LETTER Communicated by Feiping Nie

Multiclass Classification and Feature Selection Based on Least Squares Regression with Large Margin

Haifeng Zhao
ahu_cs@163.com
Siqi Wang
wangsiqi_1112@163.com
College of Computer Science and Technology, Anhui University, Hefei 230601, China

Zheng Wang
wang77802@163.com
Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China

Least squares regression (LSR) is a fundamental statistical analysis technique that has been widely applied to feature learning. However, limited by its simplicity, the local structure of data is easy to neglect, and many methods have considered using orthogonal constraint for preserving more local information. Another major drawback of LSR is that the loss function between soft regression results and hard target values cannot precisely reflect the classification ability; thus, the idea of the large margin constraint is put forward. As a consequence, we pay attention to the concepts of large margin and orthogonal constraint to propose a novel algorithm, orthogonal least squares regression with large margin (OLSLM), for multiclass classification in this letter. The core task of this algorithm is to learn regression targets from data and an orthogonal transformation matrix simultaneously such that the proposed model not only ensures every data point can be correctly classified with a large margin than conventional least squares regression, but also can preserve more local data structure information in the subspace. Our efficient optimization method for solving the large margin constraint and orthogonal constraint iteratively proved to be convergent in both theory and practice. We also apply the large margin constraint in the process of generating a sparse learning model for feature selection via joint $\ell_{2,1}$-norm minimization on both loss function and regularization terms. Experimental results validate that our method performs better than state-of-the-art methods on various real-world data sets.

Neural Computation **30, 2781–2804** (2018)
doi:10.1162/neco_a_01116

1 Introduction

Numerous real-world applications, such as text mining (Forman, 2003), bioinformatics (Ding & Peng, 2005; Wang et al., 2012), and medical image analysis (Tang, Cui, & Jiang, 2017), contain enormous amounts of high-dimensional data that can be difficult to process because they contain millions of features. In addition, the process of handling the high-dimensional data is complex, and the redundant features may deteriorate the performance. Feature extraction and feature selection are two main techniques for dimensionality reduction. The major difference between them is that feature extraction finds a meaningful low-dimensional representation of high-dimensional data by learning a linear or nonlinear transformation matrix, while feature selection aims to extract the most representative features from the original features and the property of each feature is unchanged.

During the past several decades, an abundance of algorithms for feature extraction have been proposed; principal component analysis (PCA; Jolliffe, 2005) and linear discriminant analysis (LDA; Duda, Hart, & Stork, 2001) are the best-known methods for linear dimensionality reduction. PCA is an unsupervised algorithm that maximizes the global variance of the data to obtain a projection matrix; the columns of the projection matrix are made of the largest eigenvalues of covariance matrix corresponding to the eigenvectors. However, it does not take the local structure of the data samples (Yu, 2012) into account. Manifold learning is another approach to subspace learning, whose most representative work is locally linear embedding (LLE; Roweis & Saul, 2000; Saul & Roweis, 2003), which constructs the reconstruction weight matrix to reflect intrinsic geometric properties of the data. However, LLE also has a very serious problem: it is sensitive to parameters and noise. The instability of the LLE method can be attributed to the factorizing of an abnormal matrix; the smallest eigenvalues are very small, and the largest eigenvalues are very large. Therefore, stable LLE (SLLE; Hou, Zhang, Wu, & Jiao, 2009) focuses on this problem of enhancing the stability of LLE and employs kernel transformation to avoid eigenproblems. The prime limitation of unsupervised methods is neglect of label information, which plays an important role in classification. Hence, in some classification tasks, supervised learning usually has better performance. LDA is the most typical supervised algorithm whose core task is to search an optimal projection by maximizing the ratio of the between-class scatter and the within-class scatter (Russell, Chiang, & Braatz, 2000). Some studies have concluded that the LDA in a two-class condition is equivalent to the least squares regression.

Least squares regression (LSR) has attracted a great deal of attention and has been used for many practical situations. Many popular models have been derived from it, including ridge regression (Hoerl & Kennard, 2000), LASSO (Tibshirani, 1996), and support vector machines (SVM; Cortes & Vapnik, 1995; Chang & Lin, 2011; Hou, Nie, Zhang, Yi, & Wu, 2014). Furthermore, LSR has been applied in many machine learning fields, such as

multilabel learning, semisupervised learning, and clustering. Recently, many researchers have attempted to use LSR to deal with dimensionality-reduction problems. However, LSR is constrained by the simplicity of the model: the local structural information cannot be maintained. In order to overcome this drawback, Zhao, Wang, and Nie (2016) have proposed a novel orthogonal least squares regression (OLSR) for feature extraction, and the method is confirmed to be a special case of the quadratic problem on the Stiefel manifold (QPSM) (Nie, Zhang, & Li, 2017).

The main work of this letter is similar to orthogonal least squares discriminant analysis (OLSDA; Nie, Xiang, Liu, Hou, & Zhang, 2012), which exploits orthogonal projection to constrain transformation matrix to avoid trivial solutions and preserve more local structure information. The orthogonal projection is desirable and often demonstrates good performance empirically. Thus, much previous work has used this property, including orthogonal neighborhood preserving projections (ONPP; Kokiopoulou & Saad, 2007), orthogonal locality preserving projections (OLPP; Cai, He, Han, & Zhang, 2006), and orthogonal locality minimizing globality maximizing projections (OLMGMP; Nie, Xiang, Song, & Zhang, 2009). In addition, the least squares loss between soft regression results and hard labels makes conventional LSR less suitable for classification (Bishop, 2006; Hastie, Tibshirani, & Friedman, 2001). To address this issue, many new LSR variants have been proposed; they can be categorized in two ways.

One way is to use a surrogate loss function to replace the least squares loss for improving the property of a model, such as hinge loss (Cortes & Vapnik, 1995), squared hinge loss (Chang, Hsieh, & Lin, 2008), and logistic loss (Hosmer & Lemeshow, 2004). The other category is to alter the absolute values with soft labels obtained through different methods, which can ensure that the new learned labels are beneficial to keep more discrimination information. For example, Xiang, Nie, Meng, Pan, and Zhang (2012) propose a framework called discriminative least squares regression (DLSR) that enlarges the distance between the true classes and false classes through using ε-dragging technique to force the regression targets of different categories in the opposite direction. Retargeted least squares regression (ReLSR; Zhang, Wang, Xiang, & Liu, 2015) introduces a target matrix from input data to guarantee each sample with a margin that is larger than one; it is much more accurate in measuring the classification error of a regression model.

Although OLSR can retain local information, the hard labels are still used in this algorithm, which is not well suited to classification. Thus, we propose a novel method: incorporating the large margin constraint into the classical least squares regression under the orthogonal subspace for multiclass classification: we call it orthogonal least squares regression with large margin (OLSLM).

Except the feature extraction algorithms, feature selection is another crucial technique for dimensionality reduction. It can roughly be divided into

three categories: filter, wrapper, and embedded methods. In filter models, features are preselected based on certain intrinsic properties of the data, such as variance, and some statistical indicators of the features. The selection is independent of the subsequent learning algorithm. Some popular filter feature selection methods encompass Fisher score (FS; Duda et al., 2001), mRMR (Peng, Long, & Ding, 2005), T-test (Montgomery, Runger, & Hubele, 2007), and information gain (IG; Raileanu & Stoffel, 2004). Wrapper methods (Kohavi & John, 1997) depend on the specific learning algorithms to yield learned results that can be used to select distinctive subsets of features. However, the computational costs of wrapper approaches are more expensive than filter methods. In addition, embedded methods integrate feature selection and classification model into a single optimization problem.

