Pattern Recognition Letters 34 (2013) 527–535

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Minimum–maximum local structure information for feature selection

Wenjun Hu [a,b,c,*], Kup-Sze Choi [c], Yonggen Gu [a], Shitong Wang [b]

[a] School of Information and Engineering, Huzhou Teachers College, Huzhou, Zhejiang, China
[b] School of Digital Media, Jiangnan University, Wuxi, Jiangsu, China
[c] Centre for Integrative Digital Health, School of Nursing, Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Article history:
Received 16 May 2012
Available online 29 November 2012

Communicated by G. Borgefors

Keywords:
Feature selection
Laplacian Score
Locality preserving
Laplacian Eigenmap
Manifold learning

ABSTRACT

Feature selection methods have been extensively applied in machine learning tasks, such as computer vision, pattern recognition, and data mining. These methods aim to identify a subset of the original features with high discriminating power. Among them, the feature selection technique for unsupervised tasks is more attractive since the cost to obtain the labels of the data and/or the information between classes is often high. On the other hand, the low-dimensional manifold of the "same" class data is usually revealed by considering the local invariance of the data structure, it may not be adequate to deal with unsupervised tasks where the class information is completely absent. In this paper, a novel feature selection method, called Minimum–maximum local structure information Laplacian Score (MMLS), is proposed to minimize the within-locality information (i.e., preserving the manifold structure of the "same" class data) and to maximize the between-locality information (i.e., maximizing the information between the manifold structures of the "different" class data) at the same time. The effectiveness of the proposed algorithm is demonstrated with experiments on classification and clustering.

1. Introduction

High dimensional data are prevalent in machine learning tasks in various areas of artificial intelligence, e.g. human face images in computer vision and pattern recognition, or text documents in data mining. The amount of data involved is enormous that requires considerable computational time and storage space for processing. The machine learning performance for tasks such as clustering and classification is thus severely affected. To overcome the deficiency, feature selection and extraction methods are designed to identify significant feature subsets or feature combinations so as to reduce the dimensionality of the data.

The existing feature selection methods can be categorized into the wrapper methods and filter methods (Kohavi and John, 1997; Lee and Landgrebe, 1993; Sun, 2007; Sun et al., 2010; Deng et al., 2010; Liang et al., 2009). The former employ classification algorithms to evaluate the goodness of the selected feature subsets, while the latter apply some criterion functions, such as Fisher Score (Bishop, 1995) and Laplacian Score (He et al., 2005), to evaluate the feature subsets. The wrapper methods in general produce better results since they are directly integrated into a specific classifier. However, the computational complexity of the wrapper methods is high due to the need to train a large number of classifiers. The filter methods are advantageous in this regard as they are independent of the number of classifiers and computationally more efficient than the wrapper methods. This category of feature selection methods has thus received increasing attention. A number of supervised learning algorithms have been used to implement the filter methods, which includes the Fisher Score (Bishop, 1995), Relief family (Kira and Rendell, 1992a,b; Kononenko, 1994; Sikonja and Kononenko, 1997, 2003), Fuzzy-Margin-based Relief (FM Relief) (Deng et al., 2010), local-learning-based feature selection (Sun et al., 2010), and the Distance-discriminant-and-Distribution-overlapping-based Feature Selection (HFSDD) (Liang et al., 2009). In contrast, few unsupervised learning algorithms are available for the filter methods, e.g. Laplacian Score (He et al., 2005) and Data Variance[1]. In real-world applications, unlabeled data are usually involved and the processes to determine the labels of the data are particularly difficult and expensive, making it unfeasible to use the supervised learning algorithms. In this paper, we focus on unsupervised feature selection methods and proposed a new approach by using the manifold learning techniques.

* Corresponding author. Address: School of Information and Engineering, Huzhou Teachers College, Huzhou 313000, Zhejiang, China. Tel.: +86 572 2321106.
E-mail address: hoowenjun@yahoo.com.cn (W. Hu).

[1] Data Variance is a simplest unsupervised method which is usually used as a criterion for the feature selection and extraction. The variance along a dimension reflects its representative power, which chooses those features of maximum variance in order to obtain the best representative power. This is similar to Principal Component Analysis (PCA). However, PCA aims to find a set of orthogonal basis functions for capturing the directions of maximum variance in the data, which is a classical feature extraction method.

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.11.012

In recent years, researchers consider that data sampled from a probability distribution may be on, or in close proximity to, a submanifold of the ambient space (Tenenbaum et al., 2000; He and Niyogi, 2003; Belkin and Niyogi, 2001; Roweis and Saul, 2000; Seung and Lee, 2000). In order to reflect the underlying manifold structure, many manifold learning approaches have been proposed, such as Laplacian Eigenmap (Belkin and Niyogi, 2001), Locally Linear embedding (LLE) (Roweis and Saul, 2000), Locality Preserving Projection (LPP) (He and Niyogi, 2003), Neighbor Preserving Embedding (NPE) (He et al., 2005), ISOMAP (Tenenbaum et al., 2000). These algorithms assume that the local structure of the manifold is invariant and the learning methods should preserve the local invariance. The Laplacian Score (He et al., 2005) and its extensions (Zhao et al., 2008), where local structures are considered, have demonstrated relatively better performance in feature selection.

Motivated by manifold learning and the Laplacian Score technique (He et al., 2005), we propose a novel unsupervised feature selection algorithm called the minimum–maximum local structure information Laplacian Score (MMLS) in this paper. The algorithm takes two different pieces of local structure information into account, the within-locality information and the between-locality information. The within-locality information is minimized to preserve local structure for identifying the manifold structure of the "same" class, whereas the between-locality information is maximized to release the information of the "different" classes and increase the discriminating power of the selected feature subset. Although MMLS is an unsupervised feature selection method, it intrinsically enables information interactions among the "different" classes by minimizing and maximizing the within-locality and between-locality information respectively.

The rest of the paper is organized as follows. Section 2 provides a brief description of the work related to the proposed algorithm. Section 3 introduces the MMLS algorithm and discusses the solving scheme. The experiments used to evaluate the performance of the MMLS algorithm on classification and clustering are presented in Section 4. The paper is concluded in Section 5.

