Unsupervised feature selection by self-paced learning regularization

Wei Zheng\textsuperscript{a,b}, Xiaofeng Zhu\textsuperscript{a,b,d,*}, Guoqiu Wen\textsuperscript{a,b}, Yonghua Zhu\textsuperscript{c}, Hao Yu\textsuperscript{a,b}, Jiangzhang Gan\textsuperscript{a,b}

\textsuperscript{a} College of Computer Science and Information Technology, Guangxi Normal University, Guilin, Guangxi 541004, PR China
\textsuperscript{b} Guangxi Key Lab of Multi-Source Information Mining and Security, Guilin, Guangxi 541004, PR China
\textsuperscript{c} School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi 530004, PR China
\textsuperscript{d} Institute of Natural and Mathematical Sciences, Massey University Albany Campus, Auckland 0632, New Zealand

\textbf{A R T I C L E   I N F O}

Article history:
Available online xxx

MSC:
41A05
41A10
65D05
65D17

Keywords:
Feature selection
Self-paced learning
Robust statistic

\textbf{A B S T R A C T}

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.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Big data has been widely appearing in various fields, such as pattern recognition and machine learning [8,31,39,40,48]. 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 [32,33,37,42]. Moreover, the unimportant features in big data easily lead to the issue of curse of dimensionality [41,50]. 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 [17,51,52,54].

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 [9,53], wrapper method [10,36], and embedded method [20,43]. 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 [12,44].

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 [16,28]. 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 [12,17] as well as employs the $\ell_2, 1$-norm regularization to penalize the coefficient matrix so 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 [16,28] 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

\textsuperscript{*} Corresponding author at: Guangxi Key Lab of Multi-Source Information Mining and Security, Guilin, Guangxi 541004, PR China.
\textit{E-mail addresses:} zwzgxsu163.com (W. Zheng), x.zhu@massey.ac.nz (X. Zhu).

https://doi.org/10.1016/j.patrec.2018.06.029
0167-8655/© 2018 Elsevier B.V. All rights reserved.
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 [16] 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 [17,48,52] and self-paced learning [16,19,24].

2.1. Feature selection

As an important dimensionality reduction technique, feature selection tries to find a most representative feature subset from original features [47,49,55]. Different from subspace learning [50,56], 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 criterion, such as evaluation score [121] and sparse learning [16,34], and then selects the most important features as the final result. Hence, the outputs of the feature selection methods are interpretable [30].

Depending on the availability of labels, existing feature selection methods can be partitioned into three subgroups, i.e., supervised method [29], semi-supervised method [1] and unsupervised method [18]. 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 [26], Laplace score [11] and variance [7], 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 [30].

2.2. Self-paced learning

Robust statistic [13,35] 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 [25], half-quadratic minimization [6] and self-paced learning [14]. M-estimation (i.e., maximum 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 [15], mean-shift [38]. 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 [2] 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 achieves stability.

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 $X = [x_1, \ldots, x^n] = [x_1, \ldots, x_d] \in \mathbb{R}^{n \times d}$, 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:

$$\min_{W} \|Y - XW\|_F^2 + \alpha \|W\|_{2,1} \quad (1)$$

where $\alpha$ is a sparse adjustment parameter, $Y \in \mathbb{R}^{n \times c}$ and $W \in \mathbb{R}^{d \times c}$ 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 $\ell_2, \gamma$-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:

$$X = XW + E$$

(2)

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

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

(3)

In Eq. (3), the coefficient matrix $W$ effectively reveals the potential relationship among features. Moreover, the $\ell_{2,1}$-norm regularization can make the rows in the matrix $W$ corresponding to the important 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:

$$\min_{W, \nu} \sum_{i=1}^{n} v_{i} \|x_{i} - x_{i}W\|_{2}^2 + \alpha \|W\|_{2,1} - \frac{1}{L} \sum_{i=1}^{n} v_{i},$$

s.t. $v_{i} \in [0, 1], i = 1, \ldots, n$ (4)

