Iterative Laplacian Score for Feature Selection

Linling Zhu, Linsong Miao, and Daoqiang Zhang

College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China
{zhulinling,linsongmiao,dqzhang}@nuaa.edu.cn

Abstract. Laplacian Score (LS) is a popular feature ranking based feature selection method both supervised and unsupervised. In this paper, we propose an improved LS method called Iterative Laplacian Score (IterativeLS), based on iteratively updating the nearest neighborhood graph for evaluating the importance of a feature by its locality preserving ability. Compared with LS, the key idea of IterativeLS is to gradually improve the nearest neighbor graph by discarding the least relevant features at each iteration. Experimental results on several high dimensional data sets demonstrate the effectiveness of our proposed method.

Keywords: feature selection, Iterative Laplacian score, Laplacian score, locality preserving.

1 Introduction

Feature selection has become an important preprocessing step in machine learning and data mining due to the rapid accumulation of high-dimensional data such as gene expression microarrays, digital images and text data. Feature selection is quite effective in reducing dimensionality, removing irrelevant and redundant features. According to [1], feature selection methods can be simply classified into "wrapper" methods and "filter" methods.

The wrapper methods need a predetermined learning algorithm to evaluate the features, while the filter methods are totally independent of any learning algorithm. They evaluate the features by examining the intrinsic properties of the data. Though the wrapper methods may usually outperform the filter methods in terms of accuracy, the later is more efficient in computation [8]. When confronted with huge number of features, the filter methods are more attractive. In this paper, we are particularly interested in the filter feature selection methods. Data variance, Pearson correlation coefficients, Fisher score and Kolmogorov-Smirnov test are all typical filter methods.

From another perspective, feature selection methods can be divided into supervised and unsupervised methods [2, 3]. Supervised methods evaluate the feature importance by the correlation of the feature and class label. Pearson correlation coefficients, Fisher score and Kolmogorov-Smirnov test are supervised method. Unsupervised

C.-L. Liu, C. Zhang, and L. Wang (Eds.): CCPR 2012, CCIS 321, pp. 80–87, 2012.

methods evaluate the importance of the feature by the capability of keep certain properties of the data, such as variance or locality preserving ability [4, 5]. Data variance, as an unsupervised method, is perhaps the simplest criterion for selecting representative features. Data variance projects the data points along the dimensions of maximum variances, finding the features useful for representing the data. While Fisher score [6] selects the most important features for discrimination.

Laplacian Score (LS) is a recently proposed feature selection method, which can be used in either supervised or unsupervised scenarios [9]. It seeks features which best reflect the underlying manifold structure. Laplacian Score has been proved effective and efficient compared with data variance and Fisher score. In this paper, we proposed an improved Laplacian score algorithm called Iterative Laplacian score (IterativeLS). A series of experiments have been implemented on several high dimensional UCI datasets and face data sets to compare our algorithm with Laplacian score and other feature selection algorithms, e.g., variance,etc. The experimental results show that our algorithm outperforms original Laplacian score algorithm on both classification and clustering tasks.

The rest of this paper is organized as follows. Section 2 briefly introduces the Laplacian score algorithm as background. In Section 3 we present the proposed IterativeLS algorithm in detail. The experimental results are reported in section 4. Finally, Section 5 concludes this paper.

2 Laplacian Score

LS constructs a nearest neighbor graph to model the local structure and then selects those features which best respect this graph structure.

Suppose we have *m* data points and each point has *n* features. Let L_r denote the Laplacian score of the r -th feature, $r = 1,\ldots, n$. Let f_{ri} denote the i -th sample of the r -th feature, $i = 1,\ldots, m$. LS algorithm is described as follows [4]:

- Step 1: Constructing a nearest neighbor graph G.

  The i -th node corresponds to the i -th data point x_i. We put an edge between nodes *i* and *j* if x_i and x_j are “close”. We define “close” in the way that, x_i is among *k* nearest neighbors of x_j or x_j is among *k* nearest neighbors of x_i.

- Step 2: Choosing the weights.

  Put $w\_ij = e^{-\frac{\|x\_i - x\_j\|^2}{t}}$ if nodes *i* and *j* are connected, otherwise $w\_ij = 0$, where *t* is a constant.

