IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 25, 2019, accepted July 2, 2019, date of publication July 23, 2019, date of current version August 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930470

Local Graph Reconstruction for Parameter Free Unsupervised Feature Selection

LIANG DU[1,2,3], (Member, IEEE), CHAOHONG REN[1], XIAOLIN LV[1], YAN CHEN[1], PENG ZHOU[4], AND ZHIGUO HU[1,2,3]

[1] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China
[2] Institute of Big Data Science and Industry, Shanxi University, Taiyuan 030006, China
[3] Key Laboratory of Computational Intelligence and Chinese Information Processing, Ministry of Education, Shanxi University, Taiyuan 030006, China
[4] School of Computer Science and Technology, Anhui University, Hefei 230601, China

Corresponding authors: Liang Du (csliangdu@gmail.com) and Zhiguo Hu (646579354@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502289, Grant 61603230, Grant 61806003, Grant 61802238, and Grant 61872226, and in part by the Natural Science Foundation of Shanxi, China, under Grant 201701D121052 and Grant 201801D221163.

ABSTRACT Facing with the absence of supervised information to guide the search of relevant features and the grid-search of model/hyper-parameters, it is more preferred to develop parameter-free methods and avoid additional hyper-parameters tuning. In this paper, we propose a new simple and effective parameter-free unsupervised feature selection algorithm by minimizing the linear reconstruction weight between the nearest neighbor graphs constructed from all candidate features and each single feature. The obtained global optimal reconstruction weights actually select those features with highest relevance and lowest redundancy simultaneously. The experimental results on many benchmark data sets demonstrate that the proposed method outperforms many of the state-of-the-art unsupervised feature selection methods.

INDEX TERMS Local graph reconstruction, parameter free, global optimal, redundancy minimization.

I. INTRODUCTION

Real world applications usually involve big data with high dimensionality, presenting great challenges such as the curse of dimensionality, huge computation and storage cost. To tackle these difficulties, feature selection techniques are developed to keep a few relevant and informative features from the original high-dimensional features for the subsequent tasks. Based on a small number of representative features, not only the learning process of the model could be accelerated, but also the generalization ability could be improved.

According to the availability of supervised information, feature selection methods can be categorized into supervised [1]–[6] semi-supervised [7]–[9] and unsupervised algorithms [10], [11]. Compared to supervised or semi-supervised counterparts, unsupervised feature selection is generally more challenging due to the lack of supervised information to guide the search of relevant features.

Generally speaking, unsupervised feature selection methods can be further categorized into two types: filter and embedded [12]–[29]. The filter-based feature selection algorithms rank the features in terms of a predefined criterion, which is completely independent on the learning methods. The embedded-based methods consider the feature evaluation criterion by incorporating into the learning procedure. Since embedded-based methods take the learning model into consideration, they usually perform better than filter-based ones. However, these methods often involve more parameters for tuning, it is hard to effectively perform unsupervised model selection due to the absence of supervised information. Besides, the embedded algorithms are also computationally expensive thereby impeding their uses in the tasks where the dimensionality and the amount of the data are large.

In this paper, we are particularly interested in the parameter-free (or at least the hyper-parameters can be easily set to a certain constant value) feature selection methods, in which data variance, Laplacian score [30], sparsity score [31], Multi Cluster Feature Selection [11], the Local and Global Discriminative algorithm [32], and LLE score [33] are representatives. However, the quadratic score

The associate editor coordinating the review of this manuscript and approving it for publication was Shangce Gao.

function in [30], [32] have several drawbacks as pointed out [33]. The LLE score [33] also requires fine tuning of the regularization parameter as suggested by the authors. Moreover, these aforementioned algorithms usually neglect the correlations of candidate features by evaluating the importance of features one by one and lead to sub-optimal result.

Facing with such difficulties, it is more preferred to develop parameter-free methods or methods without additional hyper-parameter tuning. In this paper, we propose a new simple and effective parameter free unsupervised feature selection algorithm, where we adopt the simple nearest neighbor graph to characterize the local structure of data. The importance of each feature is evaluated by minimizing the reconstruction weight between the weight matrices of all features and single feature. It can be further verified that our method actually selects highly relevant and lower redundant features [32], [34]. Experimental results on many benchmark data sets demonstrate that the proposed method outperforms many state-of-the-art parameter-free unsupervised methods.

The rest of the paper is organized as follows. In Section 2, we review the most closely related filter-based unsupervised algorithms. The proposed algorithm is derived in Section 3. In Section 4, we evaluate the proposed method on many data sets. We make concluding remarks in Section 5.

II. RELATED WORK

In this section, we will first review some related unsupervised feature selection methods.

A. LAPLACIAN SCORE

Laplacian score is a filter-based unsupervised feature selection method. The main idea of Laplacian score is that the data points which locate nearby are probably related to the same class. Therefore, the local structure of the data is more important than the global structure. In this way, Laplacian score evaluates the feature by its ability of preserving the local structure of data.

It computes the neighborhood relationship according to

$$w_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{t^2}} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Then, the score of each feature is computed as follows

$$\text{LapScore}_r = \frac{\sum_{i=1}^n \sum_{j=1}^n (f_{ri} - f_{rj})^2 w_{ij}}{\sum_{i=1}^n (f_{ri} - \mu_r)^2 d_{ii}} \tag{2}$$

where $d_{ii} = \sum_{j=1}^n w_{ij}$ and μ_r is the average of r-th feature.

B. LGD SCORE

The LGD score [32] takes an unsupervised Local and Global Discriminative (LGD) feature selection criterion. The score of each feature is defined as the ratio between global variance and local variance:

$$\text{LGD}_r = \frac{\sum_{i=1}^n (f_{ri} - \mu_r)^2}{\sum_j \sum_{f_{ri}\in o(f_{rj})} (f_{ri} - \bar{f}_{rj})^2} \tag{3}$$

where f_{ri} is the r-th feature of the i-th sample, μ_r is the mean of the r-th feature, $o(f_{rj})$ is the set of neighbor points of the j-th sample. $\bar{f}_{rj}$ is the mean computed from the set of neighbor points.

The LGD score prefers to select features with large global variance and small local variance. Such features are expected to be discriminative for classification/clustering tasks.

C. LLE SCORE

LLE score [33] takes the Local Linear Embedding method [35] to characterize the local structure of data. It evaluates the importance of each feature according to the difference of reconstruction weights computed from the single feature and all the features.

