Eigenvalue Sensitive Feature Selection

Yi Jiang JIANGYI5@STUDENT.SYSU.EDU.CN
Jiangtao Ren ISSRJT@MAIL.SYSU.EDU.CN
Sun Yat-sen University, Guangzhou, Guangdong, 510006, P.R.China

Abstract

In recent years, some spectral feature selection methods are proposed to choose those features with high power of preserving sample similarity. However, when there exist lots of irrelevant or noisy features in data, the similarity matrix constructed from all the un-weighted features may be not reliable, which then misleads existing spectral feature selection methods to select 'wrong' features. To solve this problem, we propose that feature importance should be evaluated according to their impacts on similarity matrix, which means features with high impacts on similarity matrix should be chosen as important ones. Since graph Laplacian(Luxburg, 2007) is defined on the similarity matrix, then the impact of each feature on similarity matrix can be reflected on the change of graph Laplacian, especially on its eigen-system. Based on this point of view, we propose an Eigenvalue Sensitive Criteria (EVSC) for feature selection, which aims at seeking those features with high impact on graph Laplacian's eigenvalues. Empirical analysis demonstrates our proposed method outperforms some traditional spectral feature selection methods.

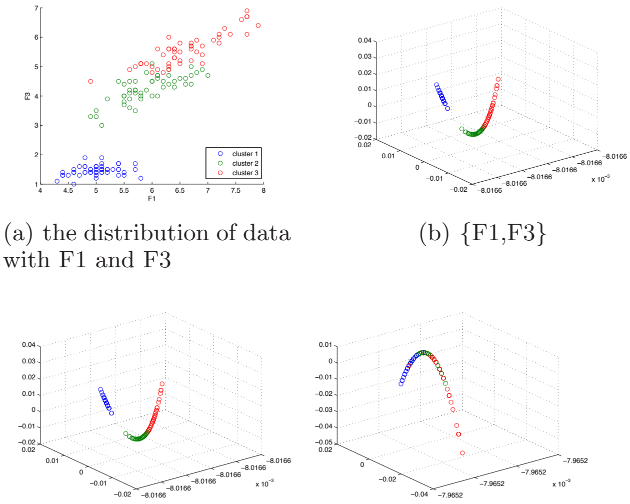
1. Introduction

In recent years, many researchers attempt to employ the spectrum of graph to design new feature selection methods, termed spectral feature selection(Zhao, 2007). According to different character of them, existing methods can be divided into two types. The first type performs feature selection based on certain evaluation criteria, which is the function of the eigen-system of graph Laplacian(Zhao, 2007). The typical ones include Laplacian score(He, 2005), trace ratio(Nie, 2008), and SPEC(Zhao, 2007). For the second type, the feature selection problem will be transformed into the regression problem. Both MRSF(Zhao, 2010) and MCFS(Cai, 2010) belong to this type. It is clear that they all depend on the eigen-system of graph Laplacian, which is defined on similarity matrix. In other words, the performance of them is determined by the similarity matrix. However, when there exist lots of irrelevant or noisy features in data, the accuracy of similarity matrix can't be guaranteed any more, which negatively affects the effectiveness of existing spectral feature selection methods. To solve this problem, we evaluate the significance of each feature based on its influence on the eigen-system of graph Laplacian.

Here, we would like to demonstrate the impacts of different features on the normalized graph Laplacian L_{rw}(Luxburg, 2007) with Iris dataset(He, 2005) using only two features: sepal length(F1) and petal length(F3), whose data distribution in the original input space with F1 and F3 is plotted in Figure 1(a). Then three normalized graph Laplacian are constructed using feature subset {F1,F3}, {F3} and {F1} respectively. Then with each graph Laplacian, the three eigenvectors with respect to the three smallest eigenvalues are selected and plotted in Figure 1(b), 1(c) and 1(d). It is clear that Figure 1(c) is similar with Figure 1(b), but Figure 1(d) is evidently different from Figure 1(b). This phenomena means the eigen-system of L_{rw} doesn't change by the elimination of F1, but changes dramatically by the elimination of F3, which indicates F3 is more important than F1.

The above example illustrates the impacts of different features on graph Laplacian are greatly different. Motivated by this viewpoint, we propose that feature importance should be evaluated according to its impact on graph Laplacian. Since graph Laplacian is highly related with its eigenvalues, we choose to perform the impact analysis of each feature on the eigenvalues of graph Laplacian. Firstly, we introduce the weighted

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.



(a) the distribution of data with F1 and F3

(b) {F1,F3}

(c) {F3}

(d) {F1}

Figure 1. (a) presents the 2-D visualization of Iris dataset. (b) ,(c) and (d) present the 3-D visualization of eigenvectors according to the graph Laplacian constructed by feature subset {F1,F3}, {F3} and {F1}, respectively.

similarity matrix into graph Laplacian, in which the t-th feature is assigned a weight w_t. Then, based on the weighted graph Laplacian, we can derive the derivative of the r-th eigenvalue of graph Laplacian(for example $\lambda(w)_r$) with respect to the t-th feature weight, that is $\frac{\partial\lambda(w)_r}{\partial w_t}$. After obtaining $\frac{\partial\lambda(w)_r}{\partial w_t}$, we can approximate the change of $\lambda(w)_r$ in response to the change of w_t from 1 to 0 while keeping all the other feature weights as 1, namely the corresponding differential $\Delta\lambda(\mathbf{1},t)_r$. $\Delta\lambda(\mathbf{1},t)_r$ represents the impact to the r-th eigenvalue of graph Laplacian when eliminating the t-th feature from data, which is the core idea of our method.

