Feature Selection via Joint Embedding Learning and Sparse Regression

Chenping Hou[1], **Feiping Nie**[2], **Dongyun Yi**[1] **and Yi Wu**[1]
[1]Department of Mathematics and Systems Science,
National Univsrsity of Defense Technology, 410073, Changsha, China.
[2]Department of Computer Science and Engineering,
University of Texas, Arlington, 76019, USA.
{hcpnudt,feipingnie,dongyun.yi}@gmail.com, wuyi_work@sina.com

Abstract

The problem of feature selection has aroused considerable research interests in the past few years. Traditional learning based feature selection methods separate embedding learning and feature ranking. In this paper, we introduce a novel unsupervised feature selection approach via Joint Embedding Learning and Sparse Regression (JELSR). Instead of simply employing the graph laplacian for embedding learning and then regression, we use the weight via locally linear approximation to construct graph and unify embedding learning and sparse regression to perform feature selection. By adding the $\ell_{2,1}$-norm regularization, we can learn a sparse matrix for feature ranking. We also provide an effective method to solve the proposed problem. Compared with traditional unsupervised feature selection methods, our approach could integrate the merits of embedding learning and sparse regression simultaneously. Plenty of experimental results are provided to show the validity.

1 Introduction

The problem of reducing data's dimensionality is a key research topic for both artificial intelligence and machine learning. In the literature, there are mainly two distinct ways for dimensionality reduction: feature selection and feature learning (or 'feature extraction'). Feature selection tries to extract a few relevant features to represent the original data while feature learning combines several original features to form new representations. Compared with feature learning which can *create* new features, feature selection does not change the original representations of data variables. If we are required to keep the original physical meanings of each feature, feature selection is preferred. Additionally, there is another advantage for feature selection. When we have determined the selected features, we only need to calculate or collect these concerning features. In feature learning, however, all features are still needed for dimensionality reduction.

Consequently, many researches have been proposed to address the problem of feature selection in the past few years. There are mainly two different kinds of feature selection approaches: supervised and unsupervised. Since we have no label information in unsupervised feature selection, it is more difficult than supervised scenario and there are relatively fewer investigations dedicated to this topic. Most unsupervised feature selection approaches are either based on filters [Dash *et al.*, 2002] [Nie *et al.*, 2008], wrappers [Roth and Lange, 2004] or embeddings [Dy *et al.*, 2004]. Although the performances of traditional unsupervised feature selection approaches are prominent in many cases, their efficiencies can also be improved since (1) from the view of manifold learning [Cai *et al.*, 2007], the high dimensional data are nearly lying on a low dimensional manifold. Traditional methods have not taken fully considerations about the manifold structure. (2) Different from feature learning, traditional feature selection approaches only employ data statistical character to rank the features essentially. They are often lack of using the learning mechanism as in feature learning, which is proved to be powerful and widely used in many areas [Nie *et al.*, 2010b].

Recently, to leverage both the manifold structure and learning mechanism, some investigations have emerged. Typical methods include: Pca Score (PcaScor) [Krzanowski, 1987], Laplacian Score (LapScor) [He *et al.*, 2005], Spectral Feature Selection (SPEC) [Zhao and Liu, 2007], Multi-Cluster Feature Selection (MCFS) [Cai *et al.*, 2010] and Minimum Redundancy Spectral Feature selection (MRSF) [Zhao *et al.*, 2010]. Commonly, these methods use various graphs to characterize manifold structure at first. LapScor and SPEC then compute different metrics to rank each feature. MCFS and MRSF, however, add sparse constraints in multi-output regression. Compared with traditional unsupervised feature selection approaches, these methods have proved to perform better in many cases [Zhao *et al.*, 2010]. Nevertheless, their performances can also be improved since these methods all separate manifold characterization and feature selection. Once the graph is determined to characterize manifold structure, it is fixed in the following ranking or regression steps. Thus, the performance of feature selection is largely determined by the effectiveness of graph construction. On the contrary, if the graph laplacian can adaptively change w.r.t. the following ranking or regression procedures, i.e., the graph not only can characterize manifold structure, but also indicate the requirements of regression, these methods would perform better.

In this paper, we introduce a novel unsupervised feature

selection approach via Joint Embedding Learning and Sparse Regression (JELSR). Instead of simply using the graph laplacian to characterize high dimensional data's structure and then regression, we propose to adopt locally linear approximation weight to construct a new graph and unify these two objectives in forming a new problem. By adding the $\ell_{2,1}$-norm regularization, we can learn a sparse transformation matrix for feature selection. We also provide an effective method to solve the proposed problem. Compared with traditional unsupervised feature selection approaches, our method could integrate the merits of manifold learning and sparse regression. Many experimental results are provided for demonstration.

The rest of this paper is organized as follows. We will formulate JELSR and provide an effective solution algorithm in Section 2. Section 3 will present the convergence behavior and relations to other approaches. Section 4 provides some comparing results on various kinds of data sets, followed by the conclusion and future works in Section 5.