The optimal reconstruction weight $\mathbf{M} \in \mathcal{R}^{n\times n}$ with all the candidate features is computed by solving

$$\min_{\mathbf{M}} \quad \|\mathbf{x}_i - \sum_{j\in\mathcal{N}_i} m_{ij}\mathbf{x}_j\|^2, \quad \text{s.t.} \sum_{j\in\mathcal{N}_i} m_{ij} = 1, \tag{4}$$

where $\mathcal{N}_i$ is the neighbor of i-sample. The reconstruction weight captured by the r-th feature is also computed by

$$\begin{aligned} &\min_{\mathbf{M}^r} \|f_{ri} - \sum_{j\in\mathcal{N}_i^r} m_{ij}^r f_{rj}\|^2 + \gamma \sum_{j\in\mathcal{N}_i^r} (m_{ij}^r)^2, \\ &\text{s.t.} \sum_{j\in\mathcal{N}_i^r} m_{ij}^r = 1, \end{aligned} \tag{5}$$

where $\mathcal{N}_i^r$ is the neighbor of i-sample from r-th feature.

Given the reconstruction weight of all features $\mathbf{M}$ and the reconstruction weight of r-th feature $\mathbf{M}^r$, the LLE score is computed by

$$\text{LLEScore}_r = \|\mathbf{M} - \mathbf{M}^r\|^2. \tag{6}$$

For each feature, the above criterion of LLE score evaluates the ability to preserve the local linear structure.

III. THE PROPOSED METHOD

The generic problem of unsupervised feature selection is to find the most informative features. Given a set of points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{R}^{d\times n}$, finding a feature subset with size m which contains the most informative features. In other words, the points$\{\mathbf{x}_1', \mathbf{x}_2', \ldots, \mathbf{x}_n'\}$ represented in the m-dimensional space $\mathcal{R}^m$ can well preserve the intrinsic structure as the data represented in the original d-dimensional space.

It has been well recognized that the local structure of the data space is more important than the global structure for the task of clustering and classification [28]. Several approaches have been developed for local structure characterization. In order to model the local geometric structure, we construct a simple and effective nearest neighbor graph $\mathbf{A}$, which is generated from all the candidate features without additional parameter except for the neighborhood size. The weight matrix can be obtained according to

$$\mathbf{A}_{ij} = \begin{cases} \frac{1}{n_i} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors,} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

where n_i is the neighbor size of i-th sample.

Given the local structure captured by the affinity graph, most existing filter-based algorithms take the quadratic function to evaluate the importance of each feature such as in Eq. (2) and Eq. (3). However, it has been pointed out that such score function suffers from at least three drawbacks [33]: 1) it fails when the elements of all the samples are equal; 2) it lacks the scaling invariant property; 3) it cannot well capture the change of the graph for each element. These weaknesses will greatly degrade its performance in feature selection.

Instead of using the quadratic function, we adopt similar idea as LLE score by constructing the local structure, i.e. $\mathbf{A}^r$, for each single feature. The weight matrix for r-th feature is computed by

$$\mathbf{A}^r_{ij} = \begin{cases} \frac{1}{n^r_i} & \text{if } f^r_i \text{ and } f^r_j \text{ are neighbors,} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

where n^r_i is the neighbor size of i-th sample with r-th feature.

Here, since both the weight matrices $\mathbf{A}$ and $\{\mathbf{A}^r\}^m_{r=1}$ capture the local structure of data, the importance of each feature then can be evaluated by their reconstruction weights to the consensus weight matrix $\mathbf{A}$. The corresponding problem can be formulated as

$$\begin{aligned} \min_{\mathbf{w}} \ & \|\mathbf{A} - \sum_{r=1}^{d} w_i \mathbf{A}^i||^2 \\ \text{s.t.} \ & \sum_{i=1}^{d} w_i = 1, \quad \mathbf{w} \geq 0. \end{aligned} \tag{9}$$

It is obvious that the above formulation does not introduce additional hyper-parameter for certain regularization, such as sparse regularization, which is often adopted by other embedded unsupervised feature selection algorithms. Such merit of parameter free is especially helpful for the task of unsupervised feature selection without supervision.

We first introduce $\mathbf{H} \in \mathcal{R}^{d\times d}$ by denoting

$$\mathbf{H}_{ij} = \text{tr}((\mathbf{A}^i)^T \mathbf{A}^j), \tag{10}$$

where $\mathbf{H}_{ij}$ is used to characterize the similarity between feature i and j, and introduce $\mathbf{b} \in \mathcal{R}^{d\times 1}$ with

$$b_i = \text{tr}(\mathbf{A}^T \mathbf{A}^i) \tag{11}$$

where b_i represents the similarity between feature i and all the candidate features. Then, the problem in Eq. (9) can be reformulated as

$$\begin{aligned} \min_{\mathbf{w}} \ & \mathbf{w}^T \mathbf{H} \mathbf{w} - 2\mathbf{w}^T \mathbf{b} \\ \text{s.t.} \ & \sum_{i=1}^{d} w_i = 1, \quad \mathbf{w} \geq 0. \end{aligned} \tag{12}$$

Clearly, the above problem is a convex quadratic programming with linear constraints, which can be easily solved by off-the-shelf optimization toolbox.

Now, we can fully investigate why the proposed algorithm is suitable for the task of unsupervised feature selection. The first term in the objective function $\mathbf{w}^T\mathbf{H}\mathbf{w}$ can be rewritten as $\sum_{i,j=1}^d \mathbf{H}_{ij} w_i w_j$. When $\mathbf{H}_{ij}$ is large, it indicates that the feature i and j have the similar neighborhood structure. That is to say, they are redundant features. By minimizing $\sum_{i,j=1}^d \mathbf{H}_{ij} w_i w_j$, the value of w_i and w_j can not be large simultaneously. In this case, we will get one large value and one small value for these two highly redundant features. Consequently, the ranking of one feature is kept and the ranking of the other feature decreases, when these two features are highly similar with each other. The second term in the objective function is equivalent to maximizing $\sum_{i=1}^d w_i b_i$. It actually selects those features which have similar neighbor structure with the structure constructed from all the candidate features, i.e., $\mathbf{A}$. Given the optimal reconstruction weight of Eq. (9) or Eq (12), we can use it to evaluate the ability of each feature to preserve the local structure. It can be seen that the features with higher scores are better for the preserving of the local structure. We list the details of the proposed method in Algorithm 1.

