



Unsupervised feature selection based on kernel fisher discriminant analysis and regression learning

Ronghua Shang¹ · Yang Meng¹ · Chiyang Liu¹ · Licheng Jiao¹ · Amir M. Ghalamzan Esfahani² · Rustam Stolkin²

Received: 25 March 2016 / Accepted: 21 September 2018 / Published online: 5 October 2018
© The Author(s) 2018

Abstract

In this paper, we propose a new feature selection method called kernel fisher discriminant analysis and regression learning based algorithm for unsupervised feature selection. The existing feature selection methods are based on either manifold learning or discriminative techniques, each of which has some shortcomings. Although some studies show the advantages of two-steps method benefiting from both manifold learning and discriminative techniques, a joint formulation has been shown to be more efficient. To do so, we construct a global discriminant objective term of a clustering framework based on the kernel method. We add another term of regression learning into the objective function, which can impose the optimization to select a low-dimensional representation of the original dataset. We use $L_{2,1}$ -norm of the features to impose a sparse structure upon features, which can result in more discriminative features. We propose an algorithm to solve the optimization problem introduced in this paper. We further discuss convergence, parameter sensitivity, computational complexity, as well as the clustering and classification accuracy of the proposed algorithm. In order to demonstrate the effectiveness of the proposed algorithm, we perform a set of experiments with different available datasets. The results obtained by the proposed algorithm are compared against the state-of-the-art algorithms. These results show that our method outperforms the existing state-of-the-art methods in many cases on different datasets, but the improved performance comes with the cost of increased time complexity.

Keywords Kernel fisher discriminant analysis · Manifold learning · Regression learning · Sparse constraint · Feature selection

Editor: Tapio Elomaa.

✉ Ronghua Shang
rhshang@mail.xidian.edu.cn

¹ Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China

² Extreme Robotics Lab, University of Birmingham, Birmingham, UK

1 Introduction

Dimensionality reduction is a widely used preprocessing of high dimensional datasets (Yan and Xu 2007). It aims to project a high dimensional dataset into a subspace (Gu and Sheng 2016), which can wipe away noise and/or redundant and irrelevant features to obtain the new representation and keep the most important variability of the dataset (Meng et al. 2018). In addition, this projection can reduce the total computation cost, because it uses lower dimensional data than the original dataset. Therefore, the efficiency of dimensionality reduction is very important (Belkin and Niyogi 2003).

Feature selection is one of the most important dimensionality reduction methods, which can remove the redundant features in the original data and rapidly deal with massive high dimensional data (Meng et al. 2018). Feature selection (a) allows us to recognize representative features in the original dataset. Hence, the further process and computation become computationally easier, e.g. classification (Gu and Sheng 2017) in the subspace; (b) results in a subspace with less influence of noise. Thus, the further computation is robust to noise; (c) inherently resolves the problem of over fitting, which is common in many contexts (Stolkin et al. 2008), e.g. model fitting. Feature selection is widely used in text-mining (Shang et al. 2017), bio-medical treatment (Ding and Peng 2005), voice recognition (Abdulla and Kasabov 2003), commodity recommendation and security monitoring (Tian and Chen 2017). In the recent years, many feature selection methods have been developed (Gu and Sheng 2017; Stolkin et al. 2008; Mitra et al. 2002; Stolkin et al. 2007; Mao and Tsang 2013; Gu et al. 2015). Depending on the available data, e.g. labeled dataset, feature selection is divided into supervised (Sikonja and Kononenko 2003; Zhao and Liu 2007), semi-supervised (Xu et al. 2010; Shi and Ruan 2015) and unsupervised (Mitra et al. 2002; Li et al. 2014; Cai et al. 2010; Yang et al. 2011).

In supervised feature selection, a subset of original features is selected by using the relationship between labels and features. Although supervised feature selection has high accuracy, it entails high computation cost (Li et al. 2010). In some cases, there is only a fraction of label information. Hence, semi-supervised feature selection methods should make full use of these label information. By adding the label information as additional constraint on unsupervised algorithms, we can enhance the performance of the method. Thus, semi-supervised algorithms can be regarded as a special type of unsupervised feature selection method (Xu et al. 2010; Cheng et al. 2011). Apparently, unsupervised feature selection methods have been proposed to handle the unlabeled datasets. Hence, some intrinsic properties of datasets are employed for feature selection, e.g. scatter separability. Unsupervised feature selection is more difficult and more computation cost than supervised and semi-supervised feature selection due to the lack of prior information. In most of real world problem, we need to deal with unlabeled or partially labeled datasets. This may indicate the future researches must mostly solve unsupervised and semi-supervised feature selection problems. Recently, there have been already many unsupervised algorithms (Stolkin et al. 2008; Shang et al. 2017; Mitra et al. 2002; Stolkin et al. 2007; Li et al. 2014; Cai et al. 2010; Constantinopoulos et al. 2006).

Comparing to the early feature selection methods, most of the latest feature selection methods are unsupervised and many other techniques are used simultaneously to enhance their performance. Here we prefer to emphasis on the application of graph spectral theory in feature selection, which has been proved to be a strong implement for dimensionality reduction (Yan and Xu 2007; Li et al. 2014; Chen et al. 2015; Yang et al. 2010; Liu et al. 2014; Doquire and Verleysen 2013; Wei et al. 2012). A large number of other methods have been also widely used including PCA (Smith 2002), Linear Discriminant Analysis (LDA) (McLachlan

2004), Locally Linear Embedding (LLE) (Roweis and Saul 2000), Isomap (Tenenbaum et al. 2000), Locality Projection Preserving (LPP) (He and Niyogi 2004) and Laplacian Eigenmaps (LE) Meng et al. (2018), etc. Some of the graph spectral methods have better performance than traditional feature selection methods including Spectral Feature Selection (SPEC) (Zhao and Liu 2007), Laplacian Score (LapScor) (He et al. 2005), Multi-cluster Feature Selection (MCFS) (Cai et al. 2010) and Minimum Redundancy Spectral Feature Selection (MRSF) (Zhao et al. 2010). It has been found that a joint framework in some algorithms can lead to better results than the “two-step” strategy in SPEC, MCFS and MRSF. For example, JELSR unifies embedding learning and sparse regress (Hou et al. 2014), LSPE solves embedding learning and feature selection simultaneously (Fang et al. 2014) and DFSC combines self-representation with manifold learning and feature selection (Shang et al. 2016). These studies demonstrated their methods outperform other multi-stage methods.

PCA and LDA have been proposed for dimensionality reduction where PCA is able to handle linear as well as nonlinear data whereas LDA can be only applied to linear data. Sebastian Mika et al. (1999) extend LDA based on kernel methods to nonlinear fields using Kernel Fisher Discriminant Analysis (KFDA). It is proved that KFDA performs better than PCA and KPCA. Besides kernel methods, Local Discriminant Models and Global Integration (LDMGI) deals with nonlinear data by applying LDA in a small neighbor of a nominal point resembling a linear subspace (Yang et al. 2010).

The abovementioned methods show a good performance, but they only use either manifold structure or discriminative technique alone (Ma et al. 2016). Inspired by the ideas above mentioned and based on graph spectral theory, we combine the global discriminative information with manifold information and propose a joint framework for feature selection. Therefore, we propose a novel joint framework of unsupervised feature selection based on kernel fisher discriminant analysis and regression learning (KFDRL) to exploit the intrinsic characters of data and select representative features. It kernelizes LDA to be a global discriminant first, adds regression learning and $L_{2,1}$ -norm regularization to construct a joint framework for feature selection. We also present update rules to compute the solution and further study the convergence and computational complexity of the proposed algorithm. The contribution of this paper is:

- (1) We propose a framework of unsupervised feature selection combining global discriminant analysis with graph spectral theory and regression learning. Therefore, our algorithm benefits all the advantages of global discriminant analysis with graph spectral theory and regression learning.
- (2) We use discriminative information to make our method superior to JELSR, which can result in a better separation of data points belonging to different classes. Our method can result in a better performance in both clustering and classification.
- (3) A mathematical model of proposed method is presented and a simple optimization strategy is applied to solve the model efficiently. We demonstrate the effectiveness of our method by a series of experiments with several datasets. We further validate our results by comparing them with the results of other feature selection algorithms.

The rest of this paper is set as follows: in Sect. 2, we introduce the formulation of the related works. In Sect. 3, the problem formulation, the algorithm and optimization process are all explained in detail. Convergence and computational complex of the algorithm are studied in Sect. 4. In Sect. 5, we present experiments and the results, which demonstrate the effectiveness of the proposed method. Final section contains conclusion and future works.

2 The related works

In this section, we will introduce some useful notations and the following two relevant works will be briefly presented (1) feature selection, MCFS, MRSF, JELSR and CGSSL (Li et al. 2014; Cai et al. 2010; Zhao et al. 2010; Hou et al. 2014) and (2) clustering algorithms, LDMGI (Yang et al. 2010), kernel method and KLDA.

2.1 Notations

We use bold capital and bold lowercase letters for matrices and vectors, for example $A \in R^{d \times n}$ is a matrix, a_i is the i th vector of A and a_{ij} is the j th element of a_i . Let's $tr(A)$ denote the trace of a square matrix A and denotes $L_{r,p}$ -norm defined as follows:

$$\|A\|_{r,p} = \left(\sum_{i=1}^d \left(\sum_{j=1}^n |a_{ij}|^r \right)^{p/r} \right)^{1/p} \tag{1}$$

The dataset represented by matrix $X = [x_1, x_2, \dots, x_n] \in R^{d \times n}$ has n sample vectors where each vector $x_i \in R^d$ has d features. We assume samples belong to c different classes. We are interested in computing a cluster assignment matrix $Y = [y_1, y_2, \dots, y_n]^T \in \{0,1\}^{n \times c}$, where $y_i \in \{0,1\}^{c \times 1}$ is a cluster assignment vector whose j th element y_{ij} is one when x_i belongs to j th cluster and zero otherwise. Furthermore, we define the scaled cluster assignment matrix $H \in R^{n \times c}$, where H satisfies $H^T H = I_c$ and $I_c \in R^{c \times c}$ is an identity matrix, as follows:

$$H = [h_1, h_2, \dots, h_n] = Y(Y^T Y)^{-\frac{1}{2}} = \left[\frac{y_1}{\|y_1\|}, \frac{y_2}{\|y_2\|}, \dots, \frac{y_n}{\|y_n\|} \right] \tag{2}$$

2.2 Feature selection

2.2.1 MCFS and MRSF

In MCFS (Cai et al. 2010) and MRSF (Zhao et al. 2010), the first step is to compute an m -D embedding representation of x_i by mapping d -D x_i into an embedding space with lower dimensions R^m where $m < d$. This mapping is represented by an embedding matrix $P = [p_1, p_2, \dots, p_n] \in R^{m \times n}$. The embedding techniques LE and LLE proposed in Meng et al. (2018), Roweis and Saul (2000) are used and regression learning MCFS and MRSF can be defined as follows:

$$\arg \min_{P P^T = I_{n \times n}} tr(PLP^T)$$

$$\arg \min_W \|W^T X - P\|_2^2 + \alpha \|W\|_1 \tag{3}$$

$$\arg \min_{P P^T = I_{n \times n}} tr(PLP^T)$$

$$\arg \min_W \|W^T X - P\|_2^2 + \alpha \|W\|_{2,1} \tag{4}$$

It is clear that the two methods are different in the regularization term where MCFS has L_1 -norm and MRSF has $L_{2,1}$ -norm. Different norms definition is used to constrain the sparse structure of the data as regression coefficient is used to rank the features. The optimization

proposed in (3) is very similar to Lasso (Gu and Sheng 2016) whereas the problem formulation in (4) uses L_2 -norm to rank each row of W . Although these two-step algorithms are efficient, they are still not as good as single-step algorithms, joint framework algorithms, which will be presented as follows.

