Pattern Recognition 46 (2013) 2798–2806

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Large Margin Subspace Learning for feature selection

Bo Liu [a,b,*], Bin Fang [a], Xinwang Liu [c], Jie Chen [d], Zhenghong Huang [b], Xiping He [b]

[a] College of Computer Science, Chongqing University, Chongqing 400044, PR China
[b] School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing, PR China
[c] School of Computer Science, National University of Defense Technology, Changsha, Hunan 410073, PR China
[d] Institut Charles Delaunay, Université de Technologie de Troyes, France

ARTICLE INFO

Article history:
Received 2 August 2012
Received in revised form
5 December 2012
Accepted 21 February 2013
Available online 7 March 2013

Keywords:
Feature selection
$\ell_{2,1}$-norm regularization
Large margin maximization
Subspace learning

ABSTRACT

Recent research has shown the benefits of large margin framework for feature selection. In this paper, we propose a novel feature selection algorithm, termed as Large Margin Subspace Learning (LMSL), which seeks a projection matrix to maximize the margin of a given sample, defined as the distance between the nearest missing (the nearest neighbor with the different label) and the nearest hit (the nearest neighbor with the same label) of the given sample. Instead of calculating the nearest neighbor of the given sample directly, we treat each sample with different (same) labels with the given sample as a potential nearest missing (hint), with the probability estimated by kernel density estimation. By this way, the nearest missing (hint) is calculated as an expectation of all different (same) class samples. In order to perform feature selection, an $\ell_{2,1}$-norm is imposed on the projection matrix to enforce row-sparsity. An efficient algorithm is then proposed to solve the resultant optimization problem. Comprehensive experiments are conducted to compare the performance of the proposed algorithm with the other five state-of-the-art algorithms RFS, SPFS, mRMR, TR and LLFS, it achieves better performance than the former four. Compared with the algorithm LLFS, the proposed algorithm has a competitive performance with however a significantly faster computational.

1. Introduction

Feature selection is to choose a subset of the original features according to some selection algorithm, it has a wide range of applications, including bioinformatics [1], object detection [2], computer vision [3]. It aims to reduce the data dimensionality by removing the redundancy and the correlation of extracted features. It has been proven in both theory and practice effective in enhancing learning efficiency, increasing predictive accuracy, and reducing complexity of learned results, due to the fact that accuracy of most classification algorithms, such as SVM, can be affected notably when applying on tasks with a small number of training data or with high-dimensional inputs [4,5]. Feature selection for high-dimensional data is one of the most important topics in machine learning research. Recently feature selection based on large margin has been widely investigated [6,7]. Feature selection based on subspace learning algorithm for high dimensional data was also proposed [8], and achieved good results. However, research which combines subspace learning with large margin has not appeared. The main contributions of this paper include:

- An efficient and novel feature selection algorithm termed as Large Margin Subspace Learning (LMSL) is proposed. Different from traditional feature selection algorithms, LMSL is a subspace learning algorithm based on large margin framework. Firstly, we utilize the expectation to estimate the nearest missing (the nearest neighbor with the different label) and the nearest hit (the nearest neighbor with the same label) of the given sample. Then these nearest neighbors are projected into the subspace $\boldsymbol{W}$ ($\in \mathbb{R}^{d\times p}, d \gg p$, and to be described in Section 3.2). After that, we define a novel metric function based on large margin in the subspace. The objective function is formulated by the metric function. To obtain row-sparsity of the solution, an $\ell_{2,1}$-norm regularization is incorporated into the objective function.
- As the proposed objective function and constraints are nonconvex. In general, a global optimal solution is hard to be obtained. We propose an efficient algorithm that obtain suboptimal solution and give detailed description of the algorithm.
- Extensive experiments are conducted to evaluate the proposed algorithm. Experimental results confirm the efficiency of our algorithm compared with different types of feature selection algorithms.

* Corresponding author at: College of Computer Science, Chongqing University, Chongqing 400044, PR China. Tel.: +86 13996283578.
E-mail addresses: flyinsky723@gmail.com (B. Liu), fb@cqu.edu.cn (B. Fang).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.02.012

The rest of this paper is organized as follows: Section 2 reviews prior work on the feature selection. In Section 3, we describe a basic introduction on large margin theory and a novel feature selection algorithm. Experimental evaluation is reported and discussed in Section 4. Finally, we conclude the paper and give a perspective of future work in Section 5.

2. Related work

According to the way of utilizing label information, feature selection algorithms can be divided into supervised [9], unsupervised [10–12] two classes. The first class includes Fisher score, ReliefF [13], SVM-RFE [14], etc. From the perspective of the selection strategy, feature selection algorithms are divided into three categories [15]: filter, wrapper and embedded algorithms. The filter algorithms compute some score of a selected feature subset by information of each feature, the algorithms are computationally much cheaper and more efficient, such as: Fisher Score [16], Laplacian Score [17], Trace Ratio [18]. These algorithms can be encompassed by algorithm named SPFS (Similarity Preserving Feature Selection) [15]. Wrapper algorithms use a procedure that wraps around a learning algorithm, and repeatedly calls the learning algorithm to evaluate how well it does using different feature subsets. The wrapper algorithm was firstly proposed in [19]. One serious problem with these wrapper algorithms is their high computational complexity because they need to train a large number of classifiers. To alleviate this problem, forward and backward selection were proposed in [20]. The algorithms are heuristic algorithms, none of them guarantee the optimal solution.

Feature selection of embedded algorithms was recently emerged in [4,21]. For large-scale feature selection problems, the novel embedded algorithms were proposed by performing feature selection directly in the SVM formulation in [4,5]. For microarray data, the embedded algorithm named RFE was proposed in [14]. In this algorithm, an SVM classifier was iteratively trained with the set of features, and those with small weights were then removed from the set. In order to obtain the sparse solution and to improve the computability, ℓ_1-SVM with a linear kernel was adopted in [22]. To jointly perform feature selection and SVM parameter learning for linear and nonlinear kernels, authors in [23] proposed a convex framework with ℓ_1-SVM. However, for an arbitrarily complex nonlinear problem, these algorithms are still no better performance. In [6], the authors proposed a notable algorithm. The main idea of the algorithm is to decompose complex nonlinear problem into a set of locally linear through local learning, and then to learn feature selection in the margin theory. To establish margin-based error function in weighted feature space, the benefits of the introduction of the Expectation–Maximization algorithm are to solve the nearest neighbor of a given sample, which is unknown before learning. To improve the performance of feature selection, the large margin principle in [6] is adopted into our paper.

Recently, the problem of subspace learning has received a lot of interests in dimensionality reduction and feature selection for high-dimensional data. Popular dimensionality reduction algorithms include principal component analysis (PCA) [24], linear discriminant analysis (LDA) [16], locality preserving projection (LPP) [25], neighborhood preserving embedding (NPE) [26], graph optimization for dimensionality reduction with sparsity constraints (GODRSC) [27]. These algorithms can be interpreted in a unified graph embedding framework based on Laplacian matrix in [28]. A framework for joint feature selection and subspace learning was presented in [8], where authors reformulated the subspace learning problem and used $\ell_{2,1}$-norm on the projection matrix to obtain row-sparsity of the solution, this enabled to select relevant features and learn transformation simultaneously.

