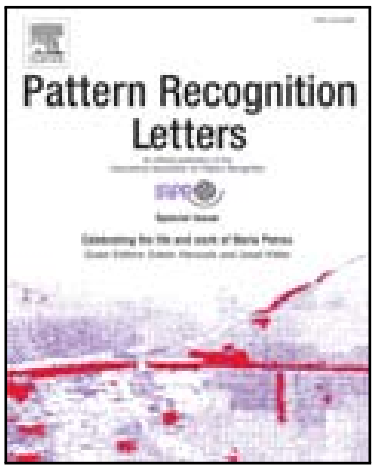

Unsupervised feature selection by self-paced learning regularization

Wei Zheng, Xiaofeng Zhu, Guoqiu Wen, Yonghua Zhu, Hao Yu, Jiangzhang Gan

PII: S0167-8655(18)30278-2
DOI: 10.1016/j.patrec.2018.06.029
Reference: PATREC 7223

To appear in: *Pattern Recognition Letters*

Received date: 11 March 2018
Revised date: 20 June 2018
Accepted date: 27 June 2018

Please cite this article as: Wei Zheng, Xiaofeng Zhu, Guoqiu Wen, Yonghua Zhu, Hao Yu, Jiangzhang Gan, Unsupervised feature selection by self-paced learning regularization, *Pattern Recognition Letters* (2018), doi: 10.1016/j.patrec.2018.06.029

Highlights

- This paper uses self-representation method to construct feature selection model.
- Self-paced learning is added into feature selection to consider the outliers.
- This paper proposes a novel optimization algorithm to solve the objective function

Pattern Recognition Letters
journal homepage: www.elsevier.com

Unsupervised feature selection by self-paced learning regularization

Wei Zheng[a,b], Xiaofeng Zhu[a,b,d,**], Guoqiu Wen[a,b], Yonghua Zhu[c], Hao Yu[a,b], Jiangzhang Gan[a,b]

[a]*College of Computer Science and Information Technology, Guangxi Normal University, Guilin, Guangxi, 541004, P.R. China*
[b]*Guangxi Key Lab of Multi-Source Information Mining and Security, Guilin, Guangxi, 541004, P.R. China*
[c]*School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, P.R. China*
[d]*Institute of Natural and Mathematical Sciences, Massey University Albany Campus, Auckland 0632, New Zealand*

ABSTRACT

Previous feature selection methods equivalently consider the samples to select important features. However, the samples are often diverse. For example, the outliers should have small or even zero weights while the important samples should have large weights. In this paper, we add a self-paced regularization in the sparse feature selection model to reduce the impact of outliers for conducting feature selection. Specifically, the proposed method automatically selects a sample subset which includes the most important samples to build an initial feature selection model, whose generalization ability is then improved by involving other important samples until a robust and generalized feature selection model has been established or all the samples have been used. Experimental results on eight real datasets show that the proposed method outperforms the comparison methods.

1. Introduction

Big data has been widely appearing in various fields, such as pattern recognition and machine learning Zhang et al. (2018); Zhu et al. (2016c); Gao et al. (2017); Song et al. (2018); Zhang et al. (2012a). A common issue in the data processing is that big data often contain unimportant features, which increase the computational cost and affect the effectiveness of the learning of big data Song et al. (2016b,a); Yang et al. (2018); Zhang et al. (2012b). Moreover, the unimportant features in big data easily lead to the issue of curse of dimensionality Zhang et al. (2017); Zhu et al. (2017b). Recently, dimensionality reduction (such as feature selection and subspace learning) has become one of the important research fields via reducing the number of features of big data Zhu et al.; Lei and Zhu (2017); Zhu et al. (2014, 2017c).

Feature selection is designed to delete the redundant features for conducting dimensionality reduction. Existing feature selection methods can be commonly partitioned into three categories, *i.e.,* filter method Gheyas and Smith (2010); Zhu et al. (2010), wrapper method Gutlein et al. (2009); Unler et al. (2011), and embedded method Liu et al. (2017); Zhang and Hancock (2017). Filter method first selects useful features (*i.e.,* important features) from all the features by certain evaluation criterion, and then uses the selected feature subset to conduct classification or clustering tasks, and thus simple and efficiently selecting features. Wrapper method directly utilizes the training model to evaluate each feature subset so that finding the best feature subset as the results of feature selection. Obviously, wrapper method is more excellent than filter method. However, wrapper method is more complex than filter method. Embedded method automatically selects the useful features during the training process, *i.e.,* integrating the feature selection procedure into the training model. Embedded method is more effective than both filter method and wrapper method. Hence, embedded method has been attracting a number of research interests Hu et al. (2017); Zheng et al..

In this paper, we introduce the self-paced learning technique into the sparse feature selection framework to consider the sample diversity, based on that different samples have different contributions to the feature selection model Kumar et al. (2010); Shah and Koltun (2017). Specifically, our proposed method, namely unsupervised feature selection by self-paced regularization (UFS_SP for short), first obtains the self-representation coefficient matrix by using the feature level self-representation Hu et al. (2017); Lei and Zhu (2017) as well as employs the $\ell_{2,1}$-norm regularization to penalize the coefficient matrix so

**Corresponding author: Xiaofeng Zhu
e-mail: seanzhuxf@gmail.com (Xiaofeng Zhu)

that the weight of the irrelevant feature will become small (even zero) and the important features are assigned large weights. Our method then integrates a self-paced learning regularization Kumar et al. (2010); Shah and Koltun (2017) into the constructed feature selection framework. In this way, the proposed method first automatically selects the most important samples as a subset to initialize the feature selection model, and then selects the most important samples from remaining samples to improve the robustness and generalization ability of the initial feature selection model. This process is repeated until all the samples have been selected or the feature selection model achieves stability. As a result, all useful samples can be chosen to take participation in the process of the feature selection model, and the outliers will be selected later or never be selected. Moreover, we propose a novel iterative optimization algorithm to optimize the resulting objective function and the optimization method enables the proposed method to fast converge.

By comparing to previous feature selection methods, the contributions of the proposed method are summarized as follows:

- Self-paced learning theory Kumar et al. (2010) is added into the sparse feature selection framework to jointly consider the sample and feature diversity. Self-paced learning implements a learning mode from simple to hard by simulating human or animal learning mechanisms. The proposed method can automatically assign a weight to each sample, and then gradually adds important samples in the iterative process to train the feature selection model. Hence, the impact of outliers can be relieved or removed.
- This paper proposes an effective optimization algorithm to optimize the proposed objective function. Important samples are iteratively selected through an iterative process, and the current optimal solution is obtained by optimizing the objective function based on the currently selected samples until all the samples are used and the final optimal solution is obtained.

