Embedded Supervised Feature Selection for Multi-class Data

Lin Chen* Jiliang Tang† Baoxin Li*

Abstract

Supervised multi-class learning arises in many application domains such as biology, computer vision, social network analysis, and information retrieval. These applications often involve high-dimensional data, which not only significantly increase the time and space requirement of the underlying algorithms but also degrade their performance due to the curse of dimensionality. Feature selection has been proven effective and efficient for preparing high-dimensional data for many learning tasks. Traditional feature selection algorithms for multi-class data assume the independence of label categories and select features with the capability to distinguish samples from different classes. However, class labels in multi-class data may be correlated and little work exists for exploiting label correlation in multi-class feature selection. In this paper, we investigate label correlation in feature selection for multi-class data. In particular, we provide a principled approach for capturing label correlation and propose an Embedded Supervised Feature Selection (ESFS) framework, which embeds label correlation modeling in supervised feature selection for multi-class data. Experiments on both synthetic data and various types of public benchmark datasets show that the proposed framework effectively captures the multi-class label correlation and significantly outperforms existing state-of-the-art baseline methods.

1 Introduction

Multi-class data is ubiquitous in many domains such as biology, computer vision, information retrieval and social network analysis[38]. For example, in object recognition, images may be labeled as one of tens or hundreds of object categories [39] and texts in information retrieval could belong to one of tens of text categories [40]. Meanwhile, such data is usually high dimensional. For example, there are millions of genes in biology applications while the word dictionaries in information retrieval could have tens of thousands of words, which not only increases the time and space requirement of algorithms but also degrades the application performance due to the curse of dimensionality. Feature selection, selecting a subset of most relevant features for a compact and accurate presentation, is proven to be an effective and efficient way to handle high-dimensional.

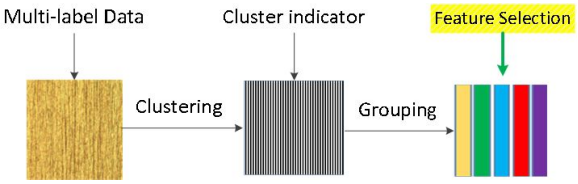
(a) Object recognition

(b) Webpage categorization

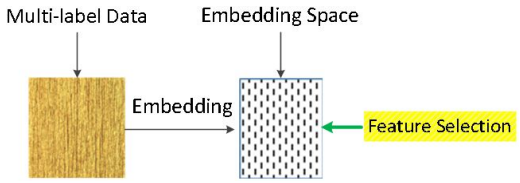
Figure 1: Examples of Multi-Label Correlation.

Supervised feature selection can be roughly divided into filter methods, wrapper methods and embedded methods. The vast majority of traditional feature selection algorithms for multi-class data treat class labels independently and select features with capability to distinguish samples from different classes. However, class labels for multi-class data in real-world applications are usually correlated in groups and each group of labels share certain features. One example is representing objects by geometric descriptors in visual object recognition as shown in Fig. 1(a) where objects are class labels. Different groups of objects (or labels) are with similar shapes inside each group while with different shapes across different groups. For example, "basketballs" and "oranges" are "spherical" while "shells" and "foldable fans" are "fan-shaped". Another example is webpage categorization as demonstrated in Fig. 1(b) where categories are class labels. For example, categories of "U.S.","world" and "local" usually share keywords related to society issues such as "charity", "economy" and "crime", while categories of "Tech" or "Science" usually share keywords like "technology" and "invention". Selecting features by treating all these class labels as individuals may not attain the best possible performance due to the existence of the correlation among class labels. Class label correlation has been explored and successfully employed in variety of applications including shape analysis [33], genes expression analysis [31] and semantic visual attribute prediction [24]. These success-

*Arizona State University, {lin.chen.cs,baoxin.li}@asu.edu
†Michigan State University, tangjili@msu.edu



(a) Traditional framework



(b) Proposed framework

Figure 2: Multi-class Feature Selection Framework

ful experiences indicate that modeling label correlation has potentials in helping feature selection; however, little work exists for modeling label correlation in feature selection for multi-class data.

In this paper, we study the problem of exploiting label correlation in feature selection for multi-class data. To achieve this goal, we provide solutions to the following two challenges: (a) how to exploit label correlation for multi-class data mathematically; and (b) how to use it for feature selection in a supervised scenario, which results in a novel embedded supervised feature selection framework ESFS that combines label correlation learning and feature selection into a coherent model. Different from the traditional multi-class feature selection approaches that first divide the class labels into different groups and then apply feature selection on the grouped classes (Fig. 2(a)), our proposed framework aims to directly learn a latent space that captures the multi-class label correlation and feature selection is directly imposed on the learned latent space.

The main contributions of this paper are summarized as below. First, we provide a principled approach to learning a latent space that captures label correlation for feature selection in multi-class data. Second, we propose an embedded supervised feature selection framework ESFS for multi-class data, which embeds the feature selection in the process of label correlation learning. Last but not least, we conduct experiments on both synthetic data and various types of real-world datasets to demonstrate the effectiveness of the proposed framework.

