Semi-supervised Feature Selection via Spectral Analysis

Zheng Zhao *   Huan Liu *

Abstract

Feature selection is an important task in effective data mining. A new challenge to feature selection is the so-called "small labeled-sample problem" in which labeled data is small and unlabeled data is large. The paucity of labeled instances provides insufficient information about the structure of the target concept, and can cause supervised feature selection algorithms to fail. Unsupervised feature selection algorithms can work without labeled data. However, these algorithms ignore label information, which may lead to performance deterioration. In this work, we propose to use both (small) labeled and (large) unlabeled data in feature selection, which is a topic has not yet been addressed in feature selection research. We present a semi-supervised feature selection algorithm based on spectral analysis. The algorithm exploits both labeled and unlabeled data through a regularization framework, which provides an effective way to address the "small labeled-sample" problem. Experimental results demonstrated the efficacy of our approach and confirmed that small labeled samples can help feature selection with unlabeled data.

Keyword: Feature Selection, Semi-supervised Learning, Machine Learning, Spectral Analysis

1 Introduction

The high dimensionality of data poses a challenge to learning tasks. In the presence of many irrelevant features, learning algorithms tend to overfitting. Various studies show that features can be removed without performance deterioration. Feature selection is one effective means to identify relevant features for dimension reduction [4]. The training data used in feature selection can be either labeled or unlabeled, corresponding to supervised and unsupervised feature selection [8]. In supervised feature selection, feature relevance can be evaluated by their correlation with the class label. And in unsupervised feature selection, without label information, feature relevance can be evaluated by their capability of keeping certain properties of the data, such

as the variance or the separability. Data are abundant and continue to accumulate in an unprecedent rate, but labeled data are costly to obtain. It is common to have a data set with huge dimensionality but small labeled-sample size. The data sets of this kind present a serious challenge, the so-called "small labeled-sample problem" [7], to supervised feature selection, that is, when the labeled sample size is too small to carry sufficient information about the target concept, supervised feature selection algorithms fail with either unintentionally removing many relevant features or selecting irrelevant features, which seems to be significant only on the small labeled data. Unsupervised feature selection algorithms can be an alternative in this case, as they are able to use the large amount of unlabeled data. However, as these algorithms ignore label information, important hints from labeled data are left out and this will generally downgrades the performance of unsupervised feature selection algorithms.

Under the assumption that labeled and unlabeled data are sampled from the same population generated by target concept, using both labeled and unlabeled data is expected to better estimate feature relevance. The task of learning from mixed labeled and unlabeled data is of semi-supervised learning [2]. In this paper, we present a *semi-supervised feature selection* algorithm based on the spectral graph theory [3]. The algorithm ranks features through a regularization framework, in which a feature's relevance is evaluated by its fitness with both labeled and unlabeled data.

2 Notations and Definitions

In semi-supervised learning, a data set of n data points $X = (\mathbf{x}_i)_{i \in [n]}$ consists of two subsets depending on the label availability: $X_L = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l)$ for which labels $Y_L = (y_1, y_2, ..., y_l)$ are provided, and $X_U = (\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, ..., \mathbf{x}_{l+u})$ whose labels are not given. Here data point $\mathbf{x}_i$ is a vector with m dimensions (features), and label y_i is an integer from $\{+1, -1\}^1$, and $l + u = n$ (n is the total number of instances). When $l = 0$, data X is for unsupervised learning; when $u = 0$, X

---

*Computer Science and Engineering (CSE) Department, Arizona State University (ASU), Tempe, AZ, 85281. {zheng.zhao, huan.liu}@asu.edu

[1]This is corresponding to data with binary classes, which is the case we will study in this paper. If the data has multiple classes, we have $y_i \in \{1, 2, \ldots, c\}$, where c is the number of classes.

is for supervised learning. Let $F_1, F_2, ..., F_m$ denote the m features of X and $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_m$ be the corresponding feature vectors that record the feature value on each instance. We give the definition of semi-supervised feature selection as:

DEFINITION 1. (SEMI-SUPERVISED FEATURE SELECTION) *Given data X_L and $X_U \subseteq R^m$, semi-supervised feature selection is to use both X_L and X_U to identify the set of most relevant features $\{F_{j_1}, F_{j_2}, ..., F_{j_k}\}$ of the target concept, where $k \leq m$ and $j_r \in \{1, 2, ..., m\}$ for $r \in \{1, 2, ..., k\}$.*

