Neural Computing and Applications (2020) 32:6443–6454
https://doi.org/10.1007/s00521-018-3853-y

MULTI-SOURCE DATA UNDERSTANDING (MSDU)

Unsupervised nonlinear feature selection algorithm via kernel function

Jiaye Li[1] · Shichao Zhang[1] · Leyuan Zhang[1] · Cong Lei[1] · Jilian Zhang[2]

Received: 29 June 2018 / Accepted: 26 October 2018 / Published online: 8 November 2018

Abstract
Feature selection is one of the important methods of data preprocessing, but the general feature selection algorithm has the following shortcomings: (1) Noise and outliers cannot be ruled out so that the algorithm does not work well. (2) They only consider the linear relationship between data without considering the nonlinear relationship between data. For this reason, an unsupervised nonlinear feature selection algorithm via kernel function is proposed in this paper. First, each data feature is mapped to a kernel space by a kernel function. In this way, nonlinear feature selection can be performed. Secondly, the low-rank processing of the kernel coefficient matrix is used to eliminate the interference of noise samples. Finally, the feature selection is performed through a sparse regularization factor in the kernel space. Experimental results show that our algorithm has better results than contrast algorithms.

Keywords Feature selection · Kernel function · Sparse regularization factor

1 Introduction

Nowadays, with the development of computer science and technology, information era is coming. As the emergence of big data and cloud computing, it brought a large number of high-dimensional data [1]. For various reasons, it is sometimes difficult to obtain a lot of data, and you will also encounter the problem of dimensional disaster when processing data [2]. In order to alleviate these problems, the typical data preprocessing method-feature selection has received more and more attention. Preprocessing the data through feature selection can improve the data quality [3], reduce the data dimension, and make the data mining algorithm achieve better results. Therefore, it is very necessary for a large number of high-dimensional data to find a subset that can represent the original data features well [4–6].

Feature selection includes linear feature selection and nonlinear feature selection. Linear feature selection represents a linear relationship between data, and then finds a subset that represents the original features [7]. In practical applications, data features may contain strong relationships [8, 9]. However, in low-dimensional space, these relationships are nonlinear and then lead to difficulties in mining, resulting in insufficient excavation. There are many previous feature selection algorithms [10–12], but they usually cannot represent the nonlinear relationship between data [13, 14]. For this reason, this paper first proposes a nonlinear feature selection with the kernel method. Specifically, this paper uses the kernel function to map each feature of the data to the high-dimensional space, so that the nonlinear relationship between them is linearly separable in the high-dimensional space. It considers the global information of the data (low-rank constraint) and the nonlinear relationship (through Gaussian kernel) for feature selection. A better feature selection algorithm is proposed, which is called the Unsupervised Nonlinear Feature Selection Algorithm via Kernel Function (KF_NFS).

Firstly, this paper processes the original data to obtain the kernel matrix by kernel function, which solves the limitation that can only perform linear feature selection. Secondly, in order to achieve the best feature selection

✉ Shichao Zhang
zhangsc@mailbox.gxnu.edu.cn

Jiaye Li
jiaye_ligxnu@126.com

[1] Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin 541004, Guangxi, People's Republic of China

[2] College of Cyber Security, Jinan University, Guangzhou 510000, China

model, we use original data itself to fit it. The kernel coefficient matrix is performed different sparsity of degree (using $\ell_{2,p}$-norm) and low-rank constraint (removing noise samples). Finally, a l_1-norm of vector is embedded to perform nonlinear feature selection. Because this algorithm considers the nonlinear relationship and global information of the data, it is better than the general linear feature selection method. The final result of the experiment shows that the feature selected by this algorithm can achieve better results in classification accuracy.

Our algorithm has the following advantages:

- No matter whether it is high-dimensional data or low-dimensional data, the Gaussian kernel function [i.e., $k(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2})$] is suitable. By adjusting its width σ, it can be found that it is usually better and more applicable than other kernel functions such as the linear kernel. At the same time, each feature of the data is mapped into a kernel matrix, so that the relationship between the features can be mined more thoroughly.
- Because the algorithm has a low-rank constraint on the kernel coefficient matrix and a nonlinear relationship between the features in the high-dimensional space, the unimportant features and noise samples are effectively removed, and the algorithm accuracy is improved. At the same time, we add a $\ell_{2,p}$-norm sparse regularization factor to the algorithm. By adjusting the value of p, we can remove unrelated features well.
- The objective function of this paper is optimized by the method of accelerated proximal gradient descent. The optimization algorithm is an accelerated gradient descent algorithm, which is much lower in time complexity than the general gradient descent algorithm. Our algorithm can quickly converge through it. At the same time, it can ensure that the objective function gradually decreases during each iterative solution process. Finally it obtains the optimal solution.

2 Related work

The kernel function was introduced into the field of machine learning long ago. It has promoted the development of SVM to some extent [15]. It was initially proposed to avoid the computational obstacles in high-dimensional data. There are many commonly used kernel functions. Since the gaussian kernel can implement a nonlinear mapping, and it has less parameters than a polynomial kernel, we use the Gaussian kernel in this paper.

Unsupervised learning was proposed early. With the development of unsupervised learning, it was applied to various fields in machine learning. It is also widely used in feature selection algorithms. For example, Shao et al. [16] proposed that online unsupervised multi-view feature selection. It improves feature selection by combining feature of different views while using consistency and complementarity. Shi et al. [17] proposed a robust spectral learning for unsupervised feature selection. The feature selection algorithm is made more stable by constructing the Laplacian matrix of the graph with local learning. In addition, the method of retaining similar data points is better than different data points; Wei et al. [18] proposed a unsupervised feature selection by preserving stochastic neighbors.

Semi-supervised learning has also been applied to feature selection algorithms, such as Chang et al. [19] proposed a convex formulation for semi-supervised multi-label feature selection. This algorithm is different from the traditional semi-supervised algorithm. By limiting the training sample tags, the algorithm can select more representative features and reduce the computational complexity. Jian et al. [20] proposed multi-label informed feature selection; the algorithm uses the tag's relevance to select partitioning features and guides feature selection by decomposing multiple tag information into a low-dimensional space. Feature selection based on global structure and local structure of data is a very novel method. For example, Liu et al. [21] proposed global and local structure preservation for feature selection. Another feature selection algorithm implements embedding learning and sparse regression at the same time, so that the effect is very obvious. For example, Hou et al. [22] proposed joint embedding learning and sparse regression—a framework for unsupervised feature selection. It combines embedded learning and sparse regression to work together.

In 1998, nonlinear feature selection had been proposed [23], which presents multiple techniques such as multidimensional scaling and Sammonia mapping in the same framework. But it does not use the kernel function for nonlinear learning, and the effect is not the best. Min et al. [24] proposed a deep nonlinear feature mapping for large-margin KNN classification. This method uses the nonlinear mapping to improve the traditional KNN algorithm and achieves good results. This explains the value of nonlinear research to a certain extent. Jawanpuria et al. [25] proposed on p-norm path following multiple kernel learning for nonlinear feature selection. This method uses the p-norm to reduce the cost of optimization, compared to other path-based algorithms, which reduces training time. We have also consulted a lot of literature, and we will not list them here. The study of nonlinear learning has been around for a while; its research and development have great significance.

3 Our method

In this section, we first introduce the symbols used in this article and then explain our proposed KF_NFS algorithm, in Sects. 3.1 and 3.2, respectively, and then elaborate the proposed optimization method in Sect. 3.3. Finally, we analyze the convergence of the objective function in Sect. 3.4.

