**Report No. UIUCDCS-R-2006-2749** **UILU-ENG-2006-1789**

# Semi-Supervised Regression using Spectral Techniques

by

**Deng Cai, Xiaofei He, and Jiawei Han**

July 2006

# Semi-Supervised Regression using Spectral Techniques*

Deng Cai[†] Xiaofei He[‡] Jiawei Han[†]

[†] Department of Computer Science, University of Illinois at Urbana-Champaign

[‡] Yahoo! Research Labs

## Abstract

Graph-based approaches for semi-supervised learning have received increasing amount of interest in recent years. Despite their good performance, many pure graph based algorithms do not have explicit functions and can not predict the label of unseen data. Graph regularization is a recently proposed framework which incorporates the intrinsic geometrical structure as a regularization term. It can be performed as semi-supervised learning when unlabeled samples are available. However, our theoretical analysis shows that such approach may not be optimal for multi-class problems. In this paper, we propose a novel method, called **Spectral Regression** (SR). By using spectral techniques, we first compute a set of responses for each sample which respects both the label information and geometrical structure. Once the responses are obtained, the ordinary ridge regression can be apply to find the regression functions. Our proposed algorithm is particularly designed for multi-class problem. Experimental results on two real world classification problems arising in visual and speech recognition demonstrate the effectiveness of our algorithm.

## 1 Introduction

The problem of learning from labeled and unlabeled data (*transductive* learning and *semi-supervised* learning) has attracted an increasing amount of attention in last decades [20], [7]. Recently, there are considerable interest and succuss on graph based semi learning algorithms [3], [6], [8], [12], [13], [22], [24], [25]. These approaches consider the graph over all the samples as a prior to guide the decision making. However, many of these algorithms can only work in a transductive setting and do not naturally extend to the semi-supervised case where novel test examples need to be classified.

Graph based learning methods are originally introduced in clustering and dimensionality reduction which are usually referred as spectral clustering [1], [15], [21] and spectral dimensionality reduction [17], [18], [2]. These spectral methods use information contained in the eigenvectors of a data affinity (i.e.,

* The work was supported in part by the U.S. National Science Foundation NSF IIS-03-08215/IIS-05-13678. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

item-item similarity) matrix to detect structure. Particularly, in $c$ classes case, at least $c-1$ eigenvectors are needed to capture the class structure [1], [15].

Recently, Belkin *et al.* [4] proposed a graph regularization framework for semi-supervised learning. In this framework, a nearest neighbor graph is constructed to model the intrinsic geometrical structure of the data space. The graph structure is then incorporated into the classification problem as a regularization term. As a result, many of the existing inductive algorithms (like SVM, Regression) can be extended to semi-supervised learning by adding this geometrically based regularization term. Since these algorithms are all inductive, they do not suffer from the out-of-sample extension problem [5]. Taking the graph structure as a regularization term is essentially similar to use only the largest eigenvector of the affinity matrix. However, in multi-class situations, the nearest neighbor graph constructed from unlabeled data is multi-class structured in nature. Previous studies in spectral clustering show that multiple eigenvectors are required to characterize the spectral properties of such graphes [1], [15]. Thus, treating the graph as a regularization term is not optimal for multi-class problem.

In data mining, one often confronts with huge amount of data. The efficiency is a key issue for data mining applications and an efficient classification algorithm is always expected. The least square classifier, also called regression in statistics, is one of the fastest classification algorithm. In text classification, it has been shown that ridge regression can get comparable performance to SVM [23], [14].

In this paper, we show that how to combine the ordinary regression with the spectral graph analysis to produce a uniform approach for learning from examples (labeled or unlabeled). We call this approach *Spectral Regression* (SR). Specifically, the labeled samples, combined with the unlabeled samples, are used to build a graph incorporating label information as well as neighborhood information of the data set. The spectral embeddings of the graph naturally characterize these information and are used as responses in the following regression.

The points below highlight several aspects of our approach:

1. Our algorithm provides a uniform approach for learning from examples. When all the samples are labeled, SR gives a solution of Regularized Discriminant Analysis [10]. When all the samples are unlabeled, SR gives a natural out-of-sample extension (linear or kernel) for many spectral clustering and embedding techniques [2][15].

2. In multi-class situation, the nearest neighbor graph constructed from unlabeled data is naturally multi-class structured. Previous studies show that multiple eigenvectors are required to characterize the spectral properties of such graphes [9][15]. Thus, those approaches which take the graph as a regularization term fail to handle multi-class problem. Our algorithm is designed to naturally handle such multi-class problem.

