Noname manuscript No.
(will be inserted by the editor)

Nonparametric Redundant Features Removal for Unsupervised Feature Selection: An Empirical Study based on Sparse Feature Graph

Pengfei Xu · Shuchu Han · Hao Huang · Hong Qin

Received: date / Accepted: date

Abstract It is known that the redundant features existing in high dimensional datasets will affect the performance of learning and mining algorithms. How to detect and remove them is a challenge in machine learning and data mining research. In this work, we propose a graph based approach to search and remove those redundant features automatically for high dimensional data. A sparse graph is generated at feature side and is used to learn the redundant relationship among features. We name this novel graph as *sparse feature graph*(SFG). To avoid the inaccurate distance information among high dimensional vectors, the construction of SFG does not utilize the pairwise relationship among samples, which means the structure info of the data is not considered. This proposed SFG based redundant feature removal algorithm is also a nonparametric one as the user does not need to tune the learning model. We treat this proposed redundant feature removal algorithm as a data preprocessing approach for existing popular unsupervised feature selection algorithms like Multi-cluster Feature Selection (MCFS) which requires accurate cluster structure information from samples. Our experimental results on benchmark datasets show that the proposed SFG and redundant feature remove algorithm can improve the performance of those unsupervised feature selection algorithms consistently.

Keywords Sparse Graph Representation · Unsupervised Feature Selection · Dense Subgraph

Pengfei Xu
College of Information Science and Technology
Beijing Normal University
E-mail: xupf@bnu.edu.cn

Shuchu Han
Department of Computer Science
Stony Brook University
E-mail: shuchu.han@gmail.com

Hao Huang
Machine Learning Laboratory
General Electric Global Research
E-mail: haohuanghw@gmail.com

Hong Qin
Department of Computer Science
Stony Brook University
E-mail: qin@cs.stonybrook.edu

1 Introduction

In unsupervised feature selection algorithms, the structure of data is used to generate indication vectors for informative features selection. The structure of data could be local manifold structure [He and Bao(2011)] [Hou and Wu(2014)], global structure [Liu and Liu(2014)] [Zhao and Ye(2013)], discriminative information [Yang and Zhou(2011)] [Li and Lu(2012)] and etc. To model such structures of data, existing methods like Gaussian similarity graph, or k-nearest neighbor graph are very popular in machine learning research. All these similarity graphs are built based on the pairwise distance like Euclidean distance ($\mathcal{L}_2$ norm) or Manhattan distance ($\mathcal{L}_1$ norm) among samples (vectors). As we can see, the pairwise distance is crucial to the quality of indication vectors, and the success of unsupervised feature selection depends on the accuracy of these indication vectors.

When the dimension of sample vector becomes high, especially for high dimensional datasets, we will meet the so-called curse of high dimensionality issue [Beyer and Shaft(1999)]. That means the differentiating ability of pairwise distance will degraded rapidly when the dimension of data goes higher, and the nearest neighbor indexing will give inaccurate results [Weber and Blott(1998)] [Aggarwal and Keim(2001)]. As a result,

the description of data structure by using similarity graphs will be not precise and even wrong. This create an embarrassing *chicken-and-egg* problem [Du and Shen(2015)] for unsupervised feature selection algorithms: "the success of feature selection depends on the quality of indication vectors which are related to the structure of data. But the purpose of feature selection is to giving more accurate data structure."

Most existing unsupervised feature selection algorithms use all *original features* [Du and Shen(2015)] to represent the (cluster) structure of data. As a result, the obtained data structure information will not as accurate as the intrinsic one it should be. To remedy this problem, dimensionality reduction techniques are required. For example, Principal Component Analysis (PCA) and Random Projection (RP) are popular methods in machine learning research. However, most of them will project the data matrix into another (lower dimensional) space with the constraint to approximate the original pairwise similarities. As a result, we lose the physical meaning or original features and the meaning of projected features are unknown.

In this study, we proposed a graph-based approach to reduce the data dimension by removing redundant features. Without lose of generality, we categorize features into three groups [Dash and Liu(1997)]: *relevant feature*,*irrelevant feature* and *redundant feature*. A feature $\boldsymbol{f_i}$ is relevant or irrelevant based on it's correlation with indication vectors (or target vectors named in other articles) $\boldsymbol{Y} = \{\boldsymbol{y_i}, i \in [1, k]\}$. For supervised feature selection algorithms [Tibshirani(1996)] [Peng and Ding(2005)] [Robnik-Šikonja and Kononenko(2003)], these indication vectors usually relate to class labels. For unsupervised scenario [Dy and Brodley(2004)] [Cai and He(2010)], as we mentioned early, they follow the structure of data. Redundant features are features that highly correlated to other features, and have no contribution or trivial contribution to the target learning task. The formal definition of redundant feature is by [Yu and Liu(2004)] based on the Markov blanket [Koller(1996)].

Based on the philosophy of sparse learning based MCFS algorithm proposed by [Cai and He(2010)], a feature could be redundant to another single feature, or to a subset of features. In this work, we propose a graph based approach to identify these two kind of redundancy at the same time. The first step is to build a Sparse Feature Graph (SFG) at feature side based on sparse representation concept from subspace clustering theory [Elhamifar and Vidal(2013)]. Secondly, we review the quality of sparse representation of each single feature vector and filtered out those failed ones. In the last, we defined Local Compressible Subgraphs (LCS) to represent those local feature groups that are very redundant. Moreover, a greedy local search algorithm is designed to discover all those LCSs. Once we have all LCSs, we pick the feature which has the highest node indegree as the representative feature and treat all other as redundant features. With this approach, we obtain a new data matrix with reduced size and alleviate the curse of dimensional issues.

To be specific, the contribution of our study can be highlighted as:

- We propose sparse feature graph to model the feature redundancy existing in high dimensional datasets. The sparse feature graph inherits the philosophy of sparse learning based unsupervised feature selection framework. The sparse feature graph not only records the redundancy between two features but also show the redundancy between one feature and a subset of features.
- We propose local compressible subgraph to represent redundant feature groups. And also design a local greedy search algorithm to find all those subgraphs.
- We reduce the dimensionality of input data and alleviate the curse of dimensional issue through redundant features removal. With a more accurate data structure, the *chicken-and-egg* problem for unsupervised feature selection algorithms are remedied in certain level. One elegant part of our proposed approach is to reduce the data dimension without any pairwise distance calculation.
- Abundant experiments and analysis over twelve high dimensional datasets from three different domains are also presented in this study. The experiment results show that our method can obtain better data structure with reduced size of dimensionality, and proof the effectiveness of our proposed approach.

The rest of paper is organized as follows. The first section describe the math notation used in our work. The Section 2 introduces the background , motivation and preliminaries of our problem. In Section 3, we define the problem we are going to solve. In Section 4, we present our proposed sparse feature graph algorithm and discuss the sparse representation error problem. We also introduce the local compressible subgraph and related algorithm. The experiment results are reported in Section 5, and a briefly reviewing of related works is given in Section 6. Finally, we conclude our study in last Section 7.

2 Math Notation

Throughout this paper, matrices are written as boldface capital letters and vectors are represented as boldface

lowercase letters. Let the data matrix be represented as $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, while each row is a sample (or instance), and each column means a feature. If we view the data matrix $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n}]^T, \boldsymbol{x_i} \in \mathbb{R}^{d\times 1}$ from feature side, it can be seen as $\boldsymbol{F} = \boldsymbol{X}^T = [\boldsymbol{f_1}, \boldsymbol{f_2}, \cdots, \boldsymbol{f_d}], \boldsymbol{f_i} \in \mathbb{R}^{n\times 1}(1 \le i \le d)$.

3 Background and Preliminaries

3.1 Unsupervised Feature Selection

In unsupervised feature selection framework, we don't have label information to determine the feature relevance. Instead, the data similarity or manifold structure constructed from the whole feature space are used as criteria to select features. Among all those algorithms of unsupervised feature selection, the most famous one is MCFS. The MCFS algorithm is a sparse learning based unsupervised feature selection method which can be illustrated as figure 1. The core idea of MCFS is to

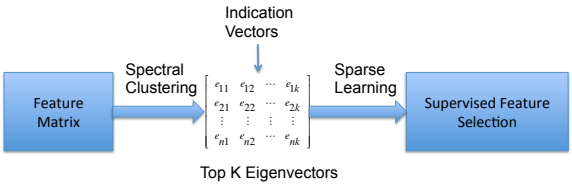

Fig. 1 The framework of sparse learning based unsupervised feature selection.

use the eigenvectors of graph Lapalcian over similarity graph as indication vectors. And then find set of features that can approximate these eigenvectors through sparse linear regression. Let us assume the input data has number K clusters that is known beforehand (or an estimated K value by the expert's domain knowledge). The top K non-trivial eigenvectors, $\boldsymbol{Y} = [\boldsymbol{y_1}, \cdots, \boldsymbol{y_k}]$, form the spectral embedding $\boldsymbol{Y}$ of the data. Each row of $\boldsymbol{Y}$ is the new coordinate in the embedding space. To select the relevant features, MCFS solves K sparse linear regression problems between $\boldsymbol{F}$ and $\boldsymbol{Y}$ as:

$$\min_{\alpha_i} \|\boldsymbol{y}_i - \boldsymbol{F}\boldsymbol{\alpha}_i\|^2 + \beta\|\boldsymbol{\alpha}_i\|_1, \tag{1}$$

where $\boldsymbol{\alpha}_i$ is a n-dimensional vector and it contains the combination coefficients for different features $\boldsymbol{f}_i$ in approximating $\boldsymbol{y}_i$. Once all coefficients $\boldsymbol{\alpha}_i$ are collected, features will be ranked by the absolute value of these coefficients and top features are selected. This can be show by a weighted directed bipartite graph as following:

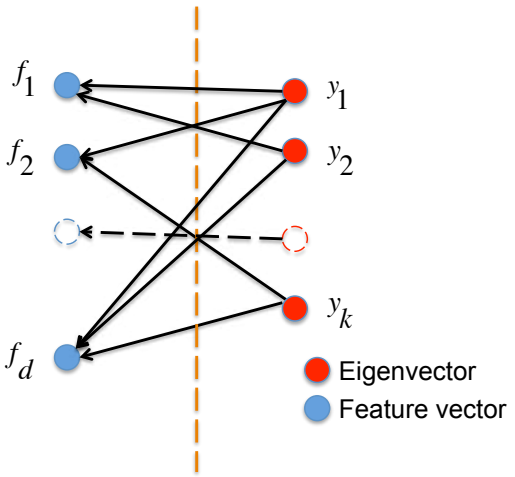

Fig. 2 Sparse learning bipartite graph for MCFS.