2. Related Work

In this section, we review the state-of-the-art methods of the topics related to our proposed method, *i.e.,* feature selection Zhu et al., 2016c); Lei and Zhu (2017) and self-paced learning Lin et al. (2018); Kumar et al. (2010); Meng et al. (2017).

2.1. Feature selection

As an important dimensionality reduction technique, feature selection tries to find a most representative feature subset from original features Zhu et al. (2016b, 2017a); Zhu and Lucey (2015). Different from subspace learning Zhu et al. (2017b,d) which utilizes the transformation matrix to project the high-dimensional data to their low-dimensional subspace, feature selection ranks all the features by a certain approach, such as evaluation score Benabdeslem and Hindawi (2011); Liu et al. (2013) and sparse learning Tan et al. (2010); Li et al. (2014), and then selects the most important features as the final result. Hence, the outputs of the feature selection methods are interpretable Smialowski et al. (2010).

Depending on the availability of labels, existing feature selection methods can be partitioned into three subgroups, *i.e.,* supervised method Shi et al. (2018), semi-supervised method Benabdeslem and Hindawi (2011) and unsupervised method Li et al. (2014). Supervised method uses the labels to test the training model, so the importance of features can be evaluated. Unsupervised method mainly utilizes certain evaluation, such as rank ratio Nie et al. (2008), Laplace score He et al. (2005) and variance Dy and Brodley (2004), to evaluate the importance of the features or feature subsets, then selects the top k important features or the best representative feature subset. Semi-supervised methods are proposed to deal with the datasets including labeled and unlabeled samples. Semi-supervised method first learns the intrinsic structure from labeled samples to construct a basic model, and then utilizes the unlabeled samples to improve the former model.

In this paper, we mainly study unsupervised feature selection because labels in the real worlds are difficult to be collected Smialowski et al. (2010).

2.2. Self-paced learning

Robust statistic Tyler (2008); Huber (2011) has been introduced into the domain of machine learning to relieve the effect of outliers. Previous robust statistic methods can be divided into three groups, *i.e.,* M-estimation Negahban et al. (2009), half-quadratic minimization Du et al. (2013) and self-paced learning Jiang et al. (2014). M-estimation (*i.e.,* maxinum likelihood type estimation) is the statistical procedure of evaluating an M-estimator, where M-estimators are obtained through the minima of sums of functions of the data. Half-quadratic minimization is a general method based on the conjugate function theory to solve the convex or non-convex minimization optimization problem, and has been widely used in various domains, such as robust feature extraction Kongmunvattana and Tiamkaew (2012), mean-shift Yuan et al. (2011). Self-paced learning utilizes the theory of curriculum learning to establish a new machine learning framework from "simple" to "hard", which will be discussed in this paper in details.

The core concept of curriculum learning Bengio et al. (2009) is simulating the learning mode of human or animal, *i.e.,* first learning the simple knowledge, and then gradually increasing the learning difficulty, followed by learning more difficult and professional knowledge. Self-paced learning is a method of using mathematical expressions to express the curriculum learning theory. Self-paced learning defines the importance of samples according to the reconstruction error *i.e.,* the value of loss function. Specifically, self-paced learning method usually defines a sample that its reconstruction error is less than a certain threshold (or equivalent to zero) as an important sample (*i.e.,* a "simple" sample), and others as a secondary sample (*i.e.,* a "difficult" sample). In the process of self-paced learning, the first step is to select a part of the samples with small construction errors (less than a certain threshold) for training, so as to obtain accurate training models, then it adds more samples by gradually increasing threshold value to enhance the generalization ability of training model until the established model achieve stability.

Table 1. The notations used in this paper.

$\mathbf{X}$	the feature matrix of the training data
$\mathbf{x}$	a vector of $\mathbf{X}$
$\mathbf{x}^i$	the i-th row of $\mathbf{X}$
$\mathbf{x}_j$	the j-th column of $\mathbf{X}$
$x_{i,j}$	the element of the i-th row and the j-th column of $\mathbf{X}$
$\|\mathbf{X}\|_F$	the Frobenius norm of $\mathbf{X}$, *i.e.,* $\|\mathbf{X}\|_F = \sqrt{\sum_{i,j} x_{i,j}^2}$
$\|\mathbf{X}\|_{2,1}$	the $\ell_{2,1}$-norm of $\mathbf{X}$, *i.e.,* $\|\mathbf{X}\|_{2,1} = \sum_i \sqrt{\sum_j x_{i,j}^2}$
$\mathbf{X}^T$	the transpose of $\mathbf{X}$
$tr(\mathbf{X})$	the trace of $\mathbf{X}$
$\mathbf{X}^{-1}$	the inverse of $\mathbf{X}$

3. Approach

3.1. Notations

In this paper, matrices, vectors, and scalars are denoted as boldface uppercase letters, boldface lowercase letters, and normal italic letters, respectively. And other used notations are summarized in Table 1.

3.2. Unsupervised feature selection

Given a data matrix $\mathbf{X} = [\mathbf{x}^1, ..., \mathbf{x}^n] = [\mathbf{x}_1, ..., \mathbf{x}_d] \in \mathbb{R}^{n\times d}$, where n and d represent the numbers of samples and features, respectively. The objective function of traditional feature selection method can be written as follows:

$$\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} \tag{1}$$

where α is a sparse adjustment parameter, $\mathbf{Y} \in \mathbb{R}^{n\times c}$ and $\mathbf{W} \in \mathbb{R}^{d\times c}$ denote the response matrix (*i.e.,* the label matrix) and the feature weight matrix, respectively. Eq. (1) obtains the weight of features by fitting the data matrix and the response matrix, then utilizes the $\ell_{2,1}$-norm regularization to conduct sparsity on the weight matrix, which can reduce the weight of unimportant features.

In the real applications, the label of samples is usually difficult to be collected due to all kinds of reasons, such as time cost and budget cost. Hence, unsupervised feature selection is very popular in machine learning and data mining. Based on the property of features that each feature can be represented by a linear combination of other features, the self-representation relationship of the features is:

$$\mathbf{X} = \mathbf{X}\mathbf{W} + \mathbf{E} \tag{2}$$

where $\mathbf{W} \in \mathbb{R}^{d\times d}$ is the representation coefficient matrix, $\mathbf{E}$ is the reconstruction error. To obtain effective matrix $\mathbf{W}$, we employ the Frobenius norm to minimize the error, *i.e.,* $\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{X}\mathbf{W}\|_F^2$ to rewrite Eq. (1) as follows:

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{X}\mathbf{W}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} \tag{3}$$

In Eq. (3), the coefficient matrix $\mathbf{W}$ effectively reveals the potential relationship among features. Moreover, the $\ell_{2,1}$-norm regularization can make the rows in the matrix $\mathbf{W}$ corresponding to the unimportant features approximate to zero (or directly equal to zero). Hence, Eq. (3) can achieve the task of unsupervised feature selection.