2. Related works

The work related to the proposed MMLS method includes Laplacian Eigenmap (Belkin and Niyogi, 2001), Locality Preserving Projection (LPP) (He and Niyogi, 2003) and Laplacian Score (He et al., 2005). These algorithms are briefly described in the section. In the discussions that follow, let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m]$ denote the dataset with $\boldsymbol{x}_i \in \Re^D$, sampled from a d-dimension submanifold embedded in $\Re^D$.

2.1. Laplacian Eigenmap and LPP

Laplacian Eigenmap is one of the most popular manifold learning methods which are based on spectral graph theory (Fan, 1997). Generally speaking, the aim of manifold learning method is to find an optimal map which maintains the intrinsic geometry structure of the data manifold. Let $\boldsymbol{y} = [y_1, \ldots, y_m]$ be the 1-dimensional map of the data set $\boldsymbol{X}$. Given a k-nearest neighbor graph G with weight matrix $\boldsymbol{W}$, which reveals the neighborhood relationship between the data points, the Laplacian Eigenmap tries to obtain the optimal maps by solving the following optimization criterion:

$$\min \sum_{i,j=1}^{m} (y_i - y_j)^2 \boldsymbol{W}_{ij},$$

where $\boldsymbol{W}_{ij}$ denotes the (i,j)th entry of the weight matrix $\boldsymbol{W}$. It can be shown with some simple algebraic steps that the above criterion is equivalent to:

$$\min \boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y},$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the Laplacian matrix (Fan, 1997; Belkin and Niyogi, 2001), and $\boldsymbol{D}$ is a diagonal matrix whose entries along the diagonal are the column sum of $\boldsymbol{W}$, i.e., $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$. In fact, a heavy penalty is applied to the objective function through the weight $\boldsymbol{W}_{ij}$ if the neighboring data $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are mapped far apart. Hence, the minimization criterion is an attempt to ensure the points y_i and y_j are also close to each other as well if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close". That is to say, the obtained optimal map attempts to preserve the local structure of the data set. To express the degree of closeness, the weight matrix $\boldsymbol{W}$ can be defined as:

$$\boldsymbol{W}_{ij} = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2t^2}\right) & \text{if } \boldsymbol{x}_i \in N(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j \in N(\boldsymbol{x}_i), \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where $\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2t^2)$ is called the heat kernel function, t is a constant which is usually called the kernel width parameter, and $N(\boldsymbol{x}_j)$ denotes the set of k-nearest neighbors or of the ε neighborhoods of $\boldsymbol{x}_j$. The weight is sometimes simplified as the so-called "binary weight" or "0–1 weight", as follows:

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

The low dimension manifold embedding can be obtained by solving a generalized minimum eigenvalue problem, i.e., $\boldsymbol{L}\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}$. Clearly, Laplacian Eigenmap is a nonlinear algorithm. A linear manifold learning algorithm based on Laplacian Eigenmap, called Locality Preserving Projection (LPP), is then proposed (He and Niyogi, 2003). In LPP, $\boldsymbol{x}_i$ is linearly projected onto a base vector $\boldsymbol{v} \in \Re^D$, i.e., $y_i = \boldsymbol{v}^T \boldsymbol{x}_i$. The corresponding objective function is given by:

$$\min \sum_{i,j=1}^{m} \left(\boldsymbol{v}^T \boldsymbol{x}_i - \boldsymbol{v}^T \boldsymbol{x}_j\right)^2 \boldsymbol{W}_{ij}. \tag{3}$$

Here, the purpose of the LPP is to find a basis $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d] \in \Re^{D \times d}$ and obtain a subspace which preserves the local structure of the data set. With the constraint $\boldsymbol{v}^T \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^T \boldsymbol{v} = 1$ and by some algebraic steps, the objective function in Eq. (3) can be solved by considering the generalized minimum eigenvalue problem $\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T \boldsymbol{v} = \lambda \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T \boldsymbol{v}$. Details of the solution can be found in (He and Niyogi, 2003).

2.2. Laplacian Score

Laplacian Score is an unsupervised feature selection method. This method is distinctive in that a feature is evaluated according to the locality preserving power integrated in the variance of that feature. In other words, the selected features are able to preserve the consistency of the embedding manifold structure. Let $\boldsymbol{f}_r$ be a vector grouped with the rth feature of the data set $\boldsymbol{X}$, i.e., $\boldsymbol{f}_r = [f_{r1}, \ldots, f_{rm}]^T$, and each f_{ri} can be treated as a value of the random variable f_r. Thus, the criterion of the Laplacian Score for feature selection is to minimize the following objective function:

$$LS_r = \frac{\sum_{i,j=1}^{m} (f_{r_i} - f_{r_j})^2 \boldsymbol{W}_{ij}}{Var(f_r)}, \tag{4}$$

where LS_r denotes the Laplacian Score of the rth feature, $\boldsymbol{W} = [\boldsymbol{W}_{ij}]$ is the weight matrix corresponding a k-nearest neighbors graph (similar to that in Laplacian Eigenmap or LPP), and $Var(f_r)$ is the variance of the rth feature, which is estimated by the diagonal matrix $\boldsymbol{D}$ based on the spectral graph theory (Fan, 1997). The estimation of $Var(f_r)$ will be discussed in Section 3.3 (He et al., 2005). The smaller the value of LS_r, the better the selected features.

To minimize the objective function in Eq. (4), it is necessary to minimize $\sum_{i,j=1}^{m}(f_{r_i}-f_{r_j})^2\boldsymbol{W}_{ij}$ and maximize $Var(f_r)$. This can be interpreted by the fact that, for a good feature, any two data points on this feature should be close to each other if they are close to each other in original space. To be specific, the larger the value of $\boldsymbol{W}_{ij}$ (i.e., the closer $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are, or the more similar $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are), the smaller the value of $(f_{r_i}-f_{r_j})$. In other words, the importance of a feature is determined by its locality preserving power. Besides, a good feature should have high representative power, which is associated with the variance of the feature. The larger the variance, the higher the representative power.

