Unsupervised Feature Selection by Graph Optimization

Zhihong Zhang[1], Lu Bai[2(✉)], Yuanheng Liang[3], and Edwin R. Hancock[4]

[1] Software School, Xiamen University, Xiamen, Fujian, China
[2] School of Information, Central University of Finance and Economics, Beijing, China
bailu69@hotmail.com
[3] School of Mathematical Sciences, Xiamen University, Xiamen, Fujian, China
[4] Department of Computer Science, University of York, York, UK

Abstract. Graph based methods have played an important role in machine learning due to their ability to encode the similarity relationships among data. A commonly used criterion in graph based feature selection methods is to select the features which best preserve the data similarity or a manifold structure derived from the entire feature set. However, these methods separate the processes of learning the feature similarity graph and feature ranking. In practice, the ideal feature similarity graph is difficult to define in advance. Because one needs to assign appropriate values for parameters such as the neighborhood size or the heat kernel parameter involved in graph construction, the process is conducted independently of subsequent feature selection. As a result the performance of feature selection is largely determined by the effectiveness of graph construction. In this paper, on the other hand, we attempt to learn a graph strucure closely linked with the feature selection process. The idea is to unify graph construction and data transformation, resulting in a new framework which results in an optimal graph rather than a predefined one. Moreover, the $\ell_{2,1}$-norm is imposed on the transformation matrix to achieve row sparsity when selecting relevant features. We derive an efficient algorithm to optimize the proposed unified problem. Extensive experimental results on real-world benchmark data sets show that our method consistently outperforms the alternative feature selection methods.

Keywords: Graph learning · Laplacian · Unsupervised feature selection

1 Introduction

Recently, graph-based methods, such as spectral embedding [2], spectral clustering [1], and semi-supervised learning [3] [4], have played an important role in machine learning due to their ability to encode the similarity relationships among data. Various applications of graph-based methods can be found in clustering [1] [5], data mining [6], manifold learning [7] [8], subspace learning [9] and speech recognition [10]. A preliminary step for all these graph-based methods

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part I, LNCS 9279, pp. 130–140, 2015.
DOI: 10.1007/978-3-319-23231-7_12

is to establish a suitable graph over the training data. Data samples are represented as vertices of the graph and the edges represent the pairwise similarity relationships between them.

In feature selection, a particularly attractive feature of graph representations is that they provide a universal and flexible framework that reflects the underlying manifold structure and the relationships between feature vectors. A frequently used criterion in graph-based feature selection methods is to select the features which best preserve the data similarity or a manifold structure derived from the entire feature set. The best known methods are the Laplacian score (LapScore) [9], spectral feature selection (SPEC) [11], multicluster feature selection (MCFS) [12] and minimum redundancy spectral feature selection (MRSF) [13]. However, a common problem in the aforementioned methods is that the graph constructing process is independent of the subsequent feature selection task. For example, MCFS [12] uses a graph to characterize the manifold structure and performs locality preserving projection (LPP) in the first-step. In the second step, MCFS performs spectral regression using a single eigenvector at a time to estimate element sparsity. Finally, a new score rule is designed to rank the goodness of the features using element sparsity. MRSF [13], on theother hand, uses the $\ell_{2,1}$-norm regularizer to replace the ℓ_1-norm regularizer in MCFS which leads to row sparsity. The row sparsity used in MRSF is better fitted for feature selection than the element sparsity used in MCFS. LapScore [9] uses a k-nearest neighbor graph to model the local geometric structure of the data and then selects the features that are most consistent with the graph structure. The SPEC [11] algorithm is an extension of LapScore aimed at making it more robust to noise.

Compared with traditional unsupervised feature selection approaches, the above methods have been in many cases been demonstrated to perform better. Nevertheless, their performance can also be further improved since they each separate the problems of estimating or learning a similarity graph and feature selection. Once the graph is determined so as to characterize the data sample similarity and underlying manifold structure, it remains fixed in the subsequent feature ranking or regression steps. As a result, the feature selection performance is largely determined by the effectiveness of graph construction. Instead, a recently proposed unsupervised feature selection algorithm called joint embedded learning and sparse regression (JELSR)[14] attempts to learn a graph embedding and a corresponding sparse transformation matrix simultaneously in one single objective function, which result in an automatically-updated graph embedding. Compared with the alternative method MCFS [12], which is to first compute the low dimensional embedding and then, regress each sample to its low dimensional embedding by adding ℓ_1-norm regularization, JELSR has been demonstrated to have superior performance by unifying two objective of MCFS. This is because the objective of sparse transformation matrix regression has also affected the derivation of low dimensional embedding. However, the optimal graph embedding in JELSR depends heavily on the transformed data, without making the best use of the original data information and the data similarity is also not

learned by the algorithm. This easily leads to the instability performance, especially when encountering a "bad" transformation matrix.

