Structure Regularized Unsupervised Discriminant Feature Analysis

Mingyu Fan
College of Maths & Info. Science,
Wenzhou University,
Wenzhou 325035, China
fanmingyu@wzu.edu.cn

Xiaojun Chang
Centre for Artificial Intelligence
University of Technology Sydney,
Sydney, NSW 2007, Australia
csj273@gmail.com

Dacheng Tao
Centre for Artificial Intelligence
University of Technology Sydney,
Sydney, NSW 2007, Australia
dacheng.tao@uts.edu.au

Abstract
Feature selection is an important technique in machine learning research. An effective and robust feature selection method is desired to simultaneously identify the informative features and eliminate the noisy ones of data. In this paper, we consider the unsupervised feature selection problem which is particularly difficult as there is not any class labels that would guide the search for relevant features. To solve this, we propose a novel algorithmic framework which performs unsupervised feature selection. Firstly, the proposed framework implements structure learning, where the data structures (including intrinsic distribution structure and the data segment) are found via a combination of the alternative optimization and clustering. Then, both the intrinsic data structure and data segmentation are formulated as regularization terms for discriminant feature selection. The results of the feature selection also affect the structure learning step in the following iterations. By leveraging the interactions between structure learning and feature selection, we are able to capture more accurate structure of data and select more informative features. Clustering and classification experiments on real world image data sets demonstrate the effectiveness of our method.

Introduction
Real world applications usually involve big data with high dimensionality, such as in computer vision (Collins, Liu, and Leordeanu 2005), bioinformatics (Saeyes, Inza, and Larranaga 2007), and data mining (Liu et al. 2010). High dimensionality generally poses great challenges, including “the curse of dimensionality”, huge computation and storage cost, to conventional machine learning algorithms. In order to address this issue, feature selection is proposed to select a subset of features from the feature pool of high dimensional data for a compact and informative representation (Guyon and Elisseeff 2003). After the implementation of feature selection, conventional machine learning algorithms can be applied on data represented by only the selected relevant features instead of all the features.

According to the availability of class labels of data, feature selection algorithms can be roughly classified into three groups, i.e., supervised feature selection (Song et al. 2007), semi-supervised feature selection (Xu et al. 2010), and unsupervised feature selection (He, Cai, and Niyogi 2006). Supervised feature selection evaluates features by computing a feature’s correlation with the class labels. Representative supervised feature selection methods include the Fisher score method (Duda, Hart, and Stork 2000), robust \( \ell_2 \) regression method (Nie et al. 2010) and the generalized Fisher score method (Gu, Li, and Han 2011). By exploiting the information of class labels, supervised feature selection is usually able to identify the discriminative and effective features for recognition and classification (Tao et al. 2016). On the other hand, with insufficient class labels, unsupervised and semi-supervised methods have to consider the capability of the features in preserving or revealing of the underlying structure of data (He, Cai, and Niyogi 2006). A frequently used criterion is to select the features which best preserve intrinsic data structure. Recent research has witnessed several important data structures that should be preserved by features, where these data structures include, but not limit to, the sparse global structure (Du and Shen 2015) and the local manifold structure (Han et al. 2015). Generally speaking, because of insufficient class labels, it is more difficult for unsupervised and semi-supervised feature selection to find the discriminative and informative features.

In practice, data structures for unsupervised feature selection are usually captured in the form of weighted graphs, such as the sample pair-wise similarity graph and the sparse graph (Du and Shen 2015) and the locally linear reconstruction graph (Hou et al. 2014). In graph based feature selection methods, the constructed graph is fixed in the following procedures and the performance of feature selection is largely determined by the quality of graph. Ideally, the quality of constructed graphs should be improved using the selected informative features instead of all the features that contain noisy and irrelevant ones. It is reasonable to alternatively optimize the data structure characterization using the selected features and then identify the selected feature set using the refined graph. Each sub-task and be iteratively boosted by using the result of the other one. Motivated by this, the main contributions of our work are:

- Data structure characterization is learned in two forms, which are referred as the soft structure and hard structure. The soft structure is defined by pair-wise similarities between data points and the hard structure is learned by data segmentation. Soft data structure is used to evaluate
the ability of features in preserving the structure of data and hard structure is used to extract the unsupervised discriminant information of data.