Feature selection is also closely related to distance metric learning. In [29], large margin component analysis (LMCA) for the low-dimensional projection of the inputs was proposed. The algorithm aimed at separating points in different classes by a large margin. Authors in [30] use the maximum margin score for discriminatively optimizing the structure of Bayesian network classifiers. For k-nearest neighbor (k-NN) classification from labeled samples, a Mahanalobis distance metric was learned by semidefinite programming in [31]. The metric was trained with the goal that the k-nearest neighbors belong to the same class while examples from different classes were separated by a large margin. In [7], authors introduced a margin based on feature selection criterion and applied it to measure the quality of sets of features. Experiments showed that the algorithms based on the large margin were effectiveness for feature selection.

In order to obtain the sparse solution, regularization was introduced into most algorithms previously mentioned. In [32,33], feature selection regularized by ℓ_1-norm showed interesting performance. However, due to the non-differentiability of ℓ_1-norm, the regularized problem was solved using sub-gradient method, which was complex and inefficient. In [34], a so-called Hybrid Huberized SVM (HHSVM) algorithm was proposed by compositing ℓ_1 and ℓ_2 norms ($\ell_{2,1}$-norm). Instead of individually using one of these two norms, this composite regularization had more favorable properties as it investigated the structure of the problem. This type of regularization was also introduced into the multi-task feature selection [35]. In [36], the Nesterov method was used to optimize the objective function with $\ell_{2,1}$-norm regularization, and an Euclidean space projection algorithm with a linear time complexity was proposed to improve the computational efficiency of the Nesterov method. In [15], the $\ell_{2,1}$-norm regularized objective function is formulated and the Nesterov method in [36] was used to solve the objective function. In [1], another efficient algorithm for the $\ell_{2,1}$-norm regularization was proposed. This algorithm did not require the gradient of the objective function and experiments showed its efficiency and fast convergence rate. Feature selection algorithm based on subspace learning and $\ell_{2,1}$-norm was proposed in [8], and the algorithm in [1] was used to solve the objective function.

3. The proposed algorithm

In this section, we will propose an algorithm termed as Large Margin Subspace Learning (LMSL). This algorithm considers the large margin of samples. It is more suitable to feature selection for high dimensional data.

In what follows, we will firstly introduce margin-based metric function. Then the motivation and theoretical basis of LMSL will be proposed. After that the objective function will be formulated and the solution method will be proposed. Finally, the implementation and analysis of the algorithm will be discussed.

3.1. Margin-based metric function

The margin plays a crucial role in current machine learning research. The basic idea of marginal feature selection is to measure the importance of features by the margin of samples.

Let $A=[x_1,\ldots,x_n]^T\in\mathbb{R}^{n\times d}$ be a training data set, where x_n is the nth data sample containing d features, $Y=[y_1,\ldots,y_n]$ is the corresponding class labels, and $d\gg n$. The margin of x_j was defined in [7] as the following:

Definition 1. Let A be a training data set, x_j be a sample from A, and w be a weight vector, then the margin of x_j about w is

$$\rho_j(w)=\sum_{k\in M_j} d_w(x_j,x_k)-\sum_{k\in H_j} d_w(x_j,x_k), \tag{1}$$

where $d_w(\cdot,\cdot)$ is a distance function about w, $H_j=\{i|1\le i\le n, y_i=y_j, i\ne j\}$ and $M_j=\{i|1\le i\le n, y_i\ne y_j\}$.

A variety of distances $d_w(\cdot,\cdot)$ have been defined in some references. The Euclidean distance in [7] defined the distance as the following:

$$d_w(x,y)=\tfrac{1}{2}\|x-y\|_w, \tag{2}$$

with $\|z\|_w=\sqrt{\sum_i(z_i^2w_i^2)}$.

In [6], the authors developed the expectation to obtain the nearest neighbors for a given sample with the same class and different classes. Two types of nearest neighbors for the sample x_j are defined in Eqs. (3) and (4), one with the same class (called nearest hit) and the other with the different class (called nearest miss):

$$NH(x_j)=d(x_j,NH(x_j))=\sum_{i\in H_j}P(x_i=NH(x_j))\|x_j-x_i\|, \tag{3}$$

and

$$NM(x_j)=d(x_j,NM(x_j))=\sum_{i\in M_j}P(x_i=NM(x_j))\|x_j-x_i\|, \tag{4}$$

where $NM(x_j)$ and $NH(x_j)$ denote the expectation computed with respect to H_j and M_j, respectively. $P(x_i=NH(x_j))$ and $P(x_i=NM(x_j))$ are the probabilities of sample x_i being the nearest hit or miss of x_j, respectively. These probabilities are estimated by the standard kernel density $k(d)=\exp(-d/\sigma)$:

$$P(x_i=NM(x_j))=\frac{k(\|x_j-x_i\|)}{\sum_{l\in M_j}k(\|x_j-x_l\|)}, \tag{5}$$

and

$$P(x_i=NH(x_j))=\frac{k(\|x_j-x_i\|)}{\sum_{l\in H_j}k(\|x_j-x_l\|)}. \tag{6}$$

This margin is defined by

$$\begin{aligned}\rho_j(w)&=d(x_j,NH(x_j))-d(x_j,NM(x_j))\\&=\left(\sum_{i\in M_j}P(x_i=NM(x_j))\|x_j-x_i\|\right)\\&\quad-\left(\sum_{i\in H_j}P(x_i=NH(x_j))\|x_j-x_i\|\right).\end{aligned} \tag{7}$$

In the sections that follow, we will define a new margin in subspace.

3.2. Basic notation for LMSL

The main step of classic subspace learning algorithms (e.g. LPP, LLE [37]) is to incorporate neighborhood information which is often represented by k nearest neighbors or Laplacian graph of the data set and compute a transformation matrix which maps the data points to projection subspace. This linear transformation optimally preserves local neighborhood information, which may not be true in the presence of copious irrelevant features [6]. We propose an algorithm to overcome the weakness of these subspace learning algorithms. This algorithm differs from the previously mentioned algorithm in two aspects. Firstly, the nearest neighbors are computed by expectation. Secondly, the margin between nearest hit and nearest miss in the projection subspace is optimally retained.

Let $\boldsymbol{W}$ ($\in\mathbb{R}^{d\times p}, d\gg p$) be a projection subspace (or matrix), where d is the number of features, p is the dimension of the subspace. $NM_W(x_j)$ and $NH_W(x_j)$ denote the expectation in the subspace $\boldsymbol{W}$. Distance of x_j to $NM_W(x_j)$ and $NH_W(x_j)$ in the subspace $\boldsymbol{W}$ can be written as the following:

$$\begin{aligned}d_W(x_j,NM_W(x_j))&=\|(x_j-NM_W(x_j))^T\boldsymbol{W}\|_2\\&=(x_j-NM_W(x_j))^T\boldsymbol{W}\boldsymbol{W}^T(x_j-NM_W(x_j)),\end{aligned} \tag{8}$$

and

$$\begin{aligned}d_W(x_j,NH_W(x_j))&=\|(x_j-NH_W(x_j))^TW\|_2\\&=(x_j-NH_W(x_j))^T\boldsymbol{W}\boldsymbol{W}^T(x_j-NH_W(x_j)),\end{aligned} \tag{9}$$

where

$$NH_W(x_j)=\sum_{i\in H_j}P_W(x_i=NH_W(x_j))\|x_j-x_i\|, \tag{10}$$

and

$$NM_W(x_j)=\sum_{i\in M_j}P_W(x_i=NM_W(x_j))\|x_j-x_i\| \tag{11}$$

the probability $P_W(\cdot)$ can be estimated via the standard kernel density $k(d)=\exp(-d/\sigma)$:

$$P_W(x_i=NM_W(x_j))=\frac{k(\|\boldsymbol{W}(x_j-x_i)\|)}{\sum_{l\in M_j}k(\boldsymbol{W}(\|x_j-x_l\|))}, \tag{12}$$

and

$$P_W(x_i=NH_W(x_j))=\frac{k(\boldsymbol{W}(\|x_j-x_i\|))}{\sum_{l\in H_j}k(\boldsymbol{W}(\|x_j-x_l\|))}. \tag{13}$$

The margin of x_j in the subspace $\boldsymbol{W}$ is defined as

$$\rho_j(W)=d_W(x_j,NM_W(x_j))-d_W(x_j,NH_W(x_j)). \tag{14}$$

3.3. Objective function

Our aim is to search a matrix $\boldsymbol{W}$ so that the margins are maximized when the nearest neighbors are projected into it.