The above-mentioned methods explicitly make use of the manifold structure and obtain the low dimension manifold by preserving local structure of the data set. The MMLS algorithm proposed in this paper also employs the manifold structure for feature extraction and is a variant of Laplacian Score. However, unlike Laplacian Score which only uses an adjacent graph, two graphs are used in the proposed algorithm: one to preserve the local structure information of the data set, and the other to retain the between-locality information hidden in the data set. Importantly, the criterion proposed for feature selection takes into the account of both the within-locality and the between-locality information.

3. Minimum and maximum information for feature selection

As described previously, the MMLS algorithm considers both the within-locality information of the dataset and the between-locality information hidden in the dataset. Both the variance of features and the geometric structures of the data set are taken into account. The algorithm is discussed in detail in this section. We begin with a description of the information in the local structure of a dataset.

3.1. Information in local structure

Given a set $\boldsymbol{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m]$ in $\Re^{D\times m}$, we first construct an undirected graph $G=(V,E)$, where V denotes the set of nodes associated with all the data points and E denotes the set of the weighted edges that connect the points in pairs. The graph is referred to as *global graph* in this paper. Let the weight matrix be $\boldsymbol{W}$, which is defined by the similarity between each pair of the points. Like Laplacian Eigenmap, LPP and Laplacian Score, we adopt the heat kernel (also called Gaussian kernel) to define the weight matrix $\boldsymbol{W}$ as follows:

$$\boldsymbol{W}_{ij}=\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2t^2}\right), \tag{5}$$

where t is a suitable constant. To obtain a nearest neighbor graph for preserving the local structure information of the data set, we split the graph $G=(V,E)$ into two subgraphs which are denoted respectively by $G_w=(V,E_w)$ and $G_b=(V,E_b)$. In the graph $G_w=(V,E_w)$, we put an edge between nodes i and j, which correspond to nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close" to each other, i.e., $\boldsymbol{x}_i\in N(\boldsymbol{x}_j)$ or $\boldsymbol{x}_j\in N(\boldsymbol{x}_i)$. On the other hand, in the graph $G_b=(V,E_b)$, we put an edge between nodes i and j if they are not "close" to each other, i.e., $\boldsymbol{x}_i\notin N(\boldsymbol{x}_j)$ and $\boldsymbol{x}_j\notin N(\boldsymbol{x}_i)$. Note that since $G=G_w+G_b$ (where the sign "+" means the union of the edge sets E_w and E_b, that is to say, $E=E_w\cup E_b$), we have $\boldsymbol{W}_w+\boldsymbol{W}_b=\boldsymbol{W}$, where $\boldsymbol{W}_w$ and $\boldsymbol{W}_b$ are weight matrices of G_w and G_b respectively. Furthermore, the entries of $\boldsymbol{W}_w$ and $\boldsymbol{W}_b$ can be obtained from the weight matrix $\boldsymbol{W}$ of the graph G as follows:

$$\boldsymbol{W}_w=\begin{cases}\boldsymbol{W}_{ij} & \text{if } \boldsymbol{x}_i\in N(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j\in N(\boldsymbol{x}_i)\\ 0 & \text{otherwise,}\end{cases} \tag{6}$$

and

$$\boldsymbol{W}_b=\begin{cases}\boldsymbol{W}_{ij} & \text{if } \boldsymbol{x}_i\notin N(\boldsymbol{x}_j) \text{ and } \boldsymbol{x}_j\notin N(\boldsymbol{x}_i)\\ 0 & \text{otherwise}\end{cases}. \tag{7}$$

The weight matrices have the following properties. First, $\boldsymbol{W}_w$ and $\boldsymbol{W}_b$ are complementary with reference to $\boldsymbol{W}$. The two matrices are also symmetric. $\boldsymbol{W}_w$ possesses all the properties of the adjacent matrix in the Laplacian Eigenmap, LPP and Laplacian Score. It reflects the within-locality structure of the dataset. On the other hand, $\boldsymbol{W}_b$ does not take into account the set of k-nearest neighbors or the ε neighborhoods. It represents the between-locality structure of the data set. Hence, two different kinds of information are defined for the local structure of the dataset: the information of between-locality structure, denoted as $\sum_{i,j=1}^{m}(y_i-y_j)^2\boldsymbol{W}_{b,ij}$, and the information of within-locality structure, denoted as $\sum_{i,j=1}^{m}(y_i-y_j)^2\boldsymbol{W}_{w,ij}$.

3.2. The minimum–maximum-information criterion for feature selection

A good feature should be able to represent the graph structure of the data set properly. For two points with the nearest relationship, a "good" feature should be able to preserve this relationship, i.e., their within-locality information should be minimized. For two points without the nearest relationship, their between-locality information should be maximized. Therefore, a reasonable criterion for feature selection, called the Minimum–maximum-information criterion (MMIC), is defined by minimizing the objective function below:

$$\mathrm{MMLS}_r=\frac{(1-\alpha)\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{w,ij}-\alpha\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{b,ij}}{Var(f_r)}, \tag{8}$$

where α is a controlling parameter and $0\leqslant\alpha\leqslant 1$, MMLS_r denotes the Minimum–maximum-information Laplacian Score of the rth feature, and $Var(f_r)$ is the estimated variance of the rth feature. Minimizing the objective function in Eq. (8) is equivalent to minimizing $\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{w,ij}$ while maximizing $\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{b,ij}$ and $Var(\boldsymbol{f}_r)$. The aim to minimize $\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{w,ij}$ is to preserve the neighbor relationship on the rth feature between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, if they are really close to each other. Otherwise, the non-neighbor relationship on the rth feature should be preserved by maximizing $\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{b,ij}$. Besides, $Var(f_r)$ is maximized to enhance the representative power, as in the case of Laplacian Score.