To address this problem, in this paper, we propose a novel unsupervised feature selection approach via graph optimization (referred to as UFSGO), which incorporates graph construction into the data transformation, and thus obtains a simultaneous learning framework for graph construction and transformation matrix optimization. More concretely, by adding the $\ell_{2,1}$-norm regularization to the transformation matrix, our new model simultaneously learns the data similarity matrix and sparse transformation matrix to achieve optimal feature selection results. Moreover, in order to fully utilise information in the original data, a square Frobenius divergence term between a predefined graph and its updated realization is added to the objective function. As a result, we formulate an elegant graph update formula which naturally fuses the original and transformed data information. We also provide an effective method to solve the proposed problem. Compared with traditional unsupervised feature selection approaches, our method integrates the merits of graph learning and sparse regression. Experimental result are provided to demonstrate the utility of the method.

2 A Brief Review of Graph-Based Unsupervised Feature Selection Methods

In this section, we review some well-known algorithms for learning-based unsupervised feature selection, all of which are closely related to our proposed method.

1) MCFS and MRSF: MCFS and MRSF are learning based feature selection methods that first compute an embedding and then use regression coefficients to rank each feature. In the first step, both methods compute a low dimensional embedding represented by the co-ordinate matrix Y. One simple way in deriving low dimensional embedding is to use Laplacian Eigenmap (LE) [8], a well known dimensionality reduction method. Denote $\mathbf{Y} = [y_1, y_2, \ldots, y_n]$ and $\hat{y}_i$ as transpose of the i-th row of $\mathbf{Y}$. The idea common to both MCFS and MRSF is to regress all x_i to $\hat{y}_i$. Their differences are used to determine sparseness constraints. MCFS uses ℓ_1-norm regularization and can be regarded as solving the following problems in sequence:

$$\begin{aligned} &arg \min_{\mathbf{Y}\mathbf{Y}^T=I} tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) \\ &arg \min_{\mathbf{W}} \|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha\|\mathbf{W}\|_1 \end{aligned} \tag{1}$$

Similarly, MRSF first computes the embedding by Eigen decomposition of graph Laplacian and then regress with $\ell_{2,1}$-norm regularization. In other words, MRSF can be regarded as solving the following two problems in sequence:

$$\begin{aligned} &arg \min_{\mathbf{Y}\mathbf{Y}^T=I} tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) \\ &arg \min_{\mathbf{W}} \|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1} \end{aligned} \tag{2}$$

MCFS and MRSF employ different sparseness constraints, i.e., ℓ_1 and $\ell_{2,1}$, in constructing a transformation matrix which is used for selecting features. Nevertheless, the low dimensional embedding, i.e., $\mathbf{Y}$, is determined in the first step and remains fixed in the subsequent ranking or regression step. In other words, we do not consider the later requirements of feature selection in deriving the embedding $\mathbf{Y}$. If it cannot only characterizes the manifold structure, but also indicates the requirements of regression, these methods would perform better.

2) JELSR [14]: Instead of simply using the graph Laplacian to characterize high dimensional data structure and then regression, JELSR (joint embedding learning and sparse regression) unifies embedding/learning and sparse regression in constructing a new framework for feature selection:

$$arg \min_{\mathbf{W},\mathbf{Y}\mathbf{Y}^T=I} tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T) + \beta(\|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_2^2 + \alpha\|\mathbf{W}\|_{2,1}) \tag{3}$$

where α and β are balance parameters. The objective function in Eq.(3) is convex with respect to $\mathbf{W}$ and $\mathbf{Y}$. $\mathbf{W}$ and $\mathbf{Y}$ can be updated in an alternative way. As we can see from Eq.(29) in [14], the objective of sparse regression, i.e. the value of $\mathbf{W}$, has also affected the low dimensional embedding, i.e., $\mathbf{Y}$. Alternative methods, such as MCFS and MRSF, minimize $tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T)$ merely. Although JELSR performs better in many cases, the optimal graph embedding in JELSR depends heavily on the transformed data, without making the best use of the original data information and the data similarity $\mathbf{S}$ is also not learned by the algorithm. This easily leads to the instability performance, especially when encountering a "bad" transformation matrix.

