Top-$k$ Supervise Feature Selection via ADMM for Integer Programming

Mingyu Fan$^1$, Xiaojun Chang$^2$, Xiaoqin Zhang$^3$, Di Wang$^1$, Liang Du$^3$

$^1$School of Maths & Info. Science, Wenzhou University, Wenzhou 325035, China
$^2$School of Computer Science, Carnegie Mellon University, PA 15213, USA
$^3$School of Computer & Information Technology, Shanxi University, Taiyuan 030006 China

{fanmingyu,xqzhang,wangdi}@wzu.edu.cn, cxj273@gmail.com, duliang@ios.ac.cn

Abstract

Recently, structured sparsity-inducing based feature selection has become a hot topic in machine learning and pattern recognition. Most of the sparsity-inducing feature selection methods are designed to rank all features by certain criterion and then select the $k$ top-ranked features, where $k$ is an integer. However, the $k$ top features are usually not the top $k$ features and therefore maybe a suboptimal result. In this paper, we propose a novel supervised feature selection method to directly identify the top $k$ features. The new method is formulated as a classic regularized least squares regression model with two groups of variables. The problem with respect to one group of the variables turn out to be a 0-1 integer programming, which had been considered very hard to solve. To address this, we utilize an efficient optimization method to solve the integer programming, which first replaces the discrete 0-1 constraints with two continuous constraints and then utilizes the alternating direction method of multipliers to optimize the equivalent problem. The obtained result is the top subset with $k$ features under the proposed criterion rather than the subset of $k$ top features. Experiments have been conducted on benchmark data sets to show the effectiveness of proposed method.

1 Introduction

In modern machine learning and pattern recognition applications, data are commonly represented by high dimensional feature vectors, such as image classification [Chatfield et al., 2011; Chang et al., 2016] and video recognition [Lan et al., 2015; Chang et al., 2017]. High dimensional data are not suitable for directly learning because the time cost and storage requirement will be very high. Furthermore, they usually contain many noise and redundant features [Peng et al., 2005] that could degrade the generalization capability of the learning algorithms. Feature Selection (FS) [Guyon and Elisseeff, 2003] is designed to identify the most relevant and important features that can give a compact and accurate data representation for learning. Compared with other feature analysis methods, such as feature extraction [Belhumeur et al., 1997], FS has better interpretability because it keeps the semantic meaning of the features. Also, the cost of feature collection can be reduced because that one only needs to collect the selected features in FS rather than all the features as in feature extraction does. As a result, FS has become a hot topic in machine learning and pattern recognition [Chang et al., 2014].

Based on the rule of the learning algorithm, there are roughly three types of FS methods in literature: the filter-type, the wrapper-type, and the embedded-type methods. The filter-type method evaluates all data features based on certain criteria, where no learning algorithm is involved. Representative filter-type feature selection methods include the reliefF [Kira and Rendell, 1992], mRMR [Peng et al., 2005], Fisher score [Duda et al., 2000], and Laplace score [He et al., 2006] methods. The wrapper-type method applies a classifier as a black box to score the features. The widely used wrapper-type methods include the Support Vector Machine Recursive Feature Elimination (SVM-RFE) [Guyon et al., 2002] and the Correlation-based Feature Selection (CFS) [Hall and Smith, 1999]. The embedded-type method [Wang et al., 2007; Argyriou et al., 2007] embeds the feature selection procedure in a classifier algorithm and only a single optimization problem is involved. Because the embedded and wrapper methods interact with a learning algorithm, they tend to achieve better classification results than the filter-type method when some specific learner is involved for evaluation.