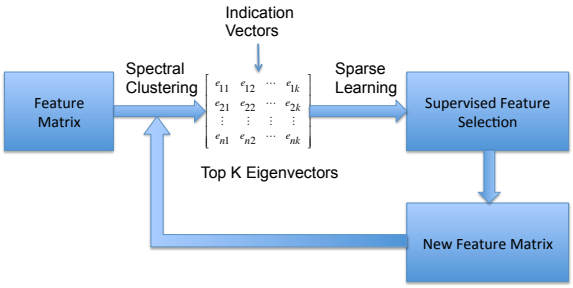

Fig. 3 Unsupervised Feature Selection with Adaptive Structure Learning.

3.2 Adaptive Structure Learning for High Dimensional Data

As we can seen, the MCFS uses whole features to model the structure of data. That means the similarity graph such as Gaussian similarity graph is built from all features. This is problematic when the dimension of data vector goes higher. To be specific, the pairwise distance between any two data vectors becomes almost the same, and as a consequence of that, the obtained structural information of data is not accuracy. This observation is the motivation of unsupervised Feature Selection with Adaptive Structure Learning (FSASL) algorithm which is proposed by Du et al. [Du and Shen(2015)]. The idea of FSASL is to repeat MCFS iteratively with updating selected feature sets. It can be illustrated as following: FASAL is an iterative algorithms which keeps pruning irrelevant and noisy features to obtain better manifold structure while improved structural info can help to search better relevant features. FASAL shows better performance in normalized mutual information and accuracy than MCFS generally. However, it's very time consuming since it is an iterative algorithm includes many eigen-decompositions.

3.3 Redundant Features

For high dimensional data $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, it exists information redundancy among features since $d \ll n$. Those redundant features can not provide further performance improvement for ongoing learning task. Instead, they impair the efficiency of learning algorithm to find intrinsic data structure.

In this section, we describe our definition of feature redundancy. Unlike the feature redundancy defined bt Markov blanket [Yu and Liu(2004)] which is popular in existing research works, our definition of feature redundancy is based on the linear correlation between two vectors (the "vector" we used here could be a feature vector or a linear combination of several feature vectors.) To measure the redundancy between two vectors $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, squared cosine similarity[Wang and Wei(2015)] is used:

$$R_{ij} = cos^2(\boldsymbol{f}_i, \boldsymbol{f}_j). \tag{2}$$

By the math definition of cosine similarity, it is straightforward to know that a higher value of $R_{i,j}$ means high redundancy existing between $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$. For example, feature vector $\boldsymbol{f}_i$ and its duplication $\boldsymbol{f}_i$ will have R_{ii} value equals to one. And two orthogonal feature vectors will have redundancy value equals to zero.

4 Problem Statement

In this work, our goal is to detect those redundant features existing in high dimensional data and obtain a more accurate intrinsic data structure. To be specific:

Problem 1 Given a high dimensional data represented in the form of feature matrix $\boldsymbol{X}$, how to remove those redundant features $f_{(\cdot)} \in \boldsymbol{X}^T$ for unsupervised feature selection algorithms such as MCFS?

Technically, the MCFS algorithm does not involve redundant features. However, the performance of MCFS depends on the quality of indication vectors which are used to select features via sparse learning. And those indication vectors are highly related to the intrinsic structure of data which is described by the selected features and given distance metric. For example, the MCFS algorithm uses all features and Gaussian similarity to represent the intrinsic structure. This is the discussed 'chicken-and-egg" problem [Du and Shen(2015)] between structure characterization and feature selection. The redundant and noise features will lead to an inaccurate estimation of data structure. As a result, it's very demanding to remove those redundant (and noise) features before the calculation of indication vectors.

5 Algorithm

In this section, we present our graph-based algorithm to detect and remove redundant features existing in high dimensional data. First, the sparse feature graph that modeling the redundancy among feature vectors will be introduced. Secondly, the sparse representation error will be discussed. In the last, the local compressible subgraph is proposed to extract redundant feature groups.

5.1 Sparse Feature Graph (SFG)

The most popular way to model the redundancy among feature vectors is correlation such as Pearson Correlation Coefficient (PCC). The correlation value is defined over two feature vectors, and it's a pairwise measurement. However, there also exiting redundancy between one feature vector and a set of feature vectors according to the philosophy of MCFS algorithm. In this section, we present SFG, which model the redundancy not only between two feature vectors but also one feature vector and a set of feature vectors.

The basic idea of sparse feature graph is to looking for a sparse linear representation for each feature vector while using all other feature vectors as dictionary. For each feature vector $\boldsymbol{f}_i$ in features set $\boldsymbol{F} = [\boldsymbol{f_1}, \boldsymbol{f_2}, \cdots, \boldsymbol{f_d}]$, SFG solves the following optimization problem:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{d-1}} \|\boldsymbol{f_i} - \boldsymbol{\Phi^i}\boldsymbol{\alpha_i}\|_2^2, \quad \text{s.t. } \|\boldsymbol{\alpha_i}\|_0 < L, \tag{3}$$

where $\boldsymbol{\Phi}^i = [\boldsymbol{f_1}, \boldsymbol{f_2}, \cdots, \boldsymbol{f_{i-1}}, \boldsymbol{f_{i+1}}, \cdots, \boldsymbol{f_d}]$ is the dictionary of f_i and each column of $\boldsymbol{\Phi}^i$ is a selected feature from data matrix $\boldsymbol{X}$. L is a constraint to limit the number of nonzero coefficients. In SFG, we set it to the number of features d. The $\boldsymbol{\alpha}_i$ is the coefficient of each atom of dictionary $\boldsymbol{\Phi}^i$. This coefficient vector not only decides the edge link to $\boldsymbol{f_i}$ but also indicates the weight of that connection. The resulted SFG is a weighted directed graph and may have multiple components.

To solve the optimization problem 3, we use Orthogonal Matching Pursuit (OMP) solver [You and Vidal(2016)] here since the number of features in our datasets is larger than 1,000. We modify the stop criterion of OMP by checking the value change of residual instead of residual itself or the maximum number of supports. The reason is that we want the number of supports (or say, the number of edge connections) to follow the raw data property. Real world datasets are always noisy and messy. It's highly possible that several feature vectors may fail to find a correct sparse linear

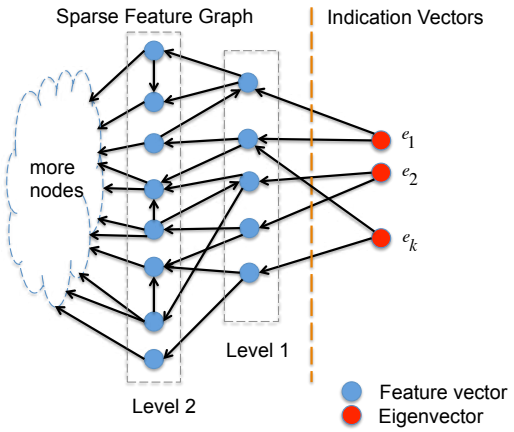

Fig. 4 Sparse feature graph and its relation with indication vectors. The level 1 features are direct sparse representation of those calculated indication vectors. The level 2 features only have representation relationship with level 1 features but not with indication vectors.

representation through OMP. If we set residual or maximum of supports as criteria, we can not differentiate the successful representations and the failed ones.

The OMP solver and SFG algorithm can be described as following.

