A Variance Minimization Criterion to Feature Selection Using Laplacian Regularization

Xiaofei He, *Senior Member*, *IEEE*, Ming Ji, Chiyuan Zhang, and Hujun Bao

Abstract—In many information processing tasks, one is often confronted with very high-dimensional data. Feature selection techniques are designed to find the meaningful feature subset of the original features which can facilitate clustering, classification, and retrieval. In this paper, we consider the feature selection problem in unsupervised learning scenarios, which is particularly difficult due to the absence of class labels that would guide the search for relevant information. Based on Laplacian regularized least squares, which finds a smooth function on the data manifold and minimizes the empirical loss, we propose two novel feature selection algorithms which aim to minimize the expected prediction error of the regularized regression model. Specifically, we select those features such that the size of the parameter covariance matrix of the regularized regression model is minimized. Motivated from experimental design, we use trace and determinant operators to measure the size of the covariance matrix. Efficient computational schemes are also introduced to solve the corresponding optimization problems. Extensive experimental results over various real-life data sets have demonstrated the superiority of the proposed algorithms.

Index Terms—Feature selection, dimensionality reduction, manifold, regularization, regression, clustering.

1 Introduction

In many applications in computer vision, pattern recognition, and data mining, the objects (e.g., images and texts) are usually represented as points in high dimensional euclidean space. High dimensionality significantly increases the time and space requirements for processing the data. Moreover, learning tasks, such as classification, clustering, and retrieval, that are analytically or computationally manageable in low dimensional spaces may become completely intractable in spaces of several hundreds or thousands dimensions [10], [16]. To overcome this problem, feature selection [6], [17], [31], [32], [35] and extraction [20], [21], [26], [27] techniques are designed to reduce the dimensionality by finding a meaningful feature subset or feature combinations. Feature selection and extraction techniques can be applied for data preprocessing and facilitate other learning tasks such as classification, clustering, and retrieval.

Feature selection methods can be classified into supervised and unsupervised methods. Supervised feature selection methods usually evaluate the feature importance by the correlation between feature and class label. The typical supervised feature selection methods include Pearson correlation coefficients, Fisher score, Kolmogorov-Smirnov test, ReliefF [23], Lasso [16], and SVM-RFE [14]. However, in practice, there is usually no shortage of unlabeled data but labels are expensive. Hence, it is of great significance to develop unsupervised feature selection algorithms which can make use of all of the data points. In this paper, we consider the problem of selecting features in unsupervised learning scenarios, which is a much harder problem due to the absence of class labels that would guide the search for relevant information.

Unsupervised feature selection algorithms can be roughly classified into two categories. The first category of approaches aims to maximize some clustering performance. Wolf and Shashua proposed a feature selection algorithm called $Q - \alpha$ [32]. The algorithm optimizes over a least-squares criterion function which measures the clusterability of the input data points projected onto the selected coordinates. The optimal coordinates are those for which the cluster coherence, measured by the spectral gap of the corresponding affinity matrix, is maximized [32]. A remarkable property of the algorithm is that it always yields sparse solutions. Some other approaches in this category include sequential unsupervised feature selection [9], wrapper approach based on expectation maximization (EM) [11], and the maximum entropy-based method [2]. The second category of approaches selects the most representative features which can best preserve the geometrical structure of the data space. Data variance is perhaps the simplest criterion for selecting representative features. This criterion essentially projects the data points along the dimensions of maximum variances. Note that the Principal Component Analysis (PCA) algorithm shares the same principle of maximizing variance, but it involves feature transformation and obtains a set of transformed features rather than a subset of the original features. Recently, the Laplacian score algorithm [17] and its extensions [34] have been proposed to select those features which can best reflect the underlying manifold structure.

In this paper, we propose a novel variance minimization criterion to feature selection. The central idea of our

- *X. He, C. Zhang, and H. Bao are with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, 388 Yu Hang Tang Rd., Hangzhou, Zhejiang 310058, China.*
E-mail: {xiaofeihe, bao}@cad.zju.edu.cn, zhangchiyuan@zjucadcg.cn.
- *M. Ji is with the Department of Computer Science, University of Illinois at Urbana Champaign, Siebel Center, 201 N. Goodwin Ave., Urbana, IL 61801. E-mail: mingji1@illinois.edu.*

Manuscript received 15 Dec. 2009; revised 21 Nov. 2010; accepted 15 Dec. 2010; published online 1 Mar. 2011.
Recommended for acceptance by M. Meila.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-12-0826.
Digital Object Identifier no. 10.1109/TPAMI.2011.44.

0162-8828/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

approaches is to explore both discriminative and geometrical information in the data. In other words, our goal is to find a feature subset which respects the underlying manifold structure and can improve the learning performance the most if the selected features are used. Suppose we have a regression task at hand after feature selection. We want the predicted response as robust as possible, and the parameter estimates have small variance. Particularly, our approaches are based on Laplacian regularized least squares (LapRLS, [4]), a regularized regression model which explicitly takes into account the underlying manifold structure in the data. It is interesting to note that the covariance matrix of the parameters is not dependent on the class label, but only on the random examples. Thus, we propose to select those features so that the size of the resulting covariance matrix, as well as the expected prediction error, is minimized. This way, our feature selection approaches explicitly take into account both discriminative and geometrical structures in the data. In statistics, there are many different optimality criteria to measure the size of the covariance matrix, leading to different algorithms. We adopt the most popular, A-optimality and D-optimality, and introduce two novel feature selection algorithms, called *Laplacian regularized A-Optimal Feature Selection* (LapAOFS) and *Laplacian regularized D-Optimal Feature Selection* (LapDOFS), respectively. In A-optimality, the trace of the parameter covariance matrix, that is, the total variance of the parameter estimates, is minimized. In D-optimality, the determinant of the parameter covariance matrix is minimized. D-optimal feature selection is motivated by reference to the ellipsoidal confidence regions for the parameters of the regression model [1]. We also introduce efficient computation schemes to solve these two optimization problems.

The organization of the paper is as follows: In the next section, we provide a brief description of the related work. In Section 3, we introduce the experimental design perspective for feature selection. We then present the proposed LapAOFS algorithm and the optimization scheme in Section 4. Section 5 presents the LapDOFS algorithm and its optimization scheme. The extensive experimental results on three real-life data sets are presented in Section 6. Finally, we provide some concluding remarks and suggestions for future work in Section 7.

2 Related Work

The work most related to our proposed approaches is Laplacian Score [17]. On the other hand, our approaches are motivated from optimal experimental design [1] in statistics. Therefore, in this section, we provide a brief description of Laplacian Score and optimal experimental design.

2.1 Laplacian Score

Laplacian score is a recently proposed unsupervised feature selection algorithm [17]. The basic idea of Laplacian score is to evaluate the feature according to its locality preserving power, or its consistency with the manifold structure.

Suppose we have m data points, $\mathbf{x}_1, \dots, \mathbf{x}_m$. Let L_r denote the Laplacian score of the rth feature. Let f_{ri} denote the ith sample of the rth feature. Define

$$\mathbf{f}_r = \left(f_{r1}, \dots, f_{rm}\right)^T.$$

In order to approximate the manifold structure, one can construct a nearest neighbor graph with weight matrix W [17]. The importance of a feature can be thought of as the degree to which it respects this graph structure. To be specific, a "good" feature should be the one on which two data points are close to each other if and only if there is an edge between these two points. A reasonable criterion for choosing a good feature is to minimize the following objective function:

$$L_r = \frac{\sum_{ij} \left(f_{ri} - f_{rj}\right)^2 W_{ij}}{Var(\mathbf{f}_r)}. \tag{1}$$

There are many choices of the weight matrix W. Let $\mathcal{N}(\mathbf{x}_i)$ denote the set of k nearest neighbors of $\mathbf{x}_i$. The simplest definition of W is as follows:

$$W_{ij} = \begin{cases} 1, & \mathbf{x}_i \in \mathcal{N}(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let D be a diagonal matrix, $D_{ii} = \sum_j W_{ij}$. The *weighted* data variance can be computed as follows:

$$Var(\mathbf{f}_r) = \widetilde{\mathbf{f}}_r^T D \widetilde{\mathbf{f}}_r,$$

where

$$\widetilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathbf{f}_r^T D\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}\mathbf{1}.$$

Define $L = D - W$, which is usually called the Laplacian matrix. It is easy to see that [17]

$$\sum_{ij} \left(f_{ri} - f_{rj}\right)^2 W_{ij} = 2\mathbf{f}_r^T L \mathbf{f}_r = 2\widetilde{\mathbf{f}}_r^T L \widetilde{\mathbf{f}}_r.$$

Finally, the Laplacian score of the rth feature is reduced to

$$L_r = \frac{\widetilde{\mathbf{f}}_r^T L \widetilde{\mathbf{f}}_r}{\widetilde{\mathbf{f}}_r^T D \widetilde{\mathbf{f}}_r}. \tag{3}$$

Both Laplacian score and our proposed algorithms explicitly make use of the manifold structure. However, unlike Laplacian score, which is based on a graph model, our proposed algorithms are motivated from the question that how the selected features can improve the performance of a manifold regularized regression model. Our proposed algorithms and framework of analysis provide a new perspective from experimental design [1] for feature selection.

2.2 Optimal Experimental Design

In statistics, the problem of selecting samples to label is typically referred to as experimental design [1]. The sample $\mathbf{x}$ is referred to as experiment, and its label is referred to as measurement. The variances of the parameter estimates and predictions depend on the particular experimental design used and should be as small as possible. Poorly designed experiments waste resources by yielding unnecessarily large variances and imprecise predictions [1].