3) LPP [18]: LPP (locality preserving projection) constructs a graph by incorporating neighborhood information derived from the data. Using the graph Laplacian, a transformation is computed to map the data into a subspace by optimally maintaining the local neighborhood information. LPP optimizes a linear transformation $\mathbf{W}$ according to

$$arg \min_{\mathbf{W}} \sum_{i,j=1}^{n} \|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|^2 s_{ij} \tag{4}$$

The basic idea underlying LPP is to find a transformation matrix W, which transforms the high-dimensional data $\mathbf{X}$ into a low-dimensional matrix $\mathbf{XW}$, so as to maximally preserve the local connectivity structure of $\mathbf{X}$ with $\mathbf{XW}$. Minimizing (4) ensures that, if x_i and x_j are close, and as a result $\mathbf{W}^T x_i$ and $\mathbf{W}^T x_j$ are close too.

As described above, LPP seeks a low-dimensional representation with the purpose of preserving the local geometry in the original data. However, such "locality geometry" is completely determined by the artificially constructed neighborhood graph. As a result, its performance may drop seriously if given a "bad" graph. Therefore, it is better to optimize graph and learn the transformation simultaneously in a unified objective function.

3 Unsupervised Feature Selection by Graph Optimization

As reported by many researchers [14–17], graph construction plays a crucial role in the success of graph-based learning methods. Typically, to construct a graph, we define neighborhood based on the k nearest neighbors(kNN) method to determine the connectivity of the graph. In general, the size of the neighborhood needs to be specified in advance and fixed throughout the entire learning process. In real-world applications, it is hard to estimate the neighborhood size and different data points have a different optimal neighborhood size. As a result, some undesirable edges and weights are unavoidable. In our study, we incorporate graph construction into the LPP objective function, and thus obtains a simultaneous learning framework for graph construction and transformation optimization. Moreover, in order to perform feature selection, it is desirable to have some rows of the transformation matrix set to be all zeros. This leads us to use the $\ell_{2,1}$-norm on the transformation matrix $\mathbf{W}$, and this leads to row-sparsity of $\mathbf{W}$. The learning problem can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{W}} \sum_{i,j=1}^{n} (\|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|_2^2 s_{ij}) + \alpha \|\mathbf{S} - \mathbf{S}^0\|_F^2 + \mu \|\mathbf{W}\|_{2,1} \\ &s.t. \quad \forall_i, s_i^T \mathbf{1} = 1, 0 \le s_i \le 1, \mathbf{W}^T S_t \mathbf{W} = I \end{aligned} \tag{5}$$

Let the transformation matrix be $\mathbf{W} \in \Re^{d \times m}$ with $m < d$ and the total scatter matrix be $S_t = \mathbf{X}^T H \mathbf{X}$, where $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix. We constrain the subspace with $\mathbf{W}^T S_t \mathbf{W} = I$ such that the data in the subspace are statistically uncorrelated. The predefined graph is $\mathbf{S}^0$ graph, $\|\cdot\|_F^2$ is the squared Frobenius divergence, and α and μ are trade-off parameters.

The first term of the objective function in (5) is similar to LPP in (4), which is designed for preserving local structure, such that if x_i and x_j are "close" then the transformed data $\mathbf{W}^T x_i$ and $\mathbf{W}^T x_j$ are also close. The second term of the objective function uses the squared Frobenius divergence to measure the fitting error of the learned graph similarity $\mathbf{S}$ to the predefined graph similarity $\mathbf{S}^0$. It constrains $\mathbf{S}$ to be close to $\mathbf{S}^0$ in order to make use of original data information.

4 Optimization Algorithm for Problem (5)

To obtain the global minimal solution of (5), we use an iterative and interleaved optimization process, which is summarized in Algorithm 1. At each iteration step, the sparse matrix $\mathbf{W}$ is updated by (9). After obtaining $\mathbf{W}$, we then update U using Eq.(6). Finally, we update s_{ij} by solving the problem given in (11) and obtain the optimal solution using Eq.(12).