Algorithm 1: Orthogonal Matching Pursuit (OMP)

Input : $\boldsymbol{\Phi} = [\boldsymbol{f_1}, \boldsymbol{f_2}, \cdots, \boldsymbol{f_{i-1}}, \boldsymbol{f_{i+1}}, \cdots, \boldsymbol{f_d}] \in \mathbb{R}^{n\times(d-1)}$, $\boldsymbol{f_i} \in \mathbb{R}^n$, ϵ.
Output: Coefficient $\boldsymbol{\alpha}_i$.
Initialize residual difference threshold $r_0 = 1.0$,
residual $\boldsymbol{q}_0 = \boldsymbol{f}_i$, support set $\Gamma_0 = \emptyset$, $k = 1$;
while $k \leq d-1$ *and* $|r_k - r_{k-1}| > \epsilon$ **do**
  Search the atom which most reduces the objective:
  $j^* = \arg\min_{j\in\Gamma^C} \left\{ \min_{\alpha} \|\boldsymbol{f}_i - \boldsymbol{\Phi}_{\Gamma\cup\{j\}}\boldsymbol{\alpha}\|_2^2 \right\}$;
  Update the active set:
  $\Gamma_k = \Gamma_{k-1} \cup \{j^*\}$;
  Update the residual (orthogonal projection):
  $\boldsymbol{q}_k = (I - \boldsymbol{\Phi}_{\Gamma_k}(\boldsymbol{\Phi}_{\Gamma_k}^T\boldsymbol{\Phi}_{\Gamma_k})^{-1}\boldsymbol{\Phi}_{\Gamma_k}^T)\boldsymbol{f}_i$;
  Update the coefficients:
  $\boldsymbol{\alpha}_{\Gamma_k} = (\boldsymbol{\Phi}_{\Gamma_k}^T\boldsymbol{\Phi}_{\Gamma_k})^{-1}\boldsymbol{\Phi}_{\Gamma_k}^T\boldsymbol{f}_i$;
  $r_k = \|\boldsymbol{q}_k\|_2^2$;
  $k \leftarrow k+1$;
end

5.2 Sparse Representation Error

In our modified OMP algorithm 1, we set a new stop criterion of searching sparse representation solution for each feature vector $\boldsymbol{f}_i$. Instead of keep searching until arriving a minimization error, we stop running while the solver could not reduce the length of residual vector

Algorithm 2: Sparse Feature Graph

Input : Data matrix $\boldsymbol{F} = [\boldsymbol{f_1}, \boldsymbol{f_2}, \cdots, \boldsymbol{f_d}] \in \mathbb{R}^{n\times d}$;
Output: Adjacent matrix $\boldsymbol{W}$ of Graph $\boldsymbol{G} \in \mathbb{R}^{d\times d}$;
Normalize each feature vector $\boldsymbol{f_i}$ with $\|\boldsymbol{f_i}\|_2^2 = 1$;
for $i = 1, \cdots, d$ **do**
  Compute $\boldsymbol{\alpha}_i$ from OMP($\boldsymbol{F}_{-i}$,$\boldsymbol{f}_i$) using algorithm 1;
end
Set adjacent matrix $W_{ij} = \boldsymbol{\alpha_i(j)}$ if $i > j$,
$W_{ij} = \boldsymbol{\alpha_i(j-1)}$, if $i < j$ and $W_{ij} = 0$ if $i == j$;

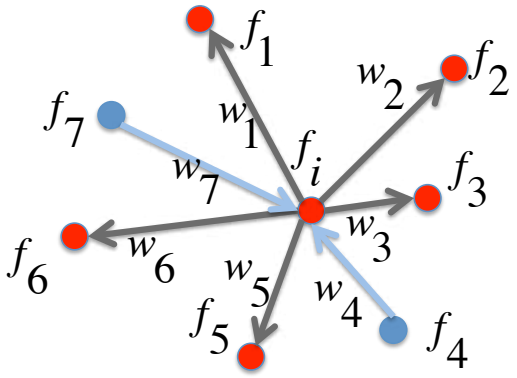

Fig. 5 Illustration of sparse representation error. SFG is a weighted directed graph.

anymore. To be specific, the 2-norm of residual vector is monitored and the solver will stop once the change of this value small than a user specified threshold.

The reason we use this new stop criterion is that several feature vectors may not find correct sparse representation in current dataset, and the ordinary OMP solver will return a meaningless sparse representation when the maximum iteration threshold arrived. Since the goal of SFG is not to find a correct sparse representation for every feature vectors, we utilize the new stop criterion and add a filter process in our algorithm to identify those failed sparse representation.

To identify those failed sparse representation, we check the angle between the original vector and the linear combination of its sparse representation. In the language of SFG, we check the angle between a node (a feature vector) and the weighted combination of its one-ring neighbor. Only the neighbors of out edges will be considered. This can be illustrated by following figure 5. As the example in Figure 5, node $\boldsymbol{f}_i$ has seven one-ring neighbors. But only
$bmf_1, bmf_2, \boldsymbol{f}_3, \boldsymbol{f}_5, \boldsymbol{f}_6$ are its sparse representation and $\boldsymbol{f}_4$ and $\boldsymbol{f}_7$ are not. Then the sparse representation error ζ is calculated by:

$$\boldsymbol{f}_i^* = w_1\boldsymbol{f}_1 + w_2\boldsymbol{f}_2 + w_3\boldsymbol{f}_3 + w_5\boldsymbol{f}_5 + w_6\boldsymbol{f}_6,$$
$$\zeta = \arccos(\boldsymbol{f}_i, \boldsymbol{f}_i^*).$$

Once we have the SFG, we calculate the sparse representation errors for all nodes. A sparse representation is treated as fail if the angle ζ less than a user specified value. We will filter out these node which has failed representation by removing its out-edges.

5.3 Local Compressible Subgraph

We group high correlated features through *local compressible subgraphs*. The SFG $\boldsymbol{G}$ is a weighted directed graph. With this graph, we need to find all feature subsets that has very high redundancy. To archive this goal, we propose a local search algorithm with seed nodes to group those highly correlated features into many subgraphs which are named as *local compressible subgraphs* in this study. Our local search algorithm involves two steps, the first step is to sort all nodes by the in-degree. By the definition of SFG, the node with higher in-degree means it appears more frequently in other nodes' sparse representation. The second step is a local bread-first search approach which finds all nodes that has higher weight connections (in and out) to the growing subgraph. The detail subgraph searching algorithm can be described by: In Alg. 3, function $label(n)$ check the current label of node n, and $BFS(n, \theta, current\_label)$ function runs a local Breadth-First search for subgraph that has edge weight large than θ.

Algorithm 3: Local Compressible Subgraphs.

Input : Weighted directed graph $G = (V, E)$, edge weight threshold θ;
Output: Local compressible subgraphs C .

```
Tag all nodes with initial label 0;
Sort the nodes by its in-degree decreasingly;
current_label = 1;
for n = 1 : |V| do
    if label(n) != 0 then
        continue;
    end
    set label of node n to current_label;
    BFS(n, θ, current_label);
    current_label += 1;
end
/* current_label now has the maximum value of
   labels.                                    */
for i = 1 : current_label do
    Extract subgraph c_i which all nodes have label i;
    if |c_i| > 1 then
        add c_i to C;
    end
end
```

5.4 Redundant Feature Removal

The last step of our algorithm is to remove the redundant features. For each local compressible subgraph we found, we pick up the node which has the highest in-degree as the representative node of that local compressible subgraph. So the number of final feature vectors equals to the number of local compressible subgraphs.

6 Experiments

In this section, we present experimental results to demonstrate the effectiveness of our proposed algorithm. We first evaluate the spectral clustering performance before and after applying our algorithms. Secondly, we show the performance of MCFS with or without our algorithm. In the last, the properties of generated sparse graphs and sensitivity of parameters are discussed.

6.1 Experiment Setup

Datasets. We select twelve real-world high dimensional datasets [J. Li and Liu(2016)] from three different domains: Image, Text and Biomedical. The detail of each dataset is listed in Table 1. The datasets have sample size different from 96 to 8293 and feature size ranging from 1,024 to 18,933. Also, the datasets have class labels from 2 to 64. The purpose of this selection is to let the evaluation results be more general by applying datasets with various characteristics.

Name	#Sample	#Feature	#Class	Type
ORL	400	1024	40	Image
Yale	165	1024	15	Image
PIE10P	210	2420	10	Image
ORL10P	100	10304	10	Image
BASEHOCK	1993	4862	2	Text
RELATHE	1427	4322	2	Text
PCMAC	1943	3289	2	Text
Reuters	8293	18933	65	Text
lymphoma	96	4026	9	Biomedical
LUNG	203	3312	5	Biomedical
Carcinom	174	9182	11	Biomedical
CLL-SUB-111	111	11340	3	Biomedical

Table 1 High dimensional datasets.

Normalization. The features of each dataset are normalized to have unit length, which means $\|\boldsymbol{f_i}\|_2 = 1$ for all datasets.

Evaluation Metric. Our proposed algorithm is under the framework of unsupervised learning. Without loss of generality, the cluster structure of data is used for evaluation. To be specific, we measure the spectral clustering performance with Normalized Mutual Information (NMI) and Accuracy (ACC). NMI value ranges from 0.0 to 1.0, with higher value means better clustering performance. ACC is another metric to evaluate the clustering performance by measuring the fraction of its clustering result that are correct. Similar to NMI, its values range from 0 to 1 and higher value indicates better algorithm performance.