2.2.2 JELSR and CGSSL

JELSR is an unsupervised algorithm, which combines embedding learning with sparse regression. It preserves local manifold structure as two variables simultaneously to get a better performance. JELSR is formulated in Eq. (5).

$$L(W, P) = \arg \min_{W, P} \text{tr}(PLP^T) + \beta \left(\|W^T X - P\|_2^2 + \alpha \|W\|_{2,1} \right) \quad (5)$$

where W denotes the importance of each feature and P is embedding matrix. The objective formulation for CGSSL is similar to JELSR. However, CGSSL impose a low-dimensional constraint upon embedding matrix with pseudo class labels. Moreover, as a sparse structure learning technique, semantic components are used in CGSSL to match pseudo class labels with truth class labels. Hence, CGSSL can be formulated as follows:

$$\begin{aligned} \min_{S, W, Q, M} \mathcal{E} &= \text{tr}(F^T L F) + \alpha \|F - X^T W\|_F^2 + \beta \|W\|_{2,1} + \gamma \|W - QS\|_F^2 \\ \text{s.t. } F^T F &= I_c, F \geq 0; \quad Q^T Q = I_r \end{aligned} \quad (6)$$

where S is a matrix of weights and Q is a transformation matrix. They are used to save original features as well as embedded features. F is a scaled cluster assignment matrix to predict labels. Both JELSR and CGSSL have great performance results due to the local manifold as well as the discriminative information.

2.2.3 MDRL and RMDRL

MDRL and RMDRL (Lu et al. 2015) are proposed for image classification with a linear regression framework. A within-class graph and a between-class graph are introduced in MDRL to get an optimal subspace. Furthermore, a nuclear norm is used to learn a robust projection matrix by a developed MDRL, i.e., RMDRL. Manifold information and discriminant information are both used in a regression learning framework by MDRL and RMDRL like the proposed algorithm KFDRL. These two algorithms are respectively formulated as follows:

$$\begin{aligned} \min_{W, M} \|XW - Y - B \odot M\|_F^2 + \alpha \text{Tr}(W^T X^T (L_w - L_b) X W) + \beta \|W\|_{2,1} + \gamma \|W\|_1 \\ \text{s.t. } M \geq 0 \end{aligned} \quad (7)$$

$$\begin{aligned} \min_{W, M} \|XW - Y - B \odot M\|_F^2 + \alpha \text{Tr}(W^T X^T (L_w - L_b) X W) + \beta \|W\|_* \\ \text{s.t. } M \geq 0 \end{aligned} \quad (8)$$

These two algorithms first construct X , Y and B by optimal methods and then compute the W as projection matrix. In the end, they use L_1 -norm and nuclear norm to select features. Note that the X is a matrix of training samples and Y is the corresponding label matrix. In other words, MDRL and RMDRL are supervised algorithms, which have different application cases with KFDRL.

2.3 Spectral clustering

Over the last few years, many studies on the graph spectral theory for clustering analysis have been published. For example, Luxburg (2007) has presented a significant spectral clustering method with graph theory. It has been shown that the spectral clustering has advantages over traditional algorithms of non-convex distribution for partitioning a complex data structure. It makes full use of geometric information in the original datasets. Hence, pseudo class labels represent a more accurate intrinsic structure information in the original datasets. In Yang et al. (2010), LDMGI computes a Laplacian matrix by using both discriminative information and manifold learning, which has a good performance in clustering images. Local discriminant model in a sufficient small local manifold area is used on spectral clustering, whose objective function can be shown as follows:

$$\arg \min_{G^{(i)}} \text{tr} \left[G^{(i)T} L_i^{S_i} G^{(i)} \right] \tag{9}$$

where $L_i = H_k (\tilde{X}_i^T \tilde{X}_i + \lambda I)^{-1} H_k$ is a local Laplacian matrix and $H_i = [h_{i1}, h_{i2}, \dots, h_{ik-1}]^T \in R^{k \times c}$ is a cluster assignment matrix. After that, a global integration method is imposed to get the global Laplacian matrix in nonlinear space:

$$L^S = \sum_{i=1}^n S_i L_i^{S_i} S_i^T = [S_1, S_2, \dots, S_n] \begin{bmatrix} L_1^{S_1} & & \\ & \dots & \\ & & L_n^{S_n} \end{bmatrix} [S_1, S_2, \dots, S_n]^T \tag{10}$$

Moreover, the global discriminative model can be defined as:

$$\begin{aligned} & \min_G \text{tr} (G^T L^S G) \\ & \text{s.t. } G^T G = I, \quad G \geq 0 \end{aligned} \tag{11}$$

where L^S contains both manifold information and discriminative information. Most of the corresponding methods employ the local-idea to handle nonlinear problems (Cai et al. 2010; Yang et al. 2010; Roweis and Saul 2000; He and Niyogi 2004; Zhao et al. 2010; Hou et al. 2014), which use (1) linear methods in each local-and-small area and (2) global integration. It has been shown to be very pragmatic in many different contexts. However, we may have some bad results due to weak robustness, low convergence rate and more complex formalism. So it is an interesting research to find a way to simplify the effectiveness and improve the robust of these algorithms.

2.4 Kernel method

The kernel method is very efficient and powerful. A group of points in a low-dimensional space can be mapped into a space with higher dimensions and become linearly separable using a proper kernel mapping. The mapping is defined by kernel function $K(x, y) = \langle \phi(x), \phi(y) \rangle$, where x and y are points in low-dimensional space, $\phi(\bullet)$ denotes the points in the higher dimensional space and \langle, \rangle denotes the inner product. According to *mercer* theorem, we can transform a pair of points in low dimensional space satisfying a specific function requirement into higher dimensional space. This transformation function can be considered as a bridge between the higher and lower dimensional spaces. Kernel fisher discriminant analysis (KFDA) is one of the applications of the kernel method, which can obtain better results than LDA and PCA in the expense of more complex optimization and higher computational cost.

3 Feature selection based on kernel discriminant and sparse regression

Here, we introduce our proposed method with three terms (1) a global kernel discriminant term based on nonnegative spectral clustering (2) a regression learning term and (3) a sparse constraint regularization term. A kernel linear discriminant model is integrated into a spectral clustering model to preserve manifold information as well as discriminative information. Therefore, our method can be applied to linear and nonlinear dataset. A regression method can fit the coefficient matrix to the scaled cluster assignment matrix. Finally, a sparse regularization is performed for feature selection.

3.1 Global kernel discriminant model based on nonnegative spectral clustering

3.1.1 Nonnegative spectral clustering

In non-negative spectral clustering, Laplacian matrix L is computed by constructing a nearest neighbor graph S of data points. The spectral embedding matrix Y can be computed by Eq. (12) to retain the manifold information:

$$\min_Y \text{tr}(Y^T LY) \quad (12)$$

In this paper, we set the embedding matrix to be cluster assignment matrix which is proposed in Li et al. (2014). Hence, $Y \in \{0,1\}^{n \times c}$ is discrete, which may make Eq. (12) be an NP-hard problem (Shi and Malik 2000). To address this problem, we use a well-known technique to relax the discrete variable Y to a continuous variable using Eq. (2). Therefore, Eq. (12) can be rewritten as:

$$\begin{aligned} \min_H \text{tr}(H^T LH) \\ \text{s.t. } H^T H = I \end{aligned} \quad (13)$$

where H is nonnegative; however, it has negative elements if (13) is directly solved (Li et al. 2014) which may deteriorate the accuracy of the results. Therefore, we add a nonnegative constraint to ensure the pseudo labels are authentic and accurate:

$$\begin{aligned} \min_H \text{tr}(H^T LH) \\ \text{s.t. } H^T H = I, \quad H \geq 0 \end{aligned} \quad (14)$$

3.1.2 Kernel discriminant model based on spectral clustering

To reveal the structure of the original datasets, we use the manifold information as well as the discriminative information. We combine the idea of LDA with spectral clustering and define between-cluster scatter matrix S_b to make the distance between different clusters the largest possible and a within-cluster scatter matrix S_w to make the distance between data points within the same clusters the smallest possible. Inspired by Mika et al. (1999), we also extend the LDA to nonlinear cases by kernel method. Let's $C_n = I_n - (1/n)I_n I_n^T$ denote a matrix used for centering the data by subtracting the mean of the data, where I_n is an identity matrix, and $\tilde{X} = X C_n$ denote the centered dataset. Hence, we define the total scatter matrix S_t and between-cluster scatter matrix S_b as follows:

$$S_t = \tilde{X} \tilde{X}^T \quad (15)$$

$$S_b = \tilde{X} H H^T \tilde{X}^T \tag{16}$$

We define mapping function $\phi(\bullet)$ to map the linearly inseparable data $\mathbf{x}_i \in R^d$ to a high-dimensional Γ :

$$\varphi : R^d \rightarrow \Gamma, \quad \mathbf{x} \rightarrow \varphi(\mathbf{x}) \tag{17}$$

We assume the dataset in high dimensional space is linearly separable. Inspired by Shang et al. (2016), we obtain the mapping matrix \hat{S}_b and \hat{S}_t as follows:

$$\hat{S}_b = \varphi(\mathbf{X}) C_n H H^T C_n^T \varphi(\mathbf{X})^T \tag{18}$$

$$\hat{S}_t = \varphi(\mathbf{X}) C_n C_n^T \varphi(\mathbf{C})^T \tag{19}$$

Then the discriminant model in Γ is obtained by the following formulation:

$$\max_H \text{tr} \left((\hat{S}_t + \mu I_n)^{-1} \hat{S}_b \right) \tag{20}$$

where $\mu > 0$, μI_n is added to guarantee the matrix $(\hat{S}_t + \mu I_n)$ is always invertible. Note that $\text{tr}(H^T C_n H) = \text{tr}(H^T (I_n - (1/n) I_n I_n^T) H)$ is constant and equivalent to $K - 1$. By subtracting this term from (20), we rewrite it as the following minimization problem:

$$\min_H \text{tr} \left(H^T \left(C_n - C_n^T (C_n + \mu K^{-1})^{-1} C_n \right) H \right) \tag{21}$$

where $K = \phi(\mathbf{x})^T \bullet \phi(\mathbf{x})$ is a kernel function. We can also design and use a kernel function satisfying the *mercer* theorem. There are already many mature kernel function developed, such as linear kernel, Gaussian kernel, Polynomial kernel and Cosine kernel (Mika et al. 1999). In this paper, we would like to use Gaussian kernel as the kernel function defined as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2} \right) \tag{22}$$

where σ is the scale parameter. We put $G = C_n - C_n^T (C_n + \mu K^{-1})^{-1} C_n$ and then rewrite (21) as follows:

$$\min_H \text{tr} \left(H^T G H \right) \tag{23}$$

Using (15)–(23) we obtain a discriminative model in Eq. (23). Next, we will show it is inherently a spectral clustering model and G is a Laplacian matrix.