2 Feature Selection via Joint Embedding Learning and Sparse Regression

In this section, we will first introduce some notations. The concrete formulation is then proposed. Finally, we provide an effective algorithm to solve this problem.

Before going into the details of our algorithm, let us introduce some notations. Denote $\{\mathbf{x}_i \in \mathbb{R}^d | i = 1, 2, \cdots, n\}$ as the unlabeled examples. We would like to select s features to represent the original data, where $s < d$. For a matrix $\mathbf{Q} \in \mathbb{R}^{u \times v}$, its $\ell_{r,p}$-norm is defined as follows.

$$\|\mathbf{Q}\|_{r,p} = \left(\sum\nolimits_{i=1}^{u} \left(\sum\nolimits_{j=1}^{v} |\mathbf{Q}_{ij}|^r \right)^{p/r} \right)^{1/p}. \quad (1)$$

For brief, the ℓ_2-norm is also denoted as $\|\cdot\|_2$ in the following. Define $\alpha > 0$, $\beta > 0$ as two balance parameters.

2.1 Formulations

There are mainly three objective functions of our algorithm. We would like to introduce them in sequence.

Considering that Spectral Regression (SR) performs well in feature learning and graph laplacian could fully characterize manifold structure, we would like to inherit their advantages in formulating our feature selection algorithm. Evoked by the intuition that nearby points should have similar properties, we would like to construct a weight graph $\mathcal{G} = (\mathcal{V}, E, \mathbf{S})$ to reveal their local connections, where $\mathcal{V} = \{\mathbf{x}_i\}$ is the graph vertex set and E contains edges of the constructed graph.

The key point in constructing graph is to determine its weight matrix $\mathbf{S}$, where $\mathbf{S}_{ij}$ reveals the similarity between points $\mathbf{x}_i$ and $\mathbf{x}_j$. Commonly, the graph is constructed by connecting every point to its k-nearest neighbors and the weights for connected points are computed by gaussian function. Motivated by the prominent performance in using locally linear approximation weight to construct graph [Roweis and Saul, 2000], we would like to employ similar strategy to measure local similarity. More concretely, the graph is constructed by following steps:

Step 1. Constructing a k-nearest graph $\mathcal{G}$. The i-th node corresponds to $\mathbf{x}_i$. For $\mathbf{x}_i$, it only connects with the points in its k-nearest neighborhood set $\mathcal{N}(\mathbf{x}_i)$. Thus, $\mathcal{G}$ is a directed graph.

Step 2. Computing the similarity matrix $\mathbf{S}$. For the i-th point $\mathbf{x}_i$, its weight $\mathbf{S}_{ij} > 0$ if and only if $\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)$. Otherwise, $\mathbf{S}_{ij} = 0$. The nonzero weight is determined by using the following locally linear approximation strategy.

$$\arg \min_{\mathbf{S},\ \sum_j \mathbf{S}_{ij}=1} = \sum_{i=1}^{n} \|\mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i)} \mathbf{S}_{ij}\mathbf{x}_j\|_2^2. \quad (2)$$

Recalling the basic idea of feature learning, we will represent the original data $\mathbf{x}_i$ by its low dimensional embedding, i.e., $\mathbf{y}_i \in \mathbb{R}^m$, where m is the dimensionality of embedding. Through this kind of replacement, the most valuable information is retained and the feature redundancies are eliminated. As a result, the first objective is

$$\arg \min_{\mathbf{Y}\mathbf{Y}^T = \mathbf{I}_{m \times m}} \sum_{i=1}^{n} \|\mathbf{y}_i - \sum_{j=1}^{n} \mathbf{S}_{ij}\mathbf{y}_j\|_2^2 = tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) \quad (3)$$

where $\mathbf{L} = (\mathbf{I}_{n \times n} - \mathbf{S})^T(\mathbf{I}_{n \times n} - \mathbf{S})$ is the graph laplacian. $\mathbf{y}_i \in \mathbb{R}^m$ is the embedding of $\mathbf{x}_i$ for $i = 1, 2, \cdots, n$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]$.

As in SR, the second objective function of our algorithm is to regress each sample to its low dimensional embedding. More concretely, assume $\{\mathbf{x}_i\}$ is centered and denote $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$ as the matrix formulated by all transformation vectors $\{\mathbf{w}_i\}_{i=1}^m$, the second objective function is

$$\arg \min_{\mathbf{W}} \sum_{i=1}^{n} \|\mathbf{W}^T\mathbf{x}_i - \mathbf{y}_i\|_2^2 = \|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2. \quad (4)$$

The third objective function is designed for feature selection. Denote $\hat{\mathbf{w}}_i$ as the i th row vector of $\mathbf{W}$, i.e.,

$$\mathbf{W} = \left[\hat{\mathbf{w}}_1^T, \hat{\mathbf{w}}_2^T, \cdots, \hat{\mathbf{w}}_d^T\right]^T. \quad (5)$$

Essentially, $\hat{\mathbf{w}}_i$ corresponds to the transformation vector of the i-th feature in regression. It can also be regarded as a vector that measures the importance of the i-th feature.