Suppose A is the clustering result and B is the known sample label vector. Let $p(a)$ and $p(b)$ denote the marginal probability mass function of A and B, and let $p(a,b)$ be the joint probability mass function of A and B. Suppose $H(A), H(B)$ and $H(A,B)$ denote the entropy of $p(a), p(b)$ and $p(a,b)$ respectively. Then the normalized mutual information NMI is defined as:

$$NMI(A,B) = \frac{H(A) + H(B) - H(A,B)}{max(H(A), H(B))} \tag{4}$$

Assume A is the clustering result label vector, and B is the known ground truth label vector, ACC is defined as:

$$ACC = \frac{\sum_{i=1}^{N} \delta(B(i), Map_{(A,B)}(i))}{N} \tag{5}$$

where N denotes the length of label vector, $\delta(a,b)$ equals to 1 if only if a and b are equal. $Map_{A,B}$ is the best mapping function that permutes A to match B.

6.2 Effectiveness of Redundant Features Removal

Our proposed algorithm removes many features to reduce the dimension size of all data vectors. As a consequence, the pairwise Euclidean distance is changed and the cluster structure will be affected. To measure the effectiveness of our proposed algorithm, we check the spectral clustering performance before and after redundant feature removal. If the NMI and ACC values are not changed to much and stay in the same level, the experiment results show that our proposed algorithm is correct and effective.

The spectral clustering algorithm we used in our experiments is the Ng-Jordan-Weiss (NJW) algorithm [Ng et al(2001)]. The Gaussian similarity graph is applied here as the input and parameter σ is set to the mean value of pairwise Euclidean distance among all vectors.

Our proposed LCS algorithm includes a parameter θ which is the threshold of redundancy. It decides the number of redundant features implicitly, and affects the cluster structure of data consequently. In our experiment design, we test different θ values ranging from 90% to 10% with step size equal to 10%: $\theta = [0.9, 0.8, 0.7, \cdots, 0.1]$.

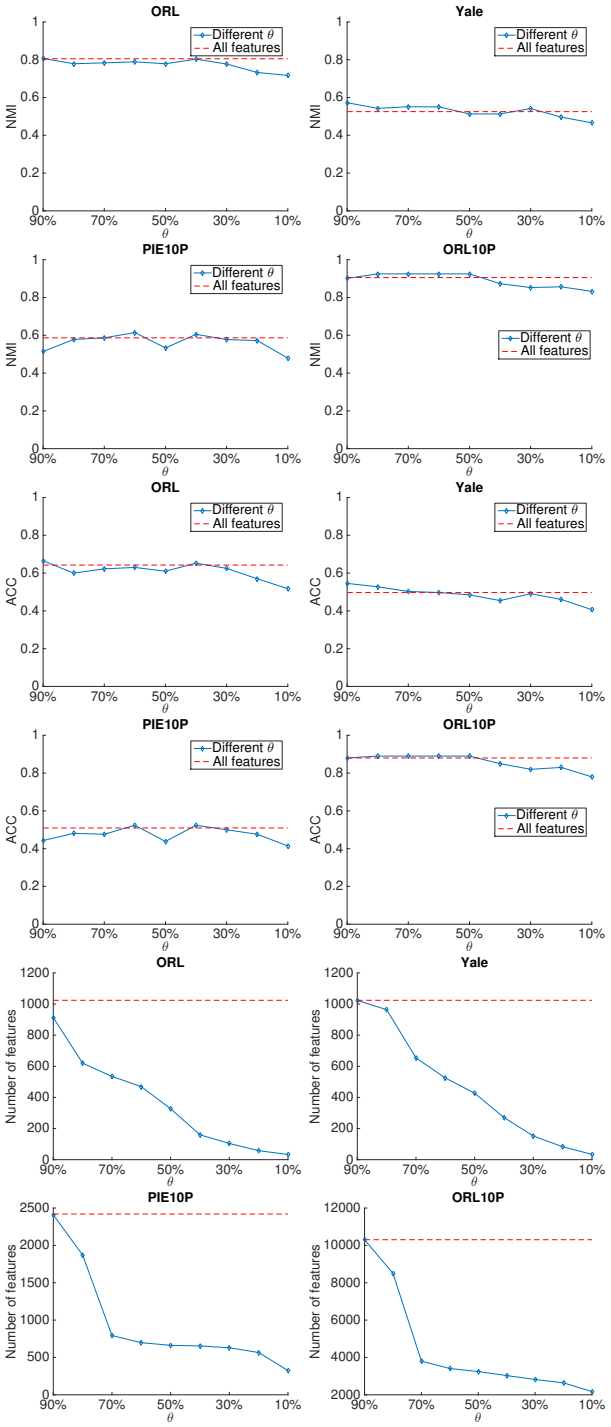

Fig. 6 Spectral clustering performance of Image datasets with different parameter θ. Top row: NMI; Middle row: ACC; Bottom row: number of features, the red dash line means the size of raw dataset.

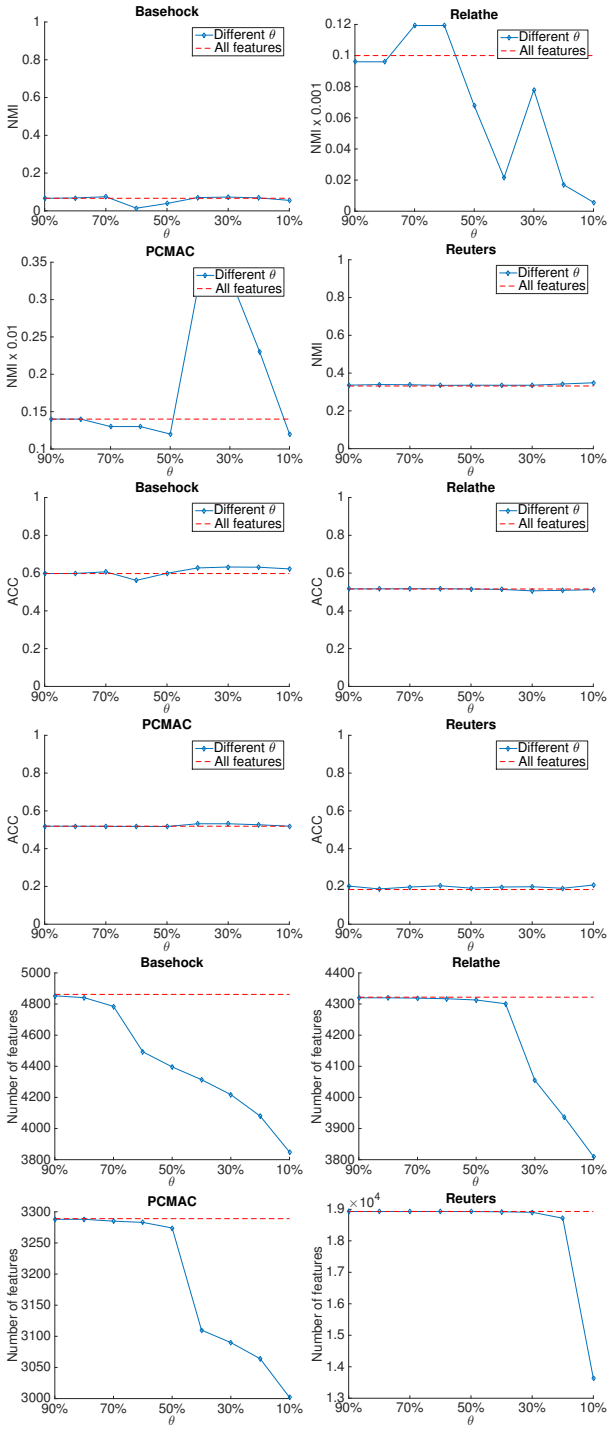

Fig. 7 Spectral clustering performance of Text datasets with different parameter θ. Top row: NMI; Middle row: ACC; Bottom row: number of features, the red dash line means the size of raw dataset.

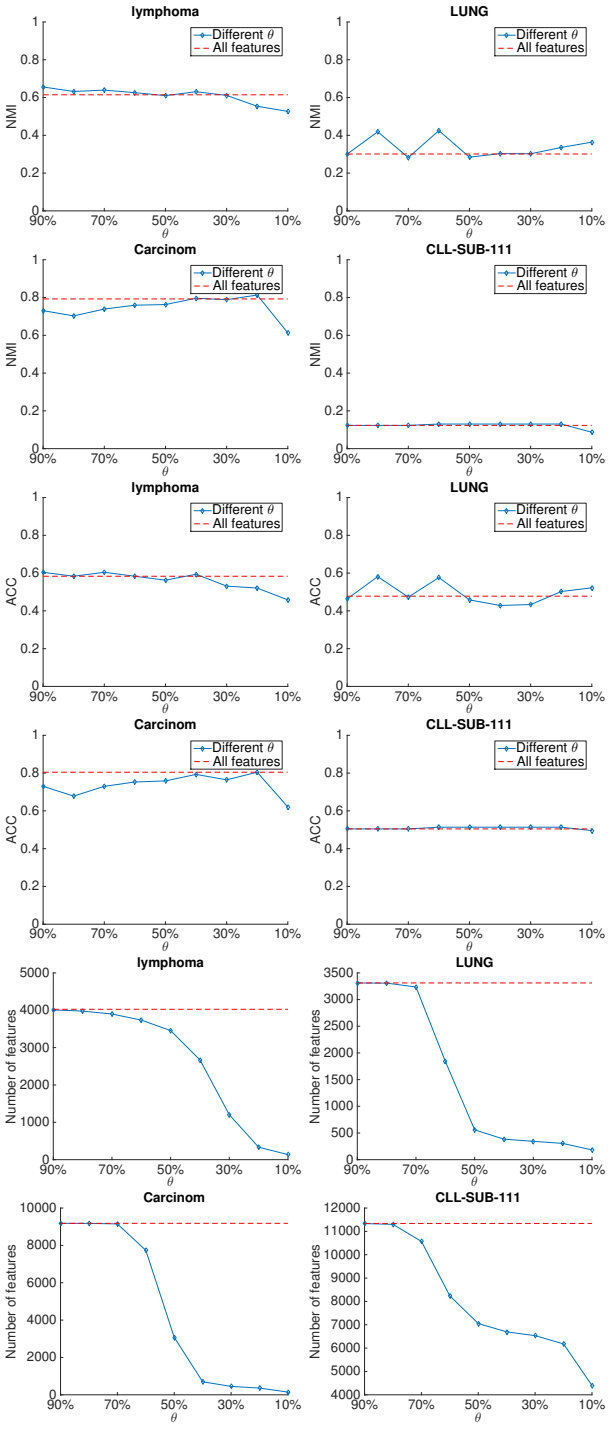

Fig. 8 Spectral clustering performance of Biomedical datasets with different parameter θ. Top row: NMI; Middle row: ACC; Bottom row: number of features, the red dash line means the size of raw dataset.