3.3. Robust unsupervised feature selection

Although Eq. (3) can effectively remove redundant features, it uses all the samples to involving outliers into the feature selection model. To address this issue, self-paced learning method uses a sampling orderly manner to train the feature selection model, *i.e.,* first selecting important samples to establish the initial model, and then gradually adds the secondary samples to improve the generalization ability of the built model. Based on self-paced learning, we proposed a robust feature selection framework as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{v}} \sum_{i=1}^{n} v_i\|\mathbf{x}^i - \mathbf{x}^i\mathbf{W}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1} - \frac{1}{k}\sum_{i=1}^{n} v_i, \\ &s.t., v_i \in [0,1], i = 1, ..., n \end{aligned} \tag{4}$$

where the element v_i of vector $\mathbf{v} \in \mathbb{R}^{n\times 1}$ is the weight of the i-th sample, k is self-paced adjustment parameter. By adding self-paced learning regularization (*i.e.,* $\varphi(\mathbf{v}) = -\frac{1}{k}\sum_{i=1}^{n} v_i$), the proposed method can automatically assign the weight of selected samples as 1. Furthermore, the parameter k can be used to determine the samples involved in the training process during the self-paced learning process. When the value of k is large, self-paced learning tends to choose a sample with a smaller error for the training process. More samples will be selected with the decrease of the value of k. This process will be stopped until the value of k is less than a certain threshold. In this way, self-paced learning can effectively avoid outliers by avoiding them into the feature selection model or involving the feature selection model later.

Eq. (4) implement a "hard" sample sampling method by assigning a binary weight (*i.e.,* $v_i \in [0, 1]$) to each sample. However, since the outliers are not evenly distributed in all the samples, the hard threshold weight cannot accurately determine whether the method should select these samples. Compared with the hard threshold weight, soft weights are assigned to each sample with a real number between 0 and 1 (including 0 and 1), which can reflect the potential importance of training samples. By using soft threshold weight, the final objective function of our proposed method is obtained as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{v}} \sum_{i=1}^{n} v_i\|\mathbf{x}^i - \mathbf{x}^i\mathbf{W}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1} + \sum_{i=1}^{n} \frac{\beta^2}{v_i+\beta k}, \\ &s.t., 0 \le v_i \le 1, i = 1, ..., n \end{aligned} \tag{5}$$

where β is an interval control parameter, which controls the "fuzzy interval" between 0 and 1. Hence, a soft threshold weight can further avoid the influence of outliers by selecting samples accurately.

3.4. Optimization

Eq. (5) contain an auxiliary variable $\mathbf{v}$ and there is a convex but no smooth constraint on matrix $\mathbf{W}$ (*i.e.,* $\|\mathbf{W}\|_{2,1}$). In this paper, we utilize the IRLS (Iteratively Reweighted Least Squares) framework Daubechies et al. (2010) to propose an alternative optimization strategy to optimize Eq. (5), *i.e.,* update $\mathbf{v}$ by fixing $\mathbf{W}$ and update $\mathbf{W}$ by fixing $\mathbf{v}$. We list the pseudo code in Algorithm 1.

Algorithm 1: Pseudo code of solving Eq.(5).

Input: $\mathbf{X} \in \mathbb{R}^{n\times d^v}$ $(v = 1, ..., m)$, control parameters α, β, k and step parameter $\mu > 1$;
Output: $\mathbf{W} \in \mathbb{R}^{d\times d}$;
Calculate loss function value $\mathbf{L} \in \mathbb{R}^{n\times 1}$ via Eq. (3) ;
Initialize t=0 ;
repeat
  Update $\mathbf{v}^{(t+1)}$ via Eq. (13) ;
  Update $\mathbf{W}^{(t+1)}$ via Algorithm 2 ;
  Update $k = \frac{k}{\mu}$, $t = t + 1$;
until *convergence*;

Algorithm 2: Pseudo code of solving $\mathbf{W}$.

Input: $\mathbf{X} \in \mathbb{R}^{n\times d^v}$ $(v = 1, ..., m)$, control parameter α ;
Output: $\mathbf{W} \in \mathbb{R}^{d\times d}$;
Initialize t=0 ;
Initialize $\mathbf{D}^{(0)}$ as random diagonal matrix;
repeat
  Update $\mathbf{W}^{(t+1)}$ via Eq. (10) ;
  Update $\mathbf{D}^{(t+1)}$ via Eq. (9);
  Update $t = t + 1$;
until *convergence*;

- Update $\mathbf{W}$ by fixing $\mathbf{v}$

  While fixing $\mathbf{v}$, the objective function Eq. (5) becomes:

$$\min_{\mathbf{W}} \sum_{i=1}^{n} v_i \|\mathbf{x}^i - \mathbf{x}^i\mathbf{W}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1} \tag{6}$$

  To facilitate the optimization, we rewrite Eq. (6) as:

$$\min_{\mathbf{W}} \|\mathbf{G} - \mathbf{G}\mathbf{W}\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} \tag{7}$$

  where $\mathbf{G} = \mathbf{U}\mathbf{X}$ and $\mathbf{U} = diag(\sqrt{\mathbf{v}})$. Eq. (7) can be seen as a function of $\mathbf{W}$. Hence, we set the derivative of Eq. (7) with respect to $\mathbf{W}$ to 0:

$$-\mathbf{G}^T\mathbf{G} + \mathbf{G}^T\mathbf{G}\mathbf{W} + \alpha\mathbf{D}\mathbf{W} = 0 \tag{8}$$

  where $\mathbf{D}$ is the diagonal matrix, its i-th element is:

$$d_{i,i} = \frac{1}{2\|\mathbf{W}^i\|_2}, s.t., i = 1, ...d \tag{9}$$

  where $\mathbf{W}^i$ is the i-th row of $\mathbf{W}$. After a simple mathematical transformation, the final solution is:

$$\mathbf{W} = (\mathbf{G}^T\mathbf{G} + \alpha\mathbf{D})^{-1}\mathbf{G}^T\mathbf{G} \tag{10}$$

- Update $\mathbf{v}$ by fixing $\mathbf{W}$

  While fixing $\mathbf{W}$, the objective function Eq. (5) can be written as follows:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{v}} \sum_{i=1}^{n} v_i\|\mathbf{x}^i - \mathbf{x}^i\mathbf{W}\|_2^2 + \sum_{i=1}^{n} \frac{\beta^2}{v_i+\beta k}, \\ &s.t., 0 \le v_i \le 1, i = 1, ..., n \end{aligned} \tag{11}$$