3. Since the neatest neighbor graph is sparse, the computation of our algorithm can be very efficient.

4. Our algorithm can be performed either in linear space or reproducing kernel Hilbert space (RKHS) which leads to kernel spectral regression.

The rest of this paper is organized as follows. In Section 2, we provide a brief review of regression. We introduce our spectral regression algorithm for semi-supervised learning in Section 3. The theoretical analysis of our algorithm is provided in Section 4. The experimental results are presented in Section 5. Finally, we conclude the paper and provide suggestions for future work in Section 6.

# 2 A Brief Review of Regression

In this section, we give a brief review of regression and different kinds of regularization, particularly, graph regularizer for semi-supervised learning.

## 2.1 Linear Regression

Suppose we have $m$ data points with responses $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$. Let $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$. Linear regression aims to fit a function

$$f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b$$

such that the residual sum of square is minimized:

$$RSS(\mathbf{a}) = \sum_{i=1}^{m} \big(f(\mathbf{x}_i) - y_i\big)^2 \tag{1}$$

For the sake of simplicity, we append a new element "1" to each $\mathbf{x}_i$. Thus, the coefficient $b$ can be absorbed into $\mathbf{a}$ and we have $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$. Let $\mathbf{y} = [y_1, \cdots, y_m]^T$. We have

$$RSS(\mathbf{a}) = \big(X^T\mathbf{a} - \mathbf{y}\big)^T\big(X^T\mathbf{a} - \mathbf{y}\big)$$

Requiring $\partial RSS(\mathbf{a})/\partial\mathbf{a} = 0$, we obtain:

$$\mathbf{a} = (XX^T)^{-1}X\mathbf{y} \tag{2}$$

When the number of features is greater than the number of samples, the matrix $XX^T$ is singular and the problem (1) is *ill posed*. A possible solution is to impose a penalty on the norm of $\mathbf{a}$:

$$RSS_{ridge}(\mathbf{a}) = \sum_{i=1}^{m} \big(\mathbf{a}^T\mathbf{x}_i - y_i\big)^2 + \alpha\|\mathbf{a}\|^2 \tag{3}$$

The solution to Eq. (3) is given below:

$$\mathbf{a} = \big(XX^T + \alpha I\big)^{-1}X\mathbf{y} \tag{4}$$

where $I$ is a $n \times n$ identity matrix. It is clear to see that $XX^T + \alpha I$ is no longer singular. The term $\|\mathbf{a}\|^2$ in Eq. (3) is called Tikhonov regularizer [19]. In statistics, such regression is called *ridge regression* [11].

### 2.2 Regression with Graph Regularization for Semi-Supervised Learning

The Tikhonov regularizer in ridge regression is data independent. For different applications, we can also develop data dependent regularizer [10]. Recently, Belkin *et al.* [4] proposed a graph regularization framework which allow us to exploit the geometry of the marginal distribution. In semi-supervised situation, the nearest neighbor graph (graph Laplacian [9][2]) can be used to estimate the geometric marginal distribution. Thus, regression with graph regularizer can naturally be used for semi-supervised learning.

Given a set of $l$ labeled examples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and a set of $m-l$ unlabeled examples $\{\mathbf{x}_i\}_{i=l+1}^{m}$, we try to estimate an unknown function by minimizing

$$\begin{aligned} f^* =& \arg\min\Bigg(\sum_{i=1}^{l}\big(f(\mathbf{x}_i)-y_i\big)^2+\alpha\|f\|^2 \\ & + \ \beta\sum_{i,j=1}^{m}\big(f(\mathbf{x}_i)-f(\mathbf{x}_j)\big)^2W_{ij}\Bigg) \\ =& \arg\min\sum_{i=1}^{l}\big(f(\mathbf{x}_i)-y_i\big)^2+\alpha\|f\|^2+\beta\mathbf{f}^TL\mathbf{f} \end{aligned} \tag{5}$$

where $W_{ij}$ are edge weights in the data adjacency graph, $\mathbf{f}=[f(\mathbf{x}_1),\cdots,f(\mathbf{x}_m)]^T$, and $L$ is the graph Laplacian given by $L=D-W$. $D$ is a diagonal matrix with its $(i,i)$-element equals to the sum of the $i$-th column of $W$.

