Multimed Tools Appl (2018) 77:22099–22113
https://doi.org/10.1007/s11042-017-5582-0

Fast unsupervised feature selection with anchor graph and $\ell_{2,1}$-norm regularization

Haojie Hu[1] · Rong Wang[1,2] · Feiping Nie[2] · Xiaojun Yang[3] · Weizhong Yu[4]

Received: 19 September 2017 / Revised: 29 November 2017 / Accepted: 26 December 2017 / Published online: 18 January 2018

Abstract Graph-based unsupervised feature selection has been proven to be effective in dealing with unlabeled and high-dimensional data. However, most existing methods face a number of challenges primarily due to their high computational complexity. In light of the ever-increasing size of data, these approaches tend to be inefficient in dealing with large-scale data sets. We propose a novel approach, called Fast Unsupervised Feature Selection (FUFS), to efficiently tackle this problem. Firstly, an anchor graph is constructed by means of a parameter-free adaptive neighbor assignment strategy. Meanwhile, an approximate nearest neighbor search technique is introduced to speed up the anchor graph construction. The $\ell_{2,1}$-norm regularization is then performed to select more valuable features. Experiments on several large-scale data sets demonstrate the effectiveness and efficiency of the proposed method.

Keywords Unsupervised feature selection · Anchor graph · $\ell_{2,1}$-norm

✉ Rong Wang
wangrong07@tsinghua.org.cn

[1] The Xi'an Research Institute of Hi-Tech, Xi'an 710025, China

[2] The Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China

[3] The School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China

[4] The School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China

1 Introduction

High-dimensional data are commonly generated in a range of real-world applications, including computer vision, pattern recognition, data mining and machine learning. However, the use of high-dimensional data can not only increase storage costs, but also introduce redundancy and irrelevant information [9, 32, 36], that degenerates the performance of learning tasks. Accordingly, feature selection, which aims to select the most representative feature set from the original features is preferred in these instances [6, 17, 20, 23, 29, 31]. Since practical large-scale data are usually collected without labels being appended, and annotating these data is a dramatically expensive and time-consuming process [2, 11], unsupervised feature selection has become a ubiquitous and challenging problem [5]. In recent years, the development of various unsupervised feature selection methods has significantly facilitated the performance of many machine learning tasks, such as classification, clustering, retrieval, and ranking [7, 8, 13, 14, 16].

In this letter, we focus on the family of graph-based unsupervised feature selection (GUFS) methods, in which the manifold geometry structure of the whole feature set is characterized in graph form. A range of typical GUFS methods have been proposed over the past decade, including Laplacian Score (LS) [9], Spectral Feature Selection (SPEC) [37], Multi-Cluster Feature Selection (MCFS) [1], and Robust Unsupervised Feature Selection (RUFS) [28]. However, most traditional GUFS methods focus on learning task performance while neglecting the underlying computational complexity, which is of great importance given the ever-increasing size of data. This complexity arises primarily due to two aspects: the first is the graph construction, and the second is the feature selection on the graph. Both of these processes are time-consuming for large-scale data, and have a time complexity of at least $O(n^2d)$, where n and d denote the number of samples and features, respectively.

To address this issue, and inspired by recent works that have scaled up graph-based learning models using anchors [3, 4, 18, 19], we propose a novel approach named Fast Unsupervised Feature Selection (FUFS) incorporating an anchor graph and $\ell_{2,1}$-norm regularization. The main contributions of our work are as follows. First, the anchor graph is constructed using a parameter-free adaptive neighbor assignment strategy. Meanwhile, an approximate nearest neighbor search (ANNS) technique is introduced to speed up the construction of the anchor graph. Second, the $\ell_{2,1}$-norm regularization [22, 26, 34] is performed in order to select more valuable features, and a simple yet efficient iterative algorithm is designed to optimize the proposed objective function. Thirdly, the computational complexity of the FUFS algorithm can be reduced to $O(ndmt)$, where m and t are the number of anchors and iteration, respectively, giving our proposed method a great advantage over conventional GUFS methods. Comprehensive experiments on several large-scale data sets demonstrate the efficiency and effectiveness of the proposed FUFS algorithm.

2 The FUFS algorithm

In this section, we introduce our FUFS method for large-scale data sets. First, the anchor graph is constructed, after which the $\ell_{2,1}$-norm regularization is adopted to select more valuable features on this graph.

2.1 Graph construction

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]^T \in \mathbb{R}^{n \times d}$ represent the data matrix, where n and d denote the number of data points and the dimension of features, respectively. Each data point $\boldsymbol{x}_i$ is represented as a vertex on the affinity graph, while each edge represents the similarity relationship of one pair of vertexes. The weight of the edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as a_{ij} and $\boldsymbol{A} = \{a_{ij}\} \in \mathbb{R}^{n \times n}$ denotes the similarity matrix of the affinity graph.

2.1.1 Traditional graph construction

The first step of all the traditional GUFS methods is to construct the similarity graph by computing all pairwise similarity between the data points. There are usually three different similarity graphs:

1. **The ε-neighborhood graph**: We connect all points whose pairwise distances are smaller than ε.
2. **k-nearest neighbor graph**: The vertexes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected if $\boldsymbol{x}_i$ is among the k-nearest neighbors of $\boldsymbol{x}_j$ or $\boldsymbol{x}_j$ is among the k-nearest neighbors of $\boldsymbol{x}_i$.
3. **The fully connected graph**: All points are connected with positive similarity with each other.

