



Latent representation learning based autoencoder for unsupervised feature selection in hyperspectral imagery

Xinxin Wang¹ · Zhenyu Wang¹ · Yongshan Zhang¹ · Xinwei Jiang¹ · Zhihua Cai¹

Received: 15 July 2020 / Revised: 2 October 2020 / Accepted: 29 December 2020 /
Published online: 27 February 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

In hyperspectral image (HSI) analysis, high-dimensional data may contain noisy, irrelevant and redundant information. To mitigate the negative effect from these information, feature selection is one of the useful solutions. Unsupervised feature selection is a data preprocessing technique for dimensionality reduction, which selects a subset of informative features without using any label information. Different from the linear models, the autoencoder is formulated to nonlinearly select informative features. The adjacency matrix of HSI can be constructed to extract the underlying relationship between each data point, where the latent representation of original data can be obtained via matrix factorization. Besides, a new feature representation can be also learnt from the autoencoder. For a same data matrix, different feature representations should consistently share the potential information. Motivated by these, in this paper, we propose a latent representation learning based autoencoder feature selection (LRLAFS) model, where the latent representation learning is used to steer feature selection for the autoencoder. To solve the proposed model, we advance an alternative optimization algorithm. Experimental results on three HSI datasets confirm the effectiveness of the proposed model.

✉ Yongshan Zhang
yszhang.cug@gmail.com

Xinxin Wang
wangxinxin@cug.edu.cn

Zhenyu Wang
zhenyuwang94@gmail.com

Xinwei Jiang
ysjxw@hotmail.com

Zhihua Cai
zhcai@cug.edu.cn

¹ School of Computer Science, China University of Geosciences, Wuhan, 430074, China

Keywords Autoencoder · Unsupervised feature selection · Latent representation learning · Hyperspectral image classification

1 Introduction

Hyperspectral imagery is commonly acquired from hyperspectral sensors and typically denoted as a data cube, where each pixel consists of plentiful spectral information to reflect the specific land cover [20, 38]. In general, hyperspectral image (HSI) data are collected from various spectral channels and represented as high-dimensional features, which may contain noisy, irrelevant and redundant information [23, 25]. To tackle the remote sensing problems with such data, the Hughes phenomenon may happen and the performance may be degraded for the limited number of instances with plentiful redundant features.

To avoid the Hughes phenomenon induced by high-dimensional hyperspectral data, dimensionality reduction should be taken into consideration [8, 15, 24]. Being a dimensionality reduction technique, feature selection is conducted according to a specific criterion, which can locate the discriminative feature subset. In [29], the multiobjective feature selection method for hyperspectral data is developed on a novel discrete sine cosine algorithm, where the redundancy is minimized and the relevance is maximized for the selected hyperspectral features. In [28], the high dimensional model representation approach for hyperspectral imagery calculates the class labels of auxiliary training instances by a k -nearest neighbor (KNN), which are used for ranking the importance of features. In [26], the Gabor hyperspectral feature selection model is related to symmetrical uncertainty and Markov blanket, where the extracted Gabor features are firstly ranked by the classification information and then evaluated their redundancy. In [6], the fast forward feature selection method is based on a Gaussian mixture model, which improves the classification accuracy via k -fold cross validation to select discriminative spectral features. However, the above feature selection methods for hyperspectral imagery are linear models and required the explicit labels. In reality, it is difficult and expensive to acquire abundant label information for hyperspectral data. Besides, the nonlinear characteristic of hyperspectral data is hard to extract by a linear model.

To address the aforementioned problems, in this paper, we propose a latent representation learning based autoencoder feature selection (LRLAFS) model, which can nonlinearly select the informative hyperspectral features for classification. The autoencoder is an unsupervised neural network model for efficient data codings [13]. By imposing the $\ell_{2,1}$ -norm regularization on the network parameters connecting the input and hidden layers, the autoencoder can be used to perform feature selection [11]. To steer the feature selection process, the adjacency matrix of hyperspectral data is decomposed to obtain the latent representation of original data, which is used for approximating a new feature representation learnt by the autoencoder. To solve the proposed model, a helpful alternative optimization algorithm is advanced. Experiments on three HSI datasets confirm the effectiveness of the proposed model.

The rest of this paper is structured as follows. Related work is briefly introduced in Section 2. Section 3 advances the proposed latent representation learning based autoencoder feature selection model, followed by the optimization algorithm in Section 4. Experiments and comparisons on HSI data are demonstrated in Section 5. We finally conclude the paper in Section 6.

2 Related work

2.1 Unsupervised feature selection

Being a data preprocessing technique, feature selection is used for dimensionality reduction, which requires less memory space and alleviates the computational cost [10, 37]. According to the availability of label information, feature selection methods are grouped into the supervised, semi-supervised and unsupervised methods. Unsupervised feature selection approaches are required to preserve the intrinsic geometric data structure and select a subset of informative features without using any label information [22, 30]. Recently, a large number of unsupervised feature selection approaches have been proposed, which are broadly split into the filter, wrapper and embedded approaches according to previous literatures [10, 19].

The filter approaches design an evaluation measure to independently score the importance of features. This is a simple kind of strategy for feature selection. The variance score is the simplest measure to select useful features with the assumption that better representation ability is along with larger variance. The laplacian score (LS) is a measure to select discriminative features for locality preservation of data with the assumption of manifold structure [12]. Filters usually require less computational complexity than wrappers, but are independent of predictive model. The wrapper approaches score the significance of each feature accompanied by a predictive model. Clustering and classification are commonly employed as predictive models in wrappers. The multi-cluster feature selection (MCFS) method utilizes a regression algorithm to promote unsupervised feature selection [4]. In addition, the embedded approaches are trained by all features and conduct feature selection during the model learning process, where the computational complexity is between filters and wrappers. Existing embedded approaches exploit the potential information and correlation to distinguish informative features by imposing different penalty terms for different purposes, such as sparsity restriction [39], nonnegative restriction [32] and redundancy control [17].