Since $W$ is the weight matrix of the adjacency graph, bigger weight of $W_{ij}$, closer of samples $\mathbf{x}_i$ and $\mathbf{x}_j$. Minimizing the graph regularizer

$$\sum_{i,j=1}^{m}\big(f(\mathbf{x}_i)-f(\mathbf{x}_j)\big)^2W_{ij}$$

is trying to find the function $f$ which has similar values for those close data samples. Thus, the graph regularizer is reasonable and natural for semi-supervised learning. The same idea also plays a key role in spectral dimensionality reduction and spectral clustering [2][15].

When a linear function $f(\mathbf{x})=\mathbf{a}^T\mathbf{x}$ is required, we have

$$\begin{aligned} \mathbf{a}^* =& \arg\min\sum_{i=1}^{l}\big(\mathbf{a}^T\mathbf{x}_i-y_i\big)^2+\alpha\|\mathbf{a}\|^2+\beta\mathbf{a}^TXLX^T\mathbf{a} \\ =& \Big(X_1X_1^T+\alpha I+\beta XLX^T\Big)^{-1}X_1\mathbf{y} \end{aligned} \tag{6}$$

where $X_1=[\mathbf{x}_1,\cdots,\mathbf{x}_l]$, $X=[\mathbf{x}_1,\cdots,\mathbf{x}_m]$ and $\mathbf{y}=[y_1,\cdots,y_l]^T$.

The experiments on several binary classification tasks showed the effectiveness of such graph regularizer for semi-supervised learning [4]. However, there are two drawbacks of this approach when we consider large scale, multi-class semi-supervised learning problems.

$$\begin{array}{cccccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

(a) $W$

$$\begin{array}{cccccc} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{array}$$

(b) $L$

Figure 1: The ideal $W$ and $L$ for six samples which belong to 3 classes

- The manifold regularization term tries to seek a function which minimize $\mathbf{f}^T L\mathbf{f}$. Apparently, such function will make $\mathbf{f}$ close to the minimum eigenvector of $L$. However, it has no discriminating power on different $\mathbf{f}$'s when these $\mathbf{f}$'s give the same value of $\mathbf{f}^T L\mathbf{f}$. Take the ideal $W$ of 6 samples in Figure 1(a) as an example. These 6 samples belong to 3 classes and the $L$ is shown in Figure 1(b). We would expect the ideal $\mathbf{f}$ has different values for different classes. Clearly, $[-1,-1,0,0,1,1]^T$ is such one. However, $[1,1,0,0,0,0]^T$,$[0,0,1,1,0,0]^T$ and $[0,0,0,0,1,1]^T$ all give the same minimum of $\mathbf{f}^T L\mathbf{f}$. Actually, all the vectors in the space spanned by these three vectors give the same minimum of $\mathbf{f}^T L\mathbf{f}$. When unlabeled data is incorporated into the graph, the graph might be intrinsically multi-class structured. However, a single eigenvector can never capture this multi-class structure.

- The explicit function $f$ can be either linear or a function in RKHS. A linear function can be represented as $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$, $\mathbf{a} \in \mathbb{R}^n$. A function in RKHS can be represented as $f(\mathbf{x}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i)$, where $K$ is a Mercer kernel. Thus, $\mathbf{f}^T L\mathbf{f}$ can be written as $\mathbf{a}^T XLX^T\mathbf{a}$ in linear case or $\boldsymbol{\alpha}^T KLK\boldsymbol{\alpha}$ in kernel case, where the $i$-th column of $X$ is $\mathbf{x}_i$ and $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_m]^T$. Although $L$ is usually a sparse matrix, the $XLX^T$ will be a $n \times n$ dense matrix and $KLK$ will be a $m \times m$ dense matrix. For high dimensional data like text, both these two matrices could be very large when a large amount of unlabeled examples exist. Such large dense matrices will cause problems in both memory storage and computation, which restricts such approach to extend to large scale.

These two observations motivate us to develop more efficient algorithm which can naturally handle large scale multi-class semi-supervised learning problem.

# 3 Regression with Spectral Analysis

In this section, we introduce a novel algorithm, called Spectral Regression, which combines the spectral graph analysis and ordinary regression to produce a uniform approach for learning from examples (labeled and unlabeled). We begin with a formal definition of the problem of semi-supervised learning.