Theorem 1 *The matrix G in Eq. (23) is a Laplacian matrix referring to Yang et al. (2010).*

For proving the Theorem 1, it’s worth proving two lemmas.

Lemma 1 *Denote $Q = I_n - (C_n + \mu K^{-1})^{-1}$, $\exists \mu$ leads Q as a positive semi-definite matrix.*

Proof Given $C_n = I_n - (1/n) I_n I_n^T$, it’s easy clear that C_n is a symmetric positive definite matrix with eigenvalues $\lambda = 1$ and $(n - 1)/n$. A befitting value μ in $(C_n + \mu K^{-1})^{-1}$ results in maximum eigenvalue $\lambda_{max} \leq 1$, i.e. the minimum eigenvalue of Q is bigger than zero. In this paper, we set $\mu = 10^{-12}$. It can be calculated that $\forall \lambda_Q \geq 0$, so Q is a positive semi-definite matrix. □

Lemma 2 *Given a positive semi-definite matrix \mathbf{Q} , \mathbf{BQB}^T must be a positive semi-definite matrix for arbitrary matrix \mathbf{B} .*

Proof Applying Cholesky decomposition to positive semi-definite matrix \mathbf{Q} (Yang et al. 2010), we obtain $\mathbf{Q} = \mathbf{M}^T \mathbf{M}$. Furthermore, we can pre-multiply and post-multiply it by \mathbf{B} to get \mathbf{BQB}^T , we can substitute \mathbf{Q} by $\mathbf{M}^T \mathbf{M}$ and get $\mathbf{BM}^T \mathbf{MB}^T = (\mathbf{MB}^T)^T (\mathbf{MB}^T)$, so \mathbf{BQB}^T is a positive semi-definite matrix. \square

We can prove the Theorem 1 on the basis of the two proved lemma above.

Proof From Lemma 1, \mathbf{G} can be rewritten as:

$$\mathbf{G} = \mathbf{C}_n - \mathbf{C}_n^T (\mathbf{C}_n + \mu \mathbf{K}^{-1})^{-1} \mathbf{C}_n = \mathbf{C}_n \left(\mathbf{I}_n - (\mathbf{C}_n + \mu \mathbf{K}^{-1})^{-1} \right) \mathbf{C}_n^T = \mathbf{C}_n \mathbf{Q} \mathbf{C}_n^T \quad (24)$$

where \mathbf{Q} is a positive semi-definite matrix. So it is easy to know \mathbf{G} is also a positive semi-definite matrix by Lemma 2. Besides, it is found that $\mathbf{C}_n \mathbf{1}_n = (\mathbf{I}_n - (1/n) \mathbf{1}_n \mathbf{1}_n^T) \mathbf{1}_n = \mathbf{0}$. Hence, $\mathbf{G} \mathbf{1}_n = \mathbf{0}$, that is, 0 is the eigenvalue of \mathbf{G} with corresponding eigenvector $\mathbf{1}$. Above all, we can draw a conclusion that \mathbf{G} is a Laplacian matrix. \square

From the Theorem 1 we realize that Eq. (23) represents not only a discriminative model but also a spectral clustering one. This implies the simultaneous consideration of manifold information and discriminative information (Luxburg 2007; Nie et al. 2010), which lays a solid foundation for feature selection later.

Combining Eqs. (14), (22) and (23), the first term of the proposed algorithm is obtained, i.e. the kernel discriminant model based on spectral clustering is:

$$\begin{aligned} & \min_{\mathbf{H}} \operatorname{tr}(\mathbf{H}^T \mathbf{G} \mathbf{H}) \\ & \text{s.t. } \mathbf{H}^T \mathbf{H} = \mathbf{I}, \quad \mathbf{H} \geq 0 \end{aligned} \quad (25)$$

where $\mathbf{G} = \mathbf{C}_n - \mathbf{C}_n^T (\mathbf{C}_n + \mu \mathbf{K}^{-1})^{-1} \mathbf{C}_n$ is a Laplacian matrix and \mathbf{K} is a kernel function.

3.2 Regression learning

Here we are going to discuss the second term of our method. We add a regression term to the proposed formulation of our method in addition to the feature selection formulation (Zhao and Liu 2007; Cai et al. 2010). In specific, we transform the samples to the corresponding low-dimensional embedding space to fit the scaled cluster assignment matrix. Let's $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$, denotes a transformation matrix where $\{\mathbf{w}_i\}_{i=1}^m$ is the transformation vector of each sample and m is the embedded dimension. In order to match the labels with embedded data, we set $m = c$, i.e. $\mathbf{W} \in \mathbb{R}^{d \times c}$. Hence, the second term of the proposed algorithm can be expressed as follows:

$$\min_{\mathbf{W}} \sum_{i=1}^n \left\| \mathbf{h}_i - \mathbf{X}^T \mathbf{w}_i \right\|_F^2 = \left\| \mathbf{H} - \mathbf{X}^T \mathbf{W} \right\|_F^2 \quad (26)$$

We use the *Frobenious-norm* in the cost function formulation. If \mathbf{H} is known, we can compute \mathbf{W} by minimizing Eq. (26), whose row vector $\hat{\mathbf{w}}_i$ represents the importance of each feature. In order to guarantee the generalization of the proposed formulation in addition to a small error value, we add a regularization constraint to Eq. (26).

3.3 Feature selection

As row vectors of \mathbf{W} have been defined above, we can rewrite the \mathbf{W} as follows:

$$\mathbf{W} = [\hat{\mathbf{w}}_1^T, \hat{\mathbf{w}}_2^T, \dots, \hat{\mathbf{w}}_d^T]^T \tag{27}$$

The third term of the purposed algorithm is to balance the fitting ability and the generalization ability. \mathbf{W} can be considered as a representation of features whose each row represents one feature. To select features, we impose a sparse structure on \mathbf{W} as well as a regression term to remove less important features. We use $L_{2,1}$ -norm for the regularization term, which can make each row of \mathbf{W} sparse and select more discriminative features. Hence, the formulation of the third term is defined as follows:

$$\min_{\mathbf{W}} \sum_{i=1}^d (\|\hat{\mathbf{w}}_i\|_2)^1 = \sum_{i=1}^d \left(\sum_{j=1}^c |\mathbf{w}_{ij}|^2 \right)^{1/2} = \|\mathbf{W}\|_{2,1} \tag{28}$$

When the \mathbf{W} is obtained, we score each row of it and rank them from large to small. The larger score the row has, the more important the feature is.

3.4 KFDRL formulations and solution

3.4.1 The framework

We use a non-negative constraint on \mathbf{W} to satisfy its physical significance and to guarantee that the result is accurate. Using the nonnegative constraint and Eqs. (25), (26) and (28), we can now write the formulation of KFDRL as follows:

$$\begin{aligned} \min L(\mathbf{W}, \mathbf{H}) &= \arg \min \text{tr}(\mathbf{H}^T \mathbf{G} \mathbf{H}) + \beta \left(\|\mathbf{H} - \mathbf{X}^T \mathbf{W}\|_F^2 + \alpha \|\mathbf{W}\|_{2,1} \right) \\ \text{s.t. } \mathbf{W} &\geq 0, \quad \mathbf{H}^T \mathbf{H} = \mathbf{I}, \quad \mathbf{H} \geq 0 \end{aligned} \tag{29}$$

where α and β are balanced parameters, α plays a role in balancing the fitting and generalization. According to Eq. (29), we would like to briefly conclude the process of KFDRL as follows. The spectral clustering model and regression learning method are used to obtain \mathbf{W} and \mathbf{H} in an unsupervised way, and the regularization term balances the fitting and generalization. The score of each row of \mathbf{W} is regarded as the importance of each feature.

3.4.2 The optimization

We cannot find a closed form solution to Eq. (29) because the $L_{2,1}$ -norm is non-smooth. Inspired by Lee and Seung (1999), we use alternate iteration method to find the optimal \mathbf{W} and \mathbf{H} . Hence, we use Lagrange relaxation and write the Lagrange multiplier form of Eq. (29) as follows:

$$\begin{aligned} \min L(\mathbf{W}, \mathbf{H}) &= \text{tr}(\mathbf{H}^T \mathbf{G} \mathbf{H}) + \beta \left(\|\mathbf{H} - \mathbf{X}^T \mathbf{W}\|_F^2 + \alpha \|\mathbf{W}\|_{2,1} \right) \\ &\quad + \frac{\lambda}{2} \|\mathbf{H}^T \mathbf{H} - \mathbf{I}\|_F^2 + \text{tr}(\psi \mathbf{H}^T) + \text{tr}(\phi \mathbf{W}^T) \end{aligned} \tag{30}$$

where λ is selected to be a large enough number, namely $\lambda = 10^8$, to control the orthogonal constraint. Furthermore, ψ and φ are two Lagrangians for constraining \mathbf{W} and \mathbf{H} to be non-negative. We consider the following cases:

- (1) Considering \mathbf{H} to be fixed, we can rewrite Eq. (30) as a function of \mathbf{W} .

$$\begin{aligned} \min L_1(\mathbf{W}) &= \|\mathbf{H} - \mathbf{X}^T \mathbf{W}\|_F^2 + \alpha \|\mathbf{W}\|_{2,1} + tr(\phi \mathbf{W}^T) \\ &= \|\mathbf{H} - \mathbf{X}^T \mathbf{W}\|_F^2 + \alpha tr(\mathbf{W}^T \mathbf{U} \mathbf{W}) + tr(\phi \mathbf{W}^T) \end{aligned} \tag{31}$$

where $\mathbf{U} \in \mathbb{R}^{d \times d}$ is a diagonal matrix whose diagonal elements satisfy the following formulation:

$$u_{ii} = \frac{1}{2 \|\hat{\mathbf{w}}_i\|_2} \tag{32}$$

If we fix \mathbf{U} , we can conclude from $\frac{\partial L}{\partial \mathbf{W}} = 0$ that:

$$2\mathbf{X}\mathbf{X}^T \mathbf{W} - 2\mathbf{X}\mathbf{H} + 2\alpha \mathbf{U} \mathbf{W} + \phi = 0 \tag{33}$$

Considering the KKT condition $\varphi_{ij} w_{ij} = 0$, we have:

$$w_{ij} \leftarrow w_{ij} \frac{(\mathbf{X}\mathbf{H})_{ij}}{(\mathbf{A}\mathbf{W})_{ij}} \tag{34}$$

where $\mathbf{A} = \mathbf{X}\mathbf{X}^T + \alpha \mathbf{U}$.

