf{f}/\|\mathbf{D}^{\frac{1}{2}}\mathbf{f}\|$, maximizing $SC_{TR}(\cdot)$ will be equivalent to solving the following problem:

$$\max_{\mathbb{F}_{sub}} \sum\nolimits_{\mathbf{f} \in \mathbb{F}_{sub}} \hat{\mathbf{f}}^\top \mathbf{K} \hat{\mathbf{f}}. \tag{3.13}$$

The above analysis shows, with another way, the connections of Laplacian score and Fisher score to feature selection based on similarity preserving. Similar analysis also holds, when general $\mathbf{K}_w$ and $\mathbf{K}_b$ are used in the trace-ratio criterion, by noticing the fact that

$$SC_{TR}\left(\mathbf{f}_{i_1}^*, \mathbf{f}_{i_2}^*, \ldots, \mathbf{f}_{i_k}^*\right) = \sum_{t=1}^k \mathbf{f}^{*\top}_{i_t} \hat{\mathbf{K}} \mathbf{f}_{i_t}^*,$$

where $\mathbf{f}_{i_1}^*, \ldots, \mathbf{f}_{i_k}^*$ is the optimal feature set of size k, $\hat{\mathbf{K}} = (\mathbf{L}_b - \lambda^* \mathbf{L}_w)$, and λ^* is the optimal λ value obtained by solving Eq. (2.3) iteratively.

3.5 ReliefF

Assume that the training data has c classes with l instances in each class; there are k instances in both $NH(\mathbf{x})$ and $NM(\mathbf{x})$; and all features have been normalized to unit length. As shown in [48], under the above assumptions, the evaluation criterion of ReliefF is equivalent to

$$\sum_{i=1}^n \left(\sum_{j=1}^k \frac{1}{k} (f_i - f_{NH(\mathbf{x}_i)_j})^2 - \sum_{y \neq y_i} \frac{\sum_{j=1}^k (f_i - f_{NM(\mathbf{x}_i, y)_j})^2}{(c-1)k} \right).$$

In the equation, f_i is the value of the feature $\mathbf{f}$ on the ith instance, $\mathbf{x}_i$; $NH(\mathbf{x}_i)_j$ denotes the jth nearest hit of $\mathbf{x}_i$; and $NM(\mathbf{x}_i, y)_j$ denotes the jth nearest miss of $\mathbf{x}_i$ in class y. Here we use the Euclidean distance to calculate the difference between two values and use all training data to train ReliefF. When $\mathbf{K}$ is defined as

$$\mathbf{K}_{i,j} = \begin{cases} 1 & i = j \\ -\frac{1}{k} & x_j \in NH(\mathbf{x}_i) \\ \frac{1}{(c-1)k} & x_j \in NM(\mathbf{x}_i, y) \end{cases}, \tag{3.14}$$

it is easy to verify that $\mathbf{D} = \mathbf{I}$, and the evaluation criterion of ReliefF is equivalent to $\text{SC}_R(\mathbf{f}) = -1 + \mathbf{f}^\top \mathbf{K} \mathbf{f}$ up to a constant [48]. Therefore using ReliefF to select k features can be formulated as

$$\max_{\mathbb{F}_{sub}} \sum\nolimits_{\mathbf{f} \in \mathbb{F}_{sub}} \mathbf{f}^\top \mathbf{K} \mathbf{f}. \tag{3.15}$$

3.6 HSIC Criterion

It is shown in [35] that when linear kernel is used to compute the $\mathbf{K}_\mathbf{F}$ in Eq. (2.4), the HSIC criterion can be formulated as

$$SC_H(\mathbf{f}_{i_1}, \ldots, \mathbf{f}_{i_k}) = \sum_{t=1}^k \mathbf{f}_{i_t}^\top \hat{\mathbf{K}} \mathbf{f}_{i_t}, \text{ where}$$

$$\hat{\mathbf{K}} = \frac{1}{n(n-3)} \left[\mathbf{K} + \left(\mathbf{1}\mathbf{1}^\top - \mathbf{I}\right) \frac{\mathbf{1}^\top \mathbf{K} \mathbf{1}}{(n-1)(n-2)} - \frac{2}{n-2} \left(\mathbf{K}\mathbf{1}\mathbf{1}^\top - \text{diag}(\mathbf{K}\mathbf{1}) \right) \right]. \tag{3.16}$$

When a nonlinear kernel function is used for constructing $\mathbf{K_F}$, features' contribution to $\mathbf{K_F}$ cannot be simply decomposed as in the linear kernel case. But in this case the HSIC criterion becomes computationally inefficient.

3.7 Discussion

The above analysis shows that many existing feature evaluation criteria can be unified under a common formulation, which suggests that although different criteria are designed to serve different purposes and are of very different formulations, they all in fact evaluate features by measuring their capability in preserving the pairwised sample similarity specified by a predefined similarity matrix. The difference between different criteria is that they use different rules to normalize features and adjust the given similarity matrix in different ways.

Eq. (3.2) also reveals that any criterion under this formulation will evaluate features individually, hence, cannot handle redundant features. Redundant features increase dimensionality unnecessarily [17], and worsen learning performance when facing the shortage of data. It is also shown empirically that removing redundant features can result in significant performance improvement [2], [7], [8], [45], [50]. Below, we propose a new framework for similarity preserving feature selection. And it forms a natural way for handling redundant features in similarity preserving feature selection.

4 THE SPFS FRAMEWORK

In this section, we propose a novel framework for *S*imilarity *P*reserving *F*eature *S*election, named SPFS. We connect SPFS to the aforementioned algorithms and show how it extends these algorithms by overcoming their limitation on handling redundant features.

4.1 The basic formulation

In similarity preserving feature selection, given m features, $(\mathbf{f}_1, \ldots, \mathbf{f}_m)$, we aim to select k features, based on which, the pairwised sample similarity specified by a predefined similarity matrix $\mathbf{K}$ is best preserved. The matrix $\mathbf{K}$ can be constructed either by using the label information in supervised learning or using certain distance metrics in unsupervised learning. Hence, $\mathbf{K}$ essentially encodes the class information or the intrinsic structure of data. By preserving the sample similarity specified in $\mathbf{K}$, we are able to select a subset of features that can maintain or even improve the performance of learning models. Below, we propose the basic formulation to capture the idea of similarity preserving feature selection, which defines a combinatorial optimization problem. The formulation uses linear kernel to measure the similarities among samples in the dimensionality reduced space generated by feature selection.

$$\begin{aligned}(P)\quad & \min_{\mathbf{W}} \ \|\hat{\mathbf{X}}\hat{\mathbf{X}}^\top - \mathbf{K}\|_F^2 \\ s.t.\quad & \hat{\mathbf{X}} = \mathbf{XW},\ \mathbf{W} \in \{0,1\}^{m\times k}, \\ & \mathbf{W}^\top \mathbf{1}_{m\times 1} = \mathbf{1}_{k\times 1},\quad \|\ \mathbf{W}\mathbf{1}_{k\times 1}\ \|_0 = k. \end{aligned} \tag{4.1}$$

$\|\cdot\|_F$ denotes the Frobenius matrix norm and $\|\cdot\|_0$ is the zero norm of a vector. These constraints imply that (1) $\mathbf{W}$ is a selection matrix with only "0" or "1" as its elements. $\hat{\mathbf{X}} = \mathbf{XW}$ exactly selects k columns out of $\mathbf{X}$; (2) each column of $\mathbf{W}$ has one and only one "1". This ensures the original features rather than a linear combination of them to be selected; (3) among the m rows of $\mathbf{W}$, only k rows contain one "1" exactly, and the remaining $m-k$ rows are just zero vectors. This guarantees that none of the m features will be repeatedly selected. Altogether, the three constraints ensure that $\hat{\mathbf{X}}$ contains k different original features of $\mathbf{X}$. The selected k features can be expressed as $\hat{\mathbf{X}} = \mathbf{XW} = (\mathbf{f}_{i_1}, \ldots, \mathbf{f}_{i_k})$, where $\{i_1, \ldots, i_k\}$ is a subset of $\{1, \ldots, m\}$ with the cardinality of k.