Note that $\|\mathbf{W}\|_{2,1}$ is convex. Nevertheless, its derivative does not exist when $\hat{w}_i = 0$ for $i = 1, 2, \ldots, d$. Therefore, we use the definition $tr(\mathbf{W}^T U \mathbf{W}) = \|\mathbf{W}\|_{2,1}/2$ in [14] when $\hat{w}_i$ is not equal to 0. The matrix $U \in \Re^{d \times d}$ is diagonal with i-th diagonal element is where

$$U_{ii} = \frac{1}{2\|\hat{w}_i\|_2} \tag{6}$$

We can then rewrite the proposed method in Eq.(5) as the following problem:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{W},U} tr(\mathbf{W}^T\mathbf{X}^T\mathbf{L}_S\mathbf{X}\mathbf{W}) + \alpha\|\mathbf{S}-\mathbf{S}^0\|_F^2 + \mu tr(\mathbf{W}^T U\mathbf{W}) \\ &s.t. \quad \forall_i, s_i^T\mathbf{1} = 1, 0 \le s_i \le 1, \mathbf{W}^T S_t\mathbf{W} = I \end{aligned} \tag{7}$$

where $\mathbf{L}_S$ is the Laplacian matrix and $\mathbf{L}_S = \mathbf{D} - \mathbf{S}$. $\mathbf{D} \in \Re^{n\times n}$ is a diagonal matrix with the i-th diagonal element as $D_{ii} = \sum_{j=1}^{n} s_{ij}$.

We first fix $\mathbf{S}$ and solve for $\mathbf{W}$ and U , then the problem (7) becomes

$$\begin{aligned} &\min_{\mathbf{W},U} tr(\mathbf{W}^T\mathbf{X}^T\mathbf{L}_S\mathbf{X}\mathbf{W}) + \mu tr(\mathbf{W}^T U\mathbf{W}) \\ &s.t. \quad \mathbf{W}^T S_t\mathbf{W} = I \end{aligned} \tag{8}$$

which can be rewritten as the following problem

$$\begin{aligned} &\min_{\mathbf{W},U} tr\big(\mathbf{W}^T(\mathbf{X}^T\mathbf{L}_S\mathbf{X} + \mu U)\mathbf{W}\big) \\ &s.t. \quad \mathbf{W}^T S_t\mathbf{W} = I \end{aligned} \tag{9}$$

When U is fixed, the optimal solution to the problem in (9) is the spectral decomposition of $S_t^{-1}(\mathbf{X}^T\mathbf{L}_S\mathbf{X} + \mu U)$, i.e., the optimal solution $\mathbf{W}$ is formed by the k eigenvectors of $S_t^{-1}(\mathbf{X}^T\mathbf{L}_S\mathbf{X} + \mu U)$ corresponding to the k smallest eigenvalues (we assume the null space of the data $\mathbf{X}$ is removed, i.e., S_t is invertible). After that, we fix $\mathbf{W}$ and update U by employing the formulation in (6) directly.

When $\mathbf{W}$ and U are fixed, the proposed method given in Eq.(7) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{S}} tr(\mathbf{W}^T\mathbf{X}^T\mathbf{L}_S\mathbf{X}\mathbf{W}) + \alpha\|\mathbf{S}-\mathbf{S}^0\|_F^2 \\ &s.t. \quad \forall_i, s_i^T\mathbf{1} = 1, 0 \le s_i \le 1 \end{aligned} \tag{10}$$

Since $\|\mathbf{S}-\mathbf{S}^0\|_F^2 = tr\big((\mathbf{S}-\mathbf{S}^0)^T(\mathbf{S}-\mathbf{S}^0)\big)$, then Eq.(10) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{n} (\|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|_2^2 s_{ij}) + \alpha\big[\sum_{i,j=1}^{n} s_{ij}^2 - 2\sum_{i,j=1}^{n} s_{ij}s_{ij}^0 + tr(s^{0T}s^0)\big] \\ &s.t. \quad \forall_i, s_i^T\mathbf{1} = 1, 0 \le s_i \le 1 \end{aligned} \tag{11}$$

Taking derivative with respect to s_{ij} and setting it to zero, we have

$$\begin{aligned} &\frac{\partial}{\partial s_{ij}}\Big[(\|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|_2^2 s_{ij}) + \alpha\big[\sum_{i,j=1}^{n} s_{ij}^2 - 2\sum_{i,j=1}^{n} s_{ij}s_{ij}^0 + tr(s^{0T}s^0)\big]\Big] = 0 \\ &\Rightarrow \|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|_2^2 + 2\alpha s_{ij} - 2\alpha s_{ij}^0 = 0 \\ &\quad\Rightarrow s_{ij} = s_{ij}^0 - \tfrac{1}{2\alpha}\|\mathbf{W}^T x_i - \mathbf{W}^T x_j\|_2^2 \end{aligned} \tag{12}$$