In this work, we propose an eigenvalue sensitive feature selection algorithm, which aims at selecting those features with high impacts on graph Laplacian. Extensive experiment results over five real-world datasets demonstrate the superiority of our method compared with traditional spectral feature selection methods.

2. Eigenvalue Sensitive Feature Selection

2.1. Spectral Feature Selection with Un-Weighted Similarity Measure

In this paper, we use X to denote a data set of n instances, and $X = (x^1, x^2, ..., x^n) = (f_1^T, f_2^T, ..., f_K^T)^T \in R^{K\times n}$, where both x_t^s and f_{ts} denote the t-th feature of instance x^s. There are several kinds of general similarity measures such as Dot-product, Square Euclidean and RBF functions, due to the limit of space, we only use RBF function as the similarity measure in this paper: $S_{ij} = e^{-\frac{\|x^i - x^j\|^2}{2\delta^2}}$. In fact, our method can be easily extended to other similarity measures.

We start from Laplacian score(He, 2005), which can be considered as the function of the eigen-system of $L_{sym} = D^{-\frac{1}{2}}SD^{-\frac{1}{2}}$(Zhao, 2007), where $D = diag(S\mathbf{1})$ and $\mathbf{1} = (1, ..., 1)^T$. The Laplacian score of the t-th feature can be also computed as(Zhao, 2007):

$$L_t = \frac{\sum_{r=2}^{n}\alpha_{tr}{}^2\lambda_r}{\sum_{r=2}^{n}\alpha_{tr}{}^2}$$

where q_r and λ_r $(1\leq r\leq n)$ are the eigenvector and eigenvalue of L_{sym}, and $\alpha_{tr} = cos\theta_{tr}$ where θ_{tr} is the angle between feature vector f_t and eigenvector q_r. The above equation shows that Laplacian score is dependent on the similarity matrix S, but S is computed with all features in the same importance. But as we know, the importance of features are different, that is why we perform feature selection. So it is nature to introduce the weighted similarity measure.

2.2. Extension to Weighted Similarity Measure

Based on the above discussion, it is clear that the eigen-system of graph Laplacian must be established on the weighted similarity measure. When we obtain the weight vector $w = (w_1, w_2, ..., w_K)^T$, the weighted RBF function can be defined as: $S(w)_{ij} = e^{-\frac{\|w\cdot(x^i - x^j)\|^2}{2\delta^2}}$. Based on the appropriate weighted similarity matrix, the corresponding eigen-system can correctly reflect the structure of sample distribution, which can guarantee the success of spectral feature selection methods. However, the choice of appropriate feature weight vector is still difficult. Hence, we can take the opposite perspective and focus on the analysis of the impact on the eigen-system of graph Laplacian by the change of each feature weight w_t, specifically the change of w_t from 1 to 0. In other words, we want to evaluate the importance of each feature by its impact on the eigen-system of graph Laplacian when it is eliminated from data. Motivated by this point of view, we propose the Eigenvalue Sensitive Criterion(EVSC), which will be discussed in detail in the next section.

2.3. Eigenvalue Sensitive Criterion(EVSC)

In this section, we will derive the criteria which evaluates the impact on eigen-system of graph Laplacian in terms of each feature. Let's start from normalized graph Laplacian(Belkin, 2001), $L(w)_{rw} =$

$D(w)^{-1}L(w)$, which is obtained by solving the following generalized eigen-problem(Belkin, 2001):

$$L(w)q(w)_r = \lambda(w)_r D(w)q(w)_r \tag{1}$$

where D(w)=diag(S(w)**1**) and L(w)=D(w)-S(w) denote the weighted degree matrix and the weighted graph Laplacian, respectively. $q(w)_r$ and $\lambda(w)_r$ denote the r-th eigenvector and the r-th eigenvalue of $L(w)_{rw}$ respectively.

Firstly, we present the following Proposition 1 for calculating the derivative of $\lambda(w)_r$ with respect to the t-th feature weight w_t, that is $\frac{\partial\lambda(w)_r}{\partial w_t}$.

