Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Unsupervised feature selection via joint local learning and group sparse regression*

Yue WU[1,2], Can WANG[‡1,2], Yue-qing ZHANG[1], Jia-jun BU[1,2]

[1]*Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*

[2]*Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China*

E-mail: wy1988@zju.edu.cn; wcan@zju.edu.cn; 704787221@qq.com; bjj@zju.edu.cn

Received Dec. 2, 2017; Revision accepted Mar. 9, 2018; Crosschecked Apr. 11, 2019

Abstract: Feature selection has attracted a great deal of interest over the past decades. By selecting meaningful feature subsets, the performance of learning algorithms can be effectively improved. Because label information is expensive to obtain, unsupervised feature selection methods are more widely used than the supervised ones. The key to unsupervised feature selection is to find features that effectively reflect the underlying data distribution. However, due to the inevitable redundancies and noise in a dataset, the intrinsic data distribution is not best revealed when using all features. To address this issue, we propose a novel unsupervised feature selection algorithm via joint local learning and group sparse regression (JLLGSR). JLLGSR incorporates local learning based clustering with group sparsity regularized regression in a single formulation, and seeks features that respect both the manifold structure and group sparse structure in the data space. An iterative optimization method is developed in which the weights finally converge on the important features and the selected features are able to improve the clustering results. Experiments on multiple real-world datasets (images, voices, and web pages) demonstrate the effectiveness of JLLGSR.

Key words: Unsupervised; Local learning; Group sparse regression; Feature selection
https://doi.org/10.1631/FITEE.1700804 **CLC number:** TP391.4

1 Introduction

Nowadays, real-world applications are confronted by big data of increasingly higher dimensionalities. High-dimensional data not only contain more information but also introduce extra redundancies and noise. Furthermore, high-dimensional data significantly increase the time and space requirements. Learning algorithms excel in low-dimensional data become completely impractical in the high-dimensional space. This phenomenon, known as "curse of dimensionality" (Bellman, 1961; Verleysen, 2003), has become a prevalent problem for learning algorithms with high-dimensional data. To address this issue, various dimensionality reduction techniques have been proposed. These methods can be categorized mainly into two classes, feature selection and feature extraction. Feature selection methods, such as the Fisher score and Laplacian score (LS) (He et al., 2005), choose a relevant feature subset to represent the original data. Feature extraction methods, such as principal component analysis (PCA) (Jolliffe, 2002), locally linear embedding (LLE) (Roweis and Saul, 2000), locality preserving projections (He and Niyogi, 2004), locality minimizing globality maximizing projections (Nie et al., 2009), and flexible manifold embedding (Nie et al., 2010b), transform the original data into reduced representations. Compared with feature

‡ Corresponding author

* Project supported by Alibaba-Zhejiang University Joint Institute of Frontier Technologies and Zhejiang Provincial Key Research and Development Plan (No. 2017C01012)

ORCID: Can WANG, http://orcid.org/0000-0002-5890-4307

extraction, feature selection does not change the representation of the original data. Consequently, feature selection methods can better preserve the actual meaning of features in the learning process and provide more interpretability for the learned results.

According to whether label information is used or not, feature selection methods can be divided into two categories, supervised and unsupervised. Supervised feature selection methods (Peng et al., 2005; Nie et al., 2010a; Tan et al., 2010) select features based on the correlation between features and labels. In contrast, unsupervised feature selection methods find the optimal feature subset that best preserves the data distribution. Considering the difficulties in obtaining labels, unsupervised feature selection methods are more widely used in practice. However, the lack of label information has also brought added challenges to the development of unsupervised feature selection methods. Therefore, unsupervised feature selection methods have attracted much more research interest compared to supervised ones in recent years.

Because existing studies have shown that data spaces are often low-dimensional manifolds embedded within high-dimensional ambient spaces (Roweis and Saul, 2000; Tenenbaum et al., 2000; Belkin and Niyogi, 2001), many feature selection methods that take advantage of the manifold structures have been proposed, including the Laplacian score (He et al., 2005), trace ratio criterion for feature selection (Nie et al., 2008), eigenvalue sensitive feature selection (Jiang and Ren, 2011), multi-cluster feature selection (Cai et al., 2010), local kernel regression score (Cheung and Zeng, 2009), and feature selection for local learning based clustering (Zeng and Cheung, 2009, 2011). These methods either explicitly consider the manifold structure in the model formulations or incorporate regularizations or constraints in the models to select features that respect the intrinsic manifold structure in the data space. Consequently, performances of the feature selection methods in image and document spaces have been widely verified.

However, the methods mentioned above use all of the features to reveal the intrinsic structures. Thus, they are quite likely affected by the noisy and redundant features in the dataset. The results obtained may be unreliable and the feature subset selected based on this structure may not be the best candidate. To address this problem, we propose a novel unsupervised feature selection method via joint local learning and group sparse regression (JLLGSR), which combines local learning based clustering with group sparse regression to perform feature selection. By local learning based clustering, the manifold structure of the original data space is learned, while by group sparse regression, related features are selected according to the clustering results. Jointly solving these two problems can simultaneously boost the structure learning process and optimize the clustering results. As a result, the feature subset that best respects the manifold structure and the group sparse structure in the data space is selected.

JLLGSR is fundamentally based on our previous work in group sparse feature selection on local learning based clustering (GSFS-llc) (Wu et al., 2016) with major improvements in the model formulation. In contrast to GSFS-llc, which performs local learning based clustering and sparse regression in two separate steps, JLLGSR combines clustering and regression into one single objective function. The omission of the intermediate steps can help JLLGSR achieve better optimization in feature selection, and JLLGSR consequently demonstrates better performance. The main contributions of this study can be summarized as follows:

1. To the best of our knowledge, JLLGSR is the first algorithm that incorporates local learning based clustering with group sparse regression in a single model, which makes JLLGSR capable of correcting the cluster structure with selected features and reducing the impact of noise and redundancies.

2. Compared with multi-cluster feature selection (MCFS) (Cai et al., 2010), GSFS-llc (Wu et al., 2016), and joint embedding learning and sparse regression (JELSR) (Hou et al., 2014), a new bias term is introduced in the sparse regression model to help improve the generalization capability of JLLGSR.

3. An alternative and iterative optimization algorithm is exploited to efficiently solve the proposed method along with its convergence and computational complexity analysis.

2 Related work

Based on how a learning algorithm is incorporated into the evaluation and selection of features, feature selection methods can be categorized into

filter methods, wrapper methods, and embedded methods (Guyon and Elisseeff, 2003). Filter methods aim to select features according to certain inner statistical properties (variance, Pearson correlation, and mutual information) of the data before running the learning algorithm (He et al., 2005; Zhao and Liu, 2007; Jiang and Ren, 2011). Wrapper methods select a feature subset based on the scores provided by a specific predictive model trained with the candidate subset (Guyon et al., 2002; Doquire and Verleysen, 2013). Embedded methods perform feature selection with specific learning machines in the training process (Cai et al., 2010; Tan et al., 2010; Yang et al., 2011; Zeng and Cheung, 2011; Hou et al., 2014).

Many existing feature selection studies focus on selecting feature subsets by respecting the intrinsic geometric structure of the data space. Existing studies such as isometric feature mapping (Tenenbaum et al., 2000) and LLE (Roweis and Saul, 2000) have shown that data samples lie on a low-dimensional manifold that is embedded in a high-dimensional ambient space. This manifold assumption has been verified in many existing datasets such as USPS, Yale, and COIL100. Nie et al. (2010b, 2011) proposed locality-based algorithms to reveal the intrinsic manifold structure in the data space. Nie et al. (2016a) proposed a parameter-free method called the constrained Laplacian rank algorithm, which exactly constructs a graph with x (equal to the number of clusters) connected components. Many recent feature selection studies attempt to incorporate manifold assumption by choosing features that respect the manifold structure. The local kernel regression (LKR) score (Cheung and Zeng, 2009) selects features that not only minimize the within-neighborhood estimation error but also maximize the overall variance, which can efficiently deal with both supervised and unsupervised scenarios using different neighborhood graphs. A local learning based feature selection method (Sun et al., 2010) decomposes an arbitrarily complex nonlinear problem into a set of locally linear problems through local learning, and then learns feature relevance globally within the large margin framework. The unsupervised feature selection method based on local learning based clustering (LLC-fs) (Zeng and Cheung, 2009, 2011) selects optimal features by incorporating a binary selection vector into the local learning based clustering objective function.

Another related research area exploring the important features in the data space is sparse regression, which is also widely used in many other machine learning applications, including image annotation (Han et al., 2012) and video segmentation (Han et al., 2015). By using different sparsity-inducing regularizations such as least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996) (using l_1-norm) and elastic net (Zou and Hastie, 2005) (using a combination of l_1-norm and l_2-norm), various sparse bases can be retrieved through sparse regression under different assumptions. Because vectors use LASSO or elastic net to achieve sparsity, the matrix gains group sparsity by incorporating the $l_{2,1}$-norm which first calculates the l_2-norm for each row of the matrix and then sums the results to form an l_1-norm. That is $\|\boldsymbol{M} \in \mathbb{R}^{p\times q}\|_{2,1} = \sum_{i=1}^{p}\sqrt{\sum_{j=1}^{q} m_{ij}^2}$. Group sparsity exhibits more stability in noisy datasets than LASSO because the coefficients related to different labels tend to share the same sparse pattern. The coefficients are more likely to include or exclude as a whole group in group sparse regression. Note that sparse regression can efficiently compute the correlation between samples and their labels (or embedding results). This characteristic of sparse regression makes it a desirable choice when feature selection is integrated with learning methods. Cai et al. (2010) proposed a multi-cluster feature selection (MCFS) which obtains the cluster information using spectral embedding and then solves the sparse coefficients through a series of l_1-norm regularized least squares regression problems. Wang et al. (2014) proposed an unsupervised feature selection method using unsupervised trace ratio formulation regularized by the $l_{2,1}$-norm of the projection matrix. Chang et al. (2016) proposed the convex sparse PCA, which incorporates $l_{2,1}$-norm minimization into a low-rank regression optimization problem and selects features based on the coefficients under the PCA criteria. Wu et al. (2016) proposed a group sparse regression based feature selection method called group sparse feature selection on local learning based clustering (GSFS-llc). It combines local learning based clustering (LLC) with group sparse regression and achieves better feature selection performance than MCFS.