3.1 Notations

For the data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, the i-th row and the j-th column are denoted as $\mathbf{X}^i$ and $\mathbf{X}_j$ respectively, and the elements of the i-th row and the j-th column are denoted as $x_{i,j}$. The trace of the matrix $\mathbf{X}$ is denoted by $tr(\mathbf{X})$, $\mathbf{X}^{\mathrm{T}}$ denotes the transpose of the matrix $\mathbf{X}$, and $\mathbf{X}^{-1}$ represents the inverse of the matrix $\mathbf{X}$. We denote the $l_{2,p}$-norm of a matrix $\mathbf{X}$ and l_1-norm of a vector as $\|\mathbf{X}\|_1 = \sum_{j=1}^{d} |x_j|$, $\|\mathbf{X}\|_{2,p} = [\sum_{i=1}^{n} (\sum_{j=1}^{d} |X_{ij}|^2)^{p/2}]^{1/p}$.

3.2 KF_NFS algorithm

Assume a given sample data set $\mathbf{X} \in \mathbb{R}^{n\times d}$, where n and d represent the number of samples and the number of features, respectively. Here we divide the d-dimensional data matrix into d matrices, each of which is a matrix $\mathbf{X}_i \in \mathbb{R}^{n\times 1}, i = 1, \ldots, d$. Then each element of $\mathbf{X}_i$ is treated as an independent sample or feature $x_{ij} \in \mathbb{R}, j = 1, \ldots, n$. $\mathbf{X}_i$ is converted into the kernel matrix $\mathbf{K}^{(i)} \in \mathbb{R}^{n\times n}$ by projecting it into the heat kernel space:

$$\mathbf{K}^{(i)} = \begin{matrix} k(x_{i1}, x_{i1})k(x_{i1}, x_{i2})\ldots k(x_{i1}, x_{in}) \\ k(x_{i2}, x_{i1})k(x_{i2}, x_{i2})\ldots k(x_{i2}, x_{in}) \\ \ldots\ldots\ldots\ldots \\ k(x_{in}, x_{i1})k(x_{in}, x_{i2})\ldots k(x_{in}, x_{in}) \end{matrix} \tag{1}$$

The unsupervised feature selection algorithm aims to mine more representative features in the data, thus paving the way for the next experiment. In the absence of the class label $\mathbf{Y}$, using the data matrix $\mathbf{X}$ as a response matrix can better preserve the internal structure of the data's original features [26]. Since there is a linear relationship between features and features, there is also a nonlinear relationship. Therefore, the algorithm first converts the data matrix $\mathbf{X}$ into d kernel matrices $\mathbf{K}^{(i)}, i = 1, \ldots, d$ through a Gaussian kernel function. In order to fully exploit the nonlinear relationship between features, get the following formula:

$$\mathbf{X} = \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{W} \tag{2}$$

where: $\mathbf{W} \in \mathbb{R}^{n\times d}$ denotes the kernel coefficient matrix, $\boldsymbol{\alpha} \in \mathbb{R}^{d\times 1}$ is used to perform feature selection and is equivalent to the feature weight vector; α_i corresponds to an element of the vector $\boldsymbol{\alpha}$, $\mathbf{K}^{(i)} \in \mathbb{R}^{n\times n}$ is the kernel matrix. In order to make $\mathbf{X}$ get a better fitting effect, people usually use the l_F-norm to detect residuals, and minimizing the residuals can better fit the data, that is:

$$\min_{\boldsymbol{\alpha},\mathbf{W}} \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{W} \right\|_F^2 \tag{3}$$

Simultaneously, in order to reduce the amount of calculation, and exclude noise samples, the following low-rank [27] constraint is applied to the kernel coefficient matrix $\mathbf{W}$:

$$\min_{\boldsymbol{\alpha},\mathbf{W}} \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{W} \right\|_F^2 \text{ s.t. } \operatorname{rank}(\mathbf{W}) \le \min(n, d) \tag{4}$$

From formula (4), we can easily see that low rank reduces the amount of computation. And in real life, if the data is noisy or outliers, it will increase the rank of the matrix of kernel coefficients [28]. The low rank indicates a degree of redundancy. We use low-rank constraint to remove noise to a certain extent and filter out some outliers. Therefore, low-rank constraints are very useful. The kernel coefficient matrix can be expressed as the product of two matrices whose rank is not greater than r, ie:

$$\mathbf{W} = \mathbf{AB} \tag{5}$$

where: $\mathbf{A} \in \mathbb{R}^{n\times r}$, $\mathbf{B} \in \mathbb{R}^{r\times d}$. Substituting Eq. (5) into Eq. (4), we can get the general form of the low rank nonlinear feature selection model:

$$\min_{\mathbf{A},\ \mathbf{B},\boldsymbol{\alpha}} \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{AB} \right\|_F^2 \tag{6}$$

At the same time, in order to improve the accuracy of nonlinear feature selection, we further optimize the above equation. That is, row sparse is performed by using $l_{2,1}$-norm to substitute the l_F-norm in the above formula to punish all the row coefficients of $\left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{AB} \right\|_F^2$. However, in practice, it is shown that the $l_{2,p}$-norm can be adjusted to p to achieve better results [29], so we use the $l_{2,p}$-norm to decompress the kernel coefficient matrix $\mathbf{AB}$, that is:

$$\min_{\mathbf{A},\mathbf{B},\boldsymbol{\alpha}} \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{AB} \right\|_{2,p} \tag{7}$$

where $0<p<2$, when $p=1$, it is the standard $l_{2,1}$-norm. When we change the value of p, different sparse structures can be implemented for the matrix $\mathbf{AB}$. Since the objective function is a convex function, we make $\mathbf{P} = \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)}$, and it is easy to know that the solution is: $\mathbf{AB} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{P}\right)^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{X}$. But, in reality, $\mathbf{P}^{\mathrm{T}}\mathbf{P}$ is not necessarily reversible, so we introduce a $l_{2,p}$-norm regularization term to make it invertible, and reject the unimportant features in the data. In addition, we also do a orthogonal restrictions for $\mathbf{B}$, and achieve a better fitting effect. At the same time, we introduce a l_1-norm of $\boldsymbol{\alpha}$ to select data features in the kernel space. In summary, the final objective function of this paper is:

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B},\boldsymbol{\alpha}} & \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{AB} \right\|_{2,p} + \lambda_1 \|\mathbf{A}\|_{2,p} + \lambda_2 \|\boldsymbol{\alpha}\|_1 \\ & \text{s.t. } \mathbf{BB}^{\mathrm{T}} = \mathbf{I}_r \end{aligned} \tag{8}$$

where $\|\mathbf{A}\|_{2,p} = \left[\sum_{i=1}^{n} \left(\sum_{j=1}^{r} \left|A_{ij}\right|^2\right)^{p/2}\right]^{1/p}$, λ_1 and λ_2 are nonnegative adjustment parameters. Orthogonal constraint conditions $\mathbf{BB}^{\mathrm{T}} = \mathbf{I}_r \in \mathbb{R}^{r\times r}$ ensure that the low rank can be studied by considering the relationship between the outputs [30], thus improving the classification accuracy. Different degrees of sparsity are applied to the coefficient matrix $\mathbf{A}$ by the $l_{2,p}$-norm, it optimize the entire low-rank nonlinear feature selection model. The kernel matrix $\mathbf{K}$ is calculated based on the Gaussian kernel function. By mapping data features to high-dimensional kernel space, the nonlinear relationship between data features is represented in high-dimensional space. This can fully consider the nonlinear relationship between data features, so that the mining of data features more thorough. The last l_1-norm of $\boldsymbol{\alpha}$ is sparse for $\boldsymbol{\alpha}$, and nonlinear feature selection is made at the same time. If the value of the corresponding element of the vector $\boldsymbol{\alpha}$ is zero, we won't select the feature.