- Step 3: Computing the Laplacian score.

  Define $\mathbf{f}_r = [f_{ri}, f_{r2},\ldots, f_{rm}]^T$, the Laplacian score of the r -th feature L_r is computed as

$$L_r = \frac{\tilde{\mathbf{f}}_r^T L \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T D \tilde{\mathbf{f}}_r},$$

Where D is a diagonal matrix with $D\_ii = \sum_j W_{ji}$, and $L = D - W$, $\tilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathbf{f}_r^T D \mathbf{1}}{\mathbf{1}^T D \mathbf{1}} \mathbf{1}$, $\mathbf{1} = [1, \ldots, 1]^T$.

Laplacian score algorithm chooses those features with lowest scores.

3 Iterative Laplacian Score Algorithm

As in LS, in the proposed IterativeLS method, we also evaluate the feature importance according to the locality preserving ability. Both algorithms are based on the observation that, data points from the same class probably are close to each other. The key idea of IterativeLS is that, at each iteration we discard the least relevant features according to their current Laplacian scores. As a result, IterativeLS improves the nearest neighbor graph every time, and selects feature subsets which respect the structure of the underlying manifold better than original Laplacian score. We develop two versions of IterativeLS, and describe them in detail as follows.

- IterativeLS-1

```
Input: data set $X$ with $n$ features, selected feature number $s$
Output: $s$ ranked feature list
Repeat:
  (1) Construct nearest neighbor graph for data $X$;
  (2) Compute Laplacian score for the features in $X$ using LS algorithm;
  (3) Rank the features according to their Laplacian scores in ascending order;
  (4) Discard the last $d$ features and update data $X$ consisting of only the rest
      features;
Until: no more than $s$ features are left.
```

- IterativeLS-2

```
Input: data set $X$ with $n$ features, selected feature number $s$
Output: $s$ ranked feature list
Repeat:
  (1) Construct nearest neighbor graph for data $X$;
  (2) Compute Laplacian score for all the features in initial $X$ using LS algorithm;
  (3) Rank the features according to their Laplacian scores in ascending order;
  (4) Discard the last $d$ features and update data $X$ consisting of only the rest
      features;

Until: no more than $s$ features are left.
```

Obviously, at each iteration, both IterativeLS-1 and IterativeLS-2 reconstruct the nearest neighbor graph for data X only using the rest features. The difference between IterativeLS-1 and IterativeLS-2 is that, the former computes Laplacian scores for the features excluding those discarded ones, but the later computes Laplacian scores for all the features. That means in algorithm IterativeLS-2 those features discarded before could be reselected in later iteration, but not the same in algorithm IterativeLS-1.

Experimental results show that, IterativeLS-1 and IterativeLS-2 almost have the same results, so in the following sections we only show the results of IterativeLS-1, and call it as IterativeLS for convenience.

4 Experiments

In this section, we investigate the performance of our proposed IterativeLS algorithm. Both supervised (classification) and unsupervised (clustering) experiments are carried out to demonstrate the effectiveness of our proposed algorithm. Experiments are conducted on several high-dimensional data sets including UCI data and ORL face database. In our experiments, we choose k=5 to construct the nearest neighbor graph.