We present our experiment results for image datasets, text datasets, and biological datasets in Figure 6, Figure 7 and Figure 8 respectively. For each dataset, we show the NMI, ACC performance with different θ and comparing with original spectral clustering performance by using all features. From the experimental results, we can read that: **Even when θ is reduced to 30%, the NMI and ACC values are staying in same level as original data.** When θ equals to 30%, it means the edges of SFG that with weights (absolute value) in the highest 70% value range are removed. (It does not mean that 70% of top weights edges are removed). This observation validate the correctness of our proposed algorithm.

6.3 Performance of MCFS

Our proposed algorithm is targeting for unsupervised feature selection. And the quality of indication vectors (or the spectral clustering performance based on eigenvectors) is an important factor evaluate the effectiveness of our proposed algorithm. In this section, we evaluate the MCFS performance over the redundant feature removed data, and comparing with the raw data that without any feature removal.

The spectral clustering performance is measured for different input data from original whole feature data to processed ones by our proposed algorithm with different θ. We report the experiment results over image datasets and biological datasets in this section. For text datasets, the feature vectors of them are very sparse, and our eigen decomposition process are always failed and we only can collect partial results. For fair evaluation, we omit the experiment results of text datasets in this work. The result of MCFS performance shows from Table 2 to Table 17.

For each dataset, we set the number of selected features ranging from $[5, 10, 15, \cdots, 60]$, which has 11 different sizes in total. The parameter θ is configured from 0.9 to 0.1 with stepsize equals to 0.1.

We report the experimental results in tables (from Table 2 to Table 17). For each table, the first row means the number of features that used as input of MCFS. The first column is the number of selected features by MCFS algorithm. The baseline is in the second column, which is the testing result of MCFS algorithm with raw data. The hyphens in the tables means the number of selected features is larger than the feature size of input data, which means invalid test. To show the effectiveness of our algorithm, we also mark those NMI and ACC scores that larger or equals to baseline.

6.4 Sparse Representation Errors

With the design of our modified OMP solvers, there will be failed/wrong sparse representations existing in generated sparse feature graph. The meaning of these edge connections and edge weights are invalid. And they should be removed from the SFG since wrong connections will deteriorate the accuracy of feature redundancy relationship. To validate the sparse representation, we check the angle between original feature vector and the linear weighted summation resulted vector (or recover signal from sparse coding point of view) from its sparse representation. If the angle lower than a threshold, we remove all out-edges from the generated sparse feature graph. To specify the threshold, we learn it from the empirical results of our selected twelve datasets. The distribution (or histogram) result of angle values is presented in figure 9.

7 Related Works

Remove redundant features is an important step for feature selection algorithms. Prestigious works include [Yu and Liu(2004)] which gives a formal definition of redundant features. Peng et al. [Peng and Ding(2005)] propose a greedy algorithm (named as mRMR) to select features with minimum redundancy and maximum dependency. Zhao et al. [Zhao and Liu(2010)] develop an efficient spectral feature selection algorithm to minimize the redundancy within the selected feature subset through $\mathcal{L}_{2,1}$ norm. Recently, researchers pay attention to unsupervised feature selection with global minimized redundancy[Wang and Wei(2015)][Wang and Huang(2015)]. Several graph based approaches are proposed in [Mairal and Yu(2013)], [Song and Wang(2013)]. The most closed research work to us is [Liu and Zhang(2012)] which build a sparse graph at feature side and ranking features by approximation errors.

8 Conclusion

In this study, we propose sparse feature graph to model both one-to-one feature redundancy and one-to-many features redundancy. By separate whole features into different redundancy feature group through local compressible subgraphs, we reduce the dimensionality of data by only select one representative feature from each group. One advantage of our algorithm is that it does not need to calculate the pairwise distance which is not accurate for high dimensional datasets. The experiment results shows that our algorithm is an effective and efficient way to obtain accurate data structure information

#f	1024	913	620	535	469	327	160	104	58	33
10	0.63	0.51	0.60	0.56	0.53	0.62	0.61	**0.65**	0.60	0.62
15	0.66	0.56	0.63	0.60	0.58	**0.67**	0.62	0.60	0.63	0.58
20	0.67	0.59	0.65	0.64	0.59	0.64	0.63	0.61	0.64	0.56
25	0.67	0.59	0.66	0.64	0.63	0.65	0.66	0.64	0.65	0.58
30	0.68	0.63	0.66	0.65	0.66	0.67	0.65	0.67	0.65	0.59
35	0.69	0.64	**0.70**	0.66	0.65	0.67	0.67	0.68	0.65	-
40	0.70	0.67	**0.71**	0.68	0.67	0.68	**0.70**	**0.70**	0.66	-
45	0.70	0.69	**0.70**	0.69	0.66	0.69	**0.70**	0.69	0.65	-
50	0.73	0.71	0.72	0.68	0.66	0.70	0.72	0.69	0.66	-
55	0.71	**0.74**	0.70	0.68	0.67	0.71	0.71	0.71	0.66	-
60	0.71	**0.74**	**0.71**	**0.72**	**0.71**	0.69	**0.72**	0.71	-	-

Table 2 NMI results of "ORL" dataset

#f	1024	1023	964	654	525	427	271	152	83	34
10	0.48	0.43	0.43	0.45	0.42	0.46	0.45	0.46	0.47	0.44
15	0.49	0.47	0.46	**0.51**	**0.49**	0.48	0.45	0.47	**0.50**	0.43
20	0.49	0.48	0.46	**0.55**	0.48	**0.51**	0.47	0.47	**0.51**	0.41
25	0.51	0.49	0.49	**0.52**	**0.52**	**0.52**	0.45	0.49	**0.54**	0.41
30	0.51	**0.51**	0.49	**0.54**	0.50	**0.51**	**0.51**	0.49	**0.50**	0.39
35	0.53	0.49	0.50	**0.54**	**0.53**	0.52	0.52	0.48	0.50	-
40	0.49	**0.50**	**0.51**	**0.53**	**0.58**	**0.55**	**0.55**	0.48	**0.51**	-
45	0.48	**0.51**	**0.51**	**0.56**	**0.59**	**0.57**	**0.52**	**0.52**	**0.49**	-
50	0.52	0.50	0.47	**0.53**	**0.59**	**0.53**	**0.53**	**0.56**	0.49	-
55	0.54	0.51	0.52	**0.55**	0.50	0.51	0.51	0.51	0.49	-
60	0.54	0.49	0.51	0.49	**0.54**	0.50	0.51	0.46	0.52	-

Table 4 NMI results of "Yale" dataset

#f	2420	2409	1871	793	698	662	654	630	566	324
10	0.44	**0.48**	**0.55**	**0.53**	**0.58**	**0.56**	**0.54**	**0.61**	**0.50**	0.38
15	0.44	**0.61**	**0.57**	**0.50**	**0.58**	**0.58**	**0.55**	**0.59**	**0.53**	0.39
20	0.43	**0.56**	**0.61**	**0.59**	**0.60**	**0.56**	**0.62**	**0.59**	**0.56**	0.41
25	0.52	**0.61**	**0.61**	**0.64**	**0.61**	**0.60**	**0.58**	**0.58**	**0.54**	**0.43**
30	0.53	**0.61**	**0.62**	**0.57**	**0.62**	**0.62**	**0.60**	**0.53**	**0.63**	0.41
35	0.59	**0.60**	**0.59**	**0.60**	**0.63**	**0.61**	**0.60**	**0.62**	**0.64**	0.43
40	0.53	**0.60**	**0.58**	**0.57**	**0.66**	**0.62**	**0.59**	**0.62**	**0.69**	0.42
45	0.55	**0.61**	**0.61**	**0.62**	**0.60**	**0.64**	0.60	**0.64**	**0.65**	0.43
50	0.56	**0.63**	**0.62**	**0.68**	**0.64**	**0.62**	**0.58**	**0.63**	**0.66**	0.37
55	0.61	0.60	**0.62**	**0.69**	**0.62**	0.60	0.57	**0.65**	0.58	0.39
60	0.55	**0.64**	**0.63**	**0.64**	**0.60**	**0.63**	0.54	**0.63**	0.51	0.39