2 Related work

Depending on the criterion adopted to measure the relevance of features, supervised feature selection can be roughly categorized into three groups - filter methods, wrapper methods and embedded methods. Filter methods filter out irrelevant features before classification by scoring and ranking the features by some ranking criterion. For example, [18] ranks the features by calculating the Pearson correlation coefficients between the variable and the class label; [36] estimates Mutual Information (MI) through Kullback-Leibler divergence to rank features; [25] develops a ranking criterion based on class densities for binary data. Wrapper methods employ a learning method as a black box and the selected features are based on the performance from the learning approach. Since too many feature subsets need to be evaluated (making the problem NP-hard), the learning approach is wrapped on a search algorithm that finds the subset giving the best performance. The Branch and Bound method [26] evaluates different feature subsets through a tree structure. The Adaptive Sequential Forward Floating Selection [34] algorithm selects features sequentially by first adding one feature giving the best performance, and then iteratively adding features from the rest features based on the performance. Genetic Algorithm [14] heuristically searches the subset of features maximizing the predictor performance wherein the chromosome bits are used to represent if the feature is included or not. However, these methods are usually computationally expensive and may not apply well on problems with large-scale data.

Embedded methods [3] "embed" feature selection as a part of the learning process without spiting the data into training and testing datasets. [29] proposes a two-stage approach by first selecting the number of features and then evaluating different subsets of features based on the performance feedback. [19] ranks the features by the weight of the SVM classifier through conducting sensitivity analysis, wherein the change in the weight can be viewed as removing a feature. [32] utilizes multilayer perceptron networks as the classifier and calculates the feature weight using a saliency measure calculated from the trained network. Recently sparsity regularization such as $\ell_{2,1}$ of matrix in dimensionality reduction has been widely investigated and also applied to feature selection[17]. [2] proposes a convex optimization approach for feature selection by $\ell_{2,1}$ group sparse regularization; [15] selects features by a non-convex multi-stage sparse based approach; [16] imposes an additional $\ell_{2,1}$-norm on row space to detect outliers. [7, 8] utilize the relatedness among different learning tasks for feature selection. The above selection approaches consider each class independently during learning.However, in reality, class labels in multi-class data could be correlated. The proposed approach employs the multi-class label correlation for feature selection, leading to better performance compared to several representative base-

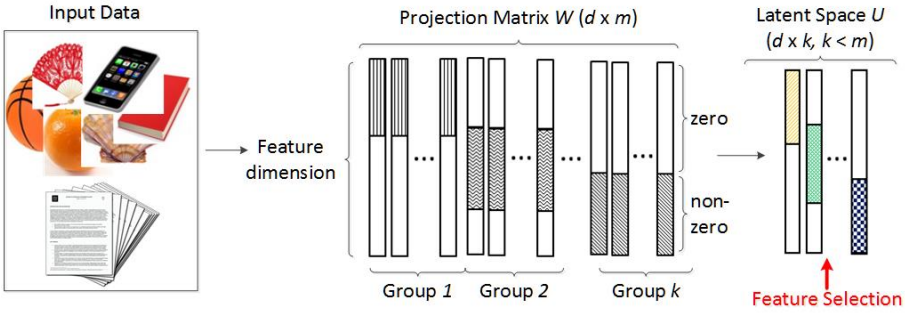


Figure 3: Concept Overview of the embedded supervised multi-class feature selection framework.

line methods.

3 Proposed Framework

Let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ denotes the multi-class dataset with n samples and d features $\mathcal{F} = \{f_1, f_2, \ldots, f_d\}$, let $Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n] \in \{0,1\}^{m\times n}$ denotes the corresponding label matrix of m classes $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$, we aim to select $K(K \leq d)$ most relevant features from $\mathcal{F}$ by leveraging X, Y and the label correlation in $\mathcal{C}$.

Let $\mathbf{s} = \pi(\overbrace{0, \ldots, 0}^{d-K}, \overbrace{1, \ldots, 1}^{K})$, where $\pi(\cdot)$ is the permutation function and K is the number of features to select where $\mathbf{s}_i = 1$ indicates that the i-th feature is selected. The original data can be represented as $\text{diag}(\mathbf{s})\mathbf{X}$ with K selected features where $\text{diag}(\mathbf{s})$ is a diagonal matrix. We assume that a linear projection matrix $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m] \in \mathbb{R}^{d\times m}$ maps the data X to its label matrix Y where $\boldsymbol{w}_i \in \mathbb{R}^d (i = 1, 2, \ldots, m)$ is the projection vector for the i-th class c_i. If we do not consider label correlation, we can select K features via solving the following optimization problem:

$$\begin{aligned} \arg\min_{W,\mathbf{s}} \quad & L(W^T \text{diag}(\mathbf{s})X, Y) \\ s.t., \quad & \mathbf{s} \in \{0,1\}^d, \ \mathbf{s}^T\mathbf{1}_d = K \end{aligned} \tag{3.1}$$

where $L(\cdot)$ is the loss function and typical choices of loss functions include least square and logistic regression.