We consider a linear regression model

$$y = \mathbf{w}^T \mathbf{x} + \epsilon,$$

where $\mathbf{w}$ is the weight vector and ϵ is an unknown error with zero mean. Different observations have errors that are independent, but with equal variances σ^2. We define $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ to be the learner's output given input $\mathbf{x}$ and the weight vector $\mathbf{w}$. Suppose we have a set of measured (labeled) samples, $(\mathbf{z}_1, y_1), \ldots, (\mathbf{z}_k, y_k)$. The maximum likelihood estimate for the weight vector, $\widehat{\mathbf{w}}$, is that which minimizes the sum squared error:

$$J_{sse}(\mathbf{w}) = \sum_{i=1}^{k} \left(\mathbf{w}^T\mathbf{z}_i - y_i\right)^2. \tag{4}$$

We define $\mathbf{y} = (y_1, \ldots, y_k)^T$ and $Z = (\mathbf{z}_1, \ldots, \mathbf{z}_k)$. The optimal solution is given by

$$\widehat{\mathbf{w}} = (ZZ^T)^{-1}Z\mathbf{y}.$$

It is easy to check that the mean and covariance matrix of the parameters have the following expression [1]:

$$E(\widehat{\mathbf{w}} - \mathbf{w}) = 0,$$

and

$$Cov(\widehat{\mathbf{w}}) = \sigma^2(ZZ^T)^{-1}.$$

Thus, the predicted value of the response given by $\widehat{y} = \widehat{\mathbf{w}}^T\mathbf{x}$ has the variance [1]

$$var(\widehat{y}) = \sigma^2\mathbf{x}^T(ZZ^T)^{-1}\mathbf{x}.$$

In order to minimize the variances of the parameter estimates and the predicted response, different optimality criteria have been proposed, out of which A and D-optimality have received the most attentions. In A-optimality, the trace of the parameter covariance matrix is minimized, equivalent to minimizing the average variance. In D-optimality, the determinant of the parameter covariance matrix is minimized. D-optimality was motivated by reference to the ellipsoidal confidence regions for the parameters of the linear model. A D-optimal design minimizes the content of this confidence region and so minimizes the volume of the ellipsoid [1].

Experimental design techniques have conventionally been used to select the most informative samples. In our work, we take a different perspective to apply experimental design techniques to select the most informative features.

3 A Variance Minimization Criterion to Feature Selection

3.1 The Problem

Let $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ be an $n \times m$ data matrix. We denote the row vectors of X by $\mathbf{f}_i^T \in \mathbb{R}^m$ $(i = 1, \ldots, n)$, each corresponding to a feature, and define the feature set $\mathcal{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$. The problem of feature selection is to find the most informative feature subset $\mathcal{S} = \{\mathbf{g}_1, \ldots, \mathbf{g}_k\} \subset \mathcal{F}$. Let $X^{\mathcal{S}}$ be a new data matrix by only keeping those features in $\mathcal{S}$ and $\mathbf{x}_i^{\mathcal{S}}$ be the ith column vector of $X^{\mathcal{S}}$.

3.2 The Objective Function

We consider the linear regression model by using the selected feature subset $\mathcal{S}$:

$$y = \mathbf{w}^T\mathbf{x}^{\mathcal{S}} + \epsilon, \tag{5}$$

where ϵ is an unknown error with zero mean. Different observations have errors that are independent, but with equal variances σ^2. Suppose y_i is the label of the ith data point. Thus, the maximum likelihood estimate for the weight vector, $\widehat{\mathbf{w}}$, is given by minimizing the sum squared error in (4).

However, the ordinary linear regression fails to take into account the intrinsic geometrical structure of the data. In this work, we consider the case where the probability distribution that generates the data is supported on a submanifold of the ambient space [3], [19], [28], [24]. Let W be a similarity matrix capturing the neighborhood structure in the data and D be a diagonal matrix, $D_{ii} = \sum_j W_{ij}$, and $L = D - W$. The matrix L is called *graph Laplacian* [8]. Belkin et al. have applied graph Laplacian as a smoothing operator to ensure the learned function varies smoothly along the geodesics of the data manifold, and hence the learning performance can be improved [4]. By incorporating the Laplacian regularizer into the sum squared error, they proposed Laplacian regularized least squares (LapRLS, [4]):

$$\begin{aligned} J_{LapRLS}(\mathbf{w}) = &\sum_{i=1}^{m} \left(\mathbf{w}^T\mathbf{x}_i^{\mathcal{S}} - y_i\right)^2 \\ &+ \frac{\lambda_1}{2}\sum_{i,j=1}^{m} \left(\mathbf{w}^T\mathbf{x}_i^{\mathcal{S}} - \mathbf{w}^T\mathbf{x}_j^{\mathcal{S}}\right)^2 W_{ij} \\ &+ \lambda_2\|\mathbf{w}\|^2, \end{aligned} \tag{6}$$

where W is a weight matrix defined in (2). The solution to the minimization problem (6) is given as follows:

$$\widehat{\mathbf{w}} = (X^{\mathcal{S}}(X^{\mathcal{S}})^T + \lambda_1 X^{\mathcal{S}}L(X^{\mathcal{S}})^T + \lambda_2 I)^{-1}X^{\mathcal{S}}\mathbf{y}, \tag{7}$$

where I is a $k \times k$ identity matrix. Define

$$H = X^{\mathcal{S}}(X^{\mathcal{S}})^T + \lambda_1 X^{\mathcal{S}}L(X^{\mathcal{S}})^T + \lambda_2 I.$$

Thus, $\widehat{\mathbf{w}} = H^{-1}X^{\mathcal{S}}\mathbf{y}$. Let $\Lambda = \lambda_1 X^{\mathcal{S}}L(X^{\mathcal{S}})^T + \lambda_2 I$. Noticing that $\mathbf{y} = (X^{\mathcal{S}})^T\mathbf{w} + \epsilon$, the bias can be computed as follows [18]:

$$\begin{aligned} &E(\widehat{\mathbf{w}} - \mathbf{w}) \\ &\quad = H^{-1}X^{\mathcal{S}}(X^{\mathcal{S}})^T\mathbf{w} - \mathbf{w} \\ &\quad = H^{-1}(X^{\mathcal{S}}(X^{\mathcal{S}})^T + \Lambda - \Lambda)\mathbf{w} - \mathbf{w} \\ &\quad = (I - H^{-1}\Lambda)\mathbf{w} - \mathbf{w} \\ &\quad = -H^{-1}\Lambda\mathbf{w}. \end{aligned} \tag{8}$$

By noticing that $Cov(\mathbf{y}) = \sigma^2 I$, the covariance matrix of the parameter $\widehat{\mathbf{w}}$ can be computed as follows [18]:

$$\begin{aligned} &Cov(\widehat{\mathbf{w}}) \\ &\quad = Cov(H^{-1}X^{\mathcal{S}}\mathbf{y}) \\ &\quad = H^{-1}X^{\mathcal{S}}Cov(\mathbf{y})(X^{\mathcal{S}})^T H^{-1} \\ &\quad = \sigma^2 H^{-1}X^{\mathcal{S}}(X^{\mathcal{S}})^T H^{-1} \\ &\quad = \sigma^2 H^{-1}\left(H - \Lambda\right)H^{-1} \\ &\quad = \sigma^2\left(H^{-1} - H^{-1}\Lambda H^{-1}\right). \end{aligned}$$

For any data point $\mathbf{x}^{\mathcal{S}}$, let $\widehat{y} = \widehat{\mathbf{w}}^T\mathbf{x}^{\mathcal{S}}$ be its predicted observation. The expected squared prediction error is

$$\begin{aligned}
&E(y-\widehat{y})^2\\
&= E(y-\widehat{\mathbf{w}}^T\mathbf{x}^{\mathcal{S}})^2\\
&= \sigma^2 + (\mathbf{x}^{\mathcal{S}})^T(E(\mathbf{w}-\widehat{\mathbf{w}})(\mathbf{w}-\widehat{\mathbf{w}})^T)\mathbf{x}^{\mathcal{S}}\\
&= \sigma^2 + (\mathbf{x}^{\mathcal{S}})^T(E(\mathbf{w}-\widehat{\mathbf{w}})E(\mathbf{w}-\widehat{\mathbf{w}})^T + Cov(\mathbf{w}-\widehat{\mathbf{w}}))\mathbf{x}^{\mathcal{S}}\\
&= \sigma^2 + (\mathbf{x}^{\mathcal{S}})^T(H^{-1}\Lambda\mathbf{w}\mathbf{w}^T\Lambda H^{-1} + \sigma^2H^{-1} - \sigma^2H^{-1}\Lambda H^{-1})\mathbf{x}^{\mathcal{S}}.
\end{aligned} \tag{9}$$

Since λ_1 and λ_2 are usually set to be very small, we have [18]

$$Cov(\widehat{\mathbf{w}}) \approx \sigma^2H^{-1}, \tag{10}$$

and

$$E(y-\widehat{y})^2 \approx \sigma^2 + \sigma^2(\mathbf{x}^{\mathcal{S}})^TH^{-1}\mathbf{x}^{\mathcal{S}}. \tag{11}$$

Inspired from experimental design principles [1], we propose selecting those features such that the size of the parameter covariance matrix is minimized. By minimizing the size of H^{-1}, the expected squared prediction error for a new point can also be minimized. Using different measures of the size of the covariance matrix, the optimal features can be obtained by solving the following optimization problems:

Laplacian Regularized A-Optimal Feature Selection:

$$\min_{\mathcal{S}\subset\mathcal{F}} \mathrm{Tr}\big(H^{-1}\big). \tag{12}$$

Laplacian Regularized D-Optimal Feature Selection:

$$\min_{\mathcal{S}\subset\mathcal{F}} \det\big(H^{-1}\big). \tag{13}$$

4 Laplacian Regularized A-Optimal Feature Section

In this section, we discuss how to solve the optimization problem (12) of LapAOFS. The matrix H can be rewritten as follows:

$$H = X^{\mathcal{S}}(I+\lambda_1L)(X^{\mathcal{S}})^T + \lambda_2I.$$

Since L is positive semidefinite (PSD), $I+\lambda_1L$ is positive definite and invertible. By using the Woodbury formula [13], we have

$$H^{-1} = \frac{1}{\lambda_2}I - \frac{1}{\lambda_2^2}X^{\mathcal{S}}\left((I+\lambda_1L)^{-1} + \frac{1}{\lambda_2}(X^{\mathcal{S}})^TX^{\mathcal{S}}\right)^{-1}(X^{\mathcal{S}})^T.$$