Table 6 NMI results of "PIE10P" dataset

#f	10304	10302	8503	3803	3408	3244	3030	2822	2638	2175
10	0.65	**0.78**	**0.77**	**0.76**	**0.77**	**0.80**	**0.74**	**0.72**	**0.75**	**0.73**
15	0.72	**0.82**	**0.79**	**0.78**	**0.81**	**0.83**	**0.79**	**0.81**	**0.75**	**0.79**
20	0.76	**0.81**	0.74	**0.78**	**0.84**	**0.83**	**0.81**	**0.76**	**0.80**	**0.78**
25	0.79	**0.84**	0.74	0.73	**0.82**	**0.86**	**0.88**	**0.83**	**0.86**	**0.81**
30	0.75	**0.77**	**0.82**	0.74	**0.88**	**0.82**	**0.83**	**0.83**	**0.86**	**0.86**
35	0.81	**0.81**	0.80	**0.83**	**0.85**	**0.83**	0.80	**0.82**	**0.85**	**0.85**
40	0.83	**0.88**	**0.84**	**0.84**	**0.90**	**0.86**	0.81	**0.93**	**0.84**	**0.87**
45	0.84	0.93	0.83	**0.85**	**0.91**	**0.86**	0.83	**0.88**	**0.84**	**0.86**
50	0.78	**0.88**	**0.88**	**0.87**	**0.89**	**0.86**	**0.82**	**0.90**	**0.84**	**0.83**
55	0.84	**0.89**	**0.86**	**0.89**	**0.91**	**0.89**	**0.88**	**0.86**	**0.84**	**0.86**
60	0.85	**0.88**	**0.86**	0.84	**0.85**	**0.91**	**0.85**	**0.88**	**0.86**	**0.85**

Table 8 NMI results of "ORL10P" dataset

#f	1024	913	620	535	469	327	160	104	58	33
10	0.38	0.28	0.36	0.31	0.28	**0.39**	**0.39**	**0.46**	**0.39**	**0.41**
15	0.45	0.33	0.41	0.40	0.34	0.43	0.40	0.38	0.42	0.36
20	0.47	0.34	0.43	0.43	0.35	0.43	0.41	0.39	0.43	0.32
25	0.48	0.35	0.45	0.44	0.37	0.42	0.47	0.41	0.45	0.34
30	0.47	0.40	0.42	0.42	0.43	0.47	0.43	0.45	0.42	0.35
35	0.49	0.41	0.48	0.46	0.44	0.44	0.47	0.47	0.42	-
40	0.51	0.46	**0.53**	0.48	0.46	0.45	0.48	**0.51**	0.43	-
45	0.49	0.47	**0.51**	**0.51**	0.44	0.48	**0.49**	**0.49**	0.43	-
50	0.55	0.51	0.52	0.47	0.47	0.50	0.52	0.48	0.46	-
55	0.53	**0.53**	0.51	0.46	0.45	0.48	0.50	**0.53**	0.46	-
60	0.51	**0.55**	**0.52**	**0.54**	0.51	0.47	**0.54**	0.51	-	-

Table 3 ACC results of "ORL" dataset.

#f	1024	1023	964	654	525	427	271	152	83	34
10	0.39	0.36	0.37	0.36	0.33	0.38	**0.41**	**0.40**	**0.41**	0.36
15	0.43	0.41	0.42	**0.44**	0.41	0.41	0.39	0.41	**0.46**	0.39
20	0.44	0.42	0.41	**0.48**	**0.44**	**0.44**	0.43	0.42	**0.44**	0.35
25	0.45	**0.45**	0.44	**0.46**	**0.47**	**0.45**	0.41	0.43	**0.49**	0.33
30	0.48	0.44	0.42	0.47	0.47	0.45	0.45	0.40	0.47	0.33
35	0.48	**0.48**	0.44	**0.50**	0.47	0.46	0.47	0.41	0.44	-
40	0.42	**0.44**	**0.45**	**0.50**	**0.55**	**0.48**	**0.53**	0.41	**0.44**	-
45	0.41	**0.48**	**0.46**	**0.51**	**0.53**	**0.54**	**0.49**	**0.47**	**0.42**	-
50	0.46	0.41	0.42	**0.48**	**0.56**	**0.50**	**0.46**	**0.52**	0.41	-
55	0.48	0.44	**0.48**	**0.48**	0.43	0.45	**0.49**	0.47	0.42	-
60	0.50	0.42	0.44	0.40	**0.50**	0.41	0.46	0.42	0.43	-

Table 5 ACC results of "Yale" dataset.

#f	2420	2409	1871	793	698	662	654	630	566	324
10	0.39	**0.45**	**0.48**	**0.50**	**0.56**	**0.50**	**0.53**	**0.59**	**0.46**	0.39
15	0.39	**0.58**	**0.51**	**0.49**	**0.51**	**0.55**	**0.56**	**0.60**	**0.50**	**0.41**
20	0.36	**0.51**	**0.53**	**0.53**	**0.55**	**0.56**	**0.60**	**0.54**	**0.50**	**0.38**
25	0.45	**0.59**	**0.53**	**0.60**	**0.54**	**0.59**	**0.60**	**0.56**	**0.52**	0.40
30	0.50	**0.58**	**0.56**	**0.58**	**0.59**	**0.60**	**0.59**	**0.49**	**0.60**	0.40
35	0.48	**0.57**	**0.51**	**0.59**	**0.61**	**0.53**	**0.54**	**0.62**	**0.61**	**0.37**
40	0.42	**0.52**	**0.53**	**0.56**	**0.63**	**0.59**	**0.53**	**0.60**	**0.64**	0.38
45	0.44	**0.52**	**0.52**	**0.58**	**0.51**	**0.63**	**0.54**	**0.62**	**0.60**	**0.41**
50	0.44	**0.61**	**0.52**	**0.64**	**0.60**	**0.59**	**0.55**	**0.62**	**0.61**	0.37
55	0.46	**0.54**	**0.53**	**0.67**	**0.58**	**0.57**	**0.57**	**0.63**	**0.54**	0.37
60	0.49	**0.60**	**0.61**	**0.61**	**0.57**	**0.61**	**0.51**	**0.61**	0.46	0.35

Table 7 ACC results of "PIE10P" dataset.

#f	10304	10302	8503	3803	3408	3244	3030	2822	2638	2175
10	0.66	**0.74**	**0.81**	**0.75**	**0.75**	**0.69**	**0.72**	**0.70**	**0.69**	**0.67**
15	0.69	**0.85**	**0.76**	**0.78**	**0.78**	**0.86**	**0.80**	**0.75**	**0.73**	**0.75**
20	0.77	**0.84**	0.74	0.76	**0.80**	**0.80**	**0.78**	0.69	0.75	0.74
25	0.71	**0.79**	0.68	**0.74**	**0.78**	**0.86**	**0.84**	**0.82**	**0.82**	**0.74**
30	0.71	**0.71**	**0.77**	0.68	**0.86**	**0.77**	**0.81**	**0.77**	**0.82**	**0.81**
35	0.74	**0.74**	**0.74**	**0.76**	**0.81**	**0.77**	0.73	**0.76**	**0.82**	**0.78**
40	0.80	**0.85**	0.74	0.77	**0.87**	**0.80**	0.75	**0.89**	**0.80**	**0.83**
45	0.82	**0.89**	0.73	0.81	**0.88**	0.78	0.77	**0.86**	0.80	0.79
50	0.73	**0.80**	**0.80**	**0.74**	**0.86**	**0.79**	**0.74**	**0.88**	**0.81**	**0.77**
55	0.79	**0.85**	**0.82**	**0.86**	**0.89**	**0.87**	**0.80**	**0.82**	**0.81**	**0.79**
60	0.82	**0.84**	0.77	0.75	**0.82**	**0.89**	0.77	**0.84**	**0.82**	**0.82**

Table 9 ACC results of "ORL10P" dataset.

which is demanding for unsupervised feature selection algorithms.