### 3.1 The Problem

Given a labeled set $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l$ and an unlabeled set $\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \cdots, \mathbf{x}_m$ in $\mathbf{R}^n$. These samples are belong to $c$ class and $l_k$ is the number of labeled samples in the $k$-th class ($\sum_{k=1}^{c} l_k = l$). The goal is to find a classifier which can predict the label of both unlabeled data and unseen data.

It is important to note that we are expecting to predict the label of unseen data. Thus, an explicit function either linear or a function in RKHS is necessary. Many of the existing pure graph based approaches [3][24] fail to satisfy such requirement.

### 3.2 The Spectral Regression Algorithm

In Section 2.2, we give a brief analysis on graph regularizer to show that a single eigenvector can not capture the multi-class structure of the affinity matrix. Naturally, we need multiple eigenvectors.

When we use regression for supervised classification, the response $y_i$ of sample $\mathbf{x}_i$ can be set based on the label of $\mathbf{x}_i$. In semi-supervised situation, the problem becomes how to get the responses of unlabeled samples. In this paper, we argue that the eigenvectors of the affinity matrix can be used as the responses of regression. Such observation is the key of our spectral regression algorithm.

The algorithmic procedure of our spectral regression algorithm is stated below.

1. **Constructing the adjacency graph**: Let $G$ denote a graph with $m$ nodes. The $i$-th node corresponds to the sample $\mathbf{x}_i$. We construct the graph $G$ through the following three steps to model the local structure as well as the label information:

   (a) Put an edge between nodes $i$ and $j$ if $\mathbf{x}_i$ is among $p$ nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among $p$ nearest neighbors of $\mathbf{x}_i$.

   (b) Put an edge between nodes $i$ and $j$ if $\mathbf{x}_i$ shares the same label with $\mathbf{x}_j$.

   (c) Remove the edge between nodes $i$ and $j$ if the label of $\mathbf{x}_i$ is different with that of $\mathbf{x}_j$.

2. **Choosing the weights**: $W$ is a sparse symmetric $m \times m$ matrix with $W_{ij}$ having the weight of the edge joining vertices $i$ and $j$.

   (a) If there is no edge between $i$ and $j$, $W_{ij} = 0$.

   (b) Otherwise,

$$W_{ij} = \begin{cases} 1/l_k, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong} \\ & \text{to the } k\text{-th class;} \\ \delta \cdot s(i,j), & \text{otherwise.} \end{cases} \tag{7}$$

   where $\delta \leq 1$ is the parameter to adjust the weight between supervised information and unsupervised neighbor information. $s(i,j)$ is a function to evaluate the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ and we have three variations.

(a) Heat kernel. [parameter $\sigma \in \mathbb{R}$].

$$s(i,j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

(b) Cosine weight. [no parameter].

$$s(i,j) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_i\|}$$

(c) Simple-minded. [no parameter].

$$s(i,j) = 1$$

3. **Eigen-decomposing**: Find $\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{c-1}$, the largest $c$ generalized eigenvectors of eigen-problem (chosen to be orthogonal to each other in the case of repeated eigenvalues):

$$W\mathbf{y} = \lambda D\mathbf{y} \tag{8}$$

which $D$ is a diagonal matrix with its $(i,i)$-element equals to the sum of the $i$-th column (or row, since $W$ is symmetric) of $W$. It is straightforward to show that the first eigenvector is a vector of all ones with eigenvalue 1 [9].

4. **Regularized least squares**: Find $c-1$ vectors $\mathbf{a}_1, \cdots, \mathbf{a}_{c-1} \in \mathbb{R}^n$. $\mathbf{a}_k$ $(k = 1, \cdots, c-1)$ is the solution of regularized least square problem:

$$\begin{aligned} \mathbf{a}_k = \arg\min_{\mathbf{a}} \Bigg( & \sum_{i=1}^{l} (\mathbf{a}^T \mathbf{x}_i - y_i^k)^2 \\ & + \sum_{i=l+1}^{m} (\gamma \mathbf{a}^T \mathbf{x}_i - y_i^k)^2 + \alpha \|\mathbf{a}\|^2 \Bigg) \end{aligned} \tag{9}$$

where $y_i^k$ is the $i$-th element of $\mathbf{y}_k$ and $\gamma \leq 1$ is a parameter to adjust the weight of unlabeled samples. Let $\widetilde{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_l, \gamma\mathbf{x}_{l+1}, \cdots, \gamma\mathbf{x}_m]$, the above regularized least square problem can be rewrite as:

$$\mathbf{a}_k = \arg\min_{\mathbf{a}} \left( \|\widetilde{X}^T \mathbf{a} - \mathbf{y}_k\|^2 + \alpha \|\mathbf{a}\|^2 \right) \tag{10}$$

The close form solution of $\mathbf{a}_i$ is given by

$$\mathbf{a}_k = (\widetilde{X}\widetilde{X}^T + \alpha I)^{-1} \widetilde{X} \mathbf{y}_k \tag{11}$$

When $n$ is large, some efficient iterative algorithms [16] can be used to solve the above regularized least square problem.