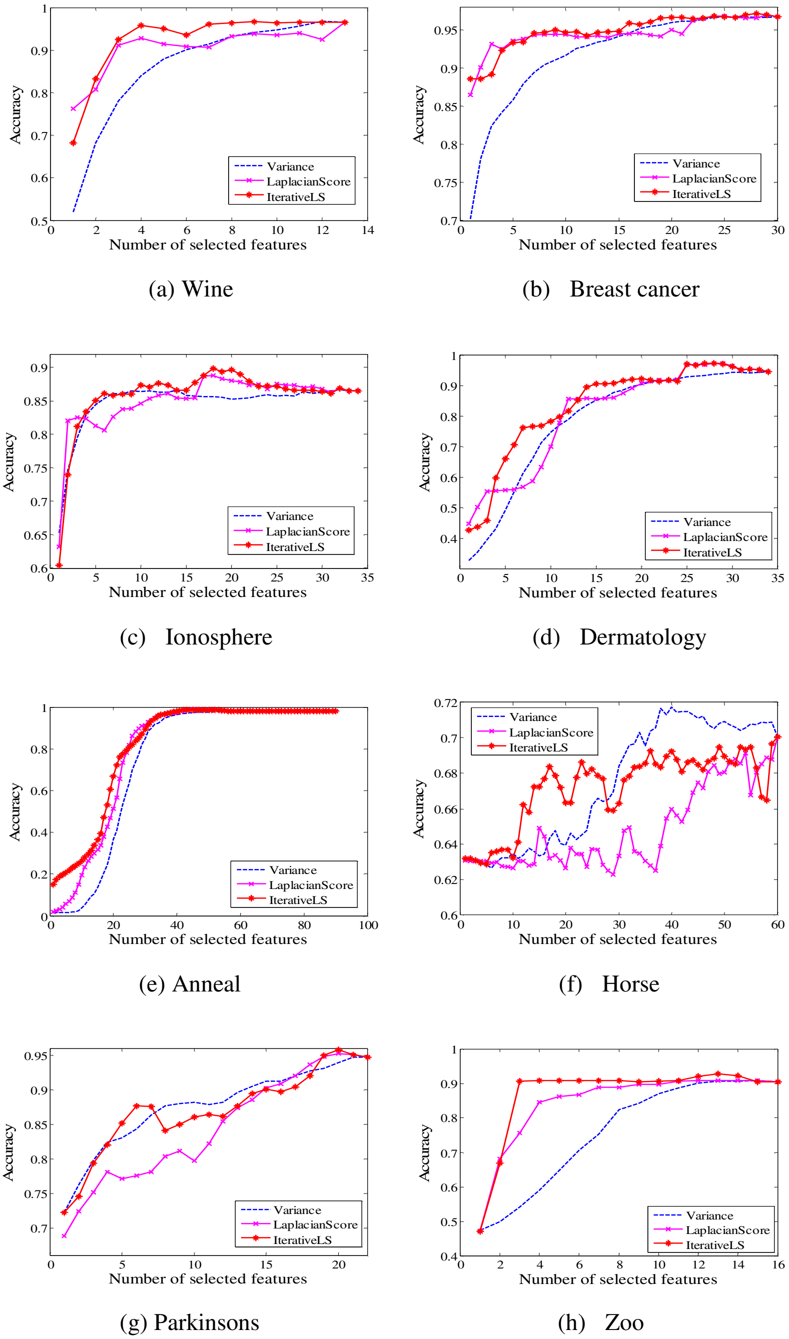
4.1 Results on UCI Data Sets

In classification experiments, we compare IterativeLS with variance and LS. We test performances of different algorithms using 10 times 10-fold cross validation, and choose the nearest neighborhood (NN) with Euclidean distance as classifier. The performances of all methods are measured by the classification accuracy using selected features on test datasets. In IterativeLS algorithm, we discard one feature each iteration (i.e., $d = 1$). The statistics of the UCI data sets are given in table 1. Fig.1 shows the curves of three algorithms with accuracy vs. different numbers of selected features.

Table 1. statistics of the UCI data sets

Data set	Sample	Feature	Class
Wine	178	13	3
Breast cancer	198	33	2
Ionosphere	351	34	2
Dermatology	358	34	6
Anneal	898	38	5
Horse	368	27	2
Parkinsons	57	16	2
Zoo	101	16	7

From Fig.1 we can see that, the performance of IterativeLS is almost always better than LS and variance. Specifically, IterativeLS significantly outperforms LS when selected feature number is much lower than the original data dimension, such as from 5 to 10.