  By defining $\mathbf{L} = \sum_{i=1}^{n} L_i = \sum_{i=1}^{n} \|\mathbf{x}^i - \mathbf{x}^i\mathbf{W}\|_2^2$, we have:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{v}} \sum_{i=1}^{n} v_i L_i + \sum_{i=1}^{n} \frac{\beta^2}{v_i+\beta k}, \\ &s.t., 0 \le v_i \le 1, i = 1, ..., n \end{aligned} \tag{12}$$

  According to Eq. (12), the closed form solution of v_i is:

$$v_i = \begin{cases} 1 & if \;\; L_i \le \frac{1}{\sqrt{k+1/\beta}}, \\ 0 & if \;\; L_i \le \frac{1}{\sqrt{k}}, \\ \beta(\frac{1}{L_i} - k) & otherwise. \end{cases} \tag{13}$$

3.5. Convergence analysis

We denote the t-th iteration of $\mathbf{v}$ and $\mathbf{W}$ as $\mathbf{v}^{(t)}$ and $\mathbf{W}^{(t)}$, respectively. Based on Algorithm 1, Eq. (5) can be written as follows:

$$\begin{aligned} E(\mathbf{W}^{(t)}, \mathbf{v}^{(t)}) = &\sum_{i=1}^{n} v_i^{(t)}\|\mathbf{x}^i - \mathbf{x}^i\mathbf{W}^{(t)}\|_2^2 \\ &+ \alpha\|\mathbf{W}^{(t)}\|_{2,1} + \sum_{i=1}^{n} \frac{\beta^2}{v_i^{(t)}+\beta k} \end{aligned} \tag{14}$$

According to self-paced learning theory Meng et al. (2015) and the fixed $\mathbf{W}^{(t)}$, we have:

$$E(\mathbf{W}^{(t)}, \mathbf{v}^{(t+1)}) \le E(\mathbf{W}^{(t)}, \mathbf{v}^{(t)}) \tag{15}$$

With the fixed $\mathbf{v}^{(t+1)}$, based on the IRLS framework, we have the following inequality:

$$E(\mathbf{W}^{(t+1)}, \mathbf{v}^{(t+1)}) \le E(\mathbf{W}^{(t)}, \mathbf{v}^{(t+1)}) \tag{16}$$

Integrating Eq. (15) into Eq. (16), the final inequality is:

$$E(\mathbf{W}^{(t+1)}, \mathbf{v}^{(t+1)}) \le E(\mathbf{W}^{(t)}, \mathbf{v}^{(t)}) \tag{17}$$

Eq. (14) is non-increasing at each iteration according to Eq. (17). Thus, the proposed Algorithm 1 converges.

3.6. Parameter determination

In Eq. (13), we can know that the value of the parameter k and β determines the choice of samples in the learning process. Hence, selecting the appropriate parameters can effectively improve the proposed algorithm. In this paper, we propose a simple and effective approach to solve the problem of parameter determination.

By denoting L_m as the maximum loss function value of initially selected samples, we have:

$$L_m = \frac{1}{\sqrt{k+1/\beta}} \tag{18}$$

To simplify the calculation, we let $k = \frac{1}{\beta}$ and obtain:

$$k = \frac{1}{2L_m^2} \tag{19}$$

By integrating Eq. (19) with Eq. (18), we have:

$$\beta = 2L_m^2 \tag{20}$$

According to Eq. (19) and Eq. (20), our proposed method can obtain the appropriate parameters k and β according to the

Table 2. The information of Datasets.

Datasets	Samples	Dimensions	Classes	Type
Umist	575	644	20	image
USPS	9298	256	10	image
Jaffe	213	1024	10	image
Coil	1440	1024	20	image
Isolet	1560	617	26	text
DBworld	64	4702	2	text
Ecoli	336	343	8	biological
Colon	62	2000	2	biological

number of samples initially selected, therefore, the dependence of the proposed algorithm on the parameters can be reduced. After the parameters k and β are fixed, other parameters still need to be adjusted. In this paper, we utilize the cross-validation approach to estimate them.

4. Experiments

In this section, we evaluated our proposed UFS_SP method and six comparison methods on eight data sets in terms of clustering performance. Specially, we first employed each feature selection method to choose the new feature subsets from original data sets, and then utilized k-means clustering to evaluate the selected subsets.

4.1. Datasets and comparison methods

The datasets (such as Ecoli and Isolet) and the datasets (such as Colon, USPS, Coil and DBWorld) are from UCI Machine Learning Repository[1] and the website of Feature Selection Data sets[2], respectively. Datasets Umist and Jaffe are from website of the University of Sheffield[3] and the paper Lyons et al. (1998). We summarized the detail of all datasets in Table 2.

We compared our proposed method (UFS_SP) with six comparison methods and listed the details of the them as follows:

- **Baseline** directly performs k-means clustering on the original data. In this paper, we used baseline as a criterion to evaluate the actual value of the feature selection method.
- Feature Selection Robust $\ell_{2,0}$-norm Augented Lagrangian Multiplier (**FSR_ALM** Cai et al. (2013)) employs an $\ell_{2,1}$-norm regularization to deal with the reconstruction error and adds an $\ell_{2,0}$-norm regularization to conduct sparsity.
- Coupled Dicitionary Learning for Unsupervised Feature Selection (**CDLFS** Zhu et al. (2016a)) uses the coupled analysis-synthesis dictionary learning to implement unsupervised feature selection and employs an $\ell_{2,p}$-norm regularization on the analysis dictionary matrix to conduct sparsity.
- Convex Semi-supervised multi-label Feature Selection (**CSFS** Chang et al. (2014)) uses the least square regression to measure the reconstruction error and conduct group sparsity on feature weight matrix by an $\ell_{2,1}$-norm regularization.
- Regularized Self-Representation (**RSR** Zhu et al. (2015)) uses the feature level self-representation property to mine the relationship between the features and construct the weight coefficient matrix, then utilizes conduct sparsity on the coefficient matrix via an $\ell_{2,1}$-norm regularization.
- General Sparsity Regularized (**GSR** Peng and Fan (2017)) proposes an $\ell_{2,r}$-norm regularization on the loss function to reduce the effect of outliers, and employs an $\ell_{2,p}$-norm regularization to achieve sparsity.

In the comparison methods, FSR_ALM and GSR belong to supervised learning methods, CSFS belongs to semi_supervised learning method, and CDLFS and RSR belong to unsupervised learning method. In this paper, we verified the effectiveness of our proposed method by comparing feature selection methods based on different learning strategies.

4.2. Experimental setting

In our experiments, we first used all the feature selection methods to select features, and then conducted k-means clustering method on the selected features to implement clustering tasks.