where the element $v_{i}$ of vector $\nu \in \mathbb{R}^{n \times 1}$ is the weight of the $i$-th sample, $k$ is self-paced adjustment parameter. By adding self-paced learning regularization (i.e., $\psi(\nu) = -\frac{1}{L} \sum_{i=1}^{n} v_{i}$), 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:

$$\min_{W, \nu} \sum_{i=1}^{n} v_{i} \|x_{i} - x_{i}W\|_{2}^2 + \alpha \|W\|_{2,1} + \sum_{i=1}^{n} \frac{\beta^2}{v_{i} + \beta k},$$

s.t. $0 \leq v_{i} \leq 1, i = 1, \ldots, n$ (5)

where $\beta$ 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 $\nu$ and there is a convex but no smooth constraint on matrix $W$ (i.e., $\|W\|_{2,1}$). In this paper, we utilize the IRLS (Iteratively Reweighted Least Squares) framework [5] to propose an alternative optimization strategy to optimize Eq. (5), i.e., update $\nu$ by fixing $W$ and update $W$ by fixing $\nu$. We list the pseudo code in Algorithm 1.

- **Update $W$ by fixing $\nu$**
  While fixing $\nu$, the objective function Eq. (5) becomes:

$$\min_{W} \sum_{i=1}^{n} v_{i} \|x_{i} - x_{i}W\|_{2}^2 + \alpha \|W\|_{2,1}$$

(6)

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

$$\min_{W} \|G - GW\|_{2}^2 + \alpha \|W\|_{2,1}$$

(7)

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

$$-G^T G + G^T GW + \alpha DW = 0$$

(8)

where $D$ is the diagonal matrix, its ith element is:

$$d_{ii} = \frac{1}{\nu_i^{1/2}}, \text{s.t. } i = 1, \ldots, d$$

(9)

where $W^t$ is the ith row of $W$. After a simple mathematical transformation, the final solution is:

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

(10)

- **Update $\nu$ by fixing $W$**
  While fixing $W$, the objective function Eq. (5) can be written as follows:

$$\min_{W, \nu} \sum_{i=1}^{n} v_{i} \|x_{i} - x_{i}W\|_{2}^2 + \sum_{i=1}^{n} \frac{\beta^2}{v_{i} + \beta k},$$

s.t. $0 \leq v_{i} \leq 1, i = 1, \ldots, n$ (11)

By defining $L = \sum_{i=1}^{n} l_{i} = \sum_{i=1}^{n} \|x_{i} - x_{i}W\|_{2}^2$, we have:

$$\min_{W, \nu} \sum_{i=1}^{n} \nu_{i} l_{i} + \sum_{i=1}^{n} \frac{\beta^2}{v_{i} + \beta k},$$

s.t. $0 \leq v_{i} \leq 1, i = 1, \ldots, n$ (12)

According to Eq. (12), the closed form solution of $v_{i}$ is:

$$v_{i} = \begin{cases} 1 & \text{if } l_{i} \leq \frac{1}{\sqrt{k+1/\beta}} \\ 0 & \text{if } l_{i} \geq \frac{1}{\sqrt{k}} \\ \beta (\frac{k}{\beta} - k) & \text{otherwise} \end{cases}$$

(13)

3.5 Convergence analysis

We denote the ith iteration of $\nu$ and $W$ as $\nu^{(i)}$ and $W^{(i)}$, respectively. Based on Algorithm 1, Eq. (5) can be written as follows:

$$E(W^{(i)}, \nu^{(i)}) = \sum_{i=1}^{n} v_{i}^{(i)} \|x_{i} - x_{i}W^{(i)}\|_{2}^2$$

$$+ \alpha \|W^{(i)}\|_{2,1} + \sum_{i=1}^{n} \frac{\beta^2}{v_{i}^{(i)} + \beta k}$$

(14)

According to self-paced learning theory [23] and the fixed $W^{(i)}$, we have:

$$E(W^{(i)}, \nu^{(i+1)}) \leq E(W^{(i)}, \nu^{(i)})$$

(15)
With the fixed \( \mathbf{v}^{(t+1)} \), based on the IRLS framework, we have the following inequality:

\[
E(\mathbf{W}^{(t+1)}, \mathbf{v}^{(t+1)}) \leq E(\mathbf{W}^{(t)}, \mathbf{v}^{(t+1)})
\]  