2.2 Autoencoder

The autoencoder [9, 14] is a special unsupervised neural network framework for learning auxiliary feature representations, where the same neurons are in the output layer as inputs to reconstruct the original input data instead of determining their predicted target values. During the learning process, the autoencoder firstly converts the input data to an auxiliary encoded representation and then decodes it to reconstruct a representation that approximates to the original input data as possible [36]. The above descriptions are respectively the encoding and decoding procedures, which can be also regarded as a nonlinear self-representation model. With the special learning process, the autoencoder can extract the discriminative encoded representation via the hidden layer, which contains the underlying information inherent in the original data. In diverse applications, different kinds of autoencoders are utilized to extract informative feature representations for further learning [3, 34].

In addition to feature extraction, the autoencoder can be also used to perform feature selection [1, 5]. When integrating a feature selection regularized term into the autoencoder, the network weights will be sparse in row to fulfill feature selection, which is shown as in Fig. 1. Since the relationships of features may be linear or nonlinear, the conventional linear feature selection methods may ignore the nonlinear characteristic and can not correctly capture the real relationships among features. In contrast, autoencoder based models can select informative features by excavating both linear and nonlinear information among features. In

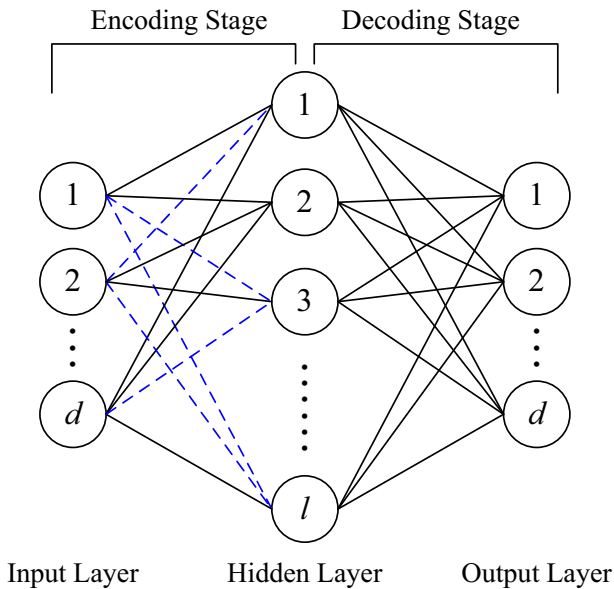


Fig. 1 Autoencoder network for feature selection. The blue dotted lines mean that the features are less important in the learning task, which can fulfill feature selection

[31], the feature selection guided autoencoder (FSAE) method is proposed, which integrates an autoencoder and a general feature selection regularizer to distinguish relevant and irrelevant features. In [11], the autoencoder feature selection (AEFS) approach is advanced by imposing the $\ell_{2,1}$ -norm regularization on the connecting parameters between the input and hidden layers, which can retain the most discriminative information among features. Following the previous work [11], the graph regularized autoencoder feature selection (GAFS) model [7] is proposed by adding the local data geometric structure regularization, which achieves promising performance. The above autoencoder based methods are superior to some linear approaches on feature selection, which mainly attribute to the special encoding and decoding procedures.

3 Proposed method

The autoencoder is an extraordinary unsupervised neural network learning model, where the output layer is to reconstruct the original inputs instead of predicting their target values [33]. Here, we employ a typical autoencoder network with only one hidden layer as an example. Given a hyperspectral data matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, the learning process of autoencoder is separated into two components: an encoder to transform the original data into a different representation $f(\mathbf{X}) = \sigma_1(\mathbf{W}^{(1)}\mathbf{X} + \mathbf{B}^{(1)})$ and a decoder to convert the new representation into the reconstructed input data $\hat{\mathbf{X}} = g(f(\mathbf{X})) = \sigma_2(\mathbf{W}^{(2)}f(\mathbf{X}) + \mathbf{B}^{(2)})$. In the above learning process, σ_1 and σ_2 are respectively set to be nonlinear and linear activation functions for the hidden and output layers, $\mathcal{H} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{B}^{(1)}, \mathbf{B}^{(2)}\}$ are the network weights to connect two adjacent layers of the autoencoder and the biases for the hidden and output layers.

During the learning process, the autoencoder aims at minimizing the difference between the original inputs and the reconstructed data, which is represented as the following equation:

$$\min_{\mathcal{H}} \|\mathbf{X} - g(f(\mathbf{X}))\|_F^2 \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. From (1), we can learn that the autoencoder encodes the data matrix \mathbf{X} to generate a representation matrix $f(\mathbf{X})$ and then decodes it to get the reconstructed data matrix $\hat{\mathbf{X}} = g(f(\mathbf{X}))$. According to nonlinear activation function used in the hidden layer, the autoencoder is normally treated as a nonlinear self-representation learning model.

When using the autoencoder model of (1), all features are adopted to solve hyperspectral image applications. However, the original hyperspectral data may contain noisy, irrelevant and redundant information. To eliminate the negative effect from these information, the associated network parameters (i.e., $\mathbf{W}^{(1)}$) of the autoencoder should be set as zeros or small values. This is typically regarded as feature selection [11]. To this end, the $\ell_{2,1}$ -norm constraint imposed on $\mathbf{W}^{(1)}$ is as follows:

$$\|\mathbf{W}^{(1)}\|_{2,1} = \sum_i^d \|\mathbf{w}_i^{(1)}\|_2 = \sum_i^d \sqrt{\sum_j^l (w_{ij}^{(1)})^2}. \tag{2}$$

where $\mathbf{w}_i^{(1)}$ is the i th row vector and $w_{ij}^{(1)}$ is the i th row and j th column element in matrix $\mathbf{W}^{(1)}$. Equation (2) can produce a row-sparse matrix $\mathbf{W}^{(1)}$, where the i th row vector $\mathbf{w}_i^{(1)}$ is to measure the importance of the i th feature. Therefore, the autoencoder based feature selection model is formulated as follows:

$$\min_{\mathcal{H}} \|\mathbf{X} - g(f(\mathbf{X}))\|_F^2 + \alpha \|\mathbf{W}^{(1)}\|_{2,1} \tag{3}$$

where α is a parameter to balance the reconstruction loss and the $\ell_{2,1}$ -norm regularizer.