According to the definition of Frobenius norm, minimizing Eq. (4.1) is equivalent to minimizing

$$\text{Trace}\left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\right) - 2\ \text{Trace}\left(\hat{\mathbf{X}}^\top\mathbf{K}\hat{\mathbf{X}}\right) + \text{Trace}\left(\mathbf{KK}\right).$$

Since $\text{Trace}\left(\mathbf{KK}\right)$ is a constant, this essentially boils down to a joint optimization of the following two terms:

$$\min_{\mathbf{W}} \text{Trace}\left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\right) \quad \& \quad \max_{\mathbf{W}} \text{Trace}\left(\hat{\mathbf{X}}^\top\mathbf{K}\hat{\mathbf{X}}\right). \tag{4.2}$$

It is not difficult to see that $\text{Trace}(\hat{\mathbf{X}}^\top\mathbf{K}\hat{\mathbf{X}}) = \sum_{j=1}^{k} \mathbf{f}_{i_j}^\top\mathbf{K}\mathbf{f}_{i_j}$. As discussed, maximizing this term leads to selecting the features that can preserve the pairwised sample similarity specified by $\mathbf{K}$.

It is interesting to study the effect of minimizing $\text{Trace}(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\hat{\mathbf{X}}\hat{\mathbf{X}}^\top)$, an extra term in our similarity preserving feature selection criterion. Without loss of generality, we assume that all the features have been centralized to have zero mean. Note that the (r,s) element of $\hat{\mathbf{X}}^\top\hat{\mathbf{X}}$ is just $\mathbf{f}_{i_r}^\top\mathbf{f}_{i_s}$. It can be obtained that

$$\begin{aligned} & \text{Trace}\left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\right) = \langle\hat{\mathbf{X}}^\top\hat{\mathbf{X}}, \hat{\mathbf{X}}^\top\hat{\mathbf{X}}\rangle_F \\ = & \sum_{r=1}^{k}\sum_{s=1}^{k}\left(\mathbf{f}_{i_r}^\top\mathbf{f}_{i_s}\right)^2 = \sum_{r=1}^{k} n^2\sigma_{i_r}^4 + 2\sum_{i_r>i_s\geq 1}^{k} n^2\sigma_{i_r}^2\sigma_{i_s}^2\rho_{i_r,i_s}^2, \end{aligned}$$

where ρ_{i_r,i_s} denotes the Pearson correlation [39] between x_{i_r} and x_{i_s}, and σ_{i_r} is the standard deviation of x_{i_r}. The last step is due to the fact that all feature vectors $\mathbf{f}_1, \ldots, \mathbf{f}_m$ have been centered to have zero means. In this case, $\mathbf{f}_{i_r}^\top\mathbf{f}_{i_s}$ equals to ρ_{i_r,j_s} scaled by $n\sigma_{i_r}\sigma_{j_s}$. From this, it can be found that minimizing $\text{Trace}\left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\right)$ prefers to select the features which are less correlated to each other. Moreover, if all the features have been standardized to conform the normal distribution $\mathcal{N}(0,1)$, minimizing $\text{Trace}\left(\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\hat{\mathbf{X}}\hat{\mathbf{X}}^\top\right)$ simply minimizes the sum of the squared correlation coefficient among the selected k features. Hence, the extra term in our selection criterion penalizes the selection of strongly correlated features. This is an important observation. It clearly shows that in evaluating features, the proposed criterion considers both similarity preservation and pairwised correlation among features in a natural way, which allows it to avoid choosing redundant features. This effect will be empirically demonstrated in the Section 6.

4.2 Problem Reformulation

Optimizing Eq. (4.1) is an integer programming problem, which is NP-hard and cannot be efficiently solved even for medium m and k. Also, in the formulation, the sample similarity measurement is restricted to the inner product between samples (linear kernel function). To address the two issues, we propose to reformulated the problem in the following form:

$$(P1) \quad \min_{\mathbf{W}} \ \|\hat{\mathbf{X}}\hat{\mathbf{X}}^\top - \mathbf{K}\|_F^2$$
$$s.t. \quad \hat{\mathbf{X}} = \mathbf{X}\mathbf{W}, \quad \mathbf{W} \in \mathbb{R}^{m\times l}, \quad \|\mathbf{W}\|_{2,0} = k. \quad (4.3)$$

$\|\mathbf{W}\|_{2,0}$ is the $L_{2,0}$-norm of the matrix $\mathbf{W}$. It counts the number of non-zero rows of $\mathbf{W}$. $\|\mathbf{W}\|_{2,0} = k$ means that only k rows of $\mathbf{W}$ are non-zero vectors. Since each feature corresponding to a row of $\mathbf{W}$, the constraint enforces that only k features will be selected, whose corresponding rows in $\mathbf{W}$ have non-zero elements. In Eq. (4.3), l is the number of columns of $\mathbf{W}$, which is determined by the character of $\mathbf{K}$. We will discuss how to choose a proper value for l in the later part of this section. For our problem, this reformulation not only makes the optimization problem easier by relaxing $\mathbf{W}$ to continuous domain, but also helps to identify features that can better preserve sample similarity. In Eq. (4.1), $\mathbf{W}_P \in \{0,1\}^{m\times k}$ enforces sample similarity be calculated from the inner product based on the k selected features:

$$< \mathbf{W}_P^\top \mathbf{x}_i, \ \mathbf{W}_P^\top \mathbf{x}_j >= \sum_{r=1}^{k} x_{i_r} x_{j_r}. \quad (4.4)$$

While in Eq. (4.3), the sample similarity is calculated using the inner product based on a linear transformation of the k selected features:

$$< \mathbf{W}_{P1}^\top \mathbf{x}_i, \ \mathbf{W}_{P1}^\top \mathbf{x}_j >= \sum_{r=1}^{k} \hat{x}_{i_r} \hat{x}_{j_r}, \quad \hat{x}_{i_r} = \mathbf{w}_r^\top \mathbf{x}_i. \quad (4.5)$$

Here $\mathbf{w}_{r=1\ldots l}$ is the rth column of $\mathbf{W}_{P1}$, which contains at most k elements to be non-zero, due to the constraint. Clearly, Eq. (4.4) is a special case of Eq. (4.5). Eq. (4.5) provides a more comprehensive way to evaluate features' capability in preserving similarity by considering their linear transformations. It is known that in the classification or clustering stage after feature selection, we can always apply a linear transform on the selected features to make the sample similarity align better with $\mathbf{K}$. Therefore we merge the two steps: *feature selection* and *linear transformation* into one through the above reformulation. More specifically, we directly select a set of k features such that they can best approximate $\mathbf{K}$ under their linear transform. This better serves our goal on maximizing the learning performance.

However, the L_0-norm in $\|\mathbf{W}\|_{2,0} = k$ is non-convex. And the optimization problem specified in Eq. (4.5) is still difficult to solve. Following the common strategy for addressing this issue, the convex L_1-norm is adopted to approximate the L_0-norm. And we reformulate the problem $(P1)$ specified in Eq. (4.5) as

$$(P2) \quad \min_{\mathbf{W}} \ \|\hat{\mathbf{X}}\hat{\mathbf{X}}^\top - \mathbf{K}\|_F^2$$
$$s.t. \quad \hat{\mathbf{X}} = \mathbf{X}\mathbf{W}, \ \mathbf{W} \in \mathbb{R}^{m\times l}, \ \|\mathbf{W}\|_{2,1} \le t. \quad (4.6)$$

Here, t ($t > 0$) is a hyper-parameter and its optimal value can be set via cross-validation. The $L_{2,1}$-norm constraint enforces the columns of $\mathbf{W}$ to share similar patterns (different columns of $\mathbf{W}$ tend to have nonzero entries on the same position) [3], [24], which is a nice property serves our feature selection purpose.

The constraint $\|\mathbf{W}\|_{2,1} \le t$ corresponds to an $L_{2,1}$ ball, which is a convex set. Unfortunately, the objective function $\|\hat{\mathbf{X}}\hat{\mathbf{X}}^\top - \mathbf{K}\|_F^2$ is not convex with respect to $\mathbf{W}$. Traditional methods such as sub-gradient descent[1] may be applied to find a local minimum. The solution is, however, still not computationally efficient, because the number of variables, mk, can be very large in a feature selection problem. It can be shown that given $\mathbf{K}$, an optimal solution of $\min_{\mathbf{A}\in\mathbb{R}^{n\times l}} \ \|\mathbf{A}\mathbf{A}^\top - \mathbf{K}\|_F^2$ is

$$\mathbf{A}^\star = \mathbf{\Gamma}_l \mathbf{\Lambda}_l^{1/2}, \quad (4.7)$$

where the columns of $\mathbf{\Gamma}_l$ are the l eigenvectors of $\mathbf{K}$ corresponding to the l largest eigenvalues, which form the diagonal elements of the diagonal matrix $\mathbf{\Lambda}_l$. Based on this observation, we further reformulate (P2) as

$$(P3) \quad \min_{\mathbf{W}} \ \|\hat{\mathbf{X}} - \mathbf{A}^\star\|_F^2$$
$$s.t. \quad \hat{\mathbf{X}} = \mathbf{X}\mathbf{W}, \ \mathbf{W} \in \mathbb{R}^{m\times l}, \ \|\mathbf{W}\|_{2,1} \le t. \quad (4.8)$$

The following thereom shows minimizing $\|\hat{\mathbf{X}} - \mathbf{A}^\star\|_F$ is a good approximation to minimizing $\|\hat{\mathbf{X}}\hat{\mathbf{X}}^\top - \mathbf{K}\|_F$.