(16)

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

\[
E(\mathbf{W}^{(t+1)}, \mathbf{v}^{(t+1)}) \leq E(\mathbf{W}^{(t)}, \mathbf{v}^{(t)})
\]  

(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 \( \beta \) 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:

\[
L_m = \frac{1}{\sqrt{\text{E}^{1/\beta}}}
\]  

(18)

To simplify the calculation, we let \( k = \frac{1}{p} \) and obtain:

\[
k = \frac{1}{2\pi}
\]  

(19)

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

\[
\beta = 21L_m^2
\]  

(20)

According to Eqs. (19) and (20), our proposed method can obtain the appropriate parameters \( k \) and \( \beta \) according to the number of samples initially selected, therefore, the dependence of the proposed algorithm on the parameters can be reduced. After the parameters \( k \) and \( \beta \) 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 \[22\]. 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 \( \ell_2, p \)-norm Augmented Lagrangian Multiplier (FSR_ALM [3])** 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 Dictionary Learning for Unsupervised Feature Selection (CDLFS [45])** 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 [4]) 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 [46]) 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 [27]) 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 \( \alpha \) of the proposed method in Eq. (5) as \([10^{-3}, 10^{-2}, \ldots, 10^2]\) and set the parameter \( \mu > 1 \). For other comparison methods, we 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.,

\[
ACC = \frac{N_c}{N}
\]

(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:

\[
NMI = 2 \sqrt{\frac{\sum_{X,Y} f(X,Y)}{\sum_{X} f(X) \sum_{Y} f(Y)}}
\]

(22)
Table 3
Clustering accuracy on benchmark data sets. Bold numbers indicate the best results.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Unist</th>
<th>USPS</th>
<th>Jaffe</th>
<th>Islet</th>
<th>DBworld</th>
<th>Ecoli</th>
<th>Coil</th>
<th>Colon</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.4400</td>
<td>0.5763</td>
<td>0.7230</td>
<td>0.4885</td>
<td>0.8125</td>
<td>0.5238</td>
<td>0.6097</td>
<td>0.3871</td>
<td>0.5701</td>
</tr>
<tr>
<td>FSR_ALM</td>
<td>0.4678</td>
<td>0.6594</td>
<td>0.7066</td>
<td>0.5679</td>
<td>0.8594</td>
<td>0.6042</td>
<td>0.6340</td>
<td>0.4556</td>
<td>0.6261</td>
</tr>
<tr>
<td>CDLFS</td>
<td>0.4609</td>
<td>0.6357</td>
<td>0.7700</td>
<td>0.5716</td>
<td>0.8378</td>
<td>0.6399</td>
<td>0.6235</td>
<td>0.4577</td>
<td>0.6246</td>
</tr>
<tr>
<td>CSFS</td>
<td>0.4539</td>
<td>0.6305</td>
<td>0.7887</td>
<td>0.5577</td>
<td>0.8559</td>
<td>0.5923</td>
<td>0.6257</td>
<td>0.4516</td>
<td>0.6185</td>
</tr>
<tr>
<td>RSR</td>
<td>0.4643</td>
<td>0.6180</td>
<td>0.7793</td>
<td>0.5538</td>
<td>0.8594</td>
<td>0.6339</td>
<td>0.6319</td>
<td>0.4566</td>
<td>0.6262</td>
</tr>
<tr>
<td>GSR</td>
<td>0.4730</td>
<td>0.6684</td>
<td>0.8122</td>
<td>0.5641</td>
<td>0.8750</td>
<td>0.6250</td>
<td>0.6340</td>
<td>0.4776</td>
<td>0.6412</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.5061</td>
<td>0.7207</td>
<td>0.8967</td>
<td>0.6147</td>
<td>0.9063</td>
<td>0.7074</td>
<td>0.6681</td>
<td>0.5484</td>
<td>0.6961</td>
</tr>
</tbody>
</table>