Considering the task of feature selection, we expect that the transformation matrix holds the sparsity property for feature selection. More concretely, we expect that only a few numbers of $\hat{\mathbf{w}}_i$ are non-zeros. As a result, the corresponding features are selected since these features are enough to regress the original data $\mathbf{x}_i$ to its low dimensional representation $\mathbf{y}_i$. When we employ the 2-norm of $\hat{\mathbf{w}}_i$ as a metric to measure its contribution in this regression, the sparsity property, i.e., a small number of $\hat{\mathbf{w}}_i$ are non-zeros, indicates the following objective function.

$$\arg \min_{\mathbf{W}} \sum_{i=1}^{d} \|\hat{\mathbf{w}}_i\|_2 = \sum_{i=1}^{d} (\sum_{j=1}^{m} W_{ij}^2)^{1/2} = \|\mathbf{W}\|_{2,1}. \quad (6)$$

Here $\|\mathbf{W}\|_{2,1}$ denotes the $\ell_{2,1}$-norm as defined in Eq. (1).

By combining the objective functions in Eq. (3), Eq. (4) with Eq. (6), our JELSR algorithm can be formulated as follows.

$$\begin{aligned} \mathcal{L}(\mathbf{W}, \mathbf{Y}) = \arg \min_{\mathbf{W}, \mathbf{Y}\mathbf{Y}^T = \mathbf{I}_{m \times m}} \\ tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) + \beta(\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1}), \end{aligned} \quad (7)$$

where α and β are two balance parameters.

After deriving $\mathbf{W}$, we use the 2-norm of $\hat{\mathbf{w}}_i$, i.e., $\|\hat{\mathbf{w}}_i\|_2$ to rank the features. The larger $\|\hat{\mathbf{w}}_i\|_2$ is, the more important this feature is. We can either select a fixed number of the most important features or set a threshold and select the feature whose $\|\hat{\mathbf{w}}_i\|_2$ is larger than this value. In the following, we select a fixed number, i.e., s, features for evaluation.

2.2 Solutions

Considering the optimization problem in Eq. (7), since we have added the $\ell_{2,1}$-norm regularization for feature selection, it is hard to derive its closed solution directly. Inspired by [Nie *et al.*, 2010a], we will solve this problem in an alternative way. As we will explain later, through this kind of procedure, we update the embedding $\mathbf{Y}$ and the sparse regression matrix $\mathbf{W}$ alternatively. In other words, we select the features by joining embedding learning and sparse regression, which has not been considered in the literature.

Note that $\|\mathbf{W}\|_{2,1}$ is convex. Nevertheless, its derivative does not exist when $\hat{\mathbf{w}}_i = \mathbf{0}$ for $i = 1, 2, \cdots, d$. For convenience, we would like to denote $\mathcal{L}(\mathbf{W}) = \|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1}$. Thus, when $\hat{\mathbf{w}}_i \neq \mathbf{0}$ for $i = 1, 2, \cdots, d$, the derivative of $\mathcal{L}(\mathbf{W})$ w.r.t. $\mathbf{W}$ is

$$\frac{\partial \mathcal{L}(\mathbf{W})}{\mathbf{W}} = 2\mathbf{X}\mathbf{X}^T\mathbf{W} - 2\mathbf{X}\mathbf{Y}^T + 2\alpha\mathbf{U}\mathbf{W}, \tag{8}$$

where $\mathbf{U} \in \mathbb{R}^{d\times d}$ is a diagonal matrix whose i-th diagonal element is

$$\mathbf{U}_{ii} = \frac{1}{2\|\hat{\mathbf{w}}_i\|_2}. \tag{9}$$

As seen from Eq. (8), we construct an auxiliary function. It is obvious that the derivative in Eq. (8) can also be regarded as the derivative of the following objective function.

$$\mathcal{C}(\mathbf{W}) = \|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha tr(\mathbf{W}^T\mathbf{U}\mathbf{W}). \tag{10}$$

Consequently, we try to solve the following problem to approximate the solution to Eq. (7).

$$\mathcal{L}(\mathbf{W}, \mathbf{U}, \mathbf{Y}) = \arg \min_{\mathbf{W},\mathbf{U},\mathbf{Y}\mathbf{Y}^T=\mathbf{I}_{m\times m}} tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) + \beta(\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha tr(\mathbf{W}^T\mathbf{U}\mathbf{W})) \tag{11}$$

where $\mathbf{U}$ is defined as in Eq. (9).

