Spectral Feature Selection for Supervised and Unsupervised Learning

Zheng Zhao ZHAOZHENG@ASU.EDU
Huan Liu HUAN.LIU@ASU.EDU
Department of Computer Science and Engineering, Arizona State University

Abstract

Feature selection aims to reduce dimensionality for building comprehensible learning models with good generalization performance. Feature selection algorithms are largely studied separately according to the type of learning: supervised or unsupervised. This work exploits intrinsic properties underlying supervised and unsupervised feature selection algorithms, and proposes a unified framework for feature selection based on spectral graph theory. The proposed framework is able to generate families of algorithms for both supervised and unsupervised feature selection. And we show that existing powerful algorithms such as ReliefF (supervised) and Laplacian Score (unsupervised) are special cases of the proposed framework. To the best of our knowledge, this work is the first attempt to unify supervised and unsupervised feature selection, and enable their joint study under a general framework. Experiments demonstrated the efficacy of the novel algorithms derived from the framework.

1. Introduction

The high dimensionality of data poses challenges to learning tasks such as the curse of dimensionality. In the presence of many irrelevant features, learning models tend to overfitting and become less comprehensible. Feature selection is one effective means to identify relevant features for dimension reduction (Guyon & Elisseeff, 2003; Liu & Yu, 2005). Various studies show that features can be removed without performance deterioration. The training data can be either labeled or unlabeled, leading to the development of supervised and unsupervised feature selection algorithms. To date, researchers have studied the two types of feature selection algorithms largely separately. Supervised feature selection determines feature relevance by evaluating feature's correlation with the class, and without labels, unsupervised feature selection exploits data variance and separability to evaluate feature relevance (He et al., 2005; Dy & Brodley, 2004). In this paper, we endeavor to investigate some intrinsic properties of supervised and unsupervised feature selection algorithms, explore their possible connections, and develop a unified framework that will enable us to (1) jointly study supervised and unsupervised feature selection algorithms, (2) gain a deeper understanding of some existing successful algorithms, and (3) derive novel algorithms with better performance. To the best of our knowledge, this work presents the first attempt to unify supervised and unsupervised feature selection by developing a general framework.

The chasm between supervised and unsupervised feature selection seems difficult to close as one works with class labels and the other does not. However, if we change the perspective and put less focus on class information, both supervised and unsupervised feature selection can be viewed as an effort to select features that are consistent with the target concept. In supervised learning the target concept is related to class affiliation, while in unsupervised learning the target concept is usually related to the innate structures of the data. Essentially, in both cases, the target concept is related to dividing instances into well separable subsets according to different definitions of the separability. The challenge now is how to develop a unified representation based on which different types of separability can be measured. Pairwise instance similarity is widely used in both supervised and unsupervised learning to describe the relationships among instances. Given a set of pairwise instance similarities $\mathbb{S}$, the separability of the instances can be studied by

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

analyzing the spectrum of the graph induced from $\mathbb{S}$. For feature selection, therefore, if we can develop the capability of determining feature relevance using $\mathbb{S}$, we will be able to build a framework that unifies both supervised and unsupervised feature selection. Based on spectral graph theory (Chung, 1997), in this work, we present a unified framework for feature selection using the spectrum of the graph induced from $\mathbb{S}$. By designing different $\mathbb{S}$'s, the unified framework can produce families of algorithms for both supervised and unsupervised feature selection. We show that two powerful feature selection algorithms, ReliefF (Robnik-Sikonja & Kononenko, 2003) and Laplacian Score (He et al., 2005) are special cases of the proposed framework. We begin with the notations used in this study below.

2. Notations

In this work, we use X to denote a data set of n instances $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, $\mathbf{x}_i \in R^m$. We use $F_1, F_2, ..., F_m$ to denote the m features, and $\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_m$ are the corresponding feature vectors. For supervised learning, $Y = (y_1, y_2, ..., y_n)$ are the class labels. According to the geometric structure of the data or the class affiliation, a set of pairwise instance similarity, $\mathbb{S}$ (or the corresponding similarity matrix S), can be constructed to represent the relationships among instances. For example, without using the class information, a popular similarity measure is the RBF kernel function:

$$S_{ij} = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{(2\delta^2)}} \tag{1}$$

Using the class labels, the similarity can be defined by:

$$S_{ij} = \begin{cases} \frac{1}{n_l}, & y_i = y_j = l \\ 0, & otherwise \end{cases} \tag{2}$$

where n_l denotes the number of instances in class l. Given X, we use $\mathbb{G}(V, E)$ to denote the undirected graph constructed from $\mathbb{S}$, where V is the vertex set, and E is the edge set. The i-th vertex v_i of $\mathbb{G}$ corresponds to $\mathbf{x}_i \in X$ and there is an edge between each vertex pair (v_i, v_j), where the weight w_{ij} is determined by $\mathbb{S}$, $w_{ij} = S_{ij}$. Given $\mathbb{G}$, its *adjacency matrix* W is defined as $W(i,j) = w_{ij}$. 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 $\mathbb{G}$ is defined by: $D(i,j) = d_i$ if $i = j$, and 0 otherwise. Here d_i can be interpreted as an estimation of the density around x_i, since the more data points that are close to $\mathbf{x}_i$, the larger the d_i. Given the adjacency matrix W and the degree matrix D of $\mathbb{G}$, the *Laplacian matrix* L and the *normalized Laplacian matrix* $\mathcal{L}$ are defined as:

$$L = D - W; \quad \mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{3}$$

It is easy to verify that D and L satisfy the following properties (Chung, 1997).