From Eq.(12), it is clear that the similarity s_{ij} is not only updated by the initial graph similarity s_{ij}^0 in the original input space, but also updated gradually

in the different transformed progressive spaces by current $\mathbf{W}$ until the algorithm converges. In fact, when α tends to $+\infty$, s_{ij} approaches s_{ij}^0. That is, the learned $\mathbf{S}$ reduces to the predefined $\mathbf{S^0}$. Intuitively, our algorithm will give better discriminative power than typical unsupervised feature selection methods. i.e. JELSR, since it simultaneously learns both the similarity matrix $\mathbf{S}$ and the sparse transformation matrix $\mathbf{W}$.

In summary, we solve the optimization problem in (5) in an iterative and interleaved way. More concretely, we first fix $\mathbf{S}$ and U, thus employing (9) to update $\mathbf{W}$, whose columns are the m eigenvectors of $S_t^{-1}(\mathbf{X}^T\mathbf{L}_S\mathbf{X} + \mu U)$ corresponding to the m smallest eigenvalues. We then fix $\mathbf{W}$ and update U using Eq.(6). Finally, we update s_{ij} by solving the problem in (11) and obtain the optimal solution as Eq.(12).

After the optimal $\mathbf{W}$ is obtained, we then sort all the original d features according to the ℓ_2-norm values of the d rows of $\mathbf{W}$ in descending order, and select the top features.

Algorithm 1. Unsupervised Feature Selection by Graph Optimization (UFSGO)

Input: $\mathbf{X} \in \Re^{n\times d}$, parameter α and μ, $\mathbf{S}^0$.
Output: the optimal sparse transformation matrix $\mathbf{W} \in \Re^{d\times m}$

1: **while** not converge **do**
2: Update $\mathbf{W}$ by (9) whose columns are the k eigenvectors of $S_t^{-1}(\mathbf{X}^T\mathbf{L}_S\mathbf{X} + \mu U)$ corresponding to the k smallest eigenvalues;
3: Update U by Eq.(6);
4: We update s_{ij} by solving the problem (11) and obtain the optimal solution as Eq.(12).
5: **end while**

5 Experiments and Comparisons

To demonstrate the effectiveness of the proposed approach, we conduct experiments on five image data sets, i.e., three face image data set AR, YaleB and ORL, one hand written digit data set MNIST and one shape image data set MPEG-7. Table. 1 summarizes the extents and properties of the four image data-sets.

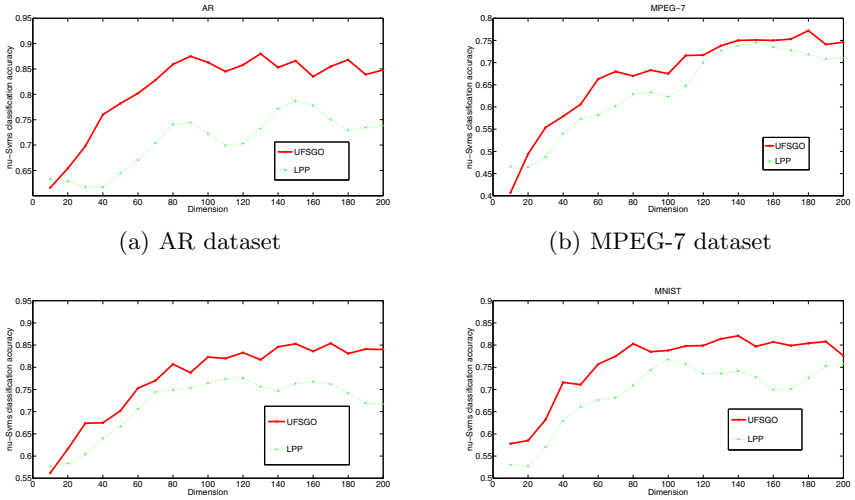
Table 1. Summary of four benchmark image data sets

Data-set	Sample	Features	Classes
AR	1680	2000	120
MPEG-7	1400	6000	70
YaleB	2414	1024	38
MNIST	2000	784	10

Table 2. The best result of all unsupervised methods and their corresponding size of selected feature subset (MEAN ± STD).