Before defining objective function, we define two column vectors $h_j\in\mathbb{R}^n$ and $m_j\in\mathbb{R}^n$ (n is the number of samples) as the following:

$$h_{ji}=\begin{cases}P_W(x_i=NH_W(x_j)), & i\in H_j\\0 & \text{otherwise}\end{cases} \tag{15}$$

and

$$m_{ji}=\begin{cases}P_W(x_i=NM_W(x_j)), & i\in M_j\\0 & \text{otherwise}\end{cases} \tag{16}$$

Then the margin $\rho_j(\boldsymbol{W})$ can be rewritten as

$$\begin{aligned}\rho_j(\boldsymbol{W})&=(e_j-m_j)^TA\boldsymbol{W}\boldsymbol{W}^TA^T(e_j-m_j)\\&\quad-(e_j-h_j)^TA\boldsymbol{W}\boldsymbol{W}^TA^T(e_j-h_j)\\&=\mathrm{tr}(\boldsymbol{W}^TA^TD_jA\boldsymbol{W}),\end{aligned} \tag{17}$$

where e_j is a column vector with j-th element equal to one and others equal to zeros and

$$D_j=(e_j-m_j)(e_j-m_j)^T-(e_j-h_j)(e_j-h_j)^T. \tag{18}$$

An $\ell_{2,1}$-norm regularization term is added into the objective function to encourage row-sparsity of the solution. Our objective function is

$$\begin{aligned}\min_{\boldsymbol{W}}F(\boldsymbol{W})&=-\sum_{j=1}^{n}\rho_j(\boldsymbol{W})+\lambda\|\boldsymbol{W}\|_{2,1}\\&=\mathrm{tr}(\boldsymbol{W}^TA^TDA\boldsymbol{W})+\lambda\|\boldsymbol{W}\|_{2,1}\\&\text{s.t. }\boldsymbol{W}^TA^TA\boldsymbol{W}=I,\end{aligned} \tag{19}$$

where $D=-\sum_{j=1}^{n}D_j$, $D\in\mathbb{R}^{n\times n}$, λ is the regularization parameter. Obviously, D_j in Eq. (18) is indefinite matrix, D is also indefinite

matrix, so $\mathrm{tr}(\boldsymbol{W}^T A^T D A \boldsymbol{W})$ is nonconvex function with respect to $\boldsymbol{W}$, and the objection function $F(\boldsymbol{W})$ is nonconvex.

The introduction of $\ell_{2,1}$-norm is based on the following two reasons:

- The $\ell_{2,1}$-norm can guarantee row-sparsity of the projection matrix $\boldsymbol{W}$ (elements in a row are all zero). Hence it is able to discard the irrelevant features and transform the relevant ones simultaneously [8].
- In [1], authors proposed a simple and efficient algorithm for $\ell_{2,1}$-norm regularization. This algorithm did not require the gradient of the objective function, and was proven convergence in theory. Hence our objective function can be efficiently solved by this algorithm.

Two reasons for the introduction of constraint $\boldsymbol{W}^T A^T A \boldsymbol{W} = \mathrm{I}$ are:

- To avoid a trivial solution of the objective function.
- As the objective function is nonconvex, it is difficult to obtain an optimal solution. The introduction of constraint enables us to obtain a sub-optimal solution of the objective function, and we will show in Section 3.4 that the objective function can be solved by using a simple and efficient algorithm.

3.4. An efficient algorithm to solve the objective function with constraints

In this section, we discuss how to solve the optimization problem equation (19). Solving process consists of two steps.

Firstly, let $A\boldsymbol{W} = \boldsymbol{U}$, then we can solve the following optimization problem:

$$\min_{\boldsymbol{U}} \quad \mathrm{tr}(\boldsymbol{U}^T D \boldsymbol{U}),$$

$$\text{s.t.} \quad \boldsymbol{U}^T \boldsymbol{U} = I, \tag{20}$$

where D is the symmetric matrix, and I is the identity matrix of the size $p \times p$. The optimal solution of Eq. (20) is

$$\boldsymbol{U}^* = \Gamma_r, \tag{21}$$

where the columns of the matrix Γ_r are r eigenvectors of D corresponding to the last r smallest eigenvalues [38].

Secondly, in order to obtain the optimal solution of the objective function, we need to solve the equation $A\boldsymbol{W} = \boldsymbol{U}^*$. Note that $A\boldsymbol{W} = \boldsymbol{U}^*$ is linear equation, there are three possibilities for the solution of the equation in [8]:

(1) The linear equation has infinitely many solutions.
(2) The linear equation has unique solution.
(3) The linear equation has no solution.

One situation in (1) which is most common is only discussed in our paper. Discussion on other situations can be found in [8].

When linear equation about $A\boldsymbol{W} = \boldsymbol{U}^*$ has infinitely many solutions, we can solve the following optimal problem to obtain optimal solution for Eq. (19):

$$\min_{\boldsymbol{W}} \quad \|\boldsymbol{W}\|_{2,1},$$

$$\text{s.t.} \quad A\boldsymbol{W} = \boldsymbol{U}^*. \tag{22}$$

The solution of optimization problem in Eq. (22) is given in [1].

3.5. LMSL algorithm

We first need to compute matrix D of Eq. (19) for LMSL. The algorithm to compute matrix D is listed in Algorithm 1. $A \in \mathbb{R}^{n \times d}$ is the training samples, $Y \in \mathbb{R}^n$ is the sample labels, and $\boldsymbol{W}_0 \in \mathbb{R}^{d \times p}$ whose elements are the initialization of the one is projection matrix. Kernel width parameter σ is determined by 5-fold cross-validation.

Algorithm 1. Compute D of Eq. (19).

Initialization:
A is the training samples, Y is the sample labels, W_0 is the projection matrix and σ is the kernel width parameter.
Output: D
1: Set $D=0$.
2: **for** $i=1$ to n **do**
3: Calculate $P_W(x_i = NM_W(x_j))$ and $P_W(x_i = NH_W(x_j))$ by Eqs. (12) and (13), respectively.
4: Calculate h_j and m_j by Eqs. (15) and (16), respectively.
5: $D = D - D_j$
6: **end for**

LMSL is listed in Algorithm 2. To be simple, p is the number class of samples. Regularization parameter λ is determined by 5-fold cross-validation. LMSL is convergence because the algorithm to solve Eq. (22) has proven to be convergence in [1].

Algorithm 2. Large Margin Subspace Learning for Feature Selection (LMSL).