***Theorem** 1: Let $\mathbf{A}^\star = \mathbf{\Gamma}_l \mathbf{\Lambda}_l^{1/2}$ and $\mathbf{\Omega} = \hat{\mathbf{X}} - \mathbf{A}^\star$. For $\hat{\mathbf{X}} \in \mathbb{R}^{n\times l}$, we have the following bounds:*

$$\|\mathbf{A}^\star\mathbf{A}^{\star\top} - \mathbf{K}\|_F \le \|\hat{\mathbf{X}}\hat{\mathbf{X}}^\top - \mathbf{K}\|_F$$
$$\le 2(\|\mathbf{A}^\star\|_F + \|\mathbf{\Omega}\|_F)\|\mathbf{\Omega}\|_F + \|\mathbf{A}^\star\mathbf{A}^{\star\top} - \mathbf{K}\|_F.$$

■

The proof of the theorem is given in Appendix. In the theorem $\|\mathbf{A}^\star\mathbf{A}^{\star\top} - \mathbf{K}\|_F$ is a constant when $\mathbf{K}$ is given, which is $\sum_{i=l+1}^{n} \lambda_i$, the summation of the $(n-l)$ smallest eigenvalues of $\mathbf{K}$. Given $m \gg \max(l, n)$, the objective of Eq. (4.8) is underdetermined and the $\|\mathbf{\Omega}\|_F$ can usually be small. As seen, by minimizing $\|\mathbf{\Omega}\|_F$, we minimize the upper bound of $\|\hat{\mathbf{X}}\hat{\mathbf{X}}^\top - \mathbf{K}\|_F$. Eq. (4.8) defines a multiple-output regression problem [18], which has a smooth and convex objective function with a convex $L_{2,1}$-norm constraint. Its global optimal solution can be efficiently obtained via convex optimization techniques[2].

1. The sub-gradient descent method needs to be used in this case, as $\|\mathbf{W}\|_{2,1}$ is non-differentiable at certain points in the space.

2. After solving problem $(P3)$, we may also use the obtained solution as a starting point to solve problem $(P2)$.

5 New Feature Selection Algorithms Based on The SPFS Framework

To turn SPFS formulations into efficient feature selection algorithms, we need to develop efficient techniques to solve the objective functions specified in SPFS. More specifically, we propose one greedy algorithm to solve Eq. (4.1), which specifies our original problem. We also propose two algorithms to solve Eq. (4.8), which corresponds to the reformulated formulation. Among the two, the first algorithm is based on Nesterov's method for constrained smooth convex optimization [24], [25], [26], which generates an exact solution for the problem. The second algorithm is based on group LAR [46], which generates an approximate solution path for the problem. The advantage of this algorithm is that it does not require the tedious process of parameter tuning, which is a limitation of the first one. The performance of the three algorithms will be compared via experiments.

5.1 SPFS-SFS

The problem specified in Eq. (4.1) is NP-hard. Treating its objective function as a feature selection criterion, we propose to combine it with the traditional forward search strategy for feature selection. The pseudo-code of the algorithm can be found in Algorithm 1. Note that card $(\cdot)$ in line 2 returns the size of a set; and lines 3 and 7 make use of the following fact:

$$\mathbf{K} - \mathbf{X}_{\mathcal{A}}\mathbf{X}_{\mathcal{A}}^{\top} = \mathbf{K} - \sum\nolimits_{i\in\mathcal{A}} \mathbf{f}_i\mathbf{f}_i^{\top}, \text{ where}$$
$$\mathbf{X}_{\mathcal{A}} = (\mathbf{f}_{i_1}, \ldots, \mathbf{f}_{i_k}),\ i_p \in \mathcal{A},\ p = 1, \ldots, k.$$

It allows us to remove $\mathbf{f}_i\mathbf{f}_i^{\top}$ from $\mathbf{R}$, once $\mathbf{f}_i$ is selected, which will make the algorithm more efficient. To select k features, SPFS-SFS needs to run k iterations. In each iteration, it greedily picks one feature that is most consistent with the current residue $\mathbf{R}$, adds the feature to $\mathcal{A}$ and updates the residue by deducting $\mathbf{f}_i\mathbf{f}_i^{\top}$ from $\mathbf{R}$. The condition specified in line 4 of Algorithm 1 guarantees that after each step, $\|\mathbf{R}\|_F^2$ monotonically decreases. It is easy to show that the time complexity of using SPFS-SFS to solve the problem specified in Eq. (4.1) is $O\left(kmn^2\right)$, where k is the number of selected features.

5.2 SPFS-NES

Eq. (4.8) defines a multiple-output regression problem with an $L_{2,1}$-norm constraint. It is a constrained smooth convex optimization problem and can be efficiently solved by using the Nesterov's method [24], [25], [26]. Nesterov's method is an optimal first-order black-box method for smooth convex optimization. The pseudo-code of Nesterov's method for constrained smooth convex optimization is shown in Algorithm 2. In Algorithm 2, $f(\mathbf{W}) = \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2$; $f'(\mathbf{W})$ is the gradient of $f(\mathbf{W})$ with respect to $\mathbf{W}$; $f_{\gamma,\mathbf{S}}(\mathbf{W})$ is the regularized tangent line of $f(\cdot)$ at $\mathbf{W}$; and $\pi_t(\mathbf{W})$ is the Euclidian projection of $\mathbf{W}$ onto the domain of the problem, which can be obtained by solving

$$\pi_t(\mathbf{W}) = \arg\min_{\mathbf{U}} \frac{1}{2}\|\mathbf{W} - \mathbf{U}\|_2^2,\ \|\mathbf{U}\|_{2,1} \le t. \tag{5.1}$$

In [24], an efficient approach is proposed to analytically compute the Euclidian projection of $\mathbf{W}$ onto an $L_{2,1}$-norm ball. We refer readers to that paper for more details on the approach. It is shown that the time complexity of using SPFS-NES to solve Eq. (4.8) is $O\left(\frac{1}{\epsilon}(mn + nk)\right)$, where ϵ is the error of the solver [24].

Algorithm 1: SPFS-SFS

Input: $\mathbf{f}_1, \ldots, \mathbf{f}_m,\ \mathbf{K},\ k$
Output: $\mathcal{A}$ - the selected features

1 $\mathcal{A} = \phi, \mathbf{R} = \mathbf{K}$;
2 **while** *card* $(\mathcal{A}) < k$ **do**
3 $\quad i^* = \arg\min_{i\notin\mathcal{A}} \|\mathbf{R} - \mathbf{f}_i\mathbf{f}_i^{\top}\|_F^2$;
4 $\quad$ **if** $\|\mathbf{R} - \mathbf{f}_i\mathbf{f}_i^{\top}\|_F^2 > \|\mathbf{R}\|_F^2$ **then**
5 $\quad\quad$ **return** $\mathcal{A}$;
6 $\quad$ **else**
7 $\quad\quad$ $\mathcal{A} = \mathcal{A} \cup \{i^*\}$; $\mathbf{R} = \mathbf{R} - \mathbf{f}_i\mathbf{f}_i^{\top}$;
8 $\quad$ **end**
9 **end**
10 **return** $\mathcal{A}$;

Algorithm 2: SPFS-NES

Input: $\mathbf{X},\ \mathbf{Y},\ t\ ,\epsilon,\ \gamma_0,\ \mathbf{W}^{[0]}$
Output: $\mathbf{W}$

1 $\mathbf{W}^{[1]} = \mathbf{W}^{[0]},\ u_{-1} = 0,\ u_0 = 1,\ i = 1$;
2 **while** $\|\mathbf{W}^{[i]} - \mathbf{W}^{[i-1]}\| > \epsilon$ **do**
3 $\quad \alpha_i = (u_{i-2} - 1)/u_{i-1}$; $j = 0$;
4 $\quad \mathbf{W}^{[i]} = \mathbf{W}^{[i]} + \alpha_i\left(\mathbf{W}^{[i]} - \mathbf{W}^{[i-1]}\right)$;
5 $\quad$ **while** $f\left(\mathbf{W}^{[i+1]}\right) > f_{\gamma,\mathbf{S}_i}\left(\mathbf{W}^{[i+1]}\right)$ **do**
6 $\quad\quad \gamma = 2^j \times \gamma_{i-1}$;
7 $\quad\quad \mathbf{W}^{[i+1]} = \pi_t\left(\mathbf{W}^{[i]} - \frac{1}{\gamma}f'\left(\mathbf{W}^{[i]}\right)\right)$;
8 $\quad\quad j = j + 1$;
9 $\quad$ **end**
10 $\quad \gamma_i = \gamma$; $u_i = \left(1 + \sqrt{1 + 4u_{i-1}^2}\ \right)/2$;
11 $\quad i = i + 1$;
12 **end**
13 **return** $\mathbf{W}^{[i]}$;

A proper t value is crucial for SPFS-NES to achieve good feature selection performance. Cross-validation provides an effective way to determine the optimal t value. However, a problem associated with the cross-validation process is that it is very time-consuming. To address the problem, below we propose an efficient path algorithm by linking our formulation to the group-LAR [29], [46] algorithm, which leads to the SPFS-LAR algorithm for feature selection.