3.1 Modeling Attributes Correlation In real-world multi-class applications, class labels might be correlated and they may form several clusters (or groups) [24]. Therefore we can model label correlation via learning clustering structures of labels. A partition of the projection matrix W into k clusters can be formed under the non-negative matrix factorization framework as:

$$\begin{aligned} \arg\min_{U,V} \ & \|W - UV^\top\|_F^2 \\ s.t. \quad & V \in \{0,1\}^{m\times k}, V^\top\mathbf{1}_m = \mathbf{1}_k \end{aligned} \tag{3.2}$$

where $U \in \mathbb{R}^{d\times k}$ is the latent feature matrix and $V \in \mathbb{R}^{m\times k}$ is the cluster indicator. The problem in Eq. 3.2 is difficult to solve due to the constraint on V. Thus we relax the constraint on the label indicator matrix V to orthogonality following [28]. After the relaxation, Eq. 3.2 can be rewritten as:

$$\begin{aligned} \arg\min_{U,V} \ & \|W - UV^\top\|_F^2 \\ s.t. \quad & V^\top V = I, V \geq 0 \end{aligned} \tag{3.3}$$

To capture the correlation among labels, similar labels should been partitioned into the same group. Inspired by the spectral analysis [28], we further add the following term to force similar label are clustered in the same group:

$$\min \mathbf{Tr}(V^\top L V) \tag{3.4}$$

where $L = D - S$ is the Laplacian matrix and D is a diagonal matrix with its elements defined as $D_{ii} = \sum_{j=1}^{m} S_{ij}$. $S \in \mathbb{R}^{m\times m}$ denotes the similarity matrix based on W, which is obtained through RBF kernel as

$$S_{ij} = e^{-\frac{\|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2}{\sigma^2}}$$

Combing Eq. (3.3) and (3.4), the label correlation can be modeled as

$$\begin{aligned} \arg\min_{U,V} \ & \|W - UV^\top\|_F^2 + \beta\mathbf{Tr}(V^\top L V) \\ s.t. \quad & V^\top V = I, V \geq 0 \end{aligned} \tag{3.5}$$

3.2 Feature Selection With the model component to capture label correlation in Eq. (3.5), the proposed embedded supervised feature selection framework (ESFS) for multi-class data is to solve the following optimization problem:

$$\begin{aligned} \arg\min_{W,F,\mathbf{s}} \ & L(W^T \text{diag}(\mathbf{s})X, Y) + \alpha\|W - UV^\top\|_F^2 \\ & + \beta\mathbf{Tr}(V^\top L V) \\ s.t. \quad & V^\top V = I, V \geq 0 \\ & \mathbf{s} \in \{0,1\}^d, \ \mathbf{s}^T\mathbf{1}_d = K \end{aligned} \tag{3.6}$$

According to [37], the feature selection on W can be relaxed to perform feature selection on U by ignoring the selection matrix on W. Thus Eq. (3.6) is equivalent to the following optimization problem:

$$\begin{aligned} \arg\min_{W,F,\mathbf{s}} & L(W^\top X, Y) + \alpha\|W - \mathrm{diag}(\mathbf{s})UV^\top\|_F^2 \\ & + \beta\mathbf{Tr}(V^\top LV) \\ s.t. \quad & V^\top V = I, V \geq 0 \\ & \mathbf{s} \in \{0,1\}^d,\ \mathbf{s}^T\mathbf{1}_d = K \end{aligned} \tag{3.7}$$

The constraint on $\mathbf{s}$ makes Eq. (3.7) a mixed integer programming problem, which is difficult to solve. We observe that $\mathrm{diag}(\mathbf{s})$ and U is as the form of $U\mathrm{diag}(\mathbf{s})$. Since $\mathbf{s}$ is a binary vector and $d-K$ rows of the $\mathrm{diag}(\mathbf{s})$ are all zeros, $U\mathrm{diag}(\mathbf{s})$ is a matrix where the elements of many rows are all zeros. This motivates us to absorb $\mathrm{diag}(\mathbf{s})$ into U as $W = U\mathrm{diag}(\mathbf{s})$, and add $\ell_{2,1}$-norm on U to ensure the sparsity of U in rows and achieve feature selection. With this relaxation, Eq. (3.7) can be rewritten as:

$$\begin{aligned} \arg\min_{W,U,V} & L(W^\top X, Y) + \alpha\|W - UV^\top\|_F^2 \\ & + \beta\mathbf{Tr}(V^\top LV) + \gamma\|U\|_{2,1} \\ s.t. \quad & V^\top V = I, V \geq 0 \end{aligned} \tag{3.8}$$

Since U is forced to sparse where some rows are close to $\mathbf{0}$, some instances of W that poorly reconstruct from U and V. These instances from the decomposition regularizer would dominate the objective function because of the squared errors. To make the model robust to these instances, we replace the decomposition regularizer by $\ell_{2,1}$-norm. Without loss of the generality, we adopt the traditional least square loss for demonstration in the following paper. The objective function of the proposed framework becomes

$$\begin{aligned} \arg\min_{W,U,V} & \|W^\top X - Y\|_F^2 + \alpha\|W - UV^\top\|_{2,1} \\ & + \beta\mathbf{Tr}(V^\top LV) + \gamma\|U\|_{2,1} \\ s.t. \quad & V^\top V = I, V \geq 0 \end{aligned} \tag{3.9}$$

4 Algorithm

We first introduce the optimization algorithm and then give an analysis of the proposed algorithm.