We would like to explain why we can deriving a sparse solution by minimizing Eq. (11). Recalling the definition of $\mathbf{U}_{ii}$ in Eq (9), we know that $tr(\mathbf{W}^T\mathbf{U}\mathbf{W}) = \|\mathbf{W}\|_{2,1}/2$ when $\hat{\mathbf{w}}_i$ is not equal to 0. Thus, the objective of minimizing $tr(\mathbf{W}^T\mathbf{U}\mathbf{W})$ will add the sparsity constraint on $\mathbf{W}$. Intuitively, if $\|\hat{\mathbf{w}}_i\|_2$ is small, then $\mathbf{U}_{ii}$ is large and the minimization of Eq. (10) tends to derive $\hat{\mathbf{w}}_i$ with much smaller ℓ_2-norm. After several times of iteration, the norms of some $\hat{\mathbf{w}}_i$s are close to zero and we get a sparse $\mathbf{W}$. Besides, the above approximation can not be used if $\hat{\mathbf{w}}_i \neq \mathbf{0}$ for $i = 1, 2, \cdots, d$.

As seen from above formulation, the objective function in Eq. (11) is convex with respect to $\mathbf{W}$ and $\mathbf{Y}$ if $\mathbf{U}$ is fixed. When $\mathbf{W}$ is fixed, we can determine $\mathbf{U}$ by Eq. (9) directly. Thus, we update $\mathbf{W}$ and $\mathbf{Y}$ when $\mathbf{U}$ is fixed and compute $\mathbf{U}$ when $\mathbf{W}$ is fixed.

When $\mathbf{U}$ is fixed, we would like to take the derivative of $\mathcal{L}(\mathbf{W}, \mathbf{U}, \mathbf{Y})$ with respect to $\mathbf{W}$ and set it to zero, i.e., we have the following equation.

$$\frac{\partial \mathcal{L}(\mathbf{W}, \mathbf{Y}, \mathbf{U})}{\mathbf{W}} = 2\mathbf{X}\mathbf{X}^T\mathbf{W} - 2\mathbf{X}\mathbf{Y}^T + 2\alpha\mathbf{U}\mathbf{W} = 0 \tag{12}$$

or equivalently,

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^T + \alpha\mathbf{U})^{-1}\mathbf{X}\mathbf{Y}^T. \tag{13}$$

By substituting above $\mathbf{W}$ into Eq. (11), we will have

$$\begin{aligned}
&\mathcal{L}(\mathbf{W}, \mathbf{U}, \mathbf{Y}) \\
=&tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) + \beta(\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha tr(\mathbf{W}^T\mathbf{U}\mathbf{W})) \\
=&tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) + \beta(tr(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}) - 2tr(\mathbf{W}^T\mathbf{X}\mathbf{Y}^T) \\
&\qquad + tr(\mathbf{Y}\mathbf{Y}^T) + \alpha tr(\mathbf{W}^T\mathbf{U}\mathbf{W})) \\
=&tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) + \beta(-tr(\mathbf{W}^T(\mathbf{X}\mathbf{X}^T + \alpha\mathbf{U})\mathbf{W}) + tr(\mathbf{Y}\mathbf{Y}^T))
\end{aligned} \tag{14}$$

Denote $\mathbf{A} = \mathbf{X}\mathbf{X}^T + \alpha\mathbf{U}$, Eq. (14) becomes

$$\begin{aligned}
&\mathcal{L}(\mathbf{W}, \mathbf{U}, \mathbf{Y}) \\
=&tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) + \beta tr(\mathbf{Y}\mathbf{Y}^T) - \beta tr(\mathbf{Y}\mathbf{X}^T\mathbf{A}^{-1}\mathbf{X}\mathbf{Y}^T) \\
=&tr(\mathbf{Y}(\mathbf{L} + \beta\mathbf{I}_{n\times n} - \beta\mathbf{X}^T\mathbf{A}^{-1}\mathbf{X})\mathbf{Y}^T)
\end{aligned} \tag{15}$$

Considering the objective function in Eq. (15) and the constraint $\mathbf{Y}\mathbf{Y}^T = \mathbf{I}_{m\times m}$, the optimization problem becomes

$$\begin{aligned}
&\arg\min_{\mathbf{Y}} tr(\mathbf{Y}(\mathbf{L} + \beta\mathbf{I}_{n\times n} - \beta\mathbf{X}^T\mathbf{A}^{-1}\mathbf{X})\mathbf{Y}^T) \\
&\text{s.t. } \mathbf{Y}\mathbf{Y}^T = \mathbf{I}_{m\times m}
\end{aligned} \tag{16}$$

If $\mathbf{A}$ and $\mathbf{L}$ are fixed, the optimization problem in Eq. (16) can be solved by eigen-decomposition of matrix $(\mathbf{L} + \beta\mathbf{I}_{n\times n} - \beta\mathbf{X}^T\mathbf{A}^{-1}\mathbf{X})$. We pick up the eigenvectors corresponding to the m smallest eigenvalues.