Theorem 1 *Given W, L and D of $\mathbb{G}$, we have:*

1. *Let $\mathbf{e} = \{1, 1, \ldots, 1\}^T$, $L * \mathbf{e} = 0$.*
2. *$\forall\ \mathbf{x} \in R^n$, $\mathbf{x}^T L \mathbf{x} = \frac{1}{2} \sum_{v_i \sim v_j} w_{i,j}(x_i - x_j)^2$*
3. *$\forall\ \mathbf{x} \in R^n, \forall\ t \in R, (\mathbf{x} - t * \mathbf{e})^T L (\mathbf{x} - t * \mathbf{e}) = \mathbf{x}^T L \mathbf{x}$*

Here, $\mathbf{e} = \{1, 1, \ldots, 1\}^T$.

3. Spectral Feature Selection

Given a set of pairwise instance similarity $\mathbb{S}$, a graph $\mathbb{G}$ can be constructed to represent it. And the target concept specified in $\mathbb{S}$ is usually reflected by the structure of $\mathbb{G}$ (Chapelle et al., 2006). A feature that is *consistent* with the graph structure assigns similar values to instances that are near each other on the graph. As shown in Figure 1, feature F assigns values to instances consistently with the graph structure, but F' does not. Thus F is more relevant with the target concept and can separate the data better (forms groups with similar instances according to the target concept). According to graph theory, the structure information of a graph can be obtained from its spectrum. *Spectral feature selection* studies how to select features according to the structures of the graph induced from $\mathbb{S}$. In this work we employ the spectrum of the graph to measure *feature relevance* and elaborate how to realize spectral feature selection.

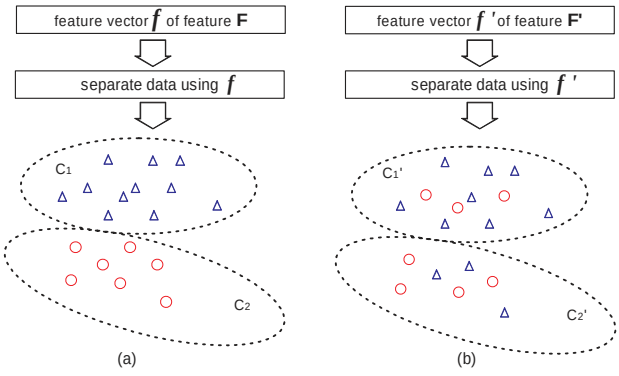

Figure 1. Consistency comparison of features. The target concept is represented by the graph structure (clusters indicated by the ellipses). Different shapes denote different values assigned by a feature.

3.1. Ranking Features on Graph

We formalize the above idea using the concept of normalized cut for graph, derive two improved functions

from the normalized cut function with the spectrum of the graph, and extend the three functions to their more general forms. These pave the way for constructing the unified framework proposed in this paper.

Evaluating Features via Normalized Cut

Given a graph $\mathbb{G}$, the Laplacian matrix of $\mathbb{G}$ is a linear operator on vectors $\boldsymbol{f} = (f_1, f_2, \ldots, f_n)^T \in \mathbb{R}^n$:

$$< \boldsymbol{f}, L\boldsymbol{f} >= \boldsymbol{f}^T L\boldsymbol{f} = \frac{1}{2}\sum_{v_i \sim v_j} w_{ij}(x_i - x_j)^2 \quad (4)$$

The equation quantifies how much $\boldsymbol{f}$ varies locally or how "smooth" it is over $\mathbb{G}$. More specifically, the smaller the value of $< \boldsymbol{f}, L\boldsymbol{f} >$, the smoother the vector $\boldsymbol{f}$ on $\mathbb{G}$. A smooth vector $\boldsymbol{f}$ assigns similar values to the instances that are close to each other on $\mathbb{G}$, thus it is consistent with the graph structure. This observation motivates us to apply L on a feature vector to measure its consistency with the graph structure. Given a feature vector $\boldsymbol{f}_i$ and L, two factors affect the value of $< \boldsymbol{f}_i, L\boldsymbol{f}_i >$: the norms of $\boldsymbol{f}_i$ and L. The two factors need to be removed, as they do not contain structure information of the data, but can cause the value of $< \boldsymbol{f}_i, L\boldsymbol{f}_i >$ to increase or decrease arbitrarily. The two factors can be removed via normalization. As $< \boldsymbol{f}_i, L\boldsymbol{f}_i >= \boldsymbol{f}_i^T L\boldsymbol{f}_i = \boldsymbol{f}_i^T D^{\frac{1}{2}}\mathcal{L}D^{\frac{1}{2}}\boldsymbol{f}_i = (D^{\frac{1}{2}}\boldsymbol{f}_i)^T\mathcal{L}(D^{\frac{1}{2}}\boldsymbol{f}_i)$. Let $\widetilde{\boldsymbol{f}}_i = (D^{\frac{1}{2}}\boldsymbol{f}_i)$ denote the weighted feature vector of F_i, and $\widehat{\boldsymbol{f}}_i = \frac{\widetilde{\boldsymbol{f}}_i}{||\widetilde{\boldsymbol{f}}_i||}$ the normalized weighted feature vector. The score of F_i can be evaluated by the following function:

$$\varphi_1(F_i) = \widehat{\boldsymbol{f}}_i^{\,T}\,\mathcal{L}\,\widehat{\boldsymbol{f}}_i \quad (5)$$

Theorem 2 *$\varphi_1(F_i)$ measures the value of the normalized cut (Shi & Malik, 1997) by using $\boldsymbol{f}_i$ as the soft cluster indicator to partition the graph $\mathbb{G}$.*