In general, the overfitting problem and the slow convergence issue commonly occur on neural network learning models. To avoid overfitting and promote convergence for an autoencoder, a weight decay regularization is imperative. By adding the weight decay regularization, the above learning model is denoted as follows:

$$\min_{\mathcal{H}} \|\mathbf{X} - g(f(\mathbf{X}))\|_F^2 + \alpha \|\mathbf{W}^{(1)}\|_{2,1} + \beta \sum_{i=1}^2 \|\mathbf{W}^{(i)}\|_F^2 \tag{4}$$

where β is a parameter for the weight decay regularization. Equation (4) describes a basic autoencoder feature selection model, which can effectively select discriminative features.

In an autoencoder, $f(\mathbf{X})$ is the representation encoding from \mathbf{X} , which denotes the latent compact or sparse information. In reality, the latent representations of \mathbf{X} are various according to different learning scenarios. Uncovering latent representations from the original data benefits diverse learning tasks and has gained increasingly attention [27]. For networked data, the link information reflects that instances often connect to each other for their underlying relationships. Usually, latent representations can be learnt from the link information via a symmetric nonnegative matrix factorization model [18]. For hyperspectral image data, the similarity matrix \mathbf{A} can be constructed by a KNN graph, which describes the correlation between instances. Motivated by this, we employ latent representation learning to steer autoencoder based feature selection. By decomposing \mathbf{A} with symmetric matrix

factorization, we can get the following equation:

$$\min_{\mathbf{V}} \|\mathbf{A} - \mathbf{V}\mathbf{V}^T\|_F^2; \quad (5)$$

where \mathbf{V} is the latent representation of \mathbf{X} . Equation (5) captures the underlying data information and represents the original data in a different representation. For a same data matrix, different latent representations should consistently share the same information [27]. Therefore, the latent representations $f(\mathbf{X})$ and \mathbf{V} should be consistent. To minimize the difference between $f(\mathbf{X})$ and \mathbf{V} , we can obtain the following equation:

$$\min_{\mathcal{H}, \mathbf{V}} \|f(\mathbf{X}) - \mathbf{V}\|_F^2. \quad (6)$$

Equation (6) builds a connection between two different latent representations. Equations (4)–(6) represent the important components of the proposed model. It is clear that these separated equations cannot guarantee an optimal result. We integrate (5) and (6) into the basic model of (4). Therefore, the formulation of our proposed model is represented as the following equation:

$$\mathcal{J}(\mathcal{H}, \mathbf{V}) = \min_{\mathcal{H}, \mathbf{V}} \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 + \alpha \|\mathbf{W}^{(1)}\|_{2,1} + \beta \sum_{i=1}^2 \|\mathbf{W}^{(i)}\|_F^2 + \gamma \|f(\mathbf{X}) - \mathbf{V}\|_F^2 + \lambda \|\mathbf{A} - \mathbf{V}\mathbf{V}^T\|_F^2 \quad (7)$$

During the learning process, we can select informative features from the advanced techniques of autoencoder and latent representation learning. Once $\mathbf{W}^{(1)}$ is learned, we can sort all features to select the top ranked features by means of the descending order of $\|\mathbf{w}_i^{(1)}\|_2$ ($i = 1, \dots, d$).

Algorithm 1 LRLAFS.

Input: Hyperspectral data $\mathbf{X} \in \mathbb{R}^{n \times l}$, adjacency matrix \mathbf{A} , parameters α , β , λ and γ ;

Output: Top ranked features;

- 1: Initialization: randomly initialize weight matrix and hidden bias vector in \mathcal{H} and latent representation matrix \mathbf{V} ;
 - 2: **repeat**
 - 3: Calculate $\delta^{(o)}$ according to (8);
 - 4: Calculate $\delta^{(h)}$ according to (9);
 - 5: Set the elements in matrix \mathbf{V} as zeros when they are less than zeros;
 - 6: Calculate $\partial \mathcal{J} / \partial \mathbf{W}^{(2)}$ according to (10);
 - 7: Calculate $\partial \mathcal{J} / \partial \mathbf{B}^{(2)}$ according to (11);
 - 8: Calculate $\partial \mathcal{J} / \partial \mathbf{W}^{(1)}$ according to (12);
 - 9: Calculate $\partial \mathcal{J} / \partial \mathbf{B}^{(1)}$ according to (13);
 - 10: Calculate $\partial \mathcal{J} / \partial \mathbf{V}$ according to (14);
 - 11: Update $\mathbf{W}^{(2)} = \mathbf{W}^{(2)} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}}$;
 - 12: Update $\mathbf{B}^{(2)} = \mathbf{B}^{(2)} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{B}^{(2)}}$;
 - 13: Update $\mathbf{W}^{(1)} = \mathbf{W}^{(1)} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}}$;
 - 14: Update $\mathbf{B}^{(1)} = \mathbf{B}^{(1)} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{B}^{(1)}}$;
 - 15: Update $\mathbf{V} = \mathbf{V} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{V}}$;
 - 16: Update \mathbf{D} as $d_{ii} = \frac{1}{2\|\mathbf{w}_i^{(1)}\|_2}$;
 - 17: **until** Convergence
 - 18: Select the top ranked features by sorting $\|\mathbf{w}_i^{(1)}\|_2$ in descending order
-

4 Optimization algorithm

Since the proposed model is based on an autoencoder network to perform feature selection, it is hard to solve (7) for the complex nonlinearity. Therefore, the gradient descent method [2] is employed as an alternating optimization algorithm to iteratively update the network parameters $\mathcal{H} = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{B}^{(1)}, \mathbf{B}^{(2)}\}$ and the latent representation \mathbf{V} . To calculate the derivatives of (7) w.r.t variables $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{B}^{(1)}, \mathbf{B}^{(2)}$ and \mathbf{V} , the calculation of the error terms for the output and hidden layers is required for the chain rule. Specifically, the error terms of the output and hidden layers are determined as follows:

$$\delta^{(o)} = 2(\mathbf{X} - g(f(\mathbf{X}))) \odot g'(f(\mathbf{X})); \tag{8}$$

$$\delta^{(h)} = (\delta^{(o)} \mathbf{W}^{(2)T}) \odot f'(\mathbf{X}); \tag{9}$$

where \odot denotes the element-wise product. Given the above error terms in (8) and (9), the derivatives of (7) w.r.t different variables can be consecutively determined.