5. **Classification in $c-1$ dimensional subspace**: Let $A = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_{c-1}]$, $A$ is a $n \times (c-1)$ transformation matrix. The samples can be embedded into $c-1$ dimensional subspace by

$$\mathbf{x} \rightarrow \mathbf{z} = A^T \mathbf{x} \tag{12}$$

The classification is based on the distances of a test example to the class centroids in the $c-1$ dimensional subspace.

# 4 Theoretical Analysis

In this section, we provide theoretical analysis of our algorithm. We first analyze our algorithm from two extreme cases: supervised setting and unsupervised setting. Then the properties of spectral regression in semi-supervised setting are analyzed.

## 4.1 Spectral Regression in Supervised Setting

In supervised situation, all the training samples are labeled, we have $l = m$. To simplify our exposition, we assume that the data points $\{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ are ordered according to their labels. Let $W^{(k)}$ be a $l_k \times l_k$ matrix with all the elements equal to $1/l_k$. We can easily see that the $W$ in the second step of our algorithm has block-diagonal structure

$$W = \begin{bmatrix} W^{(1)} & 0 & \cdots & 0 \\ 0 & W^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W^{(c)} \end{bmatrix}$$

Thus, we have $D = I$ and our generalized eigen-problem (8) reduces to an ordinary eigen-problem

$$W\mathbf{y} = \lambda\mathbf{y} \tag{13}$$

Since the $W$ is block-diagonal, its eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks (the latter padded appropriately with zeros). It is straightforward to show that $W^{(k)}$ has eigenvector $\mathbf{e}^{(k)} \in \mathbb{R}^{l_k}$ associated with eigenvalue 1, where $\mathbf{e}^{(k)} = [1, 1, \cdots, 1]^T$. Also $Rank(W^{(k)}) = 1$, there is only one non-zero eigenvalue of $W^{(k)}$. Thus, there are exactly $c$ eigenvectors of $W$ with the same eigenvalue 1. These eigenvectors are

$$\mathbf{y}_k = [\ \underbrace{0, \cdots, 0}_{\sum_{i=1}^{k-1} l_i}, \underbrace{1, \cdots, 1}_{l_k}, \underbrace{0, \cdots, 0}_{\sum_{i=k+1}^{c} l_i}\ ]^T \quad k = 1, \cdots, c \tag{14}$$

If the above $\{\mathbf{y}_k\}$ are taken as the responses for regression, we can immediately see that it is the *one-against-rest* approach for extending a binary classifier to multi-class case.

Actually, a subtlety needs to be addressed here. Since 1 is a repeated eigenvalue of $W$, we could just pick any other $c$ orthogonal vectors in the space spanned by $\{\mathbf{y}_k\}$, and defined them to be our $c$ eigenvectors. The vector of all ones is naturally in the spanned space. Such vector is useless since the responses of all the data points are the same. In reality, we can always pick the vector of all ones as our first eigenvector and use Gram-Schmidt process to orthogonalize the remaining eigenvectors. The vector of all ones then can be removed which leaves us $c - 1$ response vectors.

It is also important to note that the value of $i$-th and $j$-th position of any vector $\mathbf{y}$ in the space spanned by $\{\mathbf{y}_k\}$ will have the same value as long as $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class.

The above analysis shows that in supervised setting, our algorithm is essentially similar to extending ridge regression, a binary classifier, to multi-class situation with *one-against-rest* approach.