It is noteworthy that the results of k-means clustering are random. In our experiment, we utilized the 10-fold cross-validation scheme to repeat k-means clustering method ten times on the selected subset. We used the average of these 10 clustering results as the final result. We set the ranges of the parameter α of the proposed method in Eq. (5) as $\{10^{-3}, 10^{-2}, ..., 10^{3}\}$ and set the parameter $\mu > 1$. For other comparison methods, we are set in strictly according to their corresponding literature.

We utilized ACC (accuracy), NMI (normalized mutual information), Purity and ARI (adjusted rand index) to evaluate the performance of all the methods on eight benchmark datasets. We listed the definition of four evaluation metric as below:

- ACC: Accuracy indicates the percentage of correctly classified samples, *i.e.,* :

$$ACC = N_c/N \tag{21}$$

  where N denote the number of samples and N_c is correctly classified samples.

- NMI: Normalized mutual information uncovers a correlation between the sample and the label. When the value of NMI is 1, the sample has the highest correlation with the label. The definition of NMI is:

$$NMI = 2\frac{I(\mathbf{X},\mathbf{Y})}{H(\mathbf{X})+H(\mathbf{Y})} \tag{22}$$

  where $I(\mathbf{X}, \mathbf{Y})$ denotes mutual information (MI) between the samples and the labels, $H(\cdot)$ is the entropy.

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

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

[3] https://www.sheffield.ac.uk/eee/research/iel/research/face.

Table 3. Clustering accuracy on benchmark data sets.

Data sets	Umist	USPS	Jaffe	Isolet	DBworld	Ecoli	Coil	Colon	Average
Baseline	0.4400	0.5763	0.7230	0.4885	0.8125	0.5238	0.6097	0.3871	0.5701
FSR_ALM	0.4678	0.6594	0.7606	0.5679	0.8594	0.6042	0.6340	0.4556	0.6261
CDLFS	0.4609	0.6357	0.7700	0.5716	0.8378	0.6399	0.6235	0.4577	0.6246
CSFS	0.4539	0.6305	0.7887	0.5577	0.8559	0.5923	0.6257	0.4516	0.6185
RSR	0.4643	0.6180	0.7793	0.5538	0.8594	0.6339	0.6319	0.4686	0.6262
GSR	0.4730	0.6684	0.8122	0.5641	0.8750	0.6250	0.6340	0.4776	0.6412
Proposed	**0.5061**	**0.7207**	**0.8967**	**0.6147**	**0.9063**	**0.7074**	**0.6681**	**0.5484**	**0.6961**

Table 4. The results of NMI, Purity and ARI on benchmark data sets.

Datasets	Umist			USPS			Jaffe			Isolet		
	NMI	Purity	ARI	NMI	Purity	ARI	NMI	Purity	ARI	NMI	Purity	ARI
Baseline	0.6492	0.5061	0.3541	0.5721	0.6651	0.4685	0.7966	0.7465	0.6422	0.7094	0.5641	0.4650
FSR_ALM	0.6364	0.5009	0.3368	0.6107	**0.7261**	0.5279	0.7929	0.7906	0.6688	0.7337	0.6013	0.5099
CDLFS	0.6361	0.5217	0.3365	0.6082	0.7053	0.5276	0.7828	0.7840	0.6209	0.7257	0.5971	0.5124
CSFS	0.6468	0.4922	0.3506	0.6065	0.7051	0.5250	0.8638	0.8216	0.7331	0.7446	0.6212	0.5050
RSR	0.6343	0.4904	0.3550	0.6073	0.7163	0.5193	0.8050	0.7793	0.6872	0.7335	0.6179	0.5091
GSR	0.6417	0.5026	0.3496	**0.6132**	0.7256	0.5235	0.8529	0.8498	0.7383	0.7286	0.5942	0.5201
Proposed	**0.6912**	**0.5565**	**0.4188**	0.6111	0.7207	**0.5463**	**0.9019**	**0.8967**	**0.8165**	**0.7633**	**0.6449**	**0.5702**

Datasets	DBworld			Ecoli			Coil			Colon		
	NMI	Purity	ARI	NMI	Purity	ARI	NMI	Purity	ARI	NMI	Purity	ARI
Baseline	0.3084	0.8125	0.3812	0.4721	0.7768	0.3520	0.7248	0.6472	0.5120	0.0014	0.6452	0.0148
FSR_ALM	0.4230	0.8594	0.5091	0.4696	0.7054	0.4706	0.7423	0.6396	0.5780	0.0199	0.6354	0.0110
CDLFS	0.3868	0.8438	0.4645	**0.5206**	0.7500	0.5411	0.7463	0.6811	0.5573	0.0510	0.6835	0.0416
CSFS	0.4203	0.8559	0.5056	0.4752	0.7470	0.4448	0.7544	0.6472	0.5977	0.0157	0.6452	0.0252
RSR	0.4212	0.6179	0.5021	0.4704	0.7173	**0.5785**	0.7345	0.6694	0.5303	**0.0513**	**0.6914**	**0.0423**
GSR	0.4539	0.8750	0.5555	0.4811	0.7619	0.4436	0.7094	0.6681	0.5120	0.0132	0.6652	0.0322
Proposed	**0.5489**	**0.9063**	**0.6547**	0.5196	**0.7976**	0.5360	**0.7933**	**0.6965**	**0.6377**	0.0466	0.6774	0.0262

- Purity: Purity reflects the ratio of correctly classified samples in each cluster, which definition is:

$$Purity = \sum_{i=1}^{K} \frac{m_i}{t} P_i \tag{23}$$

where K is number of clusters, m_i and t are number of i-th cluster of samples and all samples, respectively. P_i is the maximum value that the probability of the member of i-th cluster belongs to each class.

- ARI: Adjusted rand index is a measure of the similarity between the prediction labels and the real labels, *i.e.,* :

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]} \tag{24}$$

where RI is a rand index and $E[RI]$ denotes the expectation of the rand index.

4.3. Experiment results and analysis

We listed the clustering performance of all methods on eight benchmark data sets in Tables 3-4. We also discussed the influence of sample selection of our proposed method on feature selection model in Fig.1 .

From Table 3 we can see that the clustering accuracy of our proposed method is excellent than all comparison methods on all data sets. For example, our method improved on average by 12.6%, 7%,7.15%, 7.76%, 6.99%, and 5,49%, respectively, compared with Baseline, FSR_ALM, CDLFS, CSFS, RSR, and GSR. Moreover, the ACC result of our proposed method is better than all the methods by 8.45% on the data set Jaffe (best performance on accuracy), and increased by 3.13% on the data set DBworld (worst performance on accuracy). The reasons are that the proposed method 1) achieves the function of feature selection because it clustering accuracy more outstanding compared with non feature selection method; and 2) is more outstanding than other comparison methods due to self-paced learning regularization has more effective than traditional method on handling outliers. Furthermore, in Table 4 the value of three evaluation indexes, *i.e.,* NMI, Purity and ARI, of our proposed method are highest on the datasets, such as Umist, Jaffe, Isolet, DBworld and Coil. This further proved that the our proposed method is superior to other comparison methods.