- 1) *Derivative w.r.t $\mathbf{W}^{(2)}$* : With a fixed $\mathbf{W}^{(1)}, \mathbf{B}^{(1)}, \mathbf{B}^{(2)}$ and \mathbf{V} , the optimization problem of (7) w.r.t $\mathbf{W}^{(2)}$ can be rewritten as $\mathcal{J}(\mathbf{W}^{(2)}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 + \beta \|\mathbf{W}^{(2)}\|_F^2$. The derivative w.r.t $\mathbf{W}^{(2)}$ is given as:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} = \mathbf{Z}^{(o)T} \delta^{(o)} + \beta \mathbf{W}^{(2)}. \tag{10}$$

- 2) *Derivative w.r.t $\mathbf{B}^{(2)}$* : With a fixed $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{B}^{(1)}$ and \mathbf{V} , the optimization problem of (7) w.r.t $\mathbf{B}^{(2)}$ can be rewritten as $\mathcal{J}(\mathbf{B}^{(2)}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$. The derivative w.r.t $\mathbf{B}^{(2)}$ is given as:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{B}^{(2)}} = \delta^{(o)} \tag{11}$$

- 3) *Derivative w.r.t $\mathbf{W}^{(1)}$* : With a fixed $\mathbf{W}^{(2)}, \mathbf{B}^{(1)}, \mathbf{W}^{(2)}$ and \mathbf{V} , the optimization problem of (7) w.r.t $\mathbf{W}^{(1)}$ can be rewritten as $\mathcal{J}(\mathbf{W}^{(1)}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 + \alpha \|\mathbf{W}^{(1)}\|_{2,1} + \beta \|\mathbf{W}^{(1)}\|_F^2 + \gamma \|f(\mathbf{X}) - \mathbf{V}\|_F^2$. The derivative w.r.t $\mathbf{W}^{(1)}$ is given as:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} = \mathbf{X}^T \delta^{(h)} + \alpha \mathbf{D} \mathbf{W}^{(1)} + \beta \mathbf{W}^{(1)} + 2\gamma \mathbf{X}^T ((f(\mathbf{X}) - \mathbf{V}) \odot f'(\mathbf{X})). \tag{12}$$

In (12), \mathbf{D} is a diagonal matrix updated in each iteration, where each diagonal element is calculated as $d_{ii} = \frac{1}{2\|w_i^{(1)}\|_2}$.

- 4) *Derivative w.r.t $\mathbf{B}^{(1)}$* : With a fixed $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{B}^{(1)}$ and \mathbf{V} , the optimization problem of (7) w.r.t $\mathbf{B}^{(1)}$ can be rewritten as $\mathcal{J}(\mathbf{B}^{(1)}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 + \gamma \|f(\mathbf{X}) - \mathbf{V}\|_F^2$. The derivative w.r.t $\mathbf{B}^{(1)}$ is given as:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{B}^{(1)}} = \delta^{(h)} + 2\gamma (f(\mathbf{X}) - \mathbf{V}) \odot f'(\mathbf{X}). \tag{13}$$

- 5) *Derivative w.r.t \mathbf{V}* : With a fixed $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$, the optimization problem of (7) w.r.t \mathbf{V} can be rewritten as $\mathcal{J}(\mathbf{V}) = \gamma \|f(\mathbf{X}) - \mathbf{V}\|_F^2 + \lambda \|\mathbf{A} - \mathbf{V}\mathbf{V}^T\|_F^2$. The derivative w.r.t \mathbf{V} is given as:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{V}} = 2\gamma (f(\mathbf{X}) - \mathbf{V}) + 4\lambda (\mathbf{A} - \mathbf{V}\mathbf{V}^T) \mathbf{V}. \tag{14}$$

Given the above derivatives, a novel alternative optimization algorithm is devised to alternately update variables $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, $\mathbf{B}^{(1)}$, $\mathbf{B}^{(2)}$ and \mathbf{V} . The update rule of the alternating solution for the proposed model is provided as follows:

$$\left\{ \begin{array}{l} \mathbf{W}^{(1)} := \mathbf{W}^{(1)} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(1)}} \\ \mathbf{B}^{(1)} := \mathbf{B}^{(1)} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{B}^{(1)}} \\ \mathbf{W}^{(2)} := \mathbf{W}^{(2)} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{W}^{(2)}} \\ \mathbf{B}^{(2)} := \mathbf{B}^{(2)} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{B}^{(2)}} \\ \mathbf{V} := \mathbf{V} - \eta \frac{\partial \mathcal{J}}{\partial \mathbf{V}} \end{array} \right. \quad (15)$$

where η is the learning rate for the update rule. The pseudocode of the proposed method is exhibited in Algorithm 1.

5 Experiments and results

5.1 Hyperspectral data

To validate the performance of different feature selection approaches, three public HSI datasets¹ are utilized for experimental studies, including Indian Pine, University of Pavia and Salinas Scene.

The Indian Pine dataset was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over Northwest Indiana. This scene consists of 145×145 pixels with 200 valid spectral bands and records 16 land covers of different plants. The false color composition and ground truth for the Indian Pines dataset are respectively shown in Figs. 2a and d.

The University of Pavia dataset was captured by the Reflective Optics System Imaging Spectrometer (ROSIS-3) sensor to record a scene from the University of Pavia, Italy. This scene contains 610×340 pixels with 103 informative spectral bands and includes 9 different land covers. The false color composition and ground truth for the University of Pavia dataset are respectively shown in Figs. 2b and e.

The Salinas Scene dataset was collected by the AVIRIS sensor to record a scene from Salinas Valley, California, USA. There are 512×217 pixels with 204 useful spectral bands in this scene, which is associated with 16 diverse land covers. The false color composition and ground truth for the Salinas Scene dataset are respectively shown in Figs. 2c and f.