### 4.2 Spectral Regression in Unsupervised Setting

When all the data points are unlabeled, $W$ in the second step of our algorithm gives us a $p$-nearest neighbor graph. Let $L = D - W$, it is easy to check that the largest eigenvectors of eigen-problem (8) will be equivalent to the smallest eigenvectors of $L\mathbf{y} = \lambda' D\mathbf{y}$ since

$$\begin{aligned} & W\mathbf{y} = \lambda D\mathbf{y} \\ \Rightarrow\ & (D - L)\mathbf{y} = \lambda D\mathbf{y} \\ \Rightarrow\ & L\mathbf{y} = (1-\lambda)D\mathbf{y} = \lambda' D\mathbf{y} \end{aligned} \tag{15}$$

Clustering by using the top eigenvectors of a matrix derive from the distance between points (largest eigenvectors of Eq. (8) or smallest eigenvectors of Eq. (15)), so called *spectral clustering methods*, recently received a lot of interests [1], [2], [21], [15]. But despite their empirical successes, there is still no theoretical proof on which eigenvectors to use and how to derive clusters from them. Those eigenvectors also play a key role in our algorithm, we provide here a informal analysis on how many eigenvectors we should use when there exist $c$ classes. Our analysis is based on graph partitioning.

We consider the problem of dividing the graph $\mathcal{G}$ into $c$ disjoint subgraphs, $\mathcal{G}_1, \cdots, \mathcal{G}_c$, such that $\mathcal{G} = \mathcal{G}_1 \cup \cdots \cup \mathcal{G}_c$ and $\mathcal{G}_k \cap \mathcal{G}_p = \emptyset,\ k \neq p$. The min-cut criteria can be stated as follows:

$$\min_{\mathcal{G}_1, \cdots, \mathcal{G}_c} \left( \sum_{i\in\mathcal{G}_1} \sum_{j\in\mathcal{G}_2} W_{ij} + \cdots + \sum_{i\in\mathcal{G}_{c-1}} \sum_{j\in\mathcal{G}_c} W_{ij} \right) \tag{16}$$

where $W$ is the weight matrix of graph $\mathcal{G}$. Now, let's recall the objective function of Lapalcian Eigenmaps [2], which is also the objective function for many spectral clustering algorithms:

$$\min \sum_{ij} \|Y_i - Y_j\|^2 W_{ij} \tag{17}$$

It is easy to verify that Eqn (17) gives the same solution of Eqn (16) as long as $Y_i$'s satisfy two conditions: (1) $\|Y_i - Y_j\| = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same subgraph; (2) $\|Y_i - Y_j\| = d$, $d$ is a constant for any $\mathbf{x}_i$ and $\mathbf{x}_j$ as long as they belong to different subgraph. To fulfil such requirement, we can immediately see the label vector $Y_i$'s reside on the vertices of the standard $(c-1)$-simplex, *i.e.*, $Y_i$ should be at least $c-1$ dimension. Let $Y = [Y_1, \cdots, Y_m]^T$. By relaxing the elements of $Y$ to take real values. The column vectors of optimal $Y$ will be given by the $c-1$ smallest eigenvectors of eigen-problem [2]:

$$L\mathbf{y} = \lambda D\mathbf{y}.$$

The above informal analysis shows that we need at least $c-1$ (after removing the first all ones trivial eigenvector) smallest eigenvectors of $L$ to reveal the $c$ class structure. Such analysis is consistent with previous study on spectral clustering [1][15]. Also, our analysis in the previous section shows that when the label information are available, we can use exactly $c-1$ eigenvectors to reveal the class structure.

It is also important to note that many spectral clustering algorithms [1], [15], [21] and spectral dimensionality reduction algorithm [2] do not have explicit embedding function, thus can not be applied to unseen data. Our approach provides natural out-of-sample extensions to Laplacian Eigenmap [2] and several other spectral clustering algorithms [15], [21].

### 4.3 Spectral Regression in Semi-Supervised Setting

Now we are ready to discuss some properties of our algorithm in semi-supervised situation. Particularly, we are interested in the impact of parameter $\delta$ (second step). We have the following proposition.

**Theorem 1** *Let $\boldsymbol{y}$ be one of the first $c$ largest eigenvectors of eigen-problem (8), $\boldsymbol{y} = [y_1, y_2, \cdots, y_m]^T$. Suppose two labeled points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ share the same label. For any $\eta > 0$, there exists $\delta$ (parameter in the second step of our algorithm) such that $|y_i - y_j| \leq \eta$.*

**Proof** Without loss of generality, we assume that the labeled data points $\{\mathbf{x}_1, \cdots, \mathbf{x}_l\}$ are ordered according to their labels, $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the first class and there is $l_1$ labeled points in the first class. The eigenvectors are normalized , *i.e.*, $\|\mathbf{y}\| = 1$.