Initialization:
A is the training samples, Y is the sample labels, W_0 is the projection matrix, σ is the kernel width parameter and λ is the regularization parameter.
Output: $\boldsymbol{W}$
1: Calculate matrix D of Eq. (19) by Algorithm 1.
2: Calculate eigenvector and eigenvalue of D.
3: Obtain Γ_r whose columns are r eigenvectors of D corresponding to the r smallest eigenvalues
4: Use algorithm in [1] to compute $\boldsymbol{W}$ by A, Γ_r and λ
5: return $\boldsymbol{W}$

Algorithm 2 consists of three major steps:

(1) Calculate margins of samples via Algorithm 1.
(2) Compute the eigenvalues of the matrix D in Eq. (19). The computational time is negligible as it is executed only once and the size of D is usually not large in the context under discussion.
(3) To solve Eq. (22), we use iterative algorithm in [1]. The convergence property of Eq. (22) has been theoretically proven in [1] and fast convergence rate has also been experimentally verified.

It should be noticed that Algorithm 2 achieves a suboptimal solution of Eq. (19), as its feasible region is limited to the solution space of Eq. (22), which is smaller than the original one. However, experiments have shown that its performance is favored even with a suboptimal solution.

4. Experiment study

In this section, several experiments are conducted to illustrate the effectiveness of our algorithm for feature selection.

Table 1
Summary of the benchmark data sets.

Data set	# features	# instance	# classes
ARP	2400	130	10
PIE	2420	210	10
LUNG	3312	203	5
CARC	9182	174	11
PROS	5966	102	2
TOX	5748	171	4
ORL	10 304	100	10
CLLS	11 340	111	3
COIL	16 384	350	10

The proposed LMSL algorithm is applied on nine high-dimensional data sets, including five image data sets: AR10P(ARP),[1] PIE10P(PIE) (see footnote 1), ORLRAWS10P(ORL) (see footnote 1), TOX-101(TOX) (see footnote 1), COIL,[2] and four micro-array data sets: LUNG,[3] CARCINOM(CARC) (see footnote 1), CLL-SUB-101(CLLS) (see footnote 3), PROSTATE_GE(PROS) (see footnote 3). Detailed information of these data sets is listed in Table 1.

The five typical feature selection algorithms are chosen for the comparison purpose as the following:

(1) RFS (Robust Feature Selection) [1].
(2) LLFS (Local Linear Feature Selection) [6].
(3) SPFS (Similarity Preserving Feature Selection) [15].
(4) mRMR (minimum Redundancy Maximum Relevance) [39].
(5) TR (Trace-Ratio) [18].

The first two algorithms are closely related with the proposed LMSL, as LMSL and LLFS both use the large margin principle, and LMSL benefits from RFS for solving the objective functions efficiently. The later ones are three state-of-the-art feature selection algorithms with the following characters: mRMR removes redundant features via considering pairwise feature correlation measurement. SPFS handles feature redundancy by similarity preserving. Both of these algorithms are in the class of filter-based model. TR represents data set structures by using the Laplacian graph, and it has a good performance compared with other similar algorithms (such as Laplacian Score [17]). The regularization parameter λ of the three algorithms (LMSL, LLFS, RFS) is determined by 5-fold cross validation. The kernel width σ of LMSL and LLFS is also tuned with 5-fold cross validation. For each data set, we randomly select 70% samples as the training data and use the rest as the test data. The above process is repeated 20 times in order to obtain the averaged performance. The linear SVM is then applied on the selected features in order to compare the classification accuracy.

4.1. Accuracy of classification

4.1.1. Average accuracy of classification

The classification accuracy results in Table 2 are obtained by SVM using the top 50,100,150, . . . ,1000 features selected for each algorithm. The boldfaced values are the highest ones or the ones without significant difference to the highest.

It can be observed in Table 2 that the proposed LMSL has an advantageous classification accuracy. It possesses the highest average accuracy on six test sets, and has only slight differences (about 1%) compared with the best ones when tested on the other four data sets. The last row shows the classification accuracy averaged on all these data sets. LMSL has a 0.5% higher classification accuracy than LLFS, which also utilizes the large margin principle. Compared with the other four algorithms, LMSL performs notably better than the other four algorithms, with 4–7% higher accuracy.

4.1.2. Classification accuracy of top n% features

Classification Accuracy of Top n% features is obtained by using the first n% selected features for SVM classification. In this experiment n% is set to 30% and 60%. Results are reported in Tables 3 and 4 show that the classification performance has the similar trend to the average accuracy in the precedent experiment.

4.2. Sensitivity of the regularization parameter λ

Fig. 1 shows that the insensitivity of LMSL with respect to the regularization parameter λ. Classification accuracy results in all data sets do not obviously change with different values of λ chosen from {0.001, 0.01, 0.1, 1, 10, 100, 1000}. But we have also noticed that the classification accuracy results of LLFS and RFS have relatively large changes in some data sets (e.g. ARP, CLLS, TOX). We conclude that the proposed algorithm is more suitable for a variety of applications.

4.3. Sensitivity of the kernel width σ

Fig. 2 shows that the insensitivity of LMSL and LLFS with respect to the kernel width σ. Classification accuracy of the test data sets does not obviously change with different values of σ chosen from {0.1, 0.5, 1,5, 10, 50, 100}. The insensitivity of the parameters favors the applications of the proposed algorithm. The insensitivity of kernel width σ for LLFS was also experimentally confirmed in [6].

4.4. Starting point $\boldsymbol{W_0}$ for LMSL

Different from the above experiments where W_0 in Algorithm 2 is initialized by the all-one matrix, in this experiment, we test the performance of the algorithm with different W_0. For each data set, LMSL is executed 100 times with randomly generated W_0 each time. One hundred results of corresponding classification accuracy are illustrated in Fig. 3. It can be observed that the accuracy values vary little with respect to different W_0. This shows the insensibility of LMSL with respect to the initialization W_0.

4.5. CPU time

LMSL is closely related with LLFS and RFS, Therefore, we compare the computational time of these three algorithms in this section. The results are obtained by averaging the running time of each algorithm. These algorithms are implemented with MATLAB (R2011b edition 64 bit)and run in PC (Microsoft Windows 7 64 bit, Intel Core 2 Duo CPU, 8 GB of RAM). As illustrated in the last row of Table 5, RFS is the less time consuming on all these sets, followed by LMSL. It should be noticed that LMSL just consumes slightly more time than RFS on some data sets, such as PROS (0.06 s). However, LLFS is much time consuming than LMSL, e.g. LLFS is 10 times slower than LMSL on ARP and COIL. LMSL is slower than RFS because it needs to compute margins of samples in order to obtain better performance for feature selection. The

[1] http://featureselection.asu.edu/datasets.php
[2] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[3] https://sites.google.com/site/feipingnie/file/NIPS2010_data.zip?attredirects=0

Table 2
Average accuracy of classification (%), p-value ≥ 0.05. (The value in the parentheses is p-value.)