- The results of feature subset selection and data structure learning are optimized alternatively. In this way, each sub-task (structure learning and feature selection) and can boost the result of the other in the proposed feature selection framework.

- Both the soft and hard data structure can be naturally formulated as regularization terms in the regresisonal feature selection framework. And the derived regression algorithm can be efficiently optimized with convergence guarantee.

The Proposed Framework

Let $X = [x_1, \ldots, x_N]$ be the given data matrix, where $x_i \in \mathbb{R}^D$ ($1 \leq i \leq N$) denotes the $i$-th data sample. Feature selection aims to evaluate the importance of all features of $X$, i.e., the row vectors of $X$. For an arbitrary matrix $A \in \mathbb{R}^{m \times n}$, its $\ell_{1,2}$-norm is defined as $\|A\|_{1,2} = \sum_{j=1}^n ||A_j||_2 = \sum_{j=1}^n \sqrt{\sum_{i=1}^m a_{ij}^2}$, where $A_j$ denotes the $i$-th column vector of matrix $A$. For simplicity, we assume that the elements of the $D$-th row of data matrix $X$ are all 1s and thus the bias term in linear regression can be integrated in matrix $A$.

Data Structure Learning

A large number of unsupervised feature selection algorithms have been proposed based on the analysis of the data structures, such as the Maximum Variance (MaxVar), manifold structure (Zhao et al. 2009; Du et al. 2013). Inspired by the recent development of compressive sensing, a popular approach to learn the affinity matrix of data is based on the self-expressiveness model. The basic assumption is that data points lie on a union of subspaces. Each data point can be expressed in terms of a linear combination of other data points.

The general problem can be formulated as below:

$$
\min_{Z,E} \|Z\|_\kappa + \lambda_E \|E\|_\omega,
$$

subject to $X = XZ + E$, diag($Z$) = 0,

where matrix $Z$ consists of self-expressive coefficients and $E$ denotes the matrix of data noise, $\| \cdot \|_\kappa$ and $\| \cdot \|_\omega$ are two properly chosen norms, $\lambda_E > 0$ is a tradeoff parameter. Many successful methods have been proposed based on different choices of the norms for coefficient $Z$ and noise $E$. For example, in Sparse Subspace Clustering (SSC) (Elhamifar and Vidal 2013), $\ell_1$ norm is used for both $\| \cdot \|_\kappa$ and $\| \cdot \|_\omega$ as a convex surrogate over $\ell_0$ norm to promote sparseness in coefficient matrix $Z$ and handle the noises $E$. In Low-Rank Representation (LRR) (Liu et al. 2013), the nuclear norm $\| \cdot \|_\kappa$ is adopted for $\| \cdot \|_\kappa$ as a convex surrogate of the matrix rank function and $\ell_{1,2}$ norm is used for $\| \cdot \|_\omega$ to handle noise or outlying entries $E$. Besides, a number of variants of (1) for data structure learning have been proposed for various applications in machine learning and pattern recognition.

Based on the motivation that nearby data points should have large similarity and far away data points should have small similarity, (Nie, Wang, and Huang 2014) proposes to compute the similarities between pair wise data points by solving the following problem:

$$
\min_{S=(s_{ij})} \sum_{i,j} \left( \|x_i - x_j\|^2 s_{ij} + \mu s_{ij}^2 \right),
$$

subject to $\sum_{j=1}^N s_{ij} = 1$, $s_{ij} \geq 0$, and $s_{ii} = 0$, where $\mu$ is the regularization parameter which is used to avoid trivial solution and add a prior of uniform distribution. It can be found that nearby data samples have large similarity $s_{ij}$. With such desirable property, the estimated similarity matrix $S$ can be considered as an effective local data structure characterization.

The self-expressive model (1) preserves global and sparse reconstruction data structure while the adaptive neighbor model (2) is based on the local similarity of data and focuses on local data structure. Once the coding $Z$ (or similarity matrix $S$) has been found, the segmentation of data can be obtained by applying Spectral Clustering (SC) (Ng, Jordan, and Weiss 2001) on the induced affinity matrix $W = [Z] + [Z^T]$ (or $W = [S] + [S^T]$). The clustering result is assumed to be given as $\{t_1, \ldots, t_N\}$, where $t_i \in \{1, \ldots, C\}$ is the assigned cluster label of $x_i$ with $C$ denotes the number of clusters. In this paper, the induced affinity matrix $W$ is referred to as the soft data structure because it describes the pair-wise similarity using nonnegative real value, meanwhile, the result of data segmentation is called as the hard data structure characterization as it provides label attribute of the data points.