PROOF The theorem holds as:

$$\varphi_1(F_i) = \widehat{\boldsymbol{f}}_i^{\,T}\,\mathcal{L}\,\widehat{\boldsymbol{f}}_i = \frac{\boldsymbol{f}_i^{\,T} L\boldsymbol{f}_i}{\boldsymbol{f}_i^{\,T} D\boldsymbol{f}_i} \qquad \square$$

Ranking Features Using Graph Spectrum

Given the normalized Laplacian matrix $\mathcal{L}$, we calculate its spectral decomposition (λ_i, ξ_i), where λ_i is the eigenvalue and ξ_i is the eigenvector $(0 \le i \le n-1)$. Assuming $\lambda_0 \le \lambda_1 \le \ldots \le \lambda_{n-1}$, according to Theorem 1, we have: $\lambda_0 = 0$ and $\xi_0 = D^{\frac{1}{2}}\mathbf{e}$. (λ_0, ξ_0), which is usually called the trivial eigenpair of the graph. Also we can show that all the eigenvalues of $\mathcal{L}$ are contained in $[0, 2]$. Given the spectral decomposition of $\mathcal{L}$, we can rewrite Equation 5 using the eigensystem of $\mathcal{L}$.

Theorem 3 *Let (λ_j, ξ_j), $0 \le j \le n-1$ be the eigensystem of $\mathcal{L}$, and $\alpha_j = \cos\theta_j$ where θ_j is the angle between $\boldsymbol{f}_i$ and ξ_j. Equation (5) can be rewritten as:*

$$\varphi_1(F_i) = \sum_{j=0}^{n-1}\alpha_j^2\lambda_j, \quad \text{where} \quad \sum_{\mathrm{j}=0}^{\mathrm{n}-1}\alpha_{\mathrm{j}}^2 = 1 \quad (6)$$

PROOF: Let $\Sigma = \textsc{Diag}(\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$ and $U = (\xi_0, \xi_1, \ldots, \xi_{n-1})$. As $||\widehat{\boldsymbol{f}}_i|| = ||\xi_j|| = 1$, we have $\widehat{\boldsymbol{f}}_i^{\,T}\xi_j = \cos\theta_j$. We can rewrite $\widehat{\boldsymbol{f}}_i^{\,T}\,\mathcal{L}\,\widehat{\boldsymbol{f}}_i$ as:

$$\widehat{\boldsymbol{f}}_i^{\,T}\mathcal{L}\widehat{\boldsymbol{f}}_i = \widehat{\boldsymbol{f}}_i^{\,T}U\Sigma U^T\widehat{\boldsymbol{f}}_i$$
$$= (\alpha_0, \ldots, \alpha_{n-1})\Sigma(\alpha_0, \ldots, \alpha_{n-1})^T = \sum_{i=0}^{n-1}\alpha_i^2\lambda_i$$

Also $\sum_{j=0}^{n-1}\alpha_j^2 = 1$, as $UU^T = I$ and $||\widehat{\boldsymbol{f}}_i|| = 1$ $\qquad \square$

Theorem 3 says that using Equation (5), the score of F_i is calculated by combining the eigenvalues of $\mathcal{L}$, and $\cos\theta_1, \ldots, \cos\theta_{n-1}$ are the combination coefficients, which measures the similarity between the the feature vector and the eigenvectors. According to spectral clustering theories (Ng et al., 2001), the eigenvalues of $\mathcal{L}$ measure the separability of the components of the graph[1] and the eigenvectors are the corresponding soft cluster indicators (Shi & Malik, 1997). Since $\lambda_0 = 0$, Equation (6) can be rewritten as $\widehat{\boldsymbol{f}}_i^{\,T}\,\mathcal{L}\,\widehat{\boldsymbol{f}}_i = \sum_{j=1}^{n-1}\alpha_j^2\lambda_j$, meaning the value obtained from Equation (5) measures the graph separability by using $\widehat{\boldsymbol{f}}_i$ as the soft cluster indicator, and the separability is estimated by measuring the similarities between $\widehat{\boldsymbol{f}}_i$ and those nontrivial eigenvectors of $\mathcal{L}$. Since $\sum_{j=0}^{n-1}\alpha_j^2 = 1$ and $\alpha_0 \ge 0$, we have $\sum_{j=1}^{n-1}\alpha_j^2 \le 1$, and the bigger the α_0^2, the smaller the $\sum_{j=1}^{n-1}\alpha_j^2$. The value of $\varphi_1(F_i)$ can be small, if $\widehat{\boldsymbol{f}}_i$ is very similar with ξ_0. However, in this case, a small $\varphi_1(F_i)$ value does not indicate better separability, since the trivial eigenvector ξ_0 only carries density information around instances and does not determine separability. To handle this case, we propose to use $\sum_{j=1}^{n-1}\alpha_j^2$ to normalize $\varphi_1(F_i)$, which gives us the following ranking function:

$$\varphi_2(F_i) = \frac{\sum_{j=1}^{n-1}\alpha_j^2\lambda_j}{\sum_{j=1}^{n-1}\alpha_j^2} = \frac{\widehat{\boldsymbol{f}}_i^{\,T}\,\mathcal{L}\,\widehat{\boldsymbol{f}}_i}{1 - \widehat{\boldsymbol{f}}_i^{\,T}\,\xi_0} \quad (7)$$

A small $\varphi_2(F_i)$ indicates that $\widehat{\boldsymbol{f}}_i$ aligns closely to those nontrivial eigenvectors with small eigenvalues, hence

[1] The separability is associated with inter-cluster dissimilarity, and the smaller the cut values the better.

provides good separability. According to spectral clustering theory, the leading k eigenvectors of $\mathcal{L}$ form the optimal soft cluster indicators that separate $\mathbb{G}$ into k parts. Therefore, if k is known, we can also use the following function for ranking:

$$\varphi_3(F_i) = \sum_{j=1}^{k-1} (2 - \lambda_j)\alpha_j^2 \tag{8}$$

By its definition, φ_3 assigns bigger scores to features which offer better separability because achieving a big score entails that a feature aligns closely to nontrivial eigenvectors $\xi_1, \ldots, \xi_{k-1}$, with ξ_1 having the highest priority. By focusing on only the leading eigenvectors, φ_3 achieves an effect of reducing noise. Similar mechanism is used in Principle Component Analysis (PCA).