Recently, empirical studies of sparse representation and compressed sensing [Elad, 2010] indicate that sparsity is one of the basic and intrinsic properties of real world data. F-S, which finds a sparse attributes to represent the input data, can be regarded as a natural application of the sparse representation theory. A large number of FS methods resort to the sparsity-inducing regularization terms/constraints, such as the $\ell_0$, $\ell_1$-norm, $\ell_{0,2}$-norm and $\ell_{1,2}$-norm based penalty terms/constraints, to achieve feature evaluation and selection (The appearance of the notations may differ with those used in previous papers, but they are essentially equivalent). From the sparsity perspective, $\ell_0$-norm and $\ell_{0,2}$-norm are more desirable to select the features because that they can induce the sparsest solution, i.e., each feature should be associated with either the zero score or a large score. However, $\ell_0$-norm and $\ell_{0,2}$-norm regularized/constrained optimization
tion problems have been proved to be NP-hard [Amaldi and Kann, 1998] and are very difficult to solve. Fortunately, theoretical results show that, under mild conditions, \( \ell_1 \) and \( \ell_0 \) are essentially equivalent [Donoho, 2004], i.e., \( \ell_1 \)-norm and \( \ell_1,2 \)-norm can be regarded as efficient approximations to \( \ell_0 \)-norm and \( \ell_0,2 \)-norm respectively. The \( \ell_1 \)-norm based problem, also known as Lasso [Tibshirani, 1994], is commonly used in FS on binary class data sets. Destrezo et al. utilize the Lagrangian form of Lasso for feature selection in face recognition [Destrezo et al., 2009]. In [Zou and Hastie, 2005], the elastic net regularization is proposed to handle feature redundancy, group Lasso is introduced to integrate the feature structure and then evaluate the importance of features, where the structures include the disjoint groups [Zhang et al., 2012], the overlapping groups [Jenatton et al., 2011], and so on. To address multi-class problems, Nie et al. [Nie et al., 2010] propose to apply \( \ell_{1,2} \)-norm instead of \( \ell_1 \)-norm as the penalty and have shown promising results. Many recent FS methods are proposed in the form of \( \ell_1,2 \)-norm regularized/constrained optimization problems [Xiang et al., 2012; Du and Shen, 2015; Han et al., 2015]. The matrix norm has been extended to \( \ell_{p,2} \), \( (p \in (0, 1)) \) [Wang et al., 2014; Tao et al., 2016] and \( \ell_{p,r} \), \( (1 < r) \) norms for robust FS. Because \( p = 0 \) is more desirable than any \( p > 0 \), an exact top-k FS via an optimization problem with the \( \ell_{0,2} \)-norm constraint is proposed in [Cai et al., 2013].

In this paper, we propose a novel efficient and robust supervised FS method, which has the following properties.

1. The proposed method is formulated as a problem with two groups of variables and an \( \ell_{0,2} \)-norm constraint. The sub-problem with respect to one group of the variables in the model is a 0-1 integer programming and the sub-problem of the other group of the variables admits a closed-form solution.

2. The 0-1 integer programming of our method is firstly transformed into an equivalent optimization problem with two additional continuous constraints. The equivalent problem is shown able to be efficiently solved by the Alternating Direction Method of Multipliers (ADMM). The Matlab code is published online\(^1\).

3. We provide an efficient algorithm to solve \( \ell_{0,2} \)-constrained supervised FS method. The proposed method guarantees to select the top \( k \) features instead of \( k \) top features under the proposed criterion. Experiments show that the proposed method has superior performance than the compared state-of-the-art supervised FS methods.

The rest of this paper is structured as follows: in Section 2, we review some related sparsity-inducing FS methods. The proposed supervised FS method is described in Section 3. Experimental comparisons with state-of-the-art supervised FS methods on benchmark data sets are presented in Section 4. Finally, the conclusion is drawn in Section 5.

\(^1\)https://github.com/cxj273/IJCAI2017_1274

2 Sparsity-Inducing Supervised Feature Selection Background

2.1 Notations and Definitions

The \( \ell_p \)-norm of a vector \( v \in \mathbb{R}^D \) is defined as \( \|v\|_p = \left( \sum_{i=1}^D |v_i|^p \right)^{1/p} \), where \( v_i \) denotes the \( i \)-th entry in \( v \). The \( \ell_0 \)-norm of a vector \( v \) is defined as \( \|v\|_0 = \sum_{i=1}^D |v_i|^0 \), i.e., the counts of nonzero entries in \( v \). \( \text{diag}(v) \in \mathbb{R}^{D \times D} \) is a diagonal matrix whose diagonal elements are the entries of vector \( v \) and \( \text{diag}(\Theta) \in \mathbb{R}^{D \times D} \) is a \( D \)-dimensional vector consists of the diagonal elements of the matrix \( \Theta \). The \( \ell_{p,r} \)-norm of a matrix \( A \in \mathbb{R}^{C \times D} \) is defined as