References

[Aggarwal and Keim(2001)] Aggarwal HA Charu C, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. In: International Conference on Database Theory, Springer, pp 420–434

[Beyer and Shaft(1999)] Beyer GJRR Kevin, Shaft U (1999) When is nearest neighbor meaningful? In: International conference on database theory, Springer, pp 217–235

[Cai and He(2010)] Cai ZC Deng, He X (2010) Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 333–342

[Dash and Liu(1997)] Dash M, Liu H (1997) Feature selection for classification. Intelligent data analysis 1(3):131–156

[Du and Shen(2015)] Du L, Shen YD (2015) Unsupervised feature selection with adaptive structure learning. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp 209–218

[Dy and Brodley(2004)] Dy JG, Brodley CE (2004) Feature selection for unsupervised learning. The Journal of Machine Learning Research 5:845–889

[Elhamifar and Vidal(2013)] Elhamifar E, Vidal R (2013) Sparse subspace clustering: Algorithm, theory, and applications. Pattern Analysis and Machine Intelligence, IEEE Transactions on 35(11):2765–2781

[He and Bao(2011)] He JMZC Xiaofei, Bao H (2011) A variance minimization criterion to feature selection using laplacian regularization. Pattern Analysis and Machine

#f	4026	4009	3978	3899	3737	3456	2671	1203	334	136
10	0.51	**0.59**	**0.58**	**0.52**	0.50	0.50	**0.51**	0.50	0.50	0.49
15	0.55	**0.60**	**0.62**	**0.56**	**0.58**	**0.58**	**0.58**	**0.56**	0.47	0.52
20	0.60	**0.61**	**0.60**	0.57	**0.62**	**0.62**	**0.64**	0.58	0.58	**0.60**
25	0.63	0.59	**0.64**	0.60	**0.63**	0.58	**0.66**	0.57	0.56	0.53
30	0.59	**0.61**	**0.62**	**0.60**	**0.62**	**0.64**	**0.65**	**0.60**	**0.60**	**0.59**
35	0.61	**0.66**	**0.62**	0.60	**0.65**	**0.62**	**0.61**	**0.62**	0.56	0.53
40	0.64	0.60	**0.66**	0.63	0.61	0.63	**0.66**	0.61	0.58	0.55
45	0.58	**0.63**	**0.62**	**0.62**	**0.58**	**0.61**	**0.63**	**0.64**	**0.60**	0.57
50	0.65	0.60	0.61	0.61	0.56	0.63	0.61	0.63	0.58	0.54
55	0.63	0.60	0.61	0.62	0.60	0.60	**0.63**	0.60	0.58	0.58
60	0.60	**0.60**	**0.63**	**0.61**	**0.63**	0.59	**0.65**	0.59	0.57	0.57

Table 10 NMI results of "Lymphoma" dataset

#f	3312	3311	3309	3236	1844	559	384	344	305	183
10	0.42	**0.42**	**0.43**	**0.49**	**0.52**	**0.53**	**0.43**	**0.46**	**0.43**	0.25
15	0.54	**0.54**	0.53	0.51	0.51	0.51	0.45	0.52	0.38	0.21
20	0.51	**0.51**	**0.52**	**0.53**	0.41	0.49	0.36	**0.52**	0.38	0.20
25	0.51	**0.51**	**0.53**	0.48	0.42	**0.52**	0.40	0.48	0.35	0.26
30	0.47	**0.48**	**0.52**	**0.49**	0.41	0.37	**0.49**	**0.48**	0.41	0.24
35	0.46	0.38	**0.46**	**0.48**	0.39	**0.52**	**0.49**	0.38	0.35	0.27
40	0.49	**0.49**	**0.50**	0.46	0.43	0.40	0.38	0.35	0.40	0.29
45	0.36	**0.42**	0.33	**0.47**	**0.40**	0.33	**0.38**	0.35	0.35	0.31
50	0.45	**0.45**	**0.47**	**0.49**	**0.52**	0.32	0.40	0.36	0.35	0.31
55	0.44	**0.44**	**0.44**	**0.49**	**0.51**	0.33	**0.49**	0.31	0.30	0.31
60	0.47	0.46	0.45	**0.51**	**0.49**	0.33	0.39	0.32	0.31	0.35

Table 12 NMI results of "LUNG" dataset

#f	9182	9180	9179	9150	7736	3072	697	449	360	144
10	0.70	**0.70**	**0.70**	0.69	0.67	0.64	0.66	0.65	0.66	0.47
15	0.71	0.70	**0.73**	**0.73**	**0.74**	0.66	0.67	0.70	0.66	0.52
20	0.77	**0.78**	**0.77**	0.72	0.75	0.72	0.73	0.71	0.73	0.54
25	0.74	**0.77**	**0.77**	**0.75**	**0.74**	0.71	**0.79**	**0.75**	**0.74**	0.53
30	0.69	**0.71**	**0.72**	**0.70**	**0.74**	**0.75**	**0.77**	**0.79**	**0.73**	0.54
35	0.77	0.76	0.76	0.76	0.74	**0.77**	**0.78**	**0.78**	**0.78**	0.60
40	0.75	0.74	**0.76**	**0.77**	0.74	**0.79**	**0.76**	**0.78**	**0.75**	0.59
45	0.77	0.76	0.74	**0.78**	0.74	**0.82**	**0.78**	**0.80**	**0.79**	0.57
50	0.79	0.76	0.75	0.75	**0.79**	0.76	**0.79**	**0.84**	**0.83**	0.58
55	0.75	**0.76**	**0.76**	0.74	**0.75**	**0.79**	**0.79**	**0.83**	**0.83**	0.59
60	0.74	0.72	**0.76**	0.73	**0.76**	**0.82**	**0.84**	**0.82**	**0.78**	0.62

Table 14 NMI results of "Carcinom" dataset

#f	11340	11335	11301	10573	8238	7053	6697	6533	6180	4396
10	0.16	**0.16**	0.15	**0.26**	**0.18**	**0.22**	**0.20**	**0.20**	**0.20**	**0.21**
15	0.14	**0.14**	**0.15**	**0.26**	**0.18**	**0.28**	0.09	**0.24**	0.07	0.06
20	0.16	**0.16**	0.15	0.08	0.14	**0.21**	0.04	**0.31**	**0.16**	0.11
25	0.14	**0.14**	**0.15**	0.09	0.08	**0.22**	**0.23**	0.10	0.09	0.11
30	0.13	**0.13**	**0.13**	0.08	0.07	**0.18**	0.03	**0.14**	0.10	0.11
35	0.17	**0.17**	0.13	0.03	0.07	0.12	0.10	0.01	0.08	0.10
40	0.14	**0.14**	**0.14**	0.07	0.08	0.13	0.12	0.05	**0.14**	0.09
45	0.09	**0.09**	**0.18**	0.08	**0.11**	**0.10**	**0.13**	0.07	**0.12**	**0.09**
50	0.15	0.14	**0.15**	0.08	0.11	0.11	0.12	0.12	0.13	0.09
55	0.15	**0.15**	0.14	**0.21**	0.08	0.13	0.13	0.12	0.13	0.07
60	0.10	**0.10**	**0.14**	**0.15**	0.08	**0.10**	**0.12**	**0.12**	**0.14**	0.07

Table 16 NMI results of "CLL-SUB-111" dataset

#f	4026	4009	3978	3899	3737	3456	2671	1203	334	136
10	0.50	**0.57**	**0.56**	**0.53**	0.49	**0.51**	**0.51**	0.48	**0.50**	**0.50**
15	0.53	**0.62**	**0.59**	**0.58**	**0.56**	**0.59**	**0.58**	**0.55**	0.50	**0.53**
20	0.59	0.56	0.55	0.56	0.56	**0.59**	**0.59**	0.54	0.55	**0.59**
25	0.60	0.57	**0.62**	0.56	**0.62**	0.58	**0.64**	0.56	0.52	0.50
30	0.56	**0.60**	**0.58**	**0.58**	**0.59**	**0.61**	**0.65**	**0.59**	**0.57**	0.55
35	0.55	**0.62**	**0.59**	**0.58**	**0.61**	**0.60**	**0.57**	**0.59**	**0.55**	0.53
40	0.66	0.57	0.61	0.61	0.61	0.59	0.60	0.58	0.59	0.54
45	0.54	**0.60**	**0.60**	**0.58**	**0.55**	**0.60**	**0.62**	**0.59**	**0.56**	**0.54**
50	0.65	0.62	0.58	0.64	0.52	0.59	0.56	0.59	0.53	0.53
55	0.57	**0.60**	**0.65**	**0.60**	0.54	**0.57**	**0.65**	**0.59**	0.54	**0.59**
60	0.56	**0.58**	**0.64**	**0.58**	**0.61**	**0.57**	**0.67**	**0.56**	0.53	**0.57**

Table 11 ACC results of "Lymphoma" dataset.

#f	3312	3311	3309	3236	1844	559	384	344	305	183
10	0.71	**0.72**	**0.73**	**0.77**	**0.77**	**0.75**	0.68	0.65	0.66	0.56
15	0.81	**0.81**	0.79	0.72	0.73	0.72	0.67	0.65	0.58	0.48
20	0.71	**0.73**	**0.74**	**0.72**	0.69	0.69	0.61	0.60	0.58	0.39
25	0.71	**0.71**	**0.74**	0.67	0.69	0.68	0.59	0.61	0.56	0.49
30	0.66	**0.66**	**0.67**	**0.71**	**0.68**	0.56	0.59	0.59	0.61	0.43
35	0.64	0.60	0.63	**0.68**	**0.66**	0.60	0.58	0.56	0.53	0.49
40	0.65	**0.65**	**0.66**	**0.65**	0.64	0.57	0.54	0.54	0.56	0.46
45	0.60	**0.63**	0.57	**0.65**	**0.61**	0.52	0.54	0.52	0.52	0.49
50	0.65	**0.65**	0.63	**0.65**	**0.65**	0.48	0.57	0.53	0.53	0.52
55	0.61	**0.61**	0.59	**0.65**	**0.62**	0.48	0.59	0.48	0.49	0.49
60	0.64	0.63	0.63	**0.64**	0.62	0.51	0.55	0.49	0.48	0.51

Table 13 ACC results of "LUNG" dataset.