In this work, we employ the spectral graph theory [3] to semi-supervised feature selection. In the following, we provide some definitions and basic concepts from the spectral graph theory used in the paper. Given a data set X, let $G(V, E)$ be the undirected graph constructed from X, with V is its node set and E is its edge set. The i-th node v_i of G corresponds to $\mathbf{x}_i \in X$ and there is an edge between each nodes pair (v_i, v_j), whose weight $w_{ij} = w(v_i, v_j)$ is determined by $\psi(\mathbf{x}_i, \mathbf{x}_j)$, where $\psi(\cdot)$ is a similarity function defined as: $\psi(\cdot): R^n \times R^n \to R^+$. The volume of a node set $S \subseteq V$ is defined as $\mathrm{vol}S = \sum_{v_i \in S, v_j \in V} \sum_{(v_i, v_j) \in E} w_{ij}$. Let (S, S^c) be a partition of V, the cut induced by (S, S^c) is defined as $\mathrm{cut}(S, S^c) = \sum_{v_i \in S, v_j \in S^c} w_{ij}$. Instead of connecting each nodes pair with an edge, v_i and v_j are connected, if and only if v_i or v_j is one of the k nearest neighbors of the other. This will form a k-neighborhood graph, G. In the paper we use I to denote the identity matrix, $\mathbf{e}$ to denote the column vector with all its elements to be 1 and $\mathbf{e} = \{1, 1, ..., 1\}^T$. Below we give the definitions of adjacency matrix, degree matrix and Laplacian matrix, which are frequently used in spectral graph theory.

DEFINITION 2. (ADJACENCY MATRIX W) *Let G be the graph construct from X, the adjacency matrix of G is defined as:*

$$(2.1) \qquad W_{ij} = \begin{cases} w_{ij} & if \ (v_i, v_j) \in E \\ 0 & otherwise \end{cases}$$

DEFINITION 3. (DEGREE MATRIX D) *Let $\mathbf{d}$ denote the vector: $\mathbf{d} = \{d_1, d_2, ..., d_n\}$, where $d_i = \sum_{k=1}^{n} w_{ik}$, the degree matrix D of the graph G is defined by: $D_{ij} = d_i$ if $i = j$, and 0 otherwise.*

According to the definition, more data points close to $\mathbf{x}_i$, means a larger d_i. Therefore d_i can be interpreted as an estimation of the density around the node v_i in graph G, which is corresponding to $\mathbf{x}_i$.

DEFINITION 4. (LAPLACIAN MATRIX L) *Given the adjacency matrix W and the degree matrix D of G, the*

Laplacian Matrix of graph G is defined as:

$$(2.2) \qquad L = D - W$$

The degree matrix D and the Laplacian matrix L satisfy the following properties [3]:

THEOREM 2.1. *Given W, L and D of G, we have:*

1. *Let $\mathbf{e} = \{1, 1, ..., 1\}^T$, $L * \mathbf{e} = 0$.*

2. *$\forall \mathbf{x} \in R^n$, $\mathbf{x}^T L \mathbf{x} = \sum_{\{v_i, v_j\} \in E} w_{ij}(x_i - x_j)^2$*

3. *$D \cdot \mathbf{e} = \mathbf{d}$ and $\mathbf{e}^T \cdot D \cdot \mathbf{e} = \mathrm{vol}V$.*

Applying the spectral graph theory to unsupervised learning results in spectral clustering algorithms [10], which have been proved to be effective in many applications. Spectral clustering algorithms, such as ratio cut and normalized cut, transform the original clustering problem to the cut problems on graph models. And the (local) optimal cluster indicator can be reconstructed from the eigenvectors of the corresponding matrix defined in the cut problem. Instead of reconstructing the cluster indicators from eigenvectors, we show a way to construct them from feature vectors. Thus, the fitness of cluster indicators can be evaluated by both labeled and unlabeled data, paving the way to evaluate feature relevance using both labeled and unlabeled data.