\[
\|A\|_{p,r} = \left( \left( \|A_1\|_r, \cdots, \|A_C\|_r \right) \right)^{1/p}
\]

where \( A_j \) \( (j = 1, \cdots, D) \) denotes the \( j \)-th column of \( A \). Consequently, the \( \|A\|_{0,2} \) is naturally defined as \( \|A\|_{0,2} = \sum_{j=1}^D \|A_j\|_2^0 \) which counts the number of nonzero columns in \( A \).

Let \( X = [x_1, \cdots, x_N] \in \mathbb{R}^{D \times N} \) be the input data matrix, where \( D \) is the input dimensionality, \( N \) is the number of data points, the \( i \)-th column, \( x_i \), denotes a data vector. For presentation clarity and simplicity, we assume that the constant value 1 has been added to the bottom of each data vector as an additional dimension and thus the bias term can be omitted throughout this paper.

2.2 Sparsity-Inducing Supervised Feature Selection

Given a binary class data set, many sparsity-inducing FS methods can be interpreted as the approximation or relaxed version of the following problem

\[
A^* = \arg \min_A \|y - AX\|_2^2, \quad \text{s.t.} \quad \|A\|_0 = k, \quad (1)
\]

where \( A \in \mathbb{R}^D, y \in \mathbb{R}^N \) and its \( i \)-th entry \( y_i \in \{0, 1\} \) is the class label of \( x_i \). The \( \ell_0 \)-norm constraint means only \( k \) entries are non-zeros and thus only \( k \) features of \( X \) are used. The purpose of FS is to find out which \( k \) features are effective. We can also write the problem (1) in the equivalent Lagrangian form:

\[
A^* = \arg \min_A \left\{ \|y - AX\|_2^2 + \gamma \|A\|_0 \right\}, \quad (2)
\]

where nonnegative \( \gamma \) is a given tradeoff parameter. However, the problems (1) and (2) have been proven to be NP-hard problems and are computationally infeasible. Many FS methods [Destrezo et al., 2009; Cai et al., 2010] replace \( \ell_0 \)-norm with its convex surrogate \( \ell_1 \)-norm and have shown promising results. Generally, the values of entries in \( A \) indicate the importance of data features. The features with high values in \( A \) are then selected.

For multi-class data set, the ideal top-k FS [Cai et al., 2013] is to optimize

\[
A^* = \arg \min_A \|Y - AX\|_{1,2}, \quad \text{s.t.} \quad \|A\|_{0,2} = k, \quad (3)
\]
or its equivalent form
\[
A^* = \arg \min_A \{ \| Y - AX \|_{1,2} + \gamma \| A \|_{0,2} \}. \tag{4}
\]
where \( A \in \mathbb{R}^{C \times D} \) be the projection matrix, \( Y \in \mathbb{R}^{C \times N} \) be the label matrix with the \( i \)-th column \( y_i \in \mathbb{R}^C \). If \( x_i \) is in the \( k \)-th class, the \( k \)-th entry of \( y_i \) is 1 and the rest entries are 0s. However, \( \ell_{0.2} \)-norm regularized/constrained optimization problem is still NP hard and is computationally infeasible. To address this problem, matrix \( \ell_{p,r} \)-norm has been proposed for inducing column sparsity to achieve feature selection, where \( p \in (0,1) \) and \( r > 1 \). The objective function of \( \ell_{1,2} \) based Robust Feature Selection (RFS) [Nie et al., 2010] is proposed as
\[
A^* = \arg \min_A \{ \| Y - AX \|_{1,2} + \gamma \| A \|_{1,2} \},
\]
To enforce further sparsity, [Wang et al., 2014] proposes the \( \ell_{p,2} \)-norm based FS method.
\[
A^* = \arg \min_A \{ \| Y - AX \|_{p,2} + \gamma \| A \|_{p,2} \},
\]
where \( p \in (0,1) \) is a given parameter.