#f	9182	9180	9179	9150	7736	3072	697	449	360	144
10	0.63	**0.66**	0.62	0.61	**0.67**	0.60	0.60	0.59	**0.64**	0.48
15	0.67	0.57	**0.70**	0.66	**0.68**	0.63	0.57	**0.67**	0.64	0.53
20	0.70	0.68	**0.74**	0.66	**0.71**	**0.71**	0.64	**0.73**	**0.74**	0.56
25	0.70	**0.72**	**0.75**	0.69	**0.75**	0.64	**0.75**	**0.72**	**0.76**	0.51
30	0.61	**0.64**	**0.70**	**0.69**	**0.67**	**0.71**	**0.74**	**0.76**	**0.71**	0.52
35	0.76	0.74	0.74	0.74	0.70	0.75	0.70	**0.76**	**0.77**	0.57
40	0.72	**0.72**	**0.73**	**0.75**	0.69	**0.76**	0.66	**0.78**	0.71	0.56
45	0.75	0.74	0.70	**0.75**	0.74	**0.79**	0.72	**0.79**	**0.76**	0.55
50	0.74	**0.74**	0.70	0.72	**0.74**	0.66	**0.74**	**0.83**	**0.79**	0.56
55	0.73	**0.74**	**0.74**	0.72	0.71	0.72	0.72	**0.82**	**0.80**	0.56
60	0.70	0.61	**0.71**	0.66	**0.72**	**0.75**	**0.82**	**0.80**	**0.77**	0.55

Table 15 ACC results of "Carcinom" dataset.

#f	11340	11335	11301	10573	8238	7053	6697	6533	6180	4396
10	0.51	**0.51**	0.50	**0.54**	**0.59**	**0.57**	**0.58**	**0.55**	**0.51**	0.50
15	0.51	**0.51**	0.50	**0.57**	**0.55**	**0.62**	0.47	**0.59**	0.45	0.43
20	0.50	**0.50**	0.48	0.46	**0.50**	**0.54**	0.40	**0.59**	**0.54**	**0.50**
25	0.48	**0.48**	**0.51**	0.44	0.46	**0.54**	**0.57**	**0.50**	0.46	**0.50**
30	0.49	**0.49**	**0.49**	0.44	0.44	**0.53**	0.42	**0.51**	0.48	0.48
35	0.51	**0.51**	0.49	0.42	0.44	0.49	0.49	0.41	0.44	0.48
40	0.51	**0.51**	0.50	0.43	0.45	0.50	0.49	0.43	0.48	0.47
45	0.46	0.45	**0.52**	0.44	**0.46**	**0.47**	**0.51**	0.45	**0.47**	**0.47**
50	0.51	0.50	**0.51**	0.45	0.46	0.50	0.49	0.49	0.49	0.48
55	0.49	**0.49**	**0.50**	**0.54**	0.46	**0.50**	**0.50**	**0.49**	**0.49**	0.45
60	0.49	**0.49**	**0.50**	**0.53**	0.43	0.48	**0.49**	**0.49**	**0.50**	0.44

Table 17 ACC results of "CLL-SUB-111" dataset.

Intelligence, IEEE Transactions on 33(10):2013–2025

[Hou and Wu(2014)] Hou NFLXYD Chenping, Wu Y (2014) Joint embedding learning and sparse regression: A framework for unsupervised feature selection. Cybernetics, IEEE Transactions on 44(6):793–804

[J. Li and Liu(2016)] J Li SWFMRTJT K Cheng, Liu H (2016) Feature selection: A data perspective URL http://featureselection.asu.edu/

[Koller(1996)] Koller D (1996) Toward optimal feature selection. In: Proc. 13th International Conference on Machine Learning, Morgan Kaufmann, pp 284–292

[Li and Lu(2012)] Li YYLJZX Zechao, Lu H (2012) Unsupervised feature selection using nonnegative spectral analysis. In: AAAI

[Liu and Zhang(2012)] Liu SD Mingxia, Zhang D (2012) Sparsity score: A new filter feature selection method based on graph. In: Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE, pp 959–962

[Liu and Liu(2014)] Liu WLZJYJ Xinwang, Liu H (2014) Global and local structure preservation for feature selection. Neural Networks and Learning Systems, IEEE Transactions on 25(6):1083–1095

[Mairal and Yu(2013)] Mairal J, Yu B (2013) Supervised feature selection in graphs with path coding penalties and network flows. The Journal of Machine Learning Research 14(1):2449–2485

[Ng et al(2001)] Ng JMIWY Andrew Y, et al (2001) On spectral clustering: Analysis and an algorithm. In: NIPS, vol 14, pp 849–856

[Peng and Ding(2005)] Peng LF Hanchuan, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. Pattern Analysis and Machine Intelligence, IEEE Transactions on 27(8):1226–1238

[Robnik-Šikonja and Kononenko(2003)] Robnik-Šikonja M, Kononenko I (2003) Theoretical and empirical analysis

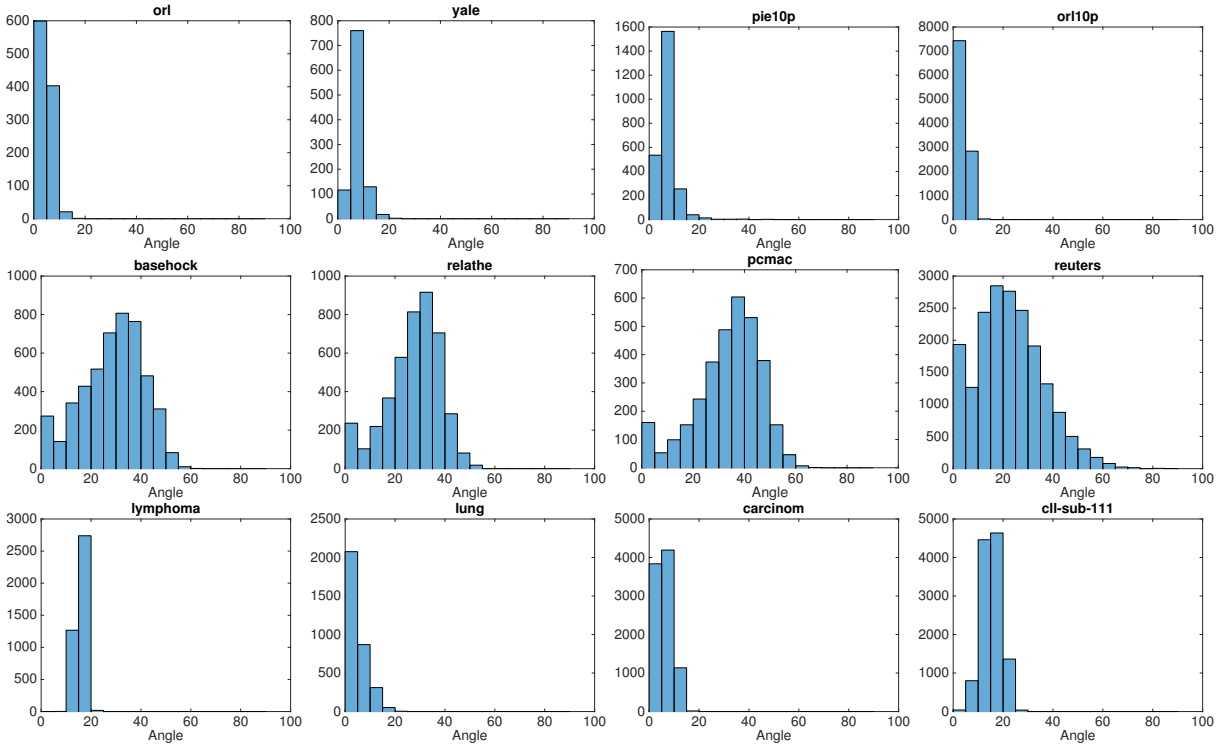

Fig. 9 The distribution of angle between original feature vector and its sparse representation.

of relieff and rrelieff. Machine learning 53(1-2):23–69

[Song and Wang(2013)] Song NJ Qinbao, Wang G (2013) A fast clustering-based feature subset selection algorithm for high-dimensional data. Knowledge and Data Engineering, IEEE Transactions on 25(1):1–14

[Tibshirani(1996)] Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological) pp 267–288

[Wang and Wei(2015)] Wang MA Xuerui, Wei X (2015) Feature selection with integrated relevance and redundancy optimization. In: Data Mining, 2015. ICDM 2015. Fifteenth IEEE International Conference on, IEEE, pp 697–702

[Wang and Huang(2015)] Wang NF De, Huang H (2015) Feature selection via global redundancy minimization. Knowledge and Data Engineering, IEEE Transactions on 27(10):2743–2755

[Weber and Blott(1998)] Weber SHJ Roger, Blott S (1998) A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: VLDB, vol 98, pp 194–205

[Yang and Zhou(2011)] Yang SHTMZHZ Yi, Zhou X (2011) l2, 1-norm regularized discriminative feature selection for unsupervised learning. In: IJCAI Proceedings-International Joint Conference on Artificial Intelligence, Citeseer, vol 22, p 1589

[You and Vidal(2016)] You RD Chong, Vidal R (2016) Scalable sparse subspace clustering by orthogonal matching pursuit. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3918–3927

[Yu and Liu(2004)] Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. The Journal of Machine Learning Research 5:1205–1224

[Zhao and Liu(2010)] Zhao WL Zheng, Liu H (2010) Efficient spectral feature selection with minimum redundancy. In: AAAI

[Zhao and Ye(2013)] Zhao WLLH Zheng, Ye J (2013) On similarity preserving feature selection. Knowledge and Data Engineering, IEEE Transactions on 25(3):619–632