Note that in most existing feature selection methods, all of the features are used to analyze the intrinsic data structure. However, this process

is prone to noises and redundancies in the original data and renders the results unreliable. Some cutting-edge feature selection methods attempt to overcome this problem by integrating feature selection in the structure learning process. Hou et al. (2014) proposed JELSR, which integrates the merits of embedding learning and sparse regression. Du and Shen (2015) proposed an unsupervised feature selection with adaptive structure learning (FSASL). The structures are adaptively learned from the results of feature selection, while the informative features are reselected to preserve the refined structures of the data. Nie et al. (2016b) proposed an unsupervised feature selection with structured graph optimization (SOGFS), which also performs feature selection and local structure learning simultaneously, where the similarity matrix can thus be adaptively determined and contains more accurate information on the data structure. Luo et al. (2018) proposed adaptive unsupervised feature selection with structure regularization, which simultaneously learns the selective matrix with the optimal reconstruction. Inspired by JELSR, we propose JLLGSR using local learning based clustering instead of embedding learning to analyze the manifold structure of the data. FSASL adaptively learns both the global and local structures with the candidate feature subset, while in JLLGSR, the candidate feature subset boosts the structure learning process, improves the clustering results obtained from all features, and reduces the impact of noisy and redundant features. This difference makes JLLGSR more stable than FSASL, because an unsupervised scenario directly using the candidate feature subset to learn the intrinsic data structure may mislead the structure learning process and result in loss of information.

3 Unsupervised feature selection via joint local learning and group sparse regression

In this section, we describe how to formulate and solve JLLGSR. Given a set of data points $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_N]^{\mathrm{T}}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\dots,\boldsymbol{f}_M]$, N is the number of samples, M is the total number of features, $\boldsymbol{x}_i\in\mathbb{R}^M$ denotes a sample point, and $\boldsymbol{f}_j\in\mathbb{R}^N$ denotes a feature. $\mathcal{N}_i$ represents the set of $\boldsymbol{x}_i$'s neighbors and $n_i=|\mathcal{N}_i|$ is its cardinality. C denotes the number of clusters. Let $K:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ be a positive definite kernel function. The kernel matrix of $\boldsymbol{x}_i$'s neighbors can be defined as $\boldsymbol{K}_i=[K(\boldsymbol{x}_u,\boldsymbol{x}_v)]\in\mathbb{R}^{n_i\times n_i}$ for $\boldsymbol{x}_u,\boldsymbol{x}_v\in\mathcal{N}_i$, and $\boldsymbol{k}_i=[K(\boldsymbol{x}_i,\boldsymbol{x}_j)]$ for all $\boldsymbol{x}_j\in\mathcal{N}_i$. Different kernel functions can be adopted, such as the linear kernel $K(\boldsymbol{x}_u,\boldsymbol{x}_v)=\boldsymbol{x}_u^{\mathrm{T}}\boldsymbol{x}_v$ and heat kernel $K(\boldsymbol{x}_u,\boldsymbol{x}_v)=\exp(-\frac{\|\boldsymbol{x}_u-\boldsymbol{x}_v\|_2^2}{2\sigma^2})$. Let d be the number of features that we want to select.

3.1 Using local learning based clustering to analyze data distribution

The data space can be regarded as linear in a small neighborhood under the manifold assumption. Therefore, using neighbors to learn a linear model to approximate the label of a sample is quite reasonable. Thus, we use kernel regression, and the l^{th} element of the sample's cluster indicator can be estimated as

$$\hat{y}_i^l=\sum_{\boldsymbol{x}_j\in\mathcal{N}_i}\beta_{ij}^l K(\boldsymbol{x}_i,\boldsymbol{x}_j). \tag{1}$$

By introducing an l_2-norm, Eq. (1) turns into a kernel ridge regression problem. In addition, the coefficient β_{ij}^l can be easily obtained by solving the following optimization problem:

$$\underset{\boldsymbol{\beta}_i^l\in\mathbb{R}^{n_i}}{\operatorname{argmin}}\|\boldsymbol{K}_i\boldsymbol{\beta}_i^l-\boldsymbol{y}_i^l\|^2+\lambda(\boldsymbol{\beta}_i^l)^{\mathrm{T}}\boldsymbol{K}_i\boldsymbol{\beta}_i^l, \tag{2}$$

where $\boldsymbol{\beta}_i^l=[\beta_{i1}^l,\beta_{i2}^l,\dots,\beta_{in_i}^l]^{\mathrm{T}}\in\mathbb{R}^{n_i}$ is the coefficient vector, $\boldsymbol{y}_i^l=[y_1^l,y_2^l,\dots,y_{n_i}^l]^{\mathrm{T}}\in\mathbb{R}^{n_i}$. $\lambda>0$ is the parameter of regularization.

The solution to problem (2) is $\boldsymbol{\beta}_i^l=(\boldsymbol{K}_i+\lambda\boldsymbol{I})^{-1}\boldsymbol{y}_i^l$. Substituting it back into Eq. (1) can lead to

$$\hat{y}_i^l=\boldsymbol{k}_i^{\mathrm{T}}(\boldsymbol{K}_i+\lambda\boldsymbol{I})^{-1}\boldsymbol{y}_i^l, \tag{3}$$

$$\boldsymbol{\alpha}_i^{\mathrm{T}}=\boldsymbol{k}_i^{\mathrm{T}}(\boldsymbol{K}_i+\lambda\boldsymbol{I})^{-1}. \tag{4}$$

$\boldsymbol{\alpha}_i^{\mathrm{T}}$ is determined only by $\boldsymbol{k}_i$, $\boldsymbol{K}_i$, and λ, and can be easily calculated without the cluster indicator $\boldsymbol{y}_i^l$. Then we can use a linear combination form to express the estimated cluster indicator $\hat{y}_i^l$ as

$$\hat{y}_i^l=\boldsymbol{\alpha}_i^{\mathrm{T}}\boldsymbol{y}_i^l. \tag{5}$$

Finally, the overall prediction error can be calculated:

$$\begin{aligned}\sum_{l=1}^{C}\sum_{i=1}^{N}(y_i^l-\hat{y}_i^l)^2&=\sum_{l=1}^{C}\|\boldsymbol{y}^l-\hat{\boldsymbol{y}}^l\|^2\\&=\sum_{l=1}^{C}\|\boldsymbol{y}^l-\boldsymbol{A}\boldsymbol{y}^l\|^2\\&=\operatorname{tr}\big(\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{A})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{Y}\big)\\&=\operatorname{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y}),\end{aligned} \tag{6}$$

where C is the number of clusters. $\boldsymbol{A}=[a_{ij}]$ is an $N\times N$ sparse matrix; a_{ij} equals the corresponding element of $\boldsymbol{\alpha}_i$ in Eq. (4) if $\boldsymbol{x}_j\in\mathcal{N}_i$, and otherwise it is set to 0. $\boldsymbol{T}=(\boldsymbol{I}-\boldsymbol{A})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{A})$. $\boldsymbol{Y}=[\boldsymbol{y}^1,\boldsymbol{y}^2,\dots,\boldsymbol{y}^C]$ is the cluster indicator matrix which we want to solve.

Eq. (6) captures the data distribution. By minimizing Eq. (6), we can obtain the partition matrix $\boldsymbol{Y}$ of the data.

3.2 Using group sparse regression to analyze the contribution of each feature

Suppose that the partition matrix $\boldsymbol{Y}$ is known. A simple idea to retrieve important features is to use regression.

$$\boldsymbol{Y}=\boldsymbol{X}\hat{\boldsymbol{W}}+\boldsymbol{1}\boldsymbol{b}, \tag{7}$$

where $\hat{\boldsymbol{W}}$ are the regression coefficients which can be used to measure the importance of each feature. $\boldsymbol{b}$ is the bias. By adding a column vector $\boldsymbol{1}$ to the right end of $\boldsymbol{X}$, the bias term $\boldsymbol{b}$ can be merged into the coefficient matrix $\boldsymbol{W}$.

$$\boldsymbol{Y}=[\boldsymbol{X},\boldsymbol{1}]\begin{bmatrix}\hat{\boldsymbol{W}}\\ \boldsymbol{b}\end{bmatrix}=[\boldsymbol{X},\boldsymbol{1}]\,\boldsymbol{W}. \tag{8}$$

For convenience, in the rest of the paper, symbol $\boldsymbol{X}$ refers to the modified data matrix $[\boldsymbol{X},\boldsymbol{1}]$.

Eq. (8) can be easily solved by optimizing the following least squares regression problem:

$$\underset{\boldsymbol{W}}{\operatorname{argmin}}\,\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2. \tag{9}$$

By incorporating an $l_{2,1}$-norm regularizer, Eq.(9) turns into

$$\underset{\boldsymbol{W}}{\operatorname{argmin}}\,\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2+\gamma\|\boldsymbol{W}\|_{2,1}. \tag{10}$$

The $l_{2,1}$-norm regularizer $\|\boldsymbol{W}\|_{2,1}=\sum_{i=1}^{M+1}\sqrt{\sum_{j=1}^{C}w_{ij}^2}$ makes $\boldsymbol{W}$ smooth in the rows and sparse in the columns, which makes it a good choice for feature selection. The sparsity in columns shows the importance of different features. The smoothness in rows means that the corresponding feature has good performance in discriminating all the clusters from the others.

3.3 Combining local learning and group sparse regression to formulate the unsupervised feature selection method

From the two subsections above, we already have a method to analyze the data distribution and a method to evaluate the importance of each feature. If we simply perform these two methods one after the other, it will lead to good feature selection results, such as the methods proposed in Cai et al. (2010) and Wu et al. (2016). Nevertheless, this is not good enough. Because the data distribution is calculated using all of the features, within which irrelevant features or various kinds of noise could exist. If we want to reduce the influence of noise, the analysis of data distribution and the selection of important features should be simultaneously performed.