The scope of research on sparsity regularization in feature selection has grown rapidly due to its robustness and efficiency. For example, ℓ_1-SVM (Bradley & Mangasarian, 1998) adopts the ℓ_1-norm regularization that typically yields a sparse solution to perform feature selection. Due to the ℓ_1-norm penalty function in ℓ_1-SVM, the number of selected variables is upper-bounded by the sample size. To remedy this drawback, Wang, Zhu, and Zou (2007) propose a hybrid Huberized SVM by combining both ℓ_1-norm and ℓ_2-norm. Recently, the structural sparsity has been extremely important in selecting the group features. FS20 (Cai, Nie, & Huang, 2013) incorporates an explicit $\ell_{2,0}$-norm equality constraint into an $\ell_{2,1}$-norm loss term. In comparison to $\ell_{2,0}$-norm, due to $\ell_{2,1}$-norm being convex, researchers prefer using $\ell_{2,1}$-norm as the regularization term. Nie, Huang, Cai, and Ding (2010) propose a robust and efficient feature selection method via joint $\ell_{2,1}$-norm minimization on both least squares loss function and regularization. It is well known that the $\ell_{2,1}$-norm-based loss function is robust to outliers and the $\ell_{2,1}$-norm regularization is used to select features across all data samples with joint sparsity. Many newly proposed algorithms for feature selection are based on $\ell_{2,1}$-norm regularization, both supervised (Xiang et al., 2012; Wang et al., 2011; Lan, Hou, & Yi, 2016) and unsupervised (Yang, Hou, Nie, & Wu, 2012; Hou, Nie, Li, Yi, & Wu, 2014) methods. In addition, correntropy can enhance robustness as well. He, Tan, Wang, and Zheng (2012) apply correntropy and $\ell_{2,1}$-norm regularization into a unified framework: correntropy regularization algorithm (CRFS). In this letter, a novel feature selection model, least squares regression with large margin for feature selection (LSLM-FS), has been proposed that joints $\ell_{2,1}$-norm on both loss function and regularization. The large margin constraint is incorporated into our model to improve the classification.

The main contributions of our letter are summarized as follows:

1. Our OLSLM model combines orthogonal constraint and the large margin constraint, which can preserve more local information in the subspace and ensure the margin between true and false classes larger than one.

2. We propose another feature selection algorithm with a large margin constraint, which achieves awesome performance on several real-world data sets.
3. We propose two efficient iterative optimization algorithms to optimize these two models. We also present the convergence of two models.

The experiments are divided into two parts: OLSLM and LSLM-FS. We use classification accuracy as the evaluation criterion of the models. Experiments are conducted on several public benchmark data sets, including the UCI and high-dimensional data sets, and it indicates that the OLSLM algorithm outperforms comparison algorithms, including an SVM classifier with four different parameters, conventional LSR, and its varieties. The validity of the extended feature selection algorithm LSLM-FS is tested on nine data sets from different fields. We also use recognition accuracy to evaluate of performance.

In section 2, we give some notations and definitions. In section 3, we describe the OLSLM model for multiclass classification in detail. In section 4, we present the LSLM-FS model. In section 5, experiments are conducted to evaluate the OLSLM and LSLM-FS methods. The conclusions are drawn in section 6.

2 Notations and Definitions

We summarize the notations and definitions of norms used in this letter. We use bold to represent matrices (uppercase) and vectors (lowercase) and regular fonts to represent scalars. For matrix $\mathbf{W} = (w_{ij})$, its ith row, jth column are denoted as $\mathbf{w}^i$, $\mathbf{w}_j$, respectively. The ℓ_p-norm of the vector $\mathbf{v} \in \mathbb{R}^n$ is defined as $\|\mathbf{v}\|_p = \left(\sum_{i=1}^n |v_i|^p\right)^{\frac{1}{p}}$. The Frobenius norm of the matrix $\mathbf{W} \in \mathbb{R}^{n\times m}$ is defined as $\|\mathbf{W}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m w_{ij}^2} = \sqrt{\sum_{i=1}^n \|\mathbf{w}^i\|_2^2}$. And the $\ell_{2,1}$-norm of matrix $\mathbf{W}$ is defined as $\|\mathbf{W}\|_{2,1} = \sum_{i=1}^n \sqrt{\sum_{j=1}^m w_{ij}^2} = \sum_{i=1}^n \|\mathbf{w}^i\|_2$.

3 Orthogonal Least Squares Regression with Large Margin

3.1 Least Squares Regression Revisited. We briefly review the least squares regression model with a class indictor matrix. Given the data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n\times d}$ and each data point $\mathbf{x}_i \in \mathbb{R}^d$ has a corresponding class label denoted by $\mathbf{y}_i \in \mathbb{R}^c$ (c is the number of classes). The purpose of LSR is to learn a transformation matrix $\mathbf{W} \in \mathbb{R}^{d\times c}$ and a bias vector $\mathbf{b} \in \mathbb{R}^c$ to estimate the target matrix $\mathbf{T}$, which can be approximately written as

$$\mathbf{XW} + \mathbf{e}_n\mathbf{b}^T \approx \mathbf{T}, \tag{3.1}$$

where $\mathbf{e}_n = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$. In the LSR model, we use a zero-one vector (1 for true class and 0 for other classes) as a class label and let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]^T \in \mathbb{R}^{n\times c}$ as the target matrix. $\mathbf{Y}$ is defined as follow. If the ith ($i = 1, 2, \ldots, n$) sample belongs to the jth class ($j = 1, 2, \ldots, c$), $\mathbf{y}_i = [0, \ldots 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^c$, only the jth element is equal to one.

The objective of LSR can be written as

$$\min_{\mathbf{W},\mathbf{b}} \left\|\mathbf{XW} + \mathbf{e}_n\mathbf{b}^T - \mathbf{Y}\right\|_F^2 + \beta\left\|\mathbf{W}\right\|_F^2, \tag{3.2}$$

where the second term is the regularization term and the β is a positive regularization parameter.

3.2 OLSLM for Multiclass Classification. Motivated by Xiang et al. (2012) and Zhao et al. (2016), the ordinary least squares regression can be extended by using orthogonal constraint to maintain more local discriminant information. In spite of this, the target values are still manually assigned as the absolute values 0/1 for a class label vector even though they are in orthogonal projection. Therefore, in this letter, we develop a novel method for multiclass classification, orthogonal least squares regression with large margin (OLSLM), which combines the target matrix constraint on the basis of OLSR so that it not only preserves local structure characteristics but also creates a large margin for targets to ensure the requirement of correct classification in theory. In this section, we describe the OLSLM model and present algorithm 1 for solving the optimization problem. Finally, we analyze the computation complexity of the algorithm and provide the convergence analysis.

3.2.1 Problem Formulation. In our method, the regression targets are used to guarantee that each sample can be correctly classified with a large margin; thus, the target matrix should be optimized according to the data with a constraint for each sample. Based on the target matrix, the margin between the targets of true and false classes should be larger than one, which can improve classification performance. In addition, compared to conventional methods, the orthogonal constraint removes the regularization term, so we can avoid the procedure of tuning parameters. From the analysis, our method can be formulated as the following problem:

$$\begin{aligned} &\min_{\mathbf{W}^T\mathbf{W}=\mathbf{I},\mathbf{b},\mathbf{T}} \left\|\mathbf{XW} + \mathbf{e}_n\mathbf{b}^T - \mathbf{T}\right\|_F^2 \\ &s.t.\ T_{i,y_i} - \max_{j\neq y_i} T_{i,j} \geq 1, i = 1, 2, \ldots, n, \end{aligned} \tag{3.3}$$

where $\mathbf{T} \in \mathbb{R}^{n\times c}$ is a target matrix and the constraint condition $T_{i,y_i} - \max_{j\neq y_i} T_{i,j} \geq 1, i = 1, 2, \ldots, n$ guarantees a large margin between the targets

of true and false classes, which ensures that our model will have a better performance of classification than OLSR in theory.