An Extension for Feature Ranking Functions

Laplacian matrix is also used by graph based learning models for designing regularization functions to penalize predictors that vary abruptly among adjacent vertices on graph. Smola and Kondor (2003) relate the eigenvectors of $\mathcal{L}$ to a Fourier basis and extend the usage of $\mathcal{L}$ to $\gamma(\mathcal{L})$, where $\gamma(\mathcal{L}) = \sum_{j=0}^{n-1} \gamma(\lambda_j)\xi_j\xi_j^T$. In the formulation, $\gamma(\lambda_j)$ is an increasing function that penalizes high frequency components[2]. As shown in (Zhang & Ando, 2006), $\gamma(\lambda_j)$ can be very helpful in a noisy learning environment. In the same spirit, we extend our feature ranking functions to the following:

$$\widehat{\varphi}_1(F_i) = \widehat{\boldsymbol{f}}_i^{\,T} \gamma(\mathcal{L})\, \widehat{\boldsymbol{f}}_i = \sum_{j=0}^{n-1} \alpha_j^2 \gamma(\lambda_j) \tag{9}$$

$$\widehat{\varphi}_2(F_i) = \frac{\sum_{j=1}^{n-1} \alpha_j^2 \gamma(\lambda_j)}{\sum_{j=1}^{n-1} \alpha_j^2} = \frac{\widehat{\boldsymbol{f}}_i^{\,T} \gamma(\mathcal{L})\, \widehat{\boldsymbol{f}}_i}{1 - \widehat{\boldsymbol{f}}_i^{\,T} \xi_0} \tag{10}$$

$$\widehat{\varphi}_3(F_i) = \sum_{j=1}^{k-1} (\gamma(2) - \gamma(\lambda_j))\alpha_j^2 \tag{11}$$

Calculating the spectral decomposition of $\mathcal{L}$ can be expensive for data with a large number of instances. However, since $\gamma(\cdot)$ is usually a rational function, $\gamma(\mathcal{L})$ can be calculated efficiently by regarding $\mathcal{L}$ as a variable and apply $\gamma(\cdot)$ on it. For example, assume $\gamma(\lambda) = \lambda^2$, then $\gamma(\mathcal{L}) = \mathcal{L}^2$. For $\widehat{\varphi}_3(\cdot)$, the $k-1$ leading eigenpairs of $\mathcal{L}$ can be obtained efficiently by using fast eigen-solvers such as the Implicitly Restarted Arnoldi method (Lehoucq 2001).

[2]Here λ_j is used to estimate frequency, as it measures how much the corresponding basis ξ_j varies on the graph.

3.2. *SPEC*- the Framework

The proposed framework is built on spectral graph theory. In the framework, the relevance of a feature is determined by its consistency with the structure of the graph induced from $\mathbb{S}$. The three feature ranking functions ($\widehat{\varphi}_1(\cdot), \widehat{\varphi}_2(\cdot)$ and $\widehat{\varphi}_3(\cdot)$) lay the foundation of the framework and enable us to derive families of supervised and unsupervised feature selection in a unified manner. We realize the unified framework in Algorithm 1. It selects features in three steps: (1) building similarity set $\mathbb{S}$ and constructing its graph representation (Line 1-3); (2) evaluating features using the spectrum of the graph (Line 4-6); and (3) ranking features in descending order in terms of feature relevance[3] (Line 7-8). We name the framework *SPEC*, stemming from the *SPEC*trum decomposition of $\mathcal{L}$.

Algorithm 1: *SPEC*

Input: X, $\gamma(\cdot)$, k, $\widehat{\varphi} \in \{\widehat{\varphi}_1, \widehat{\varphi}_2, \widehat{\varphi}_3\}$
Output: SF_{SPEC} - the ranked feature list

1 construct $\mathbb{S}$, the similarity set from X (and Y);
2 construct graph G from $\mathbb{S}$;
3 build W, D and L from G;
4 **for** *each feature vector* $\boldsymbol{f}_i$ **do**
5 $\quad \widehat{\boldsymbol{f}}_i \leftarrow \frac{D^{\frac{1}{2}}\boldsymbol{f}_i}{||D^{\frac{1}{2}}\boldsymbol{f}_i||}$; $\quad SF_{SPEC}(i) \leftarrow \widehat{\varphi}(F_i)$;
6 **end**
7 ranking SF_{SPEC} in ascending order for $\widehat{\varphi}_1$ and $\widehat{\varphi}_2$, or descending order for $\widehat{\varphi}_3$;
8 return SF_{SPEC};

The time complexity of *SPEC* largely depends on the cost of building the similarity matrix and the calculation of $\gamma(\cdot)$. If we use the *RBF* function to build the similarity matrix and $\gamma(\cdot)$ is in the form of $\mathcal{L}^r$, the time complexity of *SPEC* can be obtained as follow. First, we need $O(mn^2)$ operations to build $\mathbb{S}$, W, D, L and $\mathcal{L}$. And we need $O(rn^3)$ operations to calculate $\gamma(\mathcal{L})$. Next, we need $O(n^2)$ operations to calculate $SF_{SPEC}(i)$ for each feature: transforming $\boldsymbol{f}_i$ to $\widehat{\boldsymbol{f}}_i$ requires $O(n)$ operations; calculating using $\widehat{\varphi}_1$, $\widehat{\varphi}_2$ and $\widehat{\varphi}_2$ need $O(n^2)$ operations[4]. Therefore, we need $O(mn^2)$ operations to calculate scores for m features. Last, we needs $O(m \log m)$ operations to rank the features. Hence, the overall time complexity of *SPEC* is $O\left((rn + m)n^2\right)$, or $O\left(mn^2\right)$ if $\gamma(\cdot)$ is not used.