Data set	LMSL proposed	RFS [1]	LLFS [6]	SPFS [15]	mRMR [39]	TR [18]
ARP	**93.89 ± 14.90 (1.00)**	70.89 ± 30.21 (0.00)	**93.04 ± 20.81 (0.53)**	87.40 ± 15.58 (0.00)	86.88 ± 13.75 (0.00)	90.60 ± 33.70 (0.04)
PIE	**98.63 ± 2.51 (0.29)**	**98.73 ± 0.74 (0.26)**	**98.89 ± 1.23 (0.58)**	95.79 ± 4.80 (0.00)	95.91 ± 5.67 (0.00)	**99.09 ± 1.20 (1.00)**
LUNG	**95.61 ± 2.67** (1.00)	**95.10 ± 3.57 (0.36)**	93.10 ± 4.28 (0.00)	**94.73 ± 1.80 (0.07)**	94.52 ± 2.74 (0.04)	**95.03 ± 1.93** (0.23)
CARC	**93.10 ± 6.48 (0.16)**	86.78 ± 7.11 (0.00)	**94.26 ± 6.85 (1.00)**	86.58 ± 7.81 (0.00)	87.14 ± 6.97 (0.00)	88.19 ± 10.14 (0.00)
PROS	**92.31 ± 7.77 (1.00)**	**91.63 ± 8.31 (0.45)**	**91.46 ± 7.82 (0.34)**	77.93 ± 17.64 (0.00)	79.27 ± 14.01 (0.00)	85.98 ± 13.50 (0.00)
TOX	**91.38 ± 20.62 (1.00)**	**88.78 ± 40.22 (0.14)**	**90.44 ± 25.79 (0.54)**	80.08 ± 28.58 (0.00)	79.79 ± 22.10 (0.00)	81.51 ± 37.17 (0.00)
ORL	**96.31 ± 10.18** (1.00)	89.04 ± 8.48 (0.00)	93.88 ± 11.89 (0.03)	87.73 ± 14.41 (0.00)	86.78 ± 11.65 (0.00)	**94.81 ± 8.92 (0.13)**
CLLS	**68.35 ± 41.92 (0.56)**	65.03 ± 19.33 (0.01)	**69.46 ± 28.36 (1.00)**	63.91 ± 17.90 (0.00)	62.35 ± 14.17 (0.00)	62.79 ± 45.83 (0.00)
COIL	**99.44 ± 2.59 (0.33)**	97.58 ± 1.81 (0.00)	**99.44 ± 2.07 (1.00)**	96.40 ± 7.43 (0.00)	96.72 ± 8.10 (0.00)	78.85 ± 8.23 (0.00)
AVE	**92.11**	87.06	91.55	85.62	85.48	86.32

Table 3
Classification accuracy of top 30% features, p-value ≥ 0.05. (The value in the parentheses is p-value.)

Data set	LMSL proposed	RFS [1]	LLFS [6]	SPFS [15]	mRMR [39]	TR [18]
ARP	**94.13 ± 22.55 (1.00)**	77.25 ± 47.30 (0.00)	**93.38 ± 26.50 (0.63)**	88.75 ± 26.64 (0.00)	87.75 ± 22.96 (0.00)	**91.00 ± 40.39 (0.09)**
PIE	**98.29 ± 4.64 (0.05)**	**99.36 ± 0.96 (1.00)**	**99.00 ± 2.38 (0.39)**	97.71 ± 3.52 (0.00)	97.71 ± 5.24 (0.01)	**99.07 ± 3.71 (0.56)**
LUNG	**96.02 ± 3.21 (0.66)**	**95.78 ± 5.68 (0.47)**	94.77 ± 5.21 (0.02)	95.16 ± 2.54 (0.03)	94.92 ± 3.57 (0.02)	**96.25 ± 2.42 (1.00)**
CARC	**93.45 ± 3.00 (0.36)**	91.90 ± 11.92 (0.02)	**93.97 ± 3.29 (1.00)**	89.91 ± 11.04 (0.00)	91.29 ± 7.03 (0.00)	91.55 ± 4.35 (0.00)
PROS	**91.88 ± 10.69 (1.00)**	87.81 ± 25.60 (0.00)	**90.94 ± 14.29 (0.41)**	88.59 ± 9.53 (0.00)	80.47 ± 32.77 (0.00)	89.06 ± 17.99 (0.02)
TOX	**92.83 ± 20.84 (1.00)**	**92.64 ± 20.95 (0.90)**	84.43 ± 23.94 (0.00)	89.43 ± 32.75 (0.04)	**91.51 ± 39.16** (0.45)	87.36 ± 31.89 (0.00)
ORL	**95.00 ± 7.24 (0.24)**	93.25 ± 19.14 (0.02)	**95.13 ± 10.84 (0.36)**	**94.13 ± 10.05 (0.05)**	**95.13 ± 10.84 (0.36)**	**96.00 ± 6.84 (1.00)**
CLLS	**70.14 ± 55.40 (1.00)**	**69.86 ± 42.51 (0.90)**	**69.14 ± 64.10 (0.68)**	64.29 ± 64.88 (0.02)	65.00 ± 65.20 (0.04)	**67.43 ± 29.56 (0.20)**
COIL	**99.73 ± 0.53 (0.64)**	**99.82 ± 0.23 (1.00)**	**99.77 ± 1.03 (0.86)**	**99.36 ± 1.92 (0.17)**	95.23 ± 7.73 (0.00)	**99.73 ± 1.49 (0.76)**
AVG	**92.38**	89.74	91.17	89.70	88.78	90.83

Table 4
Classification accuracy of top 60% features, p-value ≥ 0.05. (The value in the parentheses is p-value.)

Data set	LMSL proposed	RFS [1]	LLFS [6]	SPFS [15]	mRMR [39]	TR [18]
ARP	**94.25 ± 22.43 (1.00)**	86.00 ± 50.26 (0.00)	**94.00 ± 29.21 (0.88)**	89.63 ± 23.21 (0.00)	89.63 ± 23.21 (0.00)	**91.63 ± 38.34 (0.14)**
PIE	**99.29 ± 0.75 (0.05)**	**99.43 ± 2.66 (0.37)**	**99.14 ± 2.23 (0.09)**	97.93 ± 3.32 (0.00)	98.00 ± 3.52 (0.00)	**99.79 ± 0.49 (1.00)**
LUNG	**96.41 ± 3.62 (1.00)**	**96.25 ± 3.19 (0.79)**	**95.55 ± 4.95 (0.20)**	**95.39 ± 2.43 (0.07)**	**95.31 ± 2.83 (0.06)**	**95.70 ± 3.57 (0.25)**
CARC	**92.84 ± 3.84 (0.80)**	**92.07 ± 7.01 (0.24)**	**93.02 ± 5.47 (1.00)**	91.29 ± 8.28 (0.04)	91.29 ± 8.28 (0.04)	91.64 ± 2.90 (0.04)
PROS	**90.94 ± 8.12 (1.00)**	**88.59 ± 22.90** (0.07)	**89.84 ± 12.21 (0.28)**	**89.38 ± 11.72 (0.12)**	**89.38 ± 11.72 (0.12)**	**88.75 ± 16.86 (0.06)**
TOX	**93.11 ± 22.96 (1.00)**	**93.11 ± 24.09 (1.00)**	85.94 ± 27.53 (0.00)	**90.75 ± 28.07 (0.15)**	**90.75 ± 33.69 (0.17)**	**90.47 ± 21.54 (0.08)**
ORL	**96.25 ± 7.57 (1.00)**	**94.88 ± 13.47 (0.19)**	**96.00 ± 9.47 (0.79)**	**96.13 ± 9.52 (0.89)**	**96.00 ± 9.47 (0.79)**	**96.00 ± 7.50 (0.77)**
CLLS	**70.00 ± 41.68 (1.00)**	**69.86 ± 46.81 (0.95)**	**69.00 ± 63.82 (0.67)**	64.86 ± 55.94 (0.03)	65.29 ± 59.53 (0.04)	**68.71 ± 24.47 (0.48)**
COIL	**99.64 ± 1.34 (0.57)**	**99.73 ± 0.27 (0.68)**	**99.82 ± 0.66 (1.00)**	**99.41 ± 1.50 (0.22)**	95.23 ± 7.73 (0.00)	**99.45 ± 1.77 (0.30)**
AVG	**92.52**	91.10	91.37	90.53	90.10	91.35

iterations of LLFS spend much time, especially when the number of samples is large or the dimension of samples is very high.