3.2.2 An Iterative Algorithm to Solve Problem 3.3. The problem with equation 3.3 is that it has three variables with two constraints that need to be optimized, which is a challenge. We present an iterative optimization algorithm to tackle this problem, and the optimum of three variables can be obtained simultaneously. We now introduce the steps of our optimization algorithm:

Step 1. We fix $\mathbf{T}$ to solve $\mathbf{W}$ and $\mathbf{b}$. Then the optimization problem, equation 3.3, becomes a regression problem:

$$J(\mathbf{W}, \mathbf{b}) = \min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}, \mathbf{b}} \left\| \mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^T - \mathbf{T} \right\|_F^2. \tag{3.4}$$

The Lagrangian function of equation 3.4 is

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \min_{\mathbf{W}, \mathbf{b}} \left\| \mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^T - \mathbf{T} \right\|_F^2 - \lambda(\mathbf{W}^T\mathbf{W} - \mathbf{I}), \tag{3.5}$$

where λ is the Lagrangian multiplier. First, we take the partial derivative of $\mathcal{L}(\mathbf{W}, \mathbf{b})$ with respect to $\mathbf{b}$ and set the partial derivative to zero; then we can solve $\mathbf{b}$:

$$\frac{\partial \mathcal{L}(\mathbf{W}, \mathbf{b})}{\partial \mathbf{b}} = 0 \Rightarrow \mathbf{b} = \frac{1}{n}(\mathbf{T} - \mathbf{X}\mathbf{W})^T\mathbf{e}_n. \tag{3.6}$$

Substituting equation 3.6 into equation 3.5, we can obtain

$$J(\mathbf{W}) = \min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \left\| \mathbf{H}\mathbf{X}\mathbf{W} - \mathbf{H}\mathbf{T} \right\|_F^2, \tag{3.7}$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix. Without loss of generality, equation 3.7 can be rewritten as

$$J(\mathbf{W}) = \min_{\mathbf{W}^T\mathbf{W}=\mathbf{I}} \left\| \mathbf{A}\mathbf{W} - \mathbf{Q} \right\|_F^2, \tag{3.8}$$

where $\mathbf{A} = \mathbf{H}\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{Q} = \mathbf{H}\mathbf{T} \in \mathbb{R}^{n\times c}$. To solve equation 3.8, we introduce an auxiliary matrix $\mathbf{G}$. Then the equation 3.8 can be reformulated as

$$J(\mathbf{G}) = \min_{\mathbf{G}^T\mathbf{G}=\mathbf{I}} \left\| \mathbf{A}\mathbf{G} - [\mathbf{Q}, \mathbf{A}\mathbf{E}] \right\|_F^2, \tag{3.9}$$

where $\mathbf{G} = [\mathbf{W}, \mathbf{E}] \in \mathbb{R}^{d\times d}$ and $\mathbf{E} \in \mathbb{R}^{d\times(d-c)}$ is an orthogonal complement of an orthonormal matrix $\mathbf{W}$. It has been verified that the solution to equation 3.9 is equal to problem 3.8 in the literature (Zhao et al., 2016). Then we can use the SVD decomposition of

$$\mathbf{A}^T[\mathbf{Q}, \mathbf{AE}] = \mathbf{U}\sum\mathbf{V}^T, \tag{3.10}$$

where $\mathbf{U} \in \mathbb{R}^{d\times d}$, $\sum \in \mathbb{R}^{d\times d}$, $\mathbf{V} \in \mathbb{R}^{d\times d}$, and we can obtain the solution

$$\mathbf{G} = \mathbf{UV}^T. \tag{3.11}$$

Then the transformation matrix $\mathbf{W}$ is obtained by the first c columns of matrix $\mathbf{G}$.

Step 2. Given $\mathbf{W}$ and $\mathbf{b}$, equation 3.3 is reduced to solving the retargeting problem:

$$\begin{aligned} J(\mathbf{T}) &= \min_{\mathbf{T}} \left\|\mathbf{XW} + \mathbf{e}_n\mathbf{b}^T - \mathbf{T}\right\|_F^2 = \left\|\mathbf{R} - \mathbf{T}\right\|_F^2 \\ &s.t. \quad T_{i,y_i} - \max_{j\neq y_i} T_{i,j} \geq 1, \end{aligned} \tag{3.12}$$

where $\mathbf{R} = \mathbf{XW} + \mathbf{e}_n\mathbf{b}^T \in \mathbb{R}^{n\times c}$; equation 3.12 can be divided into n subproblems, with each subproblem corresponding to the learning of one row of $\mathbf{T}$. Denote r_{ij}, t_{ij} as the i, jth elements of $\mathbf{R}$, $\mathbf{T}$, and let $\mathbf{r}^i$ and $\mathbf{t}^i$ be the ith row of $\mathbf{R}$ and $\mathbf{T}$, respectively. Then $\mathbf{t}^i$ can be obtained from the following optimization problem. Problem 3.12 can decomposed into n subproblems with a general form:

$$\begin{aligned} &\min_{\mathbf{t}^i} \left\|\mathbf{r}^i - \mathbf{t}^i\right\|_2^2 = \sum_{i=1}^{c} (r_{ij} - t_{ij})^2, \\ &s.t. \quad t_{i,y_i} - \max_{j\neq y_i} t_{i,j} \geq 1. \end{aligned} \tag{3.13}$$

Problem 3.13 can be solved using the retargeting algorithm in Zhang et al. (2015).

In summary, we have described the optimization algorithm for solving problem 3.3 and listing the detailed steps in algorithm 1.

3.2.3 Convergence Analysis. To analyze the convergence of algorithm 1, we denote the objective function value in equation 3.3 by $J(\mathbf{W}, \mathbf{b}, \mathbf{T})$. In the tth iteration, we use the value of the objective function by $J(\mathbf{W}^t, \mathbf{b}^t, \mathbf{T}^t)$. During the $(t+1)$th iteration, we fix $\mathbf{T}^t$ and update $\mathbf{W}^t$, $\mathbf{b}^t$ respectively. Then we have

Algorithm 1: An Iterative Algorithm to Solve Problem 3.3.

1 **Input:**
2 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T \in \mathbb{R}^{n\times d}$; $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T \in \mathbb{R}^n$;
3 **Initialization:**
4 $t = 0$;
5 $T_{ij}^{(0)} = \begin{cases} 0 & \text{if } y_i = j \\ 1 & \text{otherwise} \end{cases}$;
6 **for** $t = 1 \rightarrow$ *iterNum* **do**
7 Obtain $\mathbf{W}^{(t)}$ and $\mathbf{b}^{(t)}$ according to equation 3.11 **W=G**(:,1:c) and equation 3.6 with $\mathbf{T}^{(t-1)}$ respectively ;
8 $\mathbf{R}^{(t)} = \mathbf{X}\mathbf{W}^{(t)} + \mathbf{e}_n\mathbf{b}^{(t)^T}$;
9 **for** $i = 1 \rightarrow n$ **do**
10 $\mathbf{T}_{i*}^{(t)} =$ retargeting($\mathbf{R}_{i*}^{(t)}, y_i$);
11 **end**
12 **end**
13 $t = t + 1$.
14 **Output:** $\mathbf{W}^* \in \mathbb{R}^{d\times c}$,$\mathbf{b}^* \in \mathbb{R}^{c\times 1}$

$$J(\mathbf{W}^t, \mathbf{b}^{t+1}, \mathbf{T}^t) \leq J(\mathbf{W}^t, \mathbf{b}^t, \mathbf{T}^t) \tag{3.14}$$

and

$$J(\mathbf{W}^{t+1}, \mathbf{b}^{t+1}, \mathbf{T}^t) \leq J(\mathbf{W}^t, \mathbf{b}^{t+1}, \mathbf{T}^t). \tag{3.15}$$

Updating $\mathbf{T}^{t+1}$ by using $\mathbf{W}^{t+1}$, $\mathbf{b}^{t+1}$, we obtain

$$J(\mathbf{W}^{t+1}, \mathbf{b}^{t+1}, \mathbf{T}^{t+1}) \leq J(\mathbf{W}^{t+1}, \mathbf{b}^{t+1}, \mathbf{T}^t). \tag{3.16}$$

Combining equations 3.14 to 3.16, we obtain

$$J(\mathbf{W}^{t+1}, \mathbf{b}^{t+1}, \mathbf{T}^{t+1}) \leq J(\mathbf{W}^t, \mathbf{b}^t, \mathbf{T}^t). \tag{3.17}$$

In a word, we can draw a conclusion that algorithm 1 can monotonically decrease the objective of $J(\mathbf{W}, \mathbf{b}, \mathbf{T})$ in each iteration.

3.2.4 Computational Complexity Analysis. Given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, the main computational costs of algorithm 1 are concentrated in steps 6 to 12:

- The complexity of step 7 is $O(mnd^2 + md^3)$.