3 Semi-supervised Feature Selection

Supervised and unsupervised feature selection methods require to measure feature relevance, but in different ways. Therefore the key for designing an effective semi-supervised feature selection algorithm is to develop a framework, under which the relevance of a feature can be evaluated by both labeled and unlabeled data in a natural way. The clustering assumption is a base assumption for most semi-supervised learning algorithms. It assumes that "if points are in the same cluster, they are likely to be of the same class" [2]. In this spirit, we propose a semi-supervised feature selection algorithm, *sSelect*. The basic idea is illustrated in Figure 1. We first transform a feature vector $\mathbf{f}_i$ into a cluster indicator, so each element $f_{i_j}, (j = 1, 2, ..., n)$ of $\mathbf{f}_i$ indicates the affiliation of the corresponding instance $\mathbf{x}_j$. The fitness of the cluster indicator can be evaluated by two factors: (1) separability - whether the cluster structures formed are well separable; and (2) consistency - whether the cluster structures formed is consistent with the given label information. The ideal case is all labeled data in each cluster coming from the same class.

Suppose we have two feature vectors $\mathbf{f}$ and $\mathbf{f}'$, and the corresponding cluster indicators are $\mathbf{g}$ and $\mathbf{g}'$ (we will elaborate how they are formed in the next section).

The cluster structures formed by $\mathbf{g}$ and $\mathbf{g}'$ are shown in Figure 1. Comparing with the cluster structures formed by $\mathbf{g}'$, those formed by $\mathbf{g}$ are preferred. From the unlabeled data point of view, both cluster indicators form clearly separable cluster structures. However, when the label information is considered, the cluster structure formed by $\mathbf{g}$ turn out to be more consistent, because all labeled data in a cluster are of the same class. Under the clustering assumption, $\mathbf{g}$ fits the data better than $\mathbf{g}'$, suggesting the feature corresponding to the feature vector $\mathbf{f}$ is more relevant with target concept than the feature corresponding to feature vector $\hat{\mathbf{f}}$. In the next, we show how to construct cluster indicators from feature vectors semi-supervised feature selection.
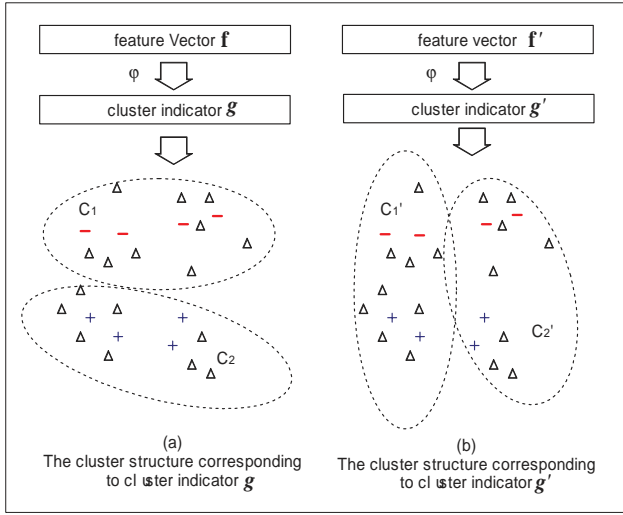

Figure 1: The basic idea for comparing the fitness of cluster indicators according to both labeled and unlabeled data for semi-supervised feature selection. "-" corresponds to instances of negative class, "+" to those of positive class, and "△" to unlabeled instances.

3.1 Clustering Indicator Construction The normalized min-cut clustering algorithm was first proposed by Shi and Malik in [10], and has been shown to be superior to other cluster algorithms, such as ratio cut[6]. Our method resorts to transforming feature vectors to the cluster indicators of normalized min-cut. Given a graph $G = (V, E)$ constructed from data X, the normalized min-cut clustering algorithm finds a cut (S, S^c) for G, that minimizes the cost function:

$$(3.3) \quad Ncut(S, S^c) = \frac{cut(S, S^c)}{\mathrm{vol}S} + \frac{cut(S^c, S)}{\mathrm{vol}S^c}$$

3.1.1 Cluster indicator space Let $\mathbf{g} = \{g_1, g_2, \ldots, g_n\}$ be the clustering indicator and $\gamma = \mathrm{vol}S/\mathrm{vol}V$, the

minimization of (3.3) can be rewritten as:

$$(3.4) \quad \min \frac{\sum_{(v_i, v_j) \in E} (g_i - g_j)^2 \times w_{ij}}{2 \sum_{v_i \in V} g_i^2 \times d_i} = \frac{\mathbf{g}^T L \mathbf{g}}{\mathbf{g}^T D \mathbf{g}}$$

$$s.t. \quad g_i \in \{2(1 - \gamma), -2\gamma\} \quad and \quad < \mathbf{g} \cdot \mathbf{d} >= 0$$