5.3 SPFS-LAR

We first convert the problem defined in Eq. (4.8) to a group lasso problem via data transformation

$$\mathbf{X}_L \in \mathbb{R}^{nk\times mk},\ \mathbf{y}_L \in \mathbb{R}^{nk\times 1},\ \mathbf{w}_L \in \mathbb{R}^{mk\times 1},$$

$$\mathbf{X}_L = \begin{bmatrix} \mathbf{f}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{f}_2 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{f}_1 & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{f}_2 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{f}_1 & \mathbf{0} & \mathbf{0} & \cdots \end{bmatrix},$$

$$\mathbf{y}_L = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \cdots \\ \mathbf{a}_k \end{bmatrix},\ \mathbf{w}_L = \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \cdots \\ \mathbf{w}_m^\top \end{bmatrix}, \tag{5.2}$$

where

$$\mathbf{X} = (\mathbf{f}_1, \cdots, \mathbf{f}_m),\ \mathbf{A}^* = (\mathbf{a}_1, \cdots, \mathbf{a}_k),\ \mathbf{W} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \cdots \\ \mathbf{w}_m \end{bmatrix}.$$

In Eq. (5.2), $\mathbf{X}_L = (\mathbf{X}_1, \ldots, \mathbf{X}_m)$, where $\mathbf{X}_i$ is a $k \times k$ block matrix, with the (j,l)-th block being $\mathbf{f}_i$ if $j = l$, and $\mathbf{0}$, otherwise. It is easy to check that with the data transformation defined above, the problem specified in Eq. (4.8) can be rewritten as

$$\min_{\mathbf{w}_L} \|\mathbf{y}_L - \mathbf{X}_L\mathbf{w}_L\|_2^2$$
$$s.t. \qquad \sum_{i=1}^{m} \|\mathbf{w}_i\|_2 \le t. \tag{5.3}$$

The problem defined in Eq. (5.3) is a group-lasso problem, and can be efficiently solved by the group-LAR algorithm [46]. The pseudo-code of group-LAR is shown in Algorithm 3. In Algorithm 3, $\mathcal{A}_i$ denotes the active set of the i-th run, which contains the features selected in this run. $\mathbf{X}_{\mathcal{A}_i}$ denotes the matrix comprised of the feature groups of $\mathbf{X}_L$ corresponding to $\mathcal{A}_i$; $\mathbf{X}_*$ denotes any feature group in $\mathcal{A}_i$; and $ext(\cdot)$ denotes the function that extend $\gamma_{\mathcal{A}_i}$ to an $m \times k$ dimensional vector by filling 0 in the positions corresponding to the feature groups that are not in $\mathcal{A}_i$. Algorithm 3 generates an approximate solution path for Eq. (5.3). It does not involve t, which is implicitly determined in group-LAR, when different numbers of features are selected. Our experimental study shows that, the SPFS-LAR can usually achieve the performance comparable to that of SPFS-NES. The advantage of SPFS-LAR is that it does not require parameter tuning.

Let m be the number of features, n the number of samples, k the number of selected features, l the number of columns in $\mathbf{Y}$. Assuming that $m \gg k$, we can show that for Algorithm 3, the computational cost of Line 2 is $O(mnc)$. The cost of Line 4 is $O(i(i+l)n)$, where i is iteration number, and in the i-th iteration, there are i features in the active set. The cost of Lines 5-7 is $O(mnl)$. And the cost of Lines 8-10 is $O(inl)$. Since Lines 4-10 need to be repeated for k times, the total computational cost of Lines 4-10 is $O(mnkl + nk^3)$, which overwhelms the cost of Line 2. Therefore, the total time complexity of SPFS-LAR is $O(mnkl + nk^3)$.

Algorithm 3: SPFS-LAR

Input: $\mathbf{X}_L = (\mathbf{X}_1, \ldots, \mathbf{X}_m),\ \mathbf{y}_L,\ k$
Output: $\mathbf{w}_L$

1 $\mathbf{w}_L^{[0]} = 0$, $i = 1$ and $\mathbf{r}^{[0]} = \mathbf{y}_L$;
2 compute the current "most correlated set": $\mathcal{A}_1 = \arg\max_j \|\mathbf{X}_j^\top \mathbf{r}^{[0]}\|_2^2$;
3 **while** $i \le k$ **do**
4 compute the current direction $\gamma_{\mathcal{A}_i}$: $\gamma_{\mathcal{A}_i} = \left(\mathbf{X}_{\mathcal{A}_i}^\top \mathbf{X}_{\mathcal{A}_i}\right)^{-1} \mathbf{X}_{\mathcal{A}_i}^\top \mathbf{r}^{[i-1]}$;
5 **for** $\forall j \notin \mathcal{A}_i$ **do**
6 compute how far (measured by α_j) we can progress in direction $\gamma_{\mathcal{A}_i}$, before $\mathbf{X}_j$ enters the most correlated set. $\|\mathbf{X}_j^\top \left(\mathbf{r}^{[i-1]} - \alpha_j \mathbf{X}_{\mathcal{A}_i}\gamma_{\mathcal{A}_i}\right)\|_2^2 = \|\mathbf{X}_*^\top \left(\mathbf{r}^{[i-1]} - \alpha_j \mathbf{X}_{\mathcal{A}_i}\gamma_{\mathcal{A}_i}\right)\|_2^2$;
7 **end**
8 $j^* = \arg\min_{j\notin\mathcal{A}_i} \alpha_j$; $\mathcal{A}_{i+1} = \mathcal{A}_i \bigcup \{j^*\}$;
9 $\mathbf{w}_L^{[i]} = \mathbf{w}_L^{[i-1]} + \alpha_{j^*} ext(\gamma_{\mathcal{A}_i})$;
10 $\mathbf{r}^{[i]} = \mathbf{y}_L - \mathbf{X}_L\mathbf{w}_L^{[i]}$; $i = i + 1$;
11 **end**
12 **return** $\mathbf{w}_L = \mathbf{w}_L^{[k]}$;

6 Experimental Study

We now empirically evaluate the performance of the three algorithms derived from the SPFS framework in both supervised and unsupervised learning context.

In the experiments, we choose nine representative feature selection algorithms for comparison purposes. And different algorithms are compared in both supervised and unsupervised learning context. For supervised learning context, six existing feature selection algorithms are chosen as baseline algorithms for comparison. They are ReliefF [34], Fisher Score [9], Trace-ratio [27], HSIC [35], mRMR [7] and AROM-SVM [41]. The first four algorithms are related to sample similarity preserving. The last two are the-state-of-the-art feature selection algorithms that can handle feature redundancy. mRMR is of filter model, which removes redundant features via considering pairwised feature correlation measured by mutual-information. AROM-SVM is of embedded model, which removes redundant features by iteratively reducing the weights of features which are less important for an SVM classifier. For unsupervised learning context, five representative algorithms are selected as baseline algorithms for comparison. They are Laplacian score [19], SPEC-1 and SPEC-3 [48], Trace-ratio [27], and HSIC [35]. These algorithms are all existing unsupervised methods related to sample similarity preserving. For SPFS-SFS, SPFS-NES and SPFS-LAR, in the supervised learning context, $\mathbf{K}$ is calculated by using Eq. (3.7); and in the unsupervised case, $\mathbf{K}$ is calculated by

using the RBF kernel function, $\mathbf{K}_{i,j} = exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\delta^2}\right)$. The adjustable parameter δ should be carefully tuned according to the problem at hand. If overestimated, the function will behave almost linearly, and lose its nonlinear power. On the other hand, if underestimated, it will lack regularization and the decision boundary will be highly sensitive to noise. In the experiment, we set

$$\delta^2 = \text{percentile}\Big(\{\parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2,\ i,j = 1\ldots,n\}, 20\Big),$$

and this usually results in good learning performance. All the algorithms are implemented in Matlab.