3 Top-k Supervised Feature Selection
3.1 Formulation
Both \( \ell_{p,2} \)-norm and \( \ell_{p} \)-norm are surrogates of their original sparsity-inducing constraints, \( \ell_{0.2} \)-norm and \( \ell_{0} \)-norm respectively. Therefore, it is more desirable to optimize the original sparsest problem instead of its relaxed formulation. In this paper, we propose a novel method that directly solves the s-sparsest feature selection model. The proposed method does not score every features, but selects the optimal feature set with \( k \) features under the proposed criteria. For convenience, the classical multi-class least squares regression model is utilized to learn the linear projection matrix \( A \) as
\[
A^* = \arg \min_A \| Y - AX \|_F^2 + \gamma \| A \|_F^2, \tag{6}
\]
\[\text{s.t. } \| A \|_{0,2} = k.\]
As can be seen, the proposed model is a classical regularized least squares regression with an additional \( \ell_{0.2} \)-norm based constraint. The \( \ell_{0.2} \)-norm constraint requires \( k \) and only \( k \) columns of the matrix \( A \) are not zero vectors. Therefore, the \( k \) corresponding features of \( X \) are active and the rest features hibernate in the regression (6). Instead of directly solving the \( \ell_{0.2} \) constrained problem, we first transform it to an equivalent \( \ell_{0} \)-norm constrained problem as follows:
\[
\{ A^*, v^* \} = \arg \min_A \| Y - \text{Adiag}(v)X \|_F^2 + \gamma \| A \|_F^2, \tag{7}
\]
\[\text{s.t. } \| v \|_0 = k, \text{ and } v \in \{0,1\}^D.\]
Once let \( \hat{A} = \text{Adiag}(v) = [v_1A_1, \ldots, v_D A_D] \), we can see that the FS problems (6) and (7) are essentially equivalent. With respect to the variable \( A \), the problem (7) has the closed-form least squares solution. However, the problem with respect to the variable \( v \) is a 0-1 integer programming and is generally very difficult to solve. Here we utilize the \( \ell_{2,1} \)-box method to efficiently address this problem [Wu and Ghanem, 2016]. The binary constraint \( \{0,1\}^D \) can be replaced with an equivalent set of continuous constraints, i.e., the intersection of a box and a shifted \( \ell_{2} \)-sphere. The result is presented in Proposition 3.1.

Proposition 3.1 [Wu and Ghanem, 2016] Let \( \mathbf{1}_D \in \mathbb{R}^D \) be the vector whose entries are all 1s, we have
\[
v \in \{0,1\}^D \iff \{ v \in [0,1]^D \} \cap \left\{ v : \| v - \frac{\mathbf{1}_D}{2} \|_2^2 = \frac{D}{4} \right\}. \tag{8}
\]
An illustrative example of the equivalence between the binary constraint and the continuous constraints in 2-D space is given in Figure 3.1. Based on Proposition 3.1, we can obtain the following problem which is equivalent to (7).
\[
\{ A^*, v^* \} = \arg \min_A \| Y - \text{Adiag}(v)X \|_F^2 + \gamma \| A \|_F^2, \tag{8}
\]
\[\text{s.t. } \mathbf{1}_D^Tv = k, \quad v = v_1, \quad v = v_2, \quad v_1 \in S_b \text{ and } v_2 \in S_p\]
where the two sets \( S_b = \left\{ v : v \in [0,1]^D \right\} \) and \( S_p = \left\{ v : \| v - \frac{\mathbf{1}_D}{2} \|_2^2 = \frac{D}{4} \right\} \). In (8), the two continuous constraints in Proposition 3.1 are separated by two additional variables \( v_1 \) and \( v_2 \). The problem (8) can now be efficiently optimized by the ADMM method.