Algorithm 1 The Proposed Filter-Based Unsupervised Feature Selection Algorithm

```
Require: Data matrix $X \in \mathbb{R}^{d\times n}$, the number of selected features $m$
    1) Construct the $k$-nearest neighbor matrix $\mathbf{A}$ using all the candidate features according to Eq. (7).
    2) Construct the $k$-nearest neighbor matrix $\mathbf{A}^r$ using the $r$-th feature according to Eq. (8).
    3) Compute the score for each feature by solving Eq. (9) or Eq. (12).
    4) Ranking features $\mathbf{w}$ in descending order, select the top $m$ features.
Ensure: top $m$ features
```

Although the above algorithm is simple and easy to understand, it still brings several nice properties for the task of unsupervised feature selection. It is worthwhile to highlight several aspects of the proposed approach here:

- It is parameter free. It can be seen that only one parameter, i.e., the neighborhood size, is involved. In practice, we set the neighborhood size to be 5 in our experiment. It is vital important to point out that the parameter or model selection is often unrealistic for the task of unsupervised feature selection due to the absence of supervised information. That is also one of the main advantages of our method compared to other filter-based methods and embedded approaches.
  Comparatively speaking, Laplacian score, LGD [32], MCFS [11] and LLE score [33] both need the neighborhood size. Besides, they often require additional hyper-parameters, such as the kernel width for Gaussian kernel function in Laplacian score, the regularization parameter in MCFS and LLE score.
- It evaluates the importance of each feature by considering both the relevance and redundancy simultaneously. As a result, the redundancy of the select feature subset

is large alleviated and features with more relevant information can be further selected.

- The global optima of the optimization problem in Eq. (9) can be easily obtained. Thus, we can avoid the greedy search or the local optimal which are often encountered by other counterparts.
- It is scale invariant. For example, let $\mathbf{f}_1 = 2\mathbf{f}_2$, $\mathbf{f}_1$ and $\mathbf{f}_2$ have the same graph structure and the score for $\mathbf{f}_1$ and $\mathbf{f}_2$ are equal. It has been pointed out that Laplacian score has different results for such features [30]. The LGD algorithm also has such problem [32].
- It is distinguished for features with equal values. The local structure of the feature with equal values does not contain any meaningful structure, and will get smaller reconstruction weight. Both the Laplacian score and the LGD algorithm will give the best score for such features.
- It is less sensitive to the change of local weight, which would be largely influenced by heavily data corruption. The binary graph weighting used in Eq. (7) only captures the neighborhood relationship while ignoring the relative sensitive closeness weights. It is believed that such weighting schema is also robust to data corruption in certain extent. It captures the change of local structure efficiently. Once the local neighborhood is changed for certain feature, it will be penalized by Eq. (9).

Now, we analyze the time complexity of the proposed method. The cost of computing the Euclidean distances between the i-th sample and the other samples is $\mathbf{O}(nd)$, then finding its k-nearest neighbors costs $\mathbf{O}(nk)$. Thus, the total computational complexity of computing $\mathbf{A}$ is $\mathbf{O}(n^2d + n^2k)$. The computational complexity for $\mathbf{A}^r$ is $\mathbf{O}(n \log n)$. The selection of top features is $\mathbf{O}(dm)$, where m is the number of selected features. In most cases, $d > k$, in this way, the computational complexity can be written into $\mathbf{O}(n^2d + dn \log n)$. The quadratic optimization problem in (12) can be solved in $\mathbf{O}(d^3)$. The time complexity of the proposed method is also comparable with other methods.

IV. EXPERIMENT

In this section, extensive experiments are conducted on 6 real-world datasets to validate the effectiveness of the proposed method. Four state-of-art relevant unsupervised feature selection methods are adopted as competitors.

A. DATA SETS

We collect a variety of data sets, including 5 image data sets and 1 text corpora and 1 biological data, most of which have been frequently used to evaluate the performance of different feature selection algorithms. The statistics of these data sets, including the number of data samples, the dimension of each sample, the types and categories of each dataset and the number of selected features are summarized in Table 1. We further present the details of these 6 data set as follows

TABLE 1. The details of data sets in our experiments.

Dataset	# instances	# features	# selected features	# classes
COIL	1440	1024	[5:5:50]	20
JAFFE	213	676	[5:5:50]	10
UMIST	575	644	[5:5:50]	20
YALEB	2414	1024	[5:5:50]	38
PIE	2856	1024	[5:5:50]	68
LUNG	203	3312	[5:5:50]	5

- **COIL20** dataset from Columbia University Image Library contains 20 classes, and each class has 72 images. Each image is of 32 × 32 pixels with some rotation.
- **JAFFE**. This database contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models.
- **UMIST**. Face Database consists of 575 images of 20 people. Each covering a range of poses from profile to frontal views. Subjects cover a range of race/sex/appearance.
- **YALEB**. The Extended Yale-B database contains 16128 face images of 38 human subjects under 9 pose and 64 illumination conditions. In our experiment, we choose the frontal pose and use all the images under different illumination, thus we get 2414 images in total. They are resized to 32 × 32 pixels, with 256 gray levels per pixel.
- **PIE** is a gray (32 × 32 pixels) scale face images. In addition, the dataset has 68 persons and each person images have different illuminations and poses.
- **LUNG** [6] data set contains in total 203 samples in five classes, which have 139, 21, 20, 6,17 samples, respectively. Each sample has 12600 genes. The genes with standard deviations smaller than 50 expression units were removed and we obtained a data set with 203 samples and 3312 genes.

B. COMPARED ALGORITHMS

To validate the effectiveness of our proposed method,[1] we compare it with one baseline (i.e., AllFea) and states-of-the-art almost parameter free unsupervised feature selection methods,

- Max Variance which selects those features of maximum variances in order to obtain the best expressive power.
- Laplacian Score [30] which selects the features most consistent with the Gaussian Laplacian matrix.
- MCFS[2] performs eigen-decomposition regarding the Laplacian matrix, and evaluates the importance of

[1] For the purpose of reproducibility, we provide all the dataset and the code at https://gitee.com/csliangdu/LGRUFS

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/code/MCFS_p.m

TABLE 2. Clustering accuracy on different datasets with $k = 5$ (mean ± std).

Data Sets	AllFea	MaxVar	LapScore	MCFS	LGD	LLEScore	Our Method
COIL	0.5917	0.4330 ± 0.0479	0.4559 ± 0.0612	0.5417 ± 0.0654	0.5060 ± 0.0286	0.5677 ± 0.0405	**0.5806 ± 0.0241**
JAFFE	0.7157	0.4816 ± 0.0620	0.6769 ± 0.0853	0.6599 ± 0.0341	0.5897 ± 0.0334	0.6458 ± 0.0401	**0.7135 ± 0.0510**
UMIST	0.4240	0.4191 ± 0.0379	0.3677 ± 0.0111	0.3994 ± 0.0233	0.4133 ± 0.0371	0.4132 ± 0.0146	**0.4922 ± 0.0073**
YALEB	0.0962	0.0907 ± 0.0013	0.0866 ± 0.0008	0.1121 ± 0.0103	0.0954 ± 0.0021	0.1266 ± 0.0046	**0.1550 ± 0.0148**
PIE	0.1809	0.1678 ± 0.0190	0.1752 ± 0.0157	0.1669 ± 0.0094	0.1547 ± 0.0472	0.1371 ± 0.0047	**0.2914 ± 0.0332**
LUNG	0.7246	0.5271 ± 0.0638	0.5329 ± 0.0349	0.4381 ± 0.0552	0.4424 ± 0.0545	0.5107 ± 0.0146	**0.6660 ± 0.0538**
Average	0.4555	0.3532	0.3825	0.3863	0.3669	0.3997	**0.4831**

TABLE 3. Clustering NMI on different datasets with $k = 5$ (mean ± std).