3.3. Solution of the criterion

To solve the minimum–maximum-information criterion in Eq. (8), we first consider the numerator in the objective criterion. Following some simple algebraic steps, we obtain

$$\begin{aligned}&(1-\alpha)\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{w,ij}-\alpha\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{b,ij}\\&=\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{w,ij}-\alpha\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2(\boldsymbol{W}_{b,ij}+\boldsymbol{W}_{w,ij})\\&=\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{w,ij}-\alpha\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{W}_{ij}\\&=\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2(\boldsymbol{W}_{w,ij}-\alpha\boldsymbol{W}_{ij}).\end{aligned}$$

Let $\boldsymbol{A}=\boldsymbol{W}_w-\alpha\boldsymbol{W}$, and since $\boldsymbol{A}$ is symmetric, the above formula can be rewritten as follows:

$$\begin{aligned}
&\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2(\boldsymbol{W}_{w,ij}-\alpha\boldsymbol{W}_{ij})=\sum_{i,j=1}^{m}(f_{ri}-f_{rj})^2\boldsymbol{A}_{ij}\\
&=\sum_{i,j=1}^{m}(2f_{ri}^2\boldsymbol{A}_{ij}-2f_{ri}f_{rj}\boldsymbol{A}_{ij})\\
&=2\boldsymbol{f}_r^T\boldsymbol{D}\boldsymbol{f}_r-2\boldsymbol{f}_r^T\boldsymbol{A}\boldsymbol{f}_r\\
&=2\boldsymbol{f}_r^T\boldsymbol{L}\boldsymbol{f}_r,
\end{aligned}\tag{9}$$

where $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}$ is the Laplacian matrix (He et al., 2005; Fan, 1997), $\boldsymbol{D}$ is a diagonal matrix and its entries along the diagonal are the column sum of $\boldsymbol{A}$, i.e., $\boldsymbol{D}_{ii}=\sum_j\boldsymbol{A}_{ij}$.

Next, the denominator of the objective criterion in Eq. (8) is considered. Recall the definition of the variance of a random variable f_r, we have

$$Var(f_r)=\int_M(f_r-\mu_r)^2p(f_r)df_r,$$

where M is the data manifold, μ_r and $p(f_r)$ are respectively the expectation and the probability density of the random variable f_r. Following the spectral graph theory (Fan, 1997), $p(f_r)$ can be estimated by the diagonal matrix $\boldsymbol{D}$ of the sampled data. Therefore, the variance of f_r, weighted by matrix $\boldsymbol{A}$, can be estimated as follows:

$$Var(f_r)=\sum_{i=1}^{m}(f_{ri}-\mu_r)^2\boldsymbol{D}_{ii},\tag{10}$$

where the expectation μ_r can be estimated by:

$$\mu_r=\frac{\sum_{i=1}^{m}f_{ri}\boldsymbol{D}_{ii}}{\sum_{i=1}^{m}\boldsymbol{D}_{ii}}=\frac{\boldsymbol{f}_r^T\boldsymbol{D}1}{1^T\boldsymbol{D}1},\tag{11}$$

where 1 is the vector of ones. Let $\tilde{f}_{ri}=f_{ri}-\mu_r$, from Eq. (10) we have

$$Var(f_r)=\sum_{i=1}^{m}\tilde{f}_{ri}^2\boldsymbol{D}_{ii}=\tilde{\boldsymbol{f}}_r^T\boldsymbol{D}\tilde{\boldsymbol{f}}_r.\tag{12}$$

Since $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}$ and $\boldsymbol{D}_{ii}=\sum_j\boldsymbol{A}_{ij}$, $\boldsymbol{L}1=(\boldsymbol{D}-\boldsymbol{A})1=0$. Then, we have

$$\begin{aligned}
&\tilde{\boldsymbol{f}}_r^T\boldsymbol{L}\tilde{\boldsymbol{f}}_r=(\boldsymbol{f}_r-\mu_r1)^T\boldsymbol{L}(\boldsymbol{f}_r-\mu_r1)\\
&=\boldsymbol{f}_r^T\boldsymbol{L}\boldsymbol{f}_r+\mu_r1^T\boldsymbol{L}\mu_r1-\boldsymbol{f}_r^T\boldsymbol{L}\mu_r1-\mu_r\boldsymbol{1}^T\boldsymbol{L}\boldsymbol{f}_r\\
&=\boldsymbol{f}_r^T\boldsymbol{L}\boldsymbol{f}_r.
\end{aligned}\tag{13}$$

Therefore, the Minimum–maximum-information Laplacian Score of the rth feature in Eq. (8) can be computed as follows:

$$\mathrm{MMLS}_r=\frac{\tilde{\boldsymbol{f}}_r^T\boldsymbol{L}\tilde{\boldsymbol{f}}_r}{\tilde{\boldsymbol{f}}_r^T\boldsymbol{D}\tilde{\boldsymbol{f}}_r}.\tag{14}$$

To summarize, the algorithm for computing MMLS_r is presented as follows:

Algorithm 1: Pseudo-code for computing the minimum–maximum-information Laplacian Score

Input: Data matrix $\boldsymbol{X}=[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m]\in\mathcal{R}^{D\times m}$, controlling parameter α ($0\leqslant\alpha\leqslant1$), kernel width parameter t, the number of nearest neighbors k (or the radius of neighborhoods ε).

Output: The minimum–maximum-information Laplacian Score MMLS_r ($1\leqslant r\leqslant D$).

1. Construct the global graph G and compute the corresponding weight matrix $\boldsymbol{W}$ using Eq. (5);
2. Construct the adjacent graph G_w and compute the corresponding weight matrix $\boldsymbol{W}_w$ using Eq. (6);
3. Compute a new weight matrix $\boldsymbol{A}=\boldsymbol{W}_w-\alpha\boldsymbol{W}$, diagonal matrix $\boldsymbol{D}$ ($\boldsymbol{D}_{ii}=\sum_j\boldsymbol{A}_{ij}$), Laplacian matrix $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}$;
4. For $r=1:D$
   Take the rth feature of $\boldsymbol{X}$, i.e., $\boldsymbol{f}_r=[\boldsymbol{X}_{r1},\cdots,\boldsymbol{X}_{rm}]^T$;
   Compute μ_r using Eq. (11);
   Compute $\tilde{\boldsymbol{f}}=\boldsymbol{f}_r-\mu_r\boldsymbol{1}$;
   Compute MMLS_r using Eq. (14);
   End For
5. Output MMLS_r.

Thus, the optimal feature subset can be determined by sorting the D features in the ascending order according to the Minimum–maximum-information Laplacian Score MMLS_r. The optimal subset with d features is then obtained from the first d features in the sorted features.