3.2 Optimization
ADMM method has been widely used in convex optimization, and there is also growing interests and applications on the advantage of ADMM in non-convex optimization. In this section, we study an efficient solution for our FS method (8) using the ADMM method, which solves the following sub-problems iteratively:

1. Fix the other variables, optimize \( A \) through solving a classic least squares regression problem.
2. Given \( A \), compute \( v \) through solving an unconstrained quadratic optimization problem.
3. Project \( v \) on to \( S_b \) and \( S_p \) to obtain \( v_1 \) and \( v_2 \) respectively.

4. Update the Lagrange multipliers.

Using a parameter \( \rho > 0 \), the augmented Lagrangian function of (8) is obtained as

\[
L(A, v, v_1, v_2, y_1, y_2, y_3) = \| A \text{diag}(v) X - Y \|^2_F + \gamma \| A \|^2_F + y^T(v - v_1) + y_2^T(v - v_2) + y_3(1_D^T v - k) + \frac{\rho}{2} \left( \| v - v_1 \|^2_F + \| v - v_2 \|^2_F + (1_D^T v - k)^2 \right),
\]

where \( y_1 \in \mathbb{R}^D \), \( y_2 \in \mathbb{R}^D \) and \( y_3 \in \mathbb{R} \) are Lagrange multipliers for the three equality constraints. The ADMM approach then iteratively optimize the variables individually. Denote by \( (A(t), v(t), v_1(t), v_2(t)) \) the optimization variables at iteration \( t \), and by \( (y_1(t), y_2(t), y_3(t)) \) the Lagrange multipliers at iteration \( t \).

**Step 1:** Solving the linear projection matrix \( A \) when other variables are fixed. (9) with respect to \( A \) is a classical least squares regression problem and the solution can be directly provided as (at the \( t + 1 \)-th iteration)

\[
A(t+1) = Y X^T \text{diag}(v(t)) \left( \text{diag}(v(t)) X X^T \text{diag}(v(t)) + \gamma I \right)^{-1}
\]

**Step 2:** Optimize \( v \) when other variables are fixed. After some mathematical deductions, we can obtain the following unconstrained quadratic optimization problem with respect to the variable \( v \) as

\[
\min_v v^T \left( \Phi \odot (\Psi(t+1))^T \right) v - 2v^T \text{diag}(\Theta(t+1)) + \frac{\rho}{2} \left( \| v - v_1(t) \|_2^2 + \| v - v_2(t) \|_2^2 \right) + (1_D^T v - k + y_3^T)^2
\]

where \( \Phi = XX^T, \Psi(t+1) = (A(t+1))^T A(t+1), \Theta(t+1) = XY^T A(t+1) \). Imposing the derivative of the objective function with respect to \( v \) to zero, we obtain the closed-form solution as

\[
v(t+1) = \left( 2 \Phi \odot (\Psi(t+1))^T + \rho (1_D 1_D^T + 2I) \right)^{-1} \left( 2 \text{diag}(\Psi(t+1)) + \rho \left( (v_1(t) - y_1(t)) / \rho + (v_2(t) - y_2(t)) / \rho \right) + (k - y_3(t) / \rho) 1_D \right)
\]

**Step 3:** Update variables \( v_1 \) and \( v_2 \) through projections onto \( S_b \) and \( S_p \). The variables are updated as follows

\[
\begin{align*}
v_1(t+1) &= P_{S_b}(v(t) + y_1(t) / \rho) \\
v_2(t+1) &= P_{S_p}(v(t) + y_2(t) / \rho)
\end{align*}
\]

For any vector \( x \in \mathbb{R}^D \), one first compute the two candidates of the projection on \( S_p \) as

\[
\begin{align*}
x_1 &= \frac{1_D}{2} + \left( \frac{\sqrt{D}}{2\|z - 1D/2\|} \right) (z - 1D/2) \\
x_2 &= \frac{1_D}{2} - \left( \frac{\sqrt{D}}{2\|z - 1D/2\|} \right) (z - 1D/2)
\end{align*}
\]

Then, the projection of \( x \) on \( S_p \) can be computed as

\[
P_{S_p}(x) = \begin{cases} 
  x_1 & \text{if } \| x - x_1 \| < \| x - x_2 \| \\
  x_2 & \text{otherwise}.
\end{cases}
\]