[3]Features selection is accomplished by choosing the desired number of features from the returned feature list.

[4]For $\widehat{\varphi}_3$, using Arnoldi method to calculate a few eigenpairs of a large sparse matrix needs roughly $O(n^2)$ operations, and calculating $\widehat{\varphi}_3$ itself needs $O(k)$ operations.

3.3. Spectral Feature Selection via *SPEC*

The framework *SPEC* allows for different similarity matrix measures, $\gamma(\cdot)$, and ranking function $\widehat{\varphi}(\cdot)$. It can generate a range of spectral feature selection algorithms for both unsupervised and supervised learning. Hence, *SPEC* is a general framework for feature selection. To demonstrate the generality and usage of the framework, we show that (1) some existing powerful feature selection algorithms, such as ReliefF and Laplacian Score, can be derived from *SPEC* as special cases, and (2) novel spectral feature selection algorithms can be derived from *SPEC* conveniently.

Connections to Existing Algorithms

The connections of the framework to Laplacian Score and ReliefF are shown in the following two theorems.

Theorem 4 *Unsupervised feature selection algorithm Laplacian Score (He et al., 2005) is a special case of SPEC, by setting* $\widehat{\varphi}(\cdot) = \widehat{\varphi}_2(\cdot)$, $\gamma(\mathcal{L}) = \mathcal{L}$ *and using weighted k-nearest neighborhood graph obtained from RBF kernel function for measuring similarities.*

PROOF: It suffices to show that the ranking function used by Laplacian Score is equivalent to $\widehat{\varphi}_2(\cdot)$, with $\gamma(\mathcal{L}) = \mathcal{L}$. The ranking function of Laplacian score is:

$$S_L = \frac{\widetilde{\boldsymbol{f}}^T L \widetilde{\boldsymbol{f}}}{\widetilde{\boldsymbol{f}}^T D \widetilde{\boldsymbol{f}}}, \text{ where } \widetilde{\boldsymbol{f}} = \boldsymbol{f} - \frac{\boldsymbol{f}^T D \mathbf{e}}{\mathbf{e}^T D \mathbf{e}} \mathbf{e}.$$

Substituting $\widetilde{\boldsymbol{f}}$ in S_L and applying Theorem 1:

$$S_L = \frac{\boldsymbol{f}^T L \boldsymbol{f}}{\boldsymbol{f}^T D \boldsymbol{f} - \frac{(\boldsymbol{f}^T D \mathbf{e})^2}{\mathbf{e}^T D \mathbf{e}}}$$

$$= \frac{(D^{\frac{1}{2}}\boldsymbol{f})^T \mathcal{L} (D^{\frac{1}{2}}\boldsymbol{f})}{(D^{\frac{1}{2}}\boldsymbol{f})^T (D^{\frac{1}{2}}\boldsymbol{f}) - \frac{\left((D^{\frac{1}{2}}\boldsymbol{f})^T (D^{\frac{1}{2}}\mathbf{e})\right)^2}{(D^{\frac{1}{2}}\mathbf{e})^T (D^{\frac{1}{2}}\mathbf{e})}}$$

As $\xi_0 = \frac{D^{\frac{1}{2}}\mathbf{e}}{||D^{\frac{1}{2}}\mathbf{e}||}$ and $\widehat{\boldsymbol{f}} = \frac{D^{\frac{1}{2}}\boldsymbol{f}}{||D^{\frac{1}{2}}\boldsymbol{f}||}$, we have:

$$S_L = \frac{\widehat{\boldsymbol{f}}^T \mathcal{L} \widehat{\boldsymbol{f}}}{1 - \widehat{\boldsymbol{f}}^T \xi_0} = \widehat{\varphi}_2(\cdot) \qquad \square$$

Theorem 5 *Assuming the training data has c classes with t instances in each class and all features have been normalized to have unit norm. Supervised feature selection algorithm ReliefF (Robnik-Sikonja & Kononenko, 2003) is a special case of SPEC by setting* $\widehat{\varphi}(\cdot) = \widehat{\varphi}_1(\cdot)$, $\gamma(\mathcal{L}) = \mathcal{L}$ *and defining W as:*

$$w_{i,j} = \begin{cases} 1 & i = j \\ \frac{1}{k} & i \neq j,\ Y_i = Y_j,\ x_j \in \text{KNN}(x_i) \\ \frac{-1}{(c-1)k} & i \neq j,\ Y_i \neq Y_j,\ x_j \in \text{KNN}(x_i) \end{cases} \tag{12}$$

Here, $x_j \in \text{KNN}(x_i)$ *indicates* x_j *is one of the k nearest neighbors of* x_i*.*

PROOF: Under the assumptions, the feature ranking function of ReliefF is equivalent to:

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{k} \frac{1}{k} (f_i - f_{H_j})^2 - \sum_{CL \neq y_i} \frac{\sum_{j=1}^{k} (f_i - f_{M(CL)_j})^2}{(c-1)k} \right)$$

Here we use Euclidean distance to calculate the difference between two feature values and use all training data to train ReliefF. According to the design of W, it is easy to verify that $D = I$. Using Theorem 1, we can show that $\boldsymbol{f}^T L \boldsymbol{f}$ is equivalent to the above ranking function up to a constant factor. Since all features are norm 1 and $D = I$, we have $\widehat{\boldsymbol{f}} = \boldsymbol{f}$. $\square$

Note that the similarity matrix defined in Theorem 5 is not positive definite. Therefore the first eigenvalue of $\mathcal{L}$ is not 0, which may cause $\widehat{\varphi}_2(\cdot)$ and $\widehat{\varphi}_3(\cdot)$ fail. Theorems 4 and 5 establish connection between the two seemingly different feature selection algorithms by showing that they all try to select features that provide the best separability for data according to the given instance similarities.