3.4. Computational complexity analysis

The proposed MMLS algorithm consists of five steps summarized in Algorithm 1. The computational cost of each step is provided as follows:

(1) Constructing the global graph needs $O(m^2D)$ to compute the pair wise distances, where m and D are the number of input data points and the dimension of input data respectively.
(2) Constructing the adjacent graph needs $O(m^2k)$ to find k neighbors for each point.
(3) The third step needs $O(m^2)$ time without considering additive.
(4) MMLS score for all the features can be computed within $O(4m^2)$ without considering additive.
(5) The optimal subset with d features can be obtained within $O(md)$.

Considering $D\gg k$, the total complexity for the proposed MMLS algorithm is $O(m^2D+md)$. Besides, for clearer comparison, the computational complexities of other unsupervised algorithms including Laplacian Score (He et al., 2005), the Variance of data (VAR), Multi-Cluster Feature Selection (MCFS) (Cai et al., 2010), are summarized in Table 1, where K is the cluster number. These algorithms will be investigated in our experiments below.

4. Experiments

A number of experiments are carried out to investigate the effectiveness of the proposed MMLS algorithm for feature selection. Classification and clustering experiments are performed with the chosen feature subset. As an unsupervised algorithm, MMLS is compared with the methods of Laplacian Score (LS) (He et al., 2005), the Variance of data (VAR) and Multi-Cluster Feature Selection (MCFS) (Cai et al., 2010) in the following discussions. The aim of the VAR method is to select the features with maximum variances in order to obtain the best representative power. MCFS selects the features according to spectral clustering and in the experiments the cluster number is set to be the class number of the data set. In the proposed MMLS algorithm, the controlling parameter α is searched from the grid $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. In MMLS, Laplacian Score and MCFS, we adopt the set of k-nearest neighbors for constructing the adjacent graph, where the parameter k is empirically set to be five in the following experiments, and the corresponding weight is defined via the heat kernel function Eq. (5), where the parameter t is set to the mean squared norm of the input data.

Table 1
Computational Complexity of LS, VAR, MCFS and MMLS.

	Laplacian Score	VAR	MCFS	MMLS
Complexity	$O(m^2D + md)$	$O(mD + md)$	$O(m^2D + md + mKd^2 + Kd^3)$	$O(m^2D + md)$

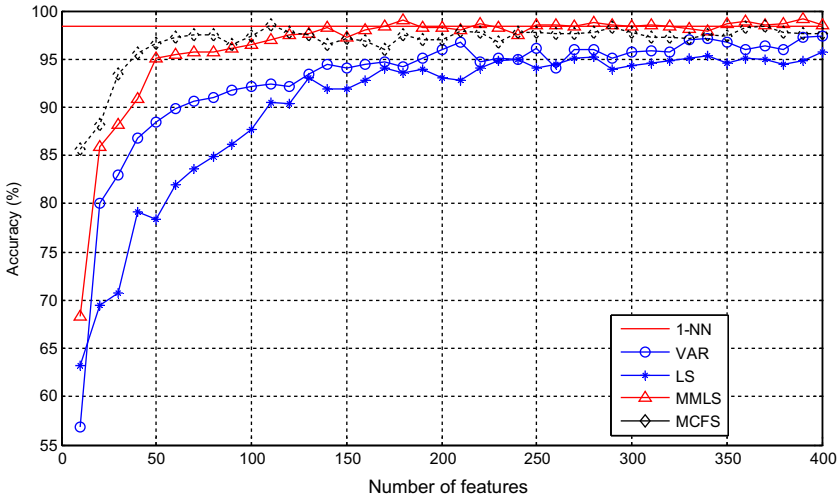


Fig. 1. Classification accuracy vs. the number of selected features on COIL20.

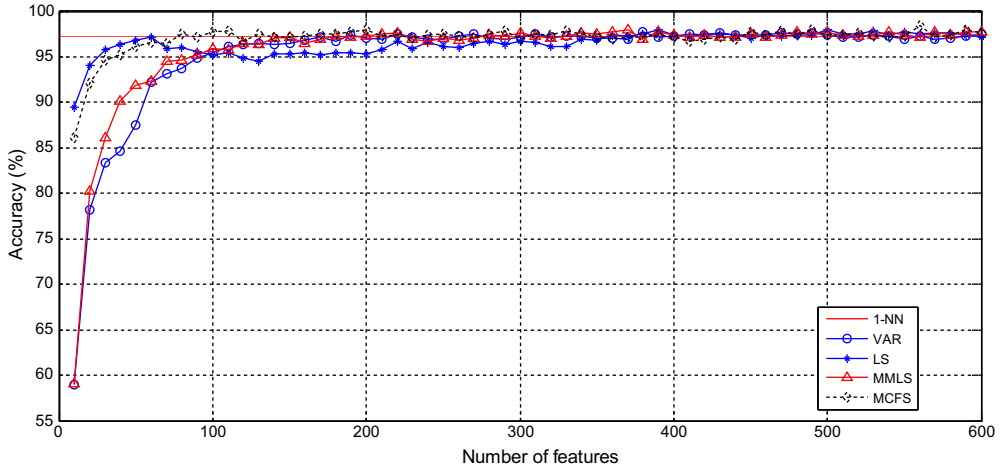


Fig. 2. Classification accuracy vs. the number of selected features on MFD.