- The complexity of step 8 is $O(ndc)$.
- We need $O(nc)$ to calculate steps 9 to 11.

m and k are the number of iterations in OLSR and OLSLM, respectively. Considering that in real-world applications, all of the data sets are resized to a proper size, the number of data is much larger than m and d. The computational complexity of OLSLM is $O(kmnd^2)$, which is linear with respect to n; therefore, our algorithm is easy to implement.

4 Least Squares Regression with Large Margin for Feature Selection

4.1 Robust Feature Selection based on $\ell_{2,1}$-norm. For simplicity, the LSR model can be written as

$$\min_{\mathbf{W}} \sum_{i=1}^{n} \left\| \mathbf{W}^T \mathbf{x}_i - \mathbf{y}_i \right\|_2^2, \tag{4.1}$$

where the bias $\mathbf{b}$ can be absorbed into $\mathbf{W}$, so the $\mathbf{W} = [\mathbf{W}^T, \mathbf{b}]^T \in \mathbb{R}^{(d+1)\times c}$ and $\mathbf{x}_i$ is the corresponding augmented vector $\mathbf{x}_i = [\mathbf{x}_i^T, 1]^T$. In the RFS (Nie et al., 2010), the robust loss function can be written as

$$\min_{\mathbf{W}} \sum_{i=1}^{n} \left\| \mathbf{W}^T \mathbf{x}_i - \mathbf{y}_i \right\|_2, \tag{4.2}$$

where the residual is not squared; thus, outliers have less influence than in a squared residual. When adding a regularization term with a parameter β, the problem becomes a robust feature selection. The objective function of the RFS algorithm has its matrix formulation as follows:

$$\min_{\mathbf{W}} \left\| \mathbf{X}\mathbf{W} - \mathbf{Y} \right\|_{2,1} + \beta \left\| \mathbf{W} \right\|_{2,1} \Rightarrow \min_{\mathbf{W}} \frac{1}{\beta} \left\| \mathbf{X}\mathbf{W} - \mathbf{Y} \right\|_{2,1} + \left\| \mathbf{W} \right\|_{2,1}. \tag{4.3}$$

The problem in equation 4.3 is equivalent to

$$\min_{\mathbf{W},\mathbf{E}} \left\| \mathbf{E} \right\|_{2,1} + \left\| \mathbf{W} \right\|_{2,1} \quad s.t.\ \mathbf{X}\mathbf{W} + \beta\mathbf{E} = \mathbf{Y}. \tag{4.4}$$

Furthermore, the following formula can be obtained,

$$\min_{\mathbf{W},\mathbf{E}} \left\| \begin{bmatrix} \mathbf{W} \\ \mathbf{E} \end{bmatrix} \right\|_{2,1} \quad s.t.\ \begin{bmatrix} \mathbf{X}^T & \beta\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{W} \\ \mathbf{E} \end{bmatrix} = \mathbf{Y}, \tag{4.5}$$

where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is an identity matrix. Letting $\mathbf{U} = \begin{bmatrix} \mathbf{W} \\ \mathbf{E} \end{bmatrix} \in \mathbb{R}^{(n+d)\times c}$, $\mathbf{A} = \begin{bmatrix} \mathbf{X} & \beta\mathbf{I} \end{bmatrix} \in \mathbb{R}^{n\times(n+d)}$, problem 4.5 can be reformulated as

$$\min_{\mathbf{U}} \left\| \mathbf{U} \right\|_{2,1} \quad s.t.\ \mathbf{AU} = \mathbf{Y}. \tag{4.6}$$

Nie et al. (2010) propose the RFS algorithm to solve such joint $\ell_{2,1}$-norm minimization problems.

4.2 LSLM for Feature Selection. In the RFS (Nie et al., 2010) algorithm, the least squares loss between soft regression results and hard zero-one labels makes LSR less suitable for classification tasks. Inspired by the ReLSR, Zhang et al. (2015) proved the regression targets are learned from data by focusing on the relative values for requirement of correct classification with a large margin. In this section, we use the idea of a large margin to extend a novel method for feature selection; least squares regression with large margin for feature selection (LSLM-FS), and present an efficient algorithm to solve this problem. Finally, we analyze the optimization algorithm and the computation complexity.

4.2.1 Problem Formulation. In the method already described, we can see the feasibility and validity of the large margin constraint; thus, we add this constraint on the basis of the RFS model. Moreover, the $\ell_{2,1}$-norm is different from the F-norm in loss function, which can reduce the influence of outliers for function value, and the $\ell_{2,1}$-norm regularization selects features across all data points with joint sparsity. From the previous discussion, the objective function of the LSLM-FS model is as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{b},\mathbf{T}} \left\| \mathbf{XW} + \mathbf{e}_n\mathbf{b}^T - \mathbf{T} \right\|_{2,1} + \beta \left\| \mathbf{W} \right\|_{2,1}, \\ &\quad s.t.\ T_{i,y_i} - \max_{j\neq y_i} T_{i,j} \geq 1. \end{aligned} \tag{4.7}$$

The first term of equation 4.7 can be viewed as a loss function, while the second term is the regularization item. The target matrix $\mathbf{T}$ under the large margin constraint can improve the performance of classification.

4.2.2 An Iterative Algorithm to Solve Problem 4.7. We present an efficient iterative optimization algorithm to address the proposed problem with three variables in our letter:

Step 1. We fix $\mathbf{T}$ to optimize $\mathbf{W}$ and $\mathbf{b}$. Then problem of 4.7 is reduced to the following subproblem:

$$\min_{\mathbf{W},\mathbf{b}} \left\|\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^T - \mathbf{T}\right\|_{2,1} + \beta\left\|\mathbf{W}\right\|_{2,1}. \tag{4.8}$$

This model is similar to the RFS (Nie et al., 2010) algorithm. In the RFS algorithm, the bias is incorporated into the transformation matrix, while at the same time, a regularization term is added. However, the bias vector is a variable that needs to be optimized in the LSLMFS model. In order to solve problem 4.8 simply and effectively, we reformulate it in terms of homogeneous coordinates.

According to Xiang et al. (2012), with homogeneous coordinate, for $\mathbf{x}_i$, its homogeneous coordinate is defined as $\widetilde{\mathbf{x}}_i = [\mathbf{x}_i^T, u]^T$, where u is a scalar number. Let $\widetilde{\mathbf{W}} = [\mathbf{W}^T, \mathbf{b}]^T \in \mathbb{R}^{(d+1)\times c}$ and $\widetilde{\mathbf{X}} = [\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \ldots, \widetilde{\mathbf{x}}_n]^T \in \mathbb{R}^{n\times(d+1)}$. Under the circumstance of $u \to \infty$, solving equation 4.8 is equivalent to solving the following problem:

$$\min_{\widetilde{\mathbf{W}}} \left\|\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T}\right\|_{2,1} + \beta\left\|\widetilde{\mathbf{W}}\right\|_{2,1}. \tag{4.9}$$

If we get the optimal solution $\left\{\mathbf{W}^*, \mathbf{b}^*\right\}$ to equation 4.9, we will obtain the optimal solution $\left\{\mathbf{W}^*, u\mathbf{b}^*\right\}$ to equation 4.8.

Considering the definition of $\ell_{2,1}$-norm, we can obtain the derivative of the term $\left\|\widetilde{\mathbf{W}}\right\|_{2,1}$ and $\left\|\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T}\right\|_{2,1}$,

$$\frac{\partial\left\|\widetilde{\mathbf{W}}\right\|_{2,1}}{\partial\widetilde{\mathbf{W}}} = \mathbf{U}\widetilde{\mathbf{W}}, \quad \frac{\partial\left\|\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T}\right\|_{2,1}}{\partial\widetilde{\mathbf{W}}} = \widetilde{\mathbf{X}}^T\mathbf{D}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T}), \tag{4.10}$$

where $\mathbf{U}$ is the diagonal matrix in $\in \mathbb{R}^{(d+1)\times(d+1)}$ with the ith diagonal component U_{ii} equal to $1/\left\|\widetilde{\mathbf{w}}^i\right\|_2$. Here, $\widetilde{\mathbf{w}}^i$ is the ith row of $\widetilde{\mathbf{W}}$. $\mathbf{D}$ is diagonal matrix in $\in \mathbb{R}^{n\times n}$ with the ith diagonal component D_{ii}, which equals $1/\left\|\widetilde{\mathbf{x}}^T\widetilde{\mathbf{W}} - \mathbf{t}_i{}^T\right\|_2$. Here, $\mathbf{t}_i$ is the ith column of matrix $\mathbf{T}$.