We define

$$M = \lambda_2(I+\lambda_1L)^{-1}.$$

Thus,

$$H^{-1} = \frac{1}{\lambda_2}I - \frac{1}{\lambda_2}X^{\mathcal{S}}(M+(X^{\mathcal{S}})^TX^{\mathcal{S}})^{-1}(X^{\mathcal{S}})^T.$$

Noticing that $\mathrm{Tr}(AB)=\mathrm{Tr}(BA)$, we have

$$\begin{aligned}
&\mathrm{Tr}(H^{-1})\\
&= \frac{k}{\lambda_2} - \frac{1}{\lambda_2}\mathrm{Tr}(X^{\mathcal{S}}\big(M+(X^{\mathcal{S}})^TX^{\mathcal{S}}\big)^{-1}(X^{\mathcal{S}})^T)\\
&= \frac{k}{\lambda_2} - \frac{1}{\lambda_2}\mathrm{Tr}((M+(X^{\mathcal{S}})^TX^{\mathcal{S}})^{-1}(X^{\mathcal{S}})^TX^{\mathcal{S}})\\
&= \frac{k}{\lambda_2} - \frac{1}{\lambda_2}\mathrm{Tr}((M+(X^{\mathcal{S}})^TX^{\mathcal{S}})^{-1}\\
&\quad (M+(X^{\mathcal{S}})^TX^{\mathcal{S}} - M))\\
&= \frac{k}{\lambda_2} - \frac{1}{\lambda_2}\mathrm{Tr}(I - \big(M+(X^{\mathcal{S}})^TX^{\mathcal{S}}\big)^{-1}M)\\
&= \frac{k-m}{\lambda_2} + \frac{1}{\lambda_2}\mathrm{Tr}((M+(X^{\mathcal{S}})^TX^{\mathcal{S}})^{-1}M).
\end{aligned} \tag{14}$$

Thus, the optimal solution of LapAOFS can be obtained by minimizing

$$\min_{\mathcal{S}\subset\mathcal{F}} \mathrm{Tr}((M+(X^{\mathcal{S}})^TX^{\mathcal{S}})^{-1}M). \tag{15}$$

As defined above, the matrix $X^{\mathcal{S}}$ only contains the selected features, i.e., $\mathbf{g}_1,\dots,\mathbf{g}_k$. Therefore, we can rewrite $(X^{\mathcal{S}})^TX^{\mathcal{S}}$ as follows:

$$(X^{\mathcal{S}})^TX^{\mathcal{S}} = \sum_{i=1}^{k}\mathbf{g}_i\mathbf{g}_i^T.$$

To simplify the optimization problem (15), we introduce n indicator variables, $\alpha_1,\dots,\alpha_n$. $\alpha_i=1$ if the ith feature is selected and 0 otherwise. Thus, we have

$$(X^{\mathcal{S}})^TX^{\mathcal{S}} = \sum_{i=1}^{n}\alpha_i\mathbf{f}_i\mathbf{f}_i^T = X^T\begin{pmatrix}\alpha_1 & 0 & 0 & 0\\ 0 & \alpha_2 & 0 & 0\\ 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & \alpha_n\end{pmatrix}X.$$

Let $\boldsymbol{\alpha}^T=(\alpha_1,\dots,\alpha_n)$. The optimization problem (15) can be rewritten as follows:

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \mathrm{Tr}\left(\left(M + X^T\begin{pmatrix}\alpha_1 & 0 & 0 & 0\\ 0 & \alpha_2 & 0 & 0\\ 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & \alpha_n\end{pmatrix}X\right)^{-1}M\right),\\
&s.t. \sum_{i=1}^{n}\alpha_i = k, \alpha_i\in\{0,1\}.
\end{aligned} \tag{16}$$

4.1 Sequential Optimization

In the following, we describe a sequential optimization scheme to select the most informative features. Suppose k (≥ 0) features have been selected, i.e., $\mathbf{g}_1,\dots,\mathbf{g}_k\in\mathcal{F}$. The $(k+1)$th feature can be selected by solving the following problem:

$$\mathbf{g}_{k+1} = \arg\min_{\mathbf{g}} \mathrm{Tr}\left(\left(M + \sum_{i=1}^{k}\mathbf{g}_i\mathbf{g}_i^T + \mathbf{g}\mathbf{g}^T\right)^{-1}M\right). \tag{17}$$

Define

$$A_k = M + \sum_{i=1}^{k} \mathbf{g}_i \mathbf{g}_i^T,$$

and

$$A_0 = M.$$

The optimization problem (17) reduces to finding

$$\mathbf{g}_{k+1} = \arg\min_{\mathbf{g}} \mathrm{Tr}\Big(\big(A_k + \mathbf{g}\mathbf{g}^T\big)^{-1} M\Big). \tag{18}$$

By using the Sherman-Morrison formula [13], we have

$$\big(A_k + \mathbf{g}\mathbf{g}^T\big)^{-1} = A_k^{-1} - \frac{A_k^{-1}\mathbf{g}\mathbf{g}^T A_k^{-1}}{1 + \mathbf{g}^T A_k^{-1}\mathbf{g}}. \tag{19}$$

Thus,

$$\begin{aligned} &\mathrm{Tr}\Big(\big(A_k + \mathbf{g}\mathbf{g}^T\big)^{-1} M\Big) \\ &= \mathrm{Tr}(A_k^{-1}M) - \frac{\mathrm{Tr}\Big(A_k^{-1}\mathbf{g}\mathbf{g}^T A_k^{-1} M\Big)}{1 + \mathbf{g}^T A_k^{-1}\mathbf{g}} \\ &= \mathrm{Tr}(A_k^{-1}M) - \frac{\mathrm{Tr}\Big(\mathbf{g}^T A_k^{-1} M A_k^{-1}\mathbf{g}\Big)}{1 + \mathbf{g}^T A_k^{-1}\mathbf{g}} \\ &= \mathrm{Tr}(A_k^{-1}M) - \frac{\mathbf{g}^T A_k^{-1} M A_k^{-1}\mathbf{g}}{1 + \mathbf{g}^T A_k^{-1}\mathbf{g}}. \end{aligned}$$

Since $\mathrm{Tr}(A_k^{-1}M)$ is a constant when selecting the $(k+1)$th feature, the $(k+1)$th optimal feature is given by

$$\mathbf{g}_{k+1} = \arg\max_{\mathbf{g}} \frac{\mathbf{g}^T A_k^{-1} M A_k^{-1}\mathbf{g}}{1 + \mathbf{g}^T A_k^{-1}\mathbf{g}}. \tag{20}$$

Once $\mathbf{g}_{k+1}$ is obtained, A_{k+1}^{-1} can be updated according to (19).

4.2 Convex Optimization

In this section, we introduce another optimization scheme for solving (16). Due to the combinatorial nature, the optimization problem (16) is NP-hard. In order to solve it efficiently, we relax the integer constraints on α_is and allow them to take real nonnegative values. Moreover, for feature selection, it is desired that α_is can be sufficiently sparse. In other words, there is only a subset of α_is whose values are positive and any other α_is are zero. This way, we can simply select those features whose corresponding α_is are positive. The sparseness of $\boldsymbol{\alpha}$ can be controlled through minimizing the ℓ_1-norm of $\boldsymbol{\alpha}$ [16], i.e., $\|\boldsymbol{\alpha}\|_1$. Since α_is are nonnegative, $\|\boldsymbol{\alpha}\|_1 = \mathbf{1}^T\boldsymbol{\alpha}$, where $\mathbf{1}$ is a column vector of all ones. Finally, the optimization problem becomes

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \mathrm{Tr}\left(\left(M + \sum_{i=1}^{n} \alpha_i \mathbf{f}_i \mathbf{f}_i^T\right)^{-1} M\right) + \gamma \mathbf{1}^T\boldsymbol{\alpha}, \\ s.t. \quad & \boldsymbol{\alpha} \succeq 0, \end{aligned} \tag{21}$$

where γ is a regularization parameter which controls the sparseness of $\boldsymbol{\alpha}$. The following theorem shows that the optimization problem (21) is convex.

Theorem 4.1. *The optimization problem (21) is convex with variable $\boldsymbol{\alpha} \in \mathbb{R}^n$.*

Proof. Since the matrix M is symmetric and positive definite, we can decompose it as follows:

$$M = UU^T, \quad U = (\mathbf{u}_1, \ldots, \mathbf{u}_m).$$

Thus, we have

$$\begin{aligned} &\mathrm{Tr}\left(\left(M + \sum_{i=1}^{n} \alpha_i \mathbf{f}_i \mathbf{f}_i^T\right)^{-1} M\right) \\ &= \mathrm{Tr}\left(U^T\left(M + \sum_{i=1}^{n} \alpha_i \mathbf{f}_i \mathbf{f}_i^T\right)^{-1} U\right). \end{aligned}$$

We define $f(A) = \mathrm{Tr}(U^T A^{-1} U) = \sum_{i=1}^{m} \mathbf{u}_i^T A^{-1}\mathbf{u}_i$. We know that the matrix fractional function $\mathbf{u}_i^T A^{-1}\mathbf{u}_i$ is a convex function of A [7]. Therefore, $f(A)$ is also convex. We define

$$g(\boldsymbol{\alpha}) = M + \sum_{i=1}^{n} \alpha_i \mathbf{f}\mathbf{f}^T.$$

Since $g(\boldsymbol{\alpha})$ is an affine function, the composition $f \circ g$ is convex [7]. Clearly, $\gamma\mathbf{1}^T\boldsymbol{\alpha}$ is a convex function of $\boldsymbol{\alpha}$. Therefore, the function $f \circ g + \gamma\mathbf{1}^T\boldsymbol{\alpha}$ is convex. It is easy to see that the constraint function $(-\boldsymbol{\alpha})$ is also convex. Thus, the optimization problem (21) is convex with variable $\boldsymbol{\alpha} \in \mathbb{R}^n$. □

By introducing a new variable $P \in \mathbb{R}^{m\times m}$, the optimization problem can be equivalently rewritten as follows:

$$\begin{aligned} \min_{P,\boldsymbol{\alpha}} \quad & \mathrm{Tr}(P) + \gamma\mathbf{1}^T\boldsymbol{\alpha}, \\ s.t. \quad & P \succeq_{\mathbb{S}_m^+} U^T\left(M + \sum_{i=1}^{n} \alpha_i \mathbf{f}_i \mathbf{f}_i^T\right)^{-1} U, \\ & \boldsymbol{\alpha} \succeq 0, \end{aligned} \tag{22}$$

where $\mathbb{S}_m^+$ denotes the set of symmetric positive semidefinite $m \times m$ matrices and $A \succeq_{\mathbb{S}_m^+} B$ denotes that $A - B$ is symmetric and positive semidefinite.