Proposition 1. *In equation (1), the calculation of $\frac{\partial\lambda(w)_r}{\partial w_t}$ is formulated as:*

$$\frac{\partial\lambda(w)_r}{\partial w_t} = q(w)_r{}^T(\frac{\partial L(w)}{\partial w_t} - \lambda(w)_r\frac{\partial D(w)}{\partial w_t})q(w)_r \tag{2}$$

Proof. For writing convenience, here we use L, D, q_r and λ_r to denote L(w), D(w), $q(w)_r$ and $\lambda(w)_r$, respectively. By differentiating both sides of $Lq_r = \lambda_r Dq_r$ with respect to w_t, we can derive the following equation:

$$\frac{\partial L}{\partial w_t}q_r + L\frac{\partial q_r}{\partial w_t} = \frac{\partial \lambda_r}{\partial w_t}Dq_r + \lambda_r\frac{\partial D}{\partial w_t}q_r + \lambda_r D\frac{\partial q_r}{\partial w_t} \tag{3}$$

Left multiply both sides of (3) by q_r^T:

$$q_r^T\frac{\partial L}{\partial w_t}q_r + q_r^T L\frac{\partial q_r}{\partial w_t}$$

$$= \frac{\partial \lambda_r}{\partial w_t}q_r^T Dq_r + \lambda_r q_r^T\frac{\partial D}{\partial w_t}q_r + \lambda_r q_r^T D\frac{\partial q_r}{\partial w_t} \tag{4}$$

Since both L and D are symmetric, we have

$$q_r^T L\frac{\partial q_r}{\partial w_t} = \lambda_r q_r^T D\frac{\partial q_r}{\partial w_t} \tag{5}$$

Thus, by (4) and (5), we can derive:

$$q_r^T\frac{\partial L}{\partial w_t}q_r = \frac{\partial \lambda_r}{\partial w_t}q_r^T Dq_r + \lambda_r q_r^T\frac{\partial D}{\partial w_t}q_r \tag{6}$$

that is,

$$\frac{\partial \lambda_r}{\partial w_t} = \frac{q_r^T(\frac{\partial L}{\partial w_t} - \lambda_r\frac{\partial D}{\partial w_t})q_r}{q_r^T Dq_r} \tag{7}$$

Since $q_r^T Dq_r = 1$, then the equation (7) can also be expressed as:

$$\frac{\partial \lambda_r}{\partial w_t} = q_r^T(\frac{\partial L}{\partial w_t} - \lambda_r\frac{\partial D}{\partial w_t})q_r \tag{8}$$

□

Secondly, since the formula (2) involves $\frac{\partial L(w)}{\partial w_t}$ and $\frac{\partial D(w)}{\partial w_t}$, it is necessary to derive the formulation of them. For writing convenience, we will use the following two notations(Ning, 2010): u_{ij} and v_{ij}. u_{ij} is a column vector with only two nonzero element: the i-th and j-th element equal to 1 and -1 respectively; v_{ij} is a column vector with i-th and j-th elements equal to 1 and other elements equal to 0. For the limit of space, the proposition 2 only involves the RBF function. $S(w)_{ij}$, $D(w)_{ij}$ and $L(w)_{ij}$ are the i-th row and j-th column element of S(w), D(w) and L(w), respectively.

Proposition 2. *Assuming $S(w)_{ij} = e^{-\frac{\|w\cdot(x^i-x^j)\|^2}{2\delta^2}}$, the calculation of $\frac{\partial D(w)}{\partial w_t}$ and $\frac{\partial L(w)}{\partial w_t}$ can be formulated as:*

$$\frac{\partial D(w)}{\partial w_t} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\boldsymbol{g(i,j,t)}diag(v_{ij}) \tag{9}$$

and

$$\frac{\partial L(w)}{\partial w_t} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\boldsymbol{g(i,j,t)}u_{ij}u_{ij}^T \tag{10}$$

respectively.

where $\boldsymbol{g(i,j,t)}=-w_t\frac{(x_t^i-x_t^j)^2}{\delta^2}S(w)_{ij}$.

Proof. ∀ i and j, $S(w)_{ij} = e^{-\frac{\sum_{t=1}^{n} w_t^2(x_t^i-x_t^j)^2}{2\delta^2}}$, then

$$\frac{\partial S(w)_{ij}}{\partial w_t} = -w_t\frac{(x_t^i - x_t^j)^2}{\delta^2}S(w)_{ij} \tag{11}$$

Since D(w)= diag(S(w)**1**), if i=j, $D(w)_{ii} = \sum_{h=1}^{n}S(w)_{ih}$, then

$$\frac{\partial D(w)_{ii}}{\partial w_t} = \sum_{h=1}^{n}\frac{\partial S(w)_{ih}}{\partial w_t} = -\sum_{h=1}^{n}w_t\frac{(x_t^i - x_t^h)^2}{\delta^2}S(w)_{ih}$$

otherwise, $D(w)_{ij} = 0$, then

$$\frac{\partial D(w)_{ij}}{\partial w_t} = 0$$

Since L(w)=D(w)-S(w), then

$$\frac{\partial L(w)_{ij}}{\partial w_t} = \frac{\partial D(w)_{ij}}{\partial w_t} - \frac{\partial S(w)_{ij}}{\partial w_t}$$