We combine Eqs. (6) and (10) and formulate the following objective function:

$$\begin{aligned}&\underset{\boldsymbol{W},\boldsymbol{Y}}{\operatorname{argmin}}\,\operatorname{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y})+\delta(\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2+\gamma\|\boldsymbol{W}\|_{2,1})\\ &\quad\text{s.t. }\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}.\end{aligned} \tag{11}$$

Note that it is hard to derive a close solution to Eq. (11), so we use an alternative and iterative method to solve Eq. (11) like the method proposed in Hou et al. (2014) and Nie et al. (2010a).

Denote $L(\boldsymbol{W},\boldsymbol{Y})=\operatorname{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y})+\delta(\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2+\gamma\|\boldsymbol{W}\|_{2,1})$. As we know, the derivative of $\|\boldsymbol{W}\|_{2,1}$ does not exist, if $\|\boldsymbol{w}_i\|_2=0$ $(i=1,2,\dots,M+1)$ where $\boldsymbol{w}_i$ is the row vector of $\boldsymbol{W}$. Thus, we add a small constraint to $L(\boldsymbol{W},\boldsymbol{Y})$. When $\|\boldsymbol{w}_i\|_2\neq 0$ $(i=1,2,\dots,M+1)$, the derivative of $L(\boldsymbol{W},\boldsymbol{Y})$ with respect to $\boldsymbol{W}$ is

$$\frac{\partial L(\boldsymbol{W},\boldsymbol{Y})}{\partial\boldsymbol{W}}=2\delta\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}-2\delta\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}+2\gamma\delta\boldsymbol{U}\boldsymbol{W}, \tag{12}$$

where $\boldsymbol{U}$ is an $(M+1)\times(M+1)$ diagonal matrix whose i^{th} diagonal element is

$$U_{ii}=\frac{1}{2\|\boldsymbol{w}_i\|_2}. \tag{13}$$

Note that $\frac{\partial\operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{W})}{\partial\boldsymbol{W}}=\boldsymbol{U}\boldsymbol{W}+\boldsymbol{U}^{\mathrm{T}}\boldsymbol{W}=2\boldsymbol{U}\boldsymbol{W}$, and it is reasonable using $\operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{W})$ to approximate $\|\boldsymbol{W}\|_{2,1}$. Thus, Eq. (11) can be approximated by

$$\underset{\boldsymbol{W},\boldsymbol{Y}}{\operatorname{argmin}}\,\operatorname{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y})+\delta(\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2+\gamma\operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{W})). \tag{14}$$

To solve Eq. (14), we first fix $\boldsymbol{U}$ and $\boldsymbol{Y}$ to optimize $\boldsymbol{W}$. Denote $L(\boldsymbol{W},\boldsymbol{Y},\boldsymbol{U})=\operatorname{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y})+\delta(\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2+\gamma\operatorname{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{W}))$. Let $\frac{\partial L(\boldsymbol{W},\boldsymbol{Y},\boldsymbol{U})}{\partial\boldsymbol{W}}=0$. We can obtain

$$\boldsymbol{W}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\gamma\boldsymbol{U})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}. \tag{15}$$

Substituting $\boldsymbol{W}$ in Eq. (15) back into Eq. (14), we have

$$\begin{aligned}
&\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y})+\delta(\|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2+\gamma\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{W}))\\
=&\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y})+\delta(\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})-2\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W})\\
&+\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W})+\gamma\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{W}))\\
=&\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y})+\delta(\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})-2\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W})\\
&+\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\gamma\boldsymbol{U})\boldsymbol{W}))\\
=&\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y})+\delta(\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})\\
&-2\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\gamma\boldsymbol{U})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y})\\
&+\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\gamma\boldsymbol{U})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}))\\
=&\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{T}+\delta(\boldsymbol{I}-\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\gamma\boldsymbol{U})^{-1}\boldsymbol{X}^{\mathrm{T}}))\boldsymbol{Y}).
\end{aligned}\tag{16}$$

Then objective function (14) is

$$\begin{aligned}
&\underset{\boldsymbol{Y}}{\mathrm{argmin}}\ \mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{T}+\delta(\boldsymbol{I}-\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\gamma\boldsymbol{U})^{-1}\boldsymbol{X}^{\mathrm{T}}))\boldsymbol{Y})\\
&\text{s.t. } \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}=\boldsymbol{I}.
\end{aligned}\tag{17}$$

If $\boldsymbol{U}$ is fixed, Eq. (17) can be easily solved by eigen-decomposition of the matrix $\boldsymbol{T}+\delta(\boldsymbol{I}-\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\gamma\boldsymbol{U})^{-1}\boldsymbol{X}^{\mathrm{T}})$. Since the number of clusters C is usually unknown in an unsupervised scenario, we introduce a preset parameter for the number of used eigenvectors u instead. Then the solution of $\boldsymbol{Y}$ is given by the eigenvectors corresponding to the top u smallest eigenvalues and u is usually set close to the number of clusters C in practice.

By iteratively optimizing/calculating $\boldsymbol{Y}$, $\boldsymbol{W}$, and $\boldsymbol{U}$ through Eqs. (17), (15), and (13), the optimization problem (14) will lead to a convergent solution, which can also be regarded as the solution to the original optimization problem (11).

After obtaining the coefficient matrix $\boldsymbol{W}$, the JLLGSR score for each feature can be defined by the summation of its corresponding coefficients' absolute values:

$$\mathrm{Score}_{\mathrm{JLLGSR}}(j)=\sum_i|w_{ji}|.\tag{18}$$

With the ranking score, the best candidate feature subset can be determined by choosing the top d features according to their JLLGSR scores in descending order. The complete JLLGSR is summarized in Algorithm 1.

Algorithm 1 Joint local learning and group sparse regression

Construct the k-nearest-neighbor graph
Compute the kernel matrix $\boldsymbol{K}$
$\boldsymbol{\alpha}_i^{\mathrm{T}}\leftarrow\boldsymbol{k}_i^{\mathrm{T}}(\boldsymbol{K}_i+\lambda\boldsymbol{I})^{-1}$
Compute the sparse matrix $\boldsymbol{A}$ based on $\boldsymbol{\alpha}_i$
$\boldsymbol{T}\leftarrow(\boldsymbol{I}-\boldsymbol{A})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{A})$
Initialize $\boldsymbol{U}\leftarrow\boldsymbol{I}$
loop
  Fix $\boldsymbol{U}$, and update $\boldsymbol{Y}$ by optimizing Eq. (17)
  Fix $\boldsymbol{U}$ and $\boldsymbol{Y}$, and update $\boldsymbol{W}$ by Eq. (15)
  Fix $\boldsymbol{W}$, and update $\boldsymbol{U}$ by Eq. (13)
  if Eq. (11) converges **then**
    break
  end if
end loop
$\mathrm{Score}_{\mathrm{JLLGSR}}(j)\leftarrow\sum_i|w_{ji}|$
Sort all features according to their JLLGSR scores in descending order and return the top d features

3.4 Convergence analysis

Because we have solved the objective function of JLLGSR iteratively in the above section, it is necessary to show its convergence. Let $\boldsymbol{Y}^t$ and $\boldsymbol{W}^t$ be the optimization results of the t^{th} iteration, and $\boldsymbol{U}^t$ can be easily calculated according to $\boldsymbol{W}^t$. In the $(t+1)^{\mathrm{th}}$ iteration, $\boldsymbol{Y}^{t+1}$ and $\boldsymbol{W}^{t+1}$ can be obtained from the optimization function with fixed $\boldsymbol{U}^t$. Since $\boldsymbol{Y}^{t+1}$ and $\boldsymbol{W}^{t+1}$ are the optimum of the $(t+1)^{\mathrm{th}}$ iteration, the following inequality holds:

$$\begin{aligned}
&\mathrm{tr}((\boldsymbol{Y}^{t+1})^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y}^{t+1})+\delta(\|\boldsymbol{Y}^{t+1}-\boldsymbol{X}\boldsymbol{W}^{t+1}\|_{\mathrm{F}}^2\\
&\quad+\gamma\mathrm{tr}((\boldsymbol{W}^{t+1})^{\mathrm{T}}\boldsymbol{U}^t\boldsymbol{W}^{t+1}))\\
\le&\mathrm{tr}((\boldsymbol{Y}^t)^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y}^t)+\delta(\|\boldsymbol{Y}^t-\boldsymbol{X}\boldsymbol{W}^t\|_{\mathrm{F}}^2\\
&\quad+\gamma\mathrm{tr}((\boldsymbol{W}^t)^{\mathrm{T}}\boldsymbol{U}^t\boldsymbol{W}^t)).
\end{aligned}\tag{19}$$

Note that

$$\begin{cases}
\mathrm{tr}((\boldsymbol{W}^{t+1})^{\mathrm{T}}\boldsymbol{U}^t\boldsymbol{W}^{t+1})=\sum_i\frac{\|\boldsymbol{w}_i^{t+1}\|_2^2}{2\|\boldsymbol{w}_i^t\|_2},\\
\mathrm{tr}((\boldsymbol{W}^t)^{\mathrm{T}}\boldsymbol{U}^t\boldsymbol{W}^t)=\sum_i\frac{\|\boldsymbol{w}_i^t\|_2^2}{2\|\boldsymbol{w}_i^t\|_2},\\
\|\boldsymbol{W}\|_{2,1}=\sum_i\|\boldsymbol{w}_i\|_2.
\end{cases}\tag{20}$$

Eq. (19) turns into

$$\begin{aligned}
&\mathrm{tr}((\boldsymbol{Y}^{t+1})^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y}^{t+1})+\delta\Bigg(\|\boldsymbol{Y}^{t+1}-\boldsymbol{X}\boldsymbol{W}^{t+1}\|_{\mathrm{F}}^2\\
&+\gamma\|\boldsymbol{W}^{t+1}\|_{2,1}+\gamma\sum_i\left(\frac{\|\boldsymbol{w}_i^{t+1}\|_2^2}{2\|\boldsymbol{w}_i^t\|_2}-\|\boldsymbol{w}_i^{t+1}\|_2\right)\Bigg)\\
\le&\mathrm{tr}((\boldsymbol{Y}^t)^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y}^t)+\delta\Bigg(\|\boldsymbol{Y}^t-\boldsymbol{X}\boldsymbol{W}^t\|_{\mathrm{F}}^2\\
&+\gamma\|\boldsymbol{W}^t\|_{2,1}+\gamma\sum_i\left(\frac{\|\boldsymbol{w}_i^t\|_2^2}{2\|\boldsymbol{w}_i^t\|_2}-\|\boldsymbol{w}_i^t\|_2\right)\Bigg).
\end{aligned}\tag{21}$$