5.2 Comparison method and experimental setting

In the experiments, the proposed LRLAFS method is compared with the following methods. SpaBS is based on sparse representation to perform feature selection, where the histogram of the coefficient matrix decomposed from hyperspectral image data is calculated to select the most discriminative features [16]. AEFS selects informative features by combining auto-encoder regression and group lasso tasks, which can take full advantage of linear and nonlinear

¹http://alweb.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

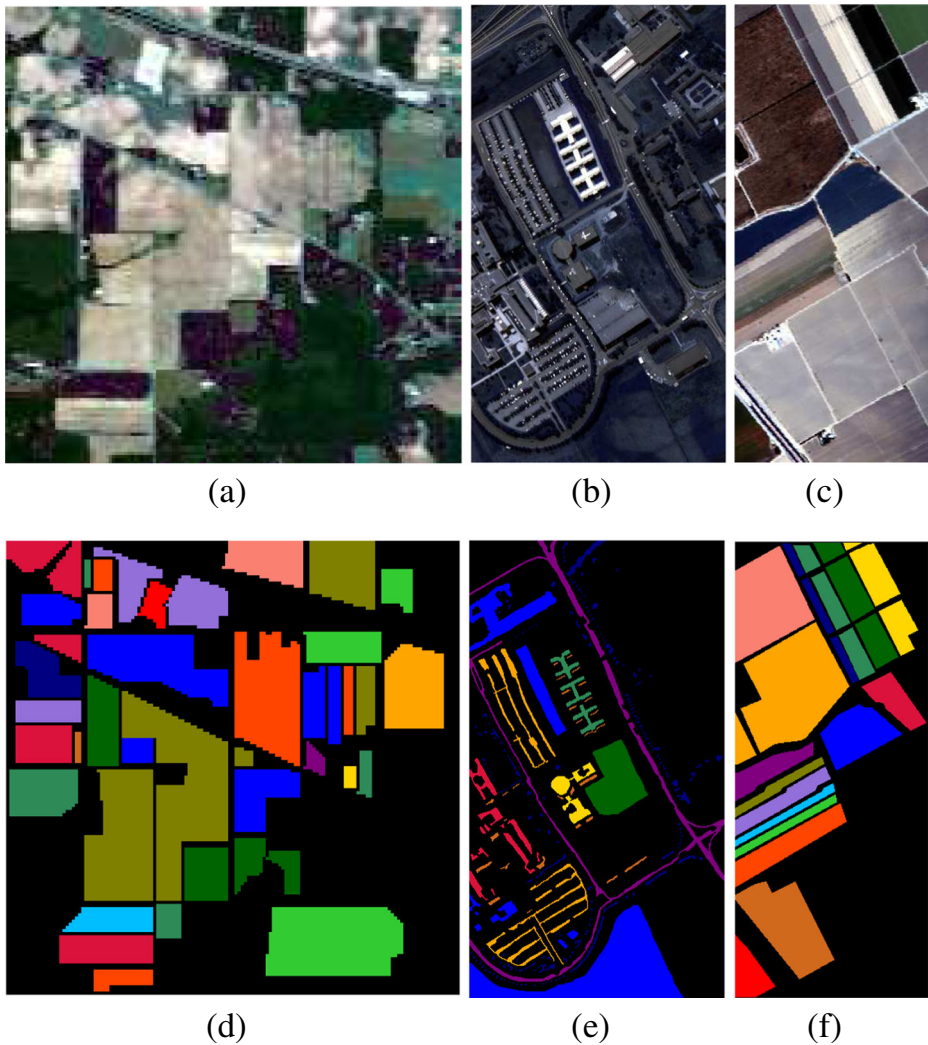


Fig. 2 The false color composition (in the upper row) and ground truth (in the bottom row) of three hyperspectral image datasets. **a** and **d** Indian Pines. **b** and **e** University of Pavia. **c** and **f** Salinas Scene

information among features [11]. GAFS integrates manifold learning into the basic autoencoder feature selection model, where spectral graph analysis is considered into the learning process for local data geometry preservation [7]. AllFeas is the baseline method with all original features for hyperspectral image classification.

For a fair comparison, k -nearest neighbors (KNN) [21] is employed as the classifier to validate all compared methods. The parameter k in KNN is empirically set to be 5. Overall accuracy (OA), average accuracy (AA) and Kappa coefficient are popularly used as evaluation measures in HSI classification. The experiments are conducted on a computer with Intel(R) Core(TM) 2.90 GHZ CPU and 8 GB RAM. In the proposed LRLAFS model, parameters α , β , γ and λ are tuned from $\{10^{-3}, 10^{-2}, \dots, 10^3\}$ by means of a grid-search

Table 1 Experimental comparison of different feature selection approaches on the Indian Pines dataset

Class	AllFeas	SpaBS	AEFS	GAFS	LRLAFS
1	69.57	68.43	71.74	67.39	86.96
2	66.04	67.25	62.75	67.37	71.43
3	64.58	65.88	61.81	65.18	71.93
4	47.68	60.02	51.48	52.32	54.85
5	91.10	89.14	88.61	92.55	92.13
6	95.34	94.62	92.74	93.97	95.62
7	89.29	88.29	75.00	89.29	89.29
8	98.54	98.94	98.33	98.54	98.12
9	75.00	76.72	65.00	50.00	70.00
10	76.75	74.83	73.77	76.34	83.02
11	77.52	75.02	76.17	77.39	82.48
12	59.87	53.17	54.30	52.61	67.96
13	95.61	94.81	93.17	97.56	98.05
14	92.09	92.33	91.86	93.20	92.96
15	53.89	52.05	48.96	47.41	53.11
16	92.47	92.43	92.47	92.47	92.47
OA(%)	77.37	76.91	75.23	77.07	81.38
AA(%)	77.83	76.34	74.88	75.85	81.27
Kappa	0.7418	0.7363	0.7172	0.7383	0.7875

strategy. Besides, the numbers of hidden neurons are tuned from {5, 15, 25, ..., 95}, while the numbers of selected features are varying from {3%, 6%, 9%, ..., 66%} of the number of features. The best results presented in the following sections are achieved by setting the optimal parameters [35].

5.3 Experimental results

The experimental results of all competing feature selection approaches on the three HSI datasets are discussed in this section.