Discriminant Feature Analysis

Linear Discriminant Analysis (LDA) (Fukunaga 1990) is a popular supervised feature extraction method. It seek directions on which data points from the same classes are close and data points from different classes are far away from each other. Given the class labels of data points, the objective function of LDA is as follows

$$
A = \arg \min_A \frac{\text{Tr} (AS_w A^T)}{\text{Tr} (AS_b A^T)},
$$

where $\text{Tr} (\cdot)$ indicates the matrix trace operator, $A \in \mathbb{R}^{D \times D}$ is the desired projective matrix and

- $S_w = \sum_{c=1}^C \left( \sum_{i=1}^{n_c} (x_i^{(c)} - \bar{x}^{(c)}) (x_i^{(c)} - \bar{x}^{(c)})^T \right)$,
- $S_b = \sum_{c=1}^C n_c (\bar{x}^{(c)} - \bar{x})(\bar{x}^{(c)} - \bar{x})^T$,

are within-class scatter matrix and between-class scatter matrix respectively, with $n_c$ indicates the number of samples in the $c$-th class, $x_i^{(c)}$ be the $i$-th sample in the $c$-th class, $\bar{x}^{(c)}$ is the mean of the samples in the $c$-th class, $\bar{x}$ denotes the mean of all the samples. Define $S_t = \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$ as the total scatter matrix, then we have $S_t = S_w + S_b$. The objective function of LDA in (3) is equivalent to

$$
A = \arg \min_A \frac{\text{Tr} (AS_A A^T)}{\text{Tr} (AS_b A^T)} = \arg \max_A \frac{\text{Tr} (AS_A A^T)}{\text{Tr} (AS_b A^T)}.
$$

The solution $A$ is given by the eigenvectors of the top eigenvalues of the generalized eigen-problem $S_b A \alpha = \lambda S_A A \alpha$, where $\lambda > 0$. 

1871
\( \lambda \) is an eigenvalue and \( \alpha \) denotes the corresponding eigenvector. Because of its simplicity and effectiveness, LDA has been widely used in machine learning research.

**Unsupervised Discriminative Feature Selection Framework**

In unsupervised scenarios, the label of data samples are unknown. As discussed above, one can implement data structure learning (Section 3) to find the clustering labels of data samples, i.e., the hard data structure characterization. Then, the with assigned clustering labels, we convert LDA into a feature selection algorithm via adopting \( \ell_{1,2} \)-norm as a regularizer:

\[
A = \arg \min_A \frac{\text{Tr} (AS_wA^T)}{\text{Tr} (AS_bA^T)} + \gamma_A \|A\|_{1,2}, \tag{5}
\]

where the regularization term \( \|A\|_{1,2} \) ensures that \( A \) is sparse in columns, making it suitable for feature selection. The balancing parameter \( \gamma_A \) controls the tradeoff between the two terms.

Furthermore, we hope the result of feature selection can affect back on the data structure learning process. When \( Ax_1 \) and \( Ax_2 \) are close after feature selection, the similarity \( w_{ij} \) between samples \( x_1 \) and \( x_2 \) should be large. The objective to minimize the disagreement between the projected data matrix \( AX \) and the data similarity matrix \( W \) can be quantified as:

\[
\min_{A,W} \sum_{i,j=1}^n w_{ij} \left( \frac{1}{2} \|Ax_i - Ax_j\|^2 \right) = \min_{A,W} \|W \circ \Theta\|_1, \tag{6}
\]

where \( \circ \) indicates the hadamard product, \( W = (w_{ij}) \) and \( \Theta = \left( \frac{1}{n} \|Ax_i - Ax_j\|^2 \right) \). Because of the fact that \( W = |Z| + |Z^T| \) (or \( W = |S| + |S^T| \)), the problem (6) is essentially equivalent to

\[
\min_{A,Z} \sum_{i,j=1}^n |z_{ij}| \left( \frac{1}{2} \|Ax_i - Ax_j\|^2 \right) = \min_{A,Z} \|Z \circ \Theta\|_1, \tag{7}
\]

or

\[
\min_{A,S} \sum_{i,j=1}^n |s_{ij}| \left( \frac{1}{2} \|Ax_i - Ax_j\|^2 \right) = \min_{A,S} \|S \circ \Theta\|_1.
\]