When $\mathbf{W}$ is fixed, we can update $\mathbf{U}$ by employing the formulation in Eq. (9) directly.

In summary, we solve the optimization problem in Eq. (7) in an alternative way. More concretely, if $\mathbf{U}$ is fixed, we can first solve the optimization problem in Eq. (16) to update $\mathbf{Y}$ and then employ Eq. (13) to update $\mathbf{W}$. After that, we fix $\mathbf{W}$ and update $\mathbf{U}$, which is defined in Eq. (9).

We now explain why our method could join embedding learning and sparse regression. Considering the above algorithm, we solve the problem in Eq. (16) to compute $\mathbf{Y}$. In other words, the objective of sparse regression has also affected the derivation of low dimensional embedding, i.e., $\mathbf{Y}$. Traditional methods, such as MCFS and MRSF, minimize $tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T)$ merely.

Additionally, since JELSR is solved in an alternative way, we would like to initialize $\mathbf{U}$ by an identity matrix. The experimental results show that our algorithm converges fast by using this kind of initialization. The number of iteration is often less than twenty. Besides, the above solving procedure can not be used when $\hat{\mathbf{w}}_i = \mathbf{0}$ for $i = 1, 2, \cdots, d$. This is another reason why we use this type of initialization.

In summary, the procedure of JELSR is listed in Table 1.

Table 1: Procedure of JELSR.

Input: Data set $\{\mathbf{x}_i|i=1,2,\cdots,n\}$; Balance parameter α β; Neighborhood size k; Dimensionality of embedding m; Selected feature number s.

Output: Selected feature index set $\{r_1, r_2, \cdots, r_s\}$.

Stage one: Graph construction

1. Construct the nearest neighborhood graph $\mathcal{G}$;
2. Compute the similarity matrix $\mathbf{S}$, graph laplacian $\mathbf{L}$;

Stage two: Alternative optimization

1. Initialize $\mathbf{U} = \mathbf{I}_{d\times d}$;
2. Alternatively update $\mathbf{U}$, $\mathbf{Y}$ and $\mathbf{W}$ until convergence.
   a. Fix $\mathbf{U}$, update $\mathbf{Y}$ by solving the problem in Eq. (16), update $\mathbf{W}$ by using Eq. (13);
   b. Fix $\mathbf{W}$, update $\mathbf{U}$ by Eq. (9);

Stage three: Feature selection

1. Compute the scores for all features $\{\|\hat{\mathbf{w}}_i\|_2\}_{i=1}^d$;
2. Sort these scores and select the largest s values. Their corresponding indexes form the selected feature index set $\{r_1, r_2, \cdots, r_s\}$.

3 Discussions

In this section, we will analyze JELSR in two aspects, i.e., the convergence behavior and the relations to other approaches.

3.1 Convergence Analysis

As seen from Table 1, since we have solved JELSR in an alternative way, we would like to show its convergence behavior. First, a lemma [Nie *et al.*, 2010a] is provided.

Lemma 1. *For any non-zero vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$, the following result follows*

$$\|\mathbf{a}\|_2 - \frac{\|\mathbf{a}\|_2^2}{2\|\mathbf{b}\|_2} \leq \|\mathbf{b}\|_2 - \frac{\|\mathbf{b}\|_2^2}{2\|\mathbf{b}\|_2}. \tag{17}$$

The convergence behavior of JELSR is summarized in the following theorem.

Theorem 1. *The optimization procedure in the second stage of Table 1 will monotonically decrease the objective of the problem in Eq. (7) in each iteration.*

Proof. As seen from the algorithm in Table 1, when we fix $\mathbf{U}$ as $\mathbf{U}^t$ in the t-th iteration and compute $\mathbf{W}^{t+1}$, $\mathbf{Y}^{t+1}$, the following inequality holds,

$$\begin{aligned} &tr(\mathbf{Y}^{t+1}\mathbf{L}(\mathbf{Y}^{t+1})^T) + \beta(\|(\mathbf{W}^{t+1})^T\mathbf{X} - \mathbf{Y}\|_2^2 \\ &\quad + \alpha tr((\mathbf{W}^{t+1})^T\mathbf{U}^t\mathbf{W}^{t+1})) \\ \leq &tr(\mathbf{Y}^t\mathbf{L}(\mathbf{Y}^t)^T) + \beta(\|(\mathbf{W}^t)^T\mathbf{X} - \mathbf{Y}\|_2^2 \\ &\quad + \alpha tr((\mathbf{W}^t)^T\mathbf{U}^t\mathbf{W}^t)). \end{aligned} \tag{18}$$