The combinatorial optimization problem specified in (3.4) is intractable. In [10] the problem is relaxed to allow g_i, $i \in \{1, \ldots, n\}$ to have any real value, instead of only one of the two discrete values, $2(1 - \gamma)$ and -2γ. The relaxed problem can be solved efficiently by calculating the harmonic eigenfunction of the normalized Laplacian matrix $\mathcal{L} = D^{-1/2} L D^{-1/2}$ [3]. It can be proved that the harmonic eigenfunction of $\mathcal{L}$ is orthogonal to $\mathbf{d}$ [10]. Since elements in $\mathbf{d}$ estimate the density around the notes in G, the orthogonality between the cluster indicator and $\mathbf{d}$ implies that the data density of clusters should be balance. For the relaxed problem, we give the definition for the *cluster indictor space* of normalized min-cut.

DEFINITION 5. (CLUSTER INDICATOR SPACE) *Given a graph G, the cluster indicator space $\mathcal{S}$ of normalized min-cut clustering on G is defined as:*

$$(3.5) \quad \mathcal{S} = \{\mathbf{g} | \, \mathbf{g} \in R^n, \, < \mathbf{g} \cdot \mathbf{d} >= 0\}$$

A vector is a member of the cluster indicator space $\mathcal{S}$, if and only if it is orthogonal to $\mathbf{d}$.

3.1.2 Transformation for features vectors Given a cluster indicator, the fitness of the indicator can be evaluated by both labeled and unlabeled data. If a feature vector $\mathbf{f}$ is orthogonal to $\mathbf{d}$, it is a cluster indicator and its fitness can be evaluated by using the way we mentioned above. However, not every feature vector of X is naturally orthogonal to $\mathbf{d}$. Therefore, we introduce an F-C transformation φ, which transforms an n dimensional feature vector $\mathbf{f} \in R^n$ to a vector in cluster space $\mathcal{S}$.

DEFINITION 6. (F-C TRANSFORMATION) *Let $\mathbf{f} \in R^n$ and $\mathbf{e} = \{1, \ldots, 1\}^T$, the F-C transformation φ is defined as:*

$$(3.6) \quad \varphi(\mathbf{f}) = \mathbf{f} - \frac{\sum_i^n f_i d_i}{\mathrm{vol}V} \cdot \mathbf{e};$$

The F-C transformation φ defines a linear transformation and has the following properties: first, it transforms $\forall \, \mathbf{f} \in R^n$ into a vector in space $\mathcal{S}$; second, working in R^n via φ, we can achieve the same optimal cut value

for Equation (3.4), as the one we achieved in $\mathcal{S}$; and third, among all linear transformations in the form of $\ell(\mathbf{f}) = \mathbf{f} + \mu \cdot \mathbf{e}$, where $\mu \in R$, the cluster indicator generated from φ upper bounds the value of Equation (3.4), and provides a reliable estimation of the fitness of $\mathbf{f}$ with data X. These properties are summarized in Theorem 3.1, and Theorem 3.2 below.

THEOREM 3.1. φ satisfies following properties:

1. $\forall \mathbf{f} \in R^n, \ \ <\varphi(\mathbf{f}) \cdot \mathbf{d}> = 0$

2. $\forall \mathbf{f}_1, \ \mathbf{f}_2 \in R^n, \ \ \mathbf{f}_1^T \cdot L \cdot \mathbf{f}_2 = (\varphi(\mathbf{f}_1))^T \cdot L \cdot \varphi(\mathbf{f}_2)$

3. $\forall \mathbf{g} \in \mathcal{S}, \ \varphi(\mathbf{g}) = \mathbf{g}$

4. $Ncut^*_{\varphi(R^n)} = Ncut^*_{\mathcal{S}}$

Here, $Ncut^*$ denotes the optimal cut value.

THEOREM 3.2. Among all linear transformations in the form: $\ell(\mathbf{f}) = \mathbf{f} + \mu \cdot \mathbf{e}$, where $\mu \in R$ and $\mathbf{e} = (1, ..., 1)^T$, $Ncut_{\varphi(\mathbf{f})}$ upper bounds the value of Equation (3.4).