In the following, we discuss how to use Schur complement theorem [7] to cast the optimization problem (22) as a SemiDefinite Programming (SDP). Suppose A, B, and C are, respectively, $p \times p$, $p \times q$, and $q \times q$ matrices, and A is invertible. Let

$$Q = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}.$$

The Schur complement of the block A of the matrix Q is the $p \times p$ matrix $C - B^T A^{-1} B$. The Schur complement theorem states that Q is positive semidefinite if and only if $C - B^T A^{-1} B$ is positive semidefinite [7]. By using the Schur complement theorem, the optimization problem (22) can be expressed as

$$\begin{aligned} \min_{P,\boldsymbol{\alpha}} \quad & \mathrm{Tr}(P) + \gamma\mathbf{1}^T\boldsymbol{\alpha}, \\ s.t. \quad & \begin{pmatrix} M + \sum_{i=1}^{n} \alpha_i \mathbf{f}_i \mathbf{f}_i^T & U \\ U^T & P \end{pmatrix} \succeq_{\mathbb{S}_{2m}^+} 0, \\ & \boldsymbol{\alpha} \succeq 0. \end{aligned} \tag{23}$$

The above optimization problem can be solved by using interior-point methods [7].

5 Laplacian Regularized D-Optimal Feature Selection

Since $\det(H^{-1}) = (\det(H))^{-1}$, minimizing $\det(H^{-1})$ is equivalent to maximizing $\det(H)$. By using matrix determinant lemma [15], we have

$$\begin{aligned}
&\det(H)\\
&= \det(X^{\mathcal{S}}(I+\lambda_1 L)(X^{\mathcal{S}})^T + \lambda_2 I)\\
&= \det(\lambda_2 I)\det\big(I+\lambda_1 L\big)\\
&\quad \times \det\left((I+\lambda_1 L)^{-1} + \frac{1}{\lambda_2}(X^{\mathcal{S}})^T X^{\mathcal{S}}\right)\\
&= (\lambda_2)^k \det(I+\lambda_1 L)(\lambda_2)^{-m}\\
&\quad \times \det(\lambda_2 (I+\lambda_1 L)^{-1} + (X^{\mathcal{S}})^T X^{\mathcal{S}}).
\end{aligned}$$

Let $c = (\lambda_2)^{k-m}\det(I+\lambda_1 L)$, which is a constant, and $M = \lambda_2(I+\lambda_1 L)^{-1}$. Notice that $(X^{\mathcal{S}})^T X^{\mathcal{S}} = \sum_{i=1}^{k} \mathbf{g}_i\mathbf{g}_i^T$; thus

$$\det(H) = c \cdot \det\left(M + \sum_{i=1}^{k}\mathbf{g}_i\mathbf{g}_i^T\right).$$

The optimization problem (13) reduces to

$$\max_{\{\mathbf{g}_1,\ldots,\mathbf{g}_k\}\subset\mathcal{F}} \det\left(M + \sum_{i=1}^{k}\mathbf{g}_i\mathbf{g}_i^T\right). \tag{24}$$

5.1 Sequential Optimization

In the following, we describe a sequential optimization scheme to solve the above problem. Suppose k features have been selected, i.e., $\mathbf{g}_1,\ldots,\mathbf{g}_k$, $k \geq 0$. We define

$$A_k = M + \sum_{i=1}^{k}\mathbf{g}_i\mathbf{g}_i^T, \tag{25}$$

and

$$A_0 = M. \tag{26}$$

Thus, the $(k+1)$th optimal feature is given by

$$\mathbf{g}_{k+1} = \arg\max_{\mathbf{g}\in\mathcal{F}} \det(A_k + \mathbf{g}\mathbf{g}^T). \tag{27}$$

Again, using the matrix determinant lemma [15], we have

$$\det(A_k + \mathbf{g}\mathbf{g}^T) = (1 + \mathbf{g}^T A_k^{-1}\mathbf{g})\det(A_k). \tag{28}$$

Notice that $\det(A_k)$ is a constant when selecting the $(k+1)$th optimal feature. Therefore,

$$\mathbf{g}_{k+1} = \arg\max_{\mathbf{g}\in\mathcal{F}} \mathbf{g}^T A_k^{-1}\mathbf{g}. \tag{29}$$

Once $\mathbf{g}_{k+1}$ is obtained, $\det(A_{k+1})$ can be obtained according to (28). A_{k+1}^{-1} can be updated according to (19).

5.2 Concave Optimization

Similarly to LapAOFS, the optimization problem (13) can also be relaxed to a concave optimization problem as

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \log\det\left(M + \sum_{i=1}^{n}\alpha_i\mathbf{f}_i\mathbf{f}_i^T\right) - \gamma\mathbf{1}^T\boldsymbol{\alpha},\\
s.t. \quad & \boldsymbol{\alpha} \succeq 0.
\end{aligned} \tag{30}$$

Since $M + \sum_{i=1}^{n}\alpha_i\mathbf{f}_i\mathbf{f}_i^T$ is an affine function of $\boldsymbol{\alpha}$ and $\log\det$ is a concave function, their composition $\log\mathrm{Det}(M + \sum_{i=1}^{n}\alpha_i\mathbf{f}_i\mathbf{f}_i^T)$ is a concave function of $\boldsymbol{\alpha}$. Thus, the objective function $\log\mathrm{Det}(M + \sum_{i=1}^{n}\alpha_i\mathbf{f}_i\mathbf{f}_i^T) - \gamma\mathbf{1}^T\boldsymbol{\alpha}$ is concave. The constraint function is linear. Therefore, the optimization problem (30) is a concave optimization problem. The optimization problem (30) is typically referred to the as determinant maximization problem and can be solved by interior-point methods [30]. Please see [30] for details.

6 Computational Complexity Analysis

In this section, we discuss the computational cost of our proposed algorithms.

6.1 LapAOFS

In Section 4, we have described two optimization schemes to solve the objective function of LapAOFS. For the sequential optimization scheme, each iteration consists of the following two steps:

1. solving the problem (20), and
2. updating A_{k+1}^{-1} according to (19).

The evaluation of $\mathbf{g}^T A_k^{-1} M A_k^{-1}\mathbf{g}/(1 + \mathbf{g}^T A_k^{-1}\mathbf{g})$ needs $O(m^2)$ operations. In order to solve problem (20), we need to calculate this equation for all of the n features. So, the complexity of the first step is $O(nm^2)$. The second step applies the Sherman-Morrison formula to computing A_{k+1}^{-1}, and it is easy to check that the complexity is $O(m^2)$. Thus, the cost per iteration is $O(nm^2)$. Suppose our goal is to select k features, then the total cost of the sequential scheme is $O(knm^2)$.

For the convex optimization scheme, the state-of-the-art SDP solvers [25], [29] typically use interior-point methods to solve the SDP problem. It has been proven that the worst-case complexity of the interior point solvers for SD's depends quadratically on the number of variables and polynomially with an exponent of roughly 2.5 on the size of the Positive Semidefinite matrix [5]. In problem (23), the number of variables is $(n + m^2)$ and the size of the PSD matrix is $2m$, so the worst-case complexity is $O((n + m^2)^2 m^{2.5})$.

6.2 LapDOFS

The sequential optimization method for solving LapDOFS is similar to that for LapAOFS. At each iteration, we need to solve the problem (29) and update A_{k+1}^{-1}. The complexity of solving the problem (29) is the same as that of solving the problem (20), i.e., $O(nm^2)$. So, the total cost of the sequential method is still $O(knm^2)$.

For the concave optimization, in the worst case, solving the determinant maximization problem needs $O(\sqrt{n})$ Newton iterations and each iteration requires $O((n^2 + m^2)n^2)$ operations [30]. So, the total cost of the concave relaxation is $O((n^2 + m^2)n^{2.5})$.

7 Experimental Results

In this section, several experiments are carried out to show the effectiveness of our proposed LapAOFS and LapDOFS methods for feature selection. We perform clustering and nearest neighbor classification experiments by only using the selected features. The following five unsupervised feature selection algorithms are compared:

- Our proposed LapAOFS algorithm.
- Our proposed LapDOFS algorithm.
- Laplacian Score [17].
- $Q-\alpha$ algorithm [32].
- Data Variance.

The Variance method selects those features of maximum variances in order to obtain the best expressive power. Laplacian score aims to preserve the local manifold structure. The $Q-\alpha$ algorithm aims to maximize the cluster coherence. In our LapAOFS and LapDOFS algorithms, the regularization parameters λ_1 and λ_2 are both set to 0.01 and the number of nearest neighbors (k) is set to 4. We have presented both sequential and convex (concave) optimization schemes for LapAOFS and LapDOFS. However, convex (concave) optimization is very time consuming. Therefore, we adopt sequential optimization schemes in our experiments. In the following, we begin with a description of the data preparation.