We can apply all graphs mentioned above to weight the edges by the similarity, while the choice of different graphs may result in different learning performance. Since local geometric structure of data can usually get better performance than global geometric structure, and the value of positive integer k is easier to tune than ε, almost all the graph-based methods tend to apply k-nearest neighbor graph (KNN) to construct similarity graph. A reasonable way to define the weight a_{ij} is by using the Gaussian kernel function, then we can define

$$a_{ij} = \begin{cases} \mathrm{eps}(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}) & \boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_j) \text{or } \boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i) \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathcal{N}(\boldsymbol{x})$ denotes the set of k-nearest neighbors of $\boldsymbol{x}$, σ is a parameter that controls the width of the neighborhoods. Since the extra parameter σ in the Gaussian kernel function is very sensitive [24, 33] and is difficult to tune in practice, we are more likely to adopt a parameter-free method to construct similarity graph.

2.1.2 Anchor graph construction

Recent studies have adopted an anchor-based strategy to construct the similarity matrix $\boldsymbol{A}$. Generally, this strategy requires two steps to construct $\boldsymbol{A}$: first, m ($m \ll n$) anchors are generated from data points, after which the similarity between data points and anchors are measured by the matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times m}$.

Anchor generation can be achieved either by random selection or by using the k-means method [3, 4, 18, 19]. Random selection selects m anchors by random sampling from data points and takes $O(1)$ computational complexity. Although random selection cannot guarantee that the selected m anchors are always good, it is extremely fast for large-scale data sets. The k-means method makes use of m clustering centers as anchors. Although use of these clustering centers results in more representative anchors, the k-means method needs

$O(ndmt)$ computational complexity , where t is the number of iterations, which makes its use for large-scale data sets impossible.

After the anchors are generated, the similarity matrix $\boldsymbol{Z}$ needs to be constructed. Conventional methods usually use kernel-based neighbor assignment strategy (e.g., Gaussian similarity function), which typically requires extra parameters. To avoid this, we adopt an adaptive neighbor assignment strategy [24] to obtain the the similarity matrix $\boldsymbol{Z}$. Let $U = [\boldsymbol{u}_1, ..., \boldsymbol{u}_m]^T \in \mathbb{R}^{m\times d}$ denotes the set of anchor points. The similarity z_{ij} between $\boldsymbol{x}_i$ and $\boldsymbol{u}_j$ can be defined as probability that $\boldsymbol{u}_i$ is to be the neighbor of $\boldsymbol{x}_i$. We use the square of Euclidean distance $\|\boldsymbol{x}_i - \boldsymbol{u}_j\|_2^2$ as the distance measure. For the i-th data point $\boldsymbol{x}_i$, all the anchor points can be connected to $\boldsymbol{x}_i$ with probability z_{ij}. Evoked by the intuition that nearby points should have similar properties [10, 25], a smaller distance should be assigned a larger probability. Thereby, a natural method to obtain neighbor probabilities for the i-th sample is by solving following problem

$$\min_{\boldsymbol{z}_i^T\mathbf{1}=1, z_i\geq 0} \sum_{j=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{u}_j\|_2^2 z_{ij}. \tag{1}$$

However, Eq. (1) has a trivial solution, only the nearest anchor point can be the neighbor of $\boldsymbol{x}_i$. To avoid this dilemma, a regularization term is added to Eq. (1), then we have

$$\min_{\boldsymbol{z}_i^T\mathbf{1}=1, z_{ij}\geq 0} \sum_{j=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{u}_j\|_2^2 z_{ij} + \gamma \sum_{j=1}^{m} z_{ij}^2, \tag{2}$$

where $\boldsymbol{z}_i^T$ denotes the i-th row of $\boldsymbol{Z}$, z_{ij} is the j-th element of $\boldsymbol{z}_i^T$ and γ is the regularization parameter. Let $d_{ij} = \|\boldsymbol{x}_i - \boldsymbol{u}_j\|_2^2$, while $\boldsymbol{d}_i \in \mathbb{R}^{m\times 1}$ is a vector with the j-th element as d_{ij}, Eq. (2) can be rewritten in vector form as

$$\min_{z_i} \|\boldsymbol{z}_i + \frac{\boldsymbol{d}_i}{2\gamma}\|_2^2 \quad s.t.\ \boldsymbol{z}_i^T\mathbf{1} = 1, z_{ij} \geq 0. \tag{3}$$

In light of [24], it is preferred to learn a sparse z_i which has exactly k nonzero values. Thus, the learned $\boldsymbol{Z}$ is sparse, and the computation burden of subsequent spectral analysis can be largely alleviated. The parameter γ can be set as $\gamma = \frac{k}{2}d_{i,k+1} - \frac{1}{2}\sum_{j=1}^{k} d_{ij}$, such that optimal solution to Eq. (3) is

$$z_{ij} = \frac{d_{i,k+1} - d_{ij}}{kd_{i,k+1} - \sum_{j=1}^{k} d_{ij}}. \tag{4}$$

For detail derivation, see [24]. The computational complexity of calculating matrix $\boldsymbol{Z}$ using Eq. (4) is $O(ndm)$. To improve the efficiency of the anchor graph construction, we investigate an ANNS technique so as to achieve k-nearest neighbors matching. Thus, the process of computing the matrix $\boldsymbol{Z}$ can be efficiently implemented in $O(nd\log(m))$ [21].

Accordingly, the similarity matrix $\boldsymbol{A}$ can be obtained by [18]

$$\boldsymbol{A} = \boldsymbol{Z}\boldsymbol{\Delta}^{-1}\boldsymbol{Z}^T, \tag{5}$$

where the diagonal matrix $\boldsymbol{\Delta} \in \mathbb{R}^{m\times m}$ is defined as $\Delta_{jj} = \sum_{i=1}^{n} z_{ij}$.