3.3 Optimization

Since the $l_{2,p}$-norm and l_1-norm are used in the objective function, the objective function cannot be closed-form solution. Therefore, this paper proposes an alternative iterative optimization method to solve this problem. Specifically divided into the following three steps.

3.3.1 Update A by fixing α and B

When $\boldsymbol{\alpha}$ and $\mathbf{B}$ are fixed, the optimization (8) problem becomes:

$$\min_{\mathbf{A}} \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{AB} \right\|_{2,p} + \lambda_1 \|\mathbf{A}\|_{2,p} \tag{9}$$

We make $\mathbf{P} = \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)}$, then the (9) formula can be transformed into: $\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{PAB}\|_{2,p} + \lambda_1 \|\mathbf{A}\|_{2,p}$, since matrix $\mathbf{B}$ has orthogonal constraint $\mathbf{BB}^{\mathrm{T}} = \mathbf{I}$, we have the following formula:

$$\begin{aligned} \|\mathbf{X} - \mathbf{PAB}\|_{2,p} &= \left\|(\mathbf{X} - \mathbf{PAB})(\mathbf{B}^{\mathrm{T}}, \mathbf{B}^{''})\right\|_{2,p} \\ &= \left\|\mathbf{XB}^{\mathrm{T}} - \mathbf{PA}\right\|_{2,p} + \left\|\mathbf{XB}^{''}\right\|_{2,p} \end{aligned} \tag{10}$$

The matrix $\mathbf{A}$ is not included in (10) $\left\|\mathbf{XB}^{''}\right\|_{2,\mathbf{p}}$, so when $\boldsymbol{\alpha}$ and $\mathbf{B}$ are fixed, the optimization of the objective function (8) can be converted to:

$$\min_{\mathbf{A}} \left\|\mathbf{XB}^{\mathrm{T}} - \mathbf{PA}\right\|_{2,p} + \lambda_1 \|\mathbf{A}\|_{2,p} \tag{11}$$

Further inference of the objective function (11) is available:

$$\min_{\mathbf{A}} tr\left[\left(\mathbf{XB}^{\mathrm{T}} - \mathbf{PA}\right)^{\mathrm{T}} \mathbf{Q} \left(\mathbf{XB}^{\mathrm{T}} - \mathbf{PA}\right)\right] + \lambda_1 tr\left(\mathbf{A}^{\mathrm{T}}\mathbf{NA}\right) \tag{12}$$

where λ_1 is the tuning parameter, $\mathbf{Q} \in \mathbb{R}^{n\times n}$ and $\mathbf{N} \in \mathbb{R}^{n\times n}$ are both diagonal matrices and their main diagonal elements are: $Q_{ii} = \frac{1}{\frac{2}{p}\left\|(\mathbf{XB}^{\mathrm{T}} - \mathbf{PA})_i\right\|_2^{2-p}} i = (1, 2, \ldots, n)$, $N_{jj} = \frac{1}{\frac{2}{p}\left\|\mathbf{A}_j\right\|_2^{2-p}} j = (1, 2, \ldots, n)$.

By setting the derivative of $\mathbf{A}$ in (12) to zero, we obtain:

$$\mathbf{A} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{QP} + \lambda_1 \mathbf{N}\right)^{-1} \mathbf{P}^{\mathrm{T}}\mathbf{QXB}^{\mathrm{T}} \tag{13}$$

3.3.2 Update B by fixing α and A

By fixing $\boldsymbol{\alpha}$ and $\mathbf{A}$, the objective function (8) can be simplified as follows:

$$\min_{\mathbf{B}} \left\|\mathbf{X} - \hat{\mathbf{P}}\mathbf{B}\right\|_{2,p}, \quad \text{s.t.}, \mathbf{BB}^{\mathrm{T}} = \mathbf{I}_r \tag{14}$$

where $\hat{\mathbf{P}} = \mathbf{PA} \in \mathbb{R}^{n\times r}$. It is easy to know that the objective function (14) is actually an orthogonal procrustes problem [31]. $\mathbf{X}^{\mathrm{T}}\hat{\mathbf{P}} = \mathbf{USV}^{\mathrm{T}}$ is performed directly singular value decomposition. It can be seen that the optimal solution of subspace matrix $\mathbf{B}$ is $\mathbf{VU}^{\mathrm{T}}$, $\mathbf{U} \in \mathbb{R}^{d\times r}, \mathbf{V} \in \mathbb{R}^{\mathbf{r}\times\mathbf{r}}$.

3.3.3 Update α by fixing A and B

After fixing $\mathbf{A}$, $\mathbf{B}$, the objective function (8) becomes:

$$\min_{\boldsymbol{\alpha}} \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{AB} \right\|_{2,p} + \lambda_2 \|\boldsymbol{\alpha}\|_1 \tag{15}$$

Here's a simple simplification:

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{AB} \right\|_{2,p} = \min_{\boldsymbol{\alpha}} \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{Z}^{(i)} \right\|_{2,p} \\
&\Leftrightarrow \min_{\boldsymbol{\alpha}} \left[\left(\mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{Z}^{(i)} \right)^{\mathrm{T}} \mathbf{Q} \left(\mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{Z}^{(i)} \right) \right] \\
&\Leftrightarrow \min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} tr \left[\left(\mathbf{X}_i - \sum_{i=1}^{d} \alpha_i \mathbf{Z}_i^{(i)} \right)^{\mathrm{T}} Q_{ii} \left(\mathbf{X}_i - \sum_{i=1}^{d} \alpha_i \mathbf{Z}_i^{(i)} \right) \right] \\
&\Leftrightarrow \min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} Q_{ii} tr \left(\mathbf{X}_i - \sum_{i=1}^{d} \alpha_i \mathbf{Z}_i^{(i)} \right)^{\mathrm{T}} \left(\mathbf{X}_i - \sum_{i=1}^{d} \alpha_i \mathbf{Z}_i^{(i)} \right) \\
&\Leftrightarrow \min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} Q_{ii} \left\| \mathbf{X}_i - \left(\alpha_1 Z_{i,\cdot}^{(1)} + \cdots + \alpha_d Z_{i,\cdot}^{(d)} \right) \right\|_2^2
\end{aligned} \tag{16}$$

We make $\mathbf{S}^{(i)} = \begin{pmatrix} Z_{i,1}^{(1)} & \cdots & Z_{i,d}^{(1)} \\ Z_{i,1}^{(d)} & \cdots & Z_{i,d}^{(d)} \end{pmatrix} \in \mathbb{R}^{d \times d}$, (16) formula is written the following form:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} Q_{ii} \left\| \mathbf{X}_i - \boldsymbol{\alpha}^{\mathrm{T}} \left(\mathbf{S}^{(i)} \right) \right\|_2^2 \tag{17}$$

After a series of simplifications, we can get (17) that is equivalent to the following formula:

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} Q_{ii} \mathbf{X}_i \mathbf{X}_i^{\mathrm{T}} - 2\boldsymbol{\alpha}^{\mathrm{T}} \sum_{i=1}^{n} Q_{ii} \mathbf{S}^{(i)} \mathbf{X}_i^{\mathrm{T}} \\
&+ \boldsymbol{\alpha}^{\mathrm{T}} \sum_{i=1}^{n} Q_{ii} \left(\mathbf{S}^{(i)} \left(\mathbf{S}^{(i)} \right)^{\mathrm{T}} \right) \boldsymbol{\alpha}
\end{aligned} \tag{18}$$

Since the objective function (15) is convex but not smooth, we design a new accelerated approximate gradient method to solve the function. We make:

$$\begin{aligned}
f(\boldsymbol{\alpha}) &= \left\| \mathbf{X} - \sum_{i=1}^{d} \alpha_i \mathbf{K}^{(i)} \mathbf{AB} \right\|_{2,p} \\
F(\boldsymbol{\alpha}) &= f(\boldsymbol{\alpha}) + \lambda_2 \|\boldsymbol{\alpha}\|_1
\end{aligned} \tag{19}$$

Notice $\|\boldsymbol{\alpha}\|_1$ is convex but not smooth. So using the approximate gradient method, we can use the following rules to update iterations $\boldsymbol{\alpha}$.

$$\begin{aligned}
\boldsymbol{\alpha}_{t+1} &= \arg\min_{\boldsymbol{\alpha}} G_{\eta t}(\boldsymbol{\alpha}, \boldsymbol{\alpha}_t) \\
G_{\eta t}(\boldsymbol{\alpha}, \boldsymbol{\alpha}_t) &= f(\boldsymbol{\alpha}_t) + \langle \nabla f(\boldsymbol{\alpha}_t), \boldsymbol{\alpha} - \boldsymbol{\alpha}_t \rangle \\
&\quad + \frac{\eta_t}{2} \|\boldsymbol{\alpha} - \boldsymbol{\alpha}_t\|^2 + \lambda_2 \|\boldsymbol{\alpha}\|_1
\end{aligned} \tag{20}$$

Here, $\nabla f(\boldsymbol{\alpha}_t) = 2\boldsymbol{\alpha}_t^{\mathrm{T}} \sum_{i=1}^{n} Q_{ii} (\mathbf{S}^{(i)} (\mathbf{S}^{(i)})^{\mathrm{T}}) - 2 \sum_{i=1}^{n} Q_{ii} \mathbf{X}_i (\mathbf{S}^{(i)})^{\mathrm{T}}$ is calculated from (18), η_t is a tuning parameter, $\boldsymbol{\alpha}_t$ is a value in the t-th iteration.

By ignoring the independent $\boldsymbol{\alpha}$ in formula (20), we can get:

$$\boldsymbol{\alpha}_{t+1} = \pi_{\eta_t}(\boldsymbol{\alpha}_t) = \arg\min_{\boldsymbol{\alpha}} \frac{1}{2} \|\boldsymbol{\alpha} - \mathbf{U}_t\|_F^2 + \frac{\lambda_2}{\eta_t} \|\boldsymbol{\alpha}\|_1 \tag{21}$$

where $\mathbf{U}_t = \boldsymbol{\alpha}_t - \frac{1}{\eta_t} \nabla f(\boldsymbol{\alpha}_t)$, $\pi_{\eta_t}(\boldsymbol{\alpha}_t)$ is a Euclidean projection on a convex set η_t, because $\|\boldsymbol{\alpha}\|_1$ has a separable form, (21) can be written as follows:

$$\alpha_{t+1}^i = \arg\min_{\alpha^i} \frac{1}{2} \left\| \alpha^i - U_t^i \right\|_2^2 + \frac{\lambda_2}{\eta_t} \left| \alpha^i \right| \tag{22}$$

where α^i and α_{t+1}^i are respectively the i-th elements of α and α_{t+1}, then according to formula (22), α_{t+1}^i can be obtained the following closed-form solution:

$$\alpha^{i*} = \begin{cases} u_t^i - \frac{\lambda_2}{\eta_t} \times \mathrm{sign}(u_t^i), & \text{if} \left\| u_t^i \right\| > \frac{\lambda_2}{\eta_t} \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

To speed up the approximate gradient algorithm in Eq. (20), we added an auxiliary variable:

$$\mathbf{V}_{t+1} = \boldsymbol{\alpha}_t + \frac{\beta_t - 1}{\beta_{t+1}} (\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_t) \tag{24}$$

where $\beta_{t+1} = \frac{1 + \sqrt{1 + 4\beta_t^2}}{2}$.

3.4 Convergence analysis

We can prove that the value of the objective function (11) is monotonically decreasing in every iteration. The objective function is equivalent to:

$$\min_{\mathbf{A}} tr \left[\left(\mathbf{XB}^{\mathrm{T}} - \mathbf{PA} \right)^{\mathrm{T}} \mathbf{Q} \left(\mathbf{XB}^{\mathrm{T}} - \mathbf{PA} \right) \right] + \lambda_1 tr \left(\mathbf{A}^{\mathrm{T}} \mathbf{NA} \right) \tag{25}$$

So we have:

$$\begin{aligned}
&tr \left[\left(\mathbf{XB}_{(t+1)}^{\mathrm{T}} - \mathbf{PA}_{(t+1)} \right)^{\mathrm{T}} \mathbf{Q}_t \left(\mathbf{XB}_{(t+1)}^{\mathrm{T}} - \mathbf{PA}_{(t+1)} \right) \right] \\
&\quad + \lambda_1 tr \left(\mathbf{A}_{(t+1)}{}^{\mathrm{T}} \mathbf{N}_t \mathbf{A}_{(t+1)} \right) \\
&\leq tr \left[\left(\mathbf{XB}_t^{\mathrm{T}} - \mathbf{PA}_t \right)^{\mathrm{T}} \mathbf{Q}_t \left(\mathbf{XB}_t^{\mathrm{T}} - \mathbf{PA}_t \right) \right] \\
&\quad + \lambda_1 tr \left(\mathbf{A}_t^{\mathrm{T}} \mathbf{N}_t \mathbf{A}_t \right)
\end{aligned} \tag{26}$$

According to the simple formula reasoning, we can get the following formula:

$$\Rightarrow \sum_{i=1}^{n} \frac{\left\| \mathbf{X}^i \mathbf{B}_{(t+1)}{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_{(t+1)} \right\|_2^{2(2-p)}}{(2/p)\left\| \mathbf{X}^i \mathbf{B}_t{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_t \right\|_2^{2-p}} + \lambda_1 \sum_{i=1}^{n} \frac{\left\| a^i{}_{(t+1)} \right\|_2^{2(2-p)}}{(2/p)\left\| a^i{}_t \right\|_2^{2-p}}$$
$$\leq \sum_{i=1}^{n} \frac{\left\| \mathbf{X}^i \mathbf{B}_t{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_t \right\|_2^{2(2-p)}}{(2/p)\left\| \mathbf{X}^i \mathbf{B}_t{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_t \right\|_2^{2-p}} + \lambda_1 \sum_{i=1}^{n} \frac{\left\| a^i{}_t \right\|_2^{2(2-p)}}{(2/p)\left\| a^i{}_t \right\|_2^{2-p}} \tag{27}$$