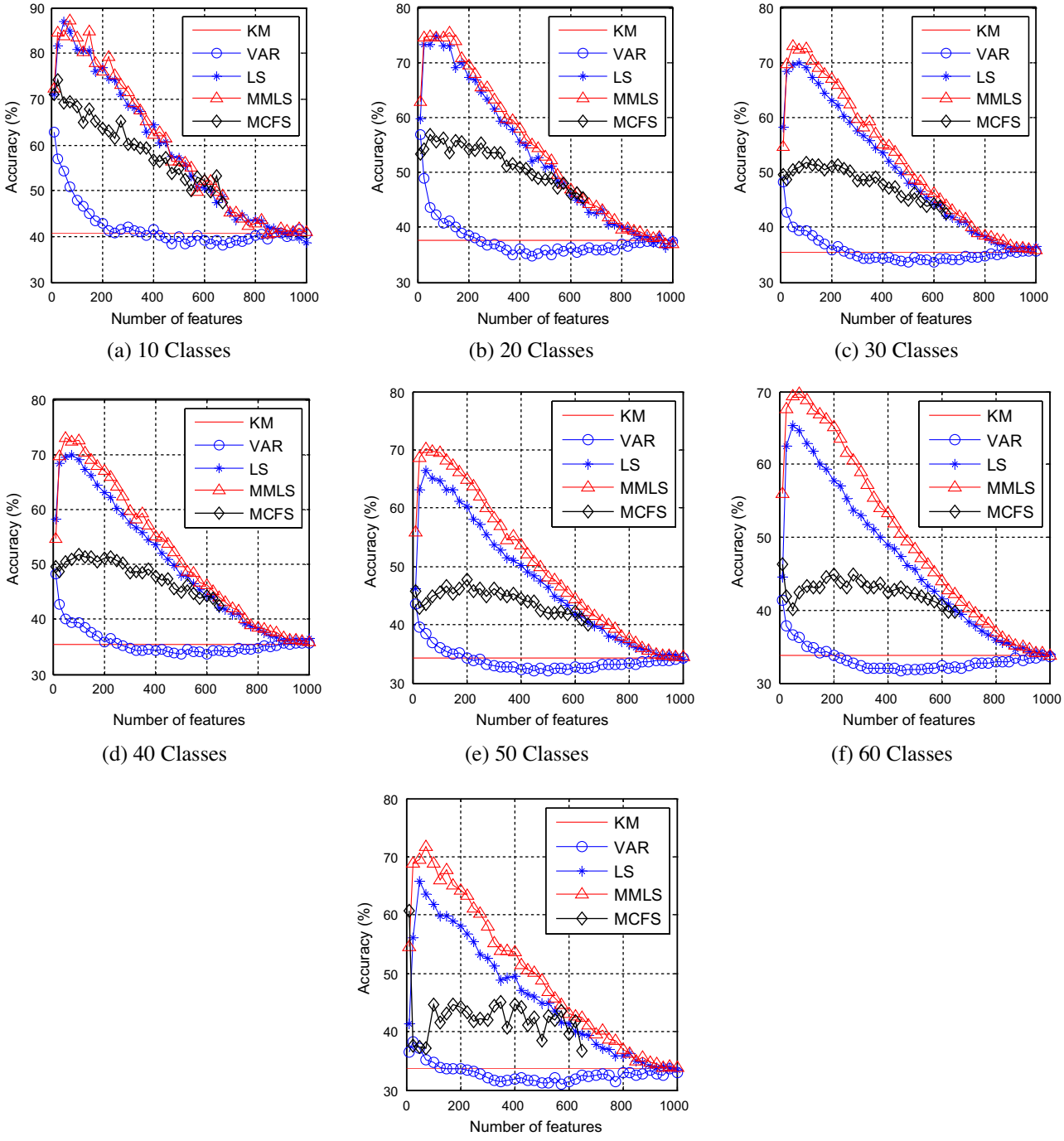
4.1. Data preparation

Three data sets are used in the experiments. The first data set is obtained from the COIL20 image library[2] from the Columbia University, which contains 1440 images generated from 20 objects. Each image is represented by a 1024-dimensional vector, and the size is 32×32 pixels with 256 grey levels per pixel.

The second dataset is the multiple feature dataset (MFD) (Frank and Asuncion, 2010), consisting of features of handwritten digits ('0' to '9') extracted from a collection of Dutch utility maps. 200 patterns per class (for a total of 2000 patterns) have been digitized into binary images. Each digit is represented by a 649-dimensional vector in terms of six feature sets: Fourier coefficients of the character shapes, profile correlations, Karhunen–Love coefficients, pixel averages in 2×3 windows, Zernike moments and morphological features. The above two datasets are used in the classification experiment.

The third dataset is obtained from the PIE face database of the Carnegie Mellon University (CMU) (downloadable from http://www.cad.zju.edu.cn/home/dengcai/). The face images are created under different poses, illuminations and expressions. The database contains 41,368 images of 68 subjects. The image size is 32×32 pixels, with 256 grey levels. We fixed the pose and expression, then a total of 1428 images under different illumination conditions are selected for the clustering experiment. In the following sections,

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



(a) 10 Classes (b) 20 Classes (c) 30 Classes

(d) 40 Classes (e) 50 Classes (f) 60 Classes

(g) 68 Classes

Fig. 3. Clustering accuracy vs. the number of selected features on PIE.

the classification experiment is first discussed, followed by the clustering experiment.

4.2. Classification experiment

In this experiment, we employ the nearest neighbor classifier to evaluate the discriminating power of the features chosen by the proposed MMLS algorithm, LS, VAR and MCFS respectively. The training data, half of the size of the data set, is obtained by randomly picking samples. The remaining samples are used for the testing data set. We performed the test with different numbers of features, starting from 10 to 400 on COIL20 (from 10 to 600 on MFD), at a step of 10. At each step, the test is randomly executed for 10 times to evaluate the average performance. Here, the performance metric used is the classification accuracy, which is defined as

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\delta(l(\boldsymbol{x}_i), l(N(\boldsymbol{x}_i))),$$

where $\boldsymbol{x}_i$ is the test sample, $l(\boldsymbol{x}_i)$ is its true class label, $l(N(\boldsymbol{x}_i)$ is the class label of the nearest neighbor in the training set, n is the size of the testing dataset, and the function $\delta(l(\boldsymbol{x}_i),l(N(\boldsymbol{x}_i))$ equals 1 if