Eight high dimensional data sets are used in the experiment. There are four image data: AR10P[3], PIE10P[4], PIX10P[5], and ORL10P[6]. Two Microarray data: TOX and CLL-SUB from the GEO gene expression data depository[7] with retrieval ID GDS1454 and GDS968, respectively. And two text data: RELATHE (BASEBALL vs. HOCKEY) and PCMAC (PC vs. MAC) from the 20-newgroup data[8]. The two text data sets are preprocessed by the TMG package [47] with standard processes. Detailed information of the data sets is listed in Table 1.

TABLE 1
Summary of the benchmark data sets

Data Set	# Features	# Instances	# Classes
RELATHE	4322	1427	2
PCMAC	3289	1943	2
TOX	5748	171	4
CLL-SUB	11340	111	3
AR10P	2400	130	10
PIE10P	2400	210	10
PIX10P	10000	100	10
ORL10P	10000	100	10

Assume $\mathbf{F}$ is the set of selected features, and $\mathbf{X}_\mathbf{F}$ is the data only contains features in $\mathbf{F}$. In the supervised learning context, the algorithms are compared on (1) **classification accuracy** and (2) **redundancy rate**. The redundancy rate is measured in the following way:

$$\text{RED}(\mathbf{F}) = \frac{1}{m(m-1)} \sum_{f_i, f_j \in \mathbf{F}, i>j} \rho_{i,j}, \qquad (6.1)$$

where, $\rho_{i,j}$ returns the Pearson correlation between two features f_i and f_j. The measurement assesses the averaged correlation among all feature pairs, and a large value indicates that many selected features are strongly correlated and thus redundancy is expected to exist in $\mathbf{F}$. For unsupervised case, three measurements are used to compare the performance of the feature selection algorithms: (1) the **redundancy rate** defined in Eq. (6.1); (2) the **scale of the residue** calculated by $\|\mathbf{X_F X_F}^\top - \mathbf{K}\|_F^2$; and (3) the **Jaccard score** computed by

$$\text{JAC}(\mathbf{K_F}, \mathbf{K}, k) = \frac{1}{n} \sum_{i=1}^{n} \frac{NB(i,k,\mathbf{K_F}) \cap NB(i,k,\mathbf{K})}{NB(i,k,\mathbf{K_F}) \cup NB(i,k,\mathbf{K})},$$

where $\mathbf{K_F} = \mathbf{X_F X_F^\top}$; $\mathbf{K_F}$ and $\mathbf{K}$ are the similarity matrix computed from the selected features and the input similarity matrix, respectively; and $NB(i,k,\mathbf{K})$ returns the k nearest neighbors of the i-th instance according to the pairwised similarity specified by $\mathbf{K}$. The Jaccard score measures the averaged overlapping of the neighborhoods specified by $\mathbf{K_F}$ and $\mathbf{K}$. A high Jaccard score indicates that the pairwised similarities specified by the two similarity matrices are consistent. The last two measures are used to assess an algorithm's capability in sample similarity preserving in the continuous and the discrete ways, respectively.

For each data set, we randomly sample 50% instances as the training data and the remaining are used as test data. The process is repeated for 20 times and results in 20 different partitions of the data. Different algorithms are evaluated on each partition. The results achieved on each partition are recorded and averaged to obtain the final results. To calculate the classification accuracy, linear SVM is used. The parameters in feature selection algorithms as well as the SVM classifier are tuned via cross-validation on the training data. In the experiment, the paired *Student's t-test* is used to evaluate the statistical significance of the obtained results and the threshold for rejecting the null hypothesis is set to 0.05.

6.1 Study of Supervised Cases

Accuracy: The classification accuracy results are shown in Table 2. The plots of the accuracy rates achieved by algorithms when different number of features are selected can be found in the appendix. Table 2 shows the "aggregated accuracy" of different algorithm on each data set. The aggregated accuracy is obtained by averaging the averaged accuracy achieved by SVM using the top 10, 20, ..., 200 features selected by each algorithm. The value in the parentheses is the p-Val. The boldfaced values are the highest ones or the ones without significant difference to the highest.

In Table 2, we can observe that the three algorithms derived from the SPFS framework produce superior classification performance comparing to the baseline algorithms. According to the aggregated accuracy, SPFS-NES achieved the highest performance, which is followed by SPFS-LAR and SPFS-SFS. The results in the second last column of Table 2 show that all three SPFS based algorithms achieve the accuracy higher than 0.82, with an averaged value of 0.826. The averaged value achieved by the baseline algorithms is 0.767, which is 8% lower than that obtained by the SPFS based algorithms. According to the accuracy achieved on each data set, we

3. http://rvl1.ecn.purdue.edu/∼leix/aleix_face_DB.html. Images are subsampled down to the size of 60×40 = 2400 from 10 persons

4. http://peipa.essex.ac.uk/ipa/pix/faces/manchester/. Images are subsampled down to the size of 60×40 = 2400 from 10 persons

5. http://www.ri.cmu.edu/projects/project_418.html. Images are subsampled down to the size of 100×100 = 10000 from 10 persons

6. http://www.uk.research.att.com/facedatabase.html. Images are subsampled down to the size of 100×100 = 10000 from 10 persons

7. http://www.ncbi.nlm.nih.gov/geo/

8. http://people.csail.mit.edu/jrennie/20Newsgroups/.

TABLE 2
Study of supervised cases: **aggregated accuracy** with p-Val. (The higher the better.)

Algorithm	RELATHE	PCMAC	TOX	CLL-SUB	ORL10P	PIX10P	AR10P	PEI10P	AVE	WIN
ReliefF	0.68 (0.00)	0.70 (0.00)	0.77 (0.00)	0.67 (0.00)	0.83 (0.00)	0.93 (0.00)	0.80 (0.02)	0.94 (0.00)	0.789	0
Fisher Score	0.73 (0.00)	**0.75 (0.28)**	0.72 (0.00)	0.54 (0.00)	0.80 (0.00)	0.92 (0.00)	0.77 (0.00)	0.93 (0.00)	0.769	1
Trace-ratio	0.73 (0.00)	**0.75 (0.28)**	0.72 (0.00)	0.54 (0.00)	0.80 (0.00)	0.92 (0.00)	0.77 (0.00)	0.93 (0.00)	0.769	1
HSIC	0.73 (0.00)	**0.75 (0.37)**	0.73 (0.00)	0.55 (0.00)	0.80 (0.00)	0.93 (0.00)	0.77 (0.00)	0.94 (0.00)	0.774	1
mRMR	**0.75 (0.58)**	**0.75 (0.31)**	0.70 (0.00)	0.64 (0.00)	0.73 (0.00)	0.87 (0.00)	0.70 (0.00)	0.95 (0.00)	0.762	2
AROM-SVM	0.73 (0.00)	0.74 (0.00)	0.64 (0.00)	0.57 (0.00)	0.78 (0.00)	0.86 (0.00)	0.63 (0.00)	0.93 (0.03)	0.739	0
SPSF-SFS	0.74 (0.03)	**0.76 (1.00)**	0.74 (0.00)	**0.69 (0.30)**	**0.90 (1.00)**	**0.96 (1.00)**	**0.83 (0.40)**	0.95 (0.00)	0.821	5
SPSF-NES	**0.75 (0.16)**	**0.75 (0.16)**	**0.79 (1.00)**	**0.71 (1.00)**	**0.89 (0.34)**	**0.94 (0.16)**	**0.84 (1.00)**	**0.97 (1.00)**	**0.830**	**8**
SPSF-LAR	**0.75 (1.00)**	**0.75 (0.93)**	0.76 (0.00)	0.70 (0.01)	**0.90 (0.40)**	**0.95 (0.20)**	**0.82 (0.41)**	0.96 (0.02)	0.827	5

TABLE 3
Study of supervised cases: **averaged redundancy rate** with p-Val. (The lower the better.)