4.6. Discussions

In this section, we provide further discussions on the proposed LMSL algorithm. The five algorithms in comparison with LMSL can be classified into three classes. The first class consists of algorithms that do not investigate local structures of sample, including RFS, SPFS and mRMR. The second class, including TR, uses Laplacian graph to represent structure of samples. The third class, including LLFS, integrates the margin into the algorithm. As illustrated by classification results, LMSL significantly outperforms the algorithms of the first class, which shows the importance of integrating the margin of the nearest neighbors between nearest hit and nearest miss for feature selection. The algorithm TR in the second class, although uses structure of samples, does not perform as well as the LMSL on most data sets. This shows the limitation of using graph for representing structure of samples in feature selection. The algorithm LLFS in the third class integrates the margin into the algorithm as well. Comparison results show that the proposed LMSL performs better than LLFS on six data sets. On the other four data sets, LMSL does not perform so well as LLFS. This may be due to that only suboptimal solution is obtained for LMSL. However it should be noticed that LMSL requires less computational time than LLFS on these test data sets. LLFS is even ten times more consuming on some high-dimensional data sets such as COIL. Moreover LMSL is insensitive to the kernel width σ, the regularization parameter λ and the initialization. These properties make it robust and stable in many applications.

5. Conclusion and future work

In this paper, we propose a novel feature selection algorithm, which maximizes the margin in the projection subspace $\boldsymbol{W}$. An $\ell_{2,1}$-norm regularization is also added to encourage row-sparsity of the solution. Experiment results show that it has competitive with some other existing algorithms. Nevertheless, only suboptimal solution of the algorithm is obtained in this paper. One of the future work is how to obtain the global optimum of LMSL. In addition, we will further investigate to incorporate the proposed algorithm within the image processing domain (such as visual-oriented grayscale frames in [40], mathematical morphology in

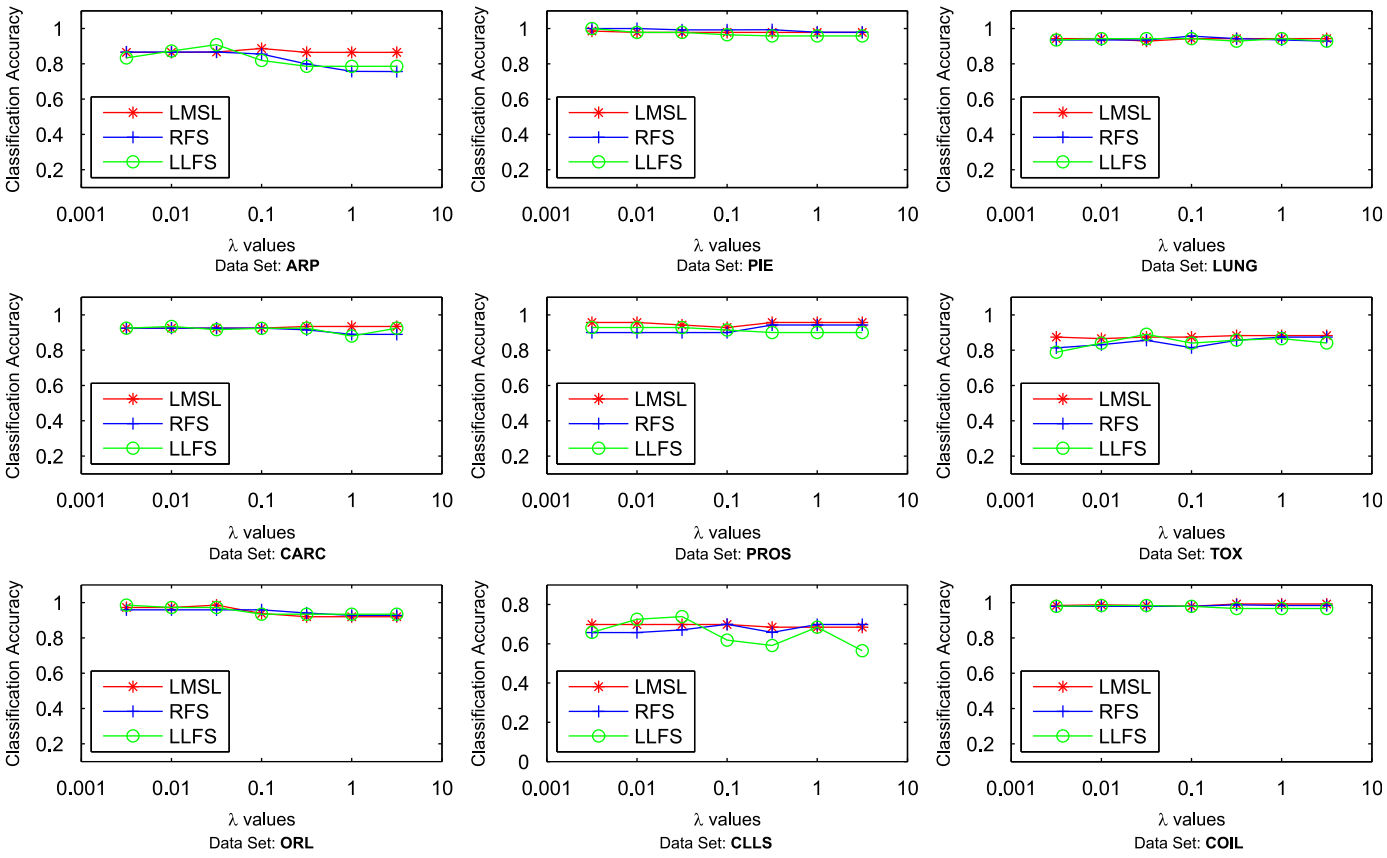

Fig. 1. Sensibility of the regularization parameter λ.