- (2) Considering \mathbf{U} and \mathbf{W} are fixed, Eq. (30) can be redefined as a function of \mathbf{H} :

$$\min L_2(\mathbf{H}) = tr(\mathbf{H}^T \mathbf{G} \mathbf{H}) + \beta \|\mathbf{H} - \mathbf{X}^T \mathbf{W}\|_F^2 + \frac{\lambda}{2} \|\mathbf{H}^T \mathbf{H} - \mathbf{I}\|_F^2 + tr(\psi \mathbf{H}^T) \tag{35}$$

The solution to Eq. (35) can be computed by $\frac{\partial L}{\partial \mathbf{H}} = 0$, as follows:

$$2\mathbf{G}\mathbf{H} + 2\beta(\mathbf{H} - \mathbf{X}^T \mathbf{W}) + 2\lambda \mathbf{H}(\mathbf{H}^T \mathbf{H} - \mathbf{I}) + \psi = 0 \tag{36}$$

Considering the KKT condition $\psi_{ij} h_{ij} = 0$, we have:

$$h_{ij} \leftarrow h_{ij} \frac{(\lambda \mathbf{H})_{ij}}{[\mathbf{G}\mathbf{H} + \beta(\mathbf{H} - \mathbf{X}^T \mathbf{W}) + \lambda(\mathbf{H}\mathbf{H}^T \mathbf{H})]_{ij}} \tag{37}$$

where $\mathbf{G} = \mathbf{G}^+ - \mathbf{G}^-$, and $\mathbf{G}^+ = (|\mathbf{G}| + \mathbf{G})/2$, $\mathbf{G}^- = (|\mathbf{G}| - \mathbf{G})/2$.

- (3) Considering \mathbf{W} is fixed, \mathbf{U} can be updated by Eq. (32). Hence, at every iteration of the proposed algorithm, we compute the updated value of \mathbf{W} and \mathbf{H} as summarized in Table 1.

4 Algorithm analysis

In this section, we present more analysis on KFDRL in detail, namely convergence and computational complex.

Table 1 The procedure of KFDRL

Input: dataset X , kernel function K , number of cluster/embedded dimension c , iteration time t , parameters $\alpha, \beta, \sigma, \mu$ and λ , number of selected features s .
 Output: selected feature subset.

1. Compute G according to kernel function.
2. Optimize W and H by alternate iteration method.
 - 2.1 Initialize $U = I_d, H = I \in R^{n \times c}$.
 - 2.2 Fixed U , update H by Eq. (37), and then update W by Eq. (34) when U and H fixed.
 - 2.3 Fixed W , update U by Eq. (32).
 - 2.4 Repeat 2.2 and 2.3 for t times (or satisfying the convergence condition).
3. Feature selection:
 - 3.1 Compute each score of features according to $\{\|\hat{w}_i\|_2\}_{i=1}^d$.
 - 3.2 Select the largest s values.

4.1 Convergence of KFDRL

Since the KFDRL is formalized as a minimization problem we need to proof that the proposed algorithm converges to an optimal solution of the objective function in Eq. (29). The convergence proof presented here is very similar to the one presented in Shang et al. (2016), Lin (2007). With U and W fixed, it is easily verified that H is convergence with an auxiliary function as in Shang et al. (2016), Lin (2007). So does the W .

Referring to Eq. (35), it is obviously that $L_2(H^{t+1}) \leq L_2(H^t)$ with the monotonically non-increased H mentioned above. That means:

$$L(W^t, U^t, H^{t+1}) \leq L(W^t, U^t, H^t) \tag{38}$$

We firstly rewrite L_1 in Eq. (31) as follows:

$$L_1(W) = \|X^T W - H\|_F^2 + tr(\phi W^T) + \alpha tr(W^T U W) \tag{39}$$

It's easily to know that $L_1(W^{t+1}) \leq L_1(W^t)$ with the monotonically non-increased W mentioned above when U and H fixed. Combining Eqs. (38) and (39), the following inequality could be presented:

$$L(W^{t+1}, U^t, H^{t+1}) \leq L(W^t, U^t, H^{t+1}) \tag{40}$$

We need to present a lemma before a further convergence proof of the proposed algorithm.

Lemma 3 For arbitrary vectors $x, y \in R^n$, the following inequality holds:

$$\|x\|_2 - \frac{\|x\|_2^2}{2\|y\|_2} \leq \|y\|_2 - \frac{\|y\|_2^2}{2\|y\|_2} \tag{41}$$

The proof of this lemma can be found in Gu and Sheng (2017). We use this lemma to proof the convergence of the proposed algorithm of KFDRL in the following.

Using Lemma 3 we can show that the following inequality holds:

$$\frac{\|\hat{w}_i^{t+1}\|_2^2}{2\|\hat{w}_i^t\|_2} - \|\hat{w}_i^{t+1}\|_2 \geq \frac{\|\hat{w}_i^t\|_2^2}{2\|\hat{w}_i^t\|_2} - \|\hat{w}_i^t\|_2 \tag{42}$$

which could be derived as:

$$\begin{aligned}
 & \frac{\|\hat{\mathbf{w}}_i^{t+1}\|_2^2 - \|\hat{\mathbf{w}}_i^t\|_2^2}{2\|\hat{\mathbf{w}}_i^t\|_2} \geq \|\hat{\mathbf{w}}_i^{t+1}\|_2 - \|\hat{\mathbf{w}}_i^t\|_2 \\
 \Rightarrow & \frac{\left(\|\hat{\mathbf{w}}_i^{t+1}\|_2 + \|\hat{\mathbf{w}}_i^t\|_2\right)\left(\|\hat{\mathbf{w}}_i^{t+1}\|_2 - \|\hat{\mathbf{w}}_i^t\|_2\right)}{2\|\hat{\mathbf{w}}_i^t\|_2} \geq \|\hat{\mathbf{w}}_i^{t+1}\|_2 - \|\hat{\mathbf{w}}_i^t\|_2 \\
 \Rightarrow & \|\hat{\mathbf{w}}_i^{t+1}\|_2 + \|\hat{\mathbf{w}}_i^t\|_2 \geq 2\|\hat{\mathbf{w}}_i^t\|_2 \\
 \Rightarrow & \|\hat{\mathbf{w}}_i^{t+1}\|_2 \geq \|\hat{\mathbf{w}}_i^t\|_2
 \end{aligned} \tag{43}$$

Combining Eqs. (32) and (43), we have:

$$\mathbf{U}^{t+1} \leq \mathbf{U}^t \tag{44}$$

Considering Eq. (39) and Eq. (40), we can obtain the following inequality formulation when \mathbf{W} and \mathbf{H} fixed:

$$L(\mathbf{W}^{t+1}, \mathbf{U}^{t+1}, \mathbf{H}^{t+1}) \leq L(\mathbf{W}^{t+1}, \mathbf{U}^t, \mathbf{H}^{t+1}) \tag{45}$$

\mathbf{U} is an intermediate variable function of \mathbf{W} , it is straight forward that any reformulation of the objective function using an intermediate variable cannot affect this convergence. We can demonstrate the convergence of the algorithm proposed in Table 1 by some experimental results in the next section.

4.2 Computational complexity analysis

In this section, we analyze the computational complexity of KFDRL. It is evident that the algorithm requires the highest computation cost when it computes Laplacian matrix \mathbf{G} , updates variables \mathbf{H} and \mathbf{W} , and so on. Computation of \mathbf{G} including matrix inversion indicates the computation complexity $O(n^3)$. The update of \mathbf{H} including both matrix inversion and multiplication indicates the computation complexity $O(d^3 + nd^2 + n^2d)$. On the contrary, the update of \mathbf{W} has relatively low computation complexity $O(nd + d^2)$. We neglect the influence of embedded dimension c where $c \ll d$ and $c \ll n$. In conclusion, the total time complexity of the algorithm is $O(n^3 + t(d^3 + nd^2 + n^2d))$ where t represents the number of iteration.

Time costs on different datasets will be shown in next section to display the computational complexity visually.

5 Experiments and analysis

In order to show the effectiveness of the method proposed in this paper, we present a number of experiments to imply the superiority of the method in different aspects. First, we introduce datasets and metrics. The parameters of KFDRL are set and a number of state-of-the-art algorithms are presented used to validate the result obtained by the algorithm proposed in this paper. Finally, we present 5 experiments.

1. The first experiment is to show the convergence rate of the objective function discussed before.
2. The second one is a toy example to intuitively show the effectiveness of KFDRL.

Table 2 Information of the experimental datasets

Dataset	Samples	Dimension	Classes
Coil20	1440	1024	20
PIE_pose27	2856	1024	68
YALE	2414	1024	38
Orl	400	1024	40
Umist	575	644	20
Ionosphere	351	34	2
BC	569	30	2
AT&T	400	10,304	40
Isolet	1560	617	26
Sonar	208	60	2

3. In the third experiment, we present feature selection and K-means clustering as a joint problem.
4. The fourth experiment is a classification problem. We first perform feature selection and then nearest neighborhood classification. Experiments 3 and 4 are the application of dimensionality reduction in clustering and classification.
5. The final experiment aims to illustrate the sensitivity of the result to the parameters.

5.1 Datasets

In this paper, to validate the performance and accuracy of the result of KFDRL, we select nine UCI datasets and three samples from AT&T to perform our experiments. The detailed information about these datasets is listed in Table 2.

5.2 Evaluation metric

In order to analyze the quality of the results obtained by the algorithm, we use the clustering accuracy (ACC) and the normalized mutual information (NMI), which are two major metrics for clustering.

5.2.1 ACC

Assuming c_i and g_i represents pseudo label and true label respectively for $\forall x_i$. ACC clustering accuracy is defined as:

$$ACC = \frac{\sum_{i=1}^n \delta(g_i, \text{map}(c_i))}{n} \quad (46)$$

where n is the total number of samples, $\delta(x, y)$ is delta function. Delta function has a value of one if $x = y$ and zero otherwise. $\text{Map}(\bullet)$ computed by Hungarian algorithm (Strehl and Ghosh 2003) is a function that maps each cluster index to the best class label. Larger value of ACC means better clustering results.