The above can indicate that any nonzero vector in (10) has:

$$\sum_i \left\| a^i{}_{(t+1)} \right\|_2^{2(2-p)} - \sum_i \frac{\|a^i{}_{t+1}\|_2^{2(2-p)}}{(2/p)\|a^i{}_{t+1}\|_2^{2-p}} \leq \sum_i \left\| a^i{}_t \right\|_2^{2-p} - \sum_i \frac{\|a^i{}_t\|_2^{2(2-p)}}{(2/p)\|a^i{}_t\|_2^{2-p}}$$

$$\sum_{i=1}^{n} \left\| \mathbf{X}^i \mathbf{B}_{(t+1)}{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_{(t+1)} \right\|_2^{2-p} - \sum_{i=1}^{n} \frac{\left\| \mathbf{X}^i \mathbf{B}_{(t+1)}{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_{(t+1)} \right\|_2^{2(2-p)}}{(2/p)\left\| \mathbf{X}^i \mathbf{B}_t{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_t \right\|_2^{2-p}}$$
$$\leq \sum_{i=1}^{n} \left\| \mathbf{X}^i \mathbf{B}_t{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_t \right\|_2^{2-p} - \sum_{i=1}^{n} \frac{\left\| \mathbf{X}^i \mathbf{B}_t{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_t \right\|_2^{2(2-p)}}{(2/p)\left\| \mathbf{X}^i \mathbf{B}_t{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_t \right\|_2^{2-p}} \tag{28}$$

To sum up, we can easily get:

$$\sum_{i=1}^{n} \left\| \mathbf{X}^i \mathbf{B}_{(t+1)}{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_{(t+1)} \right\|_2^{2-p} + \lambda_1 \sum_{i=1}^{n} \left\| a^i{}_{t+1} \right\|_2^{2-p} \leq \sum_{i=1}^{n} \left\| \mathbf{X}^i \mathbf{B}_t{}^{\mathrm{T}} - \mathbf{P}^i \mathbf{A}_t \right\|_2^{2-p} + \lambda_1 \sum_{i=1}^{n} \left\| a^i{}_t \right\|_2^{2-p} \tag{29}$$

Theorem 1 Let $\boldsymbol{\alpha}_t$ be the sequence generated by Algorithm 1, then for $\forall t \geq 1$, (29) holds:

$$F(\boldsymbol{\alpha}_t) - F(\boldsymbol{\alpha}^*) \leq \frac{2\gamma L \|\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}^*\|_F^2}{(t+1)^2} \tag{30}$$

Algorithm 1: Pseudo code for solving (8).

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$, λ_1, λ_2, *p, and r;*
Output: $\mathbf{A}$,$\mathbf{B}$,$\boldsymbol{\alpha} \in R^{d \times 1}$;
Initialize *t*=1;
Randomly initialize $\mathbf{A}(0)$;;
Compute $\mathbf{K}^{(i)}$ by (1)formula ;
repeat
  Update $\mathbf{B}$ via (14);
  Compute $\mathbf{Q}(t+1)$ as $Q_{ii} = \frac{1}{\frac{2}{p}\left\| (\mathbf{X}\mathbf{B}^T - \mathbf{P}\mathbf{A})_i \right\|_2^{2-p}} i = (1, 2, \ldots, n)$;
  Compute $\mathbf{N}(t+1)$ as $\mathrm{N}_{jj} = \frac{1}{\frac{2}{p}\|\mathbf{A}_j\|_2^{2-p}} j = (1, 2, \ldots, n)$;
  Update $\mathbf{A}(t+1)$ via (9) ;
  Update $\boldsymbol{\alpha}$ according to the following rules ;
  while $F(\boldsymbol{\alpha}_t) > G_{\eta t-1}(\pi_{\eta t-1}(\boldsymbol{\alpha}_t), \boldsymbol{\alpha}_t)$;
  Set $\eta_{t-1} = \gamma \eta_{t-1}$;
  end while ;
  Set $\eta_t = \gamma \eta_{t-1}$;
  Compute $\boldsymbol{\alpha}_{t+1} = \arg\min_{\boldsymbol{\alpha}} G_{\eta t}(\boldsymbol{\alpha}, V_t)$;
  Compute $\beta_{t+1} = \frac{1+\sqrt{1+4\beta_t{}^2}}{2}$;
  Compute formula (24) ;
  t = *t*+1;
until *converge*;

According to reference [32], γ is a constant defined in advance, L is the Lipschitz constant of the $f(\boldsymbol{\alpha})$ gradient in Eq. (19) , and $\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} F(\boldsymbol{\alpha})$.

Through the above inequality and Theorem 1, we can easily see that our algorithm is convergent.

4 Experiments

In this section, we compare the KF_NFS algorithm with the comparison algorithms. The dimensionality is reduced by the feature selection algorithms, and then, the data after dimension reduction is conducted SVM classified. Finally, the performance of the algorithms is measured according to the classification accuracy.

4.1 Experiment settings

We tested our proposed nonlinear feature selection algorithm with five binary-class data sets and seven multi-class data sets. They are Glass, SPECTF, Sonar, Clean, Arrhythmia, Movements, Ecoli, Urban_land, Ionosphere, Yale, Colon, Lung_discrete, where the first nine are all from UCI Machine Learning Repository[1] and the last three are from feature selection data.[2] The details of the data set are shown in Table 1.

Table 1 The information of the data sets

Datasets	Samples	Dimensions	Classes
Glass	214	9	6
Movements	360	90	15
SPECTF	267	44	2
Ecoli	336	343	8
Sonar	208	60	2
Urban_land	168	147	9
Clean	476	167	2
Ionosphere	351	34	2
Arrhythmia	452	279	13
Colon	62	2000	2
Yale	165	1024	15
Lung_discrete	73	325	7

At the same time, we found eight comparison algorithms to compare with the KF_NFS algorithm. The information of the comparison algorithm is as follows:

RSR: It constrains the self-representation coefficient matrix by a $l_{2,1}$-norm, so that the representative features are selected and the robustness of outliers are ensured [33].

SOGFS: It is an embedded unsupervised feature selection algorithm. It introduces local constraints on manifold structure learning by the reasonable constraints. It also performs feature selection and local structure learning simultaneously to select more valuable features [34].

EUFS: It is an unsupervised feature selection algorithm. It uses sparse learning to embed the feature selection algorithm into the clustering algorithm so as to achieve a better feature selection effect [35].

FSASL: It is also an unsupervised feature selection algorithm that combines feature learning with structural learning. Two learning methods are mutually promoted to achieve good results [36].

RFS: It is a supervised feature selection algorithm. It combines the $l_{2,1}$-norm to limit the loss function and the regularization term, and achieves a very good robust effect [37].

LS: According to the distance of two data points, then they are likely to have many similar relationships. Calculating its Laplacian score to reflect the holding ability of the local structure. Finally achieve good feature selection effect [38].

NetFS: It is a robust unsupervised feature selection algorithm, which embeds potential representation learning into feature selection to mitigate the effects of noise and achieve good results [39].

RUFS: It is also a robust and unsupervised feature selection algorithm. It performs clustering and feature selection simultaneously, and reduces the time and space complexity of the algorithm [40].