4.1 Optimization The objective function in Eq. (3.9) is convex if we update the variables U, V and W alternatively. Following [21], we use Alternating Direction Method of Multiplier (ADMM) [4] to optimize the objective function. By introducing two auxiliary variables $E = W - UV^\top$ and $Z = V$, we can convert Eq. (3.9) into the following equivalent problem:

$$\begin{aligned} \arg\min_{W,U,V,E,Z} & \|W^\top X - Y\|_F^2 + \alpha\|E\|_{2,1} \\ & + \beta\mathbf{Tr}(V^\top LV) + \gamma\|U\|_{2,1} \\ s.t. \quad & E = W - UV^\top, \\ & Z = V, V^\top V = I, Z \geq 0 \end{aligned} \tag{4.10}$$

which is equivalent to solve the following ADMM problem

$$\begin{aligned} \arg\min_{W,U,V,E,Z,Y_1,Y_2,\mu} & \|W^\top X - Y\|_F^2 + \alpha\|E\|_{2,1} \\ & + \beta\mathbf{Tr}(Z^\top LV) + \gamma\|U\|_{2,1} \\ & + \mathbf{Tr}(Y_1^\top(Z - V)) \\ & + \mathbf{Tr}(Y_2^\top(W - UV^\top - E)) \\ & + \frac{\mu}{2}(\|Z - V\|_F^2 + \|W - UV^\top - E\|_F^2) \\ s.t. \quad & V^\top V = I, Z \geq 0 \end{aligned} \tag{4.11}$$

where Y_1, Y_2 are two Lagrangian multipliers and μ is a scalar to control the penalty for the violation of equality constraints $E = W - UV^\top$ and $Z = V$.

4.1.1 Update E By fixing the other variables except E, Eq. (4.11) can be reformed as follows by removing terms that are irrelevant to E:

$$\arg\min_E \frac{1}{2}\|E - (W - UV^\top + \frac{1}{\mu}Y_2)\|_F^2 + \frac{\alpha}{\mu}\|E\|_{2,1} \tag{4.12}$$

This problem can be solved according to the following lemma due to [5].

LEMMA 4.1. *Let $Q = [\boldsymbol{q}_1; \boldsymbol{q}_2; \cdots; \boldsymbol{q}_m]$ be a given matrix and λ a positive scalar. If the optimal solution of*

$$\arg\min_W \frac{1}{2}\|W - Q\|_F^2 + \lambda\|W\|_{2,1}$$

is W^*, then the i-th row of W^* is

$$\boldsymbol{w}_i^* = \begin{cases} (1 - \frac{\lambda}{\|\boldsymbol{q}_i\|})\boldsymbol{q}_i, & if \|\boldsymbol{q}_i\| > \lambda \\ 0, & otherwise \end{cases} \tag{4.13}$$

Thus, let $Q = W - UV^\top + \frac{1}{\mu}Y_2$, E can be updated as follows according Lemma 4.1:

$$\boldsymbol{e}_i = \begin{cases} (1 - \frac{\alpha}{\mu\|\boldsymbol{q}_i\|})\boldsymbol{q}_i, & if \|\boldsymbol{q}_i\| > \frac{\alpha}{\mu} \\ 0, & otherwise \end{cases} \tag{4.14}$$

4.1.2 Update U By fixing the other variables except U and remove terms that are irrelevant to U, Eq. (4.11) becomes

$$\arg\min_{U} \frac{\mu}{2}\|W - UV^\top - E + \frac{1}{\mu}Y_2\|_F^2 + \gamma\|U\|_{2,1} \tag{4.15}$$

Since $V^\top V = I$, Eq. (4.15) can be rewritten as

$$\arg\min_{U} \frac{1}{2}\|U - (W - E + \frac{1}{\mu}Y_2)V\|_F^2 + \frac{\gamma}{\mu}\|U\|_{2,1}$$

According to Lemma 4.1, let $K = (W - E + \frac{1}{\mu}Y_2)V$, U can be updated as

$$\boldsymbol{u}_i = \begin{cases} (1 - \frac{\gamma}{\mu\|\boldsymbol{k}_i\|})\boldsymbol{k}_i, & if \|\boldsymbol{k}_i\| > \frac{\gamma}{\mu} \\ 0, & otherwise \end{cases} \tag{4.16}$$

4.1.3 Update Z Similarly, to update Z, we fix the other variables and remove terms irrelevant to Z, which makes Eq. (4.11) become

$$\begin{aligned} \arg\min_{Z;Z\geq 0} & \frac{\mu}{2}\|Z - V\|_F^2 + \beta\mathbf{Tr}(Z^\top LV) \\ & + \mathbf{Tr}(Y_1^\top(Z - V)) \end{aligned} \tag{4.17}$$

Let $T = V - \frac{1}{\mu}Y_1 - \frac{\beta}{\mu}LV$, Eq. (4.17) can be written as

$$\arg\min_{Z;Z\geq 0} \|Z - (V - \frac{1}{\mu}Y_1 - \frac{\beta}{\mu}LV)\|_F^2$$

This is equivalent to the following element-wise optimization problem

$$\arg\min_{Z_{ij};Z_{ij}\geq 0} \|Z_{ij} - T_{ij}\|^2$$

where the optimal solution is achieved by

$$Z_{ij} = \max(T_{ij}, 0)$$

4.1.4 Update V By removing terms irrelevant to V and fixing other variables, Eq. (4.11) becomes

$$\begin{aligned} \arg\min_{V;V^\top V=I} & \mathbf{Tr}(Y_1^\top(Z - V)) + \beta\mathbf{Tr}(Z^\top LV) \\ & + \frac{\mu}{2}(\|Z - V\|_F^2 + \|W - UV^\top - E\|_F^2) \\ & + \mathbf{Tr}(Y_2^\top(W - UV^\top - E)) \end{aligned} \tag{4.18}$$