In Fig.1, we use ACC to evaluate the performance of the feature selection model (*i.e.,* Eq. (3)) based on different sample sampling methods (*i.e.,* Non sampling, Random sampling and Self-paced learning). From Fig.1 we can see that 1) random sampling may obtain the best performance than non-sampling method when it selected "correct" sample subset (*i.e.,* that not include outliers) to train the model. 2) self-paced learning method obtained the excellent performance than random sampling and non-sampling methods.

4.4. Parameter sensitivity and convergence analysis

After the parameters k and β were fixed, we still need to tune the parameters α and μ. In this paper, we set the range of α and μ as $\{10^{-3}, ..., 10^{3}\}$ and $\{1.1, 1.15, ..., 1.35, 1.4\}$, respectively, and listed the corresponding results in Fig.2. As showed in Fig.2, we can find that our proposed method is sensitive to the setting of parameters. That is, the result of our method can be improved by tuning parameters. Hence, tuning the parameters is necessary to our method.

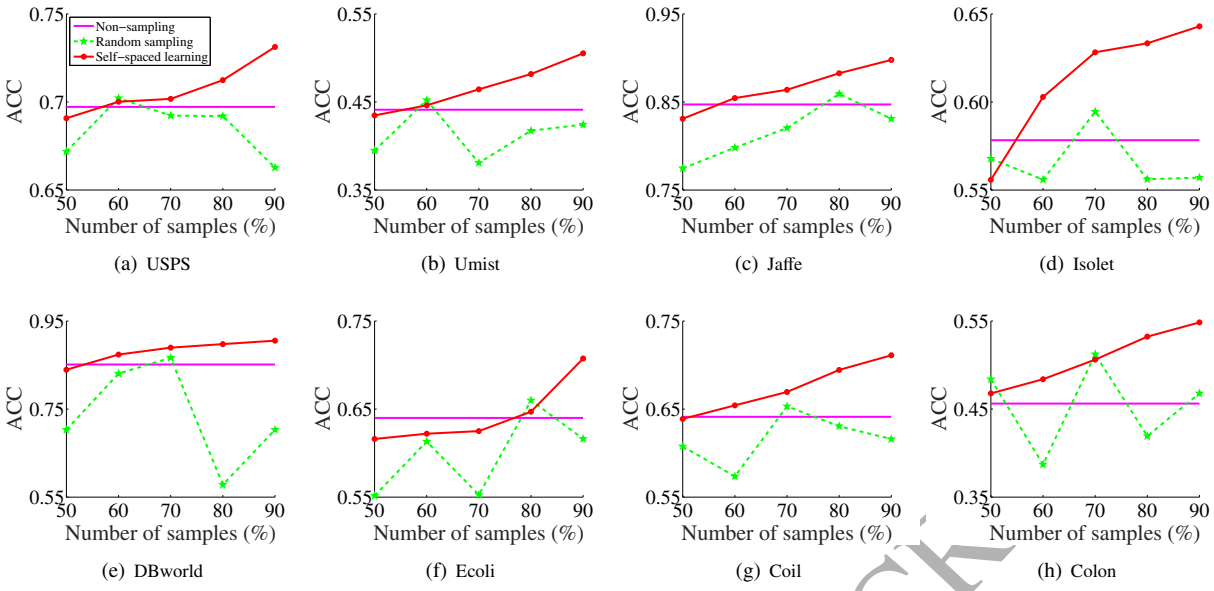

Fig. 1. ACC result of proposed methods on all data sets at different number of samples.

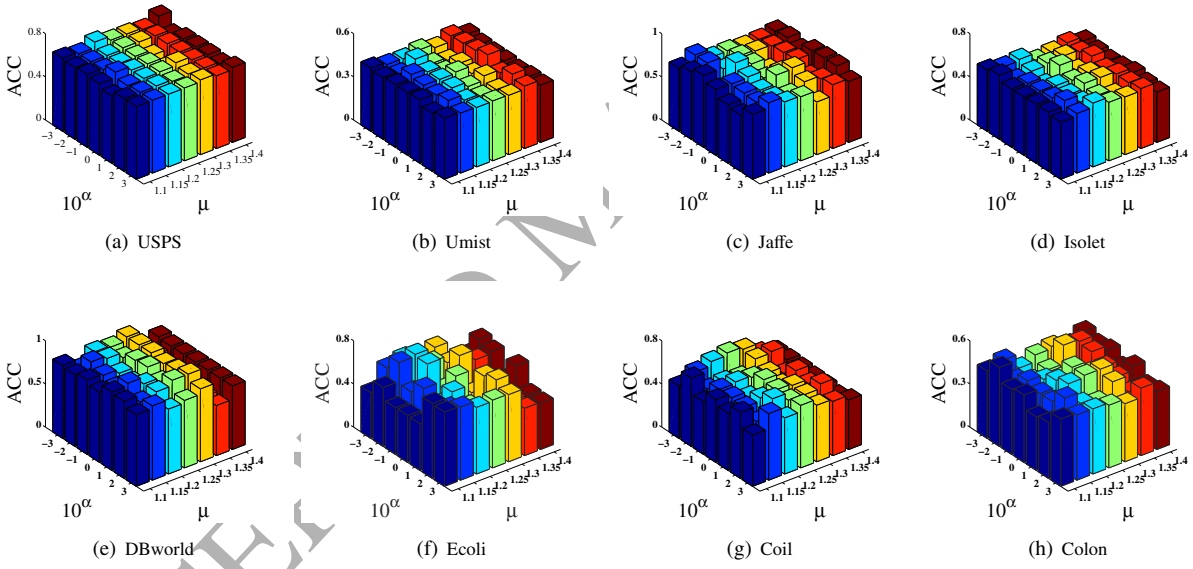

Fig. 2. ACC result of the proposed method at different parameters setting on the variables α and μ.

Fig.3 showed the behavior of the objective function value of our proposed method with the increase of the iterations. In experiments, we set the stop criteria of both Algorithm 1 and Algorithm 2 as 10^{-5}, *i.e.*, $\frac{\|obj(t+1)-obj(t)\|_2^2}{obj(t)} \leq 10^{-5}$, where $obj(t)$ represents the objective value of t-th iteration of Eq. (5). From Fig.3 we can find that 1) the proposed objective function values is monotonously decreased until the proposed Algorithm 1 achieves converges; 2) the iterations of the proposed Algorithm 1 reach the convergence are less than 20. Hence, the proposed Algorithm 1 is very efficient.

