21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Sparsity Score: A New Filter Feature Selection Method Based on l_1 Graph

Mingxia Liu[1, 2], Dan Sun[1], and Daoqiang Zhang[1*]
[1]*Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China*
[2]*Department of Information Science and Technology, TaiShan College, TaiAn 271021, China*
* *E-mail Address: {mingxialiu, dansun, dqzhang}@nuaa.edu.cn*

Abstract

Recently, l_1 graph based analysis using sparse representation has received much attention in pattern recognition and related communities. In this paper, motivated by the success of l_1 graph in dimensionality reduction, we extend it to feature selection and propose a novel filter-type method called Sparsity Score (SS) which ranks features according to their respective sparsity preserving capability. For that aim, a l_1 graph is constructed based on sparse representationon samples, where a l_1-norm based optimization is used to simultaneously determine the graph adjacency structure and corresponding graph weights of the l_1 graph.Experimental results on a series of benchmark data sets show that the proposed SS method achieves better performance than conventional feature selection methods.

1. Introduction

As an important preprocessing step in pattern recognition systems, feature selection aims to identify an optimal feature subset that is most useful in capturing the intrinsic properties of samples. It's known that feature selection is beneficial to facilitate data visualization and data understanding, to reduce the storage requirements and training time, and to overcome the curse of dimensionality for improved prediction performance [1].

Generally, there are two groups of different feature selection algorithms which are feature ranking methods and subset search ones [2]. For each candidate feature subset, the subset search methods evaluate its importance and select the optimal one based on given evaluation measures, such as consistency and correlation [2]. On the other hand, each feature is considered by feature ranking methods individually. Thus, in practice feature ranking methods are usually computationally more efficient than subset selection methods, and are scalable to high-dimensional data sets. In this paper, we are particularly interested in the feature ranking methods.

At present, there exists several well-known feature ranking methods, such as Variance [3], Fisher Score [3], Laplacian Score [4] and Constraint Score [5]. It has been shown that some feature selection methods can be reformulated in a graph-preserving way, i.e., to preserve a predefined graph and its adjacent weight matrix in the reduced feature space [4, 5]. On the other hand, l_1 graph analysis based on sparse representation has attracted a lot of attentions in recent pattern recognition and related communities, and has proven to be robust to data noise [6, 7]. However, to the best of our knowledge, no previous works have used l_1 graph in feature selection studies. In this paper, l_1 graph is introduced into feature selection and a new method called Sparsity Score (SS) is proposed where features are ranked based on their respective sparsity preserving capability.

The rest of this paper is organized as follows. Section 2 introduces the background by briefly reviewing some popular feature selection methods. In Section 3, we present the proposed l_1 graph-based Sparsity Score method. Section 4 reports the experimental results on a series of benchmark data sets. Finally, we conclude this paper in Section 5.

2. Background

Among various feature ranking methods, Variance, Fisher Score, and Laplacian Score are typical ones. Now we briefly introduce these methods as below.

Given a set of data samples $X = [x_1, \dots, x_m], x_i \in R^n, i = 1, \dots, m$, where m is the number of samples and n is the feature dimension. Let f_{ri} denotes the r-th feature of the i-th sample x_i, $i = 1, \dots, m$, $r = 1, \dots, n$. Define $\mu_r = \frac{1}{m}\sum_{i=1}^{m} f_{ri}$. For supervised learning problems, the class labels of the samples are all given as $\{1,2, \dots, K\}$, where K is the number of classes, and the number of samples belonging to the k-th class is denoted as m_k.

Variance utilizes the variance along a feature dimension to reflect the representative power. The Variance score of the r-th feature Var_r should be maximized, and can be obtained as follows [3]:

$$Var_r = \frac{1}{m}\sum_{i=1}^{m}(f_{ri} - \mu_r)^2 \quad (1)$$

With full class labels, Fisher Score is supervised and aims to find features with best discriminative ability. The Fisher score of the r-th feature FS_r, which should be maximize, is computed as follows[3]:

$$FS_r = \frac{\sum_{k=1}^{K} m_k(\mu_r^k - \mu_r)^2}{\sum_{k=1}^{K}\sum_{i=1}^{m_k}(f_{ri}^k - \mu_r^k)^2} \quad (2)$$

Features with larger variances and stronger locality preserving ability are preferred by Laplacian Score. The Laplacian score for each feature should be minimized. The score of the r-th feature LS_r is computed in the following formulation [4]:

$$LS_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}}{\sum_i (f_{ri} - \mu_r)^2 D_{ii}} \quad (3)$$

Here, D is a diagonal matrix with $D_{ii} = \sum_j S_{ij}$, where S_{ij} is defined by the neighborhood relationship between samples $x_i (i = 1, \dots, m)$ as follows:

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, & \text{if } x_i \text{ and } x_j \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where t is a constant to be set.