Due to the space limit, we omit the proofs for the above theorems. The complete proofs of the two theorems can be found in [12].

3.2 Algorithm sSelect The F-C transformation φ transforms a feature vector $\mathbf{f}$ into a cluster indicator $\mathbf{g}$, which forms a basis for us to evaluate the feature on both labeled and unlabeled data. Given a cluster indicator $\mathbf{g}$, labeled data X_L and unlabeled data X_U, the fitness should be evaluated by: (1) whether the clusters formed by the indicator are well separable (renders a small cut value), and (2) whether it is consistent with the label information as shown in Figure 1. In this spirit we design a regularization framework, which enables us to evaluate the fitness of the cluster indicator using both labeled and unlabeled data. Let $\mathbf{g}$ be the cluster indicator generated from a feature vector $\mathbf{f}$ and $\hat{\mathbf{g}} = sign(\mathbf{g})$,[2] the regularization framework is defined as:

$$(3.7) \quad \lambda \frac{\sum\limits_{v_i \sim v_j} (g_i - g_j)^2 \times w_{ij}}{2 \sum\limits_{v_i \in V} g_i^2 \times d_i} + (1-\lambda)(1 - NMI(\hat{\mathbf{g}}, \mathbf{y}))$$

In Equation (3.7), $NMI(\hat{\mathbf{g}}, \mathbf{y})$ is the normalized mutual information [9] between $\hat{\mathbf{g}}$ and $\mathbf{y}$, which is used to measures the consistency between the discretized cluster indicator and the label data, irrespective to

---

[2]For the 2nd term (labeled part) in Equation (3.7), we need discretized cluster indicator. To transform a continuous cluster indicator to a discretized one, we use 0 as the cut point and binarize the continuous cluster indicator, which is one option suggested in [10]

how the cluster indicator is mapped to classes (i.e. $(1, -1) \rightarrow (+, -)$ or $(1, -1) \rightarrow (-, +)$). The first term of Equation (3.7) calculates the cut value of using $\mathbf{g}$ as the cluster indicator for data X. The second term estimates the corresponding classification loss of $\hat{\mathbf{g}}$ according to the labeled data. In this framework, the evaluation with either labeled or unlabeled data is based on the cluster indicator $\mathbf{g}$, which serves as a common base and makes the integration of the two terms of Equation (3.7) reasonable. Given the framework, we propose a **s**emi-supervised feature **Select**ion algorithm, sSelect, below:

Algorithm 1: The spectral graph based semi-supervised feature selection algorithm (sSelect)

Input: X, Y_L, λ, k
Output: $SF_{sSelect}$, the ranked feature list
1 construct k-neighborhood graph G from X;
2 build W, $\mathbf{d}$ and L from G;
3 **for** each feature vector $\mathbf{f}_i$ **do**
4   construct $\mathbf{g}_i$ from $\mathbf{f}_i$ using φ;
5   calculate s_i, the score of F_i using Eq. (3.7);
6 **end**
7 $SF_{sSelect} \leftarrow$ ranking F_i in descending order;
8 return $SF_{sSelect}$;

Algorithm 1 has three parts. (1) Line 1-2, graph matrices are built using the training data. (2) Line 3-6, features are transformed and evaluated based on the graph. (3) Line 7-8, features are ranked in descending order in terms of relevance (the smaller the s_i, the more relevance the feature), so features selection can be done based on the returned feature list according the desired number of features. The time complexity of sSelect can be obtained as follow. First, we need $O(mn^2)$ operations to build $W, \mathbf{d}$ and L. Next, we need $O(n^2)$ operations to calculate s_i for each feature: transforming $\mathbf{f}_i$ to $\mathbf{g}_i$ requires $O(n)$ operations; calculating the cut value needs $O(n^2)$ operations; and using the confusion table to calculate NMI takes $O(c^2 n)$ operations (for binary class data $c^2 = 4$). Therefore, we need $O(mn^2)$ operations to calculate scores for m features. Last, we need $O(m \log m)$ operations to rank the features. Hence, the overall time complexity of sSelect is $O(mn^2)$.