**Step 4:** Update variables \( y_1, y_2, y_3 \) and \( \rho \). Having variables \( A(t+1), v(t+1), v_1(t+1), v_2(t+1) \) fixed, the Lagrange multipliers and step size \( \rho \) are updated as follows

\[
\begin{align*}
y_1(t+1) &= y_1(t) + \rho (v(t) - v_1(t)) \\
y_2(t+1) &= y_2(t) + \rho (v(t) - v_2(t)) \\
y_3(t+1) &= y_3(t) + \rho (1_D^T v(t) - k)
\end{align*}
\]

where \( \mu > 1 \) is a given parameter.

3.3 Computational Complexity Analysis

To optimize the objective function of the proposed method, the step 1 and step 2 are time consuming operations. At step 1, the algorithm needs to inverse an \( D \times D \) matrix and the time complexity is \( O(D^3) \). At step 2, an matrix inversion is computed on a \( D \times D \) matrix whose time complexity is \( O(D^3) \). Assuming there are \( T \) iterations before the algorithm stops, the total cost of the proposed method is \( O(T(D^3)) \). As can be seen, the computation complexity is comparably higher than most of the sparsity-inducing supervised FS methods. We plan to reduce the computation complexity of the proposed method in the future work.
4 Experiments

In this section, the proposed method is compared with state-of-the-art supervised feature selection methods on benchmark image data sets. The experiments include the supervised classification by the Nearest Neighbor classifier (1-NN) and the Support Vector Machine (SVM) under various experimental settings. The numerical convergence analysis of the proposed method is also included.

4.1 Datasets Description

Three real world data sets are used in our experiments. The important statistics of these data sets are briefly summarized as bellow:

- The Coil-20 data set\(^2\) contains 1440 image samples from 20 classes and each image is transformed into a 1024-dimensional data point. There are 72 samples in each class.
- The MNIST handwritten digital image data set\(^3\) has 6996 data points of digits ‘0’ - ‘9’. Each sample is a 784 dimensional feature vector.
- There are 2114 frontal-face images of 38 individuals in the Yale-B face image data set\(^4\). Each image is stacked to a 1024-dimensional data vector.

\(^3\)http://www.escience.cn/people/fpnie/
\(^4\)http://www.cad.zju.edu.cn/home/dengcai/Data/faceData.html

4.2 Experimental Setup

The following state-of-the-art supervised feature selection algorithms are compared in this paper. The Fisher score method [Duda et al., 2000] evaluates each features independently by computing the score using the Fisher criterion. The Spectral method [Zhao and Liu, 2007] for supervised feature selection. The Relieff method [Kira and Rendell, 1992] for multi-class supervised feature selection. Robust Feature Selection (RFS) [Nie et al., 2010] selects features by solving an $\ell_{1,2}$-norm regularized regression problem. The $\ell_{0,2}$-FS method [Cai et al., 2013] which exactly selects the top $k$ features in the supervised scenarios. The Discriminative Least Squares Regression (DLSR) [Xiang et al., 2012] which takes the $\ell_{1,2}$-norm regularized least squares regression formulation. Also, the original data with all features for classification is compared as the baseline.

The methods, Fisher, Relieff and $\ell_{0,2}$-FS, are parameter free. The Spectral method requires the neighborhood size as a key parameter, which is tuned in the range $\{4, 6, 8, 10\}$. The regularization parameter $\lambda_A$ for the RFS and DLSR methods is searched in the range $\{0.001, 0.05, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}$. The best results are reported with these parameters. To make our results reproducible, the regularization parameter $\gamma = 0.2$ is used for our method throughout the experiments.