Deriving Novel Algorithms from *SPEC*

The framework can also be used to systematically derive novel spectral algorithm by using different $\mathbb{S}$, $\gamma(\cdot)$ and ranking functions $\widehat{\varphi}(\cdot)$. For example, given the options of $\mathbb{S}$, $\gamma(\cdot)$ in Table 1, we can generate families of new supervised and unsupervised feature selection algorithms. We will conduct experiments to evaluate how effective these algorithms are and how they fare in comparison with the baseline algorithms: Laplacian Score and ReliefF.

$\widehat{\varphi}(\cdot)$	$\widehat{\varphi}_1(\cdot)$, $\widehat{\varphi}_2(\cdot)$ and $\widehat{\varphi}_3(\cdot)$
$\gamma(\cdot)$	$\gamma(r) = r$, $\gamma(r) = r^4$
$\mathbb{S}_u$	RBF kernel function, Diffusion kernel function
$\mathbb{S}_s$	Similarity matrix defined in Equations 2 and 12

Table 1. The components for *SPEC* tried in this paper. $\mathbb{S}_u$ and $\mathbb{S}_s$ stand for $\mathbb{S}$ used in unsupervised and supervised feature selection respectively.

4. Empirical Study

We empirically evaluate the performance of *SPEC*. In the experiments, we compared the algorithms specified in Table 1 with Laplacian Score (unsupervised) and ReliefF (supervised). Laplacian Score and ReliefF are both state-of-the-art feature selection algorithms, comparing with them enables us to examine

the efficacy of the algorithms derived from *SPEC*. We implement *SPEC* with the spider toolbox[5].

4.1. Data Sets

Four benchmark data sets are used for experiments: HOCKBASE[6] , RELATHE[6] , PIE10P[7] and PIX10P[8].

HOCKBASE & RELATHE are text data sets generated from the 20-new-group data: BASEBALL *vs.* HOCKEY (HOCKBASE) and (2) RELIGION *vs.* ATHEISM (RELATHE). PIE10P & PIX10P are face image data sets containing 10 persons in each. And we subsample the images down to a size of $100 \times 100 = 10000$.

Data Set	Instance	Feature	Classes
HOCKBASE	1993	8298	2
RELATHE	1427	8298	2
PIE10P	210	10000	10
PIX10P	100	10000	10

Table 2. Summary of four benchmark data sets

4.2. Evaluation of Selected Features

We apply 1-nearest-neighbor (1NN) classifier on data sets with selected features, and use its accuracy to measure the quality of the feature set. All results reported in the paper are obtained by averaging the accuracy from 10 trials of experiments.

Study of Unsupervised Cases

In the experiment, we use weighted 10-nearest neighborhood graph to represent the similarity among instances. The first two columns of Figure 2 are the 8 plots of accuracy vs. different numbers of selected features, different ranking functions, different $\gamma(\cdot)$ and different similarity measures. As shown in the plots, in most cases, the majority of the algorithms proposed in Table 1 work better than Laplacian score. Using the diffusion kernel function (Kondor & Lafferty, 2002), *SPEC* achieves better accuracy than using RBF kernel function. Generally, the more features we select, the better accuracy we can achieve. However, in many cases, this trend is less pronounced when more than 40 features are selected. Using $\widehat{\varphi}_2(\cdot)$, $\mathcal{L}$ and the *RBF* kernel function, *SPEC* works exactly the same as Laplacian Score, as expected in our theoretical analysis. Table 3 shows the accuracy when 100 features are selected: the differences between Laplacian Score and the algorithms performing best on each data set are: 0.17 for HOCKBASE, 0.08 for RELATHE, 0.18 for PIE10P and 0.06 for PIX10P. A trend can be observed in the table is that $\widehat{\varphi}_1(\cdot)$ performs well on the two text data sets, which contain binary classes; $\widehat{\varphi}_3(\cdot)$ performances well on the two face image data sets, which contain 10 different classes, while $\widehat{\varphi}_2(\cdot)$ works robustly in both cases. We also observed that comparing with using $\mathcal{L}$, using $\mathcal{L}^4$ never downgrades performance significantly, while in certain cases, applying it can improve the performance by a big margin. We averaged the accuracy over different numbers of selected features, different data sets, different similarity measures. Results show that $(\widehat{\varphi}_2, r^4)$ works best on benchmark data sets with an averaged accuracy of 0.69 which is followed by $(\widehat{\varphi}_3, r^4)$ with an average accuracy of 0.68. The averaged accuracy of Laplacian score is 0.62.