(a) Wine

(b) Breast cancer

(c) Ionosphere

(d) Dermatology

(e) Anneal

(f) Horse

(g) Parkinsons

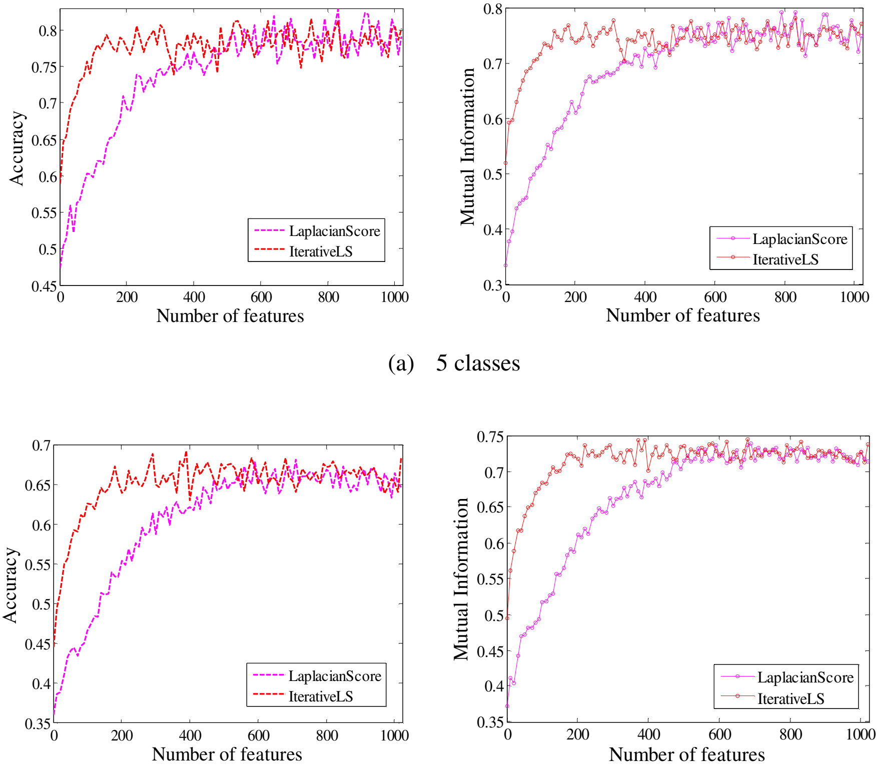
(h) Zoo

Fig. 1. Accuracy vs. different number of selected features on eight UCI data sets

4.2 Face Clustering on ORL Data Set

In this section, we applied IterativeLS to face clustering on ORI face database with comparison to LS. The ORL database contains images from 40 individuals, 10 different images for each individual. We crop the original images of ORL to size of 32x32, and use the original pixel intensity as features, so a face image is represented by a 1024 dimensional feature vector.

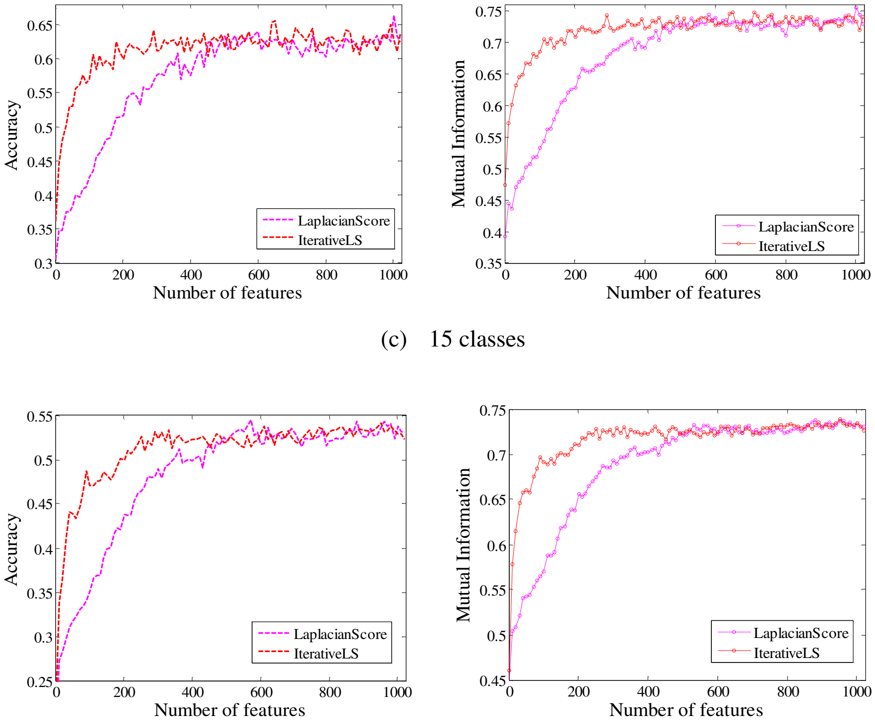
In our experiment, several tests are performed with different number of clusters (i.e.,5,10,15,40). LS and IterativeLS are used to select features, and then K-means is performed in the selected feature subspace. K-means is repeated 10 times with different initializations and recorded the best result. This process is repeated 20 times, and the average performance is computed. We evaluate the clustering performance of different algorithm using two measurements, i.e. the accuracy and the normalized mutual information metric [7]. In IterativeLS, we set the number of discarded features to be 5 (i.e.,*d*=5). Fig.2 shows the results with different number of selected features.