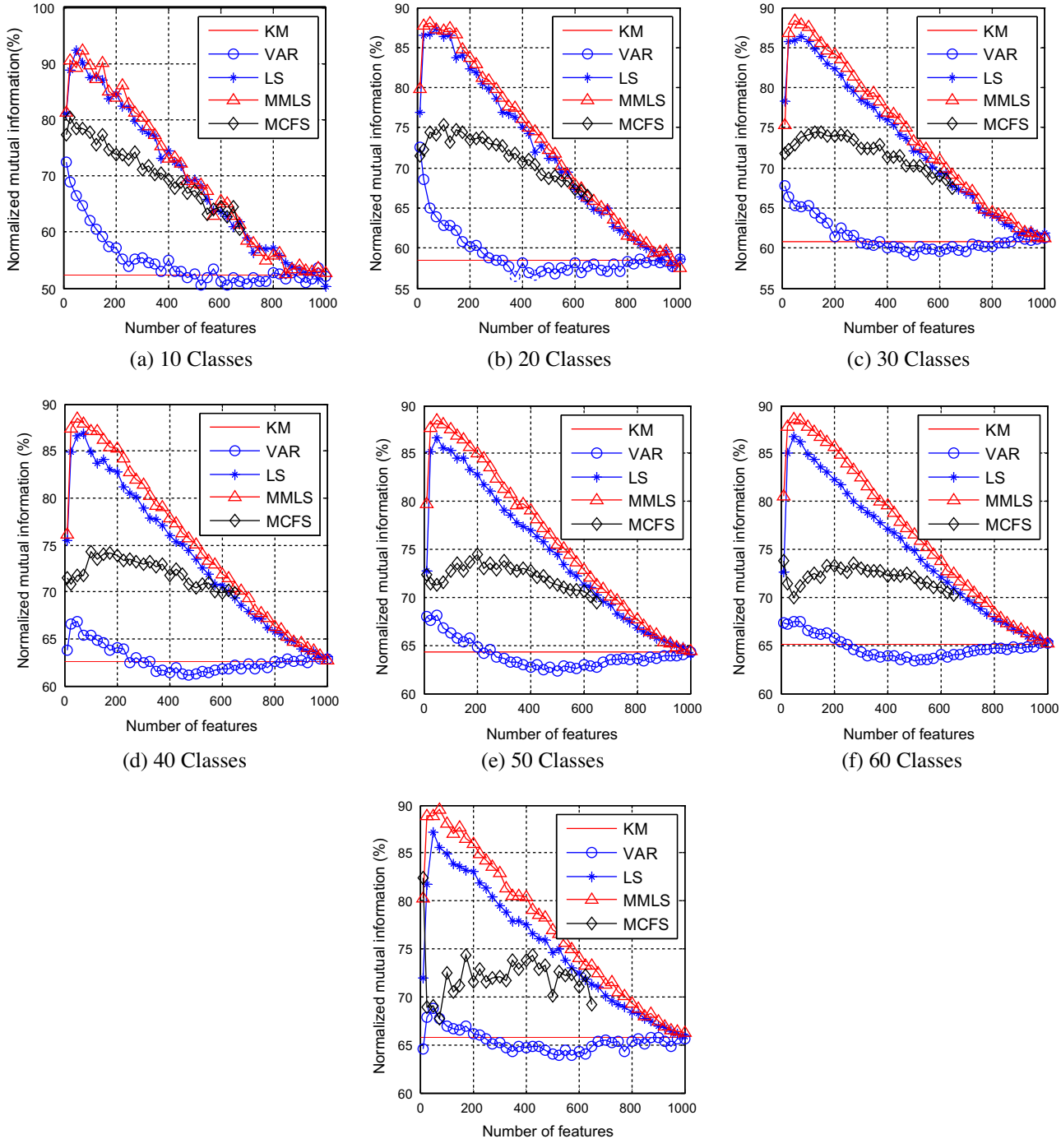


Fig. 4. Normalized mutual information vs. the number of selected features on PIE.

$l(\boldsymbol{x}_i) = l(N(\boldsymbol{x}_i)$ and 0 otherwise. The higher the classification accuracy, the better the selected features.

Fig. 1 shows the classification results on the COIL20 dataset. It can be seen that the proposed MMLS algorithm outperforms the LS and VAR methods. The classification performance of LS is the worst. MCFS outperforms the proposed MMLS under 110 features, while MMLS outperforms it over 110 features. Here, it is worthy of note that MCFS uses the information of the class number of the dataset. Particularly, MMLS can achieve 95% classification accuracy by using only 50 features and fastest converges to the best results. With a mere of 140 features, MMLS is able to achieve the same classification accuracy as the 1-nearest neighbor method (1-NN) which requires all the features of the data set.

Fig. 2 shows the classification accuracy on MFD. As can be seen, MCFS obtains the best performance and MMLS stably converges to the best results. LS achieves 97% classification accuracy with a mere of 60 features, but its accuracy vacillates when using more features. On this dataset, MMLS is better than LS and VAR. With only 140 features, MMLS achieves comparable results with those using all the features.

4.3. Clustering experiment

To evaluate the clustering performance of the MMLS algorithm, the PIE face data set of images created under different illumination conditions are used in this experiment. The performance metrics are the clustering accuracy and normalized mutual information (NMI). Details about these two metrics can be found in (He et al., 2005; Cai et al., 2005). The K-means clustering method (KM) is employed to perform clustering on the features obtained from the proposed MMLS, LS, VAR and MCFS methods.

The experiment is conducted repeatedly with different number of clusters, with K = 10,20,...,60,68, where $K = 68$ refers to the use of the entire data set. For a given value of K, we randomly select K classes from the data set and the selection process is repeated 20 times, except for the case when $K = 68$ (i.e., no selection is needed). K-means clustering method is then executed at each value of K for 10 times with different initialization settings. That is, for a given cluster number (except at $K = 68$), the test is run for a total of 200 times (10 times when $K = 68$). Meanwhile, we also conduct the experiments with different feature number according to the grid {10,25,50,75,...,1000}. Figs. 3 and 4 show the clustering performance in terms of the accuracy and the NMI, respectively. Note that in these two figures, all features are used by KM. In these experiments, MCFS has to terminate because of its too long running time when the chosen features are over 650. The computational complexity of MCFS is $O(m^2D + md + mKd^2 + Kd^3)$ (Cai et al., 2010), and then it takes too much time when the number of the chosen features d is comparatively big.