Algorithm	RELATHE	PCMAC	TOX	CLL-SUB	ORL10P	PIX10P	AR10P	PEI10P	AVE	WIN
ReliefF	0.06 (0.00)	0.06 (0.00)	0.34 (0.00)	0.59 (0.00)	0.92 (0.00)	0.79 (0.00)	0.77 (0.00)	0.36 (0.00)	0.487	0
Fisher Score	0.07 (0.00)	0.07 (0.00)	0.56 (0.00)	0.76 (0.00)	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.516	0
Trace-ratio	0.07 (0.00)	0.07 (0.00)	0.56 (0.00)	0.76 (0.00)	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.516	0
HSIC	0.07 (0.00)	0.07 (0.00)	0.56 (0.00)	0.76 (0.00)	0.79 (0.00)	0.83 (0.00)	0.67 (0.00)	0.37 (0.00)	0.515	0
mRMR	**0.04 (0.28)**	**0.03 (1.00)**	0.26 (0.00)	0.26 (0.00)	0.25 (0.03)	0.33 (0.00)	0.26 (0.00)	0.29 (0.00)	0.214	2
AROM-SVM	0.05 (0.00)	0.04 (0.00)	**0.15 (1.00)**	0.59 (0.00)	**0.25 (0.21)**	0.26 (0.01)	0.25 (0.00)	0.32 (0.00)	0.241	2
SPSF-SFS	0.05 (0.00)	0.04 (0.00)	0.30 (0.00)	0.44 (0.00)	0.26 (0.00)	0.26 (0.00)	0.25 (0.00)	**0.24 (1.00)**	0.230	1
SPSF-NES	**0.04 (1.00)**	0.03 (0.00)	**0.16 (0.07)**	**0.22 (1.00)**	**0.24 (1.00)**	**0.25 (1.00)**	**0.22 (1.00)**	0.28 (0.00)	**0.179**	**6**
SPSF-LAR	**0.04 (0.21)**	0.03 (0.00)	0.22 (0.00)	0.24 (0.01)	0.35 (0.00)	0.41 (0.00)	0.27 (0.00)	0.26 (0.00)	0.227	1

can also see that the SPFS based algorithms achieve good performance on all three types of data (image, microarray and text). In contrast, the baseline algorithms can only give good performance on one or two types of data, suggesting that the criterion used in the SPFS framework provides more stable feature selection performance.

Redundancy rate: Table 3 presents the averaged redundancy rates of the top n features selected by different algorithms, where n is the instance number of the data. We choose n, since when the number of selected features is larger than n, any feature can be expressed by a linear combination of the remaining ones, which will introduce unnecessary redundancy in the evaluation stage. In the table, the boldfaced values are the lowest redundancy rates or the ones without significant difference to the lowest. The results from the redundancy rates show that SPFS-SFS, SPFS-LAR, and SPFS-NES all attain low redundancy rates, which suggests that the redundancy removal mechanism in SPFS is effective. We also observe that the two baseline algorithms mRMR and AROM-SVM also produce low redundancy rates. Since the two algorithms are able to remove redundant features, the observation is consistent with our expectation.

The results from accuracy and redundancy rates together indicate that among all the algorithms related to sample similarity preserving, SPFS based algorithms select features containing the least redundancy, and result in the highest accuracy, which suggests the necessity of removing redundant features for improving learning performance. It is observed that the two baseline algorithms mRMR and AROM-SVM do not perform as well as the SPFS based ones in terms of classification accuracy. We conjecture that the reasons are: (1) In mRMR, feature's contribution to classification is considered individually by evaluating the correlation between each feature and the class label. However, the class label may be jointly determined by a set of features, and this interaction among features is not considered by mRMR. (2) In AROM-SVM, it is very hard to find a suitable regularization parameter C for the SVM used in the embedded model, since AROM-SVM iteratively modifies the data via feature reweighting [41].

6.2 Study of Unsupervised Cases

Residue scale & Jaccard score: Tables 4 and 5 present the averaged residue scale and Jaccard score achieved by different algorithms on the benchmark data sets. Again, top n features are selected. The two measures assess algorithms' capability in similarity preserving. The results show that comparing to the baseline algorithms, the three SPFS based algorithms achieve better performance on all eight data sets, which demonstrates their strong capability in similarity preserving.

Among the three SPFS based algorithms, SPFS-SFS achieves the best performance, followed by SPFS-NES and SPFS-LAR. We observe that the performance of SPFS-LAR is inferior to that of SPFS-SFS. The reason is: SPFS-LAR optimizes $\|\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top - \mathbf{K}\|_F^2$, while in residue scale and Jaccard score, $\mathbf{X_F}\mathbf{X_F}^\top$ is used to compute $\mathbf{K_F}$. It is possible that the performance of SPFS-LAR is underestimated by the two measures, since SPFS-LAR may select features, whose linear combination can produce a similarity matrix that well preserves the similarity specified by $\mathbf{K}$. To clearly show this, Table 6 lists the aggregated residue scale and Jaccard score of SPFS-SFS and SPFS-LAR, when $\mathbf{X}\mathbf{W}\mathbf{W}^\top\mathbf{X}^\top$ is used to compute $\mathbf{K_F}$ in the two measures. Here the aggregated residue scale and Jaccard score are obtained by averaging the averaged residue scale and Jaccard score over eight benchmark data sets. The results show that when the weight matrix $\mathbf{W}$ is taken into account, SPFS-LAR achieves better results than SPFS-SFS. This indicates

TABLE 6
The aggregated residue scale and Jaccard score of SPFS-SFS and SPFS-LAR when the weight matrix $\mathbf{W}$ of SPFS-LAR is considered in calculating the measures.

Algorithm	Residue	JAC k_{NB} = 1	JAC k_{NB} = 5
SPFS-SFS	94.15	0.38	0.35
SPFS-LAR-W	**91.45**	**0.63**	**0.53**

that the features selected by SPFS-LAR also have strong capability in similarity preserving through their linear combinations. This fact is also verified by the good classification performance achieved by the SVM classifier on the features selected by SPFS-LAR.

Redundancy rate: Table 7 shows the averaged redundancy rates achieved with the top n features selected by different algorithms on the benchmark data sets. The results show that the features selected by the three SPFS based algorithms contain much less redundancy comparing with the baseline algorithms. This is expected, since the latter cannot remove redundant features in feature selection. We observe that the performance of SPFS-LAR is inferior to that of SPFS-SFS in terms of redundancy rate, which is not the case in the supervised learning. We conjecture that this is related to the structure of the similarity matrix $\mathbf{K}$. In the supervised case, $\mathbf{K}$ is a block matrix. Its structure is much simpler than that of the $\mathbf{K}$ in the unsupervised case, which is calculated with the RBF kernel function. The SPFS-LAR only produces a solution which approximates that obtained by SPFS-NES. When the problem becomes harder, the SPFS-LAR may not be able to give a sufficiently good solution and this can result in some redundant features to be selected. We will conduct a closer study on this issue in our future work.

The experimental results from both supervised and unsupervised learning cases show that the three algorithms derived from the SPFS framework can select features containing less redundancy and producing excellent learning performance. Each of the three algorithms has its own advantages in terms of computational complexity and selection performance. With cross-validation[9], SPFS-NES performs robustly in both supervised and unsupervised learning context. Especially in the supervised case, it achieves both the highest accuracy and the lowest redundancy rates. However, the cross-validation process can be very time-consuming. In contrast to the SPFS-NES, SPFS-SFS and SPFS-LAR do not need parameter tuning, therefore they are more efficient. And at the same time, they provide feature selection performance that is comparable to that of SPFS-NES. When k is smaller than n, where k is the number of the selected features and n is the number of instances, SPFS-LAR is more efficient than SPFS-SFS. While in the unsupervised cases, SPFS-SFS may select features containing less redundancy.

9. In the unsupervised case, the regularization parameter of SPFS-NES is tuned by cross-validation to minimize the residue scale.

7 Conclusions

In this work, we study the problem of feature selection from the perspective of sample similarity preserving. We explicitly propose the concept of "Similarity Preserving Feature Selection" and develop the SPFS framework in a direct and rigorous way. We show, through theoretical analysis, the connections between the proposed framework and the existing feature selection algorithms that can be related to similarity preserving. The proposed SPFS framework improves the existing algorithms by overcoming their common drawback in handling feature redundancy. This is important in both supervised and unsupervised learning. As illustrated by the extensive experimental study, the proposed SPFS framework achieves superior performance in various learning context, which demonstrates its efficacy.

Given a high dimensional space, many approaches have been proposed to find a low dimensional space, where the geometric structure of the data is preserved according to certain criterion. These methods in general fall into the category of dimensionality reduction via feature extraction, instead of feature selection. The representative algorithms in this category include: Multidimensional Scaling (MDS) [6], ISOMAP [37], Locally Linear Embedding (LLE) [30], Laplacian Eigenmaps [4], Semidefinite Embedding (SDE) [40], Neighborhood Preserving Embedding (NPE) [20] and Structure Preserving Embedding (SPE) [33], to name a few. The difference between feature extraction and feature selection is that to reduce dimensionality, feature extraction generates a small set of new features by combining the original features, while feature selection selects a small set of the original features. By keeping the original features, feature selection improves the interpretability of learning models, which is preferred in many real applications, such as text mining and genetic analysis. Many frameworks have been proposed to unify the aforementioned *feature extraction* methods in various ways [5], [32], [36], [42], [44]. Comparing with these works, our work connects existing *feature selection* algorithms with the concept of "sample similarity preserving" and overcomes their common drawback on handling redundant features in feature selection.