Because we know that for any nonzero vectors $\boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^m$, $\frac{\|\boldsymbol{a}\|_2^2}{2\|\boldsymbol{b}\|_2}-\|\boldsymbol{a}\|_2\ge\frac{\|\boldsymbol{b}\|_2^2}{2\|\boldsymbol{b}\|_2}-\|\boldsymbol{b}\|_2$, which was

proven in Nie et al. (2010a), the following inequality holds:

$$\sum_i \left(\frac{\|\boldsymbol{w}_i^{t+1}\|_2^2}{2\|\boldsymbol{w}_i^t\|_2} - \|\boldsymbol{w}_i^{t+1}\|_2 \right) \geq \sum_i \left(\frac{\|\boldsymbol{w}_i^t\|_2^2}{2\|\boldsymbol{w}_i^t\|_2} - \|\boldsymbol{w}_i^t\|_2 \right). \tag{22}$$

Considering inequalities (21) and (22), we can easily obtain the following:

$$\begin{aligned} &\mathrm{tr}((\boldsymbol{Y}^{t+1})^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y}^{t+1}) + \delta(\|\boldsymbol{Y}^{t+1} - \boldsymbol{X}\boldsymbol{W}^{t+1}\|_{\mathrm{F}}^2 \\ &+ \gamma\|\boldsymbol{W}^{t+1}\|_{2,1}) \\ \leq &\mathrm{tr}((\boldsymbol{Y}^t)^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y}^t) + \delta(\|\boldsymbol{Y}^t - \boldsymbol{X}\boldsymbol{W}^t\|_{\mathrm{F}}^2 + \gamma\|\boldsymbol{W}^t\|_{2,1}). \end{aligned} \tag{23}$$

This means that the objective function (11) is monotonically decreasing in each iteration. Note that $\mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{Y}) = \mathrm{tr}(\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{A})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{A})\boldsymbol{Y}) \geq 0$ and $\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_{\mathrm{F}}^2 + \gamma\|\boldsymbol{W}\|_{2,1} \geq 0$. Thus, the objective function is also larger than or equal to zero. Therefore, the iteration in JLLGSR can converge.

3.5 Computational complexity analysis

We present a brief analysis of the computational complexities of JLLGSR (N, sample size; M, feature size; k, number of nearest neighbors; u, number of eigenvectors used):

1. Computing the kernel matrix $\boldsymbol{K}$ will incur a cost of $O(N^2M)$.

2. Computing the sparse matrix $\boldsymbol{A}$ requires finding the k nearest neighbors of $\boldsymbol{x}_i$ and computing $\boldsymbol{\alpha}_i$ in Eq. (4) for each $\boldsymbol{x}_i$, and the time complexity is about $O(N^2k + Nk^3)$.

3. The optimization problem (11) is solved iteratively, where fixing $\boldsymbol{U}$ and updating $\boldsymbol{Y}$ by optimizing Eq. (17) take about $O(NM^2+M^3+N^2M+N^3)$, fixing $\boldsymbol{U}$ and $\boldsymbol{Y}$ and updating $\boldsymbol{W}$ by Eq. (15) take about $O(NM^2 + M^3 + MNu)$, and fixing $\boldsymbol{W}$ and updating $\boldsymbol{U}$ by Eq. (13) take about $O(Mu)$. Since the iteration converges very quickly ($O(1)$ can be omitted), the total time complexity is about $O(N^3 + N^2M + NM^2 + M^3 + MNu + Mu)$.

4. Computing the JLLGSR score requires $O(Mu)$ and sorting features by their JLLGSR scores requires $O(M \log M)$.

Because k and u are usually very small constants, the corresponding terms can be ignored. Thus, JLLGSR's computational complexity is about $O(N^3 + N^2M + NM^2 + M^3 + M \log M)$, which is comparable to those of other group sparsity based feature selection methods.

4 Experiments

In this section, we conduct experiments on various datasets to evaluate the performance of JLLGSR. First, we use a small handwritten digit dataset to show the effectiveness of JLLGSR. Then we compare JLLGSR with state-of-the-art feature selection algorithms on various real-world benchmarks/datasets to demonstrate the superiority of JLLGSR. Finally, experiments on varying parameters are performed to study the influence of different parameter selections.

4.1 Data sets and evaluation metrics

The experiments are conducted on six datasets, including handwritten digits (USPS08), voices (ISOLET4), human faces (YaleB), object images (COIL100, CIFAR10), and web sites (WPAE). USPS08 contains 2261 handwritten digit images of 16×16 size of digit zero and digit eight chosen from the famous handwritten digit database USPS (Hull, 1994). ISOLET4 is the fourth subset of the spoken letter recognition dataset ISOLET (Fanty and Cole, 1990), which contains 1558 samples with 617 features. YaleB is a combination of the Yale face database B (Georghiades et al., 2001) and the extended Yale face database B, which contains 2414 near frontal images under different illuminations of 38 individuals and is cropped to a 32×32 size (Lee et al., 2005). The Columbia object image library of 100 objects is COIL100, which contains a total of 7200 (72 images on each object) 32×32 images taken five degrees apart as the object is rotated on a turntable with 256 grey levels per pixel. CIFAR10 is a dataset containing 60 000 color images of 32×32 size in 10 classes (Krizhevsky, 2009), where we extract a 512-dimensional gist feature vector to represent each image. WPAE is used in web accessibility evaluation, consisting of 4300 web pages crawled from 43 web sites. Each sample contains 57 features, which are the related HTML tags appearing in the web page. All datasets used in the experiments are summarized in Table 1.

Two evaluation metrics are used to quantitatively evaluate the clustering performance, clustering accuracy (AC), and normalized mutual information (NMI). Let t_i and l_i denote the cluster label obtained and the true label of sample x_i, respectively, and map$(\cdot)$ the permutation mapping function

Table 1 Statistics for the datasets

Dataset	Number of samples	Number of features	Number of classes
USPS08	2261	256	2
ISOLET4	1558	617	26
YaleB	2414	1024	38
COIL100	7200	1024	100
CIFAR10	60 000	512	10
WPAE	4300	57	43

which uses the Hungarian algorithm (Kuhn, 1955; Munkres, 1957) to find the optimal label mapping that can produce the largest number of matching pairs between the cluster labels obtained and the true labels. AC is defined as

$$\mathrm{AC}=\frac{1}{n}\sum_{i=1}^{n}\delta(\mathrm{map}(t_i),l_i), \tag{24}$$

where n is the number of samples. $\delta(u,v)=1$ if $u=v$; otherwise, $\delta(u,v)=0$. Let C denote the ground truth set of clusters and C' the set of clusters obtained from the clustering algorithm. The mutual information between C and C' is then defined as

$$\mathrm{MI}(C,C')=\sum_{c\in C,c'\in C'}p(c,c')\cdot\log_2\frac{p(c,c')}{p(c)\cdot p(c')}, \tag{25}$$

where $p(c)$ and $p(c')$ are the probabilities that a sample randomly selected from the data belongs to the clusters c and c', respectively, and $p(c,c')$ is the joint probability that the arbitrarily selected sample belongs to clusters c and c' simultaneously. Then we can define NMI as

$$\mathrm{NMI}(C,C')=\frac{\mathrm{MI}(C,C')}{\max(\mathrm{H}(C),\mathrm{H}(C'))}, \tag{26}$$

where $\mathrm{H}(C)$ and $\mathrm{H}(C')$ denote the entropies of C and C', respectively. It is quite straightforward to see that $\mathrm{NMI}(C,C')\in[0,1]$. $\mathrm{NMI}(C,C')=1$ if C and C' are identical and $\mathrm{NMI}(C,C')=0$ if C and C' are independent.

4.2 Experiment setup

To validate the effectiveness of JLLGSR, we compare JLLGSR with six state-of-the-art feature selection algorithms:

1. LS (He et al., 2005) selects features that preserve local similarities and maximize the variances.

2. LKR (Cheung and Zeng, 2009) seeks the features that minimize the within-neighborhood estimation error and maximize the variance over all the data samples.

3. MCFS (Cai et al., 2010) uses spectral embedding to obtain the cluster structure in a dataset and l_1-norm regularized least squares regression to select features that best preserve the cluster structure.

4. GSFS-llc (Wu et al., 2016) uses local learning based clustering to analyze data distribution and then uses group sparse regression to select the candidate feature subset.

5. JELSR (Hou et al., 2014) combines embedding learning with sparse regression to obtain the best candidate feature subset.

6. FSASL (Du and Shen, 2015) performs structure learning and feature selection simultaneously, and selects the features that best preserve the refined structures which are adaptively learned from the results of feature selection.

All the experiments are conducted on a 64-bit Linux server with two 2.4 GHz 6-core 12-thread CPUs and 256 GB memory. The parameters used in the following experiments are set as follows. For all these methods and all the datasets, we choose five nearest neighbors to build the neighbor graph. For LS, LKR, MCFS, GSFS-llc, and JLLGSR, the similarity matrix is calculated based on the heat kernel; for JELSR, the similarities are solved by the locally linear embedding algorithm. The number of eigenvectors used or the dimensionality of embedding in MCFS, GSFS-llc, JELSR, FSASL, and JLLGSR is set to be equal to the number of clusters. The regularization parameters γ and δ in JLLGSR are determined by a grid search within $\{0.001,0.01,0.1,1,10,100,1000\}$. After all the feature selection methods issue their candidate feature subset, k-means is employed to cluster the data with the selected features. Then AC and NMI are calculated based on the clustering results to evaluate the performance of each feature selection method.