Due to the success of SSC method in subspace clustering, here we adopt the \( \ell_1 \)-norm for both \( \| \cdot \|_k \) and \( \| \cdot \|_2 \) in (1) for structure learning. Combining our discriminative feature selection term (5) and the disagreement term (6), the unified optimization framework for the Sr-UDFS algorithm is proposed as follows:

\[
\min_{Z,E,A} \left\{ \|Z\|_1 + \lambda_Z \|E\|_1 + \lambda_Z \|\Theta \circ Z\|_1 \right\}
\]

\[
+ \frac{\text{Tr} (AS_wA^T)}{\text{Tr} (AS_bA^T)} + \gamma_A \|A\|_{1,2}, \tag{7}
\]

Subj. to: \( X = XZ + E, \quad \text{diag}(Z) = 0 \).

As can be seen that when \( A \) is fixed, our method learns the data structure with the consideration of the refined data features (the third term). When the coding matrix \( Z \) is fixed, both the soft data structure and hard data structure are transformed into regularizers for the problem of feature selection. In our method, both the adverse effect of data noise \( E \) and noisy features are largely alleviated. The two sub tasks, data structure learning and feature selection, boosts each other within the unified learning framework.

**Optimization**

In this subsection, an efficient solution to the optimization problem in (7) is proposed based on solving the following subproblems iteratively:

1. Given \( A \), optimize \( Z \) and \( E \) through solving a weighted sparse coding problem.

2. Implement SC on weight matrix \( W = |Z| + |Z^T| \) to obtain the clustering labels of data.

3. Find the optimal \( A \) given \( Z \) and \( E \).

Given the projection matrix \( A \) (initialized as the identity matrix \( I \)), we solve for matrix \( Z \) and \( E \) through optimizing the following structured sparse problem:

\[
\min_{Z,E} \left\{ \|Z\|_1 + \lambda_Z \|E\|_1 + \lambda_Z \|\Theta \circ Z\|_1 \right\} \tag{8}
\]

Subj. to: \( X = XZ + E, \quad \text{diag}(Z) = 0 \).

To implement Alternating Direction Method of Multipliers (ADMM) method, an augmented matrix \( Q \) should be introduced, and the problem (8) is equivalent to:

\[
\min_{Z,E} \left\{ \|Z\|_1 + \lambda_Z \|E\|_1 + \lambda_Z \|\Theta \circ Z\|_1 \right\} \tag{9}
\]

Subj. to: \( X = XQ + E, \quad Q = Z - \text{diag}(Z) \).

The augmented Lagrangian function is given by:

\[
L(Z, Q, E, Y_1, Y_2) = \|Z\|_1 + \lambda_Z \|E\|_1 + \lambda_Z \|\Theta \circ Z\|_1
+ (Y_1, X - XQ - E) + (Y_2, Q - Z - \text{diag}(Z))
+ \frac{\mu}{2} \left( \|X - XQ - E\|^2_F + \|Q - Z - \text{diag}(Z)\|^2_F \right),
\]

where \( Y^{(1)}, Y^{(2)} \) are matrices of Lagrange multipliers, and \( \mu > 0 \) is a adaptive parameter. The iterative scheme of ADMM method for (8) can be presented as (at the \( t \) + 1-th iteration) where \( \rho > 1 \) is a given parameter.