Let

$$N = \frac{1}{\mu}Y_1 + Z - \frac{\beta}{\mu}L^\top Z + (W - E + \frac{1}{\mu}Y_2)^\top U, \tag{4.19}$$

utilizing $V^\top V = I$, Eq. (4.18) can be further written as

$$\arg\min_{V,V^\top V=I} \|V - N\|_F^2 \tag{4.20}$$

This problem can be solved according to the following lemma due to [21]:

LEMMA 4.2. *Let P and Q are the left and right singular vectors of the economic singular value decomposition (SVD) of N where $N = P\Sigma Q$, the optimal V of the objective function in Eq. (4.20) is defined as*

$$V = PQ^\top$$

4.1.5 Update W Removing terms irrelevant to W and fixing other variables, we rewrite Eq. (4.11) as

$$\begin{aligned} F(W) = \arg\min_{W} & \|W^\top X - Y\|_F^2 + \beta\mathbf{Tr}(Z^\top LV) \\ & + \mathbf{Tr}(Y_2^\top(W - UV^\top - E)) \\ & + \frac{\mu}{2}\|W - UV^\top - E\|_F^2 \end{aligned} \tag{4.21}$$

Let $\tilde{U} = UV^\top - E$ and $\tilde{V} = VZ^\top$, the gradient of 4.21 corresponding to $\boldsymbol{w}_i$ can be represented as

$$\begin{aligned} \nabla\frac{F(W)}{\boldsymbol{w}_i} = & \boldsymbol{y}_{2,i} + \mu(\boldsymbol{w}_i - \tilde{\boldsymbol{u}}_i) + \sum_{j=1}^{n}(2\boldsymbol{x}_j(\boldsymbol{x}_j^\top)\boldsymbol{w}_i - y_{ij}) \\ & + \beta\sum_{j=1;j\neq i}^{m}((\tilde{v}_{ii} - \tilde{v}_{ij} - \tilde{v}_{ji})\frac{2(\boldsymbol{w}_j - \boldsymbol{w}_i)}{\sigma^2}e^{-\frac{\|\boldsymbol{w}_i - \boldsymbol{w}_j\|^2}{\sigma^2}}) \end{aligned}$$

where $\boldsymbol{y}_{2,i}$, $\boldsymbol{w}_i$, $\tilde{\boldsymbol{u}}_i$ and $\boldsymbol{x}_i$ are the i-th column vector of matrix Y_2, W, $\tilde{U}$ and X; y_{ij} and $\tilde{v}_{ij}$ are the (i,j)-th element of matrix Y and $\tilde{V}$.

Thus, the new weight vector $\boldsymbol{w}_i'$ can be updated by gradient descent

$$\boldsymbol{w}_i' = \boldsymbol{w}_i - \eta\nabla\frac{F(W)}{\boldsymbol{w}_i}$$

where η is the step size.

4.1.6 Update Y_1,Y_2 **and** μ According to [4], the ADMM parameters can be updated as

$$\begin{aligned} Y_1 &= Y_1 + \mu(Z - V) \\ Y_2 &= Y_2 + \mu(W - UV^\top - E) \\ \mu &= \min(\rho\mu, \mu_{max}) \end{aligned} \tag{4.22}$$

where $\rho > 1$ is a parameter controlling the convergence speed and μ_{max} is a large number preventing μ becomes too large.

Following these updating rules, the proposed algorithm is summarized in Algorithm 1. The importance of the i-th feature is indicated by $\|\boldsymbol{u}^i\|_2$. Therefore, we rank features in descending order according to $\|\boldsymbol{u}^i\|_2$ and select the top-K ranked ones.

Algorithm 1 The Proposed Feature Selection Framework for Multi-class Data

Input:

1. Multi-class data $\{X, Y\}$;
2. Parameters α, β, γ, k and the number of selected features K;
3. The initial projection matrix W_0;

Output:

Top-K selected features;

Initialize $W = W_0$, $\mu = 10^{-3}$, $\rho = 1.1$, $\mu_{max} = 10^{10}$, $U = 0$, $V = 0$ (or initialized by K-means);
repeat
  Calculate $Q = W - UV^\top + \frac{1}{\mu}Y_2$ and update E according to Eq. (4.14);
  Calculate $K = (W - E + \frac{1}{\mu}Y_2)V$ and update U according to Eq. (4.16);
  Calculate $T = V - \frac{1}{\mu}Y_1 - \frac{\beta}{\mu}LV$ and update Z according to Eq. (4.18);
  Calculate N according to Eq. 4.19 and update V according to Lemma 4.2;
  Update Y_1,Y_2,μ according to Eq. (4.22);
until Converges
Sort each feature according to $\|\boldsymbol{v}^i\|_2$ in descending order;
return The top-K ranked features;

4.2 Clustering Structure Acquisition Although our approach does not explicitly utilize the clustering structure for feature selection, we are still able to acquire the clusters through matrix V, which may be useful for some applications. Specifically, we first perform the sum-to-one normalization according to [9]:

$$U \leftarrow U D_U^{-1}; \quad V \leftarrow D_U V$$

where $D_U = \mathrm{diag}(\mathbf{1}^\top A)$. Denote $\boldsymbol{c} \in \mathbb{R}^m$ as the cluster identification vector where c_i records which cluster the i-th class belongs to, then $\boldsymbol{c}$ be calculated by

$$c_i = \arg\max_j v_{ij} \tag{4.23}$$

where v_{ij} is the (i, j)-th entry of V.