5.2.2 NMI

Given two arbitrary variables P and Q , the NMI is defined as:

$$NMI(P, Q) = \frac{MI(P, Q)}{\sqrt{H(P)H(Q)}} \quad (47)$$

where $MI(P, Q)$ is mutual information between P and Q . $H(\cdot)$ denotes the information entropy (Papadimitriou and Steiglitz 1998). By definition $NMI(P, Q) = 1$ if $P = Q$ and $NMI(P, Q) = 0$ otherwise. We can also formulate NMI in Eq. (47) using the pseudo labels t_l and true labels t_h , as follows:

$$NMI = \frac{\sum_{l=1}^c \sum_{h=1}^c t_{l,h} \log\left(\frac{n \cdot t_{l,h}}{t_l t_h}\right)}{\sqrt{\left(\sum_{l=1}^c t_l \log \frac{t_l}{n}\right) \left(\sum_{h=1}^c t_h \log \frac{t_h}{n}\right)}} \quad (48)$$

where $t_{l,h}$ is the number of samples identical among the two label sets. Obviously, the larger NMI is, the better clustering results have.

5.2.3 Classification accuracy

We use Euclidean distance to measure the classification accuracy. The dataset is divided in training and test sets. We use parts of dataset to train the classifier whereas the rest of the dataset is used for testing. Euclidian distance is used to measure the distance between the sample points. Hence, the samples of training and test sets belong to the same cluster if they are close.

5.3 Settings in experiments

5.3.1 Parameters settings for KFDRL

Based on the experiments in different algorithmic contexts, parameters selection affects the result of the corresponding algorithm. Therefore, it is very important to select the best parameters of KFDRL. With reference to Table 1, we can see that KFDRL has five parameters including α , β , σ , μ and λ . Some of these parameters can be set using existing method (Li et al. 2014; Hou et al. 2014; Fang et al. 2014; Shang et al. 2016). Since the algorithm shows no sensitivity to the value of λ if it has large enough value, we set $\lambda = 10^8$, with reference to Li et al. (2014). We set the scaled parameter for Gaussian kernel function $\mu = 10^{-12}$, based on the argument presented in Sect. 3. On the other hand, the other parameters must be set carefully as the result is sensitive to their value. Hence, we set $\sigma = \{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000\}$ based on our experience and a grid-search strategy is used to determine a small range of parameter values including the best value. Finally, the best value is computed with small enough step size tuning of the parameters in that region resulting in $\alpha = [0.1, 3.5]$ and $\beta = \{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000\}$.

5.3.2 The compared algorithms

We compare the results obtained by the proposed method with the ones obtained by some classical feature selection algorithms for clustering and classification problems, including

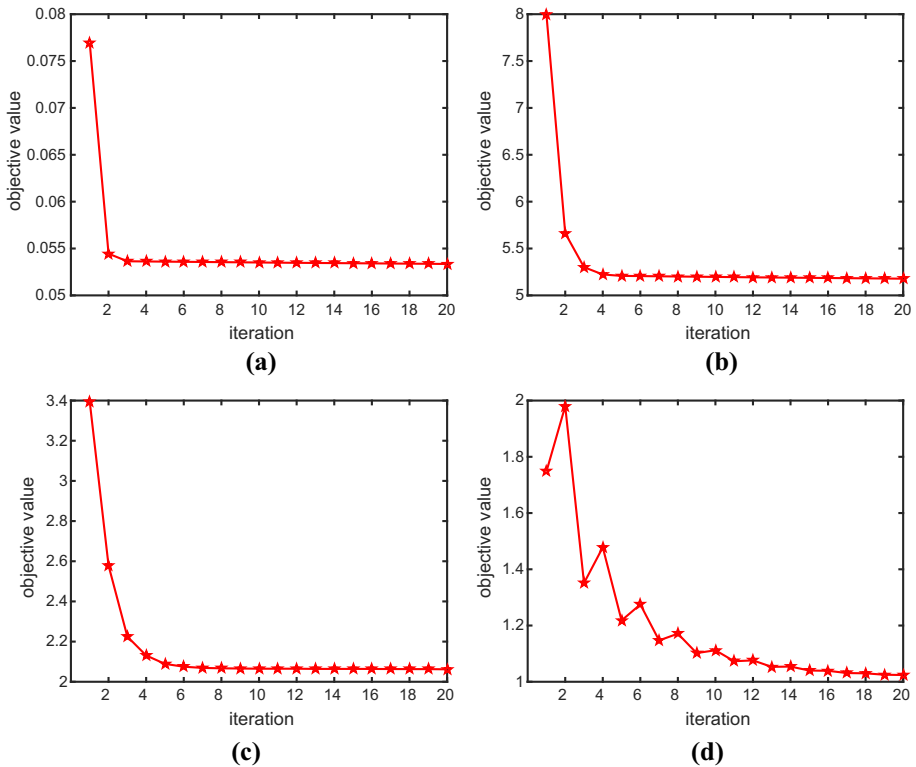


Fig. 1 The convergence of objective function of KFDRL on four selected datasets. **a** Convergence on BC, **b** convergence on Orl, **c** convergence on Sonar, **d** convergence on Umist

LapScore, SPEC, MCFS, MRSF, JELSR, LSPE and DFSC (Zhao and Liu 2007; Cai et al. 2010; He et al. 2005; Zhao et al. 2010; Hou et al. 2014; Fang et al. 2014; Shang et al. 2016). To be fair, we use the parameters reported in the corresponding original works (Zhao and Liu 2007; Cai et al. 2010; He et al. 2005; Zhao et al. 2010; Hou et al. 2014; Fang et al. 2014; Shang et al. 2016). Hence, we use the best result of every algorithms with the parameters. We will also discuss the variable of different experiments in the later sections.

5.4 The convergence of KFDRL

This experiment is presented here to intuitively show the convergence and the rate of convergence. We apply the algorithm on four datasets including BC, Umist, Orl and Sonar. The evolutions of the objective values are shown in Fig. 1. These four pictures show the convergence of KFDRL.

It can be seen from Fig. 1, the value of objective function decreases very fast over the first three iterations for all four datasets. This evidences that KFDRL is very efficient in terms of convergence rate. This testifies that we can set the maximum number of iteration to five during the next experiment to decrease the computation time.



Fig. 2 Top row: an image of a face belonging to the 5th sample of class 2 of AT&T dataset; Middle row: an image of a face belonging to the 7th sample of the class 3 of AT&T dataset; Bottom row: an image of a face belonging to the 10th sample of the class 6 of AT&T dataset

5.5 Toy example

We randomly choose three different pictures from AT&T dataset to illustrate the effectiveness of KFDRL algorithm. This shows KFDRL always trends to select more discriminative features. We draw a number of pictures from the dataset respectively with {1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192, 9220, 10244} features selected from each sample. The pixel point is considered as black when it is not chosen. In Fig. 2, three samples are displayed in three rows. From left to right, the pictures correspond to {1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192, 9220, 10244} features respectively. This figure shows how the picture drawn using different number of features approximates the original image. The more features we select, the more similar picture to the original image is drawn. As it is shown in Fig. 2, KFDRL tends to preserve more discriminative features even with small and fixed number of features. For example, the main profile of the face are recognizable in sample 1 and 3 using the picture drawn with only 1024 features, that is almost 10% of the total number of features. Nonetheless, sample 2 shows a bit less effectiveness of the method since the picture drawn by the algorithm with 1024 features does not show the nose, forehead and rim of the eye very important features for recognizing a face. In conclusion, KFDRL can automatically characterize the most important features in an image. The 4096 features (almost 40% of the total features) selected by KFDRL are necessary and sufficient to draw a picture identical to the original image while the rest of features can be regarded as redundant and irrelevant ones.

5.6 Feature selection for K-means

In this section, we show the performance of K-means clustering by KFDRL where ACC and NMI are used as metrics. We set the number of selected features to be {10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120} on Coil20, Orl, Isolet and Umist, and {2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22} on Ionosphere, BC and Sonar. The number of feature for each algorithm is chosen such that we obtain the best clustering result. To set the parameters of the KFDRL, namely α and β , we follow the procedure presented in Sect. 5.3.1. Nonetheless, σ is set to 1

since the result does not show any sensitivity to its value. This will be discussed and showed in detail in the later experiments.

- 1) Algorithms used for validation of the results are LapScore, SPEC, MCFS, MRSF, JELSR, LSPE and DFSC. We classify these algorithms into three categories (I) LapScore is a classical method (II) SPEC, MCFS and MRSF are two-step methods and (III) the rest of them are one-step methods.
- 2) We use the following datasets to test the algorithms: Coil20, Orl, Ionosphere, Isolet, Umist, BC and Sonar.
- 3) Procedure of the experiment: we first run all feature selection algorithms including KFDRL on each dataset. Then, the K-means clustering is applied to the datasets obtained by using feature selection algorithms. Finally, the best results of each algorithm are reported in Table 3. Since the influence from initial cases cannot be ignored for K-means clustering, we run the K-means for 100 times and then compute the average value to reduce the error.

Tables 3 and 4 display the best ACC and NMI obtained by different feature selection algorithms on all the datasets. The second rows in the two tables show clustering results on the original data. The best results on each dataset are marked as bold. We omit the NMI results on Sonar dataset in Table 4 because NMIs of all the algorithms including KFDRL are no larger than 10%. We can know that it lacks the mutual information and has a poor performance of the algorithm in clustering Sonar dataset.