4.3 Experimental results

4.3.1 Results for handwritten digits

The first experiment is conducted on USPS08 to directly show the effectiveness of JLLGSR. The visualized feature selection results from USPS08 are shown in Fig. 1. Digit zero is obviously differentiated from digit eight by the top 10 selected features. Figs. 1a and 1b display the mean images of digit zero and digit eight, respectively. Figs. 1c and 1d present the top 10 selected features of digit zero and digit

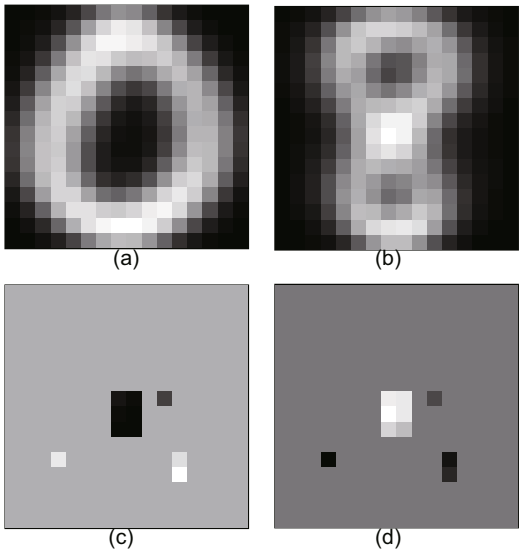

Fig. 1 Visualized feature selection results for USPS08: (a) mean image of digit zero; (b) mean image of digit eight; (c) top 10 features of zero; (d) top 10 features of eight

eight, respectively. The pixels of the top 10 selected features in the images show a clear contrast.

The clustering results for USPS08 are shown in Fig. 2 and Table 2. The number of selected features ranges from 10 to 50 and the best result of each row in Table 2 is displayed in bold font. The clustering results using all features are recorded in the last row of the table. As we can see, JLLGSR outperforms the other algorithms in terms of both clustering accuracy and normalized mutual information. Another important observation is that the clustering results from JLLGSR beat the results with all the features. This proves that JLLGSR is able to reduce the impact of redundancies and noise in the data.

We conduct another experiment on USPS08 to further evaluate the performance of JLLGSR in the presence of noise. Fig. 3 shows the original images and the images with salt & pepper noise in USPS08. The clustering results for the top 10 selected features with a noise density ranging from 0% to 30%

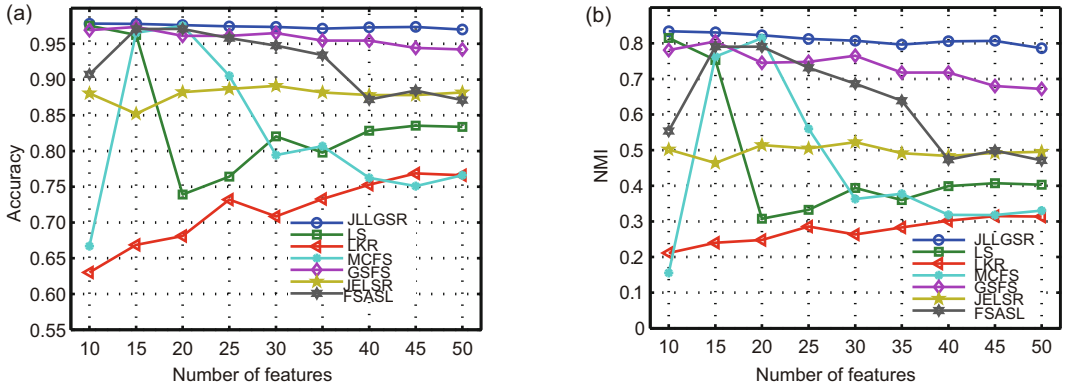

Fig. 2 Clustering results using features ranging from 10 to 50 in USPS08: (a) clustering accuracy; (b) normalized mutual information

Table 2 Clustering results for USPS08

d	Clustering accuracy (%)							Normalized mutual information (%)						
	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR
10	97.52	63.03	66.70	96.90	88.06	90.71	**97.83**	81.52	21.12	15.55	78.10	50.17	55.39	**83.38**
15	96.24	66.87	96.55	97.35	85.23	97.08	**97.79**	75.31	24.00	76.26	80.51	46.40	79.07	**83.08**
20	73.91	68.11	97.52	96.11	88.24	97.08	**97.61**	30.76	24.78	81.46	74.56	51.44	79.07	**82.28**
25	76.43	73.20	90.54	96.11	88.68	95.80	**97.43**	33.25	28.58	56.00	74.81	50.50	73.13	**81.27**
30	82.04	70.85	79.43	96.51	89.12	94.74	**97.35**	39.37	26.34	36.29	76.53	52.23	68.65	**80.75**
35	79.79	73.29	80.67	95.44	88.19	93.41	**97.13**	36.00	28.29	37.77	71.82	49.13	63.90	**79.67**
40	82.84	75.28	76.25	95.44	87.84	87.22	**97.30**	39.91	30.19	31.82	71.82	48.36	47.33	**80.57**
45	83.55	76.87	75.10	94.43	87.84	88.46	**97.35**	40.72	31.48	31.78	68.01	49.22	49.86	**80.69**
50	83.37	76.60	76.60	94.21	88.19	87.13	**96.99**	40.28	31.33	33.02	67.21	49.54	47.15	**78.64**
All	84.30	84.30	84.30	84.30	84.30	84.30	84.30	41.83	41.83	41.83	41.83	41.83	41.83	41.83

Parameter d denotes the number of selected features. Bold numbers denote the best results.

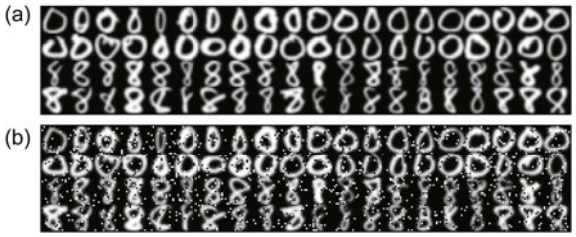

Fig. 3 Original images and images with noise in USPS08: (a) original images; (b) images with 10% salt & pepper noise

are shown in Fig. 4 and Table 3. As noise density increases, the clustering accuracy of JLLGSR slightly decreases from 97.83% to 91.82%, while the clustering accuracies of other methods fluctuate wildly. Therefore, JLLGSR tends to have more robust performance than any other feature selection method in the presence of noise.

4.3.2 Clustering results for real-world benchmarks

In this section, JLLGSR is compared with state-of-the-art feature selection algorithms on various real-world benchmarks. The clustering results are explained briefly to show the superiority of JLLGSR and to aid in understanding why it achieves such results.

The clustering results for ISOLET4 are presented in Fig. 5 and Table 4. Except for the results from the selection of the top 30 features, JLLGSR outperforms all the other algorithms in terms of both clustering accuracy and normalized mutual information. Even when choosing the top 30 selected features, the gap between the JLLGSR result and the best result is quite small. Furthermore, JLLGSR results beat the clustering results with all the features once again.

The clustering results from YaleB are shown in Fig. 6 and Table 5. The clustering accuracy of JLLGSR is lower than that of JELSR when selecting a small number of features (10–15), but the gap between them is quite small. When choosing a large enough feature subset ($d > 15$), JLLGSR outperforms the other methods. For normalized mutual information, JLLGSR is lower than JELSR only if choosing 20 or 35 features, and the gap between them is just 0.01, which can be ignored. The clustering results for JLLGSR beat the results with all the

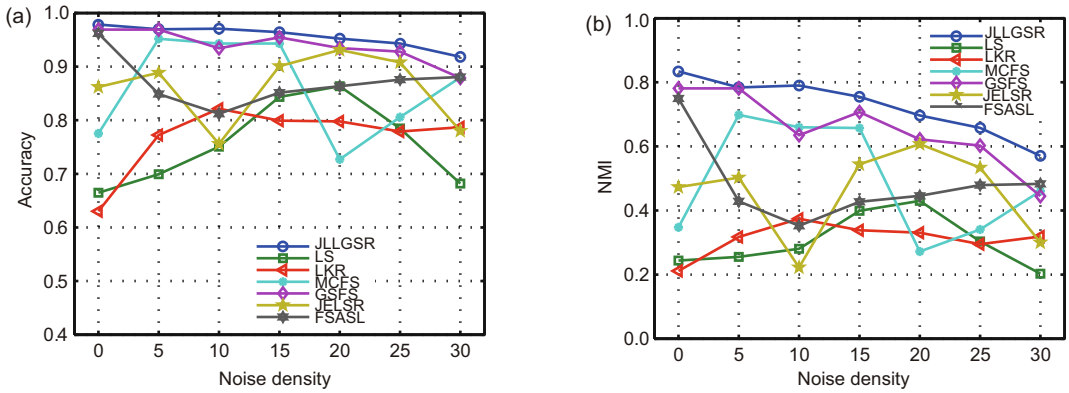

Fig. 4 Clustering results with noise density ranging from 0% to 30% for USPS08: (a) clustering accuracy; (b) normalized mutual information

Table 3 Clustering results for USPS08 with noise

Noise density	Clustering accuracy (%)							Normalized mutual information (%)						
	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR
0%	66.48	63.03	77.53	96.90	86.20	96.20	**97.83**	24.39	21.12	34.76	78.10	47.32	74.96	**83.38**
5%	69.92	77.22	95.18	96.90	88.85	84.87	**96.95**	25.51	31.73	69.86	78.12	50.26	42.84	**78.37**
10%	75.06	82.13	94.29	93.41	75.59	81.20	**97.08**	28.03	37.43	65.98	63.50	22.21	35.22	**79.01**
15%	94.34	79.92	94.29	95.44	90.09	85.14	**96.42**	39.90	33.81	65.76	70.71	54.44	42.73	**75.47**
20%	86.33	79.79	72.71	93.45	93.10	86.33	**95.22**	42.99	33.09	27.19	62.21	60.72	44.60	**69.66**
25%	78.46	77.89	80.58	92.79	90.80	87.57	**94.29**	30.34	29.49	34.07	60.24	53.36	47.93	**65.79**
30%	68.20	78.68	87.97	87.88	78.02	88.06	**91.82**	20.29	31.90	46.05	44.48	30.02	48.33	**57.03**