Data Sets	AllFea	MaxVar	LapScore	MCFS	LGD	LLEScore	Our Method
COIL	0.7376	0.5627 ± 0.0582	0.5816 ± 0.0595	0.6461 ± 0.0780	0.6381 ± 0.0367	0.6686 ± 0.0462	**0.6728 ± 0.0414**
JAFFE	0.7921	0.5099 ± 0.0971	0.7474 ± 0.0890	0.7096 ± 0.0333	0.6317 ± 0.0194	0.6987 ± 0.0476	**0.7841 ± 0.0466**
UMIST	0.6336	0.5652 ± 0.0488	0.5466 ± 0.0237	0.5322 ± 0.0380	0.5658 ± 0.0496	0.5872 ± 0.0244	**0.6503 ± 0.0139**
YALEB	0.1290	0.1333 ± 0.0053	0.1358 ± 0.0012	0.1875 ± 0.0232	0.1396 ± 0.0067	0.2017 ± 0.0111	**0.2552 ± 0.0199**
PIE	0.4081	0.3985 ± 0.0330	0.4112 ± 0.0226	0.3924 ± 0.0154	0.3798 ± 0.1176	0.3480 ± 0.0082	**0.5545 ± 0.0304**
LUNG	0.5220	0.3645 ± 0.0769	0.3912 ± 0.0389	0.2469 ± 0.0874	0.2561 ± 0.0720	0.3275 ± 0.0220	**0.4771 ± 0.0556**
Average	0.5371	0.4223	0.4690	0.4525	0.4352	0.4720	**0.5657**

TABLE 4. Clustering purity on different datasets with $k = 5$ (mean ± std).

Data Sets	AllFea	MaxVar	LapScore	MCFS	LGD	LLEScore	Our Method
COIL	0.6432	0.4890 ± 0.0529	0.5113 ± 0.0640	0.5833 ± 0.0709	0.5534 ± 0.0291	0.6030 ± 0.0397	**0.6140 ± 0.0283**
JAFFE	0.7526	0.5101 ± 0.0657	0.7137 ± 0.0876	0.6977 ± 0.0359	0.6237 ± 0.0298	0.6808 ± 0.0402	**0.7510 ± 0.0506**
UMIST	0.4978	0.4720 ± 0.0469	0.4287 ± 0.0187	0.4503 ± 0.0306	0.4635 ± 0.0460	0.4857 ± 0.0251	**0.5561 ± 0.0123**
YALEB	0.1046	0.1022 ± 0.0024	0.0962 ± 0.0014	0.1242 ± 0.0142	0.1071 ± 0.0040	0.1341 ± 0.0052	**0.1803 ± 0.0160**
PIE	0.2027	0.1916 ± 0.0189	0.1956 ± 0.0167	0.1946 ± 0.0090	0.1755 ± 0.0465	0.1656 ± 0.0047	**0.3139 ± 0.0324**
LUNG	0.8532	0.7957 ± 0.0414	0.8090 ± 0.0273	0.7455 ± 0.0382	0.7420 ± 0.0328	0.7825 ± 0.0032	**0.8440 ± 0.0372**
Average	0.5090	0.4268	0.4591	0.4654	0.4442	0.4753	**0.5432**

features via sparse spectral regression [11]. The neighbor size is set to 5.

- The local and global discriminative (LGD) feature selection criterion [32]. The score of feature is determined by the ratio between global variance and local variance.
- LLEScore [33]. It is a filter-based unsupervised feature selection method, which is based on LLE and the graph-preserving feature selection framework. The difference between structures of the graphs constructed by each feature and the original data was used to measure the importance of each feature.

C. EVALUATION METRICS

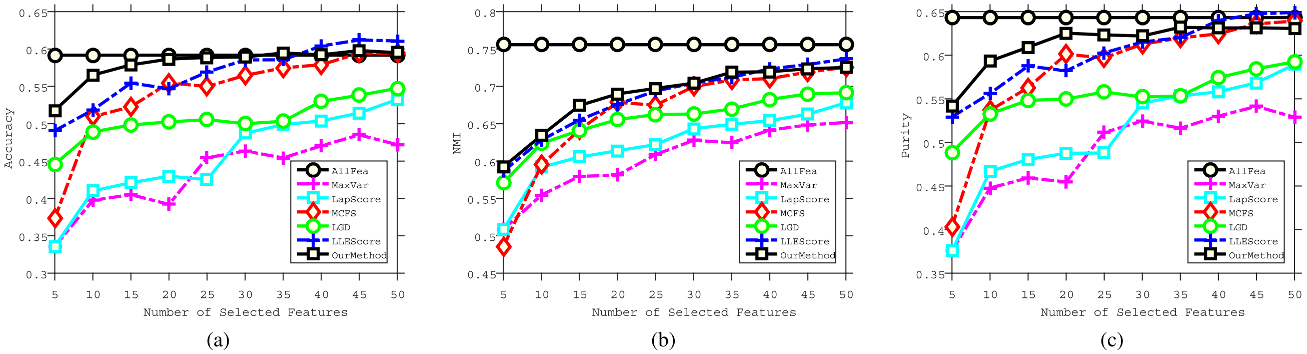
To evaluate their performance, we compare the generated clusters with the ground truth by computing the following three performance measures.