We can summarize the results presented in Tables 3 and 4 as follows:

1. ACC and NMI illustrate that KFDRL performs better than other feature selection algorithms on most of the datasets.
2. The traditional dimensionality reduction algorithm LapScore, which is a modified for Laplacian Eigenmaps (LE), recognizes and utilizes manifold structure embedded in high dimensional data by graph Laplacian without learning mechanism. The results reported in Tables 3 and 4 show that the joint-framework or two-step algorithms can perform better than traditional and one-step algorithms.
3. In comparison to other algorithms, KFDRL demonstrated superiorities. First, it simultaneously benefits from manifold and discriminative information, because the objective function proposed in KFDRL combines manifold learning and discriminative regression learning. Second, the constraints imposed on the scaled assignment cluster matrix \mathbf{H} and on the transformation matrix \mathbf{W} can help it to find physically meaningful features (a forced constraint imposed to the scaled assignment cluster matrix \mathbf{H} and the transformation matrix \mathbf{W} can fit the physical meaning). The best results obtained with feature selection for each dataset have better performance than the clustering results obtained on the original datasets. It indicates that feature selection can not only reduce the size of data to increase the computation speed, but also efficiently remove the redundant and noisy information demonstrating the great significance of data pre-processing.
4. KFDRL cannot show a better performance than the other algorithms in a few cases. For example, it shows that JELSR performs better than KFDRL with ACC metric on Sonar dataset. Moreover, it shows that KFDRL performs worse than all features selected and DFSC with NMI metric on the Isolet dataset, as well as JELSR and LSPE on the Umist dataset. It is noted that these methods mentioned above are all single-step methods, which describe their better performance. We explain several reasons for the worse performance of KFDRL. First, Sonar and Isolet are both voice data. The advantage of KFDRL can be identified in the task of clustering image data because the embedded dimension and the number of classes are considered to be identical (Cai et al. 2008, 2011; Liu et al. 2012),

Table 3 The best results of ACC for each algorithm in different datasets (mean \pm STD%)

	Coil20	Orl	Isolet	Sonar	Umist	Ionosphere	BC
All features	61.39 \pm 0.24	50.15 \pm 0.35	50.69 \pm 0.86	54.72 \pm 1.10	44.33 \pm 1.02	66.21 \pm 0.58	77.24 \pm 0.29
LapScore	50.28 \pm 2.67	44.76 \pm 0.23	49.02 \pm 0.47	59.13 \pm 1.27	37.23 \pm 1.04	66.85 \pm 2.21	71.23 \pm 0.46
SPEC	55.68 \pm 2.43	50.03 \pm 0.21	50.34 \pm 0.83	61.20 \pm 1.17	44.01 \pm 1.13	68.58 \pm 2.09	75.39 \pm 0.19
MCFS	56.11 \pm 2.54	50.10 \pm 0.98	55.31 \pm 0.62	51.60 \pm 0.65	47.29 \pm 0.78	58.38 \pm 2.03	71.02 \pm 0.68
MRSF	59.43 \pm 2.33	48.86 \pm 1.02	52.45 \pm 0.45	66.35 \pm 1.02	49.56 \pm 1.86	61.66 \pm 1.56	72.89 \pm 0.15
JELSR	60.54 \pm 2.56	52.55 \pm 1.12	54.65 \pm 2.36	67.91 \pm 0.22	52.10 \pm 2.00	68.18 \pm 2.57	74.86 \pm 0.41
LSPE	62.32 \pm 2.24	51.78 \pm 0.50	55.31 \pm 0.66	66.35 \pm 1.78	48.89 \pm 1.21	70.45 \pm 2.84	76.69 \pm 0.19
DFSC	62.28 \pm 3.23	53.36 \pm 2.43	55.98 \pm 3.44	58.88 \pm 2.11	49.48 \pm 3.62	72.23 \pm 0.00	86.12 \pm 0.00
KFDRL	65.15 \pm 2.36	56.90 \pm 2.27	61.69 \pm 2.46	66.83 \pm 0.00	53.01 \pm 3.32	82.62 \pm 0.00	91.21 \pm 0.00

Table 4 The best results of NMI for each algorithm in different datasets (mean \pm STD %)

	Coil20	ORL	Isolet	Umist	Ionosphere	BC
All features	74.77 \pm 1.21	70.16 \pm 1.16	73.12 \pm 0.86	60.30 \pm 1.45	12.95 \pm 0.00	18.00 \pm 0.00
LapScore	60.61 \pm 1.36	67.80 \pm 1.56	66.43 \pm 1.65	56.22 \pm 1.54	8.68 \pm 0.00	17.55 \pm 0.00
SPEC	60.66 \pm 1.58	71.03 \pm 1.12	66.90 \pm 1.85	57.44 \pm 1.35	9.05 \pm 0.33	16.79 \pm 0.00
MCFS	68.34 \pm 1.78	71.74 \pm 1.33	71.06 \pm 1.82	68.87 \pm 1.48	1.32 \pm 0.65	18.28 \pm 0.00
MRSF	68.85 \pm 2.03	71.55 \pm 2.00	70.32 \pm 1.56	66.67 \pm 1.41	3.82 \pm 0.00	17.91 \pm 0.00
JELSR	73.18 \pm 2.37	72.23 \pm 1.88	70.65 \pm 1.88	69.80 \pm 1.70	8.48 \pm 1.15	19.67 \pm 0.00
LSPE	75.54 \pm 1.98	72.96 \pm 1.26	70.54 \pm 1.90	70.67 \pm 1.61	13.10 \pm 0.44	19.63 \pm 0.00
DFSC	74.87 \pm 2.77	73.36 \pm 1.44	73.56 \pm 1.35	65.39 \pm 1.54	30.52 \pm 0.68	42.32 \pm 0.00
KFDRL	76.28 \pm 1.14	76.03 \pm 1.35	71.26 \pm 1.20	69.13 \pm 2.01	35.02 \pm 0.00	55.12 \pm 0.00

i.e. $c = m$. However, it cannot have better results on the Sonar and Isolet datasets. We believe the embedding learning employed in JELSR and LSPE on the Umist dataset has superiority over other techniques. As reported in Tables 3 and 4 an algorithm resulting in a better performance of clustering different datasets than others does not exist. Hence, the best algorithm still needs to be chosen based on the nature of the dataset to be clustered and finding an algorithm best for clustering all the different datasets will be an open research question for future.

5. In general, KFDRL performs better than JELSR on most of the datasets, which indicates that the manifold learning alone is not efficient in many cases. Nonetheless, JELSR outperforms KFDRL on the Sonar and Umist dataset, which may be a result of the inherently discriminant structure of the datasets. Hence, a discriminative based algorithm cannot perform as well as it does in the case of non-discriminant datasets. It is worth reminding that KFDRL includes discriminative and manifold learning whereas JELSR is based on manifold learning only.

5.7 Classification

In this section, we apply the dimensionality reduction algorithms to the classification problems. This will improve the classifier efficiency by reducing the corresponding feature dimensions. To study and show the performance improvement, we use the classification accuracy (AC) as the evaluation metric to show how the performance of nearest neighborhood classifier (NN) improves by using KFDRL to select the relevant features. We use the Ionosphere and Coil20 dataset because they have different sizes. From Table 2, we know that Ionosphere is a small size dataset with 34 features and 2 classes while Coil20 is a large size dataset with 1024 features and 20 classes.

We select one algorithm from each three categories (namely traditional, one-step and two-step algorithms) of algorithms recognized in this paper to ensure the representativeness of the experiment. We choose LapScore belonging to the first category, SPEC belonging to the second category, and JELSR belonging to the third category.

Similarly to the method presented in Zhao et al. (2010), we first do dimensionality reduction for original datasets by feature selection. Next, we use 50% of the dataset for training and the rest 50% for testing. NN-classifier is used to perform classification where the results are shown in Fig. 3. The vertical axis in Fig. 3 shows the AC value and the horizontal axis

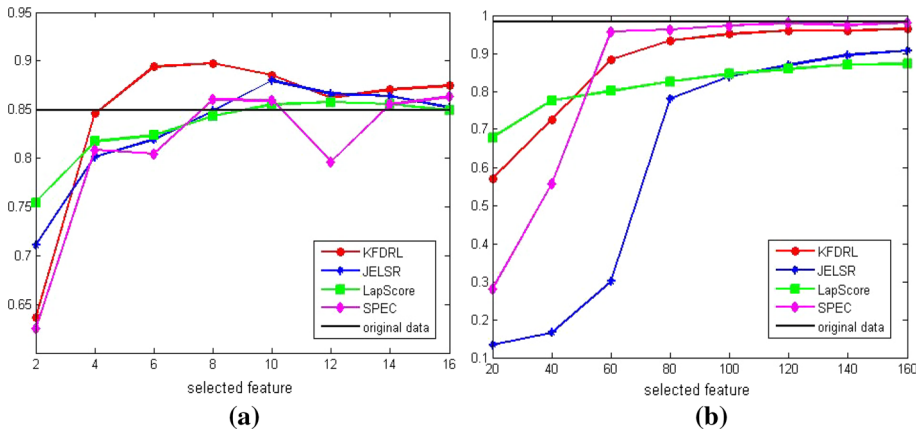


Fig. 3 Classification accuracy (AC) by NN-classifier on two typical datasets. **a** AC with different selected features by different algorithms on Ionosphere, **b** AC with different selected features by different algorithms on Coil20

Table 5 The time cost of classification on Coil20 when selected features $s = 100$ (s)

Method	LapScore	SPEC	KFDRL	Original data
Computational time	94	95	432	940

show the number of features selected. The black lines in two figures are both the classification results using the original dataset. We have chosen the numbers of features {2, 4, 6, 8, 10, 12, 14, 16} for classifying Ionosphere dataset and {20, 40, 60, 80, 100, 120, 140, 160} for classifying Coli20 dataset. The process is repeated 50 times resulting in 50 different partitions. Figure 3 shows some interesting results, which we discuss in the following:

1. KFDRL results in the best performance in classifying small size dataset Ionosphere when the features selected are more than four. When the number of features larger than eight, the results obtained by almost all the algorithms are better than the original dataset for classification.
2. The joint-framework algorithms have a better performance than the traditional algorithm in classifying Coil20 dataset with relatively large size. However, the joint-framework algorithms result in better performance than the two-step algorithm only if the features selected are more than 60. Moreover, KFDRL shows superiority over JELSR. Although the joint-framework algorithms have better performance in most cases, the two-step algorithms show superiority over joint-framework algorithm in some cases.
3. SPEC and KFDRL result in 98% of classification performance using the original dataset by using only 100 selected features, almost 10% of the total number of features, which illustrates that most of the discriminative features are included in the 100 selected features. This significantly accelerates the classification rate due to the significantly fewer number of features used, as it is reported in Table 5.
4. Based on the results of classification of two datasets, we recognize the Ionosphere has noisy and irrelative features as such a better performance is obtained if feature selection algorithm is used. Nonetheless, it describes the fact that the more features selected the less

Table 6 *t* Test for AC in Fig. 3

Dataset	Algorithm	s=2	s=4	s=6	s=8	s=10	s=12	s=14	s=16
Ionosphere	LapScore	F(.00)	W(.00)	W(.00)	W(.00)	W(.00)	B(–)	W(.00)	W(.00)
	SPEC	B(–)	W(.00)	W(.00)	W(.00)	W(.00)	W(.00)	W(.00)	W(.01)
	JELSR	F(.01)	W(.00)	W(.00)	W(.00)	B(–)	B(–)	W(.00)	W(.00)
Dataset	Algorithm	s=20	s=40	s=60	s=80	s=100	s=120	s=140	s=160
Coil20	LapScore	F(.00)	F(.00)	W(.00)	W(.00)	W(.00)	W(.00)	W(.00)	W(.00)
	SPEC	W(.00)	W(.00)	F(.00)	F(.00)	F(.00)	F(.00)	F(.00)	F(.00)
	JELSR	W(.00)	W(.00)	W(.00)	W(.00)	W(.00)	W(.00)	W(.00)	W(.00)

AC value obtained. For Coil20, the best classification results obtained by using KFDRL and SPEC as the feature selection. These results are almost identical to the result obtained by classifying the original dataset. It indicates that Coil20 has many redundant features and it is less noisy. Therefore, dimensionality reduction plays an important role in cutting the computation cost and memory space usage for classifying the Coil20 dataset.