2.2 Feature selection with anchor graph and $\ell_{2,1}$-norm regularization

According to the manifold learning theory, high-dimensional data lies in or is close to a low-dimensional manifold, and there is always a matrix $\boldsymbol{W} \in \mathbb{R}^{d\times l}$ that can preserve the manifold structure after projection, where d is the original dimension, l is the projection dimension. A typical dimensionality reduction algorithm is proposed in [12, 27] to solve the following problem:

$$\min_{\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}} Tr(\boldsymbol{W}^T\boldsymbol{X}^T\boldsymbol{L}\boldsymbol{X}\boldsymbol{W}), \tag{6}$$

where $\boldsymbol{I}$ denotes the identity matrix, $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ is Laplacian matrix which is defined by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, $\boldsymbol{D}$ is a diagonal matrix, and the i-th entry is defined as $D_{ii} = \sum_{j=1}^{n} a_{ij}$. Problem (6) is a dimensionality reduction model and the projected feature is a linear combination of all original features. However, in many applications, we are more interested in the feature selection model. Since only a few important features are involved in the projection and the i-th row of matrix $\boldsymbol{W}$ could be used to measure the importance of i-th feature of original data. The task of our proposed method is to find the optimal projection matrix that is constrained to be a row sparse matrix, which can be achieved by means of $\ell_{2,1}$-norm regularization. Therefore, problem (6) can be rewritten as follows for feature selection:

$$\min_{\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}} Tr(\boldsymbol{W}^T\boldsymbol{X}^T\boldsymbol{L}\boldsymbol{X}\boldsymbol{W}) + \alpha\|\boldsymbol{W}\|_{2,1}, \tag{7}$$

where α is the regularization parameter, $\|\boldsymbol{W}\|_{2,1}$ is defined as $\sum_{i=1}^{d}\|\boldsymbol{w}_i\|_2$, where $\boldsymbol{w}_i \in \mathbb{R}^{l\times 1}$ is the transpose of the i-th row of $\boldsymbol{W}$. From (5), $\boldsymbol{A}$ can be written as $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{B}^T$, where $\boldsymbol{B} = \boldsymbol{Z}\boldsymbol{\Delta}^{-\frac{1}{2}}$. For the degree of each data point, we have $D_{ii} = \sum_{sj} Z_{is}(\Delta_{ss})^{-1}Z_{js} = \sum_s Z_{is} = 1$. Therefore, it is easy to see that $\boldsymbol{A}$ is a double stochastic matrix, and we obtain the diagonal matrix $\boldsymbol{D} = \boldsymbol{I}$ and $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T$. Accordingly, we propose our fast unsupervised feature selection (FUFS) model by solving the following problem:

$$\min_{\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}} Tr[\boldsymbol{W}^T\boldsymbol{X}^T(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{X}\boldsymbol{W}] + \alpha\sum_{i=1}^{d}\|\boldsymbol{w}_i\|_2. \tag{8}$$

Obviously, $\|\boldsymbol{w}_i\|_2$ can be zero in theory, however, this will make Eq. (8) non-differentiable. To avoid this issue, $\|\boldsymbol{w}_i\|_2$ is replaced by $\sqrt{\boldsymbol{w}_i^T\boldsymbol{w}_i + \varepsilon}$ to make Eq. (8) differentiable, where ε is a small enough constant. Therefore, we obtain

$$\min_{\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}} Tr[\boldsymbol{W}^T\boldsymbol{X}^T(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{X}\boldsymbol{W}] + \alpha\sum_{i=1}^{d}\sqrt{\boldsymbol{w}_i^T\boldsymbol{w}_i + \varepsilon}, \tag{9}$$

which is evidently equal to problem (8) when ε is infinitely close to zero. The Lagrangian function of problem (9) is

$$\begin{aligned}\mathcal{L}(\boldsymbol{W}, \boldsymbol{\Lambda}) = {} & Tr[\boldsymbol{W}^T\boldsymbol{X}^T(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{X}\boldsymbol{W}] \\ & +\alpha\sum_{i=1}^{d}\sqrt{\boldsymbol{w}_i^T\boldsymbol{w}_i + \varepsilon} + Tr(\boldsymbol{\Lambda}(\boldsymbol{W}^T\boldsymbol{W} - \boldsymbol{I})),\end{aligned} \tag{10}$$

where $\boldsymbol{\Lambda}$ is the Lagrangian multiplier. By taking the derivative of $\mathcal{L}(\boldsymbol{W}, \boldsymbol{\Lambda})$ w.r.t $\boldsymbol{W}$, and set the derivative to zero, we obtain

$$\frac{\partial\mathcal{L}(\boldsymbol{W},\boldsymbol{\Lambda})}{\partial\boldsymbol{W}} = \boldsymbol{X}^T(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{X}\boldsymbol{W} + \alpha\boldsymbol{Q}\boldsymbol{W} + \boldsymbol{W}\boldsymbol{\Lambda} = 0. \tag{11}$$

where $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$ is a diagonal matrix, and the i-th element is defined as

$$\boldsymbol{Q}_{ii} = \frac{1}{2\sqrt{\boldsymbol{w}_i^T\boldsymbol{w}_i + \varepsilon}}. \tag{12}$$