In our proposed model, we set $\{\lambda_1, \lambda_2\} \in \{10^{-4}, \ldots, 10^8\}$, the rank of the kernel coefficient matrix $r \in \{1, \ldots, \min(n, d)\}$, and the parameter of $l_{2,p}$-norm $p \in \{0.1, \ldots, 1.9\}$. The parameters $c \in \{2^{-5}, \ldots, 2^5\}$ and $g \in \{2^{-5}, \ldots, 2^5\}$ are used to select the best SVM for classification, and distinguish different types of samples. The experiment divides the data set randomly into a training set and a test set through a tenfold cross-validation. In order to maintain fairness, we carry out tenfold cross-validation of 10 times, and finally we take the average of the classification accuracy.

We use the classification accuracy and standard deviation as the evaluation criteria for our experiments. We define the classification accuracy as follows:

$$\text{acc} = X_{\text{correct}}/X \tag{31}$$

where X represents the total number of samples and X_{correct} represents the correct number of samples for classification. At the same time we define the standard deviation to measure the stability of our algorithm, as follows:

[1] http://archive.ics.uci.edu/ml.

[2] http://featureselection.asu.edu/datasets.php.

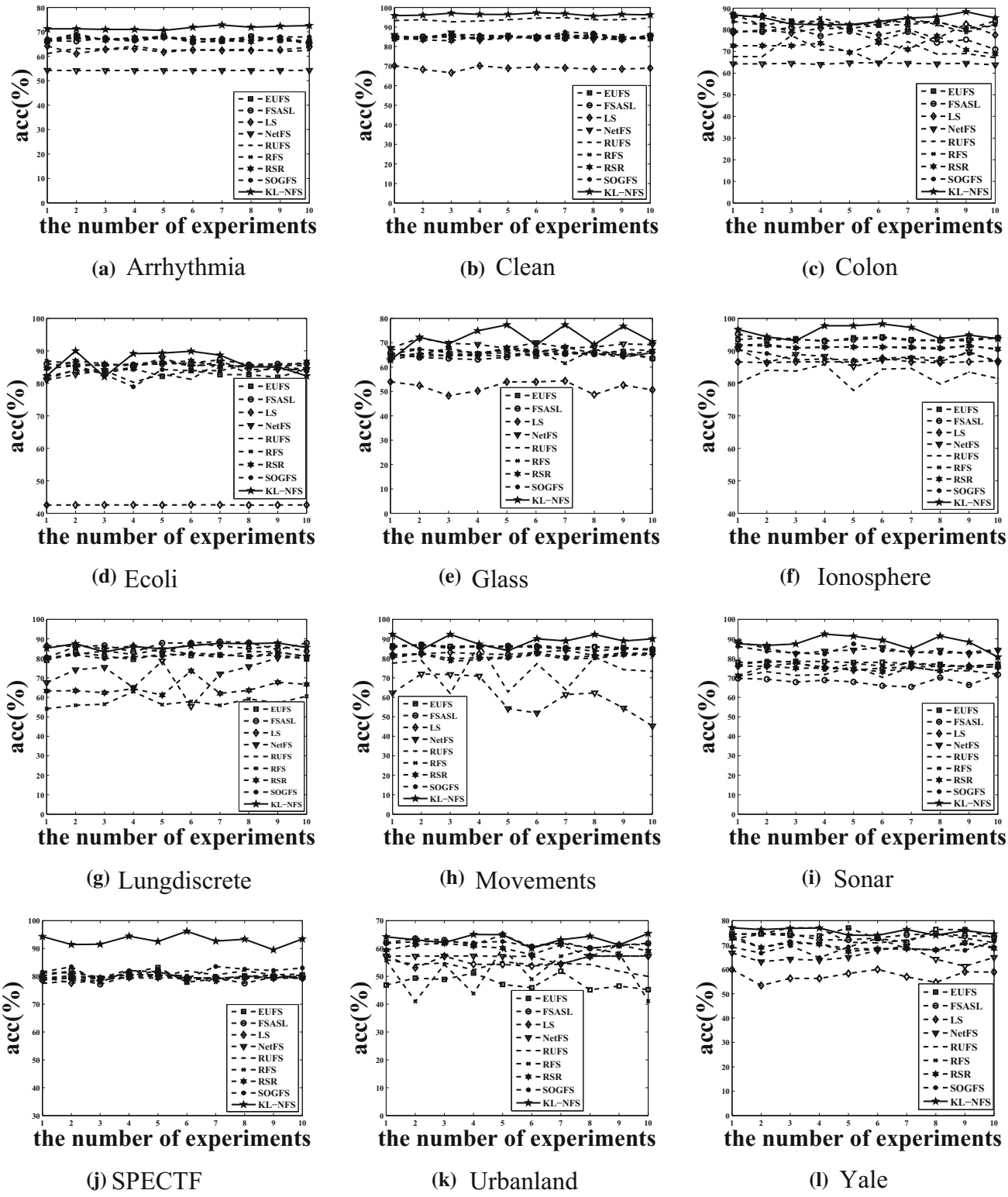

Fig. 1 Average classification accuracy of all methods for all datasets

Table 2 Average classification accuracy [acc (%)]

Datasets	EUFS	FSASL	LS	NetFS	RUFS	RFS	RSR	SOGFS	KF_NFS
Glass	65.14	64.62	51.90	69.12	66.40	64.97	66.02	66.11	**71.97**
Movements	85.83	85.75	82.50	60.61	73.75	80.56	81.36	85.44	**89.00**
SPECTF	80.26	79.29	79.43	79.63	80.31	80.14	79.40	81.69	**92.93**
Ecoli	83.15	86.01	42.56	84.47	83.29	85.83	85.81	83.79	**86.36**
Sonar	77.52	68.27	75.68	84.23	72.72	76.70	74.44	83.96	**87.96**
Urban_land	47.81	62.05	55.61	56.96	53.75	52.06	60.01	61.72	**63.36**
Clean	84.61	84.83	68.88	84.69	93.76	84.77	84.64	85.18	**96.41**
Ionosphere	93.61	93.67	86.69	87.70	82.52	91.11	91.22	88.10	**95.72**
Arrhythmia	66.91	66.71	62.65	54.20	62.61	67.13	66.8	67.19	**71.65**
Colon	82.88	77.07	79.24	64.38	69.79	82.48	72.29	83.62	**84.90**
Yale	74.30	73.69	57.36	65.04	73.32	69.76	69.46	68.81	**75.60**
Lung_discrete	81.059	86.07	85.59	72.45	82.20	57.61	64.79	81.36	**86.18**
Average value	76.92	77.34	69.01	71.96	74.54	74.43	74.69	78.08	**83.50**

Bold values indicate highest classification accuracy

$$\text{std} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{acc}_i - \mu)^2} \tag{32}$$

where N represents the number of experiments, acc_i represents the classification accuracy of the i-th experiment, μ represents the average classification accuracy, and the smaller the std, the more stable the representative algorithm.

4.2 Experiment results and analysis

In Fig. 1, the classification accuracy of 10 experiments for all algorithms is shown. To avoid the randomness of the training set and the test set, we use tenfold cross-validation to divide the data into training and test sets. At the same time, the average of 10 results is used to evaluate the accuracy of the algorithm. In this way, ten experiments were finally carried out. From Fig. 1, we can clearly see that our proposed algorithm has the highest classification accuracy in most cases. In Table 2, we show the results of all algorithms on 12 datasets. It can be seen that the KF_NFS algorithm has the highest classification accuracy compared with the other eight algorithms. Specifically, it is 6.58% higher than the EUFS algorithm on average, 14.49% higher than the LS algorithm on average, and 9.07% higher than the RFS algorithm on average. It shows that our algorithm is better than the general linear feature selection algorithm. Compared with the FSASL, NetFS, RUFS, RSR, and SOGFS algorithms, It increased by an average of 6.16%, 11.54%, 8.96%, 8.81%, and 5.42%, respectively.