7.1 Data Preparation

Three real world data sets were used in our experiments. The first one is the MNIST handwritten digit database,[1] which has a training set of 60,000 images (denoted as set A) and a testing set of 10,000 images (denoted as set B). In our experiments, we take the first 1,000 images from set A and the first 1,000 images from set B as our data set. Each class (digit) contains around 200 images. Each digit image is of size 28×28 and therefore represented by a 784-dimensional vector.

The second one is the COIL20 image library[2] from Columbia. It contains 20 objects. The images of each objects were taken 5 degrees apart as the object is rotated on a turntable and each objects has 72 images. The size of each image is 32×32 pixels, with 256 gray levels per pixel. Thus, each image is represented by a 1,024-dimensional vector.

The third one is the AT&T face database[3] which consists of a total of 400 face images, of a total of 40 subjects (10 samples per subject). The images were captured at different times and have different variations including expressions (open or closed eyes, smiling or nonsmiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. The original images were normalized (in scale and orientation) such that the two eyes were aligned at the same position. Then, the facial areas were cropped into the final images for matching. The size of each cropped image is 32×32 pixels, with 256 gray levels per pixel. Thus, each face image can be represented by a 1,024-dimensional vector.

7.2 Data Clustering

We perform K-means clustering by using the selected features and compare the results of different algorithms in this test.

7.2.1 Evaluation Metric

The clustering algorithm generates a cluster label for each data point. The clustering performance is evaluated by comparing the generated class label and the ground truth. In our experiments, the accuracy (AC) and the normalized mutual information metric (NMI) are used to measure the clustering performance [33]. Given a point $\mathbf{x}_i$, let r_i and s_i be the obtained cluster label and the label provided by the ground truth, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{m} \delta(s_i, map(r_i))}{m},$$

where m is the total number of samples and $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise, and $\mathrm{map}(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data set. The best mapping can be found by using the Kuhn-Munkres algorithm [22].

Let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)},$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a sample point arbitrarily selected from the data set belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected data point belongs to the clusters c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information as follows:

$$NMI(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))},$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $NMI(C, C')$ ranges from 0 to 1. $NMI = 1$ if the two sets of clusters are identical, and $NMI = 0$ if the two sets are independent.

7.2.2 Clustering Results

For each data set, the evaluations were conducted by using different numbers of clusters (k). For MNIST, $k = 3, 5, 7, 9$. For each given cluster number k $(= 3, 5, 7)$, 20 test runs were conducted on different randomly chosen clusters, and the final performance scores were computed by averaging the scores from the 20 tests. For $k = 9$, there are only 10 possible combinations by removing each one of the 10 digit classes. In this case, we averaged the scores from the 10 tests. For each test, we applied different algorithms to select ℓ features and applied K-means for clustering. The K-means was applied 10 times with different start points and the best result in terms of the objective function of K-means was recorded. Fig. 1 shows the plots of clustering performance, in terms of accuracy and normalized mutual information, versus the number of selected features (ℓ). For

[1] http://yann.lecun.com/exdb/mnist/.

[2] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[3] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

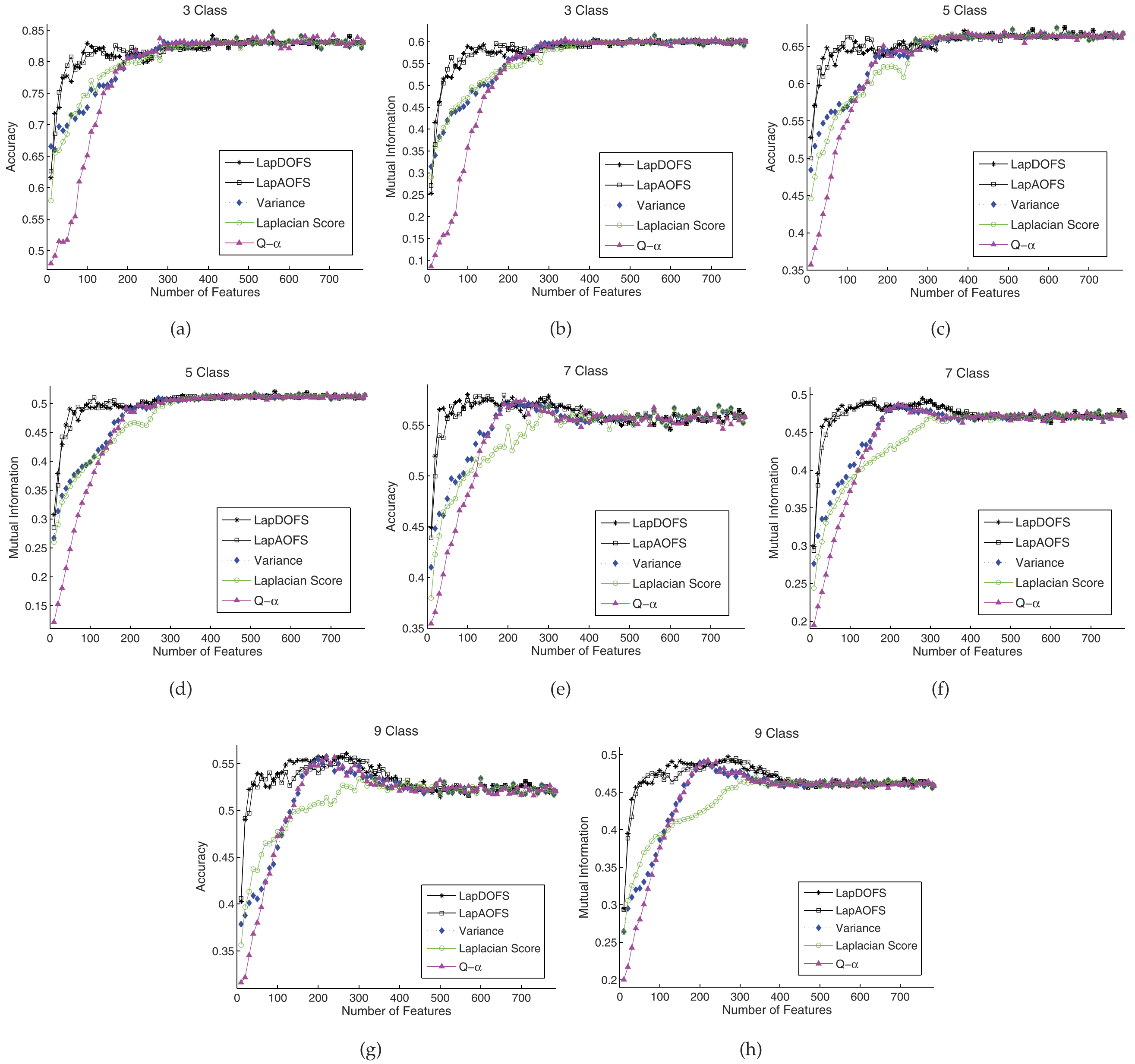
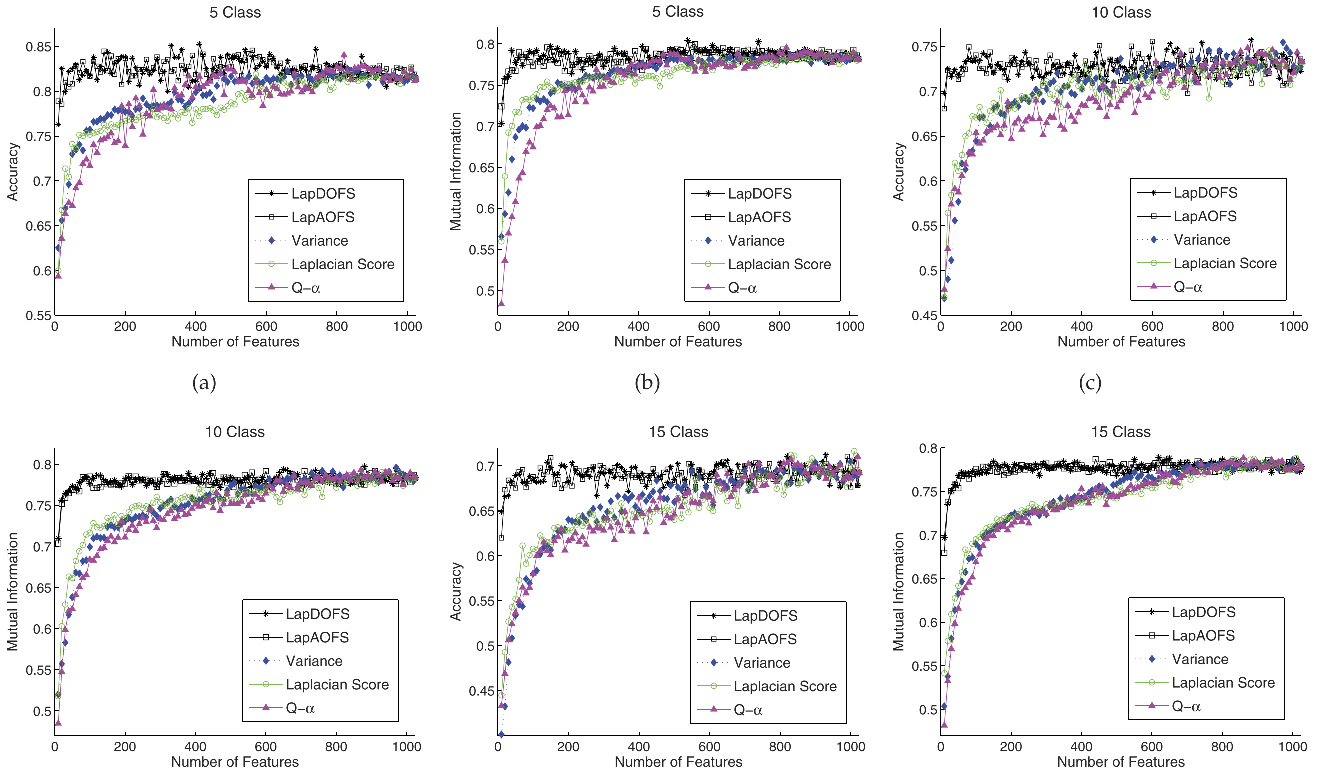


Fig. 1. Clustering accuracy and normalized mutual information versus the number of selected features on the MNIST data set.