4 Empirical Study

We now empirically evaluate the performance of sSelect. We compare the proposed algorithm with two representative feature selection algorithms: Laplacian Score [5] is a recent spectral graph-based unsupervised feature selection algorithm and Fisher Score [1] is a popular supervised feature selection algorithm which is employed in [5] for comparison. We implement sSelect algorithm

in the Matlab environment. All experiments were conducted on a PENTIUM IV 2.4G PC with 1.5GB RAM. In the experiment, the λ value is set to 0.1 and the RBF kernel function are used for building a neighborhood graph with the neighborhood size of 10.

4.1 Data sets We test the three feature selection algorithms on three real data sets generated from the 20-new-group data. The three data sets are: (1) PC *vs.* MAC (PCMAC), (2) BASEBALL *vs.* HOCKEY (HOCKBASE) and (3) MAC *vs.* BASEBALL (MACBASE). The four topics addressed in the three data sets are widely used for performance evaluation for learning algorithms. The three data sets are generated from the version 20-news-18828 using TMG package [11] with standard process. Detail information of the three benchmark data sets is listed in Table 1.

Data Set	PCMAC	HOCKBASE	MACBASE
instances	1943(982:961)	1993(994:999)	1955(961:994)
features	8298	8298	8298

Table 1: Summary of the three benchmark data sets

4.2 Evaluation framework A common hypothesis used for evaluating the quality of a feature subset is: if a feature subset is more relevant with the target concept than others, a classifier learning with the feature subset should achieve better accuracy. In the normal evaluation framework, feature selection is carried out on the training data, and a classifier is trained and evaluated on the training and testing data, respectively, using selected features. To simulate the small labeled sample context, we set l, number of labeled data, to be 6 and 10 respectively. So few labeled instances, however, induce insufficiency for sensibly obtaining a classifier, whose estimated accuracy is used to evaluate the quality of a feature subset. Hence, we use 5-fold cross validation (CV) on the whole data X (recall that all instances in X have class labels). to estimate the accuracy for evaluating the quality of a feature subset. The details of the evaluation framework is shown in Algorithm 2. We define a projection operator $\Pi_{SF}(X)$ which retains the selected features in SF and removes unselected features. The process specified in Algorithm 2 is repeated for 20 times. The obtained accuracy is averaged and used for evaluating the quality of the feature subset selected according to each algorithm.

4.3 Comparison of feature quality Using the framework defined in Algorithm 2, we test the three algorithms on the three benchmark data sets. Figure 2 shows the plots for accuracy vs. different numbers of

---

Algorithm 2: Feature Evaluation Framework

1 for *each data set* **do**
2   Generate labeled data X_L by randomly sampling $\frac{1}{2} \cdot l$ instances from each class;
3 $X_U = X - X_L$;
4   /*using each algorithm to rank features*/;
5   **begin**
6 $SF_{sSelect} \leftarrow$ sSelect with $X_L + X_U$;
7 $SF_{LP} \leftarrow$ Laplacian Score with X_U;
8 $SF_F \leftarrow$ Fisher Score with X_L;
9   **end**
10   /*evaluating the quality of feature sets*/;
11   **for** $i =5$ *to* 50 *step* 5 **do**
12   Select top i features from $SF_{sSelect}$, SF_{LP}, SF_F and SF_{IG};
13 $X_{sSelect} \leftarrow \Pi_{SF_{sSelect}}(X)$;
14 $X_{LP} \leftarrow \Pi_{SF_{LP}}(X)$;
15 $X_F \leftarrow \Pi_{SF_F}(X)$;
16   Run 5-fold CV on $X_{sSelect}$, X_{LP} and X_F using 1NN and record accuracy;
17   **end**
18 end

---

selected features and different numbers of labeled data. As shown in the figure, *sSelect* works consistently better than the other two feature selection algorithms. Generally, *sSelect* works best and is followed by Fisher Score and Laplacian Score. From the figure, we can see, generally, the more features we select, the better accuracy we can achieve. A closer study reveals, generally, the accuracy of *sSelect* increases fast in the beginning (the number of selected feature is small) and slows down at the end (the number of selected feature is already large). This suggests that *sSelect* ranks features properly as important features are selected first.