4.3 Algorithm Analysis Since we adopt ADMM as the optimization algorithm, the convergence is guaranteed due to the proof in [4]. The convergence criteria can be set as $\frac{J_{t+1}-J_t}{J_t} < \epsilon$ where J_t is the objective function in Eq. (3.9) and ϵ is a tolerance value. In our implementation we control the iteration by setting a maximun number of iteration, e.g., 100 in our experiment.

For the time complexity, the update of E and U involves the computation of Q and K. Since U is sparse, the computation cost is $O(Nd)$. The main computation cost during updating Z is the calculation of T, which is $O(k^2)$. The computation cost for updating V involves the computation of N and the SVD decompostion, which is $O(Ndk)$ and $O(Nk^2)$. The computation cost for update W mainly includes matrix multiplication and matrix inverse whose total time complexity is $O(ndk)$. The computational cost for Y_1 and Y_2 are both $O(Nd)$. Since $d \gg k$, the final computation cost is $O(Ndk)$ for each iteration.

5 Experiments

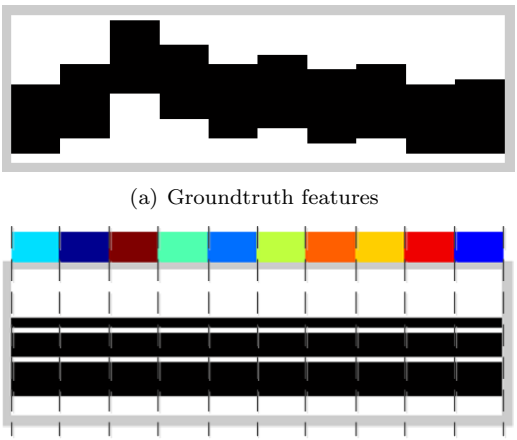
In this section we conduct experiments to evaluate the effectiveness of our proposed framework. We first describe the experiment of a simulated dataset to verify the effectiveness of the proposed framework in finding the cluster structure of class labels. Then we focus on the empirical evaluation by introducing the public datasets involved in our experiments and the baseline approaches we compared with followed by the experiment results and parameter analysis.

5.1 Experiment using Simulated Data Since it is difficult to obtain the groundtruth cluster structure for real applications, we first verify the effectiveness of the proposed approach in obtaining the cluster structures of the proposed approach on simulated dataset. Following [23, 41], we construct the synthetic data containing 10 clusters with 10 class labels in each cluster, generating a total number of 100 class labels. For the i-th class label, a dataset $X_i \in \mathbb{R}^{d \times n}$ is randomly drawn from a normal distribution $N(0, 1)$ for learning, with the dimension $d = 30$ and the sample size $n = 60$.

We construct the projection model as follows. For the i-th cluster, a cluster weight vector $\boldsymbol{w}_i^c \in \mathbb{R}^d$ is drawn from the normal distribution $N(0, 900)$. Then 15 dimensions of $\boldsymbol{w}_i^c$ are randomly but carefully selected and assigned as zeros, to ensure all $\boldsymbol{w}^c$ are orthogonal to each other. Similarly, for the j-th class label belonging to cluster i, a class-specific weight vector $\boldsymbol{w}_j^s \in \mathbb{R}^d$ is drawn from the normal distribution $N(0, 16)$ with the same dimensions of $\boldsymbol{w}_i^c$ assigned to zeros. Thus, the ultimate weight vector of the j-th class label is the linear combination of the cluster and class-specific weight vector $\boldsymbol{w}_j = \boldsymbol{w}_i^c + \boldsymbol{w}_j^s$.

The corresponding response $\boldsymbol{y}_i$ of the i-th samples $\boldsymbol{x}_i$ in the class j is then obtained by $\boldsymbol{y}_i = \boldsymbol{w}_j^T \boldsymbol{x}_i + \boldsymbol{\varepsilon}_i$ where $\boldsymbol{\varepsilon}$ is the noise vector drawn from $N(0, 0.1)$. We choose 0.5 as the threshold to assign binary label to each sample.

We verify the effectiveness of our proposed approach by comparing the learned cluster structure and the selected features with the groundtruth. Based on the prior knowledge implied by the construction of the



(a) Groundtruth features

(b) Selected features

Figure 4: The selected features of our approach and the corresponding groundtruth features in the simulation experiment. The horizontal coordinates denote class labels while the vertical coordinates denote features. The colored bins represent the detected cluster structure corresponding to the class labels, where each color denotes an independent cluster. The black parts are zeros and the white parts are non-zeros.

groundtruth, We set $k = 10$ and the number of selected features as $K = 15$.

Figure 4 demonstrates the detected cluster structures and selected features by our approach (4(b)) and the corresponding groundtruth features (4(a)). The results show that our approach can detect the correct cluster structures of the class labels, and select important features which exist in majority of labels. For example, for the majority of class labels in Figure 4(a), the first and the last several features are important, which are correctly selected by the proposed approach as shown in Figure 4(b).