The SPFS framework forms our initial study for similarity preserving feature selection. Our future work includes: (1) the current SPFS formulation is based on comparing a linear kernel constructed from the selected features, $\hat{\mathbf{X}}^\top\hat{\mathbf{X}}$, to a given similarity matrix $\mathbf{K}$. This strategy, on one hand, makes the formulation tractable. On the other hand, it may restrict model complexity and reduce the estimation precision. We will try to extend the current formulation by allowing to use general nonlinear kernels. (2) When both labeled and unlabeled data are presented, Semi-supervised learning technique can be utilized to boost the learning performance. We will study how to extend the current SPFS formulation to the semi-supervised learning context. (3) In our study, we found

TABLE 4
Study of supervised cases: **averaged residue scale** with p-Val. (The lower the better.)

Algorithm	RELATHE	PCMAC	TOX	CLL-SUB	ORL10P	PIX10P	AR10P	PEI10P	AVE	WIN
Laplacian Score	219.19 (0.00)	224.14 (0.00)	90.78 (0.00)	61.99 (0.00)	56.92 (0.00)	54.30 (0.00)	67.83 (0.00)	106.13 (0.00)	110.16	0
SPEC-1	219.37 (0.00)	224.55 (0.00)	90.88 (0.00)	62.03 (0.00)	57.20 (0.00)	54.37 (0.00)	67.26 (0.00)	108.47 (0.00)	110.52	0
SPEC-3	240.42 (0.00)	232.01 (0.00)	87.63 (0.00)	60.73 (0.00)	54.53 (0.00)	54.05 (0.00)	65.74 (0.00)	101.64 (0.00)	112.09	0
Trace-Ratio	200.57 (0.00)	203.75 (0.00)	90.74 (0.00)	60.27 (0.00)	57.21 (0.00)	54.37 (0.00)	67.26 (0.00)	106.69 (0.00)	105.11	0
HSIC	199.63 (0.00)	199.61 (0.00)	90.68 (0.00)	59.41 (0.00)	57.07 (0.00)	54.11 (0.00)	67.03 (0.00)	103.56 (0.00)	103.89	0
SPFS-SFS	**196.43 (1.00)**	**197.01 (1.00)**	**81.03 (1.00)**	**51.40 (1.00)**	**45.02 (1.00)**	**41.68 (1.00)**	**54.31 (1.00)**	**86.32 (1.00)**	**94.15**	**8**
SPFS-NES	196.59 (0.00)	197.18 (0.00)	81.32 (0.00)	51.85 (0.00)	45.58 (0.00)	42.63 (0.00)	54.35 (0.00)	86.44 (0.00)	94.49	0
SPFS-LAR	196.71 (0.00)	197.30 (0.00)	87.15 (0.00)	56.82 (0.00)	48.70 (0.00)	43.39 (0.00)	56.14 (0.00)	88.53 (0.00)	96.84	0

TABLE 5
Study of unsupervised cases: **averaged Jaccard score** with p-Val. (The higher the better.)

$k_{NB} = 1$

Algorithm	RELATHE	PCMAC	TOX	CLL-SUB	ORL10P	PIX10P	AR10P	PEI10P	AVE	WIN
Laplacian Score	0.00 (0.00)	0.00 (0.00)	0.10 (0.00)	0.06 (0.00)	0.07 (0.00)	0.05 (0.00)	0.07 (0.00)	0.04 (0.00)	0.05	0
SPEC-1	0.00 (0.00)	0.00 (0.00)	0.11 (0.00)	0.07 (0.00)	0.06 (0.00)	0.05 (0.00)	0.07 (0.00)	0.04 (0.00)	0.05	0
SPEC-3	0.03 (0.00)	0.02 (0.00)	0.12 (0.00)	0.05 (0.00)	0.15 (0.00)	0.05 (0.00)	0.09 (0.00)	0.05 (0.00)	0.07	0
Trace-Ratio	0.05 (0.00)	0.02 (0.00)	0.12 (0.00)	0.08 (0.00)	0.06 (0.00)	0.05 (0.00)	0.08 (0.00)	0.03 (0.00)	0.06	0
HSIC	0.07 (0.00)	0.04 (0.00)	0.12 (0.00)	0.10 (0.00)	0.08 (0.00)	0.05 (0.00)	0.07 (0.00)	0.04 (0.00)	0.07	0
SPFS-SFS	**0.09 (1.00)**	**0.05 (0.54)**	**0.52 (1.00)**	**0.25 (0.40)**	**0.69 (1.00)**	**0.60 (1.00)**	0.49 (0.01)	**0.32 (0.11)**	**0.378**	**7**
SPFS-NES	0.08 (0.00)	0.04 (0.00)	**0.51 (0.70)**	**0.26 (1.00)**	0.58 (0.00)	0.48 (0.00)	**0.52 (1.00)**	**0.34 (1.00)**	0.352	4
SPFS-LAR	**0.09 (0.18)**	**0.05 (1.00)**	0.16 (0.00)	0.11 (0.00)	0.30 (0.00)	0.43 (0.00)	0.28 (0.00)	0.18 (0.00)	0.201	2

$k_{NB} = 5$

Algorithm	RELATHE	PCMAC	TOX	CLL-SUB	ORL	PIX	AR	PIE	AVE	WIN
Laplacian Score	0.01 (0.00)	0.02 (0.00)	0.17 (0.00)	0.16 (0.00)	0.16 (0.00)	0.11 (0.00)	0.13 (0.00)	0.08 (0.00)	0.10	0
SPEC-1	0.01 (0.00)	0.01 (0.00)	0.17 (0.00)	0.15 (0.00)	0.15 (0.00)	0.11 (0.00)	0.14 (0.00)	0.08 (0.00)	0.10	0
SPEC-3	0.02 (0.00)	0.01 (0.00)	0.19 (0.00)	0.14 (0.00)	0.28 (0.00)	0.11 (0.00)	0.16 (0.00)	0.11 (0.00)	0.13	0
Trace-Ratio	0.04 (0.00)	0.02 (0.00)	0.18 (0.00)	0.17 (0.00)	0.15 (0.00)	0.11 (0.00)	0.14 (0.00)	0.08 (0.00)	0.11	0
HSIC	0.05 (0.00)	0.02 (0.00)	0.18 (0.00)	0.16 (0.00)	0.16 (0.00)	0.13 (0.00)	0.14 (0.00)	0.10 (0.00)	0.12	0
SPFS-SFS	**0.05 (0.48)**	0.02 (0.00)	**0.42 (0.22)**	**0.28 (0.59)**	**0.63 (1.00)**	**0.66 (1.00)**	**0.43 (0.77)**	0.32 (0.00)	**0.353**	**6**
SPFS-NES	**0.05 (0.06)**	0.02 (0.00)	**0.43 (1.00)**	**0.29 (1.00)**	0.57 (0.00)	0.58 (0.00)	**0.43 (1.00)**	**0.34 (1.00)**	0.338	5
SPFS-LAR	**0.06 (1.00)**	**0.03 (1.00)**	0.22 (0.00)	0.19 (0.00)	0.38 (0.00)	0.54 (0.00)	0.33 (0.00)	0.20 (0.00)	0.244	2

TABLE 7
Study of unsupervised cases: **averaged redundancy rate** with p-Val. (The lower the better.)