Clustering accuracy (ACC). The first performance measure is the clustering accuracy, which discovers the one-to-one relationship between clusters and classes. Given a point $\boldsymbol{x}_i$, let p_i and q_i be the clustering result and the ground truth label, respectively. The ACC is defined as follows:

$$\mathrm{ACC} = \frac{1}{n}\sum_{i=1}^{n}\delta(q_i, map(p_i)), \tag{13}$$

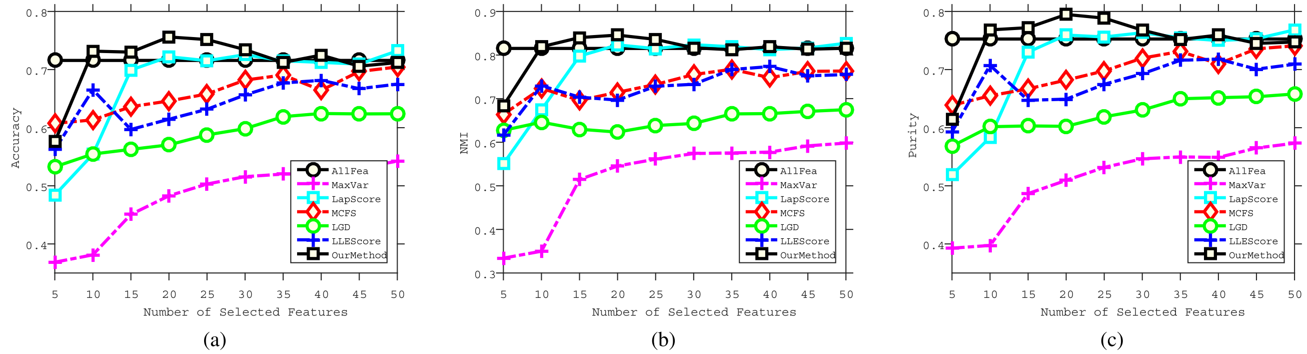
where n is the total number of samples and $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise, and $map(\cdot)$ is the permutation mapping function that maps each cluster index to a true class label. The best mapping can be found by using the Kuhn-Munkres algorithm [36]. The greater clustering accuracy means the better clustering performance.

Normalized mutual information (NMI). Another evaluation metric that we adopt here is the normalized mutual information, which is widely used for determining the quality of clustering. Let $\mathcal{C}$ be the set of clusters from the ground truth and $\mathcal{C}'$ obtained from a clustering algorithm. Their mutual



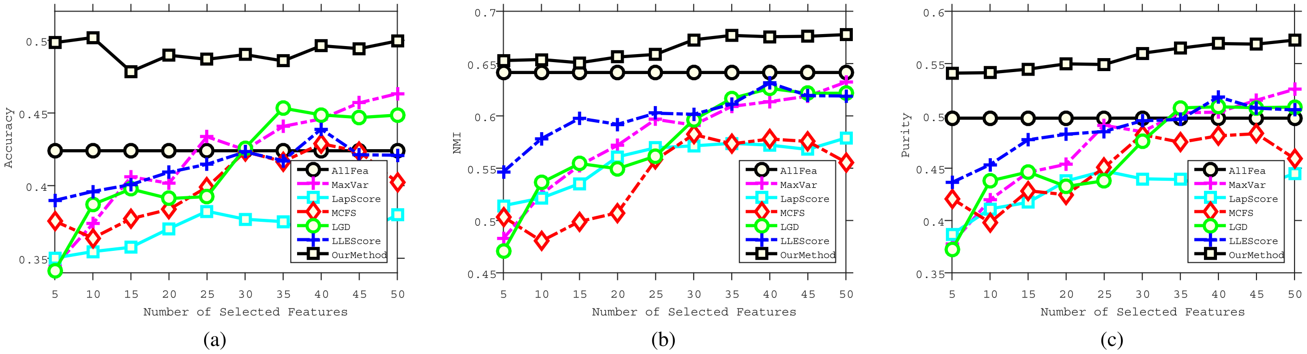
(a) (b) (c)

FIGURE 1. Clustering results w.r.t. different number of selected features on COIL.



(a) (b) (c)

FIGURE 2. Clustering results w.r.t. different number of selected features on JAFFE.



(a) (b) (c)

FIGURE 3. Clustering results w.r.t. different number of selected features on UMIST.

information $MI(\mathcal{C}, \mathcal{C}')$ is defined as follows:

$$\mathrm{MI}(\mathcal{C}, \mathcal{C}') = \sum_{c_i \in \mathcal{C}, c_j' \in \mathcal{C}'} p(c_i, c_j') \log \frac{p(c_i, c_j')}{p(c_i)p(c_j')}, \tag{14}$$

where $p(c_i)$ and $p(c_j')$ are the probabilities that a data point arbitrarily selected from the data set belongs to the cluster c_i and c_j', respectively, and $p(c_i, c_j')$ is the joint probability that the arbitrarily selected data point belongs to the cluster c_i as well as c_j' at the same time. In our experiments, we use the normalized mutual information as follows:

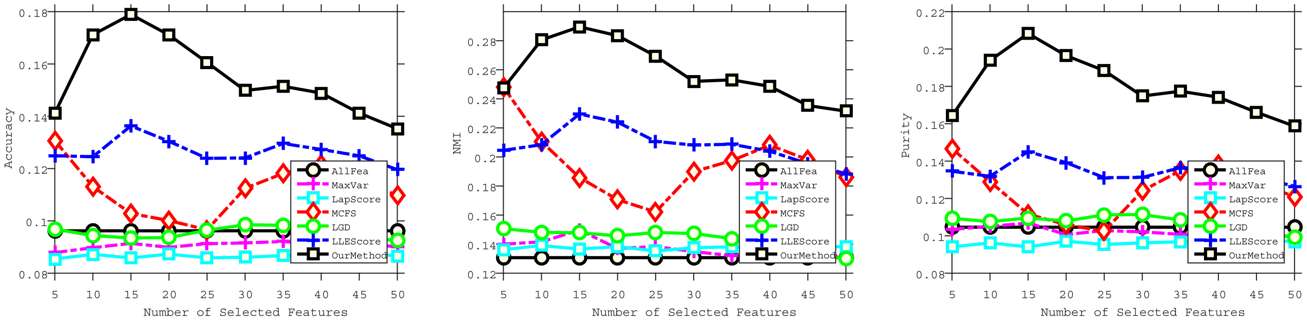
$$\mathrm{NMI}(\mathcal{C}, \mathcal{C}') = \frac{\mathrm{MI}(\mathcal{C}, \mathcal{C}')}{\max(H(\mathcal{C}), H(\mathcal{C}'))}, \tag{15}$$

where $H(\mathcal{C})$ and $H(\mathcal{C}')$ are the entropies of $\mathcal{C}$ and $\mathcal{C}'$, respectively. Again, a larger NMI indicates a better performance.