Note that $\boldsymbol{Q}$ is also a unknown variable and dependent on $\boldsymbol{W}$. We now propose an alternative iterative algorithm to solve problem (9). When $\boldsymbol{W}$ is fixed, then $\boldsymbol{Q}$ is obtained by Eq. (12). When $\boldsymbol{Q}$ is fixed, solving Eq. (11) is equivalent to solving

$$\min_{\boldsymbol{W}^T\boldsymbol{W}=\boldsymbol{I}} Tr[\boldsymbol{W}^T\boldsymbol{X}^T(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{X}\boldsymbol{W}]+\alpha Tr(\boldsymbol{W}^T\boldsymbol{Q}\boldsymbol{W}), \tag{13}$$

and problem (13) can be solved directly to obtain $\boldsymbol{W}$. Let $\boldsymbol{M} = \boldsymbol{X}^T(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{X}$; as such, the details of this algorithm are summarized in Algorithm 1. The convergence of Algorithm 1 has been proved in our previous work. For detail and proof, see lemma 1 in [26].

Algorithm 1 Algorithm to solve problem (8)

```
Input:  Data matrix $\boldsymbol{X} \in \mathbb{R}^{n\times b}$, projection dimension $l$. Initialize $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$ as $\boldsymbol{Q} = \boldsymbol{I}$,
        calculate $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ as $\boldsymbol{B} = \boldsymbol{Z}\boldsymbol{\Delta}^{-\frac{1}{2}}$.
        repeat
            1. Fix $\boldsymbol{Q}$, $\boldsymbol{W}$ is formed by the $l$ eigenvectors of $(\boldsymbol{M} + \gamma\boldsymbol{Q})$
               corresponding to the $l$ smallest eigenvalues.
            2. Fix $\boldsymbol{W}$, $\boldsymbol{Q}$ is obtained by (12).
        until converge
Output: Sort each feature according to $||\boldsymbol{w}_i||_2 (i = 1, \cdots, d)$ in descending order, select
        the top-ranked ones.
```

2.3 Computational complexity analysis

Our proposed method (FUFS) consists of four steps:

1. We need $O(1)$ and $O(ndmt)$ to generate m anchors by random selection and the k-means method respectively, where t is the iterative number of the k-means.
2. We need $O(nd\log(m))$ to obtain the matrix $\boldsymbol{Z}$.
3. We need $O(ndm + nd^2 + d^2m)$ to calculate $\boldsymbol{M}$.
4. We need $O(d^2l)$ to obtain projection matrix W by performing eigenvalue decomposition on $(\boldsymbol{M} + \gamma\boldsymbol{Q})$.

Considering that $d \ll m \ll n$ for very large-scale data sets, l and the iterative number of our proposed iterative algorithm are usually fairly small; the overall computational complexity of FUFS-R and FUFS-K is $O(ndm)$ and $O(ndmt)$ respectively.

3 Experiments

In this section, several experiments are performed to demonstrate the effectiveness and efficiency of our proposed method (FUFS), and then show several analysis of experimental results. As there are two common ways to generate anchor points, FUFS can be further subdivided into FUFS-R and FUFS-K, where anchor points are generated by random selection and the k-means method respectively.

3.1 Experimental setup

We conduct experiments on four benchmark data sets in terms of clustering and running time. These data sets include one face data set (MSRA25), three hand written digit image data sets (USPS, MNIST and Extended MNIST). The above four data sets can be categorized into small, medium and large sizes. In our experiment, we regard MSRA25 and USPS as small-sized data sets, MNIST as medium-sized data set, while Extended MNIST is considered large-sized data sets. The important statistics of these data sets are summarized in Table 1.

For our proposed method (FUFS), there are two unique parameters that need to be set in advance: projection dimension (i.e., l) and the number of anchor points (i.e., m). In our experiment, we set the projection dimension as the number of clusters for FUFS-R and FUFS-K, then set the number of anchor points as 500 for small-sized data sets, 1000 for medium-sized data sets, and 2000 for large-sized data sets. To ensure a fair comparison between the different unsupervised feature selection algorithms, we fix $k = 5$ for all data sets to specify the size of neighborhoods. The number of selected features is set from half of the total number to the full feature size, while all other parameters are tuned from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. After the different combinations of parameters are fixed, each feature selection algorithm is first executed to rank features, after which the k-means was repeated 30 times in the selected feature subspace and compute the average results to alleviate the stochastic effect. Owing to space limitations, only the best results from the optimal parameters are reported here. All these compared methods are implemented in MATLAB R2015b, and run on a Windows 7 machine with 3.40 GHz i7-6700 CPU, 16 GB main memory.

3.2 Evaluation metrics

To evaluate the clustering results, we adopt two widely used evaluation metrics to measure the learning performance:

Clustering Accuracy (ACC) discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. Clustering Accuracy is defined as follows:

$$ACC = \frac{\sum_{i=1}^{n} \delta(map(r_i), l_i)}{n}, \tag{14}$$

where r_i denotes the cluster label of $\boldsymbol{x}_i$ and l_i denotes the true class label, n is the total number of samples, $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data set. The larger the value of ACC is, the better performance is.

Table 1 Data Set Description

Data Set	Samples	Features	Classes
MSRA25	1799	256	12
USPS	9298	256	10
MNIST	70000	784	10
Extended MNIST	630000	900	10

Normalized Mutual Information (NMI) is used for determining the quality of clusters. According to the definition in [30], NMI is estimated by

$$NMI = \frac{\sum_{i=1}^{c}\sum_{i=1}^{c} n_{i,j} \log \frac{n_{i,j}}{n_i \hat{n}_j}}{\sqrt{(\sum_{i=1}^{c} n_i \log \frac{n_i}{n})(\sum_{j=1}^{c} \hat{n}_j \log \frac{\hat{n}_j}{n})}}, \tag{15}$$

where n_i is the number of data contained in the cluster $C_i (1 \leq i \leq c)$, which is generated by a clustering algorithm. While $\hat{n}_j$ is the number of data belonging to the j-th ground truth cluster, and $n_{i,j}$ denotes the number of data that are in the intersection between cluster C_i and the class L_j. Similarly,a larger NMI indicates a better clustering result.