HOCKBASE (**Laplacian Score** = 0.58)						
	$\widehat{\varphi}_1, r$	$\widehat{\varphi}_1, r^4$	$\widehat{\varphi}_2, r$	$\widehat{\varphi}_2, r^4$	$\widehat{\varphi}_3, r$	$\widehat{\varphi}_3, r^4$
RBF	**0.61**	**0.61**	0.58	0.58	0.60	0.60
DIF	0.74	0.74	**0.75**	0.74	0.70	0.70
RELATHE (**Laplacian Score** = 0.59)						
	$\widehat{\varphi}_1, r$	$\widehat{\varphi}_1, r^4$	$\widehat{\varphi}_2, r$	$\widehat{\varphi}_2, r^4$	$\widehat{\varphi}_3, r$	$\widehat{\varphi}_3, r^4$
RBF	**0.63**	**0.63**	0.59	0.59	0.55	0.55
DIF	**0.67**	**0.67**	**0.67**	**0.67**	0.61	0.61
PIE10P (**Laplacian Score** = 0.74)						
	$\widehat{\varphi}_1, r$	$\widehat{\varphi}_1, r^4$	$\widehat{\varphi}_2, r$	$\widehat{\varphi}_2, r^4$	$\widehat{\varphi}_3, r$	$\widehat{\varphi}_3, r^4$
RBF	0.75	0.75	0.74	0.78	**0.87**	0.86
DIF	0.81	0.81	**0.92**	0.91	0.91	0.91
PIX10P (**Laplacian Score** = 0.88)						
	$\widehat{\varphi}_1, r$	$\widehat{\varphi}_1, r^4$	$\widehat{\varphi}_2, r$	$\widehat{\varphi}_2, r^4$	$\widehat{\varphi}_3, r$	$\widehat{\varphi}_3, r^4$
RBF	0.78	0.78	0.88	**0.94**	0.93	0.91
DIF	0.79	0.79	0.84	0.85	**0.93**	0.92

Table 3. Study of unsupervised cases: comparison of accuracy with 100 selected features. DIF stands for diffusion kernel, and bold typeface indicates the best accuracy.

Study of Supervised Cases

Due to the space limit, we skip $\gamma(\cdot) = r^4$ for supervised feature selection[9]. The last column of Figure 2 shows the 4 plots for supervised feature selection. Table 4 shows the accuracy with 100 selected features. From the results we can observe (1) generally supervised feature selection algorithms perform better than unsupervised feature selection algorithms, as they use label information; and (2) $\widehat{\varphi}_2(\cdot)$ works robustly with both similarity matrix. We averaged the accuracy over different numbers of selected features and different data sets. Results show that using $\widehat{\varphi}_2$ and the similarity matrix defined in Equation 2, *SPEC* performs best with

[5]http://www.kyb.tuebingen.mpg.de/bs/people/spider/
[6]http://people.csail.mit.edu/jrennie/20Newsgroups/
[7]http://www.ri.cmu.edu/projects/project 418.html
[8]http://peipa.essex.ac.uk/ipa/pix/faces/manchester/

[9]Also, the similarity matrix defined in Equation 2 is a block matrix with rank k. It can be verified that in this case the first k eigenvalues of $\mathcal{L}$ are 0 and the others are 1. Hence, varying $\gamma(\cdot)$ will not affect the values of the three feature ranking functions.

averaged accuracy of 0.82, followed by ReliefF whose accuracy is 0.81.

Based on our experimental results and observations, we offer the following guidelines for configuring *SPEC*. For data with a small number of classes, use φ_1, otherwise use φ_3, while φ_2 is robust in both cases. φ_3 is suitable for noisy data, and modifying the spectrum with an increasing function also helps remove noise.

RLF	R, $\widehat{\varphi}_1$	R, $\widehat{\varphi}_2$	R, $\widehat{\varphi}_3$	C, $\widehat{\varphi}_1$	C, $\widehat{\varphi}_2$	C, $\widehat{\varphi}_3$
0.74	0.74	**0.80**	0.57	0.69	0.78	0.63
0.73	**0.73**	0.68	0.53	0.65	**0.73**	0.64
0.97	0.97	0.97	**0.99**	0.97	0.97	0.98
0.97	**0.97**	**0.97**	0.82	0.95	**0.97**	**0.97**

Table 4. Study of supervised cases: comparison of accuracy with 100 selected features. The rows are the results for HOCKBASE, RELATHE, PIE10P and PIX10P, respectively. RLF stands for ReliefF, C and R stand for the similarity matrix defined in Equations 2 and 12 respectively.

5. Discussions and Conclusions

Feature selection algorithms can be either supervised or unsupervised (Liu & Yu, 2005). Recently, an increasing number of researchers paid attention to developing unsupervised feature selection. One piece of work closely related to this work is (Wolf & Shashua, 2005). The authors proposed an unsupervised feature selection algorithm based on iteratively calculating the soft cluster indicator matrix and the feature weight vector. They then extended the algorithm to handling data with class labels. Since the input of the algorithm restricts to covariance matrix, it does not handle general similarity matrix and cannot be extended as a general framework for designing new feature selection algorithms and covering existing algorithms.

In this paper, we propose a general framework of spectral feature selection for both supervised and unsupervised learning, which facilitates the joint study of supervised and unsupervised feature selection. We show that some powerful existing feature selection algorithms can be derived as special cases from the framework; and families of new effective algorithms can also be derived. Extensive experiments exhibit the generality and usability of the proposed framework. Our work is based on general similarity matrix. It is natural to extend the framework with existing kernel and metric learning methods (Lanckriet et al., 2004) to a variety of applications. Another line of our future work is to study semi-supervise feature selection (Zhao & Liu, 2007) using this framework.

6. Acknowledgements

The authors would like to express their gratitude to Jieping Ye for his helpful suggestions.

References

Chapelle, O., Schölkopf, B., and Zien, A. (Eds.) (2006). *Semi-supervised learning*, chapter Graph-Based Methods. The MIT Press.

Chung, F. (1997). *Spectral graph theory*. AMS.

Dy, J., & Brodley, C. E. (2004). Feature selection for unsupervised learning. *JMLR.*, *5*, 845–889.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *JMLR.*, *3*, 1157–1182.