Algorithm	RELATHE	PCMAC	TOX	CLL-SUB	ORL10P	PIX10P	AR10P	PEI10P	AVE	WIN
Laplacian Score	0.27 (0.00)	0.33 (0.00)	0.57 (0.00)	0.65 (0.00)	0.88 (0.00)	0.97 (0.00)	0.82 (0.00)	0.84 (0.00)	0.67	0
SPEC-1	0.27 (0.00)	0.34 (0.00)	0.57 (0.00)	0.67 (0.00)	0.88 (0.00)	0.97 (0.00)	0.81 (0.00)	0.87 (0.00)	0.67	0
SPEC-3	0.66 (0.00)	0.51 (0.00)	0.47 (0.00)	0.59 (0.00)	0.72 (0.00)	0.97 (0.00)	0.75 (0.00)	0.77 (0.00)	0.68	0
Trace-Ratio	0.19 (0.00)	0.19 (0.00)	0.57 (0.00)	0.67 (0.00)	0.88 (0.00)	0.97 (0.00)	0.81 (0.00)	0.87 (0.00)	0.65	0
HSIC	0.17 (0.00)	0.12 (0.00)	0.57 (0.00)	0.64 (0.00)	0.88 (0.00)	0.97 (0.00)	0.80 (0.00)	0.82 (0.00)	0.62	0
SPFS-SFS	**0.07 (1.00)**	**0.05 (1.00)**	**0.16 (1.00)**	**0.15 (1.00)**	**0.27 (1.00)**	**0.34 (1.00)**	0.28 (0.00)	0.38 (0.00)	**0.21**	**6**
SPFS-NES	**0.07 (0.33)**	0.06 (0.00)	0.18 (0.00)	0.19 (0.00)	**0.28 (0.62)**	**0.36 (0.15)**	**0.26 (1.00)**	**0.36 (1.00)**	0.22	4
SPFS-LAR	0.08 (0.00)	0.06 (0.00)	0.46 (0.00)	0.55 (0.00)	0.57 (0.00)	0.44 (0.00)	0.43 (0.00)	0.50 (0.00)	0.39	0

that the SPFS formulation can be linked to a wide range of learning models, such as PCA, LDA and SVM through their least square formulations. We will reveal these connections to gain more insights on similarity preserving feature selection for developing more novel feature selection algorithms.

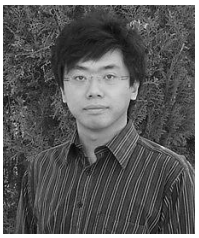
ACKNOWLEDGMENTS

This work is, in part, supported by NSF Grant (0812551).

REFERENCES

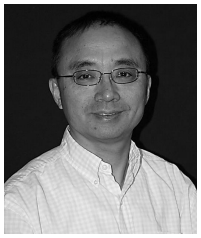
[1] D. W. Aha. *Feature Weighting for Lazy Learning Algorithms*, pages 13–32. 1998.
[2] A. Appice, M. Ceci, S. Rawles, and P. Flach. Redundant feature elimination for multi-class problems. In *Proceedings of ICML*, 2004.
[3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.
[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Advances in Neural Information Processing Systems, 15*, 2003.
[5] D. Cai, X. He, and J. Han. Spectral regression: A unified approach for sparse subspace learning. In *Proceedings of ICDM*, 2007.
[6] T. Cox and etal. *Multidimensional Scaling*. Chapman& Hall, 2001.
[7] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of CSB*, 2003.
[8] R. Duangsoithong. Relevant and redundant feature analysis with ensemble classification. In *Proceedings of ICAPR*, 2009.
[9] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2 edition, 2001.
[10] J. G. Dy and etal. Unsupervised feature selection applied to content-based retrieval of lung images. *Transactions on Pattern Analysis and Machine Intelligence*, 25(3):373–378, 2003.
[11] J.G. Dy and C.E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, 2004.
[12] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–49, 2004.
[13] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.
[14] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.
[15] A. Gretton, O. Bousquet, A. Smola, and B. Scholkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Proceedings of ALT*, 2005.

[16] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
[17] M. Hall. *Correlation Based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, Computer Science, 1999.
[18] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
[19] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems 18*, 2005.
[20] X. He, D. Cai, S. Yan, and H.J. Zhang. Neighborhood preserving embedding. In *Proceedings of ICCV*, 2005.
[21] K. Kira and L.A. Rendell. A practical approach to feature selection. In *Proceedings of ICML*, 1992.
[22] I. Kononenko. Estimating attributes: Analysis and extension of RELIEF. In *Proceedings of ECML*, 1994.
[23] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers, 1998.
[24] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient l2,1-norm minimization. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.
[25] A. Nemirovski. Efficient methods in convex programming. Lecture Notes, 1994.
[26] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2003.
[27] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan. Trace ratio criterion for feature selection. In *Proceedings of Conference on Artificial Intelligence (AAAI)*, 2008.
[28] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
[29] V. Roth and B. Fischer. The group-lasso for generalized linear models: Uniqueness of solutions and efficient algorithms. In *Proceedings of ICML*, 2008.
[30] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
[31] Y. Saeys and etal. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.
[32] L.K. Saul, K.Q. Weinberger, F. Sha, J. Ham, and D.D. Lee. *Spectral Methods for Dimensionality Reduction*, chapter 16, pages 279–293. The MIT Press, 2006.
[33] B. Shaw and T. Jebara. Structure preserving embedding. In *Proceedings of ICML*, 2009.
[34] M.R. Sikonja and I. Kononenko. Theoretical and empirical analysis of Relief and ReliefF. *Machine Learning*, 53:23–69, 2003.
[35] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 2007.
[36] L. Sun, S. Ji, and J. Ye. A least squares formulation for a class of generalized eigenvalue problems in machine learning. In *Proceedings of ICML*, 2009.
[37] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
[38] U. von Luxburg. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, 2007.
[39] R. Walpole, R. Myers. *Probability and Statistics for Engineers and Scinentists*. Macmillan Publishing Company, 1993.
[40] K. Q. Weinberger, B. D. Packer, and L. K. Saul. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *In Proceedings of ISTATS*, 2005.
[41] J. Weston, A. Elisseff, B. Schoelkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.
[42] L. Xiao, J. Sun, and S. Boyd. A duality view of spectral methods for dimensionality reduction. In *Proceedings of ICML*, 2006.
[43] Z. Xu, R. Jin, J. Ye, M.R. Lyu, and I. King. Discriminative semi-supervised feature selection via manifold regularization. In *Proceedings of IJCAI*, 2009.
[44] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 29 (1):40–51, 2007.
[45] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the ICML*, 2003.
[46] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, 68:49–67, 2005.
[47] D. Zeimpekis and E. Gallopoulos. Tmg: A matlab toolbox for generating term-document matrices from text collections. Technical report, University of Patras, Greece, 2005.
[48] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the ICML*, 2007.
[49] Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. In *Proceedings of the SDM*, 2007.
[50] L. Zhou, L. Wang, and C. Shen. Feature selection with redundancy -constrained class separability. *IEEE Transactions on Neural Networks*, 21:853–858, 2010.



Zheng Zhao is a research statistician of the SAS institute, Inc. He obtained his Ph.D. in Computer Science and Engineering from Arizona State University (ASU), and his M.Eng. and B.Eng. in Computer Science and Engineering from Harbin Institute of Technology (HIT). His research interests are in high-performance data mining. In recent years, he focuses on designing and developing novel analytic approaches for handling large-scale data of extremely high dimensionality. He has published more than 20 research papers in the top conferences and journals. He has served as a reviewer for over 10 journals and conferences. He waw also a co-chair for the PAKDD Workshop on Feature Selection in Data Mining 2010.

Lei Wang received the B.Eng degree and the M.Eng degree from Southeast University, China in 1996 and 1999, respectively, and the PhD degree from Nanyang Technological University, Singapore in 2004. He worked as Research Associate and Research Fellow at Nanyang Technological University from 2003 to 2005. After that, he joined the Australian National University and worked as Research Fellow and then Fellow from 2005 to 2011. In April 2011, he joined Faculty of Informatics University of Wollongong as Senior Lecturer. He was awarded the Australian Post-doctoral Fellowship by Australian Research Council in 2007 and the Early Career Researcher Award by Australian Academy of Science in 2009, respectively. His research interests include machine learning, pattern recognition, and computer vision.



Huan Liu is a professor of Computer Science and Engineering at Arizona State University. He obtained his Ph.D. in Computer Science from University of Southern California and his B.Eng. in Computer Science and Electrical Engineering from Shanghai Jiaotong University. He was recognized for excellence in teaching and research in Computer Science and Engineering at Arizona State University. His research interests are in data mining, machine learning, social computing, and artificial intelligence, investigating problems that arise in many real-world applications with high-dimensional data of disparate forms such as social media, group interaction and modeling, data preprocessing (feature selection), and text/web mining. His well-cited publications include books, book chapters, and encyclopedia entries as well as conference and journal papers. He serves on journal editorial boards and numerous conference program committees, and is a founding organizer of the International Conference Series on Social Computing, Behavioral-Cultural Modeling, and Prediction.

Jieping Ye is an Associate Professor of Computer Science and Engineering at Arizona State University. He received his Ph.D. in Computer Science from University of Minnesota, Twin Cities in 2005. His research interests include machine learning, data mining, and biomedical informatics. He won the outstanding student paper award at ICML in 2004, the SCI Young Investigator of the Year Award at ASU in 2007, the SCI Researcher of the Year Award at ASU in 2009, the NSF CAREER Award in 2010, and the KDD best research paper award honorable mention in 2010.