- Although JELSR and KFDRL are joint-frame algorithms, the results of classification problem demonstrate that KFDRL obtains the best results of classification on most datasets by adding discriminative analysis technology. Though different methods are compared in clustering problem in many papers, we show the classification is also worth considering since it is popular application of the dimensionality reduction. We show that the discriminative analysis can improve the performance of classification problem.
- For demonstrating the rationality of the classification experiment and suppressing interference from the randomly sampling, the *t* test is employed to test the reliability of the above AC results. The threshold of statistical significance is set to 0.05 in Table 6. “W” means KFDRL performs better than other algorithms discussed in this paper, “F” indicates KFDRL fails and “B” implies we cannot distinguish the results using statistical method. The value in brackets is *p* value, which indicates the probability of other methods is worse than KFDRL. The smaller the *p* value is, the more confidence we have on the corresponding statement.

The results reported in Table 6 illustrate that the hypothesis matches the results of AC in most cases. Hence, AC can be used as a valid metric to analyze the results of the experiments.

5.8 Parameter sensitivity

5.8.1 Sensitivity analysis of σ

We use part of Ionosphere, Sonar and Coil20 as our test datasets and use $\sigma = \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$ to test the sensitivity of the clustering algorithm with different choice of σ value whereas the other parameters are fixed. The results are shown in Fig. 4a on the BC dataset, Fig. 4b on the Ionosphere dataset, and Fig. 4c on the Coil20 dataset, where the blue and green lines show ACC and NMI, respectively.

It is apparent that ACC and NMI are constant for different value of σ on the BC and Ionosphere datasets because they show a slight change on the Coil20 dataset. These three datasets are different in terms of size and dimension. In Fig. 4, the result of clustering Coil20 dataset is more sensitive to changes of σ than BC and Ionosphere datasets. Considering

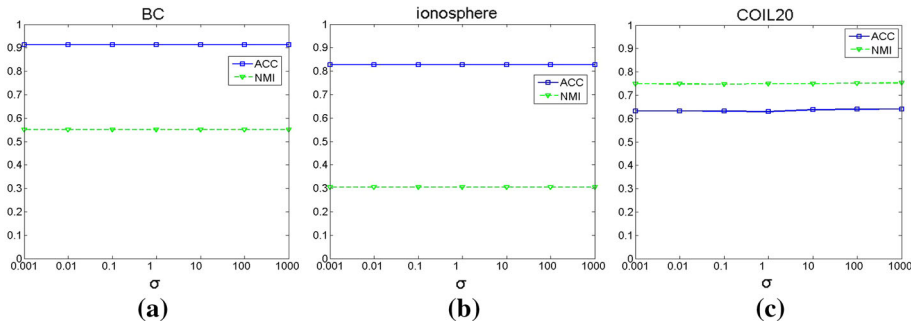


Fig. 4 Clustering stability with different value of σ . **a** the sensitivity of clustering BC to σ , **b** the sensitivity of clustering Ionosphere to σ , **c** the sensitivity of clustering Coil20 to σ

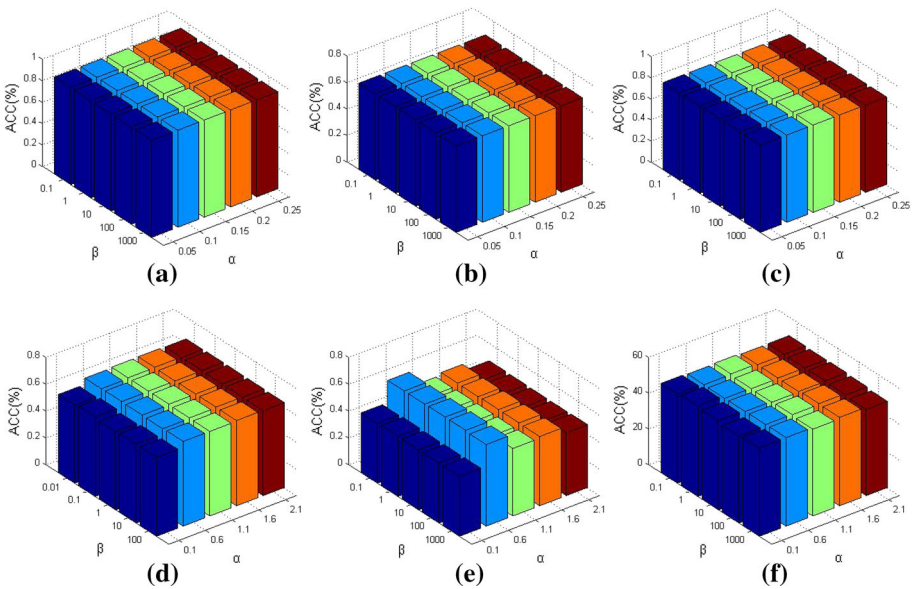


Fig. 5 ACC with different α and β . Different values of α and β are selected for different datasets within the certain range. **a** Sensitivity on BC, **b** sensitivity on Sonar, **c** sensitivity on Ionosphere, **d** sensitivity on Coil20, **e** sensitivity on Isolet, **f** sensitivity on Umist

that Coil20 is much bigger than BC and Ionosphere, we believe that the size of dataset is an important factor of the robustness of kernel function. In conclusion, the σ is almost insensitive to the datasets in this paper.

5.8.2 Sensitivity analysis of α and β

Here we focus on the sensitivity of α and β with other parameters fixed. After a grid-search we set α to in the range of 0.01 and 2.50 and β to be {0.001, 0.01, 0.1, 1, 10, 100, 1000}. We applied K-means clustering to six UCI datasets respectively with different α and β . The results shown in Fig. 5 are averaged over 15 times clustering.

We can interpret the results shown in Fig. 5 as follows:

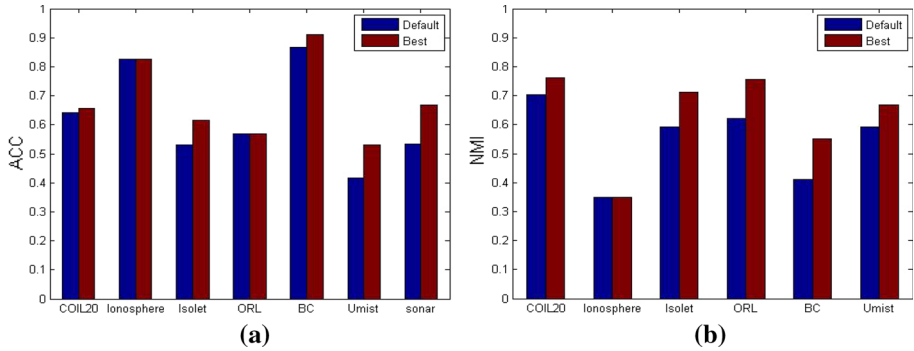


Fig. 6 The performance comparison between the cases with the default (i.e. $\alpha = 1.5$, $\beta = 10$) and best parameters. **a** The performance comparison of ACC, **b** The performance comparison of NMI

1. The ACC is constant for the Umist dataset with different values of α and β . On the other hand, for larger datasets with the number of features over 1000, such as Isolet and Coil20, the ACC changes mostly as a function of α . For the dataset whose dimension is no more than 10^2 , such as Umist, the best ACC obtained by KFDRL can be found using α and β tuned by grid-search only. However, for the larger datasets, it is important to tune the parameters in the certain range determined by the grid-search. Similarly with most of the feature selection algorithms, it is still an open problem to find an efficient method to search a suitable value for parameters. At present, it mostly depends on experiences and test.
2. It is evident that α has a stronger influence on the results than β . ACC value is not sensitive to β value, as shown in Fig. 5, which indicates there is no relevance between the discriminative term and learning framework term according to Eq. (31). In fact, α is the parameter that balances fitting term and generalization term in learning process. It is why ACC is more sensitive to α value than other parameters. Improper choice of α can easily lead to the under-fitting or the over-fitting problem.

Figure 6 shows a comparison of results with the default parameters and the best parameters. This figure illustrates the importance of α and β in KFDRL algorithm. In Fig. 6, blue bars show ACC and NMI corresponding with the parameters randomly selected whereas brown bars show the ACC and NMI with the best-tuned parameters. Here we randomly select $\alpha = 1.5$ and $\beta = 10$.

It can be seen from Fig. 6 that a fixed α results in different performance of ACC and NMI. In other words, different α is needed for different datasets. Considering that different datasets have different intrinsic information, we believe α is impacted by a dataset itself including its size and dimension.

5.9 Time costs

Obviously, time cost is depended on the size of dataset in most cases. In this part, computational complexity in different algorithms is shown visually in the one-off running time form. Here, several representative algorithms are selected as compared algorithms, including LapScore (a classical algorithm), SPEC and MCFS (two-step algorithms), DFSC and JELSR (joint-framework algorithms). Furthermore, six medium scaled with 100 selected features and three small scaled datasets with 30 selected features are tested.

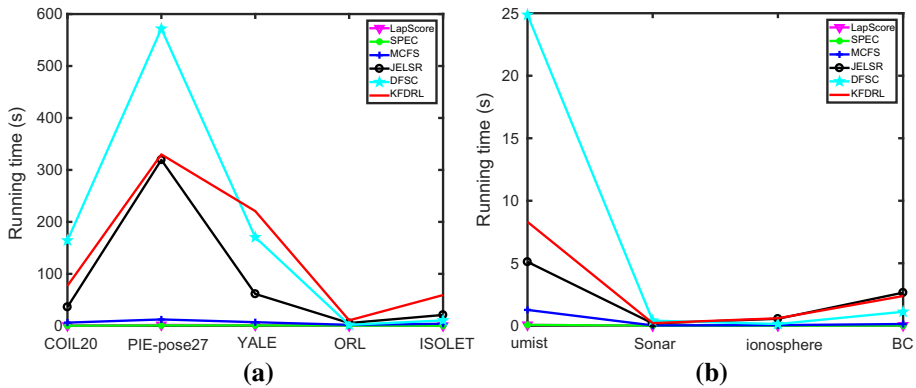


Fig. 7 Running time with different algorithms (s). **a** Running time on the first five datasets (s). **b** Running time on the last four datasets (s)

We draw the results in Fig. 7 as follows, which are obtained in seconds by MATLAB 2014a, 6 GB RAM and a 2.50 GHz CPU.