All these three feature selection methods actually aim to preserve a predefined graph. To be specific, Variance and Fisher Score seek to preserve global graph structures with equal weight for all edges and edges within one class respectively, while Laplacian Score aims to preserve a local graph constructed by connecting data samples in a predefined neighborhood. Inspired by the discriminative power and robustness of sparse representation on a number of tasks such as face recognition [6, 8], we will develop a l_1 graph-based feature selection method below.

3. l_1 Graph-based Feature Selection

In this section, we proposea novel graph-based feature selection method called Sparsity Score (SS), which is based on sparse representation and uses the l_1 graph to construct the graph adjacency and weights simultaneously.

3.1. Sparse Representation and l_1 Graph

In recent years, much attention have been focused on the sparse linear representation with respect to an over-complete dictionary of base elements [9], through which a l_1 graph can be constructed automatically. The main idea of sparse representation is to reconstruct each sample x_i by using as few samples as possible. Through solving the following l_1-norm based optimization problem, a sparse reconstructive weight vector s_i for each x_i can be obtained [7, 8, 10]:

$$min_{s_i} \|s_i\|_1 \ s.t. \ x_i = Xs_i, \mathbf{1} = \mathbf{1}^T s_i \quad (5)$$

where $s_i = [s_{i,1}, \dots, s_{i,i-1}, 0, s_{i,i+1}, \dots s_{i,m}]^T$ is an m-dimensional vector. The i-th element of s is equal to zero implying that x_i is removed from X. For each sample $x_i, i = 1, \dots, m$, we can compute the optimal sparse reconstructive weights vector $\hat{s}_i$, and get the sparse reconstructive weight matrix $S = (\hat{s}_{i,j})_{m \times m}$:

$$S = [\hat{s}_1, \hat{s}_2, \dots, \hat{s}_m]^T \quad (6)$$

where $\hat{s}_i$ is the optimal solution of Eq. (5). It is worth noting that discriminative information may be naturally preserved in the weight matrix S, even if no class label information is used. The reason is that the non-zero entries in $\hat{s}_i$usually correspond to the samples from the same class, which implies that $\hat{s}_i$ may help to distinguish that class from the others. After obtaining the reconstruction weight matrix S by Eq.(6), the l_1 graph including both graph adjacency structure and graph weights can be simultaneously determined from$S$.

To overcome noise and small sample size problems, two modified objective functions are presented as:

$$min_{s_i} \|s_i\|_1, s.t. \| x_i - Xs_i\| < \varepsilon, \mathbf{1} = \mathbf{1}^T s_i \quad (7)$$

$$min_{[s_i^T t_i^T]^T} \|s_i^T t_i^T\|_1, s.t. \begin{bmatrix} x_i \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} X & I \\ \mathbf{1}^T & 0^T \end{bmatrix} \begin{bmatrix} s_i \\ t_i \end{bmatrix} \quad (8)$$

where ε is the error tolerance, and t_i is an n-dimensional vector which is incorporated as a reconstructive compensation term.

3.2. Sparsity Score

We are now in the position to derive our $l_1$graph-based feature selection method, called Sparsity Score (SS). The objective function of Sparsity Score is defined as the following formulation:

$$SS_r = \frac{\sum_{i=1}^{m}(f_{ri} - \sum_{i=1}^{m}\hat{s}_{i,j} f_{rj})^2}{\frac{1}{m}\sum_{i=1}^{m}(f_{ri} - \mu_r)^2} \quad (9)$$

where $\hat{s}_{i,j}$is the entry in sparse reconstruction weight matrix. In Eq. (9), we prefer those features which can best preserve the l_1 graph structure and those with large variance with greater representative power simultaneously. With smaller reconstruction error, as

well as larger variance for r-th feature, the Sparsity score tends to be small, which means the feature is more important.

By simple derivation, we can get the formulation of Sparsity Score rewritten in the following form:

$$SS_r = \frac{f_r^T(I - S - S^T + SS^T)f_r}{f_r^T\left(I - \frac{1}{m}\mathbf{1}\mathbf{1}^T\right)f_r} \quad (10)$$

where S is the sparse reconstruction weight matrix with all training samples, and I is identity matrix. The computational complexity of SS algorithm is $O(m^2)$.