By using u_{ij} and v_{ij}, $\frac{\partial D(w)}{\partial w_t}$ and $\frac{\partial L(w)}{\partial w_t}$ can be finally expressed as:

$$\frac{\partial D(w)}{\partial w_t} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{\partial S(w)_{ij}}{\partial w_t}diag(v_{ij})$$

and

$$\frac{\partial L(w)}{\partial w_t} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{\partial S(w)_{ij}}{\partial w_t} u_{ij} u_{ij}^T$$

, respectively. □

Thirdly, based on (9) and (10), the calculation of $\frac{\partial \lambda(w)_r}{\partial w_t}$ can be expressed as:

$$\frac{\partial \lambda(w)_r}{\partial w_t} = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathbf{g(i,j,t)}\{(q(w)_{ri} - q(w)_{rj})^2\}$$

$$- \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathbf{g(i,j,t)}\{\lambda(w)_r(q(w)_{ri}^2 + q(w)_{rj}^2)\} \quad (12)$$

where **g(i,j,t)** is defined in Proposition 2, and $q(w)_{ri}$ and $q(w)_{rj}$ denotes the i-th element and the j-th element of the r-th eigenvector $q(w)_r$, respectively. As can be seen in (12), the $\frac{\partial \lambda(w)_r}{\partial w_t}$ is actually the function of w which represents all feature weights(including w_t).

Finally, according to different feature weight vector w(including w_t), we can derive the derivative of $\lambda(w)_r$ with respect to w_t from (12), that is, $\frac{\partial \lambda(w)_r}{\partial w_t}$. Loosely speaking, this derivative can be thought of as how much the r-th eigenvalue is changing in response to the change of the t-th feature weight. In general, it is hard to obtain the accurate feature weight vector w. Thus the calculation of (12) is also intractable. However, we can consider the differential $d\lambda(w)_r$ of $\lambda(w)_r$ at w_t, where $w_k(k{\neq}t \ and \ 1{\leq}k{\leq}n)$ is constant and $w_t{\in}[0,1]$, which is formally expressed as:

$$d\lambda(w)_r = (\frac{\partial \lambda(w)_r}{\partial w_t})dw_t \quad (13)$$

Specifically, when w=$\mathbf{1}$=$(1,...,1)^T$, and dw_t=1-0=1, that is, the t-th feature is eliminated from data, the corresponding change of the r-th eigenvalue (denoted $\Delta\lambda(\mathbf{1},t)_r$) can be approximated as follows:

$$\Delta\lambda(\mathbf{1},t)_r{\approx}(\frac{\partial \lambda(w)_r}{\partial w_t}|_{w=\mathbf{1}}) \quad (14)$$

The equation (14) just reveals the change of the r-th eigenvalue with respect to the elimination of the t-th feature, which is the core idea of this paper. Therefore, we can employ the equation (14) to evaluate the importance of the t-th feature for the r-th eigenvalue of L_{rw}. To reflect the influence of the t-th feature on all n eigenvalues of L_{rw}, the eigenvalue sensitive criterion(EVSC) can be defined as:

$$EVSC(t) = \sum_{r=1}^{n}|\Delta\lambda(\mathbf{1},t)_r| \quad (15)$$

3. Algorithm for Eigenvalue Sensitive Feature Selection(ESFS)

For saving the space of paper, here we only give the eigenvalue sensitive feature selection algorithm based on the RBF function, described in **Algorithm 1**.

The time complexity of **ESFS** can be analyzed as follows: (a) In step 1 and 2, we need $O(n^2K)$ operations to build S, D and L; (b) In step 3, we need $O(n^3)$ operations to get the eigenvalues and eigenvectors of graph Laplacian by Lanczos algorithm; (c) In step 4, we need $O(n^2K)$ operations to calculate the EVSC score for all features; (d) In step 5, the top m features can be found within O(KlogK). Thus, the overall time complexity of **ESFS** is $MAX(n^3, n^2K, KlogK)$.

Algorithm 1 Eigenvalue Sensitive Feature Selection

```
Input: data set X, feature number $m$
Output: feature subset $F_m$
1. Construct the similarity matrix S with RBF function
2. Build L and D based on S
3. Calculate the eigen-system $(\lambda_r, q_r), 1{\leq}r{\leq}n$, from $Lq_r = \lambda_r Dq_r$
for $t = 1$ to $K$ do
    4. Calculate the EVSC of the t-th feature according to (15).
end for
5. Rank the features decreasingly according to the value of EVSC and select the leading m features, that is $F_m = \{F_{K_1}, ..., F_{K_m}\}$
6. return $F_m$
```

4. Empirical Analysis

In this section, we will empirically analyze the performance of our proposed algorithm EVSC compared with Laplacian Score and SPEC.

4.1. Dataset decription

Five data sets[1] are used for experiments, which are briefly described in table 1.