3.3 Compared algorithms

In the experiments, we have compared our methods (FUFS-R and FUFS-K) with following unsupervised feature selection approaches:

- Baseline: All original features are adopted as the baseline in the experiments.
- LS: Laplacian Score [9] where features are ranked according to their power of locality preserving in a descending order.
- MCFS: Multi-Cluster Feature Selection [1] which selects those features that can best preserve multi-cluster structure of the data by using spectral regression with ℓ_1-norm regularization.
- UDFS: Unsupervised Discriminative Feature Selection [35] which simultaneously exploits discriminative information and feature correlations.
- NDFS: Nonnegative Discriminative Feature Selection [15] which selects features by a joint framework of nonnegative spectral analysis and $\ell_{2,1}$-norm regularized regression.

3.4 Results and analysis

The average running time of all methods is shown in Table 2, while the clustering performance is shown in Tables 3 and 4. From the results, the following observations can be made. First, feature selection can not only make the subsequent processing more efficient by selecting a subset of original features, but also significantly improves the learning performance. All feature selection methods exhibit better performance than Baseline in terms of ACC. This is mainly caused by the removal of redundant and noisy features. Second, for small-sized data sets, our FUFS method has no obvious advantages over traditional graph-based unsupervised feature selection methods; however, the performance is much better than LS, which has the lowest computational complexity of all the compared methods. Moreover, for medium-sized and large-sized data sets, the proposed FUFS-R and FUFS-K achieve

Table 2 Running time (in seconds) for different selection methods

Data Set	LS	MCFS	UDFS	NDFS	FUFS-R	FUFS-K
MSRA25	**0.128**	1.036	1.252	2.012	**0.747**	1.063
USPS	**0.967**	22.239	23.665	7.969	**2.454**	5.041
MNIST	918.7	1147.5	3980.3	5086.1	**33.4**	**95.4**
Extended MNIST	OM	OM	OM	OM	**293.8**	**3458.5**

(Top 2 highest-ranked methods are highlighted in bold, 'OM' means 'out-of-memory error')

Table 3 Clustering results (ACC%) for different selection methods

Data Set	Baseline	LS	MCFS	UDFS	NDFS	FUFS-R	FUFS-K
MSRA25	56.5	59.6	**63.1**	59.0	**64.2**	61.8	62.0
USPS	67.7	68.0	69.1	68.1	**75.6**	72.5	**76.5**
MNIST	57.9	59.4	59.9	**60.2**	**60.4**	58.7	59.9
Extended MNIST	46.3	–	–	–	–	**50.1**	**50.9**

(Top 2 highest-ranked methods are highlighted in bold)

significant improvements in running time. FUFS-R and FUFS-K only need 33.474 and 95.466 seconds, respectively, which is 28 and 6 times faster than the third-fastest method (LS) on the medium-sized data set (MNIST). Meanwhile, our methods achieve competitive performance for almost all data sets, and FUFS-K achieves the best performance on USPS and Extended MNIST. In addition, we can expect to obtain a more accurate similarity matrix as the number of anchor points increases, meaning that the performance of both FUFS-R and FUFS-K could conceivably achieve even higher performance. Third, although FUFS-R selects anchor points randomly, its performance is not significantly lower than that of FUFS-K, while the running time is much smaller than that of FUFS-K. Compared with FUFS-R, the extra running time of FUFS-K is derived from the use of the k-means method to generate anchors, and this method may require a large number of iterations to converge in some cases. Therefore, considering both accuracy and efficiency, FUFS-R is the best unsupervised feature selection method among all of the compared approaches, especially for very large-scale data sets.

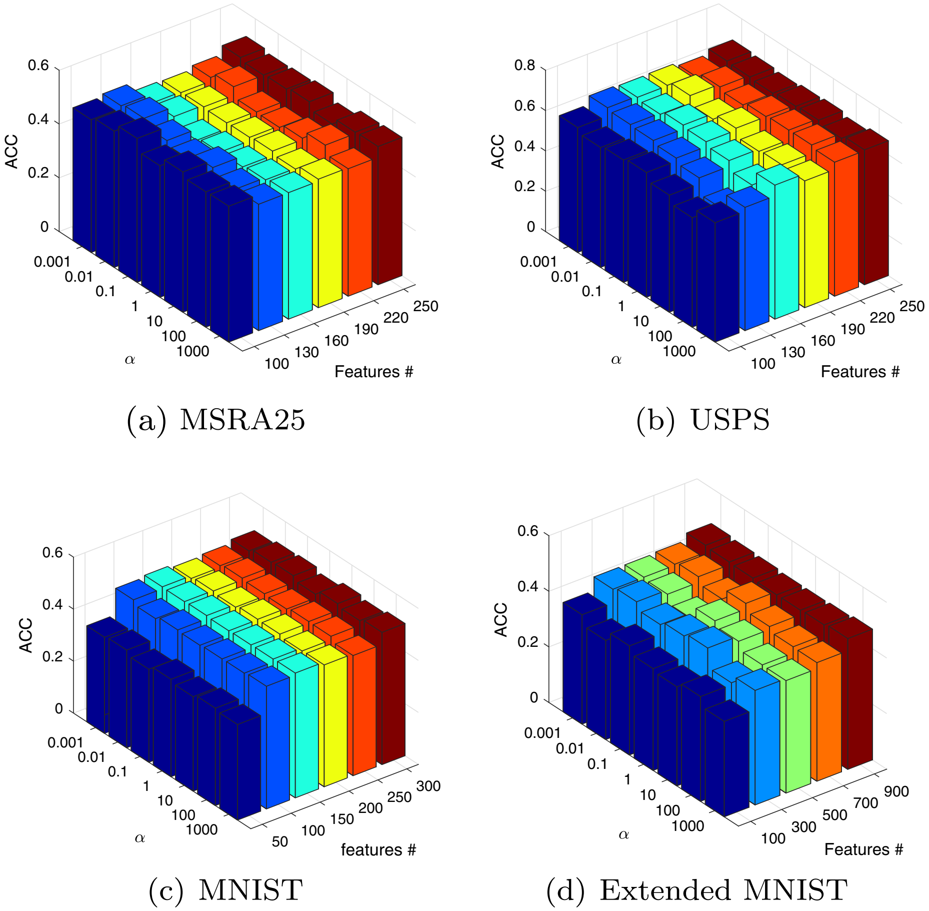
3.5 Studies on parameter sensitivity and convergence