(a) 5 classes

(b) 10 classes

Fig. 2. Clustering results with different number of selected features



(c) 15 classes

(d) 40 classes

Fig. 2. (*Continued*)

As can be seen from Fig.2, our proposed algorithm is superior in performance compared to original LS. Take a closer look at Fig.2, we can find out that, when selected feature number is smaller than 400, IterativeLS is significantly better than LS. All these experimental results demonstrate that our proposed algorithm is more effective than LS.

4.3 Further Discussion

In order to investigate the influence of discarded feature number (i.e., *d*) at each iteration, we conducted experiment on face data ORL with different number of discarded feature. In our experiment, we randomly select 2, 5 and 8 images for each individual as train data, and the rest of images as test data respectively. Then IterativeLS and LS are applied on the face data to select features, and nearest neighborhood classifier is used to measure the performance of the algorithms. Fig.3 plots the curves of accuracy vs. different number of selected features.

From Fig.3 we can draw the conclusion that, when the dimension of original data is large, the number of discarded features at each iteration does not matter much to the final results.

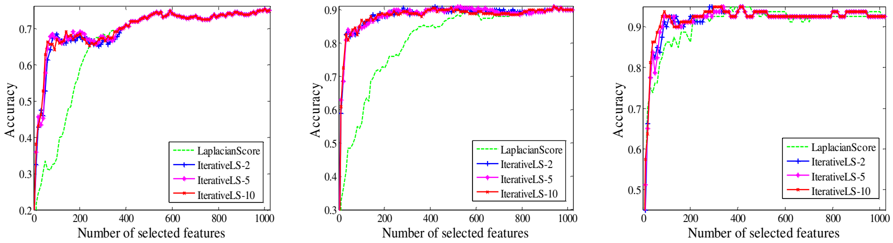


Fig. 3. Accuracy vs. different number of selected features as different discarded feature number

5 Conclusion

In this paper, we proposed a new Iterative Laplacian Score (IterativeLS) method for feature selection, based on evaluating the feature importance according to the locality preserving ability for the underlying manifold structure. In each iteration of IterativeLS, we discard the least significant features according to corresponding current scores of features, and reconstruct the nearest neighbor graph with the rest of the features. The experimental results on both UCI data sets and face databases demonstrate the effectiveness of our proposed algorithm. Particularly, when very few features are selected, IterativeLS can achieve significantly better performance than conventional Laplacian Score (LS).

References

1. Belkin, M., Niyogi, P.: Laplacian Egienmaps and Spectral Techniques for Embedding and Clustering. In: Advances in Neural Information Processing System, vol. 14 (2001)
2. Dy, J.G., Brodley, C.E., Kak, A.C., Broderick, L.S., Aisen, A.M.: Unsupervised feature selection applied to content-based retrieval of lung image. IEEE Trans. Pattern Anal. Mach Intell. 25, 373–378 (2003)
3. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised feature learning. J. March. Learning Res. 5, 845–889 (2004)
4. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Advances in Neural Information Processing System, vol. 17. MIT press, Cambridge (2005)
5. He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing System, vol. 16. MIT Press, Cambridge (2004)
6. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
7. Xu, W., Liu, X., Gong, Y.: Document Clustering Based on Non-negative Matrix Factorization. In: ACM SIGIR Conference on Information Retrieval (2003)
8. Sun, Y., Todorovic, S., Goodison, S.: Local-learning-Based Feature Selection for High-Dimensional Data Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligent 32(9) (2010)
9. Xu, J., Man, H.: Dictionary Learning Based on Laplacian Score in Sparse Coding. In: Perner, P. (ed.) MLDM 2011. LNCS (LNAI), vol. 6871, pp. 253–264. Springer, Heidelberg (2011)