For each data set and different numbers of labeled data, we average the accuracy for different number of selected features. The differences of the averaged accuracy among algorithms are list in Table 2. We can see that in terms of average accuracy gains, *sSelect* is 0.0921 better than Fisher Score and 0.1998 better than Laplacian Score. One trend can be clearly observed is that comparing with Laplacian Score, the accuracy differences become bigger when more labeled data is provided for training *sSelect*. This observation suggests that the label information is important for feature selection. This is also consistent with our understanding for the role of the label information in semi-supervised learning. The experiment results on the benchmark data sets confirm that using both labeled and unlabeled data does help feature selection.

L	Fisher Score	Laplacian Score
	PCMAC	
6	+0.0431	+0.1447
10	+0.0873	+0.1753
	HOCKBASE	
6	+0.0836	+0.1775
10	+0.1150	+0.1997
	MACBASE	
6	+0.0859	+0.2338
10	+0.1377	+0.2682
	AVERAGE	
	+0.0921	+0.1998

Table 2: A comparison of the average accuracy. L is for number of labeled data.

5 Conclusion

This work presents a concrete initial attempt to the new problem of semi-supervised feature selection. We propose an algorithm based on the spectral graph theory. We show that one can construct cluster indicators for normalized min-cut clustering from feature vectors which allows to evaluate fitness on both labeled and unlabeled data in determining feature relevance. Experimental results confirm that using labeled and unlabeled data together does help feature selection. Extending *sSelect* to multi-class data and studying ways for effectively tuning the regularization parameter are in our plan of future work. Another direction for semi-supervised feature selection is to iteratively propagate labels from labeled data to unlabeled data while carrying out feature selection. This requires feature selection and label propagation to be considered in an EM framework. Our preliminary experiment (to be reported elsewhere) shows that the performance of this method is unstable and heavily depends on the starting point (initial labeled data). Further work is needed to deepen our understanding of this approach.

References

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

[2] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, 2006.

[3] F. Chung. *Spectral graph theory*. AMS, 1997.

[4] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis: An International Journal*, 1(3):131–156, 1997.

[5] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2005.

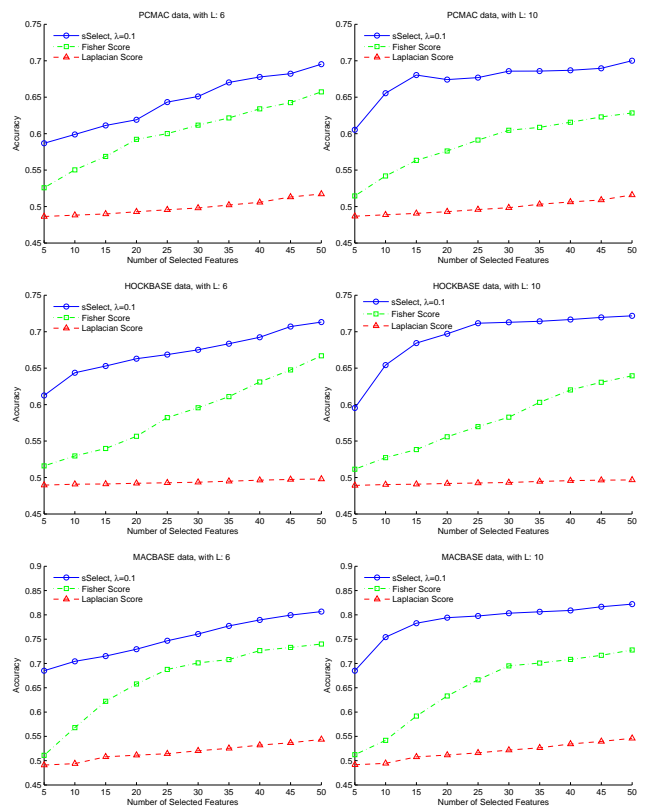[6] J. Huang. A combinatorial view of graph laplacians.

Figure 2: Accuracy vs. different numbers of selected features and different numbers of labeled data.

Technical report, Max Planck Institute for Biological Cybernetics, 2005.

[7] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[8] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17:491–502, 2005.

[9] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1988.

[10] J. B. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[11] D. Zeimpekis and E. Gallopoulos. Tmg: A matlab toolbox for generating term-document matrices from text collections. Technical report, University of Patras, Greece, 2005.

[12] Z. Zhao and H. Liu. Semi-supervised feature selection via spectral analysis. Technical Report TR-06-022, Department of Computer Science and Engineering, Arizona State University, 2006.