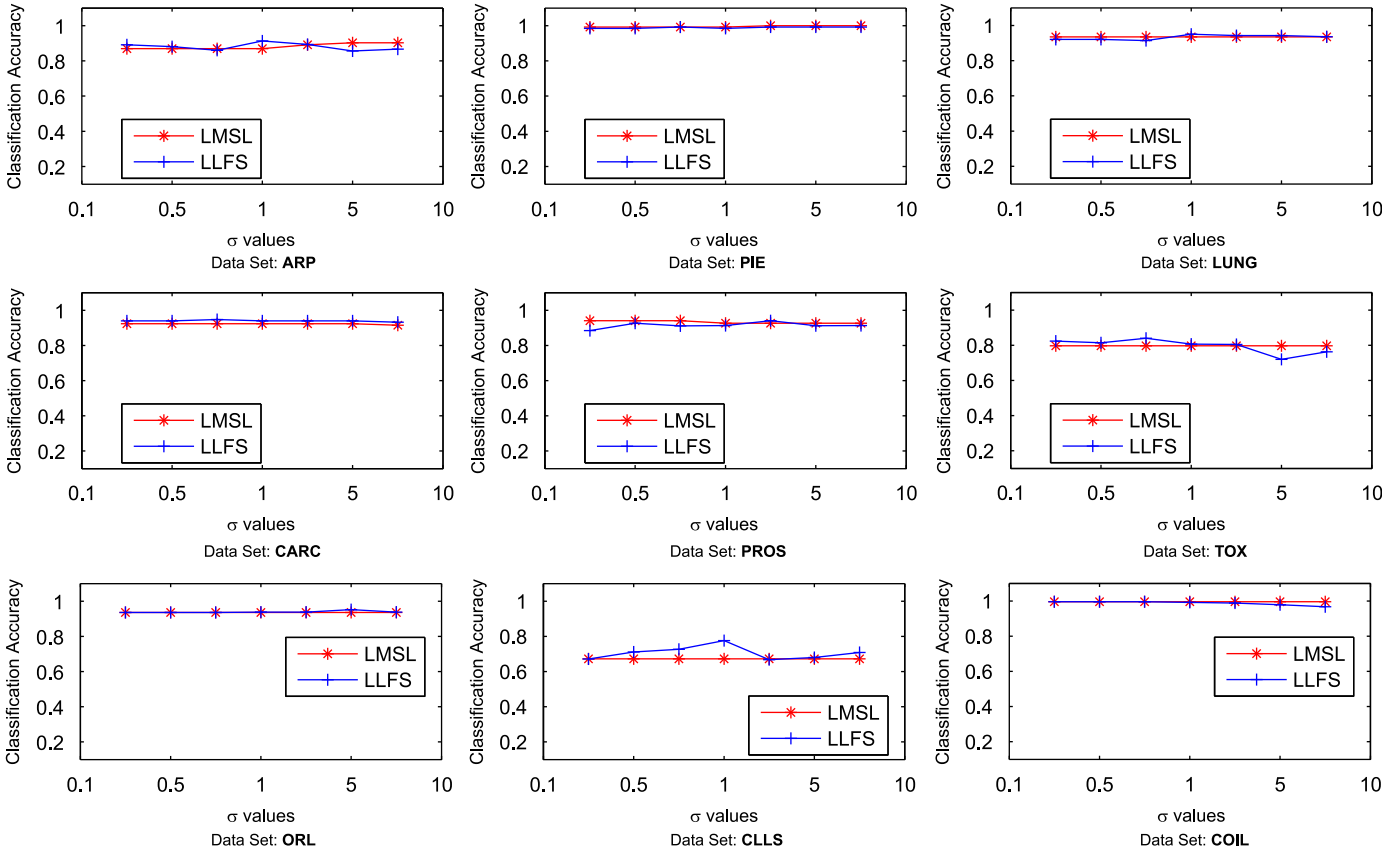

Fig. 2. Sensibility of the parameter σ.

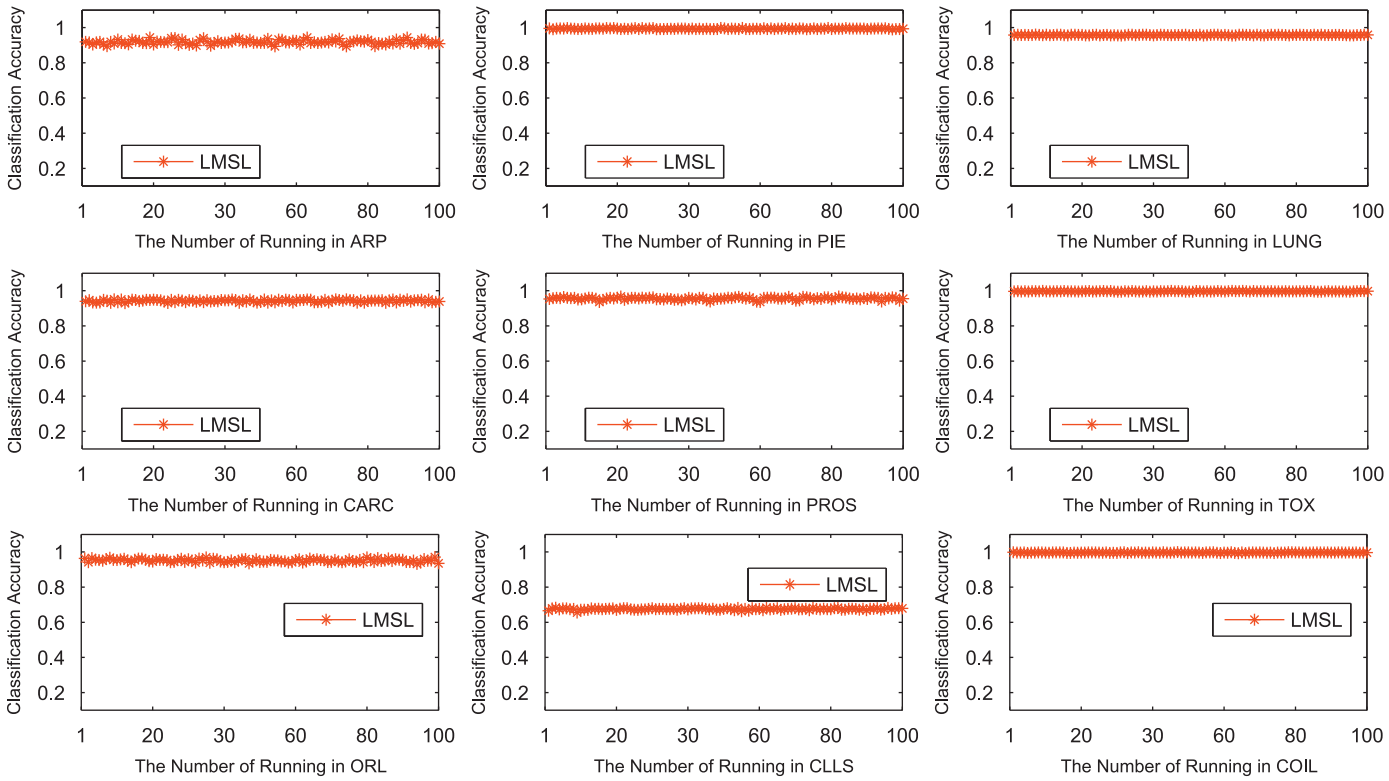

Fig. 3. Sensibility of the starting point $\boldsymbol{W_0}$.

Table 5
CPU time (in seconds) of three algorithms performed on nine data sets.

Data set	LMSL proposed	RFS [1]	LLFS [6]
ARP	5.79	**2.25**	60.51
PIE	13.56	**4.30**	131.12
LUNG	14.99	**5.83**	15.36
CARC	30.19	**12.09**	472.56
PROS	3.73	**4.10**	15.33
TOX	16.09	**8.02**	84.70
ORL	9.77	**6.38**	70.84
CLLS	7.75	**9.31**	58.61
COIL	219.16	**51.36**	2725.59
AVG	35.67	**11.52**	403.85

[41-43], etc.) to improve accordingly the performance of existed algorithms.

Conflict of interest statement

None declared.

Acknowledgments

The work is supported by Program for Natural Science Foundation of China (61173129), the Specialized Research Fund for the Doctoral Program of Higher Education of China (20120191110026), the Fundamental Research Funds for the Central Universities (CDJXS11182240), the National Natural Science Foundation of China (Project No. 60970034 and 61170287), the Foundation for the Author of National Excellent Doctoral Dissertations (Grant No. 2007B4), and the Foundation for the Author of Hunan Provincial Excellent Doctoral Dissertations, Chongqing Municipal Education Commission funded projects (KJ120723) and (KJ110730), Chongqing, China.