- (1) *Comparison for the Indian Pines data:* The experimental results on the Indian Pines data are summarized in Table 1. From Table 1, the proposed LRLAFS method is superior to AllFeas and other competing methods are inferior to AllFeas. It demonstrates that the proposed LRLAFS method can select informative features and remove uninformative features from the original Indian Pines data, which benefits the classification performance. While SpaBS, AEFS and GAFS may select uninformative features from the original feature space, which degrades the further classification performance. For autoencoder based models, AEFS, GAFS and LRLAFS show the increasing performance on feature selection, which mainly attributes to the manifold learning in GAFS and the latent representation learning in LRLAFS. Accordingly, the classification maps of all competing approaches are presented in Fig. 3, which are consistent with our previous observation.
- (2) *Comparison for the University of Pavia data:* We report the experimental results on the University of Pavia data in Table 2. According to Table 2, AEFS and GAFS

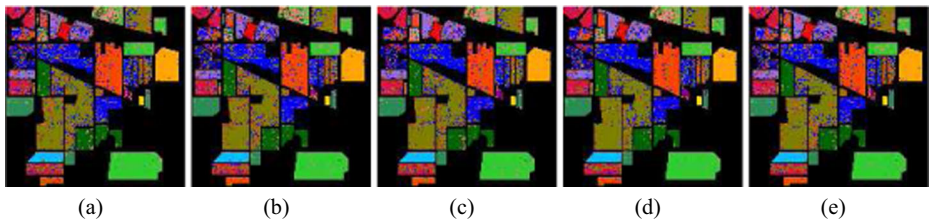


Fig. 3 Classification maps of all competing approaches on the Indian Pines dataset. **a** AllFeas. **b** SpaBS. **c** AEFS. **d** GAFS. **e** LRLAFS

demonstrate the worst and the second worst performance compared to other feature selection methods, while LRLAFS shows the best performance. Compared to AEFS, the improvement of GAFS and LRLAFS is at 1.4% and 6.6% for overall accuracy, 1.2% and 7.8% for average accuracy, and 1.9% and 9.5% for Kappa coefficient. Compared to AllFeas, LRLAFS can choose discriminative features from the original University of Pavia features, which alleviates computational burden and improves the further classification performance. For SpaBS, the overall accuracy is 85.42%, the average accuracy is 83.28%, and the Kappa coefficient is 0.8051, which shows inferior performance compared to AllFeas and achieves superior performance compared to AEFS and GAFS. More detailed comparisons can be discovered in Table 2 and the classification maps for all compared methods are shown in Fig. 4.

- (3) *Comparison for the Salinas Scene data:* Table 3 presents the experimental results on the Salinas Scene data. According to Table 3, LRLAFS exhibits superior performance (90.76% for overall accuracy, 95.47% for average accuracy, and 0.8972 for Kappa coefficient) than AllFeas and other feature selection methods, which mainly attributes to the latent representation learning embedding in the feature selection process. AllFeas and SpaBS present comparative performance and slightly outperform AEFS and GAFS, which means that SpaBS is good at choosing informative features on the Salinas Scene data. In details, the overall accuracy of SpaBS, AEFS and GAFS is 90.51%, 90.20% and 90.43%, respectively. From Table 3, more detailed

Table 2 Experimental comparison of different feature selection approaches on the University of Pavia dataset

Class	AllFeas	SpaBS	AEFS	GAFS	LRLAFS
1	87.71	86.37	84.31	85.52	88.56
2	94.64	93.88	91.85	93.11	94.67
3	67.14	66.43	57.62	56.90	63.52
4	85.97	84.87	83.36	84.83	85.32
5	98.88	98.88	98.88	98.88	98.88
6	69.78	68.02	62.82	64.21	70.91
7	78.20	74.51	67.67	70.68	75.19
8	82.34	80.37	77.58	78.40	82.03
9	99.47	99.47	99.47	99.47	100.00
OA(%)	87.34	85.42	83.39	84.52	88.89
AA(%)	84.90	83.28	80.40	81.33	86.79
Kappa	0.8308	0.8051	0.7781	0.7930	0.8517

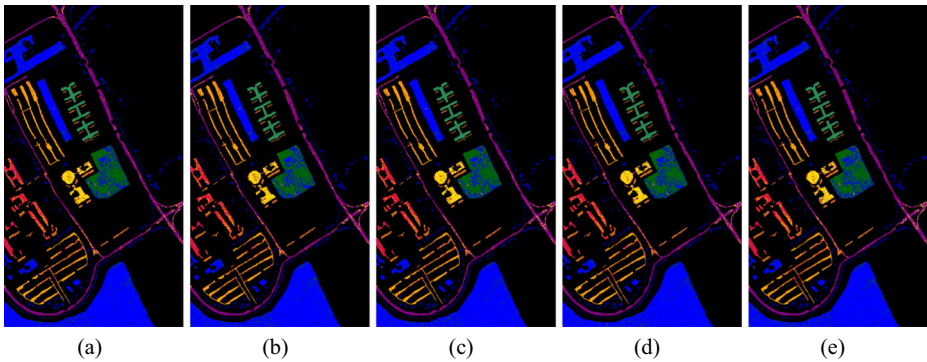


Fig. 4 Classification maps of all competing approaches on the University of Pavia dataset. **a** AllFeas. **b** SpaBS. **c** AEFS. **d** GAFS. **e** LRLAFS

comparisons can be observed. Besides, Fig. 5 shows the classification maps for all compared methods.

5.4 Parameter sensitivity study

In LRLAFS, parameters α , β , γ and λ are utilized in the objective function, which are needed to be set in advance. Here, we study the sensitivity of these four parameters and observe how they influence the final results. These four parameters are tuned from

Table 3 Experimental comparison of different feature selection approaches on the Salinas Scene dataset

Class	AllFeas	SpaBS	AEFS	GAFS	LRLAFS
1	98.76	98.34	97.51	99.00	98.76
2	99.60	99.52	99.46	99.33	99.60
3	99.24	99.20	99.24	98.99	99.24
4	100.00	100.00	100.00	100.00	100.00
5	97.76	97.48	97.39	97.39	97.95
6	99.75	99.77	99.87	99.75	99.75
7	99.16	99.23	99.44	99.16	99.30
8	78.93	79.03	79.15	78.66	79.55
9	99.44	99.42	99.36	99.36	99.36
10	94.66	94.17	93.75	93.75	94.51
11	96.73	96.02	95.79	95.79	97.20
12	99.48	99.37	99.22	99.48	100.00
13	98.36	98.28	97.27	98.36	98.36
14	95.79	95.80	96.26	95.79	95.33
15	69.53	68.89	67.19	69.46	69.46
16	99.17	99.08	98.89	98.89	99.17
OA(%)	90.63	90.51	90.20	90.43	90.76
AA(%)	95.40	95.25	94.99	95.20	95.47
Kappa	0.8957	0.8944	0.8909	0.8940	0.8972