The first $l_1$ elements in the $i$-th and $j$-th rows of $W$ are $1/l_1$ and the next $m - l_1$ elements are either zero or $\delta \cdot s(i,j)$ as defined in Eq. (7). It is easy to check the rank of $W$ is grater than $c$. Thus, $\mathbf{y}$ is an eigenvector with non-zero eigenvalue, $W\mathbf{y} = \lambda D\mathbf{y}$. We have

$$\begin{aligned} y_i &= \frac{1}{\lambda}\frac{1}{D_{ii}}\sum_{q=1}^{m} W_{iq}y_q \\ &= \frac{1}{\lambda}\frac{1}{D_{ii}}\left(\frac{1}{l_1}\sum_{q=1}^{l_1} y_q + \sum_{q=l_1+1}^{m} W_{iq}y_q\right) \\ &\qquad\qquad\qquad\qquad (1 \leq i \leq l_1) \end{aligned}$$

Since $s(i,j) \leq 1,\ (\forall\ i,j)$ and $\|\mathbf{y}\| = 1$, we have

$$\begin{aligned} &W_{iq} \leq \delta, \quad (q = l_1 + 1, \cdots, m), \\ &W_{jq} \leq \delta, \quad (q = l_1 + 1, \cdots, m), \\ &D_{ii} = \sum_q W_{iq} \leq 1 + (m - l_1)\delta, \\ &D_{jj} = \sum_q W_{jq} \leq 1 + (m - l_1)\delta, \\ &|y_i| \leq 1, \quad (i = 1, \cdots, m). \end{aligned}$$

Thus,

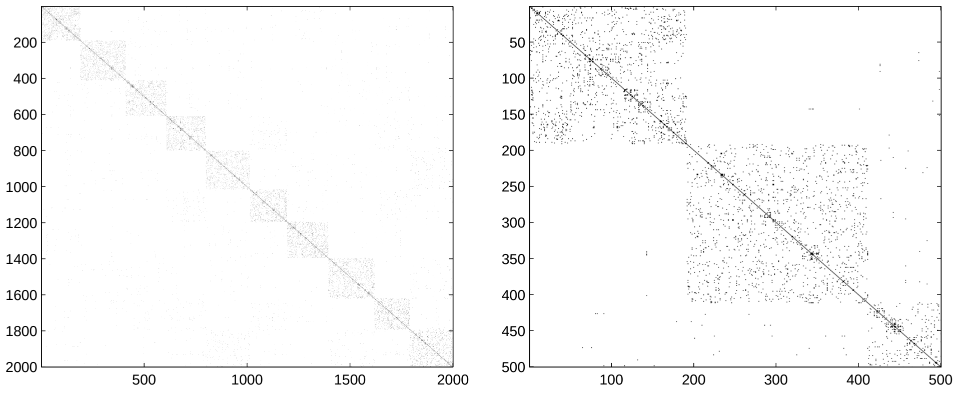
$$
\begin{aligned}
|y_i - y_j| &= \left| \frac{1}{\lambda}\frac{1}{l_1}\sum_{q=1}^{l_1} y_q \left(\frac{1}{D_{ii}} - \frac{1}{D_{jj}}\right) \right. \\
&\quad \left. + \frac{1}{\lambda}\left(\frac{1}{D_{ii}}\sum_{q=l_1+1}^{m} W_{iq}y_q - \frac{1}{D_{jj}}\sum_{q=l_1+1}^{m} W_{jq}y_q\right) \right| \\
&\leq \frac{1}{\lambda}\frac{1}{l_1}\sum_{q=1}^{l_1} |y_q| \left|\frac{D_{jj} - D_{ii}}{D_{ii}D_{jj}}\right| \\
&\quad + \frac{1}{\lambda}\frac{1}{D_{ii}}\sum_{q=l_1+1}^{m} W_{iq}|y_q| + \frac{1}{\lambda}\frac{1}{D_{jj}}\sum_{q=l_1+1}^{m} W_{jq}|y_q| \\
&\leq \frac{1}{\lambda}\frac{1}{l_1}\sum_{q=1}^{l_1} |y_q| \Big(2(m - l_1)\delta\Big) + \frac{2}{\lambda}\sum_{q=l_1+1}^{m} |y_q|\delta \\
&= \left(\frac{2(m - l_1)}{\lambda l_1}\sum_{q=1}^{l_1} |y_q| + \frac{2}{\lambda}\sum_{q=l_1+1}^{m} |y_q|\right)\delta \\
&\leq \left(\frac{4(m - l_1)}{\lambda}\right)\delta
\end{aligned}
$$

Given $\eta > 0$, we choose $\delta = \eta\lambda/4(m - l_1)$. Thus $|y_i - y_j| \leq \eta$.