References

[1] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint $l_{2,1}$ norms minimization, in: Advances Neural Information Processing Systems (NIPS), vol. 23, 2010, pp. 1813–1821.
[2] Z. Suna, G. Bebisa, R. Miller, Object detection using feature subset selection, Pattern Recognition 37 (2004) 2165–2176.
[3] X. Zhu, Z. Huang, H. Shen, J. Cheng, Dimensionality reduction by mixed kernel canonical correlation analysis, Pattern Recognition 45 (2012) 3003–3016.
[4] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, V.Vapnik, Feature selection for svms, in: Advances in Neural Information Processing Systems (NIPS), vol. 13, 2001, pp. 668–674.
[5] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Machine Learning 46 (2002) 131–159.
[6] Y. Sun, S. Todorovic, S. Goodlson, Local-learning-based feature selection for high-dimensional data analysis, IEEE Transaction on Pattern Analysis and Machine Intelligence 32 (2010) 1610–1626.
[7] R. Bachrachy, A. Navotz, N. Tishbyy, Margin based feature selection theory and algorithms, in: Fifteenth International Conference on Machine Learning (ICML 2004), vol. 21, 2004, pp. 43–50.
[8] Q. Gu, Z. Li, J. Han, Joint feature selection and subspace learning, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, vol. 2, 2011, pp. 1294–1299.
[9] J. Sotoca, F. Pla, Supervised feature selection by clustering using conditional mutual information-based distances, Pattern Recognition 43 (2010) 2068–2081.
[10] X. He, M. Ji, C. Zhang, H. Bao, A variance minimization criterion to feature selection using Laplacian regularization, IEEE Transaction on Pattern Analysis and Machine Intelligence 33 (2012) 2013–2025.
[11] J. Dy, C. Brodley, Feature subset selection and order identification for unsupervised learning, in: 17th Fifteenth International Conference on Machine Learning, vol. 17, 2000, pp. 247–254.
[12] X. Zhu, Z. Huang, Y. Yang, H. Shen, C. Xu, J. Luo, Self-taught dimensionality reduction on the high-dimensional small-sized data, Pattern Recognition 46 (2013) 215–229.
[13] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of relief and relief, Machine Learning 53 (2003) 23–69.
[14] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (2002) 389–422.

[15] Z. Zhao, L. Wang, H. Liu, J. Ye, On similarity preserving feature selection, IEEE Transactions on Knowledge and Data Engineering 25 (3) (2013) 619–632.
[16] R. Duda, P. Hart, D. Stork, Pattern Classification, John Wiley Sons, New York, 2001.
[17] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, 2005, pp. 507–514.
[18] F. Nie, S. Xiang, Y. Jia, C. Zhang, S. Yan, Trace ratio criterion for feature selection, in: The Twenty-Third AAAI Conference on Artificial Intelligence, vol. 2, 2008, pp. 671–676.
[19] R. Kohavi, G. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1997) 273–324.
[20] P. Pudil, J. Novovicova, Novel methods for subset selection with respect to problem knowledge, IEEE Intelligent Systems 13 (1998) 66–74.
[21] T. Lal, O. Chapelle, J. Weston, A. Elisseeff, Feature Extraction Foundations and Applications, Springer-Verlag, 2006.
[22] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, 1-norm support vector machines, in: Neural Information Processing Systems, MIT Press, 2003, p. 16.
[23] M. Nguyen, F. Torre, Optimal feature selection for support vector machines, Pattern Recognition 43 (2010) 584–591.
[24] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1996) 711–720.
[25] X. He, P. Niyogi, Locality preserving projections, in: Advances in Neural Information Processing Systems, vol. 16, 2003.
[26] X. He, D. Cai, S. Yan, H. Zhang, Neighborhood preserving embedding, in: 10th IEEE International Conference on Computer Vision, vol. 2, 2005, pp. 1208–1213.
[27] L. Zhang, S. Chen, L. Qiao, Graph optimization for dimensionality reduction with sparsity constraints, Pattern Recognition 45 (2012) 1205–1210.
[28] S. Yan, B.Z.D. Xu, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 40–51.
[29] L. Torresani, K. Lee, Large margin component analysis, in: 18th in Neural Information Processing Systems (NIPS), 2006, pp. 1385–1392.
[30] F. Pernkopf, M. Wohlmayr, Stochastic margin-based structure learning of Bayesian network classifiers, Pattern Recognition 46 (2013) 464–471.
[31] K. Weinberger, L. Saul, Distance metric learning for large margin nearest neighbor classification, Journal of Machine Learning Research 10 (2009) 207–244.
[32] P. Bradley, O. Mangasarian, Feature selection via concave minimization and support vector machines, in: Proceedings of the Fifteenth International Conference on Machine Learning, vol. 1, 1998, pp. 82–90.
[33] C. Ding, D. Zhou, X. He, H. Zha, R1-pca: rotational invariant ℓ_1-norm principal component analysis for robust subspace factorization, in: Proceedings of the 23rd International Conference on Machine Learning, vol. 1, 2006, pp. 281–288.
[34] L. Wang, J. Zhu, H. Zou, Hybrid huberized support vector machines for microarray classification and gene selection, in: Proceedings of the 24th International Conference on Machine Learning, ACM Press, 2007, pp. 983–990.
[35] A. Argyriou, T. Evgeniou, M. Pontil, Multi-task feature learning, in: Advances in Neural Information Processing Systems (NIPS), vol. 20, 2007, pp. 41–48.
[36] J. Liu, S. Ji, J. Ye, Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization, in: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, 2009, pp. 339–348.
[37] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[38] R. Horm, C. Johnson, Topics in Matrix Analysis, Cambridge University Press, 1994.
[39] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transaction on Pattern Analysis and Machine Intelligence 27 (2005) 1226–1238.
[40] D. Vizireanu, R. Udrea, Visual-oriented morphological foreground content grayscale frames interpolation method, Journal of Electronic Imaging 45 (2009) 020502, pp. 1–3.
[41] R. Udrea, D. Vizireanu, Iterative generalization of morphological skeleton, Journal of Electronic Imaging 16 (2007) 010501, p. 1–3.
[42] B. Fang, Y. Tang, Elastic registration for retinal images based on reconstructed vascular trees, IEEE Transactions on Biomedical Engineering 53 (6) (2006) 1183–1187.
[43] B. Fang, Mi. Cheng, Y. Tang, Improving the discriminant ability of local margin based learning method by incorporating the global between-class separability criterion, Neurocomputing 73 (1-3) (2009) 536–541.

Bo Liu is a PhD candidate in College of Computer Science, The Chongqing University, China. His research interests include machine learning, pattern recognition, and convex optimization.

Bin Fang is currently a Professor in the Department of Computer Science at the Chong Qing University. His research interests include computer vision, pattern recognition, medical image processing, biometrics applications, and document analysis.

Xinwang Liu is currently pursuing his PhD. His research interests focus on kernel learning and feature selection.

Jie Chen is PhD candidate in Institut Charles Delaunay, Université de Technologie de Troyes, and France. His research interests lie at the intersection of statistical signal processing and machine learning. Specific interests include kernel methods, supervised and unsupervised learning.

Zhenghong Huang is currently a Professor in the School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing, China. His research interests include pattern recognition, medical image processing, and wavelet analysis.

Xiping He is currently a Professor in the School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing, China. His research interests include machine learning, wavelet analysis.