We can obtain $\widetilde{\mathbf{W}}$ by setting the derivative of the objective function in equation 4.9 with respect to $\widetilde{\mathbf{W}}$ to zero:

$$\widetilde{\mathbf{X}}^T\mathbf{D}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T}) + \beta\mathbf{U}\widetilde{\mathbf{W}} = 0. \tag{4.11}$$

Thus, we arrive at

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^T\mathbf{D}\widetilde{\mathbf{X}} + \beta\mathbf{U})^{-1}\widetilde{\mathbf{X}}^T\mathbf{D}\mathbf{T}. \tag{4.12}$$

Step 2: Given $\mathbf{W}$ and $\mathbf{b}$, the problem can be rewritten as

$$\begin{aligned} &\min_{\mathbf{T}} \left\|\mathbf{X}\mathbf{W} + \mathbf{e}_n\mathbf{b}^T - \mathbf{T}\right\|_{2,1} = \left\|\mathbf{R} - \mathbf{T}\right\|_{2,1}, \\ &s.t.\ T_{i,y_i} - \max_{j\neq y_i} T_{i,j} \geq 1, \end{aligned} \tag{4.13}$$

where $\mathbf{R} = \mathbf{XW} + \mathbf{e}_n\mathbf{b}^T \in \mathbb{R}^{n\times c}$. Indeed, since the $\ell_{2,1}$-norm, problem 4.13 can be divided into n subproblems, with each subproblem corresponding to the learning of one row of $\mathbf{T}$. Thus, problem 4.13 can be decomposed into n subproblems with a general form as

$$\min_{\mathbf{t}^i} \left\|\mathbf{r}^i - \mathbf{t}^i\right\|_2^1 = \sqrt{\sum_{i=1}^{c}(r_{ij} - t_{ij})^2},$$
$$s.t.\ t_{i,y_i} - \max_{j\neq y_i} t_{i,j} \geq 1. \tag{4.14}$$

Note that functions $f(x) = \sqrt{x}$ and $g(x) = x$ have the same monotonicity for $x \geq 0$, and thus the solution of problem 4.14 is equivalent to the optimization of $\sum_{i=1}^{c}(r_{ij} - t_{ij})^2$ with respect to t_{ij}. Thus, problem 4.14 and ReLSR (Zhang et al., 2015) have the same solution.

Based on this analysis, the optimization problem in equation 4.7 has been solved. The steps of the optimization procedure are briefly given in algorithm 2.

4.2.3 Algorithm Analysis. We provide an iterative algorithm to solve problem 4.7 in algorithm 2 and obtain the optimal $\mathbf{W}$. Then we can select z features from d original features. First, we calculate the scores for all features $\left\|\mathbf{w}^i\right\|_2$, $(i = 1, 2, \ldots d)$. Then, we sort these scores and select the top z ranked features as the ultimate result. The convergence of algorithm 2 is proved to be similar to the convergence proof of algorithm 1.

4.2.4 Computational Complexity Analysis. In this part, we analyze the computational complexity of the LSLM-FS model in algorithm 2. It is not hard to deduce that the main computational costs are concentrated in the steps 10 and 15 to 18. The complexity of step 10 is $O((d+1)^2(n+c) + (d+1)nc)$. We need $O(ndc)$ to calculate step 15. Another computationally demanding operation needs to be performed in steps 16 to 18. The complexity is $O(nc)$. Suppose the number of iterations is k; thus, the maximum computational complexity of algorithm 2 is about $O(k((d+1)^2(n+c)))$.

5 Experiments

5.1 Experimental Results of OLSLM. In order to verify the performance of the OLSLM model, we compare our approach with traditional LSR, DLSR, ReLSR, OLSR, L1-SVM with hinge loss, L2-SVM with squared hinge loss, logistic regression (LR), and the multiclass SVM (MC-SVM) with multiclass hinge loss, on a range of real-world data sets. We give a brief introduction to all of the data sets used in our experiments. Then we introduce the setting of parameters and analyze the experimental results.

Algorithm 2: An Optimization Algorithm to Solve Problem 4.7.

Input:
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$;
$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T \in \mathbb{R}^{n \times c}$;
parameter β; large positive number u, and the maximum number of iterations $iters$, $\mathbf{I}_{d+1}$ and $\mathbf{I}_n$ are identity matrixs ;
Initialization:
$k = 1, u = 1000; \mathbf{W}_0 = \mathbf{0}, \mathbf{b}_0 = \mathbf{0}, \widetilde{\mathbf{W}}_0 = \mathbf{0}$;
$\mathbf{T}_0 = \mathbf{Y}$;
$\mathbf{U} = \mathbf{I}_{d+1}, \mathbf{D} = \mathbf{I}_n$;
while $k < iters$ **do**
  $\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^T \mathbf{D} \widetilde{\mathbf{X}} + \beta \mathbf{U})^{-1} \widetilde{\mathbf{X}}^T \mathbf{D} \mathbf{T}_0$;
  $U_{ii} = 1/\|\widetilde{\mathbf{w}}^i\|_2, i = 1, 2, ..., (d+1)$;
  $D_{ii} = 1/\|\widetilde{\mathbf{x}}_i{}^T \widetilde{\mathbf{W}} - \mathbf{t}_i{}^T\|_2, i = 1, 2, ..., n$;
  $\mathbf{W} = \widetilde{\mathbf{W}}(1{:}d, :)$, $\mathbf{b} = \widetilde{\mathbf{W}}(d{+}1{:}\text{end}, :)$;
  $\mathbf{b} = u\mathbf{b}$;
  $\mathbf{R} = \mathbf{X}\mathbf{W} + \mathbf{e}_n \mathbf{b}^T$;
  for $i = 1 \to n$ **do**
    $\mathbf{T}_{i*}^{(t)}$ =retargeting($\mathbf{R}_{i*}^{(t)}, y_i$) ;
  end
  if $(\|\boldsymbol{W} - \boldsymbol{W}_0\|_F^2 + \|\boldsymbol{b} - \boldsymbol{b}_0\|_F^2 < 10^{-4})$ **then**
    break ;
  end
  $\mathbf{W}_0 = \mathbf{W}$, $\mathbf{b}_0 = \mathbf{b}$, $\mathbf{T}_0 = \mathbf{T}$;
  $k = k + 1.$;
end
Output: $\mathbf{W}^* \in \mathbb{R}^{d \times c}$,$\mathbf{b}^* \in \mathbb{R}^{c \times 1}$

5.1.1 Data Sets. In our experiment, we use 12 public data sets to evaluate the performance of the proposed method. Table 1 presents some parameters of these data sets. The first six UCI data sets are taken from the UCI Machine Learning Repository.[1] The following six high-dimensional data sets are downloaded from the website.[2] The high-dimensional data sets are a palm print data set, and five face data sets (AR, Georgia Tech, CMU PIE, Yale, and YaleB).

The AR (Martinez, 1998) data set contains over 4000 color images corresponding to 126 people's faces; we use a subset of this data set: 100 persons,

[1] https://archive.ics.uci.edu/ml/datasets.html.
[2] http://www.face-rec.org/databases/.

Table 1: Brief Description of the Data Sets.