Dataset	AR	MPEG-7	YaleB	MNIST
SPEC	80.8% ± 2.23 (180)	72.8% ± 1.88(180)	76% ± 1.88 (140)	80.9% ± 1.91 (200)
JELSR	82.2% ± 1.77 (170)	70% ± 2.98 (190)	81.9% ± 2.34 (180)	77% ± 1.21 (140)
MCFS	78.7% ± 2.89 (120)	73.2% ± 1.03 (190)	77.4% ± 3.66 (200)	76.4% ± 2.56(170)
LapScore	76.9% ± 3.32 (130)	72.3% ± 2.51 (200)	78.7% ± 4.33(200)	76.1% ± 1.73(130)
UFSGO	**89.9**% ± **2.38**(**130**)	**75.6**% ± **1.45**(**180**)	**86.7**% ± **1.67**(**140**)	**81.6**% ± **2.03**(**140**)

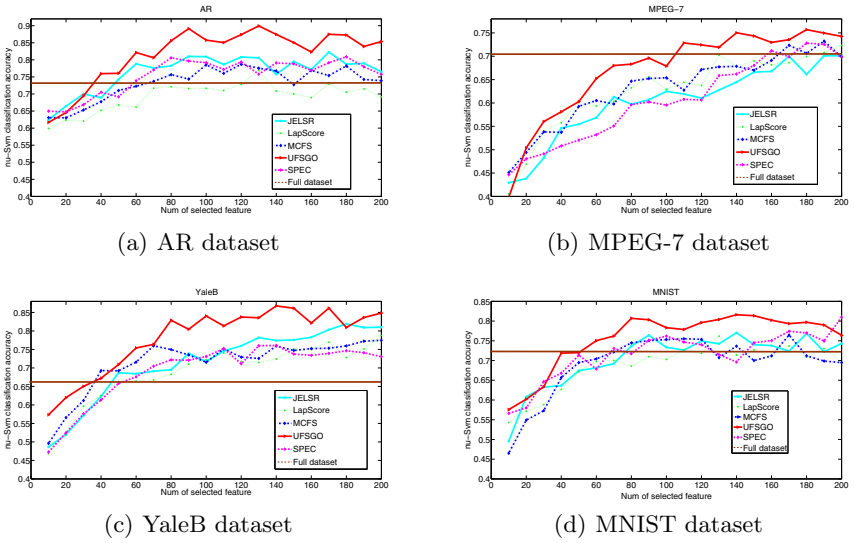
Since our proposed model (see Eq.5) can be interpreted as more generalized version of LPP with additional graph similarity preservation and sparse feature selection capabilities, we begin with evaluating the classification performance based on the proposed method (UFSGO) on the above four publicly available image datasets, compared with LPP. Then, we compare the classification results from UFSGO with four representative unsupervised feature selection algorithms. These methods are LapScore [9], SPEC [11], MCFS [12], JELSR [14]. A 10-fold cross-validation strategy using the nu-Support Vector Machine (nu-SVM) [19] is employed to evaluate the classification performance. The parameters in feature selection algorithms as well as the nu-SVM classifier are tuned via cross-validation on the training data. Specifically, the entire sample is randomly partitioned into 10 subsets and then we choose one subset for test and use the remaining 9 for training, and this procedure is repeated 10 times. The final accuracy is computed by averaging of the accuracies from all experiments.



(a) AR dataset (b) MPEG-7 dataset

(c) YaleB dataset (d) MNIST dataset

Fig. 1. Accuracy rate vs. the variation of dimension on four benchmark image datasets by LPP and UFSGO.



(a) AR dataset (b) MPEG-7 dataset

(c) YaleB dataset (d) MNIST dataset

Fig. 2. Accuracy rate vs. the number of selected features on four benchmark image datasets by nu-SVM.

From Fig.1, we observe that UFSGO can consistently outperform LPP on all the used datasets. This states that UFSGO is more discriminative than LPP, and really benefits from graph updating and optimization process.