Purity measures the extent to which each cluster contained data points from primarily one class. The purity of a clustering solution is obtained as a weighted sum of individual cluster purity [37] values and is given by

$$\mathrm{Purity} = \sum_{i=1}^{c} \frac{n_i}{n} P(S_i), \quad P(S_i) = \frac{1}{n_i} \max_j(n_i^j) \tag{16}$$

where S_i is a particular cluster of size n_i, n_i^j is the number of samples of the i-th input class that were assigned to the j-th cluster, c is the number of clusters and n is the total number

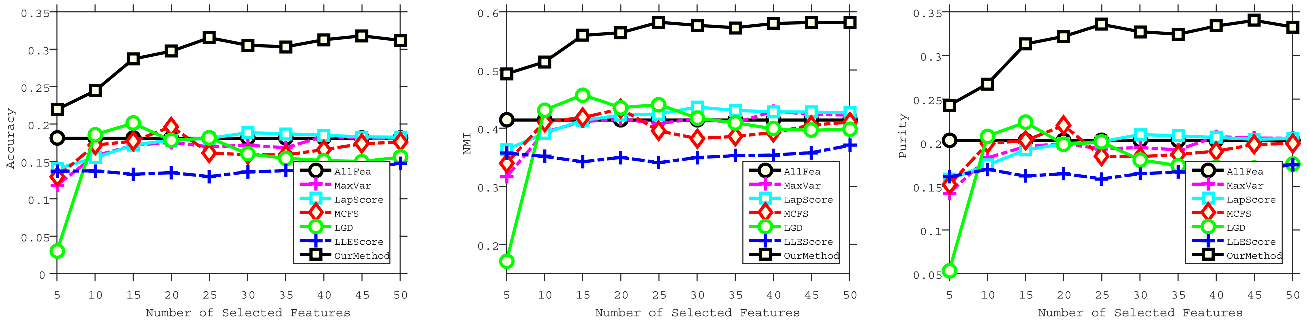


(a)

(b)

(c)

FIGURE 4. Clustering results w.r.t. different number of selected features on YALEB.

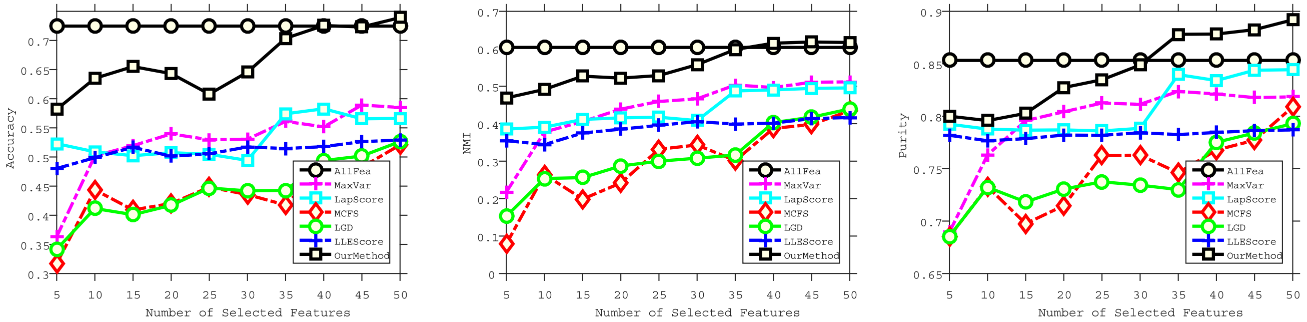


(a)

(b)

(c)

FIGURE 5. Clustering results w.r.t. different number of selected features on PIE.



(a)

(b)

(c)

FIGURE 6. Clustering results w.r.t. different number of selected features on LUNG.

of points. In general, the larger the values of purity, the better the clustering solution is.

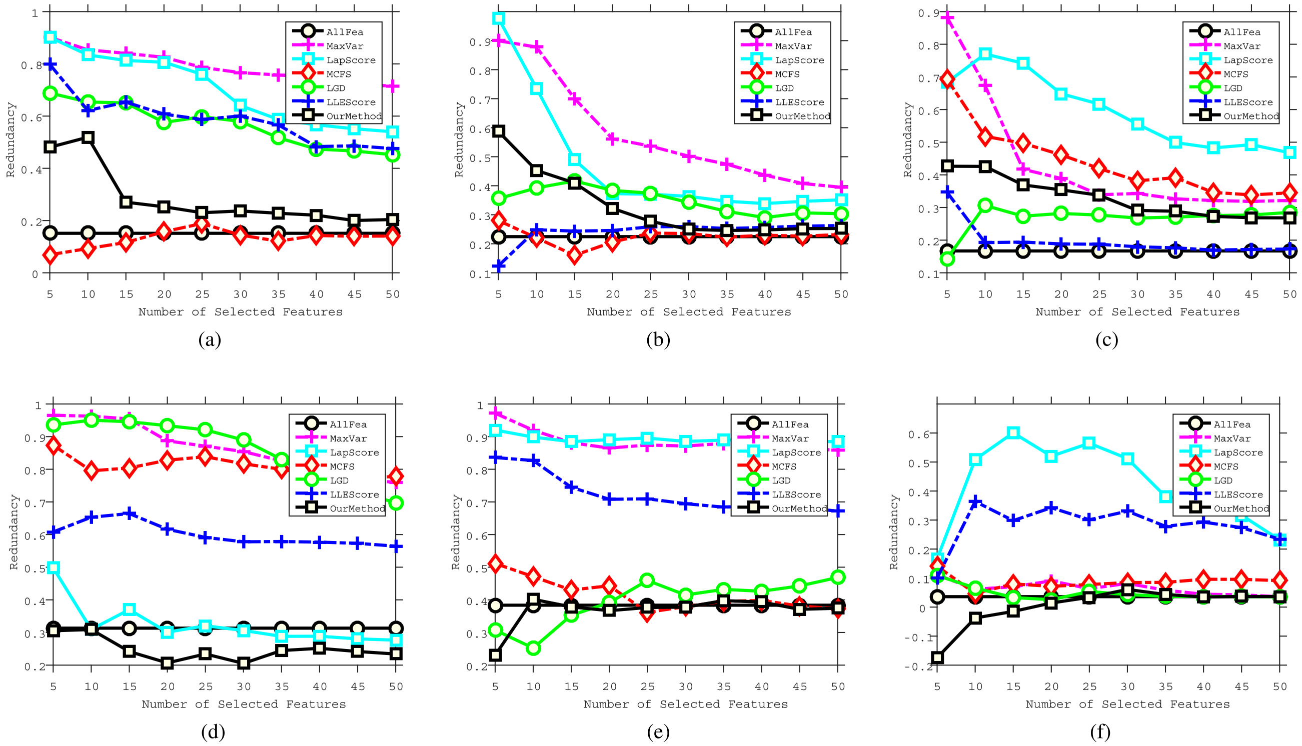
In order to measure the **redundancy** [38] among the selected features, the following formula is adopted as

$$\mathrm{Red}(\mathbf{S}) = \frac{1}{m(m-1)} \sum_{\mathbf{f}_i, \mathbf{f}_j \in \mathbf{S}, i \neq j} \rho_{ij} \tag{17}$$

where ρ_{ij} is the Pearson correlation between two features $\mathbf{f}_i$ and $\mathbf{f}_j$. The measurement assesses the averaged correlation among all feature pairs, and a large value indicates that many selected features are strongly redundant. In general, the smaller the values of redundancy or correlation, the better the selected features.