4.3 Convergence Analysis

To solve the proposed formulation, we develop an iterative update algorithm. It is important to provide the experimental
study of the convergence of the proposed method. The convergence rates on the Coil-20 data set are shown in Fig. 2. As can be seen from Fig. 2(b), most of the entries of vector $v$ are either 0 or 1. Only a few entries lie in the range $[0, 1]$. This means the proposed method is effective in solving $0 - 1$ integer programming. Fig. 2(a) presents that if an entry of $v$ is 0, the corresponding column of matrix $A$ would be a zero vector. We can see from Fig. 2(c) that the proposed method converges within 50 iterations, demonstrating that the proposed optimization algorithm is effective. There is a low-lying pit between the 3rd and the 10th iterations. This is because that at the early stage of the algorithm, the impact of the continuous constraints in our method is trivial, i.e., few of the entries of $v$ are 0 or 1 or even in the range of $[0, 1]$. The problem is essentially equivalent to a classic least squares at the early steps. After some iterations, the two continuous constraints start exert significant influence on the objective function and then the objective value can reasonably rise.

4.4 Performance Evaluation

Given a data set, we randomly select $p$ percents from each class to formulate the training data $X_{train}$, and the remaining data are used as the test data. The supervised feature selection methods are performed on $X_{train}$ to rank the features. The classifiers, both 1-NN and SVM, are trained on $X_{train}$ represented by the selected features and then tested on the test data. For each setting, the experiments are repeated 10 times and both the mean of results and the deviation variances are reported.

Fig. 4 shows the plots of classification performance versus the number of selected features on the data sets, Coil-20, MNIST and Yale-B. The percentage of labeled training data is $p = 30$. As can be seen, our proposed method consistently outperforms or show comparable performance with the compared FS methods. The accuracy curves of the proposed method converge very fast, with typically around 80 features.

The filter-type FS methods (Fisher, Spectral and ReliefF) evaluate features individually and do not consider the relevance among them. Therefore, the accuracies of these methods are generally lower than those obtained by group sparsity-inducing based FS methods (RFS, $\ell_{0,2}$-FS, DLSR and our method). Our proposed method can efficiently remove noise and redundant features and get comparably better performance with fewer features. By comparing with the $\ell_{0,2}$ FS method, which also selects the exact number of the top features, it is observed that our method is superior to $\ell_{0,2}$ FS on the data sets.

Due to the limited pages, the results on additional data sets and the results when the percentage of training data $p = 50$ are not shown here. The readers are encouraged to try other data sets with the provided Matlab code.

4.5 Parameter Sensitivity

The proposed method requires one parameter $\gamma$ to be set in advance. In this subsection, we discuss the sensitivity of the proposed method over this parameter. The parameter $\gamma$ is searched in the range of $[0, 3]$. 50 percents of data in each class are used as the training data ($p = 50$) and the top 200 features are utilized. The results on Coil-20 and YaleB data sets are presented in Fig. 3. As can be seen, both large and small $\gamma$ degrades the performance of the proposed method. On the other hand, any $\gamma \in [0.2, 0.5]$ seems to be able to give reasonable results. We set $\gamma = 0.2$ and do not tune the parameter in the experiments for different data sets under different experimental settings.

5 Conclusion

In this paper, we propose a novel supervised FS method to identify the best $k$ features rather than the $k$ top features. The proposed method is a classic least squares under the $\ell_0$ or $\ell_{0,2}$ constraint, where the non-smoothed constraint means the number of selected features. We transformed the original integer programming problem into an optimization problem with two continuous constraints. And then the problem is efficiently solved by ADMM method. It is shown that the proposed method outperforms the state-of-the-art supervised FS method on benchmark image data sets.

Acknowledgments

This work is supported by NSFC (Grants nos. 61472285, 61473212, 61503263, 61502289, 61511130084), Zhejiang Provincial Natural Science Foundation (Grants nos. LY15F030011, LY17F030004 and LR17F030001), Project of science and technology plans of Zhejiang Province (Grants no. 2015C31168). Project of science and technology plans of Wenzhou City (Grants Nos. G20150017, G20160002).
References


Figure 4: 1-NN and SVM classification results of the comparing feature selection methods with 30% percents training data the on (a) Coil20(1-NN), (b) MNIST(1-NN), (c) YaleB(1-NN), (d) Coil20(SVM), (e) MNIST(SVM), and (f) YaleB(SVM) data sets. (1-NN and SVM in the legend means the classification results on all features)