For the \( Z \)-subproblem in (9), we solve the following problem:

\[
Z^{(t+1)} = \arg \min_Z \left\{ \| (11^T + \lambda_Z \Theta) \circ Z\|_1 + \frac{\mu(t)}{2} \|Q^{(t)} - Z + \text{diag}(Z)\|_2^2 \right\},
\]

where \( 1 \) indicates the vector with entries are all 1s. The closed-form solution for \( Z \) can be given as

\[
Z^{(t+1)} = Z^{(t+1)} - \text{diag}(Z^{(t+1)}),
\]

where \( Z_{ij}^{(t+1)} = S_{ij} \frac{1}{\mu(t)} (1 + \lambda_Z \Theta_{ij}) U_{ij}^{(t)} \) with \( U^{(t)} = Q^{(t)} + \frac{Y_{ij}^{(t)}}{\mu(t)} \). Here, \( S_{ij}(\cdot) \) is the element-wise shrinkage thresholding operator. It should be noted that instead of soft-thresholding all entries of matrix \( U^{(t)} \) with a constant value as in SSC (Elhamifar and Vidal 2013), the proposed method thresholds the entries of \( U^{(t)} \) with different values.

To optimize \( Q \) in (9), taking derivative of the objective function with respect to \( Q \) and the solution is given by

\[
Q^{(t+1)} = \left( X^T X + I \right)^{-1} \left( X^T (X - E^{(t)} + \frac{Y^{(t)}}{\mu(t)}) \right)
+ Z^{(t+1)} - \text{diag}(Z^{(t+1)}),
\]

1872
When other variables are fixed, the subproblem to find $E$ is:

$$E^{(t+1)} = \arg\min_{E} \lambda_E \|E\|_1 + \frac{\rho}{2} \|X - XQ^{(t+1)} - E\|^2_F.$$  

The closed-form solution for $E$ can be given as:

$$E^{(t+1)} = S \frac{\lambda_E}{\mu^{(t)}} \left(V^{(t)}\right), \quad (12)$$

where $V^{(t)} = X - XQ^{(t+1)} + \frac{\rho}{\mu^{(t)}} Y^{(t)}$.

After the convergence of ADMM method (9), the self-expressive sparse representation matrix $Z$ is obtained. The next step is to infer the segmentation of data into different clusters. To address this problem, one can directly compute the similarity matrix as $W = [Z] + [Z^T]$ and then apply SC method to $W$. The segmentation of data can be obtained as $T = \{t_1, \cdots, t_N\}$, where $t_i \in \{1, \cdots, C\}$ is the assigned cluster label of $x_i$.

Solving Structure regularized Unsupervised Discriminant Feature Selection. Given the labels $T$, the problem (7) reduces to the following problem:

$$\min_{A} \frac{\text{Tr} \left( AS_w A^T \right)}{\text{Tr} \left( A S_h A^T \right)} + \frac{\alpha}{2} \sum_{i,j} w_{ij} \|Ax_i - Ax_j\|^2 + \gamma_A \|A\|_{1,2}. \quad (13)$$

where the scatter matrices $S_w$ and $S_h$ are calculated based on the labels $T$. Because matrix $A$ exists in the numerator, denominator and the summed terms, it is difficult to directly solve (13). In this paper, we resort to the spectral regression method (Cai, He, and Han 2008) which can transform the intricate problem (13) to an equivalent regression form and make it easier and more efficient to solve.

Let $\tilde{X} = [x_1 - \bar{x}, \cdots, x_N - \bar{x}]$ be the centered data matrix. The between-class scatter matrix $S_b$ can be rewritten as:

$$S_b = \sum_{c=1}^{C} n_c (\bar{z}(c)) (\bar{z}(c))^T = \tilde{X} W \tilde{X}^T,$$

where $\tilde{W}_{ij} = \frac{1}{n_c}$ if $x_i$ and $x_j$ are of the same class $c$ and 0 otherwise. The following Theorem can be obtained.

**Theorem 0.1.** Let $Y \in \mathbb{R}^{(C-1) \times N}$ be a matrix of which each row vector is an eigenvector of the eigen-problem

**Algorithm 2** The algorithm for solving problem (13)

**Input:** Data matrix $X$, Sparse coding matrix $Z$

**Output:** Converged matrix $A$

1: Implement SC on $W = [Z] + [Z^T]$ to obtain data labels $T = \{t_1, \cdots, t_N\}$;
2: Compute the regression target $Y$;
3: while not converged do
4: Compute the diagonal matrix $D_A^{(t+1)}$ as (16);
5: Update $A^{(t+1)}$ as (17);
6: If not converged, set $t \leftarrow t + 1$.
7: end while

$\tilde{W} = \lambda y$. If there exist a matrix $A \in \mathbb{R}^{(C-1) \times D}$ such that $A \bar{X} = Y$, then each row vector of $A$ is an eigenvector of the generalized eigen-problem $X \tilde{W} X^T \alpha = \lambda \bar{X} \tilde{X}^T \alpha$ (i.e., eigen-problem for LDA) with the same eigenvalue $\lambda$.