Data Sets	Classes	Features	Total Number	Train Number
cancer	2	9	683	274
cars	3	8	392	157
glass	6	9	214	86
heart	2	13	270	108
iris	3	4	150	60
vowel	11	13	990	396
AR	100	165×120 (17×12)	2600	800
GT	50	480×640 (15×20)	750	150
POSE07	68	64×64 (16×16)	1629	339
Yale	15	243×320 (15×20)	165	30
YaleB	38	32×32 (16×16)	2414	490
Palm	100	16×16 (16×16)	2000	200

each person with 26 color images. Georgia Tech (Chen, Man, & Nefian, 2005) contains images of 50 people. For each individual, there are 15 color images. Most of the images were taken in two sessions to take into account the variations in illumination conditions, facial expression, and appearance. The CMU PIE (Sim, Baker, & Bsat, 2002) data set contains 4,1368 images of 68 people under 13 poses, 4 expressions, and 43 illumination conditions; POSE07 is a subset of this data set. The Yale (Georghiades, Belhumeur, & Kriegman, 2001) data set consists of 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with/glasses, happy, left-light, with/no glasses, normal, right-light, sad, sleepy, surprised, and wink. The YaleB (Georghiades et al., 2001) data set is an extension of the Yale data set, which consists of 16,128 face images of 38 people. In this experiment, we chose the 2414 frontal images. Palm (Nie, Wang, & Huang, 2014) data set contains 2000 palm images of 100 classes. In the data sets, the color images should be converted to gray images. It is worth noting that in order to accelerate the speed of calculation, images from all of the data sets were resized to an equal proportional scale (shown in Table 1).

5.1.2 Parameter Settings. Recall the comments by Zhang et al. (2015) that the hyperparameter β in regression models such as LSR, DLSR, and ReLSR is a regularization parameter to avoid overfitting. We set it as

$$\beta = \widehat{\beta}\frac{1}{d}\mathbf{tr}(\mathbf{X}^T\mathbf{H}\mathbf{X}) \tag{5.1}$$

where $\mathbf{tr}(\cdot)$ is the trace of a matrix. Here, we use 10-fold cross-validation to select the optimal hyperparameter β by setting $\widehat{\beta}$ from the interval

[0:0.1:1] when the number of training samples was more than 200; the three-fold cross-validation approach was performed on the remaining data sets. For SVMs, there exists a regularization parameter C in LIBLINEAR.[3] We also use the above cross-validation to select it from the candidate set $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$.

In each experiment, we randomly select a small set of samples from each class for training. Table 1 lists the number of the training samples. The classifier is a one-nearest-neighbor classifier. We use recognition accuracy to evaluate the performance of our method in comparison with different models; the average classification accuracy and standard deviation are obtained by 10 random splits.

5.1.3 Experimental Results Analysis. Figure 1 shows the average classification accuracy and the standard deviation of different models on these 12 data sets. Figure 2 shows the convergence curve of OLSLM on all data sets. As we have seen, in most cases, our model gets better results than other approaches. Especially, in the AR, POSE07, Yale, and YaleB data sets, our method has outstanding performance compared with other methods. In addition, comparable performance is acquired compared with LSR and DLSR on the Palm and GT data sets, respectively.

Previously we proposed an efficient optimization algorithm to minimize the objective function value and have already proved the convergence on the theoretical level. In addition, we evaluated the convergence of our model on all the data sets used in our experiment. As Figure 2 shows, the objective function values consistently fall in the iterative process, which is consistent with the theoretical analysis. And the algorithm converges to a stable value within 50 iterations. The convergence is very fast and almost within 30 iterations.

In addition, we exhibit the averaged training time (per run) of our method and LSR and its variant methods on real data sets. The results are shown in Table 2. As we analyzed in section 3.2.2, calculating the transformation matrix by using the OLSR algorithm is inevitable, and it is performed in each iteration; thus, OLSLM takes longer.

5.2 Experimental Results of LSLM-FS. We evaluated the performance of the LSLM-FS model in a series of real-world high-dimensional data sets. We compared our method with six benchmark feature selection algorithms: T-test, RFS, CRFS, FS20, FS, and mRMR. We give a brief description of all data sets in Table 3. Then we introduce the parameter settings and analyze the experimental results.

[3] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/.

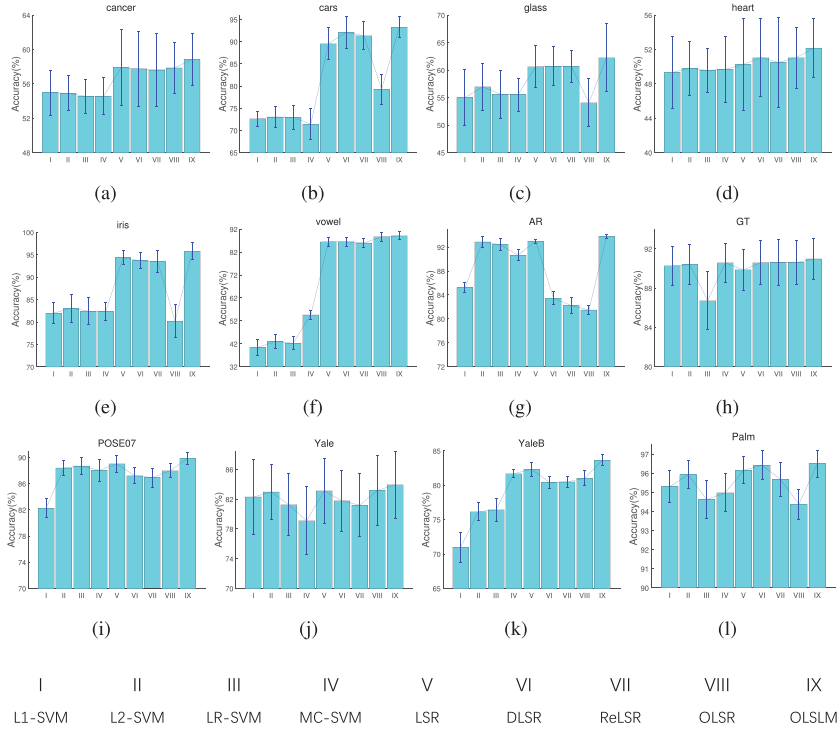

Figure 1: The classification accuracy and standard deviation of different methods on the cancer, cars, glass, heart, iris, vowel, AR, GT, Palm, POSE07, Yale, and YaleB data sets, respectively. (a) Cancer. (b) Cars. (c) Glass. (d) Heart. (e) Iris. (f) Vowel. (g) AR. (h) GT. (i) POSE07. (j) Yale. (k) YaleB. (l) Palm.

5.2.1 Data Sets. In order to test the performance of our proposed approach, we used nine public data sets from different fields. Among them, the two data sets listed in Table 1 are also included.

The Coil20 (Nene, Nater, & Murase, 1996) data set includes 20 objects, each of which has 72 gray images taken from different view directions. The LUNG (Bhattacharjee et al., 2001) data set contains 203 samples in five classes. Genes with standard deviations smaller than 50 expression units were removed, which produces a data set with 203 samples and 3312 genes. Details about the Faces95,[4] PIE10P, CLL-SUB-111, TOX,[5] and WebKB-WC[6] data sets are in Table 2.

[4] http://cswww.essex.ac.uk/mv/allfaces/index.html.
[5] http://featureselection.asu.edu/datasets.php.
[6] http://images.ee.umist.ac.uk/danny/database.html.