5.2 Experiment using Real Data The real data experiment is conducted on 6 public benchmark feature selection datasets including one object image dataset, i.e., **COIL100** [1], one hand written digit image dataset **USPS** [22], one spoken letter speech dataset **Isolet** [12], three face image dataset **YaleB** [13], **ORL** [30] and **PIX10P**[1]. All the datasets are standardized to zero-mean and normalized by the standard deviation. The statistics of the datasets are summarized in Table 1.

Following the common way to evaluate supervised feature selection, we assess the quality of selected features in terms of the classification performance [20, 6].

[1]PIX10P is publicly available from https://featureselection.asu.edu/datasets.php

Table 1: Statistics of the Datasets

Dataset	# Samples	# Features	# Classes
COIL100	7200	1024	100
YaleB	2414	1024	38
ORL	400	4096	40
PIX10P	100	10000	10
USPS	9298	256	10
Isolet	7797	617	150

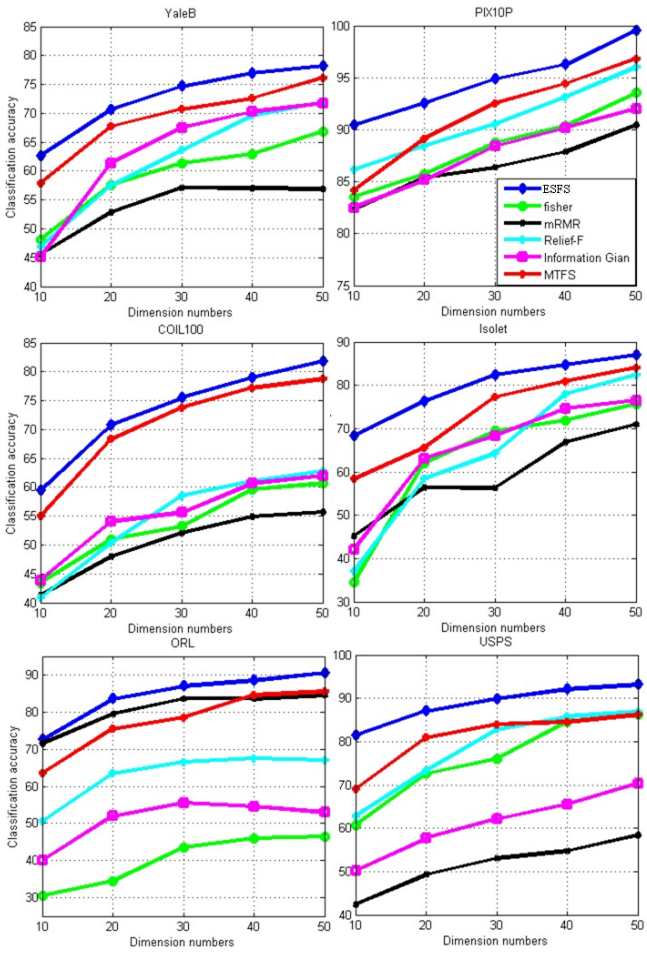


Figure 5: Classification accuracies with different dimensions of features selected of SVM.

The larger classification accuracy is, the better performance the corresponding feature selection approach achieves. In our experiments, we employ linear Support Vector Machine (**SVM**) and k-nearest neighbors (k**NN**) classifier with $k = 3$ for evaluation. How to determine the optimal number of selected features is still an open question for feature selection; hence we vary the number of selected features as {10,20, ...,50} in this work. In each setup 50% samples are randomly selected for feature selection and training for classification

and the remaining is for testing. Specific constrains are imposed to make sure the class labels of the training set are balanced. The whole experiment is conducted 10 rounds and average accuracies are reported.

We compare the proposed approach with the following representative feature selection algorithms:

- Fisher Score [11] determines the most relevant features with the best discriminating ability on fully labeled training data.
- mRMR [29] selects features that correlate the strongest with a classification variable and makes the features mutually different from each other.
- Relief-F [27] chooses instances randomly and update the weight of the feature relevance based on the nearest neighbors.
- Information Gain [10] selects features by computing information gain.
- MTFS [2] applies a $\ell_{2,1}$ norm on the column space of W to constrain a sparse structure for feature selection.

For the parameter setup, we tune the parameters for all methods by cross-validation for a fair comparison. We will further discuss some key parameters of the proposed framework in the following subsection.

Figure 5 shows ihe comparison results for **SVM** on the 6 benchmark datasets and we make the following observations:

- MTFS and the proposed framework ESFS outperform Fisher Score, mRMR and Information Gain. For example, the proposed framework achieves a performance gain of 6%∼15% compared with the traditional approaches. Fisher Score, mRMR and Information Gain select features one by one while MTFS and ESFS select features in a batch model. It is consistent with what was suggested in [35] that it is better to analyze features jointly for feature selection.
- Most of the time, the proposed framework ESFS outperforms MTFS. Better performance gain is usually achieved when fewer number of features are selected. For example, ESFS obtains about 10% relative improvement over MTFS in the USPS dataset when 10 features are selected. This performance gain suggests that modeling label correlation can significantly improve feature selection performance for multi-class data.