COIL20, the number of clusters is taken to be 5, 10, and 15. Fig. 2 shows the plots of clustering performance versus the number of selected features. For AT&T, the number of clusters is taken to be 10, 20, and 30. Fig. 3 shows the clustering performance comparison.

As we can see, our proposed LapAOFS and LapDOFS algorithms consistently outperform all of the other feature selection algorithms on the MNIST, COIL20, and AT&T data sets. Both LapAOFS and LapDOFS converge to the best results very fast, with typically no more than 100 features. For all of the other methods, they usually require $300 \sim 600$ features to achieve a reasonably good result, as can be seen from Figs. 1, 2, and 3. It would be interesting to note that, on the COIL20 data set, our proposed algorithms perform surprisingly well by using only 10 features. For example, when five classes are used and only 10 features are selected, the clustering accuracy (normalized mutual information) for LapAOFS and LapDOFS are 78.9 (72.4 percent) and 76.3 percent (70.4 percent), respectively. These results are even comparable to the clustering results by using *all* of the 1,024 features, that is, 81.4 percent in accuracy and 78.2 percent in normalized mutual information. We can see similar results when 10 and 15 classes are used for clustering. The Variance, Laplacian score, and $Q-\alpha$ algorithms perform comparably to one another on the MNIST and COIL20 data sets. On the AT&T data set, Laplacian score performs slightly better than Variance and $Q-\alpha$, especially when the number of selected features is less than 100.

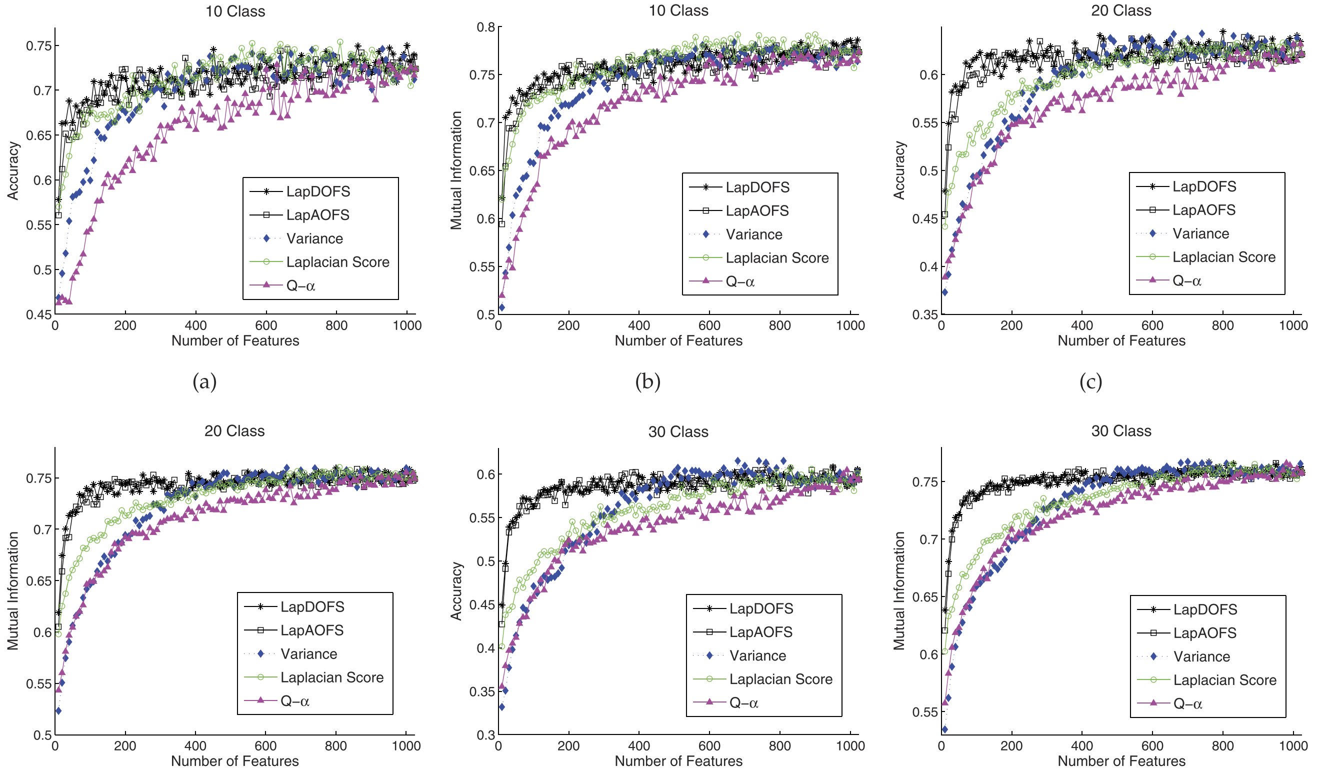
Since the goal of feature selection is to reduce the dimensionality of the data, in Tables 1, 2, and 3 we report the detailed clustering performance (accuracy and normalized mutual information) by using 100 features for each algorithm. The last two columns of each table record the average clustering performance over different numbers of



(a) (b) (c)

(d) (e) (f)

Fig. 2. Clustering accuracy and normalized mutual information versus the number of selected features on the COIL20 data set.



(a) (b) (c)

(d) (e) (f)

Fig. 3. Clustering accuracy and normalized mutual information versus the number of selected features on the AT&T data set.

clusters. As can be seen, LapAOFS and LapDOFS significantly outperform the other three methods on all the three data sets. Laplacian score performs the second best. Comparing with Laplacian score, LapAOFS achieves 16.9 percent (16.2 percent), 22.0 percent (21.8 percent), and 10.9 percent (10.7 percent) relative error reduction in average

TABLE 1
Clustering Performance (Percent) by Using 100 Features on the MNIST Data Set

	3 clusters		5 clusters		7 clusters		9 clusters		average	
	AC	NMI	AC	NMI	AC	NMI	AC	NMI	AC	NMI
LapDOFS	**83.0**	**59.0**	64.3	49.3	**58.0**	48.3	**53.9**	**47.9**	**64.8**	**51.1**
LapAOFS	81.2	56.9	**66.3**	**50.4**	57.5	**48.4**	53.7	47.4	64.7	50.8
Laplacian Score	74.7	47.1	57.3	39.9	50.3	38.7	47.7	39.3	57.5	41.3
Variance	72.8	46.1	57.0	39.8	51.6	40.5	46.0	38.7	56.9	41.3
$Q-\alpha$	65.1	35.8	55.0	36.0	48.1	37.2	47.2	37.6	53.9	36.7
All	83.1	60.1	66.7	51.5	55.8	47.2	52.1	45.9	64.4	51.2

The last row records the clustering performance by using all of the 784 features.

accuracy (average normalized mutual information) on the MNIST, COIL20, and AT&T data sets, respectively. Similarly, LapDOFS achieves 17.2 percent (16.7 percent), 21.1 percent (22.2 percent), and 11.4 percent(12.0 percent) relative error reduction in average accuracy (average normalized mutual information) on the MNIST, COIL20, and AT&T data sets, respectively. The last row of each table records the clustering performances by using all the features.

7.3 Nearest Neighbor Classification

In this section, we evaluate the discriminating power of different feature selection algorithms. We consider the nearest neighbor classifier. The good features should yield high classification accuracy.

We perform leave-one-out cross validation as follows: For each data point $\mathbf{x}_i$, we find its nearest neighbor $\mathbf{x}'_i$. Let $c(\mathbf{x}_i)$ be the class label of $\mathbf{x}_i$. The nearest neighbor classification accuracy is thus defined as

$$\text{Accuracy} = \frac{1}{m}\sum_{i=1}^{m}\delta\big(c(\mathbf{x}_i), c(\mathbf{x}'_i)\big),$$

where m is the number of data points and $\delta(a,b)=1$ if $a=b$ and 0 otherwise. Figs. 4, 5, and 6 show the plots of nearest neighbor classification accuracy versus the number of selected features.

TABLE 2
Clustering Performance (Percent) by Using 100 Features n the COIL20 Data Set

	5 clusters		10 clusters		15 clusters		average	
	AC	NMI	AC	NMI	AC	NMI	AC	NMI
LapDOFS	**81.7**	**78.9**	73.4	77.7	68.4	77.0	74.5	**77.9**
LapAOFS	**81.7**	77.4	**74.0**	**78.5**	**68.8**	**77.6**	**74.8**	77.8
Laplacian Score	75.2	73.5	67.1	71.7	60.8	69.5	67.7	71.6
Variance	75.7	72.2	64.4	70.0	58.0	69.0	66.0	70.4
$Q-\alpha$	71.7	67.4	65.4	68.4	58.0	66.9	65.0	67.6
All	81.4	78.2	73.3	78.4	69.2	77.8	74.6	78.1

The last row records the clustering performance by using all of the 1,024 features.

TABLE 3
Clustering Performance (Percent) by Using 100 Features on the AT&T Data Set

	10 clusters		20 clusters		30 clusters		average	
	AC	NMI	AC	NMI	AC	NMI	AC	NMI
LapDOFS	67.5	**73.9**	**61.8**	**73.7**	**56.3**	73.5	**61.9**	**73.7**
LapAOFS	**68.6**	72.8	60.1	73.5	**56.3**	**73.6**	61.7	73.3
Laplacian Score	67.9	72.9	54.3	69.0	48.9	68.4	57.0	70.1
Variance	60.0	65.8	49.4	64.6	47.1	65.9	52.2	65.4
$Q-\alpha$	54.5	62.9	49.4	64.9	45.9	66.1	49.9	64.6
All	72.4	77.3	62.1	74.9	59.4	75.8	64.6	76.0

The last row records the clustering performance by using all of the 1,024 features.