As seen from Fig.2, from the statistical view, we can see that our proposed method (UFSGO) achieves significantly better results comparing to the baseline algorithms in all cases. This is obviously because the proposed UFSGO simultaneously learns the graph and a sparse transformation matrix, to achieve the optimal feature selection results, but each of the rival algorithms dichotomise the process of constructing or learning the underlying data graph and subsequent feature ranking. Although both MCFS and JELSR lead to the element sparsity, the classification performance of MCFS is worse than JELSR (see Fig.2(a),(c) and (d)). This occurs because JELSR simultaneously performs manifold learning and regression, but MCFS sequentially performs them. This demonstrated that simultaneously performing manifold learning and regression is better. Comparatively, LapScore gives the worst performance. This is because it does not take feature redundancy into account and is prone to selecting redundant features.

The best result for each method together with the corresponding size of the selected feature subset are shown in Table. 2. In the table, the classification accuracy is shown first and the optimal number of features selected is reported in brackets. Table. 2 clearly show that the proposed method (UFSGO) outperformed all the competing methods in all experiments. For example, our method improved the classification accuracy by 7.7% (AR), 2.4%(MPEG-7), 4.8%(YaleB), 0.7%(MNIST), respectively, compared to the best performances among the competing methods. Based on these results, we argue that the proposed joint graph

optimization and feature selection method help enhance the classification performance. Although JELSR performs better in many cases, the optimal graph embedding in JELSR depends heavily on the transformed data, without making the best use of the original data information and the data similarity $\mathbf{S}$ is also not learned by the algorithm. This easily leads to the instability performance, especially when encountering a "bad" transformation matrix. Comparatively, our proposed method UFSGO simultaneously learns the data similarity matrix and sparse transformation matrix to achieve optimal feature selection results, while the original data information is also embedded into the graph optimization.

6 Conclusion

In this paper, we proposed a novel unsupervised feature selection algorithm. The approach not only investigates a graph optimization method by learning the data similarity matrix but also presents a simultaneously learning of the sparse matrix for feature selection. As a result, the graph in UFSGO is adjustable instead of predefined as in alternative graph based feature selection methods. Moreover, a square Frobenius divergence term between a predefined graph and its updated realization is added to the objective function which can fully utilize information in the original data. Experimental results from unsupervised feature selection cases demonstrate the effectiveness and efficiency of the proposed UFSGO framework.

Acknowledgments. This work is supported by National Natural Science Foundation of China (Grant No.61402389). Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 888–905 (2000)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in neural information processing systems **14**, 585–591 (2001)
3. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. In: Proceedings of The 22nd International Conference on Machine Learning, vol. 74, no. 1, pp. 457–464 (2005)
4. Chung, F.: Spectral Graph Theory. American Mathematical Society (1992)
5. Jain, V., Zhang, H.: A spectral approach to shape-based retrieval of articulated 3D models. Computer-Aided Design **39**(5), 398–407 (2007)
6. Jin, R., Ding, C., Kang, F.: A probabilistic approach for optimizing spectral clustering. Advances in Neural Information Processing systems, vol. 18. MIT Press, Cambridge (2005)
7. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. Proceedings of the Twentieth International Conference on Machine Learning **20**(2), 912–919 (2003)

8. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation **15**(6), 1373–1396 (2003)
9. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in neural information processing systems, pp. 507–514 (2005)
10. Bach, F., Jordan, M.: Learning spectral clustering, with application to speech separation. The Journal of Machine Learning Research **7**, 1963–2001 (2006)
11. Zhao, Z., Liu, H.: Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th International Conference on Machine Learning, pp. 1151–1157 (2007)
12. Cai, D., Zhang, C., He, X.: Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 333–342 (2010)
13. Zhao, Z., Wang, L., Liu, H.: Efficient spectral feature selection with minimum redundancy. In: Proceedings of AAAI, pp. 673–678 (2010)
14. Hou, C., Nie, F., Yi, D., Wu, Y.: Joint embedding learning and sparse regression: A framework for unsupervised feature selection. IEEE Transactions on Cybernetics **44**(6), 793–804 (2014)
15. Liu, X., Wang, L., Zhang, J., Liu, H.: Global and local structure preservation for feature selection. IEEE Transactions on Neural Networks and Learning Systems **25**(6), 1083–1095 (2014)
16. Nie, F., Wang, X., Huang, H.: Clustering and projected clustering with adaptive neighbors. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 977–986 (2014)
17. Zhang, L., Qiao, L., Chen, S.: Graph-optimized locality preserving projections. Pattern Recognition **43**(6), 1993–2002 (2010)
18. He, X., Niyogi, P.: Locality preserving projections. Neural information processing systems. MIT Press, Cambridge (2003)
19. Chang, C., Lin, C.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (2011)