Table 1: Statistics of the five data sets

Data Set	*Instance*	*Feature*	*Class*
PIX10P(PIX)	100	10000	10
ORL10P(ORL)	100	10304	10
GLA-BRA(GLA)	180	4915	4
CLL-SUB(CLL)	111	11340	3
TOX-171(TOX)	171	5748	4

[1] http://featureselection.asu.edu/datasets.php

4.2. Evaluation criterions

For investigating the performance of our methods, several tests are performed on the following two evaluation metrics:

Clustering Accuracy(ACC) The accuracy of clustering(He, 2005) is defined as comparing the predefined label(c(s)) and the obtained label(km(s)) by k-means clustering of each sample s:

$$ACC = \frac{\sum_{h=1}^{n} \delta(c(h), km(h))}{n}$$

where $\delta(c(s), km(s)) = 1$ only if c(s)=km(s), otherwise 0. Since the starting points of k-means algorithm are randomly chosen each time, here we apply the k-means algorithm to the same data set with the same feature subset 10 times and record the best result.

Jaccard Score(JAS) As (Zhao, 2010), the Jaccard Score is used to evaluate the ability of selected feature subset in preserving sample structure, which is computed by:

$$JAC = \frac{1}{n}\sum_{h=1}^{n} \frac{NB(h, m, S_F) \cap NB(h, m, S)}{NB(h, m, S_F) \cup NB(h, m, S)}$$

where S_F is the similarity matrix on the selected features while S is the original similarity matrix, and $NB(s, m, S)$ donates the m nearest neighbors of s-th sample according to S. The similarity matrix is computed by using inner product. A high Jaccard Score indicates that sample similarity are well preserved.

4.3. Experiment setup

In the experiments, two representative spectral feature selection algorithms are chosen as base lines: Laplacian Score(He, 2005)[2] and SPEC(Zhao, 2007)[3]. For experimental convenience, we only choose **RBF function** as similarity measure, whose parameter is determined by cross-validation. For each data set, we apply Laplacian Score, SPEC and EVSC to calculate the corresponding feature score. For EVSC, all features are ranked decreasingly, while all features are ranked increasingly for the two others. After such preparation, we calculate the corresponding Clustering Accuracy and Jaccard Score of each data set with the leading 100, 200, ...,1900 features.

4.4. Analysis of the result

Clustering Accuracy Figure 2(a-e) show the curves of Clustering Accuracy versus the number of selected features on five different data sets based on Laplacian Score, SPEC and EVSC. Figure 2(g) shows the average Clustering Accuracy over all five data sets versus each feature number. As we can see, our proposed EVSC algorithm outperforms Laplacian Score and SPEC on each data set. Especially on ORL10P and PIX10P, the clustering error can be nearly zero by using around 1300 features.

Table 2 reports the detailed average accuracy over all chosen feature number on each data set based on Laplacian Score, SPEC and EVSC. The last row of it records the average clustering performance over all five data sets and all chosen feature numbers for each method. Compared with Laplacian Score, our method achieves 15.5%, 13.5%, 13.01%, 40% and 17.5% relative improvements on CLL, GLA, ORL10P, PIX10P and TOX respectively, while 15.1%, 30.67%, 25.67%, 12.6% and 17.5% relative improvements compared with SPEC.

Jaccard Score Figure 3(a-e) show the curves of Jaccard Score versus the number of selected features on each data set based on Laplacian Score, SPEC and EVSC. Figure 3(f) show the trend of the average Jaccard Score over all five data sets versus feature number, which largely justifies the superiority of our method on average. All figures show that the ability of our selected features in preserving sample similarity is better than Laplacian Score and SPEC. Especially on CLL, PIX10P and TOX, the Jaccard Score is nearly 0.8 by using only 100 features.

Table 3 shows the average Jaccard Score over all chosen feature numbers of each data set based on Laplacian Score, SPEC and EVSC. As we can see, the score differences between EVSC with Laplacian Score and SPEC are at least 0.2098 for CLL, 0.4512 for CLA, 0.0928 for ORL10P, 0.5396 for PIX10P and 0.3216 for TOX. The last row of table 3 reveals that the average Jaccard Score over all feature numbers and all five data sets is increased with at least 0.37988 by EVSC.

Best feature number For each data set, we calculate the EVSC of each feature, and rank the features decreasingly according to the values of EVSC. Figure 4(a-c) describe the trend of EVSC score, Jaccard Score, Clustering Accuracy versus the leading features selected by EVSC according to dataset CLA, CLL and TOX, respectively. The vertical line of each figure denotes the turning point of the EVSC trend. It is clear that we are likely to achieve nearly best results for Jaccard Score and Clustering Accuracy at the turning point of EVSC, which suggests that our method can help to determine the best feature number of each dataset.

[2] http://www.zjucadcg.cn/dengcai/Data/code

[3] http://featureselection.asu.edu/software.php

Difference between 'good' and 'bad' features For EVSC, Laplacian Score and SPEC, they may rank the features according to the scores, then the leading m features are considered as 'good' features while the last m features are taken as 'bad' features. To compare the performance of good and bad features selected by the three methods, we can plot the curves of Clustering Accuracy and Jaccard Score versus the best m or worst m features. Due to the space limit, we only use GLA dataset as the example. The green curve of Figure 5(a-c) denotes the Clustering Accuracy when we use the last 100,200,...,1900 features('bad' features) selected by each related scores, while the curves of other colors correspond to the top 100,200,...,1900 features('good' features). The Figure 5(d-f) show the Jaccard Score curves of 'good' and 'bad' features selected by three different methods. From these figures, we can conclude that EVSC can achieve the largest and most stable performance difference between the 'good' and 'bad' features on GLA.