D. PARAMETERS SETTINGS

There are some parameters to be set in advance. We set the size of neighborhoods $k = 5$ on all the datasets for all these compared feature selection algorithms except MAXVAR. The weight of k-nn graph for LapScore is based on the Gaussian kernel, where the kernel width is set to the mean distance between any two data examples as suggested in [39]. For LLEScore, the regularization parameter is set to 10^{-5} as



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 7. Redundancy of the selected features with $k = 5$ on COIL, JAFFE, UMIST, YALEB, PIE and LUNG, respectively.

TABLE 5. Redundancy on different datasets with $k = 5$ (mean ± std).

Data Sets	AllFea	MaxVar	LapScore	MCFS	LGD	LLEScore	Our Method
COIL	0.1512	0.7916 ± 0.0615	0.7002 ± 0.1366	**0.1313 ± 0.0335**	0.5655 ± 0.0849	0.5880 ± 0.0968	0.2842 ± 0.1159
JAFFE	0.2245	0.5791 ± 0.1855	0.4692 ± 0.2164	**0.2249 ± 0.0297**	0.3477 ± 0.0436	0.2412 ± 0.0417	0.3297 ± 0.1168
UMIST	0.1669	0.4332 ± 0.1910	0.5957 ± 0.1123	0.4392 ± 0.1097	0.2657 ± 0.0444	**0.1979 ± 0.0530**	0.3302 ± 0.0621
YALEB	0.3133	0.8639 ± 0.0786	0.3239 ± 0.0668	0.8086 ± 0.0314	0.8590 ± 0.0978	0.6004 ± 0.0347	**0.2474 ± 0.0350**
PIE	0.3836	0.8846 ± 0.0352	0.8923 ± 0.0109	0.4144 ± 0.0485	0.3948 ± 0.0702	0.7247 ± 0.0597	**0.3668 ± 0.0493**
LUNG	0.0359	0.0659 ± 0.0234	0.4168 ± 0.1471	0.0870 ± 0.0236	0.0476 ± 0.0244	0.2816 ± 0.0737	**0.0033 ± 0.0690**
Average	0.2126	0.6031	0.5664	0.3509	0.4134	0.4390	**0.2603**

in [33]. Compared to most embedded unsupervised feature selection methods which often have the difficulty of unrealistic grid-search of several parameters, it can be seen that the parameters for all these compared unsupervised feature selection can be relatively easy to set in advance.

E. CLUSTERING WITH SELECTED FEATURES

With the selected features, we evaluate the performance in terms of k-means clustering by three widely used metrics, i.e., Accuracy (ACC) and Normalized Mutual Information (NMI) and Purity. The results of k-means clustering depend on the initialization. For all the compared algorithms with different parameters and different number of selected features, we first repeat the clustering 20 times with random initialization and record the average results.

Since the optimal number of selected features is unknown in advance, to better evaluate the performance of unsupervised feature selection algorithms, we finally report the averaged results over different number of selected features (the range of selected features for each data set can be found in Table 1) with standard derivation.

The clustering results in terms of ACC, NMI and Purity are reported in Table 2, Table 3 and Table 4, respectively. For different feature selection algorithms, the results in each cell of Table 2, Table 3 and 4 are the mean ± standard deviation. The last row of Table 2, Table 3 and 4 show the averaged results of all the algorithms over the 6 datasets. The best results with highest value of these unsupervised feature selection algorithms are highlighted in boldface.

Compared with clustering using all features, our method can achieve comparable results with less than 50 features.

These results can well demonstrate the effectiveness and efficiency of unsupervised feature selection algorithm. These unsupervised feature selection algorithms not only can largely reduce the number of features facilitating the latter learning process, but can also often improve the clustering performance. It can also be observed that our method consistently produces better performance than the other unsupervised feature selection algorithms. In particular, our method achieves 20.87%, 19.85% and 14.29% improvement in terms of accuracy, NMI and purity respectively with the second best algorithm.

The details of clustering results on each data set have been shown in Fig 1, 2, 3, 4, 5, 6. The squared black line denotes the proposed method, other methods include AllFea, MaxVar, LapScore, MCFS, LGD and LLEScore. It can be observed that our method largely improves the clustering results under most of the feature numbers.

F. COMPARISON OF REDUNDANCY

To fully investigate the effectiveness of the proposed method, we take the redundancy metric to measure the redundancy of the selected subset of features. The results in terms of redundancy are reported in Table 5. For different feature selection algorithms, the results in each cell of Table 5 are the mean ± standard deviation. The last row of Table 5 shows the averaged results of all the algorithms over the 6 datasets. The best results with lowest value of these unsupervised feature selection algorithms are highlighted in boldface.

From Table 5, we can see that: our method achieves the lowest redundancy on selected features. Compared with the second lowest algorithm MCFS, we further get 25.83% improvement. The details of redundancy on different number of selected features are provided in Fig. 7. The redundancy for features selected by filter-based methods, i.e., MaxVar, LapScore and LLE score are higher than other two methods. These results well demonstrate that the clustering results could be improved with more compact subset of features by redundancy minimization.

V. CONCLUSION

In this paper we construct the simple k-nearest neighbor graphs using all candidate features and single feature to characterize the local structure of data. We further propose to minimize the linear reconstruction weight between the nearest neighbor graphs constructed from all candidate features and each single feature. Our method with the global optimal weights actually selects those features with highest relevance and lowest redundancy simultaneously, which makes it more suitable for the task of unsupervised feature selection. Experimental results on many benchmark data sets demonstrate that the proposed method outperforms many state-of-the-art unsupervised feature selection methods.

REFERENCES

[1] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

[2] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, Oct. 2003.

[3] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[4] J. Yin, W. Zeng, and L. Wei, "Optimal feature extraction methods for classification methods and their applications to biometric recognition," *Knowl.-Based Syst.*, vol. 99, no. 1, pp. 112–122, 2019.

[5] N. Gu, M. Fan, L. Du, and D. Ren, "Efficient sequential feature selection based on adaptive eigenspace model," *Neurocomputing*, vol. 161, pp. 199–209, Aug. 2015.

[6] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $l_{2,1}$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1813–1821.

[7] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proc. SIAM Int. Conf. Data Mining*, 2007, pp. 641–646.

[8] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010.

[9] M. Fan, X. Zhang, L. Du, L. Chen, and D. Tao, "Semi-supervised learning through label propagation on geodesics," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1486–1499, May 2017.

[10] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, Jan. 2004.

[11] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2010, pp. 333–342.

[12] L. Shi, L. Du, and Y.-D. Shen, "Robust spectral learning for unsupervised feature selection," in *Proc. IEEE 14th Int. Conf. Data Mining*, Dec. 2014, pp. 977–982.