Since $\|\mathbf{W}\|_{2,1} = \sum_{i=1}^d \|\hat{\mathbf{w}}_i\|_2$, the above inequality indicates

$$\begin{aligned} &tr(\mathbf{Y}^{t+1}\mathbf{L}(\mathbf{Y}^{t+1})^T) + \beta(\|(\mathbf{W}^{t+1})^T\mathbf{X} - \mathbf{Y}\|_2^2 \\ &\quad + \alpha\|\mathbf{W}^{t+1}\|_{2,1} + \alpha\sum_{i=1}^d(\frac{\|\hat{\mathbf{w}}_i^{t+1}\|_2^2}{2\|\hat{\mathbf{w}}_i^t\|_2} - \|\hat{\mathbf{w}}_i^{t+1}\|_2)) \\ \leq &tr(\mathbf{Y}^t\mathbf{L}(\mathbf{Y}^t)^T) + \beta(\|(\mathbf{W}^t)^T\mathbf{X} - \mathbf{Y}\|_2^2 \\ &\quad + \alpha\|\mathbf{W}^t\|_{2,1} + \alpha\sum_{i=1}^d(\frac{\|\hat{\mathbf{w}}_i^t\|_2^2}{2\|\hat{\mathbf{w}}_i^t\|_2} - \|\hat{\mathbf{w}}_i^t\|_2)). \end{aligned} \tag{19}$$

Recalling the results in Lemma 1, we know that

$$\frac{\|\hat{\mathbf{w}}_i^{t+1}\|_2^2}{2\|\hat{\mathbf{w}}_i^t\|_2} - \|\hat{\mathbf{w}}_i^{t+1}\|_2 \geq \frac{\|\hat{\mathbf{w}}_i^t\|_2^2}{2\|\hat{\mathbf{w}}_i^t\|_2} - \|\hat{\mathbf{w}}_i^t\|_2. \tag{20}$$

Combining Eq. (19) with Eq. (20), we have the following results.

$$\begin{aligned} &tr(\mathbf{Y}^{t+1}\mathbf{L}(\mathbf{Y}^{t+1})^T) + \beta(\|(\mathbf{W}^{t+1})^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha\|\mathbf{W}^{t+1}\|_{2,1}) \\ &\leq tr(\mathbf{Y}^t\mathbf{L}(\mathbf{Y}^t)^T) + \beta(\|(\mathbf{W}^t)^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha\|\mathbf{W}^t\|_{2,1}). \end{aligned} \tag{21}$$

This inequality indicates that the objective function in Eq. (7) will monotonically decrease in each iteration. □

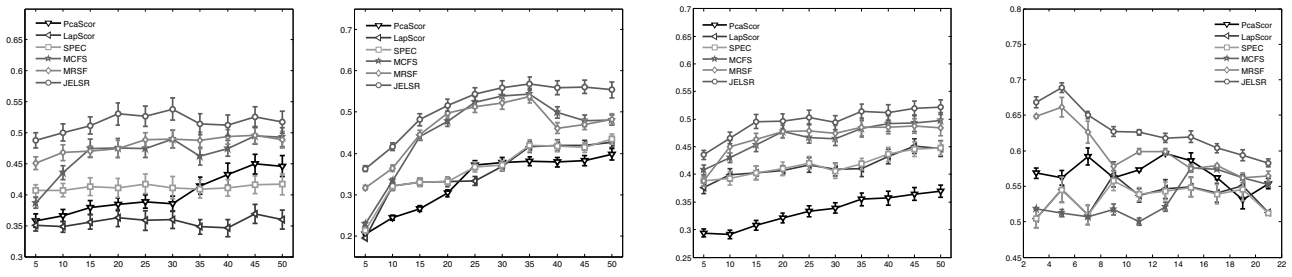
Additionally, since the objective function has lower bounds, such as zero, the above iteration will converge. Besides, the following experimental results show that it converges fast, the time of iteration is often less than 20.

3.2 Relations to other Approaches

First, considering the above deduction of our algorithm, we know that JELSR is related to LapScor [He *et al.*, 2005]. LapScor selects features that can best preserve the similarity revealed by $\mathbf{S}$. Both JELSR and LapScor construct a graph to characterize data manifold. JELSR uses the locally linear approximation weights while LapScor employs the gaussian function. Compared with LapScor, JELSR uses a more prominent way, i.e., sparse regression, to learn feature weights. It could inherit the metric of SR [Cai *et al.*, 2007]. Moreover, SPEC is also based on spectral analysis and it can be regarded as an extension of LapScor. Nevertheless, SPEC exploits both labeled and unlabeled data through a regularization framework and emphasizes the problem of semi-supervised feature selection. JELSR, however, focuses on the unsupervised case.

Second, JELSR has close relationship with MCFS [Cai *et al.*, 2010]. MCFS computes the same embedding as Laplacian Eigenmaps [Belkin and Niyogi, 2003] and then regresses each point to this embedding by adding the ℓ_1 norm regularization. Compared with MCFS, JELSR uses a different graph to characterize data structure. More importantly, JELSR unifies the procedures of embedding learning and sparse regression, which are separated in MCFS. Thus, it performs better than MCFS in many cases.