Table 4 and 5 describe the average Clustering Accuracy and Jaccard Score over all chosen 'bad' feature subsets of each dataset respectively. By comparing table 2 with table 4, we can find that the average clustering performance differences between 'good' and 'bad' features on datasets are: 0.0636 for Laplacian score, 0.0728 for SPEC, and 0.159 for EVSC. By comparing table 3 with table 5, we can observe that the average Jaccard Score difference between 'good' and 'bad' features for EVSC is more than 2.3 times larger than that of Laplacian Score and SPEC.

Stability Figure 5(a-f) also show the stability of EVSC compared with Laplacian Score and SPEC. For Clustering Accuracy and Jaccard Score, the trend of EVSC is almost rising, while the curves of Laplacian Score and SPEC fluctuate greatly. For ACC with SPEC and JAS with Laplacian Score, 'bad' features are even better than 'good' features in some data sets.

5. Conclusion

In this paper, we propose a new spectral feature selection method called EVSC, which performs the impact analysis of each feature on the eigenvalues of graph Laplacian. In comparison with two methods, Laplacian Score and SPEC, EVSC demonstrates excellent performance and high stability. Besides, our method can be useful to determine the best feature number.

6. Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant No. 60703110.

References

Ng, A. Y., Jordan, M. I. and Weiss, Y. *On Spectral Clustering: Analysis and an Algorithm.* In Advances in Neural Information Processing Systems, 2001.

Hastie, T., Tibshirani, R. and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* . New York: Springer-Verlag, 2001.

Belkin, M., and Niyogi, P. *Laplacian eigenmaps and spectral techniques for embedding and clustering.* In Advances in Neural Information Processing Systems, 2001.

Duda, R., Hart, P. and Stork, D. *Pattern Classification.* New York, 2001.

He, X. F., Cai, D. and Niyogi P. *Laplacian Score for Feature Selection.* In Advances in Neural Information Processing Systems, 2005.

Luxbury, U. V. *A Tutorial on Spectral Clustering.* Statistics and Computing, 2007.

Zhao, Z. and Liu, H. *Semi-supervised Feature Selection via Spectral Analysis.* SIAM International Conference on Data Mining, 2007.

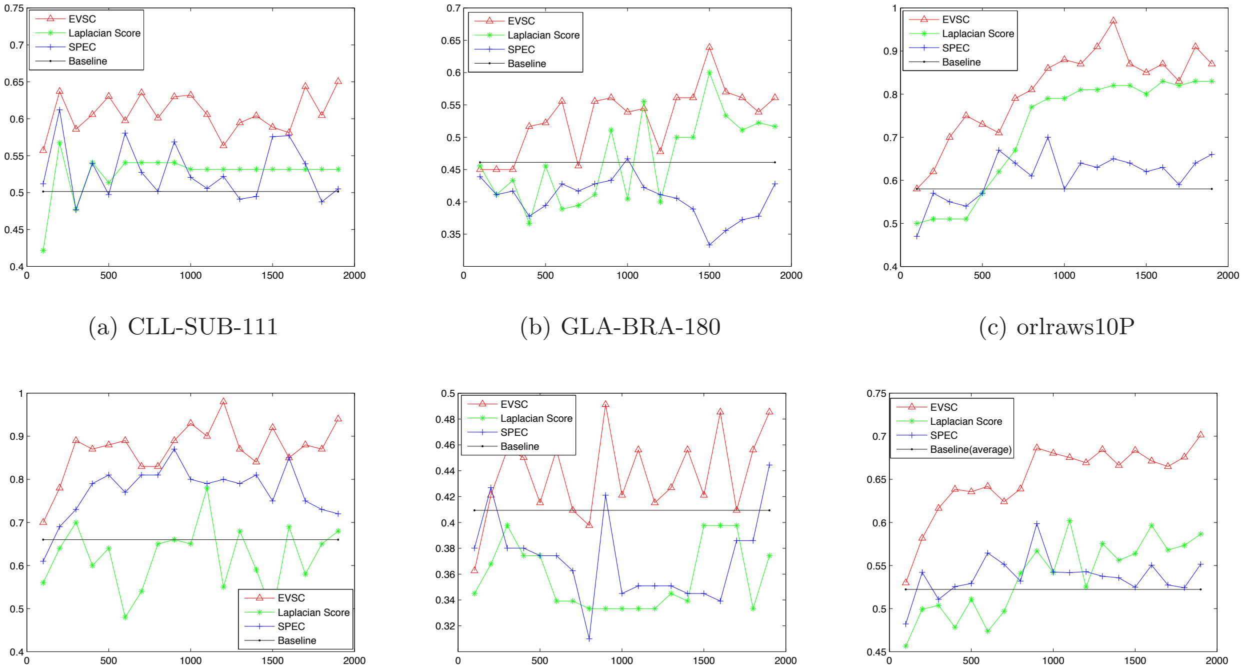
Zhao, Z. and Liu, H. *Spectral Feature Selection for Supervised and Unsupervised Learning.* In Proceedings of the 24th International Conference on Machine Learning, 2007.