**Proof.** With $A \bar{X} = Y$ and $\tilde{W} = \lambda y$, we have the following equation

$$X \tilde{W} X^T \alpha = \bar{X} \lambda y = \lambda \bar{X} \tilde{X}^T \alpha,$$

where $\alpha$ is the transpose of a row vector of matrix $A$ and $y$ is the transpose of a row vector of $Y$.

**Theorem 0.1** indicates that under mild condition, the LDA problem (4) is essentially equivalent to the regression problem: $A = \arg \min_{A} \|A \bar{X} - Y\|^2_F$, where the row vectors in $Y$ are eigenvectors of eigen-problem $\tilde{W} \lambda y$ and $\lambda y$. One advantage of LDA is that one need not to really solve the eigenproblem to obtain the eigenvectors $Y$. The $C + 1$ eigenvectors of $W$ can be directly given as $\{1\} \bigcup \{v_{c}\}_{c=1}^{C} \subset \{0, 1\}^N$, with the $j$-th entry of $v_c$ is 1 if and only if $x_j$ is in class $c$. Consequently, we can get the $C + 1$ useful orthogonal eigenvectors $\{y_{c}\}_{c=1}^{C}$ by implementing the Gram-Schmidt orthogonalization algorithm on $\{1\} \bigcup \{v_{c}\}_{c=1}^{C}$. As is shown in (Cai, He, and Han 2008), the $C + 1$ orthogonal eigenvectors are sufficient to represent a $C$ class problem.

Based on the above discussions, the problem (13) is equivalent to the following problem:

$$\min_{A} \|A \bar{X} - Y\|^2_F + \lambda \sum_{i} TrAXLX^T A^T + \gamma_A \|A\|_{1,2}. \quad (14)$$

where $L = D - W$ is the graph Laplacian matrix, $D$ is a diagonal matrix with diagonal elements $D_{ii} = \sum_{j=1}^{N} w_{ij}, i = 1, \cdots, N$. Motivated by the recent progress on $\ell_{1,2}$ norm minimization, the problem (14) can be efficiently solved by an iterative re-weighted approach which solve the following problems (at the $(t+1)$-th iteration):

$$A^{(t+1)} = \arg \min_{A} \|A \bar{X} - Y\|^2_F + \lambda \sum_{i} TrAXLX^T A^T + \gamma_A \|A\|_{1,2}, \quad (15)$$

$$\left[D_A^{(t+1)} \right]_{ii} = \|A^{(t+1)}\|_2. \quad (16)$$
The solution to (15) can be given as

$$A^{(t+1)} = YX^T D_A^{(t)} \left( X X^T D_A^{(t)} + \lambda_2 X L X^T D_A^{(t)} + \gamma A I \right)^{-1}$$  (17)

We can show that the objective function of problem (14) is nonincreasing under the updating rules of $A$ and $D$ in Algorithm 2.

**Theorem 2.** The Algorithm monotonically decrease the objective function of the problem (14) in each iteration, and converge to the global optimum of the problem.

**Proof.** The proof follows the work (Nie et al. 2010) and can be found in the supplement material. □

**Discussions**

In this section, we discuss the relationships between the proposed method and several algorithms, including TRACK (Wang, Nie, and Huang 2014), CGSSL (Li et al. 2014), and DFS (Tao et al. 2016).

TRACK proposed an unsupervised feature selection by integrating Fisher criterion and clustering as below

$$\min_{A^T A = I, A \in \text{Ind}} \left\{ \frac{\text{Tr} A S_w A^T}{\text{Tr} A S_l A^T} + \gamma A \| A \|_{1,2} \right\},$$  (18)

where $G$ is the $\{0,1\}$ cluster indicator matrix. In TRACK, $G$ is computed by the k-means method and $A$ is given by the eigenvectors of the smallest eigenvalues of the matrix

$$S_w - \frac{\text{Tr} A S_w A^T}{\text{Tr} A S_l A^T} S_l + \gamma_A \text{Tr} (A S_l A^T) D_A,$$  (19)