Finally, JELSR also has close relationship with MRSF [Zhao *et al.*, 2010]. MRSF first computes the embedding by eigen-decomposition of graph laplacian and then regresses



(a) (b) (c) (d)

Figure 1: Acc of Kmeans on four data sets with different number of selected features. The x-axis represents the number of selected features and the y-axis is the Acc results. (a) Umist; (b) Isolet; (c) Orl; (d) Sonar.

with $\ell_{2,1}$ norm regularization. More concretely, MRSF can be regarded as solving the following two problems in sequence.

$$\begin{aligned} &\arg \min_{\mathbf{Y}\mathbf{Y}^T=\mathbf{I}_{m\times m}} tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) \\ &\arg \min_{\mathbf{W}} \|\mathbf{W}^T\mathbf{X}-\mathbf{Y}\|_2^2+\alpha\|\mathbf{W}\|_{2,1}. \end{aligned} \tag{22}$$

Comparing the formulation in Eq. (22) with that in Eq. (7), we can see the main difference between JELSR and MRSF is that JELSR unifies the two objectives of MRSF. In other words, JELSR could join the procedures of embedding learning and sparse regression. MRSF separates these two steps. Thus, its performance is largely determined by the effectiveness of graph construction. More importantly, we have the following result.

Theorem 2. *If we compute the graph laplacian* $\mathbf{L}$ *in Eq. (3) by employing gaussian function, MRSF in Eq. (22) can be regarded as a special case of JELSR in Eq. (7) when* $\beta \to 0$.

The above theorem indicates that JELSR can be regarded as a unified framework in viewing different learning based feature selection approaches.

4 Experiments

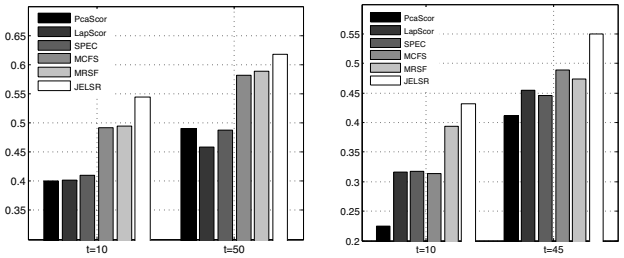
We present three different groups of experiments. The first group is the clustering results of Kmeans on different data with different numbers of selected features. To show whether the performance comparison of different algorithms are dominated by clustering method, we propose to use another approach, i.e, Normalized Cut(Ncut), for clustering in the second group. Finally, since there are mainly two different parameters, i.e., α and β, we would like to provide the results with different parameters. Let us describe the experimental data sets at first.

There are mainly four different types of data sets. They are images, including Umist, Orl, and others, including Isolet5 and Sonar data. Their sizes range from about 200 to about 1600. The dimensionality ranges from about 30 to about 1500. Two different metrics, including clustering Accuracy (Acc) and Normalized Mutual Information (NMI) are employed to measure the clustering performances. We compare our algorithm with other learning based unsupervised feature selection approaches, including PcaScor, LapScor, SPEC, MCFS and MRSF.

In the first group, we employ Kmeans for clustering by repeating 100 times. With different numbers of selected features, the Acc and NMI results are show in Fig. 1, Table 2 and Table 3, where the parameters are selected by grid search in a heuristic way. Other parameters, such as k and m, are empirically determined as in traditional subspace learning approaches.

As seen from Fig. 1, Table 2 and Table 3, it is clear that JELSR performs better than other approaches in most cases. Besides, although Acc and NMI are two different metrics, they both indicate the advantages of our algorithm.

In the second group, we employ another clustering algorithm, i.e., Ncut, for evaluation. On Umist data, we select $s = 10$ and $s = 50$ features. On Isolet data, we set $s = 10$ and $s = 45$ for illustration. The Acc results are in Fig. 2.



(a) (b)

Figure 2: Acc of Ncut on different data sets with different number of selected features. (a) Umist with the selected feature numbers s=10 and $s = 50$; (b) Isolet with the selected feature numbers s=10 and $s = 45$.

Similarly, although we employ different clustering methods, JELSR also outperforms other algorithms.

Finally, we first determine two parameters by grid search and then change them within certain ranges. The Acc results of Kmeans with different α and β are shown in Fig. 3.

As seen from Fig. 3, when two parameter are changed within a certain range, the performance of JELSR also changes within a certain range.