Nie, F. P., Xiang, S. P. Jia, Y. Q. Zhang, C. S. and Yan, S. C. *Trace Ratio Criterion for Feature Selection.* In Proceedings of the 23th AAAI Conference on Artificial Intelligence, 2008.

Cai, D., Zhang, C. Y. and He, X. F. *Unsupervised Feature Selection for Multi-cluster Data.* In Proceeding of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD'10), 2010.

Zhao, Z., Wang, L. and Liu, H. *Efficient Spectral Feature Selection with Minimum Redundancy.* In Proc. 2010 AAAI Conf. on Artificial Intelligence(AAAI'10), 2010.

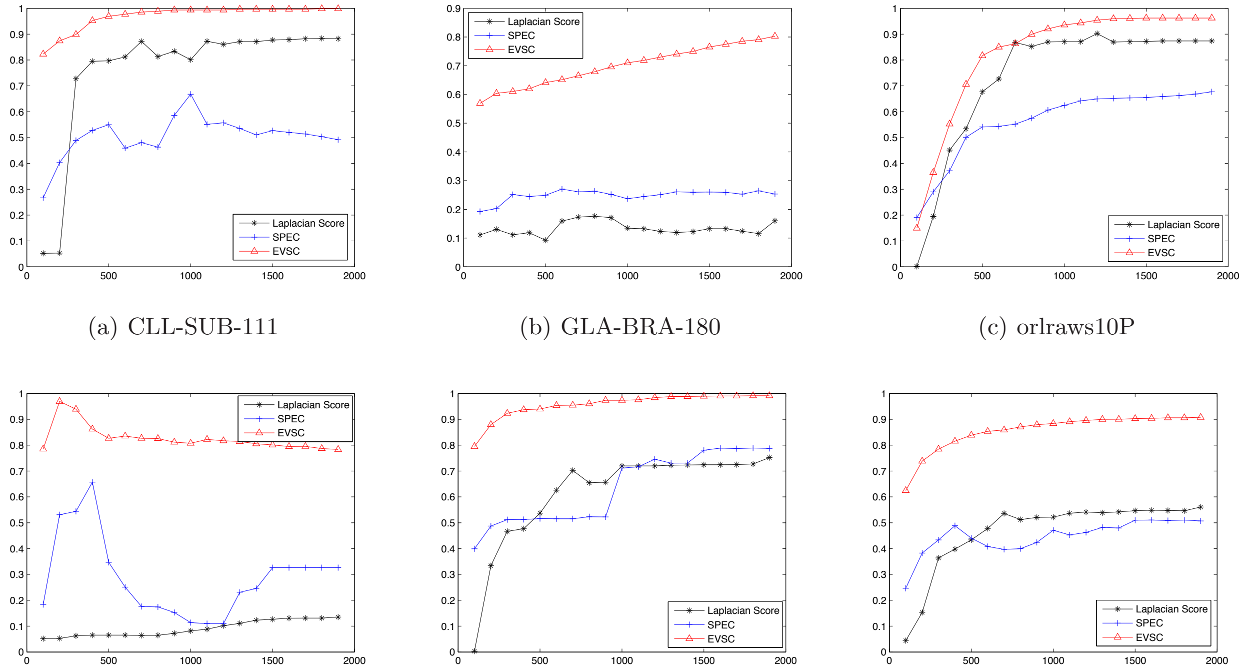
Ning, H. Z., Xu, W., Chi, Y., Gong, Y. H. and Huang, T. S. *Incremental Spectral Clustering by Efficiently Updating The Eign-System.* Pattern Recognition (PR) 43(1):113-127 (2010).



(a) CLL-SUB-111 (b) GLA-BRA-180 (c) orlraws10P

(d) pixraw10P (e) TOX-171 (f) ACC on all data sets

Figure 2. Clustering performance vs. the number of selected features based on Laplacian Score, SPEC and EVSC.



(a) CLL-SUB-111 (b) GLA-BRA-180 (c) orlraws10P

(d) pixraw10P (e) TOX-171 (f) Average JAS on all data sets

Figure 3. Jaccard Score vs. the number of selected features based on Laplacian Score, SPEC and EVSC.