Bold numbers denote the best results

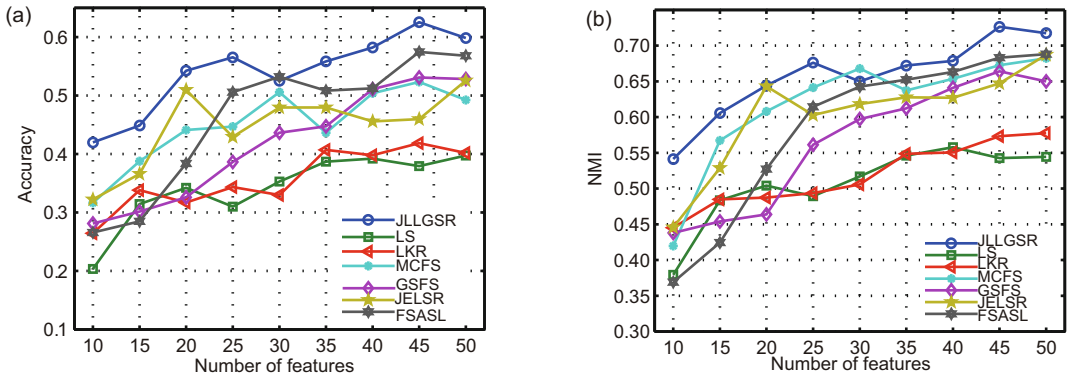

Fig. 5 Clustering results using features ranging from 10 to 50 in ISOLET4: (a) clustering accuracy; (b) normalized mutual information

Table 4 Clustering results for ISOLET4

d	Clustering accuracy (%)							Normalized mutual information (%)						
	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR
10	20.35	26.44	31.64	28.11	32.22	26.57	**41.98**	37.94	44.52	41.99	43.79	44.56	36.91	**54.13**
15	31.45	33.83	38.77	30.17	36.59	28.50	**44.87**	48.36	48.49	56.73	45.40	52.87	42.44	**60.56**
20	34.21	31.71	44.09	32.61	50.96	38.38	**54.24**	50.42	48.75	60.80	46.39	64.31	52.72	**64.42**
25	31.00	34.34	44.67	38.64	42.94	50.58	**56.55**	48.94	49.36	64.14	56.12	60.27	61.43	**67.61**
30	35.30	32.99	50.58	43.65	47.95	**53.21**	52.50	51.70	50.59	**66.80**	59.74	61.83	64.27	64.99
35	38.70	40.69	43.65	44.74	47.95	50.83	**55.84**	54.61	54.87	63.72	61.22	62.77	65.22	**67.21**
40	39.22	39.79	50.39	51.09	45.57	51.22	**58.22**	55.79	55.06	65.35	64.09	62.70	66.33	**67.86**
45	37.93	41.85	52.31	53.08	45.96	57.45	**62.52**	54.26	57.32	67.26	66.41	64.72	68.32	**72.63**
50	39.73	40.18	49.23	52.82	52.57	56.80	**59.82**	54.45	57.76	68.22	64.98	68.72	68.81	**71.72**
All	55.07	55.07	55.07	55.07	55.07	55.07	55.07	71.31	71.31	71.31	71.31	71.31	71.31	71.31

The parameter d denotes the number of selected features. Bold numbers denote the best results.

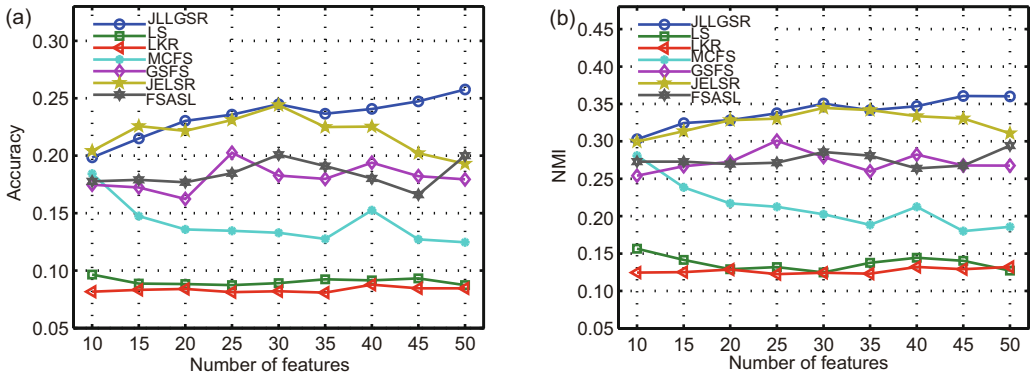

Fig. 6 Clustering results using features ranging from 10 to 50 in YaleB: (a) clustering accuracy; (b) normalized mutual information

features once again, demonstrating the advantage in reducing the impact of noise and redundancies.

The clustering results for the COIL100 dataset are displayed in Fig. 7 and Table 6. JLLGSR achieves the best performance again in terms of both clustering accuracy and normalized mutual information. However, the clustering results of JLLGSR cannot beat those with all the features. The reason is that at most 50 selected features do not have enough discriminating power and are not capable of dealing with 100 clusters.

The clustering results for the CIFAR10 dataset

Table 5 Clustering results for YaleB

d	Clustering accuracy (%)							Normalized mutual information (%)						
	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR
10	9.65	8.16	18.43	17.48	**20.42**	17.77	19.84	15.65	12.45	28.09	25.40	29.96	27.29	**30.27**
15	8.86	8.33	14.75	17.23	**22.58**	17.90	21.50	14.16	12.52	23.86	26.68	31.35	27.29	**32.43**
20	8.82	8.41	13.59	16.24	22.16	17.69	**23.03**	12.91	12.87	21.71	27.24	**32.83**	26.96	32.82
25	8.74	8.12	13.46	20.26	23.12	18.48	**23.57**	13.18	12.26	21.25	30.10	33.03	27.14	**33.75**
30	8.91	8.20	13.30	18.27	24.40	20.09	**24.48**	12.47	12.45	20.25	27.90	34.48	28.57	**35.04**
35	9.24	8.08	12.76	17.98	22.49	19.10	**23.65**	13.77	12.30	18.83	26.00	**34.17**	28.08	34.16
40	9.15	8.78	15.24	19.39	22.54	18.02	**24.07**	14.45	13.22	21.26	28.23	33.36	26.40	**34.68**
45	9.32	8.45	12.72	18.23	20.22	16.61	**24.73**	14.03	12.91	18.00	26.78	33.05	26.76	**36.06**
50	8.74	8.45	12.47	17.94	19.30	20.01	**25.77**	12.72	13.22	18.58	26.76	31.07	29.43	**36.03**
All	9.20	9.20	9.20	9.20	9.20	9.20	9.20	11.70	11.70	11.70	11.70	11.70	11.70	11.70

The parameter d denotes the number of selected features. Bold numbers denote the best results.

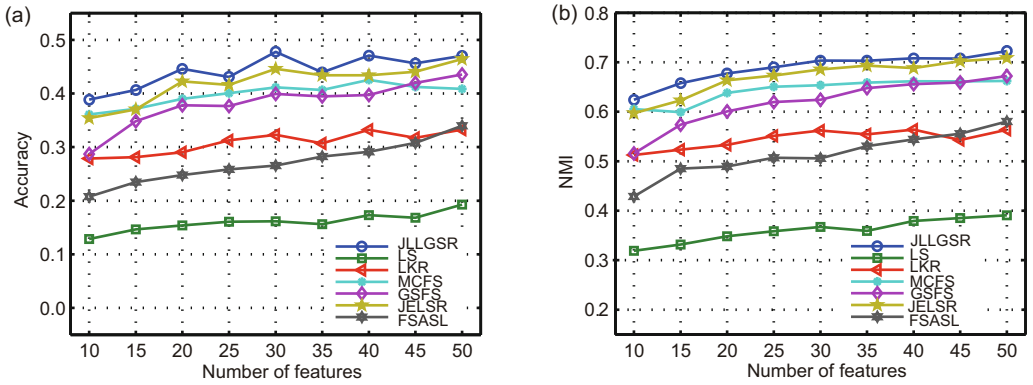

Fig. 7 Clustering results using features ranging from 10 to 50 in COIL100: (a) clustering accuracy; (b) normalized mutual information

Table 6 Clustering results for COIL100

d	Clustering accuracy (%)							Normalized mutual information (%)						
	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR
10	12.86	27.88	36.08	28.71	35.40	20.75	**38.85**	31.90	51.26	60.53	51.55	59.71	42.87	**62.41**
15	14.67	28.13	37.15	34.83	37.04	23.49	**40.67**	33.18	52.33	59.93	57.38	62.29	48.52	**65.76**
20	15.40	29.04	39.01	37.79	42.26	24.78	**44.60**	34.84	53.30	63.82	60.06	66.34	48.93	**67.75**
25	16.08	31.28	40.06	37.65	41.60	25.83	**43.08**	35.85	55.13	65.06	61.96	67.30	50.71	**68.98**
30	16.15	32.29	41.15	39.90	44.60	26.57	**47.76**	36.72	56.21	65.36	62.41	68.53	50.59	**70.34**
35	15.63	30.68	40.65	39.46	43.40	28.24	**43.93**	35.91	55.45	65.89	64.76	69.32	53.06	**70.31**
40	17.31	33.22	42.53	39.71	43.39	29.13	**47.07**	37.92	56.39	66.15	65.57	68.83	54.42	**70.79**
45	16.82	31.74	41.24	41.96	44.04	30.78	**45.63**	38.51	54.28	66.12	65.92	70.21	55.56	**70.74**
50	19.28	33.25	40.85	43.54	46.42	33.99	**47.03**	39.06	56.35	66.21	67.21	70.87	58.03	**72.27**
All	48.63	48.63	48.63	48.63	48.63	48.63	48.63	75.88	75.88	75.88	75.88	75.88	75.88	75.88

The parameter d denotes the number of selected features. Bold numbers denote the best results.

are shown in Fig. 8 and Table 7. FSASL needs more than 256 GB memory to run, which goes beyond our server's capacity, so its results are not provided. Compared with other methods, JLLGSR achieves the best performance in terms of clustering accuracy when selecting at least 30 features. In terms of normalized mutual information, JLLGSR beats all the other methods when selecting at least 25 features. Therefore, we can say that JLLGSR outperforms others when selecting a large enough feature subset.