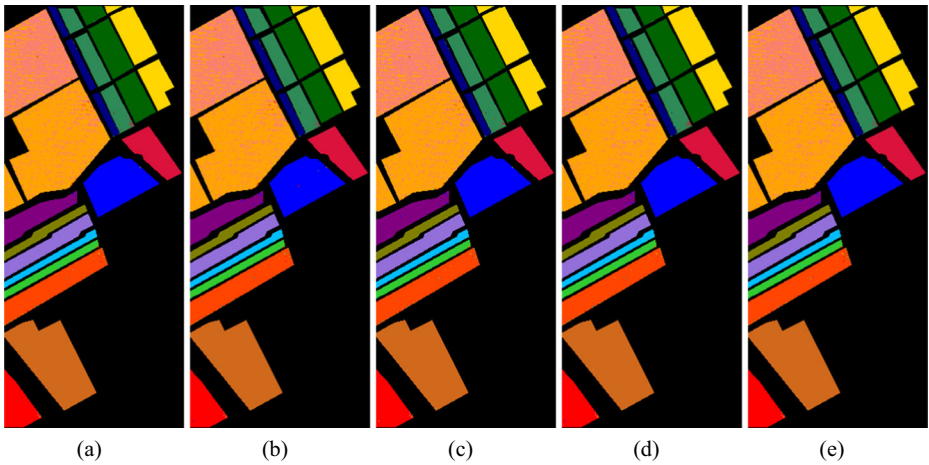


Fig. 5 Classification maps of all competing approaches on the Salinas Scene dataset. **a** AllFeas. **b** SpaBS. **c** AEFS. **d** GAFS. **e** LRLAFS

$\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3\}$. The experimental results of parameter sensitivity study on three HSI datasets are depicted in Fig. 6. For the Indian Pine data, the proposed LRLAFS shows superior performance with $\alpha = 10$ (presented in Fig. 6a). For the University of Pavia data, the inferior result is achieved with $\omega = 1$, which is shown in Fig. 6k. For the Salinas Scene data, the superior results is obtained with $\beta = 0.001$, which is displayed

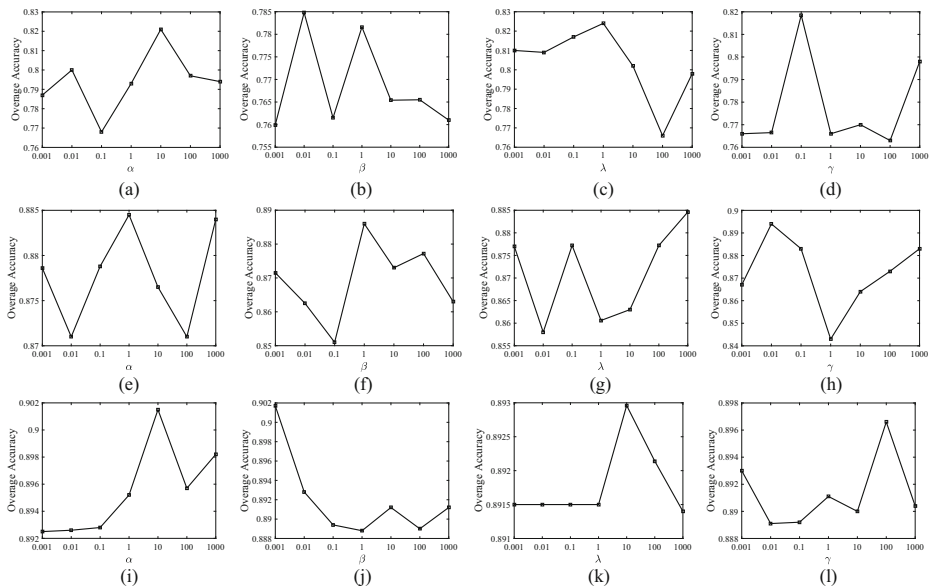


Fig. 6 Performance variation of LRLAFS for α (in the first column), β (in the second column), γ (in the third column) and λ (in the fourth column) on hyperspectral image datasets. **a, b, c** and **d** Indian Pine. **e, f, g** and **h** University of Pavia. **i, j, k** and **l** Salinas Scene

in Fig. 6f. Therefore, it is important to determine appropriate values for the four parameters to conduct feature selection.

6 Conclusions

In this paper, we propose a novel latent representation learning based autoencoder feature selection (LRLAFS) model for hyperspectral image (HSI) data. To select informative features from HSI data, we incorporate the latent representation learning into the basic autoencoder feature selection model, where the latent representations learnt by the hidden layer of the autoencoder and decomposed by matrix factorization from the adjacency matrix are used to steer the nonlinear feature selection. An alternative optimization algorithm is introduced to solve the optimization for LRLAFS. Experimental comparisons on three HSI datasets prove the potential of LRLAFS and LRLAFS consistently outperforms other compared methods.

Acknowledgments This work is supported in part by the National Nature Science Foundation of China under Grant 61703355, the Natural Science Foundation of Hubei Province of China under Grant 2020CFB328, the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan).