where $D_A$ is a diagonal matrix whose $i$-th diagonal entry $D_A(i,i) = \frac{1}{2(1 + \| A^T S_l A^T \|^2)}$. As can be seen, TRACK implements clustering directly on transformed data matrix $AX$, meanwhile, Sr-UDFS implements SC on refined similarity matrix $W$. DFS proposed a supervised feature selection based on the Fisher criterion, which compute the projection matrix $A$ by solving the generalized eigen-problem:

$$\left( \gamma_A A - S_l \right) \alpha = \lambda S_l \alpha. \tag{20}$$

Compared with both TRACK and DFS, our Sr-UDFS efficiently transforms objective of Fisher criterion into a regression model and avoid solving the eigen-problem (19) or the generalized eigen-problem (20).

CGSSL is an one-stop unsupervised feature selection method. The objective function of CGSSL can be presented as

$$\min_{A,P,Q,Y} \text{Tr} (Y L Y^T) + \alpha \| Y - AX \|^2_F + \beta \| A \|_{1,2}$$

$$+ \gamma \| A - PQ \|^2_F,$$

subj. to. $Y Y^T = I, \quad Y \geq 0, \quad Q Q^T = I,$

where $L$ is the graph Laplacian matrix, $\alpha, \beta$ and $\gamma$ are given parameters. As can be seen, different from Sr-UDFS, the entries of target $Y$ are nonnegative reals instead of $\{0,1\}$ values. Besides, the data structure learned in CGSSL is based only on the local manifold assumption, which cannot utilize the refined data features to improve the quality of data structure learning.

---

**Experiments**

In this section, we evaluate the proposed Sr-UDFS to data clustering and classification on benchmark image datasets. Besides, state-of-the-art unsupervised feature selection methods are compared under various experimental settings.

**Datasets Description**

The experiments are conducted on publicly available image data sets: the Coil-20 data set (Coil-20)\(^1\), the USPS handwritten digits data set (USPS)\(^2\), the Yale-B Extended (YaleB) and the CMUPIE face data sets\(^3\). The statistics of the data sets are summarized in Table 1.

**Experiment Setup**

The validate the effectiveness of the proposed Sr-UDFS, we compare it with several state-of-the-art unsupervised feature selection methods, which includes LapScore (He, Cai, and Niyogi 2006), SPFS (Zhao et al. 2013), UDFS (Yang et al. 2011), MCFS (Cai, Zhang, and He 2010), RUFS (Qian and Zhai 2013), and JELSR (Hou et al. 2014). Also, one baseline (all features) for data clustering and classification is also compared.

There are some parameters to be set for the comparing methods. For methods require neighborhood sizes for data structure learning, the neighborhood size is searched in $\{4,6,8,10\}$. For UDFS, RUFS, and JELSR, the regularization parameters are searched in the range $\{10^{-5}, 10^{-4}, \ldots, 10^2, 10^3\}$. The parameter $\gamma_A$ is searched in the range $\{0.01, 0.05, 0.1, 0.5, 1\}$. To make the experimental results reproducible, $\lambda_2$ for Sr-UDFS is set as 0.1 respectively throughout the experiments. The experimental results of the parameters sensitivity for Sr-UDFS on Coil-20 data set is shown in Fig. 2.

Given a data set, we randomly select $p$ percents from every class in data X to formulate the training set $X_{train}$, and the left data are used as the test data $X_{test}$. The results when $p = 30$ is presented here and the results when $p = 10$ is provided in the supplement material. The feature selection methods are implemented on training data to rank all the features. In our data clustering experiments, the test data is clustered by $k$-means method with some selected features.


---

Table 1: Statistics of the data sets

<table>
<thead>
<tr>
<th>Data sets</th>
<th># of samples</th>
<th># of Dimension</th>
<th># of Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coil-20</td>
<td>1440</td>
<td>1024</td>
<td>20</td>
</tr>
<tr>
<td>YaleB</td>
<td>2414</td>
<td>1024</td>
<td>38</td>
</tr>
<tr>
<td>USPS</td>
<td>9298</td>
<td>256</td>
<td>10</td>
</tr>
<tr>
<td>CMUPIE</td>
<td>11554</td>
<td>1024</td>
<td>68</td>
</tr>
</tbody>
</table>

---

1874
Table 2: Clustering results when 30% of data are used as the training data. The result is in bold when one feature selection method outperforms the comparing feature selection methods.