Table 2: Average Clustering Accuracy

	Laplacianscore	*SPEC*	*EVSC*
PIX	0.6216	0.7726	0.8705
ORL	0.7163	0.6105	0.8095
GLA	0.4669	0.4056	0.53
CLL	0.5262	0.5281	0.6078
TOX	0.3712	0.3712	0.4364
Ave	0.54044	0.5376	0.65084

Table 3: Average Jaccard Score

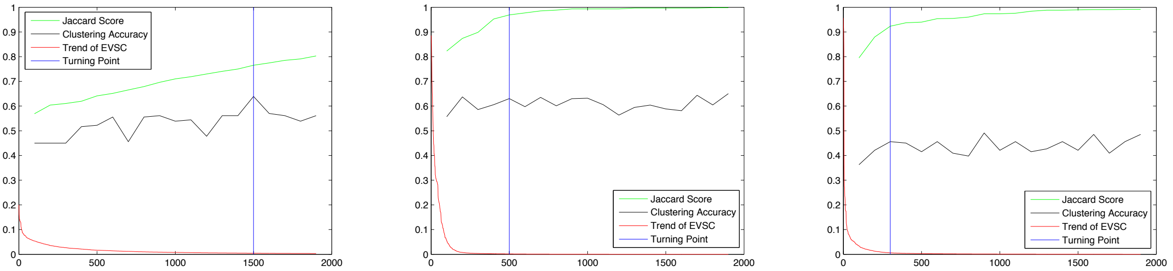
	Laplacianscore	*SPEC*	*EVSC*
PIX	0.0908	0.2873	0.8269
ORL	0.7329	0.5638	0.8257
GLA	0.249	0.1337	0.7002
CLL	0.5053	0.7598	0.9696
TOX	0.6165	0.6353	0.9569
Ave	0.4389	0.47598	0.85586

Table4: Average ACC on 'bad' features

	Laplacianscore	*SPEC*	*EVSC*
PIX	0.6742	0.6053	0.6563
ORL	0.5195	0.5268	0.5311
GLA	0.4269	0.436	0.3929
CLL	0.3751	0.3722	0.5301
TOX	0.3881	0.3838	0.3488
Ave	0.47676	0.46482	0.49184

Table 5: Average JAS on 'bad' features

	Laplacianscore	*SPEC*	*EVSC*
PIX	0.014	0.0364	0.0014
ORL	0.1073	0.0016	0.086
GLA	0.4343	0.114	0.048
CLL	0.6521	0.3261	0.1038
TOX	0.0891	0.0645	0.0016
Ave	0.2568	0.1085	0.0482

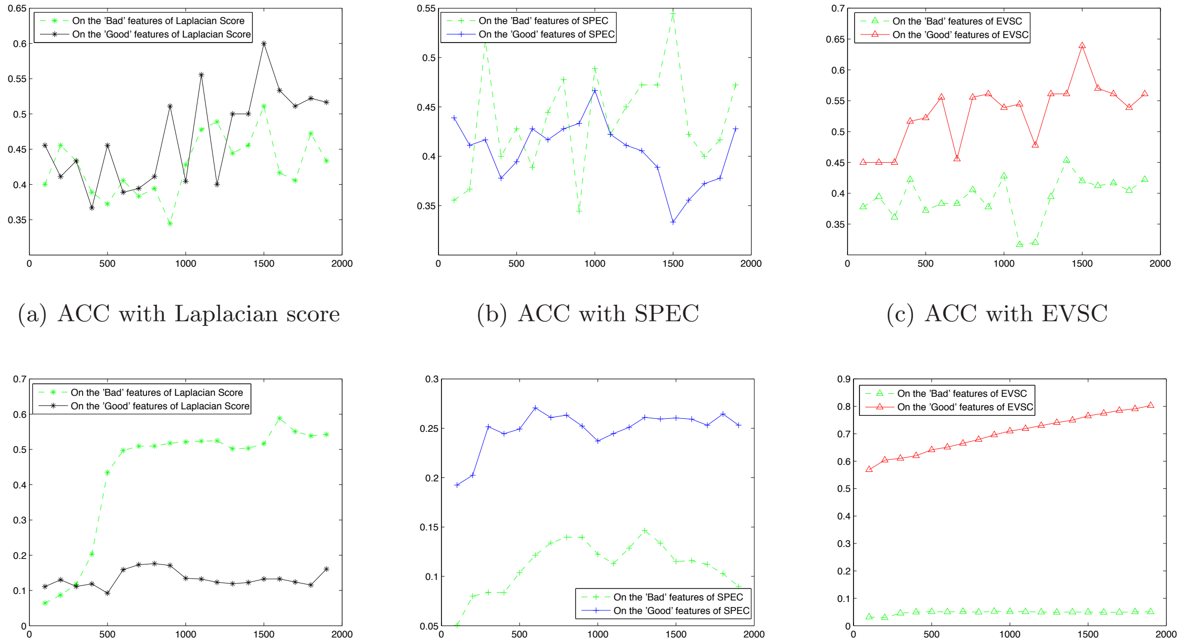


(a) CLA-BRA-180

(b) CLL-SUB-111

(c) TOX-171

Figure 4. The trend of the feature scores of the leading 2000 features selected by EVSC.



(a) ACC with Laplacian score

(b) ACC with SPEC

(c) ACC with EVSC

(d) JAS with Laplacian score

(e) JAS with SPEC

(f) JAS with EVSC

Figure 5. Performance difference between 'good' and 'bad' features on CLA-BRA-180.