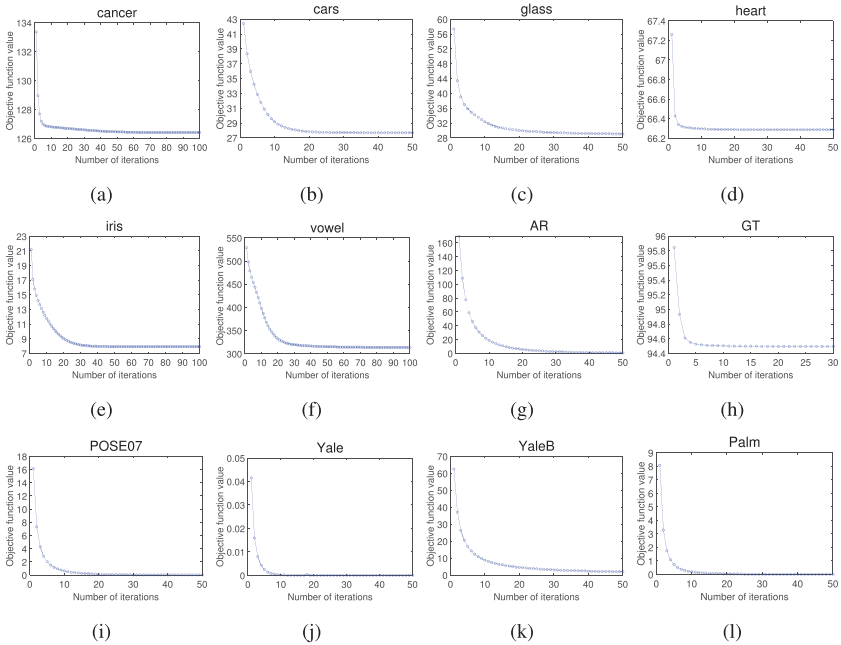

Figure 2: The convergence curve of OLSLM on the cancer, cars, glass, heart, iris, vowel, AR, GT, POSE07, Yale, YaleB, and Palm data sets, respectively. (a) Cancer. (b) Cars. (c) Glass. (d) Heart. (e) Iris. (f) Vowel. (g) AR. (h) GT. (i) POSE07. (j) Yale. (k) YaleB. (l) Palm.

Table 2: Training Time (Seconds) on 12 Real Data Sets.

Data Sets	LSR	DLSR	ReLSR	OLSR	OLSLM
cancer	0.031	0.047	0.047	0.031	0.125
cars	0.031	0.047	0.047	0.078	0.094
glass	0	0	0.031	0.063	0.109
heart	0	0.016	0.047	0.047	0.078
iris	0.031	0.063	0.047	0.047	0.078
vowel	0.047	0.078	0.063	0.047	0.219
AR	0.156	0.218	0.578	0.203	3.766
GT	0.031	0.031	0.109	0.141	1.484
POSE07	0	0.063	0.141	0.141	2.625
Yale	0	0.031	0	0.171	2.125
YaleB	0.063	0.141	0.094	0.203	3.125
Palm	0.016	0.125	0.156	0.172	1.844

Table 3: Brief Description of the Data Sets.

Data Sets	Classes	Features	Total Number	Train Number
Faces95	72	$200 \times 180(20 \times 18)$	1440	576
PIE10P	10	55×44	210	80
Yale	15	$243 \times 320(27 \times 36)$	165	60
YaleB	38	$32 \times 32(16 \times 16)$	2414	978
Coil20	20	$128 \times 128(16 \times 16)$	1440	580
CLL-SUB-111	3	11340	111	44
LUNG	5	3312	203	81
TOX	4	5748	171	69
WebKB-WC	7	4189	1210	484

5.2.2 Parameter Settings. We compared our algorithm with several typical feature selection algorithms, including the Fisher score (FS), the minimum redundancy maximum (mRMR), the student's T-test (T-test), RFS, CRFS, and FS20. The LSLM-FS model has a parameter β that needs to be tuned. The feature selection performance is evaluated by average classification accuracy using the LibSVM classifier. Here, we use 10-fold cross-validation to select a proper β and the regularization parameter C of LibSVM when there are more than 200 training samples. A three-fold cross-validation approach is performed on the remaining data sets. We tune them by grid search from $\beta \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$ and $C \in \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. For all competitors, we also tune parameters repeatedly and report the results under the best parameter settings. In each experiment, we randomly select a percentage of samples from each class for training. Table 3 lists the number of the training samples. We use recognition accuracy to evaluate the performance of our comparison method of different models, the average classification accuracy and standard deviation are obtained by 10 random splits.

5.2.3 Experimental Results Analysis. The accuracy curves of all feature selection methods on the nine data sets are shown in Figure 3. The final classification accuracy is calculated as the average of the 10 trials.

Compared with the traditional feature selection algorithms, including FS, T-test, and mRMR, our algorithm can achieve higher mean accuracies on almost all data sets with different numbers of selected features. In most cases, our method has obvious improvements compared with RFS, which indicates that the idea of the large margin is feasible for improving classification ability. The CRFS algorithm also achieves very promising results. For some data sets, in particular TOX and WebKB-WC, we can see that when the number of selected features is small, CRFS outperforms LSLM-FS. However, as shown in Figure 3, with the increasing number of selected features, our algorithm achieves higher mean accuracies, which is better than the

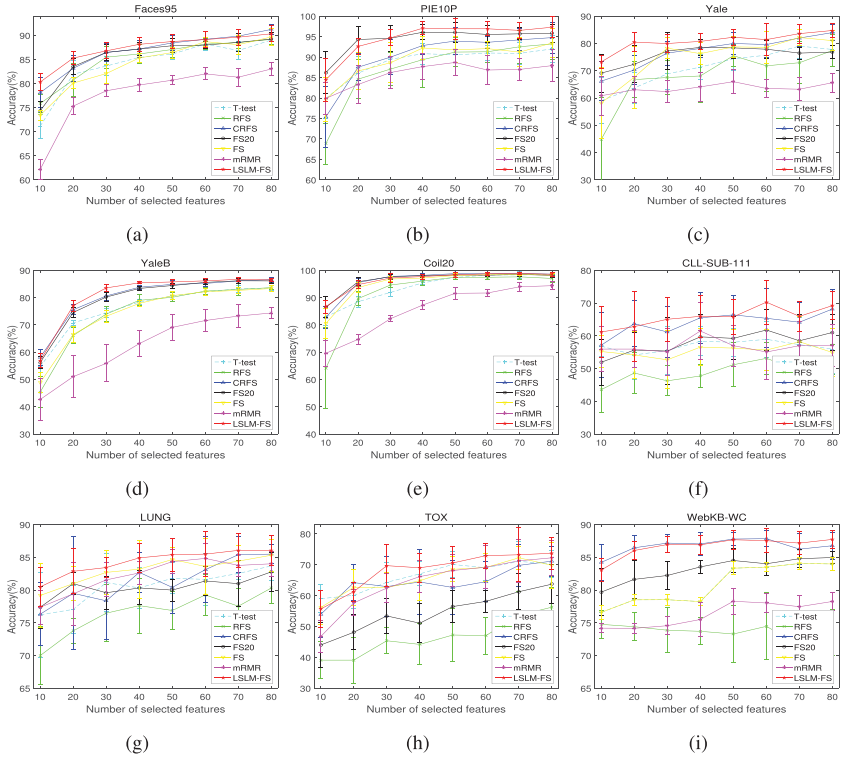

Figure 3: Classification accuracy and standard deviation of different methods on Faces95, PIE10P, Yale, YaleB, Coil20, CLL-SUB-111, LUNG, TOX, and WebKB-WC data sets with different numbers of selected features. Eight selected features are evaluated, as indicated by the horizontal axis. (a) Faces95. (b) PIE10P. (c) Yale. (d) YaleB. (e) Coil20. (f) CLL-SUB-111. (g) LUNG. (h) TOX. (i) WebKB-WC.

CRFS algorithm. In the PIE10P data set, our algorithm, compared with FS20, generates slightly low accuracy under a low dimension.

Table 4 reports the average training time of running one split with the seven algorithms. As seen from the results in Table 4, compared with the three classic algorithms—T-test, FS, and mRMR, our method cost less time on most data sets, particular in high-dimensional data sets. The remaining algorithms take more time as the remaining algorithms are implemented iteratively, especially the FS20 algorithm, because this method needs to iterate several times to converge.

Table 4: Training Time (Seconds) on Nine Real Data Sets.