Algorithm 1 gives the detailed procedure of the proposed algorithm where the error tolerance parameter can be tuned through cross validation. We set it as 0.001 in our experiments empirically.

Algorithm1. Sparsity Score (SS)

Input: Data matrix $X = [x_1, x_2, \dots, x_m]$, $x_i \in R^n$; Error tolerance ε.

Output: The ranked feature list

Procedure:

Step 1. Solve the constrained optimization problem in Eq. (7) or Eq. (8);

Step 2. Construct the sparse reconstructive weight matrix$S$using Eq. (6).

Step 3. Compute the sparsity score for each of all the n features using Eq. (10);

Step 4. Rank features based on their sparsity scores in ascending order.

4. Experiments

To evaluate the performance of our method, we apply them on four data sets from UCI machine learning repository and on two gene expression data sets including *Colon Cancer* and *Prostate Cancer*. Characteristics of these data sets are shown in Table 1.

Table 1. Data sets used in our experiments

Data Set	*Dimension*	*Class*	*Size*
Wine	13	3	178
Hepatitis	19	2	155
Ionosphere	33	2	351
Steel Plate Faults	27	7	1941
Colon Cancer	2000	2	62
Prostate Cancer	12600	2	136

4.1. Experimental Setting

We compare our proposed Sparsity Score (SS) with unsupervised methods including Variance (Var) and Laplacian Score (LS), and supervised one including Fisher Score (FS) and Fisher-Markov selector with polynomial kernel (LFS) [11]. The Support Vector Machine (SVM) with RBF kernel and a 10-fold cross validation strategy are adopted to perform classification and compute the average classification accuracy, respectively. To reduce the bias deduced by randomly portioning dataset in cross-validation, this portioning process is repeated for 10 times independently. The average classification accuracies are computed as the final results.

4.2. Experimental Results

Fig. 1 plots the curves of classification accuracy and different selected feature numbers on six data sets, comparing our SS method with other four feature selection algorithms. It's worth noting that SS method is unsupervised while FS and LFS are fully supervised.

Fig. 1 indicates that in most cases the proposed SS method achieves much better performances than all the other four methods on these data sets, especially when small number of features is selected.This is more obvious on two high-dimensional datasets (i.e., *Colon Cancer* and *Prostate Cancer*), where SS achieves much higher accuracies than those of other methods for a large range of numbers of selected features.

Table 2 reports the highest accuracies as well as the numbers of optimal feature dimensions. Note that *a* in the entry '*a (b)*' is the average classification accuracy obtained by cross validation and *b* is the corresponding number of selected features, while accuracies using all features are used as *Baseline*.

Table 2. Average classification accuracies of different feature selection methods

Dataset	*SS*	*Var*	*LS*	*FS*	*LFS*	*Baseline*
Wine	**97.1 (12)**	95.4 (11)	95.8 (10)	95.7 (13)	95.7 (13)	95.7 (13)
Hepatitis	70.2 (3)	66.7 (18)	67.4 (15)	**73.6 (4)**	66.7 (18)	66.0 (19)
Ionosphere	**95.0 (16)**	94.8 (20)	94.9 (23)	94.7 (25)	94.9 (23)	94.5 (33)
Steel Plate Faults	**54.3 (13)**	53.5 (21)	53.9 (21)	53.6 (18)	53.5 (21)	53.2 (24)
Colon Cancer	85.2 (140)	83.2 (20)	81.4 (50)	**85.5 (330)**	82.4 (30)	71.9 (2000)
Prostate Cancer	**70.1 (60)**	61.3 (40)	61.3 (40)	66.4 (1)	61.3 (40)	57.1 (12600)

Similarly, from Table 2, we can see that SS achieves better performances than LS and LFS, and is superior to FS and Var in most cases. On the *Wine* dataset, SS achieves an accuracy of 97.1% which is much higher than those of other methods. Meanwhile, SS got the highest accuracy on *Prostate Cancer* data set with 70.1%.This validates the effectiveness of sparsity (due to l_1 graph structure) in learning features scores. Also, it is worth noting that these experiments are carried not only on two-class data sets but also

multi-class ones, which is much meaningful for practical applications.