Table 4
The results of NMI, Purity and ARI on benchmark data sets. Bold numbers indicate the best results.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Unist</th>
<th>USPS</th>
<th>Jaffe</th>
<th>Islet</th>
<th>DBworld</th>
<th>Ecoli</th>
<th>Coil</th>
<th>Colon</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.6492</td>
<td>0.5061</td>
<td>0.3541</td>
<td>0.5721</td>
<td>0.6651</td>
<td>0.4685</td>
<td>0.7966</td>
<td>0.7465</td>
<td>0.6422</td>
</tr>
<tr>
<td>FSR_ALM</td>
<td>0.6364</td>
<td>0.5009</td>
<td>0.3368</td>
<td>0.6107</td>
<td>0.7261</td>
<td>0.5279</td>
<td>0.7929</td>
<td>0.7906</td>
<td>0.6668</td>
</tr>
<tr>
<td>CDLFS</td>
<td>0.6361</td>
<td>0.5217</td>
<td>0.3365</td>
<td>0.6082</td>
<td>0.7053</td>
<td>0.5276</td>
<td>0.7828</td>
<td>0.7840</td>
<td>0.6209</td>
</tr>
<tr>
<td>CSFS</td>
<td>0.6468</td>
<td>0.4922</td>
<td>0.3506</td>
<td>0.6965</td>
<td>0.7051</td>
<td>0.5250</td>
<td>0.8638</td>
<td>0.8216</td>
<td>0.7331</td>
</tr>
<tr>
<td>RSR</td>
<td>0.6343</td>
<td>0.4904</td>
<td>0.3550</td>
<td>0.6073</td>
<td>0.7163</td>
<td>0.5193</td>
<td>0.8050</td>
<td>0.7793</td>
<td>0.6872</td>
</tr>
<tr>
<td>GSR</td>
<td>0.6417</td>
<td>0.5026</td>
<td>0.3496</td>
<td>0.6132</td>
<td>0.7256</td>
<td>0.5235</td>
<td>0.8529</td>
<td>0.8498</td>
<td>0.7383</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.6912</td>
<td>0.5565</td>
<td>0.4188</td>
<td>0.6111</td>
<td>0.7207</td>
<td>0.5463</td>
<td>0.9019</td>
<td>0.8967</td>
<td>0.8165</td>
</tr>
</tbody>
</table>

where \( l(X, Y) \) denotes mutual information (MI) between the samples and the labels, \( H(.) \) is the entropy.

- **Purity**: Purity reflects the ratio of correctly classified samples in each cluster, which definition is:
  \[
  \text{Purity} = \frac{K}{\sum_{i=1}^{K} m_i P_i} 
  \]
  where \( K \) is the number of clusters, \( m_i \) and \( P_i \) are the 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.,
  \[
  ARI = \frac{R(I) - E(R(I))}{\max(R(I)) - E(R(I))} 
  \]
  where \( R(I) \) is a rand index and \( E(R(I)) \) 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 and 4. We also discussed the in-

Please cite this article as: W. Zheng et al., Unsupervised feature selection by self-paced learning regularization, Pattern Recognition Letters (2018), https://doi.org/10.1016/j.patrec.2018.06.029
fluence 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 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 $\beta$ were fixed, we still need to tune the parameters $\alpha$ and $\mu$. In this paper, we set the range of $\alpha$ and $\mu$...
μ as $10^{-3}, \ldots, 10^3$ and $[1.1, 1.15, \ldots, 1.35, 1.4]$, respectively, and listed the corresponding results in Fig. 2. As shown 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. 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 Algorithms 1 and 2 as $10^{-5}$, i.e., $\frac{\|f(t+1) - f(t)\|}{\|f(t)\|} \leq 10^{-5}$, where $f(t)$ 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 monotonically 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 $l_2,1$-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 since graph learning can further enhance the effect of feature selection model [17,44].

Acknowledgments

This work is partially supported by the China Key Research Program (Grant No. 2016YFB1000905); the National Natural Science Foundation of China (Grants No. 616573270 and 61672177); the Project of Guangxi Science and Technology (GuixiAEd17195062); the Guangxi Natural Science Foundation (Grant No. 2015GXNSFCB189011); 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