As can be seen, on all three data sets, LapAOFS and LapDOFS consistently outperform the other three methods. Similarly to clustering, both LapAOFS and LapDOFS converge to the best result very fast, with no more than 100 features. Particularly on the COIL20 data set, LapAOFS and LapDOFS can achieve 100 and 99.51 percent classification accuracy by using only 30 features, respectively. That is, out of 1,440 data points, only seven data points are misclassified by

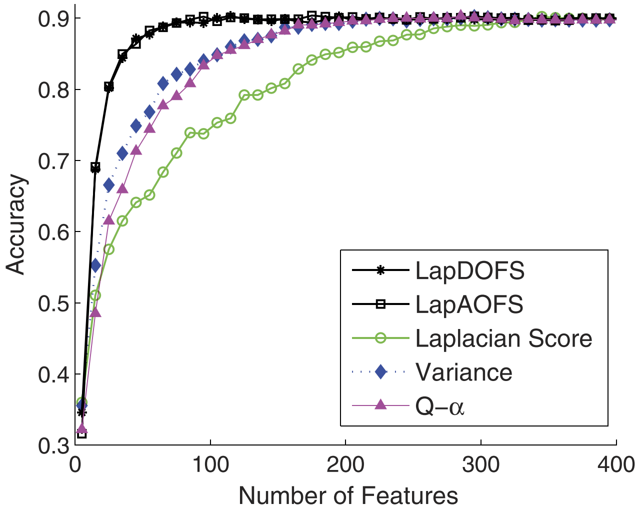


Fig. 4. Nearest neighbor classification accuracy versus the number of selected features on the MNIST data set.

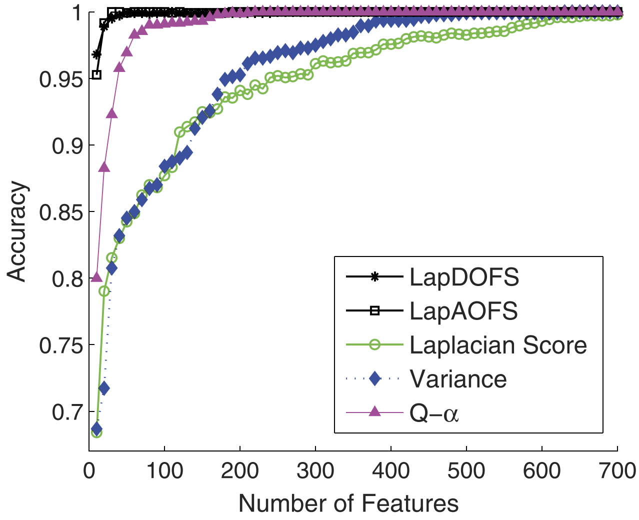


Fig. 5. Nearest neighbor classification accuracy versus the number of selected features on the COIL20 data set.

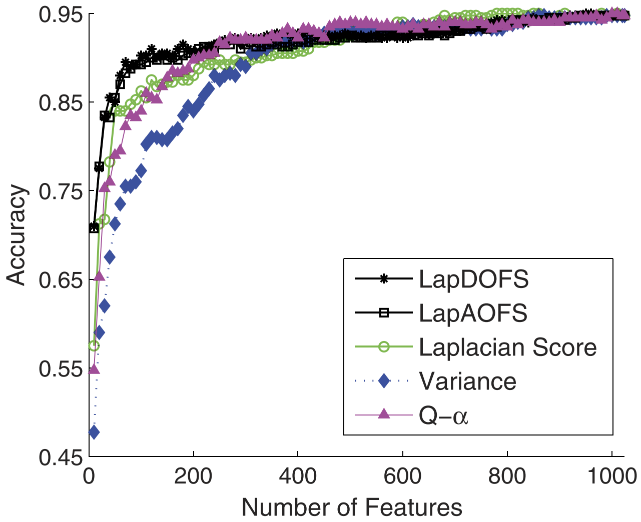


Fig. 6. Nearest neighbor classification accuracy versus the number of selected features on the AT&T data set.

LapDOFS and all of the data points are correctly classified by LapAOFS. On this data set, the $Q-\alpha$ algorithm performs comparably to our algorithms and much better than Variance and Laplacian Score, especially when the number of selected features is less than 100. On the MNIST data set, the Variance and $Q-\alpha$ algorithms perform comparably to each other, and Laplacian Score performs the worst. On the AT&T data set, Laplacian Score and $Q-\alpha$ perform comparably and Variance performs the worst.

Similarly to clustering, in Table 4 we show the nearest neighbor classification accuracy for each algorithm using only 100 features. As can be seen, both LapDOFS and LapAOFS achieve comparable results to that using all of the features.

TABLE 4
Nearest Neighbor Classification Accuracy Comparison by Using 100 Selected Features

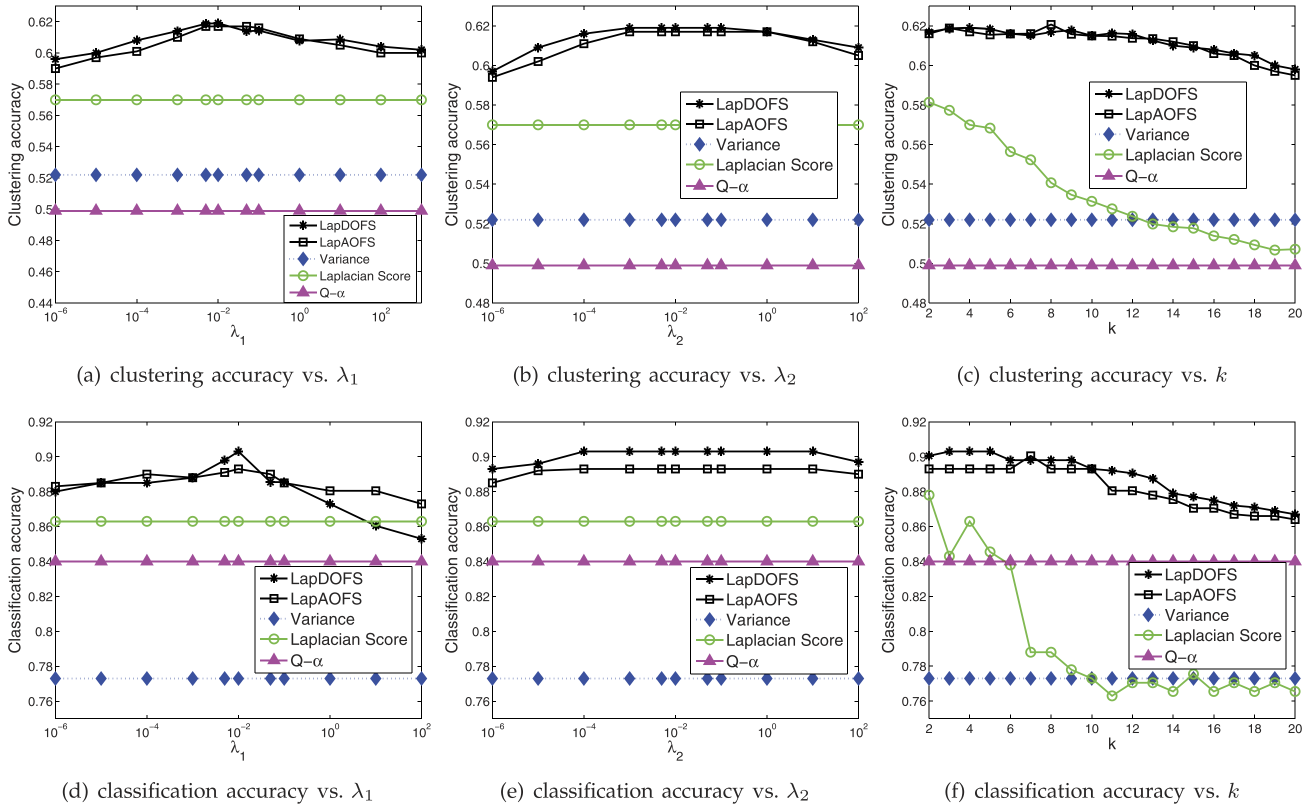
	MNIST (%)	COIL20 (%)	AT&T (%)
LapDOFS	**89.6**	**100**	**90.3**
LapAOFS	89.3	**100**	89.3
Laplacian Score	74.9	87.7	86.3
Variance	84.1	88.4	77.3
$Q-\alpha$	85.0	99.1	84.0
All	89.6	100	94.8

The last row records the performance by using all of the features.

7.4 Parameters Selection

Our algorithms have three parameters, that are, regularization parameters (λ_1 and λ_2) and number of nearest neighbors (k). For unsupervised feature selection, model selection is especially difficult since there is no label information available. Thus, standard model selection method such as cross validation cannot be applied. In this section, we evaluate how the algorithms perform with different values of the parameters. The data set used for this test is the AT&T face database. By applying LapAOFS and LapDOFS, we select 100 features.

Figs. 7a, 7b, and 7c show the average clustering accuracy over 10, 20, and 30 clusters as a function of each of these three parameters. Figs. 7d, 7e, and 7f show the classification accuracy as a function of each of these three parameters. As



(a) clustering accuracy vs. λ_1

(b) clustering accuracy vs. λ_2

(c) clustering accuracy vs. k

(d) classification accuracy vs. λ_1

(e) classification accuracy vs. λ_2

(f) classification accuracy vs. k

Fig. 7. The clustering and classification accuracies versus parameters λ_1, λ_2, and k.

we can see, the performance of LapAOFS and LapDOFS is very stable with respect to these parameters. Moreover, LapAOFS and LapDOFS can achieve significantly better performance than the other three algorithms over a large range of λ_1 and λ_2. The Laplacian Score algorithm shares the same parameter k with our proposed algorithms, which defines the "locality" of the data manifold. Recall that the Laplacian regularization is based on the assumption that two data points may share the same label if they are sufficiently close to each other. However, as k increases, two points within the same neighborhood may have different labels and the Laplacian regularizer can no longer capture the local geometrical and discriminant structures. From Figs. 7c and 7f, we can see that the performance of both our algorithms and Laplacian Score decreases as k increases. However, comparing to Laplacian Score, our algorithms are much more stable. From these results, we see that the selection of parameters is not a very crucial problem in our proposed algorithms.