References

1. Abid A, Balin MF, Zou J (2019) Concrete autoencoders for differentiable feature selection and reconstruction. In: International conference on machine learning, pp 444–453
2. Andrychowicz M, Denil M, Gomez S, Hoffman MW, Pfau D, Schaul T, Shillingford B, De Freitas N (2016) Learning to learn by gradient descent by gradient descent. In: Advances in neural information processing systems, pp 3981–3989
3. Ap SC, Lauly S, Larochelle H, Khapra M, Ravindran B, Raykar VC, Saha A (2014) An autoencoder approach to learning bilingual word representations. In: Advances in neural information processing systems, pp 1853–1861
4. Cai D, Zhang C, He X (2010) Unsupervised feature selection for multi-cluster data. In: ACM SIGKDD international conference on knowledge discovery and data mining, pp 333–342
5. Chandra B, Sharma RK (2015) Exploring autoencoders for unsupervised feature selection. In: International joint conference on neural networks, pp 1–6
6. Fauvel M, Dechesne C, Zullo A, Ferraty F (2015) Fast forward feature selection of hyperspectral images for classification with gaussian mixture models. *IEEE J Sel Top Appl Earth Observ Remote Sens* 8(6):2824–2831
7. Feng S, Duarte MF (2018) Graph regularized autoencoder-based unsupervised feature selection. In: Asilomar conference on signals, systems, and computers, pp 55–59
8. Feng J, Jiao L, Liu F, Sun T, Zhang X (2016) Unsupervised feature selection based on maximum information and minimum redundancy for hyperspectral images. *Pattern Recognit* 51:295–309
9. Gnouma M, Ladjailia A, Ejbali R, Zaied M (2019) Stacked sparse autoencoder and history of binary motion image for human activity recognition. *Multimed Tools Appl* 78(2):2157–2179
10. Gui J, Sun Z, Ji S, Tao D, Tan T (2016) Feature selection based on structured sparsity: a comprehensive study. *IEEE Trans Neural Netw Learn Syst* 28(7):149–1507
11. Han K, Wang Y, Zhang C, Li C, Xu C (2018) Autoencoder inspired unsupervised feature selection. In: IEEE international conference on acoustics, speech and signal processing, pp 2941–2945
12. He X, Cai D, Niyogi P (2006) Laplacian score for feature selection. In: Advances in neural information processing systems, pp 507–514
13. Hong C, Yu J, Wan J, Tao D, Wang M (2015) Multimodal deep autoencoder for human pose recovery. *IEEE Trans Image Process* 24(12):5659–5670
14. Jia W, Muhammad K, Wang SH, Zhang YD (2019) Five-category classification of pathological brain images based on deep stacked sparse autoencoder. *Multimed Tools Appl* 78(4):4045–4064

15. Jiang J, Ma J, Chen C, Wang Z, Cai Z, Wang L (2018) Superpca: a superpixelwise pca approach for unsupervised feature extraction of hyperspectral imagery. *IEEE Trans Geosci Remote Sens* 56(8):4581–4593
16. Li S, Qi H (2011) Sparse representation based band selection for hyperspectral images. In: *IEEE international conference on image processing*, pp 2693–2696
17. Li Z, Tang J (2015) Unsupervised feature selection via nonnegative spectral analysis and redundancy control. *IEEE Trans Image Process* 24(12):5343–5355
18. Li J, Hu X, Wu L, Liu H (2009) Robust unsupervised feature selection on networked data. In: *SIAM international conference on data mining*, pp 387–395
19. Li Z, Liu J, Yang Y, Zhou X, Lu H (2013) Clustering-guided sparse structural learning for unsupervised feature selection. *IEEE Trans Knowl Data Eng* 26(9):2138–2150
20. Lunga D, Prasad S, Crawford MM, Ersoy O (2014) Manifold-learning-based feature extraction for classification of hyperspectral data: a review of advances in manifold learning. *IEEE Signal Process Mag* 31(1):55–66
21. Maillou J, García S, Luengo J, Herrera F, Triguero I (2019) Fast and scalable approaches to accelerate the fuzzy k-nearest neighbors classifier for big data. *IEEE Trans Fuzzy Syst* 28(5):874–886
22. Mitra P, Murthy C, Pal SK (2002) Unsupervised feature selection using feature similarity. *IEEE Trans Pattern Anal Mach Intell* 24(3):301–312
23. Pal M, Foody GM (2010) Feature selection for classification of hyperspectral data by svm. *IEEE Trans Geosci Remote Sens* 48(5):2297–2307
24. Prasad S, Bruce LM (2008) Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geosci Remote Sens Lett* 5(4):625–629
25. Serpico SB, Bruzzone L (2001) A new search algorithm for feature selection in hyperspectral remote sensing images. *IEEE Trans Geosci Remote Sens* 39(7):1360–1367
26. Shen L, Zhu Z, Jia S, Zhu J, Sun Y (2012) Discriminative gabor feature selection for hyperspectral image classification. *IEEE Geosci Remote Sens Lett* 10(1):29–33
27. Tang C, Bian M, Liu X, Li M, Zhou H, Wang P, Yin H (2019) Unsupervised feature selection via latent representation learning and manifold regularization. *Neural Netw* 117:163–178
28. Taşkın G, Kaya H, Bruzzone L (2017) Feature selection based on high dimensional model representation for hyperspectral images. *IEEE Trans Image Process* 26(6):2918–2928
29. Wan Y, Ma A, Zhong Y, Hu X, Zhang L (2020) Multiobjective hyperspectral feature selection based on discrete sine cosine algorithm. *IEEE Trans Geosci Remote Sens* 58(5):3601–3618
30. Wang S, Tang J, Liu H (2015) Embedded unsupervised feature selection. In: *AAAI conference on artificial intelligence*, pp 470–476
31. Wang S, Ding Z, Fu Y (2017) Feature selection guided auto-encoder. In: *AAAI conference on artificial intelligence*, pp 2725–2731
32. Yang Y, Shen HT, Nie F, Ji R, Zhou X (2011) Nonnegative spectral clustering with discriminative regularization. In: *Twenty-fifth AAAI conference on artificial intelligence*, pp 555–560
33. Yang X, Deng C, Zheng F, Yan J, Liu W (2019) Deep spectral clustering using dual autoencoder network. In: *IEEE conference on computer vision and pattern recognition*, pp 4066–4075
34. Zeng K, Yu J, Wang R, Li C, Tao D (2015) Coupled deep autoencoder for single image super-resolution. *IEEE Trans Cybern* 47(1):27–37
35. Zhang Y, Jiang X, Wang X, Cai Z (2019) Spectral-spatial hyperspectral image classification with superpixel pattern and extreme learning machine. *Remote Sens* 11(17):1–20
36. Zhang Y, Wu J, Cai Z, Du B, Philip SY (2019) An unsupervised parameter learning model for rvfl neural network. *Neural Netw* 112:85–97
37. Zhang Y, Wu J, Cai Z, Yu P (2020) Multi-view multi-label learning with sparse feature selection for image annotation. *IEEE Trans Multimed* 1–14. <https://doi.org/10.1109/TMM.2020.2966887>
38. Zhou Y, Peng J, Chen CP (2015) Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification. *IEEE Trans Geosci Remote Sens* 53(2):1082–1095
39. Zhu X, Li X, Zhang S, Ju C, Wu X (2016) Robust joint graph sparse coding for unsupervised spectral feature selection. *IEEE Trans Neural Netw Learn Syst* 28(6):1263–1275