It can be seen in Fig. 7, KFDRL and DFSC are high-complexity algorithms. Thus, offline cases may be more applicable for the proposed algorithm KFDRL, especially handling big data. Here, larger scaled datasets are not chosen for their out-of-costs on the configuration. In real situation, that problem could be addressed with some developed platform, such as hadoop and spark.

6 Conclusion

A variety of feature selection algorithms have been proposed for dimensionality reduction. However, in most cases, either the manifold information or discriminative information is utilized alone. In contrast, both the manifold information and discriminative information are important for clustering, classification and other applications. Thus, in this paper, a novel unsupervised feature selection algorithm based on discriminant analysis and regression learning (KFDRL) is proposed to reduce dimensionality by better exploiting the underlying information. In particular, both manifold information and discriminative information data are used together. To achieve this goal, the kernel method is used in LDA to handle nonlinear spaces. At the same time, this LDA model is constructed and proved to be form of a spectral clustering. Thus the intrinsic information, i.e. both manifold information and discriminative information are preserved. Next, the kernel model and regression learning are unified into a joint-framework to get better performance. To select features effectively, $L_{2,1}$ -norm is imposed to be a sparse constraint. A simple and efficient method, i.e. the alternative iteration update rule, is used to optimize the objective function and get a sparse representation matrix. Finally, our experiments demonstrate that KFDRL outperforms other algorithms in clustering and classification by removing noise and redundancy more effectively. In addition, the experiment demonstrates the fast convergence properties of KFDRL. The simple example problem is further used to illustrate the validity of KFDRL intuitively. The parameter sensitivity experiment implies that only one parameter in KFDRL is significantly sensitive on different datasets, and needs tuning via optimization. In conclusion, KFDRL performs highly compared to other state-of-the-art methods.

The most pressing problem is the time complexity. The experiment in terms of time costs indicates that the KFDRL has a relatively long time compared with other algorithms, especially tuning parameters. It means that if parameters in KFDRL had been decided, cost of running time can be accepted. Otherwise, the time on tuning parameters cannot be tolerant at present.

There are some remaining aspects of KFDRL which might be improved in future work, either. First, though the alternative update rule is fast and simple, a limitation is that it may converge on local optima, and is easily affected by initial values. Second, how to tune parameters efficiently is still an open problem. Third, the stability is worse when KFDRL is applied to big datasets, where the results significantly depend on parameter α . Finally, the measurement KFDRL adopts is Euclidean distance, which is not always ideal for some real problems. In future work, we will concentrate on the global optimization for feature selection and find a better way to tune parameters. Furthermore, another interesting research question is how to uncover suitable measurements for different data automatically.

Acknowledgements We would like to express our sincere appreciation to the editors and the anonymous reviewers for their insightful comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Natural Science Foundation of China (Nos. 61773304 and 61371201).

References

- Abdulla, W., & Kasabov, N. (2003). Reduced feature-set based parallel CHMM speech recognition systems. *Information Sciences*, 156(1–2), 21–38.
- Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6), 1373–1396.
- Cai, D., He, X., Han, J., & Huang, T. (2011). Graph regularized non-negative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), 1548–1560.
- Cai, D., He, X., Wu, X., & Han, J. (2008). Non-negative matrix factorization on manifold. In *Processing of the 8th IEEE international conference on data mining*, 2008 (pp. 63–72).
- Cai, D., Zhang, C., & He, X. (2010). Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining*, 2010 (pp. 333–342).
- Chen, M., Tsang, W., Tan, M., & Cham, T. (2015). A unified feature selection framework for graph embedding on high dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 27(6), 1465–1477.
- Cheng, Q., Zhou, H., & Cheng, J. (2011). The Fisher–Markov selector: Fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6), 1217–1233.
- Constantinopoulos, C., Titsias, M., & Likas, A. (2006). Bayesian feature ad model selection for gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6), 1013–1018.
- Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2), 185–205.
- Doquire, G., & Verleysen, M. (2013). A graph Laplacian based method to semi-supervised feature selection for regression problems. *Neurocomputing*, 121(18), 5–13.
- Fang, X., Xu, Y., Li, X., Fan, Z., Liu, H., & Chen, Y. (2014). Locality and similarity preserving embedding for feature selection. *Neurocomputing*, 128, 304–315.
- Gu, B., & Sheng, V. (2016). A robust regularization path algorithm for ν -support vector classification. *IEEE Transactions on Neural Networks and Learning Systems*. <https://doi.org/10.1109/TNNLS.2016.2527796>.
- Gu, B., & Sheng, V. S. (2017). A robust regularization path algorithm for ν -support vector classification. *IEEE Transactions on Neural Networks and Learning Systems*, 28(5), 1241–1248.
- Gu, B., Sheng, V. S., Wang, Z., Ho, D., Osman, S., & Li, S. (2015). Incremental learning for ν -support vector regression. *Neural Networks*, 67, 140–150.
- He, X., Cai, D., & Niyogi, P. (2005). Laplacian score for feature selection. In *International conference on neural information processing systems*, 2005 (pp. 507–514), MIT Press.

- He, X., & Niyogi, P. (2004). Locality preserving projections. *Advances in Neural Information Processing Systems*, 16, 153–160.
- Hou, C., Nie, F., Li, X., Yi, D., & Wu, Y. (2014). Joint embedding learning and sparse regression: A framework for unsupervised feature selection. *IEEE Transactions on Cybernetics*, 44(6), 793–804.
- Lee, D., & Seung, H. (1999). Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401, 788–791.
- Li, Z., Liu, J., Yang, Y., Zhou, X., & Lu, H. (2014). Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 26(9), 2138–2150.
- Li, Z., Liu, J., Zhu, X., Liu, T., & Lu, H. (2010). Image annotation using multi-correlation probabilistic matrix factorization. In *Proceedings of the 18th ACM international conference on multimedia*, 2010 (pp. 1187–1190).
- Lin, C. (2007). On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transaction on Neural Networks*, 18(6), 1589–1596.
- Liu, X., Wang, L., Zhang, J., Yin, J., & Liu, H. (2014). Global and local structure preservation for feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 25(6), 1083–1095.
- Liu, H., Wu, Z., Li, X., Cai, D., & Huang, T. (2012). Constrained non-negative matrix factorization for image representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7), 1299–1311.
- Lu, Y., Lai, Z., & Fan, Z. (2015). Manifold discriminant regression learning for image classification. *Neurocomputing*, 166, 475–486.
- Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 395–416.
- Ma, T., Wang, Y., Tang, M., Cao, J., Tian, Y., Al-Dhelaan, A., et al. (2016). LED: A fast overlapping communities detection algorithm based on structural clustering. *Neurocomputing*, 207, 488–500.
- Mao, Q., & Tsang, I. (2013). A feature selection method for multivariate performance measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9), 2051–2063.
- McLachlan, G. (2004). *Discriminant analysis and statistical pattern recognition* (Vol. 544). New York, NY: Wiley.
- Meng, Y., Shang, R., Jiao, L., Zhang, W., & Yang, S. (2018a). Dual-graph regularized non-negative matrix factorization with sparse and orthogonal constraints. *Engineering Applications of Artificial Intelligence*, 69, 24–35.
- Meng, Y., Shang, R., Jiao, L., Zhang, W., Yuan, Y., & Yang, S. (2018b). Feature selection based dual-graph sparse non-negative matrix factorization for local discriminative clustering. *Neurocomputing*, 290, 87–99.
- Mika, S., Rätsch, G., Weston, J., Schölkopf, B., & Müller, K. (1999). Fisher discriminant analysis with kernels. In: *Proceeding of IEEE neural networks for signal processing workshop (NNSP)*, 1999 (pp. 41–48).
- Mitra, P., Murthy, C., & Pal, S. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3), 301–312.
- Nie, F., Xu, D., Tsang, I. W., & Zhang, C. (2010). Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7), 1921–1932.
- Papadimitriou, C., & Steiglitz, K. (1998). *Combinatorial optimization: Algorithms and complexity*. Mineola, NY: Dover Publications.
- Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally embedding. *Science*, 290(5500), 2323–2326.
- Shang, R., Liu, C., Meng, Y., Jiao, L., & Stolkin, R. (2017). Nonnegative matrix factorization with rank regularization and hard constraint. *Neural Computation*, 29, 2553–2579.
- Shang, R., Zhang, Z., & Jiao, L. (2016a). Global discriminative-based nonnegative spectral clustering. *Pattern Recognition*, 55, 172–182.
- Shang, R., Zhang, Z., Jiao, L., Liu, C., & Li, Y. (2016b). Self-representation based dual-graph regularized feature selection clustering. *Neurocomputing*, 171, 1242–1253.
- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 888–905.
- Shi, C., & Ruan, Q. (2015). Hessian semi-supervised sparse feature selection based on $l_{2,1/2}$ -matrix norm. *IEEE Transactions on Multimedia*, 17(1), 16–28.
- Sikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of relief and relieff. *Machine Learning*, 53(1), 23–69.
- Smith, L. (2002). A tutorial on principal components analysis. *Cornell University*, 58(3), 219–226.
- Stolkin, R., Greig, A., Hodgetts, M., & Gilby, J. (2008). An EM/E-MRF algorithm for adaptive model based tracking in extremely poor visibility. *Image and Vision Computing*, 26(4), 480–495.
- Stolkin, R., Hodgetts, M., Greig, A., & Gilby, J. (2007). Extended Markov random fields for predictive image segmentation. In *Proceedings of the 6th international conference on advances in pattern recognition*, 2007.

- Strehl, A., & Ghosh, J. (2003). Cluster ensembles: A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3, 583–617.
- Tenenbaum, J., Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2319–2323.
- Tian, Q., & Chen, S. (2017). Cross-heterogeneous-database age estimation through correlation representation learning. *Neurocomputing*, 238, 286–295.
- Wei, D., Li, S., & Tan, M. (2012). Graph embedding based feature selection. *Neurocomputing*, 93(2), 115–125.
- Xu, Z., King, I., Lyu, M., & Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transaction on Neural Networks*, 21(7), 1033–1047.
- Yan, S., & Xu, D. (2007). Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1), 40–51.
- Yang, Y., Shen, H., Ma, Z., Huang, Z., & Zhou, X. (2011). $l_{2,1}$ -norm regularized discriminative feature selection for unsupervised learning. In *Proceedings of the twenty-second international joint conference on artificial intelligence*, 2011 (pp. 1589–1594).
- Yang, Y., Xu, D., Nie, F., Yan, S., & Zhuang, Y. (2010). Image clustering using local discriminant models and global integration. *IEEE Transactions on Image Processing*, 19(10), 2761–2773.
- Zhao, Z., & Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In: *Proceedings of the 24th International Conference on Machine learning*, 2007 (pp. 1151–1157).
- Zhao, Z., Wang, L., & Liu, H. (2010). Efficient spectral feature selection with minimum redundancy. In *Proceedings of the 24th AAAI conference on artificial intelligence*, 2010 (pp. 673–678).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.