5. Conclusion

In this paper, we proposed a novel unsupervised feature selection method by embedding a self-paced learning regularization into the sparse feature selection model. Specifically, we integrated feature self representation, self-paced learning regularization and an $\ell_{2,1}$-norm regularization into a unified framework. Experimental results showed that our proposed method can achieve the best clustering performance compared with all the comparison methods on real data sets.

In the future work, we will try to add graph learning to extend our proposed framework to conduct spectral feature selection

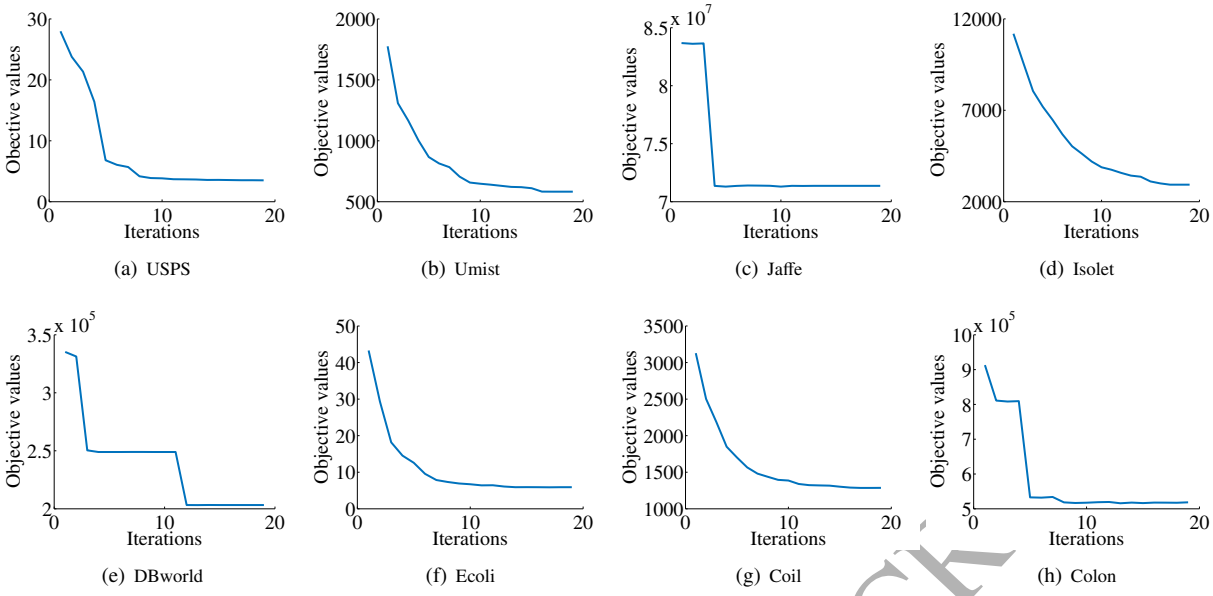

Fig. 3. The convergence of Algorithm 1 on all data sets.

since graph learning can further enhance the effect of feature selection model Zheng et al.; Lei and Zhu (2017).

Acknowledgements

This work is partially supported by the China Key Research Program (Grant No: 2016YFB1000905); the Natural Science Foundation of China (Grants No: 661573270 and 61672177); the Project of Guangxi Science and Technology (GuiKeAD17195062); the Guangxi Natural Science Foundation (Grant No: 2015GXNSFCB139011); Innovation Project of Guangxi Graduate Education (Grant No: YCSW2018093); the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing; the Guangxi Bagui Teams for Innovation and Research; the Guangxi High Institutions Program of Introducing 100 High-Level Overseas Talents; and the Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (18-A-01-01).

References

Benabdeslem, K., Hindawi, M., 2011. Constrained laplacian score for semi-supervised feature selection, in: Machine Learning and Knowledge Discovery in Databases - European Conference, pp. 204–218.

Bengio, Y., Louradour, J., Collobert, R., Weston, J., 2009. Curriculum learning. Journal of the American Podiatry Association 60, 6.

Cai, X., Nie, F., Huang, H., 2013. Exact top-k feature selection via $\ell_{2,0}$-norm constraint, in: International Joint Conference on Artificial Intelligence, pp. 1240–1246.

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.

Daubechies, I., Devore, R., Fornasier, M., Gunturk, S., 2010. Iteratively reweighted least squares minimization for sparse recovery. Communications on Pure & Applied Mathematics 63, 1–38.

Du, L., Li, X., Shen, Y.D., 2013. Robust nonnegative matrix factorization via half-quadratic minimization, in: IEEE International Conference on Data Mining, pp. 201–210.

Dy, J.G., Brodley, C.E., 2004. Feature selection for unsupervised learning. Journal of Machine Learning Research 5, 845–889.

Gao, L., Guo, Z., Zhang, H., Xu, X., Shen, H.T., 2017. Video captioning with attention-based lstm and semantic consistency. IEEE transactions on Multimedia 19, 2045–2055.

Gheyas, I.A., Smith, L.S., 2010. Feature subset selection in large dimensionality domains. Pattern Recognition 43, 5–13.

Gutlein, M., Frank, E., Hall, M., Karwath, A., 2009. Large-scale attribute selection using wrappers, in: Computational Intelligence and Data Mining, pp. 332–339.

He, X., Cai, D., Niyogi, P., 2005. Laplacian score for feature selection., in: International Conference on Neural Information Processing Systems, pp. 507–514.

Hu, R., Zhu, X., Cheng, D., He, W., Yan, Y., Song, J., Zhang, S., 2017. Graph self-representation method for unsupervised feature selection. Neurocomputing 220, 130–137.

Huber, P.J., 2011. Robust statistics, in: International Encyclopedia of Statistical Science, pp. 1248–1251.

Jiang, L., Meng, D., Yu, S.I., Lan, Z., Shan, S., Hauptmann, A., 2014. Self-paced learning with diversity, in: Neural Information Processing Systems.

Kongmunvattana, A., Tiamkaew, E., 2012. Spa: A scalable per-address branch predictor, in: IEEE International Conference on Systems, Man, and Cybernetics, pp. 975–980.

Kumar, M.P., Packer, B., Koller, D., 2010. Self-paced learning for latent variable models, in: International Conference on Neural Information Processing Systems, pp. 1189–1197.

Lei, C., Zhu, X., 2017. Unsupervised feature selection via local structure learning and sparse learning. Multimedia Tools and Applications , https://doi.org/10.1007/s11042–017–5381–7.

Li, Z., Liu, J., Yang, Y., Zhou, X., Lu, H., 2014. Clustering-guided sparse structural learning for unsupervised feature selection. IEEE Transactions on Knowledge & Data Engineering 26, 2138–2150.

Lin, L., Wang, K., Meng, D., Zuo, W., Zhang, L., 2018. Active self-paced learning for cost-effective and progressive face identification. IEEE transactions on pattern analysis and machine intelligence 40, 7–19.