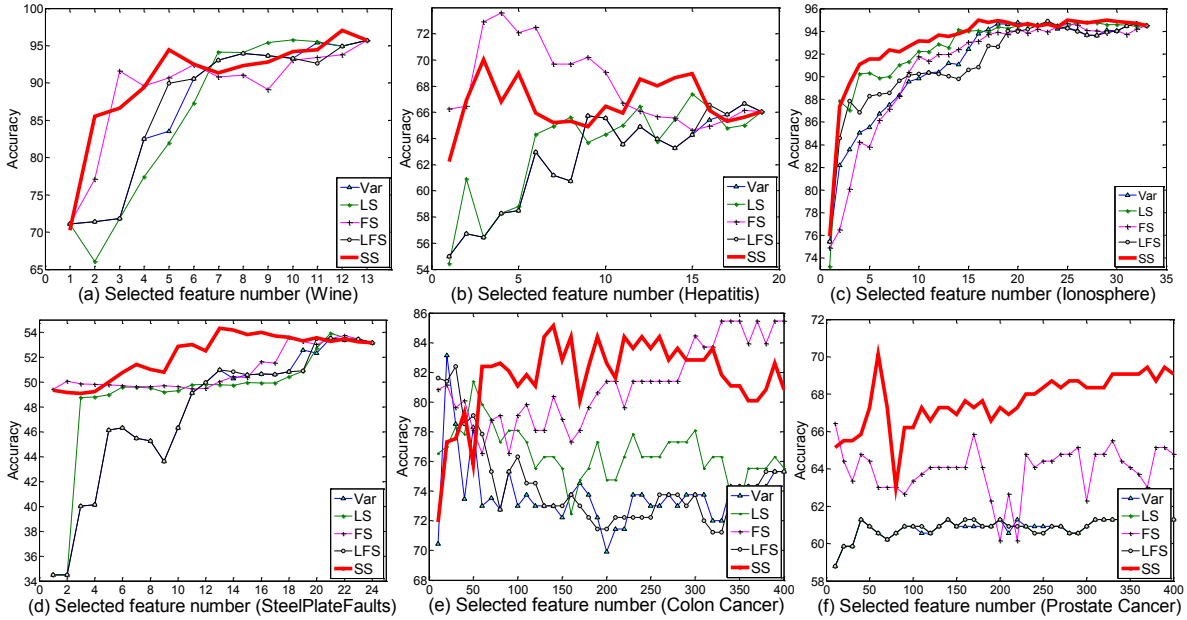

Fig. 1. Accuracy vs. selected feature number

5. Conclusion

In this paper, we propose a new graph-based feature selection algorithm based on sparsity preserving power, aiming to preserve l_1 graph structure. Extensive experimental results have validated the effectiveness of our proposed method. In the current work, we only consider using the l_1 graph for graph construction. In fact, besides the l_1 graph, there may be other kinds of graphs (e.g., l_2 graph). It's interesting to investigate whether using other kinds of graphs can also lead to performance improvement, which will be our future work.

6. Acknowledgment

This work is supported by the National Natural Science Foundation of China under the grant Nos. 60875030 and 61072148.

References

[1] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3:1157-1182, 2003.

[2] L. Yu and H. Liu. Feature Selection for High-dimensional Data: AFast Correlation-based Filter Solution. in *Proceedings of the 20th International Conferences on Machine Learning*,601-608, 2003.

[3] C. M. Bishop. *Neural Networks for Pattern Recognition*.Oxford University Press, Oxford, 1995.

[4] X. He, et al. Laplacian Score for Feature Selection. in *Advances in Neural Information Processing Systems*, 2005.

[5] D. Q. Zhang, et al. Constraint Score: A New Filter Method for Feature Selection with Pairwise Constraints.*Pattern Recognition*, 41:1440-1451, 2008.

[6] B. Cheng, et al. Learning with L1-graph for Image Analysis. *IEEE Transactions on Image Process*, 19:858-66, 2010.

[7] S. Yan, et al. Graph Embedding and Extensions: AGeneral Framework for Dimensionality Reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 29:40-51, Jan 2007.

[8] J. Wright, et al. Robust Face Recognition via Sparse Representation.*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31: 210-27, Feb 2009.

[9] S. S. B. Chen, et al. Atomic Decomposition by Basis Pursuit. *Siam Review*, 43:129-159, Mar 2001.

[10]A. Y. Yang, et al. Feature Selection in Face Recognition: A Sparse Representation Perspective. 2007.

[11]Q. Cheng, et al. The Fisher-Markov Selector: Fast Selecting Maximally Separable Feature Subset for Multiclass Classification with Applications to High-Dimensional Data.*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1217-1233, 2011.