[13] L. Du and Y.-D. Shen, "Unsupervised feature selection with adaptive structure learning," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2015, pp. 209–218.

[14] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. K. Shiu, "Unsupervised feature selection by regularized self-representation," *Pattern Recognit.*, vol. 48, no. 2, pp. 438–446, 2015.

[15] Y. Liu, K. Liu, C. Zhang, J. Wang, and X. Wang, "Unsupervised feature selection via diversity-induced self-representation," *Neurocomputing*, vol. 219, pp. 350–363, Jan. 2017.

[16] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, and S. Zhang, "Graph self-representation method for unsupervised feature selection," *Neurocomputing*, vol. 220, pp. 130–137, Jan. 2017.

[17] W. Zhou, C. Wu, Y. Yi, and G. Luo, "Structure preserving non-negative feature self-representation for unsupervised feature selection," *IEEE Access*, vol. 5, pp. 8792–8803, 2017.

[18] L. Du, Z. Shen, X. Li, P. Zhou, and Y.-D. Shen, "Local and global discriminative learning for unsupervised feature selection," in *Proc. IEEE 13th Int. Conf. Data Mining*, Dec. 2013, pp. 131–140.

[19] M. Fan, X. Chang, X. Zhang, D. Wang, and L. Du, "Top-k supervise feature selection via ADMM for integer programming," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1646–1653.

[20] S. Wang and H. Wang, "Unsupervised feature selection via low-rank approximation and structure learning," *Knowl.-Based Syst.*, vol. 124, pp. 70–79, May 2017.

[21] J. Dai, Y. Chen, Y. Yi, J. Bao, L. Wang, W. Zhou, and G. Lei, "Unsupervised feature selection with ordinal preserving self-representation," *IEEE Access*, vol. 6, pp. 67446–67458, 2018.

[22] P. Zhu, Q. Xu, Q. Hu, and C. Zhang, "Co-regularized unsupervised feature selection," *Neurocomputing*, vol. 275, pp. 2855–2863, Jan. 2018.

[23] R. Zhang, F. Nie, Y. Wang, and X. Li, "Unsupervised feature selection via adaptive multimeasure fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[24] Y. Zhang, Z. Zhang, S. Li, J. Qin, G. Liu, M. Wang, and S. Yan, "Unsupervised nonnegative adaptive feature extraction for data representation," *IEEE Trans. Knowl. Data Eng.*, to be published.

[25] X. Du, F. Nie, W. Wang, Y. Yang, and X. Zhou, "Exploiting combination effect for unsupervised feature selection by $\ell_{2,0}$ norm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 201–214, Jan. 2019.

[26] X. Li, H. Zhang, R. Zhang, Y. Liu, and F. Nie, "Generalized uncorrelated regression with adaptive graph for unsupervised feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1587–1595, May 2019.

[27] J. Yin, Z. Lai, W. Zeng, and L. Wei, "Local sparsity preserving projection and its application to biometric recognition," *Multimedia Tools Appl.*, vol. 77, no. 1, pp. 1069–1092, 2018.

[28] X. Chen, G. Yuan, W. Wang, F. Nie, X. Chang, and J. Z. Huang, "Local adaptive projection framework for feature selection of labeled and unlabeled data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6362–6373, Dec. 2018.

[29] L. Wei, R. Zhou, J. Yin, C. Zhu, X. Zhang, and H. Liu, "Latent graph-regularized inductive robust principal component analysis," *Knowl.-Based Syst.*, vol. 177, no. 1, pp. 68–81, 2019.

[30] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18, 2005, pp. 507–514.

[31] M. Liu and D. Zhang, "Sparsity score: A novel graph-preserving feature selection method," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 28, no. 4, pp. 14500091–145000929, 2014.

[32] D. Wang, F. Nie, and H. Huang, "Feature selection via global redundancy minimization," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2743–2755, Oct. 2015.

[33] C. Yao, Y.-F. Liu, B. Jiang, J. Han, and J. Han, "LLE score: A new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5257–5269, Nov. 2017.

[34] F. Nie, S. Yang, R. Zhang, and X. Li, "A general framework for auto-weighted feature selection via global redundancy minimization," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2428–2438, Dec. 2018.

[35] P. Zhou, L. Du, M. Fan, and Y.-D. Shen, "An LLE based heterogeneous metric learning for cross-media retrieval," in *Proc. SIAM Int. Conf. Data Mining*, 2015, pp. 64–72.

[36] L. Lovász and M. Plummer, "Matching theory," *Ann. statist.*, vol. 29, 1986.

[37] L. Du and Y.-D. Shen, "Joint clustering and feature selection," in *Proc. Int. Conf. Web-Age Inf. Manage.*, 2013, pp. 241–252.

[38] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.

[39] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Neural Inf. Process. Syst.*, 2005, pp. 1601–1608.

LIANG DU received the B.E. degree in software engineering from Wuhan University, in 2007, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, in 2013. He is currently a Lecturer with Shanxi University. Prior to that, he was an Assistant Researcher with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. From 2013 to 2014, he was a Software Engineer with the Alibaba Group. He is particularly interested in the following topics: clustering with noise and heterogeneous data, ranking for feature selection, active learning, and document summarization. He has published over 30 papers in top conferences and journals, including KDD, IJCAI, AAAI, ICDM, TKDE, SDM, and CIKM.

CHAOHONG REN received the B.S. degree from Jinzhong University, Shanxi, China, in 2017. He is currently pursuing the M.S. degree with Shanxi University, Taiyuan, China. His research interest includes unsupervised feature selection.

XIAOLIN LV received the B.S. degree from Taiyuan Normal University, Shanxi, China, in 2017. She is currently pursuing the M.S. degree with Shanxi University, Taiyuan, China. Her research interest includes unsupervised feature selection.

YAN CHEN received the B.S. degree from the North University of China, Taiyuan, China, in 2016. She is currently pursuing the M.S. and Ph.D. degrees with Shanxi University, Taiyuan, China. Her research interest includes unsupervised feature learning with applications.

PENG ZHOU received the B.E. degree in computer science and technology from the University of Science and Technology of China, in 2011, and the Ph.D. degree in computer science from the Institute of Software, University of Chinese Academy of Sciences, in 2017. He is currently a Lecturer with Anhui University. His research interests include machine learning, data mining, and artificial intelligence.

ZHIGUO HU received the B.S. degree in command and control engineering from Artillery College, Shenyang, China, in 2001, the M.S. degree in command and control engineering from Artillery College, Hefei, China, in 2006, and the Ph.D. degree in computer science from TongJi University, China, in 2012. In 2015, he joined Shanxi University, Taiyuan, where he is currently a Lecturer with the School of Computer and Information Technology. His research interests include network measurement, data mining, and machine leaning.