In Fig. 2, the value of each iteration of our objective function over 12 data sets is shown. We set the condition for our algorithm to converge on $\frac{|\text{obj}(t+1)-\text{obj}(t)|}{\text{obj}(t)} \leq 10^{-5}$. From Fig. 2, we can easily see that the value of the objective function gradually decreases as the number of iterations increases and finally converges to a certain value. Moreover, our objective function converges very quickly, and most converge before the fifth iteration.

In Table 3, we can see that the average standard deviation of accuracy rate of our algorithm is the smallest. On the one hand, it shows that our algorithm is more stable, and on the other hand, it shows that its overall performance is better.

The KF_NFS algorithm can achieve good results. It is mainly related to the following two points: (a) considering the nonlinear relationship between data; (b) Iteratively performing low-rank feature selection steps. At the same time, according to the standard deviation, we can easily find that our proposed algorithm is the most stable.

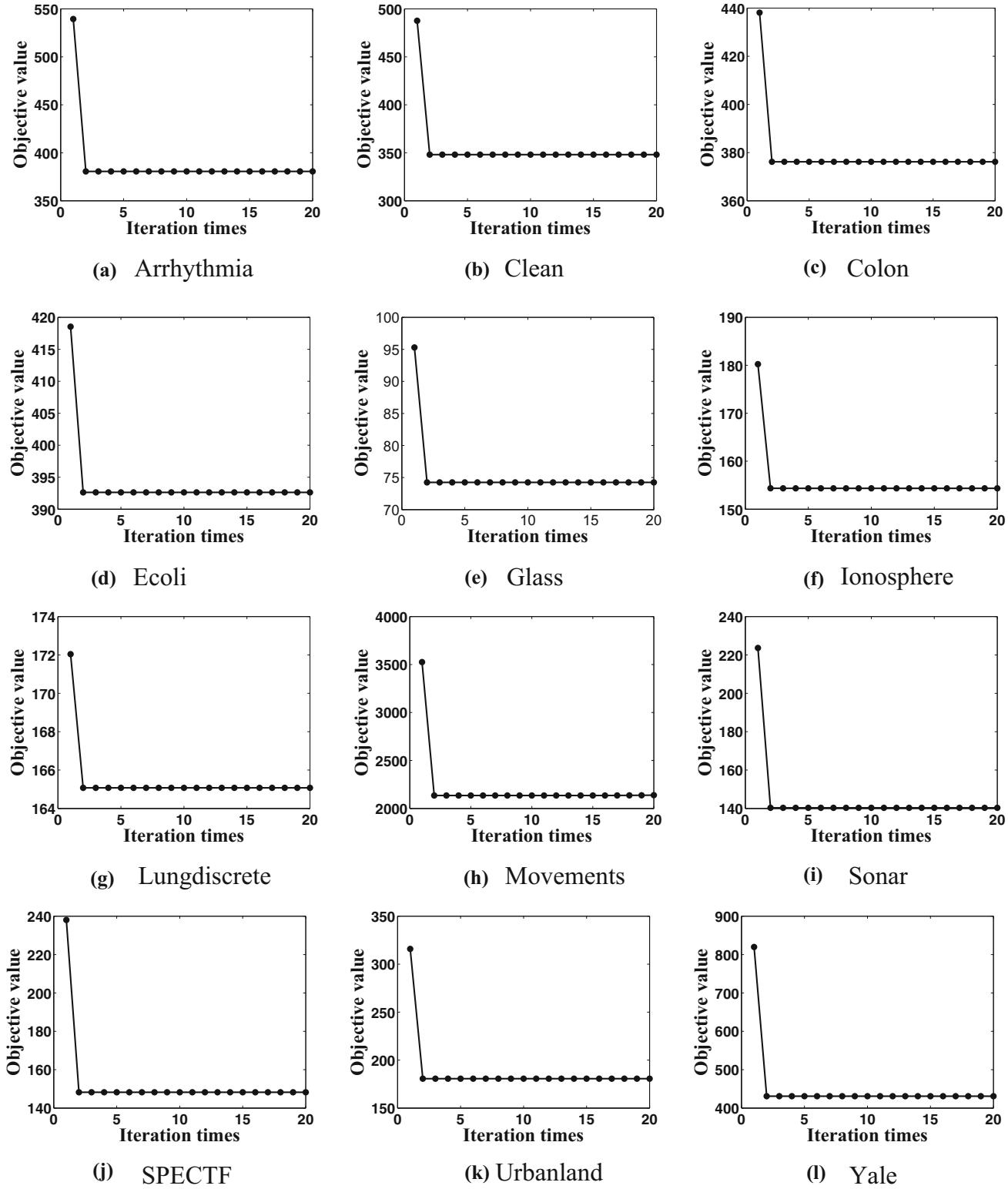

Fig. 2 Convergence rate of Algorithm 1 on all tested datasets

Table 3 Standard deviation of classification accuracy [std (%)]

Datasets	EUFS	FSASL	LS	NetFS	RUFS	RFS	RSR	SOGFS	KF_NFS
Glass	**0.82**	1.03	2.16	1.25	0.86	1.54	1.52	1.23	1.45
Movements	0.69	0.62	0.75	8.66	7.99	1.25	1.04	1.01	**0.33**
SPECTF	1.48	1.29	0.78	0.57	1.52	0.90	**0.03**	1.75	**0.03**
Ecoli	0.93	0.47	**0.03**	1.78	1.84	0.54	0.82	1.82	0.62
Sonar	**0.82**	1.90	0.85	1.76	1.87	1.26	1.42	1.58	1.21
Urban_land	2.27	1.38	1.69	0.88	1.73	7.30	1.57	**0.83**	1.67
Clean	0.54	0.94	0.99	1.08	0.64	0.74	0.97	1.26	**0.51**
Ionosphere	0.44	0.58	0.39	1.61	2.44	0.44	0.40	1.05	**0.25**
Arrhythmia	0.84	0.85	0.85	**0.02**	0.75	1.00	1.05	0.56	0.71
Colon	1.88	2.99	1.83	**0.29**	3.56	2.08	2.31	1.77	1.93
Yale	1.76	**0.95**	2.12	2.28	2.23	1.74	1.60	2.14	1.25
Lung_discrete	1.29	2.52	1.34	7.48	1.57	2.45	3.50	**0.95**	1.44
Average value	1.15	1.29	1.15	2.31	2.25	1.77	1.35	1.33	**0.95**

Bold values indicate lowest standard deviation

5 Conclusion

This paper proposes a new unsupervised nonlinear feature selection algorithm through the nonlinear relationship between data features. That is, using the kernel function, applying the $l_{2,p}$-norm to both the loss function and the regularization, and combining the low rank and the l_1-norm sparse methods are used to further refine the proposed model; therefore, it achieves very good results. The algorithm has discovered the nonlinear relationship between the data features and has a more significant mining effect than the general feature selection algorithm. The experimental results show that the algorithm of this paper has greatly improved the classification accuracy and stability. In the future work, we will attempt to combine more advanced theoretical for improvement algorithms.