In this subsection, we first evaluate parameter sensitivity for our proposed method FUFS-R. From the results shown in Figs. 1 and 2, we can see that different combinations of parameters may result in different learning performance. To illustrate the influence of the regularization parameter and number of features on the learning performance, we conduct experiments to assess Clustering Accuracy on four benchmark data sets. Figure 1 shows that our method is robust to the parameter α with wide ranges, while comparatively sensitive to the number of selected features. We also study the parameter sensitivity with regard to the number of anchor points. Figure 2 shows that as the number of anchor points increases, the performance does not always improve, while the running time increases in a linear fashion. Thus, we can make a tradeoff between computational complexity and learning performance by selecting an appropriate number of anchor points. Theoretically, the optimal number of

Table 4 Clustering results (NMI%) for different selection methods

Data Set	Baseline	LS	MCFS	UDFS	NDFS	FUFS-R	FUFS-K
MSRA25	55.8	62.4	**68.9**	65.3	**67.7**	64.9	67.6
USPS	60.4	61.5	61.8	61.5	**64.1**	62.5	**64.3**
MNIST	48.1	51.4	**52.2**	49.5	**51.5**	49.4	51.2
Extended MNIST	38.8	–	–	–	–	**40.1**	**40.6**

(Top 2 highest-ranked methods are highlighted in bold)



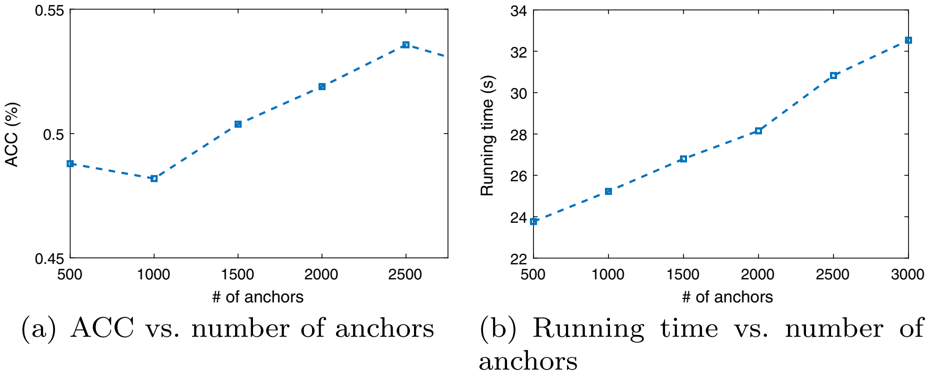
(a) MSRA25

(b) USPS

(c) MNIST

(d) Extended MNIST

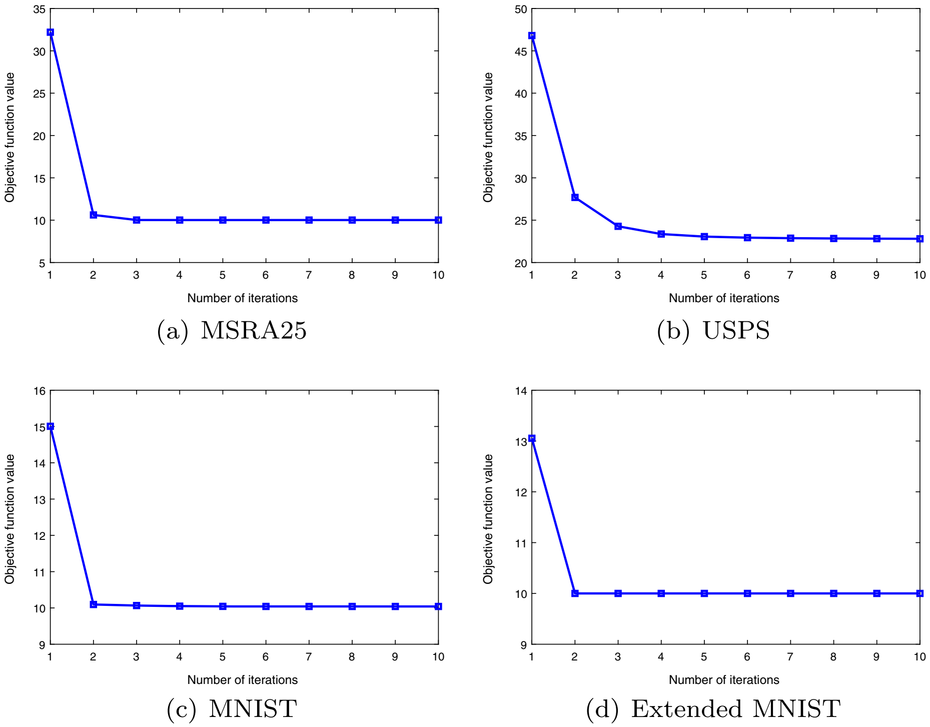
Fig. 1 Clustering Accuracy versus of different α and feature numbers on four data sets