Data Sets	T-test	FS	mRMR	RFS	CRFS	FS20	LSLM-FS
Faces95	13.219	0.672	0.703	0.031	0.078	0.094	1.641
PIE10P	1.611	1.156	2.297	0.031	0.688	0.016	0.141
Yale	1.578	0.656	2.297	0.031	0.094	0.016	0.031
YaleB	4.406	0.406	0.941	0.063	0.031	0.266	3.281
Coil20	0.984	0.234	0.794	0.047	0.062	0.218	0.984
CLL-SUB-111	0.531	1.859	2.766	0.031	39.734	0.094	0.281
LUNG	0.500	0.641	2.313	0.031	1.391	0.047	0.219
TOX	0.531	1.406	2.453	0.047	6.672	0.031	0.703
WebKB-WC	0.578	1.109	3.406	0.031	6.750	0.031	0.218

6 Conclusion

In this letter, we propose two novel algorithms: the orthogonal least squares regression with large margin (OLSLM) algorithm for multiclass classification and the least squares regression with large margin for feature selection (LSLM-FS). The OLSLM is different from most traditional LSR and its extensions. The proposed method makes use of orthogonal characteristic and soft labels information to boost the ability of classification. The core idea is to impose the large margin constraint on an OLSR model to replace the hard labels with relative values. Thus, our method can preserve the local discriminant information and guarantee that each sample can be correctly classified with large margins simultaneously. An efficient iterative algorithm is also proposed to solve the optimization problem. In addition, the idea of a large margin is applied to explore a novel feature selection method. The LSLM-FS adopts the $\ell_{2,1}$-norm on both loss function and regularization to produce a sparse learning model under the large margin constraint condition, and an efficient iterative algorithm is proposed to solve the optimization problem as well.

The OLSLM model, however, is sensitive to noise, that is, it cannot suppress the interference of noise. We can use other norms to replace the F-norm to ensure that the model can be more robust in the future. In addition, many graph-based methods have been widely used recently. Thus one part of our future work is to incorporate the theory of graph into our framework.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China (61402002, 61502002, 61300057); the Project Sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (48,2014-1685); the Natural Science Foundation of Anhui Province (1408085QF120, 1408085MKL94); the Key Natural

Science Project of Anhui Provincial Education Department (KJ2016A040); and Open Project of IAT Collaborative Innovation Center of Anhui University (ADXXBZ201511).

References

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., & Vasa, P., . . . Meyerson, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(24), 13790–13795.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer-Verlag.

Bradley, P. S., & Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector Machines. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 82–90). New York: ACM.

Cai, D., He, X., Han, J., & Zhang, H. J. (2006). Orthogonal Laplacianfaces for face recognition. *IEEE Transactions on Image Processing*, *15*(11), 3608–3614.

Cai, X., Nie, F., & Huang, H. (2013). Exact top-k feature selection via $\ell_{2,0}$-norm constraint. In *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 1240–1246). San Francisco: Morgan Kaufmann.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machine. *ACM*, *2*(3), 1–27.

Chang, K. W., Hsieh, C. J., & Lin, C. J. (2008). Coordinate descent method for large-scale L2-loss linear support vector machines. *Journal of Machine Learning Research*, *9*(3), 1369–1398.

Chen, L., Man, H., & Nefian, A. V. (2005). Face recognition based on multi-class mapping of Fisher scores. *Pattern Recognition*, *38*(6), 799–811.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, *3*(2), 185–205.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York: Wiley.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, *3* 1289–1305.

Georghiades, A. S., Belhumeur, P. N., & Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(6), 643–660.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). The elements of statistical learning. *Mathematical Intelligencer*, *27*(2), 83–85.

He, R., Tan, T., Wang, L., & Zheng, W. S. (2012). $\ell_{2,1}$ Regularized correntropy for robust feature selection. *Computer Vision and Pattern Recognition*, *157*(10), 2504–2511.

Hoerl, A. E., & Kennard, R. W. (2000). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*, 55–67.

Hosmer Jr, D. W., & Lemeshow, S. (2004). Applied logistic regression. *Journal of the American Statistical Association*, *85*(411), 81–82.

Hou, C., Nie, F., Li, X., Yi, D., & Wu, Y. (2014). Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Transactions on Cybernetics*, *44*(6), 793–805.

Hou, C., Nie, F., Zhang, C., Yi, D., & Wu, Y. (2014). Multiple rank multi-linear SVM for matrix data classification. *Pattern Recognition*, *47*(1), 454–469.

Hou, C., Zhang, C., Wu, Y., & Jiao, Y. (2009). Stable local dimensionality reduction approaches. *Pattern Recognition*, *42*(9), 2054–2066.

Jolliffe, I. T. (2005). *Principal component analysis*. Berlin: Springer-Verlag.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324.

Kokiopoulou, E., & Saad, Y. (2007). Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2143–2156.

Lan, G., Hou, C., & Yi, D. (2016). Robust feature selection via simultaneous capped ℓ_2–norm and $\ell_{2,1}$–norm minimization. In *Proceedings of the IEEE International Conference on Big Data Analysis* (pp. 1–5). Piscataway, NJ: IEEE.

Martinez, A. M. (1998). *The AR face database* (CVC Technical Report, 24) Barcelona: Computer Vision Center.

Montgomery, D. C., Runger, G. C., & Hubele, N. F. (2007). *Engineering statistics* (4th ed.). Hoboken, NJ: Wiley.

Nene, S., Nater, S., & Murase, H. (1996). *Columbia Object Image Library (COIL-20)*. New York: Columbia University.

Nie, F., Huang, H., Cai, X., & Ding, C. (2010). Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Neural information processing systems*, *23* (pp. 1813–1821). Red Hook, NY: Curran.

Nie, F., Wang, X., & Huang, H. (2014). Clustering and projected clustering with adaptive neighbors. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 977–986). New York: ACM.

Nie, F., Xiang, S., Liu, Y., Hou, C., & Zhang, C. (2012). Orthogonal vs. uncorrelated least squares discriminant analysis for feature extraction. *Pattern Recognition Letters*, *33*(5), 485–491.

Nie, F., Xiang, S., Song, Y., & Zhang, C. (2009). Orthogonal locality minimizing globality maximizing projections for feature extraction. *Optical Engineering*, *48*(1), 017202 (1–5).

Nie, F., Zhang, R., & Li, X. (2017). A generalized power iteration method for solving quadratic problem on the Stiefel manifold. *Science China*, *60*(11), 112101.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238.

Raileanu, L. E., & Stoffel, K. (2004). Theoretical comparison between the Gini Index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, *41*(1), 77–93.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Russell, E. L., Chiang, L. H., & Braatz, R. D. (2000). *Fisher discriminant analysis*. London: Springer.

Saul, L. K., & Roweis, S. T. (2003). Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, *4*(2), 119–155.

Sim, T., Baker, S., & Bsat, M. (2002). The CMU pose, illumination, and expression (PIE) database. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 46–51). Piscataway, NJ: IEEE.

Tang, Z., Cui, Y., & Jiang, B. (2017). Groupwise registration of MR brain images containing tumors via spatially constrained low-rank based image recovery. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 397–405). Cham: Springer.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society*, *58*(3), 267–288.

Wang, H., Nie, F., Huang, H., Kim, S., Nho, K., Risacher, S. L., & Shen, L. (2012). Identifying quantitative trait loci via group-sparse multitask regression and feature selection: An imaging genetics study of the ADNI cohort. *Bioinformatics*, *28*(2), 229–237.

Wang, H., Nie, F., Huang, H., Risacher, S., Ding, C., Saykin. A. J., & Shen, L. (2011). Sparse multi-task regression and feature selection to identify brain imaging predictors for memory performance. In *Proceedings of the International Conference on Computer Vision* (pp. 557–562). Washington, DC: IEEE Computer Society.

Wang, L., Zhu, J., & Zou, H. (2007). Hybrid Huberized support vector machines for microarray classification. *Bioinformatics*, *24*, 983–990.

Xiang, S., Nie, F., Meng, G., Pan, C., & Zhang, C. (2012). Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(11), 1738–1754.

Yang, S., Hou, C., Nie, F., & Wu, Y. (2012). Unsupervised maximum margin feature selection via $\ell_{2,1}$-norm minimization. *Neural Computing and Applications*, *21*(7), 1791–1799.

Yu, J. (2012). Local and global principal component analysis for process monitoring. *Journal of Process Control*, *22*(7), 1358–1373.

Zhang, X. Y., Wang, L., Xiang, S., & Liu, C. L. (2015). Retargeted least squares regression algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(9), 2206–2213.

Zhao, H., Wang, Z., & Nie, F. (2016). Orthogonal least squares regression for feature extraction. *Neurocomputing*, *216*, 200–207.

Received December 24, 2017; accepted May 6, 2018.