All of the experimental results above show that JLLGSR performs better than the state-of-the-art feature selection algorithms. To further understand the behavior of JLLGSR, we offer a comparison of

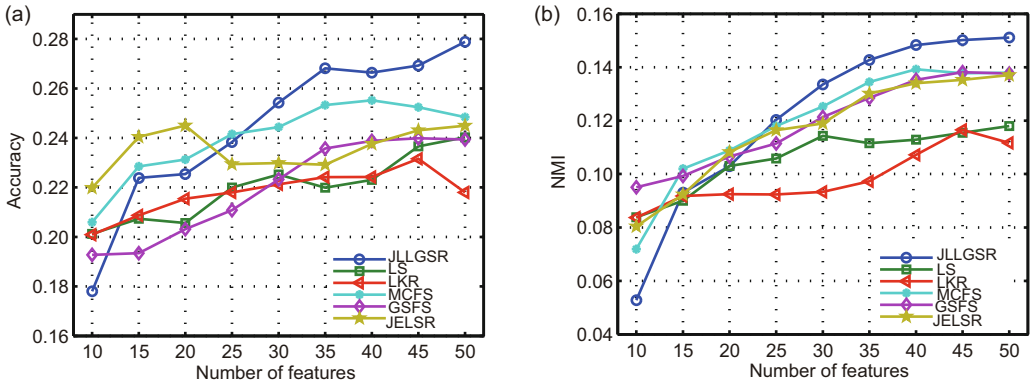

Fig. 8 Clustering results using features ranging from 10 to 50 in CIFAR10: (a) clustering accuracy; (b) normalized mutual information

Table 7 Clustering results for CIFAR10

d	Clustering accuracy (%)							Normalized mutual information (%)						
	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR
10	20.12	20.09	20.59	19.27	**21.98**	–	17.80	8.38	8.37	7.19	**9.50**	8.04	–	5.28
15	20.74	20.87	22.85	19.34	**24.04**	–	22.39	9.00	9.18	**10.20**	9.93	9.24	–	9.30
20	20.56	21.55	23.13	20.32	**24.50**	–	22.54	10.30	9.25	**10.89**	10.66	10.82	–	10.29
25	22.00	21.80	**24.16**	21.08	22.94	–	23.84	10.58	9.24	11.80	11.14	11.64	–	**12.03**
30	22.52	22.12	24.44	22.32	22.99	–	**25.43**	11.43	9.33	12.53	12.13	11.89	–	**13.36**
35	21.99	22.42	25.33	23.57	22.92	–	**26.82**	11.15	9.73	13.45	12.85	13.01	–	**14.27**
40	22.30	22.42	25.52	23.88	23.76	–	**26.64**	11.29	10.72	13.92	13.53	13.40	–	**14.83**
45	23.66	23.16	25.24	23.99	24.31	–	**26.93**	11.55	11.65	13.78	13.81	13.53	–	**15.02**
50	24.03	21.80	24.84	23.94	24.50	–	**27.89**	11.80	11.17	13.80	13.76	13.70	–	**15.11**
All	28.20	28.20	28.20	28.20	28.20	28.20	28.20	16.81	16.81	16.81	16.81	16.81	16.81	16.81

The parameter d denotes the number of selected features. Bold numbers denote the best results.
The results for FSASL are not provided because our server does not have enough memory (more than 256 GB).

JLLGSR, JELSR, and GSFS-llc. JLLGSR applies local learning based clustering for data distribution analysis, while JELSR uses locally linear embedding to capture the data manifold structure, and they both use group sparse regression to analyze the importance of each feature. Locally linear embedding assumes that one sample point can be approximated by its neighbors and its low dimensional embedding shares the same local linear approximation weights as the original data. Its graph Laplacian matrix is $\boldsymbol{L}=(\boldsymbol{I}-\boldsymbol{S})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{S})$. The local linear approximation weight $\boldsymbol{S}$ is formed by $\boldsymbol{s}_i=\mathbf{1}^{\mathrm{T}}\boldsymbol{C}^{-1}$, where $\boldsymbol{C}$ is the local covariance matrix of $\boldsymbol{x}_i$'s neighbor. Local learning based clustering uses $\boldsymbol{x}_i$'s neighbors to train a linear regression model and obtain its low dimensional embedding, and its graph Laplacian matrix $\boldsymbol{T}=(\boldsymbol{I}-\boldsymbol{A})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{A})$, where $\boldsymbol{A}$ is calculated by $\boldsymbol{a}_i=\boldsymbol{k}_i^{\mathrm{T}}(\boldsymbol{K}_i+\lambda\boldsymbol{I})^{-1}$. The Laplacian matrix of JLLGSR, where the kernel method is used, is much more complicated and may contain more information than that used in JELSR. Moreover, local learning based clustering is known to have a better clustering performance than locally linear embedding. JLLGSR and GSFS-llc both use local learning based clustering and group sparse regression to select the feature subset. The difference is that JLLGSR jointly solves these two problems while GSFS-llc solves local learning based clustering first and then group sparse regression based on the results of LLC. The joint optimization process enables the candidate feature subset to improve the clustering results, so JLLGSR performs better than GSFS-llc.

4.3.3 Clustering results for real-world datasets

Web pages from one web site usually share similar templates and the templates can be revealed by the structure of the HTML tags. Thus, in a web accessibility evaluation scenario, web pages are simplified to HTML tags to automatically carry out some evaluation processes. Fig. 9 and Table 8 show the clustering results for the WPAE dataset. JLLGSR outperforms the other methods in terms of clustering

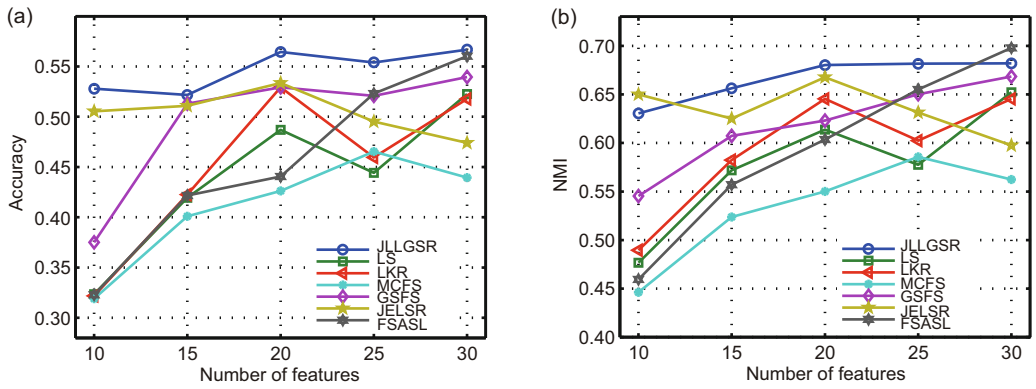

Fig. 9 Clustering results using features ranging from 10 to 30 in WPAE: (a) clustering accuracy; (b) normalized mutual information

Table 8 Clustering results for WPAE

d	Clustering accuracy (%)							Normalized mutual information (%)						
	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR	LS	LKR	MCFS	GSFS-llc	JELSR	FSASL	JLLGSR
10	32.35	32.19	31.91	37.51	50.53	32.28	**52.79**	47.67	48.96	44.62	54.54	**65.01**	45.95	63.06
15	41.93	42.26	40.09	51.30	51.09	42.16	**52.16**	57.18	58.24	52.37	60.73	62.51	55.70	**65.62**
20	48.70	52.93	42.63	52.93	53.35	44.05	**56.44**	61.38	64.55	55.02	62.31	66.77	60.36	**68.02**
25	44.42	45.95	46.53	52.07	49.51	52.30	**55.40**	57.76	60.24	58.57	65.02	63.15	65.50	**68.16**
30	52.26	51.77	43.95	53.95	47.40	56.02	**56.67**	65.22	64.55	56.25	66.84	59.75	**69.79**	68.20
All	44.79	44.79	44.79	44.79	44.79	44.79	44.79	57.06	57.06	57.06	57.06	57.06	57.06	57.06

The parameter d denotes the number of selected features. Bold numbers denote the best results.

accuracy and achieves relatively stable results for normalized mutual information. The clustering performance for the selected feature subsets is higher than that for all the features, which means the web page template structure is more related to the selected HTML tags. Therefore, these selected tags will be more useful in the following web accessibility evaluation process.

4.3.4 Parameter selection

JLLGSR has a total of four parameters to tune: the number of neighbors k, the number of eigenvectors used u, and the regularization parameters γ and δ. Under the manifold assumption, the similarities between samples can be preserved only within a small neighborhood in the original data space. Thus, the parameter k should be set to a small enough number. Based on our experience, we use five neighbors to calculate the kernel matrix. The parameter u is used to determine the dimension of matrix $\boldsymbol{Y}$, and $\boldsymbol{Y}$ represents the clustering results. According to local learning based clustering, the parameter u should be set to equal to the number of clusters. Therefore, we set the parameter u as the number of classes of each dataset.

In previous experiments, the regularization parameters γ and δ are determined by a grid search. Now, we conduct a series of experiments to investigate the influences of these parameters. Fig. 10 shows the clustering results using the top 30 features with the regularization parameters γ and δ ranging from 10^{-3} to 10^{3} in ISOLET4, YaleB, and COIL100. The best combination for each dataset is $\gamma = 0.1$, $\delta = 0.01$ in ISOLET4, $\gamma = 10$, $\delta = 0.001$ in YaleB, and $\gamma = 0.001$, $\delta = 1000$ in COIL100. As seen from these results, a relatively small γ is a good choice to achieve a good result; for δ, however, the use of a grid search may be the best way to tune the parameter.