These experiments reveal several interesting findings:

(1) The clustering performance varies with the number of features. Both MMLS and LS achieve the best performance at very low dimensionality (in the range of 50 and 125). This indicates that the face images of the PIE dataset contain many irrelevant features.
(2) Both MMLS and LS consider the geometrical structure of the data set and achieve better performance than the other algorithms. This implies that the geometrical structure cannot be ignored in the machine learning methods.
(3) Among the five comparison algorithms, MMLS gives the best performance. It is able to discover the most discriminative features, even for unsupervised problems. The finding demonstrates that the strategy to maximize the information of the between-locality structure while minimizing that of the within-locality structure is effective for feature selection.
(4) In particular, although the MMLS and LS algorithms both utilize the geometrical structure of the dataset for feature extraction, MMLS considerably outperforms LS when the cluster number is large. This shows that the information of the between-locality structure plays a significant role in the identification of information of different clusters.

5. Conclusion and future work

This paper proposes the novel feature selection algorithm minimum–maximum-information Laplacian Score by taking the local structure information of data set into consideration. The algorithm not only preserves the manifold structure of the "same" class data by minimizing the within-locality information, but also maximizes the information between the manifold structures of the "different" class data by maximizing the between-locality information. The experimental results show that MMLS is able to identify a subset of the original features with more discriminating power and that the classification and clustering performance of MMLS are superior to LS and VAR methods.

The MMLS algorithm is proposed to deal with unsupervised feature selection problems. By considering the between-locality information, MMLS applies the information among the manifolds of the "different" classes hidden in the data set. However, theoretical selection of the sizes of the k-nearest neighbors or the ε neighborhoods to match the local structure and obtain more information among the manifolds of the "different" classes remains an issue. Further investigation will be carried out to split the global graph into two or more subgraphs. Besides, it is also not clear about how to theoretically select the parameter α in MMLS that controls the tradeoff between the within-locality and the between-locality information. We are currently exploring these parameter identification problems which deserve further research effort.

Acknowledgments

This work was supported in part by the Hong Kong Polytechnic University under Grants A-PJ38, 1-ZV6C and 87RF; the National Natural Science Foundation of China under Grants 61170122, 61170029, 61272210, and 61202311; by the Natural Science Foundation of Jiangsu Province under Grant BK2011003 and BK2011417; by JiangSu 333 expert engineering Grant (BRA2011142); 2012 Postgraduate Student's Creative Research Fund of Jiangsu Province (CXZZ12_0759); and the Natural Science Foundation of Zhejiang Province under Grant LY12F03008.

References

Belkin, M., Niyogi, P., 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems, vol. 14, MIT Press, pp. 585–591.
Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press.
Cai, D., He, X., Han, J., 2005. Document clustering using locality preserving indexing. IEEE Trans. Pattern Anal. Machine Intell. 17 (12), 1624–1637.
Cai, D., Zhang, C., He, X., 2010. Unsupervised feature selection for multi-cluster data. In: Proc. Sixth ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD'10), July 2010, pp. 333–342.
Deng, Z., Chung, F., Wang, S., 2010. Robust relief-feature weighting, margin maximization, and fuzzy optimization. IEEE Trans. Fuzzy Systems 18 (4), 726–744.
Fan, R.K., 1997. Chung, Spectral Graph Theory (CBMS Regional Conference Series in Mathematics). American Mathematical Society, New York.
Frank, A., Asuncion, A., 2010. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, School of Information and Computer Science, Irvine, CA.
He, X., Niyogi, P., 2003. Locality preserving projections. In: Proc. Conf. on Advances in Neural Information Processing Systems (NIPS), MIT Press, pp. 153–160.
He, X., Cai, D., Niyogi, P., 2005. Laplacian score for feature selection. In: Advances in Neural Information Processing Systems, vol. 18, MIT Press, pp. 507–514.
He, X., Cai, D., Yan, S., Zhang, H., 2005. Neighborhood preserving embedding. In: Proc. Internat. Conf. on Computer Vision (ICCV).
Kira, K., Rendell, L.A., 1992a. The feature selection problem: Traditional methods and new algorithm. In: Proc. Tenth National Conf. on Artificial intelligence, AAAI Press, pp. 129–134.
Kira, K., Rendell, L.A., 1992b. A practical approach to feature selection. In: Sleeman, D., Edwards, P. (Eds.), Proceedings of the 344 Tenth National Conference on Artificial intelligence. Morgan Kaufmann, pp. 249–256.
Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artificial Intell. 97 (1–2), 273–324.
Kononenko, I., 1994. Estimating attributes: Analysis and extensions of relief. In: Raedt, L.D., Bergadano, F. (Eds.), Proceedings of the European conference on machine learning on Machine Learning. Springer Verlag, pp. 171–182.
Lee, C., Landgrebe, D.A., 1993. Feature extraction based on decision boundaries. IEEE Trans. Pattern Anal. Machine Intell. 15 (4), 388–400.
Liang, J., Yang, S., Wang, Y., 2009. An optimal feature subset selection method based on distance discriminant and distribution overlapping. Int. J. Pattern Recognition Artificial Intell. 23 (8), 1577–1597.
Roweis, S., Saul, L., 2000. Nonlinear dimensionality reduction by locally linear embedding. Science 290 (5500), 2323–2326.
Seung, H.S., Lee, D.D., 2000. The manifold ways of perception. Science 290 (12), 2268–2269.
Sikonja, M.R., Kononenko, I., 1997. An adaptation of relief for attribute estimation in regression. In: Fisher, D.H. (Ed.), Proceedings of the Fourteenth International Conference on Machine Learning. Morgan Kaufmann, pp. 296–304.
Sikonja, M.R., Kononenko, I., 2003. Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learn. 53 (1–2), 23–69.

Sun, Y., 2007. Iterative RELIEF for feature weighting: Algorithms, theories, and applications. IEEE Trans. Pattern Anal. Machine Intell. 29 (6), 1035–1051.
Sun, Y., Todorovic, S., Goodison, S., 2010. Local-learning-based feature selection for high-dimensional data analysis. IEEE Trans. Pattern Anal. Machine Intell. 32 (9), 1610–1626.
Tenenbaum, J., de Silva, V., Langford, J., 2000. A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500), 2319–2323.
Zhao, J., Lu, K., He, X., 2008. Locality sensitive semi-supervised feature selection. Neurocomputing 71 (10–12), 1842–1849.