7.5 Summary

The clustering and nearest neighbor classification experiments on three databases have been systematically performed. These experiments reveal a number of interesting points:

1. On all three data sets, LapAOFS and LapDOFS consistently outperform the other three algorithms for both clustering and nearest neighbor classification. As the number of selected features increases, the clustering and nearest neighbor classification performance for all the methods increase and the performance difference among different methods gets smaller.
2. Our proposed algorithms perform especially well when the number of selected features is small (e.g., $\ell < 100$). By using only 100 features, the performance of our proposed LapAOFS and LapDOFS algorithms are comparable to and sometimes even better than (see Tables 1, 2, 3, and 4) the performance by using all the features. Therefore, comparing to Variance, Laplacian Score, and $Q-\alpha$, our algorithms can achieve much more compact representation without sacrifice of discriminating power.
3. In all the cases, the difference between LapAOFS and LapDOFS is very small. This indicates that the choice of different experimental design criteria may not be very critical.

8 Conclusions and Future Work

This paper presents two novel feature selection algorithms, called LapAOFS and LapDOFS, from an experimental design perspective. By using A-optimality criterion, LapAOFS selects those features such that the trace of the parameter (corresponding to the selected features) covariance matrix is minimized. Likewise, LapDOFS aims to minimize the determinant of the parameter covariance matrix. Since our proposed algorithms essentially aim to minimize the expected prediction error of the data points, they can have more discriminating power. In comparison with one simple method, that is, Variance, and two state-of-the-art methods, namely, Laplacian Score and $Q-\alpha$, the experimental results validate that the new methods achieve significantly higher accuracy for clustering and classification. Our proposed LapAOFS and LapDOFS algorithms perform especially well when the number of selected features is less than 100.

In this paper, we applied A and D-optimality criteria to evaluate the size of the parameter covariance matrix. There are also many other choices of the design criteria, such as E and G-optimality. E-optimality maximizes the minimum eigenvalue of the parameter covariance matrix and G-optimality minimizes the maximum variance of the predicted values [1]. Although our empirical tests show that the feature selection approach is not sensitive to the choice of optimality criteria, it is worthwhile to further investigate the performance of other optimality criteria for feature selection. Moreover, in this paper we consider the unsupervised feature selection problem. It remains unclear how to make use of label information to enhance the feature selection algorithms when it is available. The simplest way might be incorporating the label information into the graph structure. For example, if two points share the same label, then we can assign a large weight on the edge connecting them. Finally, the framework of analysis presented here is primarily focused on feature selection. However, the experimental design techniques have been conventionally applied to data selection (or active learning) [1], [12], [18]. It is thus natural to perform both data and feature selection simultaneously within the experimental design framework. We are currently exploring these problems in theory and practice.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 60633070 and 60875044, and the National Key Basic Research Foundation of China under Grant 2009CB320801.

References

[1] A.C. Atkinson and A.N. Donev, *Optimum Experimental Designs.* Oxford Univ. Press, 2007.

[2] S. Basu, C.A. Micchelli, and P. Olsen, "Maximum Entropy and Maximum Likelihood Criteria for Feature Selection from Multivariate Data," *Proc. IEEE Int'l Symp. Circuits and Systems,* 2000.

[3] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems 14,* pp. 585-591, MIT Press, 2001.

[4] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Examples," *J. Machine Learning Research,* vol. 7, pp. 2399-2434, 2006.

[5] T.D. Bie, "Deploying sdp for Machine Learning," *Proc. 15th European Symp. Artificial Neural Networks,* Apr. 2007.

[6] S. Boutemedjet, N. Bouguila, and D. Ziou, "A Hybrid Feature Extraction Selection Approach for High-Dimensional Non-Gaussian Data Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 8, pp. 1429-1443, Aug. 2009.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge Univ. Press, 2004.

[8] F.R.K. Chung, *Spectral Graph Theory,* vol. 92 of Regional Conf. Series in Math. AMS, 1997.

[9] M. Dash and H. Liu, "Unsupervised Feature Selection," *Proc. Pacific Asia Conf. Knowledge Discovery and Data Mining,* 2000.

[10] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. Wiley-Interscience, 2000.

[11] J.G. Dy and C.E. Brodley, "Feature Subset Selection and Order Identification for Unsupervised Leanring," *Proc. 17th Int'l Conf. Machine Learning,* 2000.

[12] P. Flaherty, M.I. Jordan, and A.P. Arkin, "Robust Design of Biological Experiments," *Advances in Neural Information Processing Systems 18,* MIT Press, 2005.

[13] G.H. Golub and C.F.V. Loan, *Matrix Computations,* third ed. Johns Hopkins Univ. Press, 1996.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification Using Support Vector Machines," *Machine Learning,* vol. 46, pp. 389-422, 2002.

[15] D.A. Harville, *Matrix Algebra from a Statistician's Perspective.* Springer-Verlag, 1997.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, 2001.

[17] X. He, D. Cai, and P. Niyogi, "Laplacian Score for Feature Selection," *Advances in Neural Information Processing Systems 18,* MIT Press, 2005.

[18] X. He, M. Ji, and H. Bao, "A Unified Active and Semi-Supervised Learning Framework for Image Compression," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2009.

[19] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant Locally Linear Embedding with High-Order Tensor Data," *IEEE Trans. Systems, Man, and Cybernetics, Part B,* vol. 38, no. 2, pp. 342-352, Apr. 2008.

[20] X. Li and Y. Pang, "Deterministic Column-Based Matrix Decomposition," *IEEE Trans. Knowledge and Data Eng.,* vol. 22, no. 1, pp. 145-149, Jan. 2010.

[21] W. Liu, D. Tao, and J. Liu, "Transductive Component Analysis," *Proc. IEEE Int'l Conf. Data Mining,* 2008.

[22] L. Lovasz and M. Plummer, *Matching Theory.* Akadémiai Kiadó, 1986.

[23] M. Robnik-Sikonja and I. Kononenko, "Theoretical and Empirical Analysis of Relief and Relieff," *Machine Learning,* vol. 53, nos. 1/2, pp. 23-69, 2003.

[24] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 5500, pp. 2323-2326, 2000.

[25] J. Sturm, "Using Sedumi 1.02, a Matlab Toolbox for Optimization over Symmetric Cones," *Optimization Methods and Software,* vol. 11, nos. 1-4, pp. 625-653, 1999.

[26] D. Tao, X. Li, X. Wu, and S.J. Maybank, "General Averaged Divergence Analysis," *Proc. IEEE Int'l Conf. Data Mining,* 2007.

[27] D. Tao, X. Li, X. Wu, and S.J. Maybank, "Geometric Mean for Subspace Selection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 2, pp. 260-274, Feb. 2009.

[28] J. Tenenbaum, V. de Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, no. 5500, pp. 2319-2323, 2000.

[29] K.C. Toh, M.J. Todd, and R.H. Tütüncü, "Sdpt3—A Matlab Software Package for Semidefinite Programming," *Optimization Methods and Software,* vol. 11, nos. 1-4, pp. 545-581, 1999.

[30] L. Vandenberghe, S. Boyd, and S.-P. Wu, "Determinant Maximization with Linear Matrix Inequality Constraints," *SIAM J. Matrix Analysis and Applications,* vol. 19, no. 2, pp. 499-533, 1998.

[31] D. Ververidis and C. Kotropoulos, "Information Loss of the Mahalanobis Distance in High Dimensions: Application to Feature Selection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 12, pp. 2275-2281, Dec. 2009.

[32] L. Wolf and A. Shashua, "Feature Selection for Unsupervised and Supervised Inference: The Emergence of Sparsity in a Weight-Based Approach," *J. Machine Learning Research,* vol. 6, pp. 1855-1887, 2005.

[33] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non-Negative Matrix Factorization," *Proc. 2003 Int'l Conf. Research and Development in Information Retrieval,* pp. 267-273, Aug. 2003.

[34] J. Zhao, K. Lu, and X. He, "Locality Sensitive Semi-Supervised Feature Selection," *Neurocomputing,* vol. 71, nos. 10-12, pp. 1842-1849, 2008.

[35] Z. Zhao and H. Liu, "Spectral Feature Selection for Supervised and Unsupervised Learning," *Proc. 24th Int'l Conf. Machine Learning,* 2007.

Xiaofei He received the BS degree in computer science from Zhejiang University, China, in 2000 and the PhD degree in computer science from the University of Chicago in 2005. He is a professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University, he was a research scientist at Yahoo! Research Labs, Burbank, California. His research interests include machine learning, information retrieval, and computer vision. He is a senior member of the IEEE.

Ming Ji received the BS degree in computer science from Zhejiang University, China, in 2009. She is currently working toward the PhD degree in computer science at the University of Illinois at Urbana Champaign. Her research interests include machine learning, data mining, and information retrieval.

Chiyuan Zhang received the BS degree in computer science from Zhejiang University, China, in 2009. He is currently working toward the master's degree in computer science at Zhejiang University. His research interests include machine learning, computer vision, and information retrieval.

Hujun Bao received the bachelor's and PhD degrees in applied mathematics from Zhejiang University in 1987 and 1993, respectively. He is currently the director of the State Key Lab of CAD&CG at Zhejiang University, China. His research interests include computer graphics, computer vision, and virtual reality.