He, X., Cai, D., & Niyogi, P. (2005). Laplacian score for feature selection. In *NIPS*. MIT Press.

Kondor, R. I., & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete structures. *ICML*.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *JMLR.*, *5*, 27–72.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE TKDE*, *17*, 491–502.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *NIPS*.

Lehoucq, R. B. (2001). Implicitly Restarted Arnoldi Methods and Subspace Iteration. SIAM J. Matrix Anal. Appl. *23*, 551–562.

Robnik-Sikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of Relief and ReliefF. *Machine Learning*, *53*, 23–69.

Shi, J., & Malik, J. (1997). Normalized cuts and image segmentation. *CVPR*.

Smola, A., & Kondor, I. (2003). Kernels and regularization on graphs. *COLT*.

Wolf, L., & Shashua, A. (2005). Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *JMLR.*, *6*, 1855–1887.

Zhang, T., & Ando, R. (2006). Analysis of spectral kernel design based semi-supervised learning. *NIPS*.

Zhao, Z., & Liu, H. (2007). Semi-supervised Feature Selection via Spectral Analysis. *SDM*.

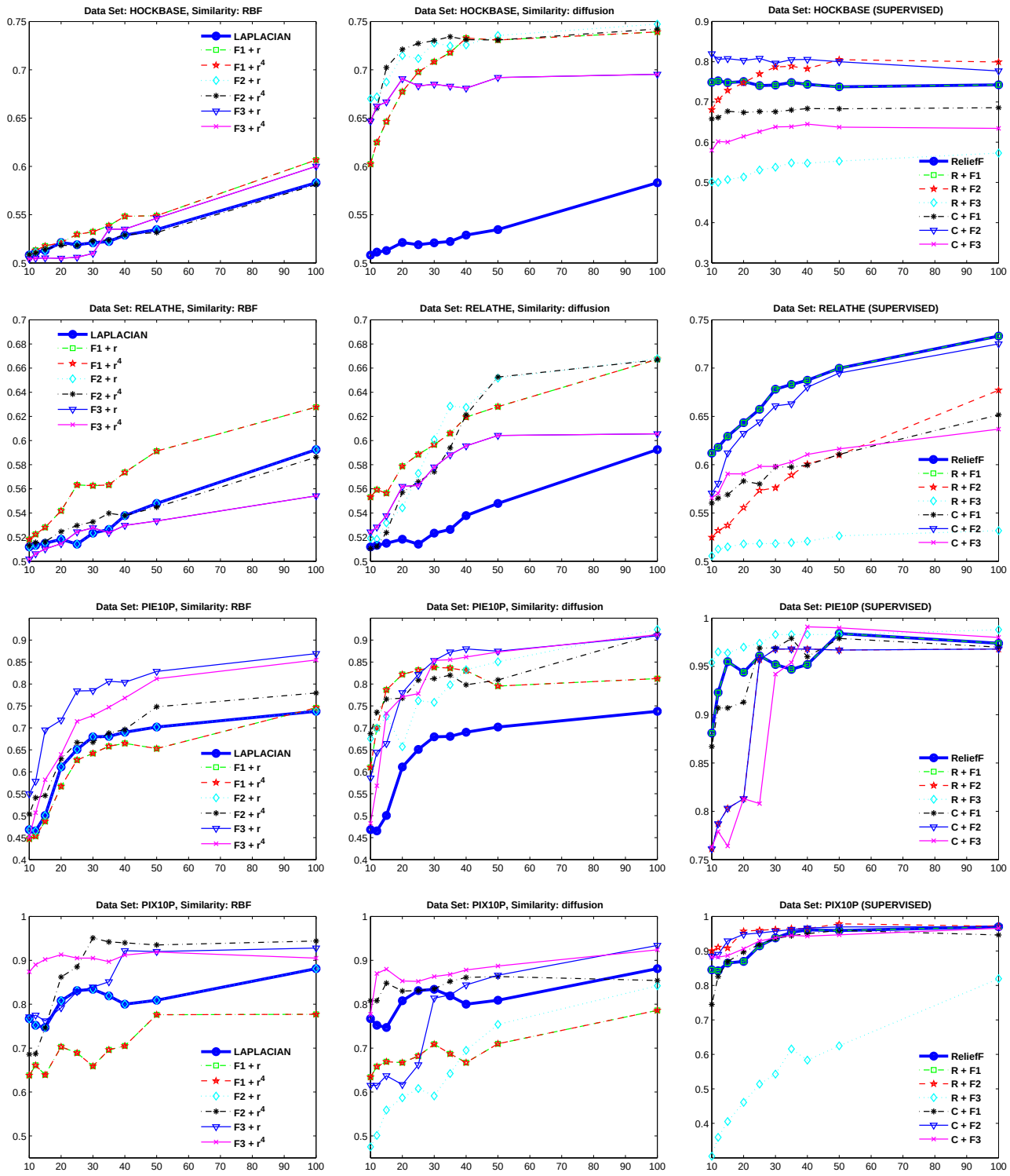

Figure 2. Accuracy (X axis) vs. different numbers of selected features (Y axis), different feature ranking functions, different $\gamma(\cdot)$ and different similarity measures. Each row stands for a different data set. The first two columns are for unsupervised feature selection and the last column is for supervised feature selection. **Thick lines in the figures are for the baseline feature selection algorithms: Laplacian Score and ReliefF**. In the legend, $F1$, $F2$ and $F3$ stand for feature ranking function $\widehat{\varphi}_1(\cdot)$, $\widehat{\varphi}_2(\cdot)$ and $\widehat{\varphi}_3(\cdot)$; C and R stand for the similarity matrix defined in Equations 2 and 12.