<table>
<thead>
<tr>
<th></th>
<th>All Fea</th>
<th>LapScore</th>
<th>SPFS</th>
<th>UDFS</th>
<th>MCFS</th>
<th>RUFS</th>
<th>JELSR</th>
<th>Sr-UDFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>p = 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coil-20</td>
<td>48.72(0.17)</td>
<td>52.81(7.16)</td>
<td>50.59(4.47)</td>
<td>48.87(4.98)</td>
<td>50.76(4.61)</td>
<td>47.67(4.73)</td>
<td>48.27(7.88)</td>
<td>57.20(5.61)</td>
</tr>
<tr>
<td>YaleB</td>
<td>8.44(0.21)</td>
<td>8.61(0.42)</td>
<td>12.72(2.65)</td>
<td>8.43(0.62)</td>
<td>9.83(1.62)</td>
<td>10.74(1.68)</td>
<td>7.94(0.50)</td>
<td>16.71(4.59)</td>
</tr>
<tr>
<td>USPS</td>
<td>65.56(0.06)</td>
<td>54.68(7.55)</td>
<td>62.91(4.15)</td>
<td>46.88(1.93)</td>
<td>54.28(5.32)</td>
<td>59.15(7.60)</td>
<td>59.46(5.95)</td>
<td>65.41(4.21)</td>
</tr>
<tr>
<td>CMUPIE</td>
<td>9.77(0.16)</td>
<td>7.91(0.18)</td>
<td>11.42(0.60)</td>
<td>8.87(0.24)</td>
<td>9.08(0.48)</td>
<td>9.64(0.45)</td>
<td>8.95(0.30)</td>
<td>11.91(1.13)</td>
</tr>
</tbody>
</table>

Performance Evaluation

For classification experiments, we choose the 1-Nearest Neighbor (1-NN) classifier because it is parameter free and the results will be easily reproducible. The results are measured by classification accuracy. Because the optimal number of features are unknown, we compare the algorithms with different percentages of training subset and varying number of selected features. The classification results are shown in Fig. 1.

As can be seen, most of the time Sr-UDFS outperforms the comparing methods on the data sets. On Coil-20, YaleB and CMUPIE data sets, Sr-UDFS outperforms the 1-NN with all features. And on USPS data set, Sr-UDFS can achieve comparable performance as 1-NN with fewer selected features. These results indicate that the proposed method can effectively remove redundant and noisy features of data.

With the selected features, we evaluate the performance of clustering by two common evaluation metrics, Accuracy (Acc) and Normalized Mutual Information (NMI). The range of the number of the selected feature for Coil-20, YaleB, and CMUPIE is \{15, 25, 35, 45, 50, 100, 150, 200, 250, 300\} and the range of selected features for USPS is \{15, 25, 35, 45, 50, 100, 150, 200, 250, 300\}. We finally report the averaged results and standard deviation over the range of selected features. The clustering results in terms of Acc and NMI are reported in Table 2.

Compared with clustering using all features, except the case on USPS data set when p = 30, the proposed method not only largely reduce the number of selected features, but also improve the clustering performance. Compared with other unsupervised feature selection methods, the Sr-UDFS produces better performance in most of the cases. And the performance are very close when some other methods outperform the proposed method.

Conclusion

In this paper, we proposed a novel unsupervised feature selection method which can simultaneously perform data structure learning and feature selection. In our method, two
data structures, soft structure and hard structure, are learned via a combination of the alternative optimization and clustering. Both the two types of data structures are formulated as regularization terms for our discriminative feature selection. An efficient algorithm for the proposed algorithm is proposed. The connections between our method with other counterparts are discussed. Experiments on benchmark image data sets have been presented to demonstrate the superior performance of our method.

Acknowledgments

This work was partially supported by the National Natural Science Foundation (NNSF) of China under Grants 61473212, 61203241, 61472285, and the Natural Science Foundation of Zhejiang Province under Grant LY15F030011, partially supported by the Australian Research Council (ARC) projects FT-130101457, DP-140102164, and LE-140100061.

References