(a) ACC vs. number of anchors

(b) Running time vs. number of anchors

Fig. 2 ACC and Running time versus number of anchors on MNIST



(a) MSRA25

(b) USPS

(c) MNIST

(d) Extended MNIST

Fig. 3 Convergence curves of the proposed algorithm (FUFS-R)

anchor points is larger corresponding to a larger scale data set; however, determining the optimal number of anchor points for different data sets is still an open problem.

To solve the objective function, we have developed an efficient iterative algorithm. And now we experimentally study the speed of its convergence. Figure 3 shows the objective function values versus the number of iteration. From the figure, we can see that the convergence curves of the objective value monotonically decreases with a faster speed, and our algorithm converges to the optimum, almost within four iterations. The fast convergence of Algorithm 1 validates the efficiency of our proposed method.

4 Conclusion

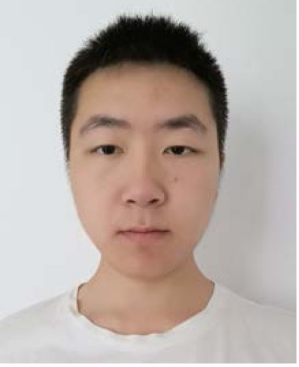
In this paper, we propose a fast graph-based unsupervised feature selection method, which applies an anchor-based strategy to construct a similarity graph by means of a parameter-free adaptive neighbor assignment strategy with fast approximate nearest neighbor matching, then adds a $\ell_{2,1}$-norm regularization into the objective function. To solve the optimization problem of FUFS, an efficient iterative algorithm was executed to obtain the projection matrix. Extensive experiments have shown that FUFS-R overcomes the limitations of existing graph-based strategy in dealing with extremely large-scale data sets.

Acknowledgements This paper is supported in part by the National Natural Science Foundation of China under Grant 61401471, Grant 61772427 and Grant 61751202 and in part by the China Postdoctoral Science Foundation under Grant 2014M562636.

References

1. Cai D, Zhang C, He X (2010) Unsupervised feature selection for multi-cluster data. In: Proceedings of the conference on knowledge discovery and data mining, pp 333–342
2. Cheng Q, Zhou H, Cheng J (2011) The fisher-markov selector: fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data. IEEE Trans Pattern Anal Mach Intell 33(6):1217–33
3. Deng C, Ji R, Liu W, Tao D, Gao X (2013) Visual reranking through weakly supervised multi-graph learning. In: Proceedings of the 2013 IEEE international conference on computer vision, pp 2600–2607
4. Deng C, Ji R, Tao D, Gao X, Li X (2014) Weakly supervised multi-graph learning for robust image reranking. IEEE Trans Multimed 16(3):785–795
5. Dy JG, Brodley CE (2004) Feature selection for unsupervised learning. J Mach Learn Res 5(4):845–889
6. Freeman C, Kulic D, Basir O (2013) Feature-selected tree-based classification. IEEE Trans Cybern 43(6):1990–2004
7. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
8. Hancock T, Mamitsuka H (2012) Boosted network classifiers for local feature selection. IEEE Trans Neural Netw Learn Syst 23(11):1767–1778
9. He X, Cai D, Niyogi P (2005) Laplacian score for feature selection. In: Proceedings of the conference on neural information processing systems, pp 507–514
10. Hou C, Nie F, Li X, Yi D, Wu Y (2014) Joint embedding learning and sparse regression: a framework for unsupervised feature selection. IEEE Trans Cybern 44(6):793
11. Hou C, Nie F, Tao H, Yi D (2017) Multi-view unsupervised feature selection with adaptive similarity and view weight. IEEE Trans Knowl Data Eng 29(9):1998–2011
12. Kokiopoulou E, Saad Y (2007) Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique. IEEE Trans Pattern Anal Mach Intell 29(12):2143–2156
13. Lai HJ, Pan Y, Tang Y, Yu R (2013) Fsmrank: feature selection algorithm for learning to rank. IEEE Trans Neural Netw Learn Syst 24(6):940–952
14. Laporte L, Flamary R, Canu S, Djean S, Mothe J (2014) Nonconvex regularizations for feature selection in ranking with sparse svm. IEEE Trans Neural Netw Learn Syst 25(6):1118–1130
15. Li Z, Yang Y, Liu J, Zhou X, Lu H (2012) Unsupervised feature selection using nonnegative spectral analysis. In: Proceedings of the AAAI conference on artificial intelligence, pp 1026–1032
16. Li Y, Si J, Zhou G, Huang S, Chen S (2015) Frel: a stable feature selection algorithm. IEEE Trans Neural Netw Learn Syst 26(7):1388
17. Ling X, Qiang MA, Min Z (2013) Tensor semantic model for an audio classification system. Sci Chin 56(6):1–9
18. Liu W, He J, Chang SF (2010) Large graph construction for scalable semi-supervised learning. In: Proceedings of the international conference on machine learning, pp 679–686
19. Liu W, Wang J, Chang SF (2012) Robust and scalable graph-based semisupervised learning. Proc IEEE 100(9):2624–2638
20. Luo M, Nie F, Chang X, Yi Y, Hauptmann AG, Zheng Q (2017) Adaptive unsupervised feature selection with structure regularization. IEEE Trans Neural Netw Learn Syst PP(99):1–13
21. Muja M, Lowe DG (2014) Scalable nearest neighbor algorithms for high dimensional data. IEEE Trans Pattern Anal Mach Intell 36(11):2227–2240
22. Nie F, Huang H, Cai X, Ding CH (2010) Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In: Proceedings of the conference on advances in neural information processing systems, pp 1813–1821
23. Nie F, Xiang S, Jia Y, Zhang C, Yan S (2008) Trace ratio criterion for feature selection. In: Proceedings of the AAAI conference on artificial intelligence, pp 671–676
24. Nie F, Wang X, Huang H (2014) Clustering and projected clustering with adaptive neighbors. In: Proceedings of the conference on knowledge discovery and data mining, pp 977–986
25. Nie F, Wang X, Jordan M, Huang H (2016) The constrained laplacian rank algorithm for graph-based clustering. In: Proceedings of the AAAI conference on artificial intelligence, pp 1969–1976
26. Nie F, Zhu W, Li X (2016) Unsupervised feature selection with structured graph optimization. In: Proceedings of the AAAI conference on artificial intelligence, pp 1302–1308
27. Peng Y, Lu BL (2017) Discriminative extreme learning machine with supervised sparsity preserving for image classification. Neurocomputing
28. Qian M, Zhai C (2013) Robust unsupervised feature selection. In: Proceedings of the international joint conference on artificial intelligence, pp 1621–1627
29. Romero E, Sopena JM (2008) Performing feature selection with multilayer perceptrons. IEEE Trans Neural Netw 19(3):431–41