Acknowledgements This work is partially supported by the China Key Research Program (Grant No. 2016YFB1000905); the Key Program of the National Natural Science Foundation of China (Grant No. 61836016); the Natural Science Foundation of China (Grant Nos. 61876046, 61573270, 81701780 and 61672177); the Project of Guangxi Science and Technology (GuiKeAD17195062); the Guangxi Natural Science Foundation (Grant Nos. 2015GXNSFCB139011 and 2017GXNSFBA198221); the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing; the Guangxi High Institutions Program of Introducing 100 High-Level Overseas Talents; and the Research Fund of Guangxi Key Lab of Multisource Information Mining and Security (18-A-01-01).

Compliance with ethical standards

Conflict of interest We wish to draw the attention of the Editor to the following facts which may be considered as potential conflicts of interest and to significant financial contributions to this work. We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. We understand that the corresponding author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from 2017010381@stu.gxnu.edu.cn.

References

1. Zhu X, Zhang S, He W, Hu R, Lei C, Zhu P (2018) One-step multi-view spectral clustering. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2018.2873378
2. Abualigah LM, Khader AT, Al-Betar MA, Alomari OA (2017) Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering. Exp Syst Appl Int J 84(C):24–36
3. Li Z, Tang J, Mei T (2018) Deep collaborative embedding for social image understanding. IEEE Trans Pattern Anal Mach Intell 1:1–1
4. Zheng W, Zhu X, Wen G, Zhu Y, Yu H, Gan J (2018) Unsupervised feature selection by self-paced learning regularization. Pattern Recogn Lett. https://doi.org/10.1016/j.patrec.2018.06.029
5. Zhu X, Zhang S, Li Y, Zhang J, Yang L, Fang Y (2018) Low-rank sparse subspace for spectral clustering. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2018.2858782
6. Zhu X, Li X, Zhang S, Zongben X, Litao Y, Wang C (2017) Graph PCA hashing for similarity search. IEEE Trans Multimedia 1:1–1
7. Kolhe S, Deshkar P (2017) Dimension reduction methodology using group feature selection. In: International conference on innovative mechanisms for industry applications, pp 789–791
8. Liimatainen K, Heikkil R, Yliharja O, Huttunen H, Ruusuvuori P (2015) Sparse logistic regression and polynomial modelling for detection of artificial drainage networks. Remote Sens Lett 6(4):311–320

9. Zhu X, Li X, Zhang S, Chunhua J, Xindong W (2017) Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE Trans Neural Netw Learn Syst 28(6):1263–1275
10. Gao L, Guo Z, Zhang H, Xu X, Shen HT (2017) Video captioning with attention-based LSTM and semantic consistency. IEEE Trans Multimedia 19(9):2045–2055
11. Hu R, Zhu X, Cheng D, He W, Yan Y, Song J, Zhang S (2016) Graph self-representation method for unsupervised feature selection. Neurocomputing 220:130–137
12. Song J, Shen HT, Wang J, Huang Z, Sebe N, Wang J (2016) A distance-computation-free search scheme for binary code databases. IEEE Trans Multimedia 18(3):484–495
13. Zhu X, Li X, Zhang S (2016) Block-row sparse multiview multilabel learning for image classification. IEEE Trans Cybern 46(2):450–461
14. Zhang S, Li X, Zong M, Zhu X, Wang R (2018) Efficient KNN classification with different numbers of nearest neighbors. IEEE Trans Neural Netw Learn Syst 29(5):1774–1785
15. Fei J, Zhao N, Shi Y, Feng Y, Wang Z (2016) Compressor performance prediction using a novel feed-forward neural network based on gaussian kernel function. Adv Mech Eng 8(1):1–1
16. Shao W, He L, Lu CT, Wei X, Yu PS (2016) Online unsupervised multi-view feature selection, pp 1203–1208. arXiv:1609.08286
17. Shi L, Du L, Shen YD (2015) Robust spectral learning for unsupervised feature selection. In: IEEE international conference on data mining, pp 977–982
18. Wei X, Yu PS (2016) Unsupervised feature selection by preserving stochastic neighbors. In: ACM sigspatial Ph.D. symposium, p 1
19. Chang X, Nie F, Yang Y, Huang H (2014) A convex formulation for semi-supervised multi-label feature selection. In: Twenty-eighth AAAI conference on artificial intelligence, pp 1171–1177
20. Jian L, Li J, Shu K, Liu H (2016) Multi-label informed feature selection. In: International joint conference on artificial intelligence, pp 1627–1633
21. Liu X, Wang L, Zhang J, Yin J, Liu H (2017) Global and local structure preservation for feature selection. IEEE Trans Neural Netw Learn Syst 25(6):1083–1095
22. Hou C, Nie F, Li X, Yi D, Yi W (2017) Joint embedding learning and sparse regression: a framework for unsupervised feature selection. IEEE Trans Cybern 44(6):793–804
23. De Backer S, Naud A, Scheunders P (1998) Non-linear dimensionality reduction techniques for unsupervised feature extraction. Elsevier Science Inc., Amsterdam
24. Min R, Stanley D, Yuan Z, Bonner A, Zhang Z (2009) A deep non-linear feature mapping for large-margin KNN classification. In: Ninth IEEE international conference on data mining, pp 357–366
25. Jawanpuria P, Manik V, Nath JS: On p-norm path following in multiple kernel learning for non-linear feature selection. In: International conference on machine learning, pp 118–126 (2014)
26. Lei C, Zhu X (2017) Unsupervised feature selection via local structure learning and sparse learning. Multimedia Tools Appl 1:1–18
27. Zhu X, Zhang L, Huang Z (2014) A sparse embedding and least variance encoding approach to hashing. IEEE Trans Image Process 23(9):3737–50
28. Canyi L, Lin Z, Yan S (2015) Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization. IEEE Trans Image Process Publ IEEE Signal Process Soc 24(2):646–654
29. Zhang M, Ding C, Zhang Y, Nie F (2014) Feature selection at the discrete limit. AAAI, pp 2232–2237
30. Wei Z, Xiaofeng Z, Yonghua Z, Rongyao H, Cong L (2017) Dynamic graph learning for spectral feature selection. Multimedia Tools Appl. https://doi.org/10.1007/s11042-017-5272-y
31. Gower J, Dijksterhuis G (2004) Procrustes problems. Oxford University Press, Oxford
32. Nesterov Y (2004) Introductory lectures on convex optimization. Appl Optim 87(5):236
33. Zhu P, Zuo W, Zhang L, Qinghua H (2015) Unsupervised feature selection by regularized self-representation. Pattern Recogn 48(2):438–446
34. Nie F, Zhu W, Li X (2016) Unsupervised feature selection with structured graph optimization. In: Thirtieth AAAI conference on artificial intelligence, pp 1302–1308
35. Wang S, Tang J, Liu H (2015) Embedded unsupervised feature selection. AAAI, pp 470–476
36. Du L, Shen YD (2015) Unsupervised feature selection with adaptive structure learning, vol 37, no 7. ACM, pp 209–218
37. Nie F, Huang H, Cai X, Ding C (2010) Efficient and robust feature selection via joint l 2,1 -norms minimization. In: International conference on neural information processing systems, pp 1813–1821
38. He X, Cai D, Niyogi P (2005) Laplacian score for feature selection. In: International conference on neural information processing systems, pp 507–514
39. Li J, Hu X, Wu L, Liu H (2016) Robust unsupervised feature selection on networked data. In: Siam international conference on data mining, pp 387–395
40. Qian M, Zhai C (2013) Robust unsupervised feature selection. In: International joint conference on artificial intelligence, pp 1621–1627