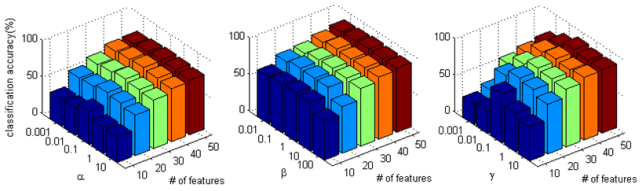


Figure 6: Parameter Analysis for the Proposed Framework

5.3 On Choosing the Parameters The proposed framework has three important parameters - α and β controlling the contribution of modeling label correlation and γ controlling the sparsity of W We study the effect of each parameter by fixing the other to see how the performance of ESFS varies with the number of selected features. Due to the page limitation, we only report the result on the **Isolet** dataset in Figure 6. However, we have similar observations in other datasets.

Figure 6 demonstrates the experiment result of how the classification accuracies varies with the increase of parameters. With the increase of *alpha* and β, the performance first increases, demonstrating the importance of modeling label correlation, and then decreases. This property is practically useful because we can use this pattern to set these parameters. When γ increases, the performance increases dramatically, which suggests the capability of $\ell_{2,1}$-norm for feature selection.

6 Conclusion

In this paper, we proposed an embedded supervised feature selection framework for multi-class data. Different from existing approaches, our method considers the label correlation among classes and utilizes such correlation to help feature selection. The proposed approach is evaluated on a synthetic dataset and 6 public benchmark datasets with comparison with representative baseline approaches. The results demonstrated (1) the proposed framework can capture clustered label correlation; (2) the importance of label correlation in feature selection for multi-class data; and (3) the proposed framework outperforms the state-of-the-art supervised feature selection algorithms.

There are some directions we need further investigations. First, the current optimization algorithm can only get a local optimal solution for the proposed framework and we will investigate optimization methods that can provide global optimal solutions for the proposed framework. Second, our successful experience in exploiting label correlation in feature selection encourages us to model label correlation in more multi-class learning problems.

Acknowledgment The work was supported in

part by ONR grant N00014-15-1-2344 and ARO grant W911NF1410371. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ONR or ARO.

References

[1] Columbia Object Image Library (COIL-100). Technical report, Columbia University, 1996.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Mach. Learn.*, 2008.

[3] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artif. Intell.*, 1997.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 2011.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] X. Cai, F. Nie, and H. Huang. Exact top-k feature selection via l2,0-norm constraint. In *IJCAI'13*.

[7] L. Chen and B. Li. Clustering-based joint feature selection for semantic attribute prediction. In *IJCAI'16*.

[8] L. Chen, Q. Zhang, and B. Li. Predicting multiple attributes via relative multi-task learning. In *CVPR'14*.

[9] S. Choi. Algorithms for orthogonal nonnegative matrix factorization. In *IJCNN'08*.

[10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[11] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. 2 edition, 2001.

[12] M. Fanty and R. Cole. Spoken letter recognition. In *NIPS'91*.

[13] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001.

[14] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st edition, 1989.

[15] P. Gong, J. Ye, and C. shui Zhang. Multi-stage multi-task feature learning. In *NIPS'12*.

[16] P. Gong, J. Ye, and C. Zhang. Robust multi-task feature learning. KDD'12.

[17] J. Gui, T. Liu, D. Tao, Z. Sun, and T. Tan. Representative vector machines: A unified framework for classical classifiers. *IEEE Transactions on Cybernetics*, 2016.

[18] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 2003.

[19] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Mach. Learn.*, 2002.

[20] Y. Han, Y. Yang, and X. Zhou. Co-regularized ensemble for feature selection. In *IJCAI'13*.

[21] J. Huang, F. Nie, H. Huang, and C. Ding. Robust manifold nonnegative matrix factorization. *ACM Trans. Knowl. Discov. Data*, 2014.

[22] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1994.

[23] L. Jacob, J. philippe Vert, and F. R. Bach. Clustered multi-task learning: A convex formulation. In *NIPS'09*.

[24] D. Jayaraman, F. Sha, and K. Grauman. Decorrelating semantic visual attributes by resisting the urge to share. CVPR'14.

[25] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. AAAI'92.

[26] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 1997.

[27] H. Liu and H. Motoda, editors. *Computational Methods of Feature Selection*. Chapman & Hall, 2008.

[28] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 2007.

[29] F. D. C. Peng, H. Long. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005.

[30] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision*, 1994.

[31] B. Santosa, T. Conway, and T. Trafalis. A hybrid knowledge based-clustering multi-class svm approach for genes expression analysis. In *Data Mining in Biomedicine*. 2007.

[32] R. Setiono and H. Liu. Neural-network feature selector. *Neural Networks, IEEE Transactions on*, 1997.

[33] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu. Statistical shape analysis: Clustering, learning, and testing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005.

[34] Y. Sun, C. Babbs, and E. Delp. A comparison of feature selection methods for the detection of breast cancers in mammograms: Adaptive sequential floating search vs. genetic algorithm. In *EMBS'05*.

[35] J. Tang and H. Liu. Feature selection with linked data in social media. In *SDM'12*.

[36] K. Torkkola. Feature extraction by non parametric mutual information maximization. *J. Mach. Learn. Res.*, 2003.

[37] S. Wang, J. Tang, and H. Liu. Embedded unsupervised feature selection. AAAI'15.

[38] Y. Wang, S. Wang, J. Tang, H. Liu, and B. Li. Unsupervised sentiment analysis for social media images. In *IJCAI'15*.

[39] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR'10*.

[40] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML'97*.

[41] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. In *NIPS'11*.