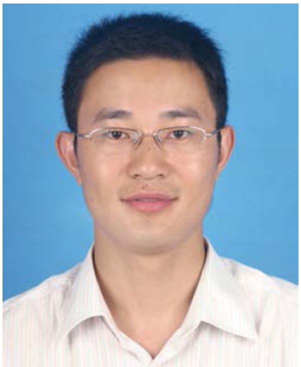
30. Strehl A, Ghosh J (2002) Cluster ensembles: a knowledge reuse framework for combining partitionings. In: Eighteenth national conference on artificial intelligence, pp 93–98
31. Wang R, Nie F, Yang X, Gao F, Yao M (2015) Robust 2DPCA with non-greedy ℓ_1-norm maximization for image analysis. IEEE Trans Cybern 45(5):1108–1112
32. Wang R, Nie F, Hong R, Chang X, Yang X, Yu W (2017) Fast and orthogonal locality preserving projections for dimensionality reduction. IEEE Trans Image Process 26(10):5019–5030
33. Xiang S, Nie F, Zhang C, Zhang C (2009) Nonlinear dimensionality reduction with local spline embedding. IEEE Trans Knowl Data Eng 21(9):1285–1298
34. Xing L, Dong H, Jiang W, Tang K (2017) Nonnegative matrix factorization by joint locality-constrained and $\ell_{2,1}$-norm regularization. Multimed Tools Appl. https://doi.org/10.1007/s11042-017-4970-9
35. Yang Y, Shen HT, Ma Z, Huang Z, Zhou X (2011) $\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning., In: Proceedings of the international joint conference on artificial intelligence, pp 1589–1594
36. Yu Q, Wang R, Yang X, Li BN, Yao M (2016) Diagonal principal component analysis with non-greedy ℓ_1-norm maximization for face recognition. Neurocomputing 171:57–62
37. Zhao Z, Liu H (2007) Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the international conference on machine learning, pp 1151–1157



Haojie Hu received his B.S. degree from High-Tech Institute of Xi'an, Xi'an, China in 2016. He is currently pursuing the M.S. degree in High-Tech Institute of Xi'an, Xi'an, China. His research interests focus on machine learning and its applications.

Rong Wang received the B.E. degree in information engineering, the M.E. degree in signal and information processing, and the Ph.D. degree in computer science from the Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004, 2007, and 2013, respectively. From 2007 to 2013, he was also with the Department of Automation, Tsinghua University, Beijing, China, for the Ph.D. degree. His current research interests include machine learning and signal processing, together with their applications including pattern recognition, image processing, and computer vision.

Feiping Nie received the Ph.D. degree in computer science from Tsinghua University, China, in 2009. He is currently a Full Professor with Northwestern Polytechnical University, China. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval. He has authored over 100 papers in the following top journals and conferences: TPAMI, IJCV, TIP, TNNLS/TNN, TKDE, Bioinformatics, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His papers have been cited over 7000 times. He is serving as an associate editor or a PC member for several prestigious journals and conferences in the related fields.



Xiaojun Yang received the Ph.D. degree from theHigh-Tech Institute of Xi'an, Xi'an, China, in 2010. He is currently an Associate Professor with the Guangdong University of Technology. His research interests include machine learning, signal processing, and data fusion.

Weizhong Yu received the Ph.D. degree from the High-Tech Institute of Xi'an, China, in 2012. He is currently a Post-Doctoral Research Associate with the School of Electronics and Information Engineering, Xi'an Jiaotong University. His main research interests include machine learning and computer vision.