5 Conclusion

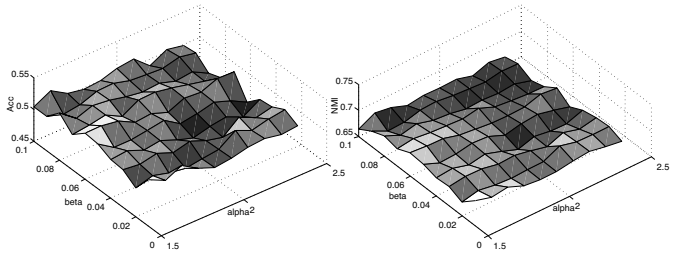
In this paper, we proposed a novel unsupervised feature selection algorithm. Different from traditional methods, our

Table 2: NMI of different methods on Umist data set with different numbers of selected features by using Kmeans for clustering (mean $\pm$ std).

s	PcaScor	LapScor	SPEC	MCFS	MRSF	JELSR
s=5	0.5049±0.0151	0.4985±0.0088	0.4614±0.0087	0.5286±0.0121	0.5839±0.0124	**0.6234**±0.0160
s=15	0.5337±0.0122	0.5178±0.0147	0.5191±0.0145	0.6260±0.0182	0.6142±0.0124	**0.6515**±0.0187
s=25	0.5450±0.0132	0.5271±0.0152	0.5268±0.0152	0.6418±0.0204	0.6535±0.0167	**0.6888**±0.0204
s=35	0.5854±0.0150	0.5268±0.0171	0.5293±0.0167	0.6352±0.0201	0.6549±0.0167	**0.6886**±0.0250
s=45	0.6248±0.0131	0.5493±0.0172	0.5339±0.0150	0.6651±0.0195	0.6699±0.0186	**0.6984**±0.0193

Table 3: NMI of different methods on Isolet data set with different numbers of selected features by using Kmeans for clustering (mean $\pm$ std).

s	PcaScor	LapScor	SPEC	MCFS	MRSF	JELSR
s=5	0.3608±0.0047	0.3588±0.0047	0.3667±0.0048	0.3778±0.0055	0.4816 ±0.0052	**0.5139**±0.0064
s=15	0.3813±0.0077	0.4971±0.0076	0.4958±0.0078	0.5769±0.0083	0.5736 ±0.0094	**0.6153**±0.0152
s=25	0.5110±0.0107	0.5035±0.0093	0.5401±0.0081	0.6612±0.0145	0.6561 ±0.0106	**0.6900**±0.0155
s=35	0.5391±0.0114	0.6026±0.0100	0.6055±0.0097	**0.7034**±0.0134	0.6799 ±0.0145	**0.7272**±0.0159
s=45	0.5583±0.0115	0.6107±0.0099	0.6059±0.0098	0.6703±0.0146	0.6544 ±0.0107	**0.7191**±0.0151



(a) (b)

Figure 3: Acc and NMI of Kmeans on Umist data set when α varies from 1.5 to 2.4 and β varies from 1e-2 to 1e-1. (a) The Acc results; (b) The NMI results.

algorithm could combine embedding learning and sparse regression. We also provided an efficient algorithm to solve the $\ell_{2,1}$-norm regularization problem. The convergence behavior was also analyzed. Further research includes the extension of JELSR to supervised case. We will also focus on the accelerating issue of our algorithm.

Acknowledgement

This work is supported by NFSC, China, No. 61005003, 60975038.

References

[Belkin and Niyogi, 2003] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15:1373–1396, 2003.

[Cai *et al.*, 2007] Deng Cai, Xiaofei He, and Jiawei Han. Spectral regression for efficient regularized subspace learning. In *Proc. Int. Conf. Computer Vision (ICCV'07)*, 2007.

[Cai *et al.*, 2010] Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. KDD '10, pages 333–342, 2010.

[Dash *et al.*, 2002] Manoranjan Dash, Kiseok Choi, Peter Scheuermann, and Huan Liu. Feature selection for clustering - a filter solution. ICDM '02, pages 115–122, 2002.

[Dy *et al.*, 2004] Jennifer G. Dy, Carla E. Brodley, and Stefan Wrobel. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, 2004.

[He *et al.*, 2005] Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In *NIPS*, 2005.

[Krzanowski, 1987] W. J. Krzanowski. Selection of Variables to Preserve Multivariate Data Structure, Using Principal Components. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36(1):22–33, 1987.

[Nie *et al.*, 2008] Feiping Nie, Shiming Xiang, Yangqing Jia, Changshui Zhang, and Shuicheng Yan. Trace ratio criterion for feature selection. In *AAAI*, 2008.

[Nie *et al.*, 2010a] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. In *NIPS 23*. 2010.

[Nie *et al.*, 2010b] Feiping Nie, Dong Xu, Ivor Wai-Hung Tsang, and Changshui Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE TIP*, 19(7):1921–1932, 2010.

[Roth and Lange, 2004] Volker Roth and Tilman Lange. Feature selection in clustering problems. In *NIPS 16*, 2004.

[Roweis and Saul, 2000] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[Zhao and Liu, 2007] Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, pages 1151–1157, 2007.

[Zhao *et al.*, 2010] Zheng Zhao, Lei Wang, and Huan Liu. Efficient spectral feature selection with minimum redundancy. In *AAAI*, 2010.