Liu, X., Wang, L., Zhang, J., Yin, J., Liu, H., 2017. Global and local structure preservation for feature selection. IEEE Transactions on Neural Networks & Learning Systems 25, 1083–1095.

Liu, Y., Nie, F., Wu, J., Chen, L., 2013. Efficient semi-supervised feature selection with noise insensitive trace ratio criterion. Neurocomputing 105, 12–18.

Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J., 1998. Coding facial expressions with gabor wavelets, in: IEEE International Conference on Automatic

Face and Gesture Recognition, pp. 200–205.

Meng, D., Zhao, Q., Jiang, L., 2015. What objective does self-paced learning indeed optimize? Computer Science .

Meng, D., Zhao, Q., Jiang, L., 2017. A theoretical understanding of self-paced learning. Information Sciences 414, 319–328.

Negahban, S.N., Ravikumar, P., Wainwright, M.J., Yu, B., 2009. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. Statistical Science 27, 538–557.

Nie, F., Xiang, S., Jia, Y., Zhang, C., Yan, S., 2008. Trace ratio criterion for feature selection, in: National Conference on Artificial Intelligence, pp. 671–676.

Peng, H., Fan, Y., 2017. A general framework for sparsity regularized feature selection via iteratively reweighted least square minimization, in: Thirty-First AAAI Conference on Artificial Intelligence, pp. 2471–2477.

Shah, S.A., Koltun, V., 2017. Robust continuous clustering. Proceedings of the National Academy of Sciences of the United States of America 114, 9814.

Shi, Y., Miao, J., Wang, Z., Zhang, P., Niu, L., 2018. Feature selection with $\ell_{2,1-2}$ regularization. IEEE Transactions on Neural Networks & Learning Systems PP, 1–16.

Smialowski, P., Frishman, D., Kramer, S., 2010. Pitfalls of supervised feature selection. Bioinformatics 26, 440.

Song, J., Gao, L., Liu, L., Zhu, X., Sebe, N., 2018. Quantization-based hashing: a general framework for scalable image and video retrieval. Pattern Recognition 75, 175–187.

Song, J., Gao, L., Nie, F., Shen, H.T., Yan, Y., Sebe, N., 2016a. Optimized graph learning using partial tags and multiple features for image and video annotation. IEEE Trans. Image Processing 25, 4999–5011.

Song, J., Shen, H.T., Wang, J., Huang, Z., Sebe, N., Wang, J., 2016b. A distance computation free search scheme for binary code databases. IEEE transactions on Multimedia 18, 484–495.

Tan, M., Wang, L., Tsang, I.W., 2010. Learning sparse svm for feature selection on very high dimensional datasets, in: International Conference on International Conference on Machine Learning, pp. 1047–1054.

Tyler, D.E., 2008. Robust statistics: Theory and methods.

Unler, A., Murat, A., Chinnam, R.B., 2011. mr^2pso : A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. Information Sciences 181, 4625–4641.

Yang, Y., Duan, Y., Wang, X., Huang, Z., Xie, N., Shen, H.T., 2018. Hierarchical multi-clue modelling for poi popularity prediction with heterogeneous tourist information. IEEE Transactions on Knowledge and Data Engineering .

Yuan, X.T., Hu, B.G., He, R., 2011. Agglomerative mean-shift clustering. IEEE Transactions on Knowledge & Data Engineering 24, 209–219.

Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R., 2018. Efficient knn classification with different numbers of nearest neighbors. IEEE Transactions on Neural Networks and Learning Systems 29, 1774–1785.

Zhang, Y., Jin, J., Qing, X., Wang, B., Wang, X., 2012a. Lasso based stimulus frequency recognition model for ssvep bcis. Biomedical Signal Processing and Control 7, 104–111.

Zhang, Y., Wang, Y., Jin, J., Wang, X., 2017. Sparse bayesian learning for obtaining sparsity of eeg frequency bands based feature vectors in motor imagery classification. International Journal of Neural Systems 27, 1650032.

Zhang, Y., Zhao, Q., Jin, J., Wang, X., Cichocki, A., 2012b. A novel bci based on erp components sensitive to configural processing of human faces. Journal of neural engineering 9, 026018.

Zhang, Z., Hancock, E.R., 2017. A graph-based approach to feature selection. Lecture Notes in Computer Science 6658, 205–214.

Zheng, W., Zhu, X., Zhu, Y., Hu, R., Lei, C., . Dynamic graph learning for spectral feature selection. Multimedia Tools and Applications .

Zhu, P., Hu, Q., Zhang, C., Zuo, W., 2016a. Coupled dictionary learning for unsupervised feature selection, in: Thirtieth AAAI Conference on Artificial Intelligence.

Zhu, P., Zuo, W., Zhang, L., Hu, Q., Shiu, S.C.K., 2015. Unsupervised feature selection by regularized self-representation. Pattern Recognition 48, 438–446.

Zhu, X., Jin, Z., Ji, R., 2016b. Learning high-dimensional multimedia data. Multimedia Systems , 1–3.

Zhu, X., Li, X., Zhang, S., 2016c. Block-row sparse multiview multilabel learning for image classification. IEEE transactions on cybernetics 46, 450–461.

Zhu, X., Li, X., Zhang, S., Ju, C., Wu, X., 2017a. Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE transactions on neural networks and learning systems 28, 1263–1275.

Zhu, X., Li, X., Zhang, S., Xu, Z., Yu, L., Wang, C., 2017b. Graph pca hashing for similarity search. IEEE Transactions on Multimedia 19, 2033–2044.

Zhu, X., Zhang, L., Huang, Z., 2014. A sparse embedding and least variance encoding approach to hashing. IEEE transactions on image processing 23, 3737–3750.

Zhu, X., Zhang, S., Hu, R., Zhu, Y., et al., . Local and global structure preservation for robust unsupervised spectral feature selection. IEEE Transactions on Knowledge and Data Engineering 30, 517–529.

Zhu, X., Zhang, S., Jin, Z., Zhang, Z., Xu, Z., 2010. Missing value estimation for mixed-attribute data sets. IEEE Transactions on Knowledge & Data Engineering 23, 110–121.

Zhu, Y., Kim, M., Zhu, X., Yan, J., Kaufer, D., Wu, G., 2017c. Personalized diagnosis for alzheimers disease, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 205–213.

Zhu, Y., Lucey, S., 2015. Convolutional sparse coding for trajectory reconstruction. IEEE transactions on pattern analysis and machine intelligence 37, 529–540.

Zhu, Y., Zhu, X., Kim, M., Kaufer, D., Wu, G., 2017d. A novel dynamic hyper-graph inference framework for computer assisted diagnosis of neurodiseases, in: IPMI, pp. 158–169.