5 Conclusions

In this study, we have proposed a novel unsupervised feature selection method called JLLGSR, which combines local learning and group sparse regression in a single model. JLLGSR obtains a clustering structure via local learning and the group sparse structure using $l_{2,1}$-norm regularized

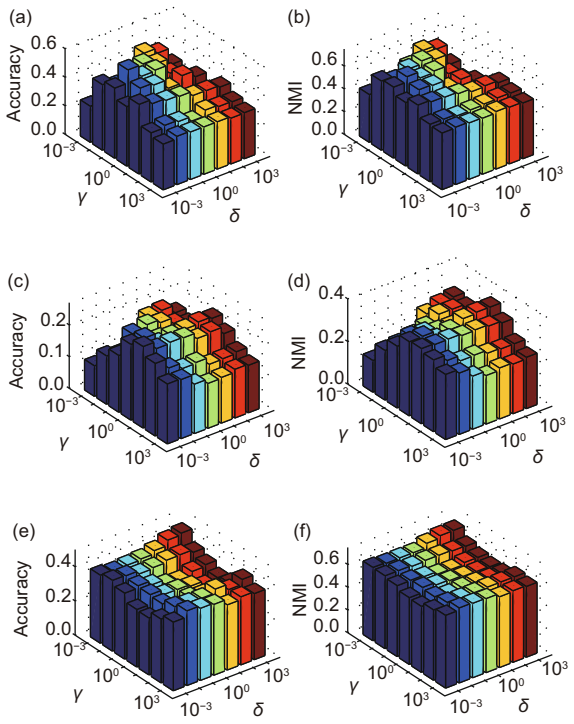

Fig. 10 Clustering accuracy and normalized mutual information (NMI) with the regularization parameters γ and δ ranging from 10^{-3} to 10^{3} from ISOLET4, YaleB, and COIL100: (a) accuracy in ISOLET4; (b) NMI in ISOLET4; (c) accuracy in YaleB; (d) NMI in YaleB; (e) accuracy in COIL100; (f) NMI in COIL100

regression. By jointly optimizing these two objectives, the resulting feature subset not only explicitly respects the manifold structure in the data space, but also exhibits the noise-resistant characteristic of a group sparsity structure. Extensive experiments show that JLLGSR outperforms state-of-the-art feature selection algorithms on various datasets and is particularly robust in the presence of noise.

In the future, we plan to further investigate the property of group sparsity and accelerate the learning process, as well as integrate group sparsity with other feature selection or feature learning algorithms to improve the learning performance.

Acknowledgements

The experiment is supported by Cheng-wei YAO in the Experiment Center of the College of Computer Science and Technology, Zhejiang University.

References

Belkin M, Niyogi P, 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. 14th Int Conf on Neural Information Processing Systems: Natural and Synthetic, p.585-591.

Bellman RE, 1961. Adaptive Control Processes: a Guided Tour. Princeton University Press, Princeton, NJ.

Cai D, Zhang C, He X, 2010. Unsupervised feature selection for multi-cluster data. 16th Int Conf on Knowledge Discovery and Data Mining, p.333-342. https://doi.org/10.1145/1835804.1835848

Chang XJ, Nie FP, Yang Y, et al., 2016. Convex sparse PCA for unsupervised feature learning. *ACM Trans Knowl Dis Data*, 11(1):3. https://doi.org/10.1145/2910585

Cheung Y, Zeng H, 2009. Local kernel regression score for selecting features of high-dimensional data. *IEEE Trans Knowl Data Eng*, 21(12):1798-1802. https://doi.org/10.1109/TKDE.2009.23

Doquire G, Verleysen M, 2013. Mutual information-based feature selection for multilabel classification. *Neurocomputing*, 122:148-155. https://doi.org/10.1016/j.neucom.2013.06.035

Du L, Shen YD, 2015. Unsupervised feature selection with adaptive structure learning. 21st Int Conf on Knowledge Discovery and Data Mining, p.209-218. https://doi.org/10.1145/2783258.2783345

Fanty M, Cole R, 1990. Spoken letter recognition. Conf on Advances in Neural Information Processing Systems, p.220-226. https://doi.org/10.3115/116580.116725

Georghiades AS, Belhumeur PN, Kriegman DJ, 2001. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Trans Patt Anal Mach Intell*, 23(6):643-660. https://doi.org/10.1109/34.927464

Guyon I, Elisseeff A, 2003. An introduction to variable and feature selection. *J Mach Learn Res*, 3:1157-1182. https://doi.org/10.1162/153244303322753616

Guyon I, Weston J, Barnhill S, et al., 2002. Gene selection for cancer classification using support vector machines. *Mach Learn*, 46(1-3):389-422. https://doi.org/10.1023/A:1012487302797

Han YH, Wu F, Tian Q, et al., 2012. Image annotation by input-output structural grouping sparsity. *IEEE Trans Image Proc*, 21(6):3066-3079. https://doi.org/10.1109/TIP.2012.2183880

Han YH, Yang Y, Yan Y, et al., 2015. Semisupervised feature selection via spline regression for video semantic recognition. *IEEE Trans Neur Netw Learn Syst*, 26(2):252-264. https://doi.org/10.1109/TNNLS.2014.2314123

He X, Niyogi P, 2004. Locality preserving projections. Conf on Advances in Neural Information Processing Systems, p.153-160.

He X, Cai D, Niyogi P, 2005. Laplacian score for feature selection. Conf on Advances in Neural Information Processing Systems, p.507-514.

Hou CP, Nie FP, Li XL, et al., 2014. Joint embedding learning and sparse regression: a framework for unsupervised feature selection. *IEEE Trans Cybern*, 44(6):793-804. https://doi.org/10.1109/TCYB.2013.2272642

Hull JJ, 1994. A database for handwritten text recognition research. *IEEE Trans Patt Anal Mach Intell*, 16(5):550-554. https://doi.org/10.1109/34.291440

Jiang Y, Ren JT, 2011. Eigenvalue sensitive feature selection. 28th Int Conf on Machine Learning, p.89-96.

Jolliffe IT, 2002. Principal Component Analysis (2nd Ed.). Springer, New York.

Krizhevsky A, 2009. Learning Multiple Layers of Features from Tiny Images. Science Department, University of Toronto, Tech, Toronto.

Kuhn HW, 1955. The Hungarian method for the assignment problem. *Nav Res Log Q*, 2(1-2):83-97. https://doi.org/10.1002/nav.3800020109

Lee KC, Ho J, Kriegman DJ, 2005. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans Patt Anal Mach Intell*, 27(5):684-698. https://doi.org/10.1109/TPAMI.2005.92

Luo MN, Nie FP, Chang XJ, et al., 2018. Adaptive unsupervised feature selection with structure regularization. *IEEE Trans Neur Netw Learn Syst*, 29(4):944-956. https://doi.org/10.1109/TNNLS.2017.2650978

Munkres J, 1957. Algorithms for the assignment and transportation problems. *J Soc Ind Appl Math*, 5(1):32-38. https://doi.org/10.1137/0105003

Nie FP, Xiang SM, Jia YQ, et al., 2008. Trace ratio criterion for feature selection. 23rd Int Conf on Artificial Intelligence, p.671-676.

Nie FP, Xiang SM, Song YQ, et al., 2009. Orthogonal locality minimizing globality maximizing projections for feature extraction. *Opt Eng*, 48(1):017202. https://doi.org/10.1117/1.3067869

Nie FP, Huang H, Cai X, et al., 2010a. Efficient and robust feature selection via joint $l_{2,1}$-norms minimization. 23rd Int Conf on Neural Information Processing Systems, p.1813-1821.

Nie FP, Xu D, Tsang IWH, et al., 2010b. Flexible manifold embedding: a framework for semi-supervised and unsupervised dimension reduction. *IEEE Trans Image Proc*, 19(7):1921-1932. https://doi.org/10.1109/TIP.2010.2044958

Nie FP, Zeng ZN, Tsang IW, et al., 2011. Spectral embedded clustering: a framework for in-sample and out-of-sample spectral clustering. *IEEE Trans Neur Netw*, 22(11):1796-1808. https://doi.org/10.1109/TNN.2011.2162000

Nie FP, Wang XQ, Jordan MI, et al., 2016a. The constrained Laplacian rank algorithm for graph-based clustering. 30th AAAI Conf on Artificial Intelligence, p.1969-1976.

Nie FP, Zhu W, Li XI, 2016b. Unsupervised feature selection with structured graph optimization. 30th AAAI Conf on Artificial Intelligence, p.1302-1308.

Peng HC, Long FH, Ding C, 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Patt Anal Mach Intell*, 27(8):1226-1238. https://doi.org/10.1109/TPAMI.2005.159

Roweis ST, Saul LK, 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323-2326. https://doi.org/10.1126/science.290.5500.2323

Sun YJ, Todorovic S, Goodison S, 2010. Local-learning-based feature selection for high-dimensional data analysis. *IEEE Trans Patt Anal Mach Intell*, 32(9):1610-1626. https://doi.org/10.1109/TPAMI.2009.190

Tan MK, Wang L, Tsang IW, 2010. Learning sparse SVM for feature selection on very high dimensional datasets. 27th Int Conf on Machine Learning, p.1047-1054.

Tenenbaum JB, de Silva V, Langford JC, 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319-2323. https://doi.org/10.1126/science.290.5500.2319

Tibshirani R, 1996. Regression shrinkage and selection via the Lasso. *J R Stat Soc B*, 58(1):267-288.

Verleysen M, 2003. Learning high-dimensional data. In: Ablameyko S, Goras L, Gori M (Eds.), Limitations and Future Trends in Neural Computation. IOS Press, Amsterdam, p.141-162.

Wang D, Nie FP, Huang H, 2014. Unsupervised feature selection via unified trace ratio formulation and K-means clustering (TRACK). European Conf on Machine Learning and Knowledge Discovery in Databases, p.306-321. https://doi.org/10.1007/978-3-662-44845-8_20

Wu Y, Wang C, Bu JJ, et al., 2016. Group sparse feature selection on local learning based clustering. *Neurocomputing*, 171:1118-1130. https://doi.org/10.1016/j.neucom.2015.07.045

Yang Y, Shen HT, Ma ZG, et al., 2011. $l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning. 22nd Int Joint Conf on Artificial Intelligence, p.1589-1594. https://doi.org/10.5591/978-1-57735-516-8/ijcai11-267

Zeng H, Cheung YM, 2009. Feature selection for local learning based clustering. 13th Pacific-Asia Conf on Advances in Knowledge Discovery and Data Mining, p.414-425. https://doi.org/10.1007/978-3-642-01307-2_38

Zeng H, Cheung YM, 2011. Feature selection and kernel learning for local learning-based clustering. *IEEE Trans Patt Anal Mach Intell*, 33(8):1532-1547. https://doi.org/10.1109/TPAMI.2010.215

Zhao Z, Liu H, 2007. Spectral feature selection for supervised and unsupervised learning. 24th Int Conf on Machine Learning, p.1151-1157. https://doi.org/10.1145/1273496.1273641

Zou H, Hastie T, 2005. Regularization and variable selection via the elastic net. *J R Stat Soc Ser B*, 67(2):301-320. https://doi.org/10.1111/j.1467-9868.2005.00503.x