The above theorem shows that when $\delta$ is sufficiently small, the responses of the labeled same class points can be as close as possible. This property is reasonable and important since for classification, either supervised or semi-supervised, we always expect the labeled same class samples have same response. One may ask how small of $\delta$ should enough. It mainly depends on the parameter $p$ of the $p$-nearest graph in the first step as well as the number of labeled points. In reality, when we use set $p = 5$, $\delta = 0.05$ is sufficiently small.

## 5 Experimental Results

In most of previous experiments for semi-supervised learning, the experimental setting is *transductive*. That is, both the training and test set (without label information) are available during the learning process. In reality, a more natural setting for semi-supervised learning is as follows. The available training set contains both labeled and unlabeled examples, and the testing set is not available during the training phrase, which we refer here as *semi-supervised setting*. In such case, many of recently proposed graph based semi-supervised learning algorithms [24][25] can not be applied since they do not have out-of-sample extension. The most recently work on manifold regularizer [4] provides a geometric framework for semi-supervised learning and has natural out-of-sample extension. LapRLS and LapSVM are two algorithms developed in this framework. Both of these two algorithms have the similar strategy to use the unlabeled data and achieves similar results.



(a)

(b)

Figure 2: 5 nearest neighbor graph of 2000 handwritten digit data examples (The first 2000 examples in MNIST training set). There are around 200 examples of each digit, ordered based on there labels. (a) the whole graph; (b) the subgraph of the first 500 data examples. Clearly, the nearest neighbor graph is multi-class structured and reflects the class relationships of the data points to some extent.

In this section, we compared our Spectral Regression (SR) with LapRLS under semi-supervised setting on two real world classification problems arising in visual and speech recognition. For comparison reason, we also show the result of Regularized Least Square (RLS), which will only use labeled data.

## 5.1 Adjacency Graph to Illustrate the Intrinsic Data Structure

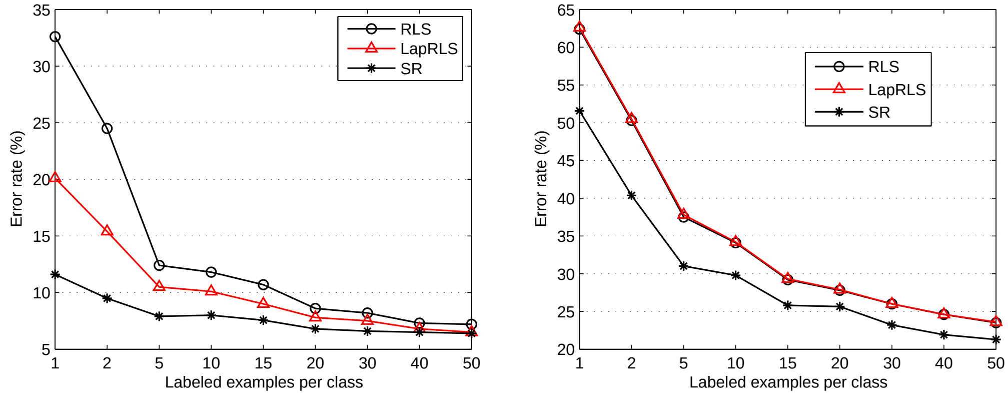
In our spectral regression algorithm, we use an adjacency graph to model the manifold structure as well as the label information. A good adjacency graph should be able to reflect the class relationships of the data points to some extent. Figure (2) shows the 5 nearest neighbor graph of 2000 handwritten digit data examples (The first 2000 examples in MNIST training set). Clearly, the block structure (multi-class structure) of the graph illustrates the intrinsic class relationship of the data points to some extent.

## 5.2 Handwritten Digit Recognition

The MNIST handwritten digit database[1] was used in this experiment. The public MNIST database has a training set of 60,000 samples, and a testing set of 10,000 samples. In our experiment, we take the first 2,000 samples from the training set as our training set and the first 2,000 samples from the test set as our test set. In both training and test set, each digit has around 200 samples.

As we analyzed before, our algorithm is particularly suit for multi-class semi-supervised learning. Thus, we designed two experiments. In the first one, we take all the samples with number 8 and 9 to form a binary classification problem. In the second one, we classify all the 10 digits. LapRLS and RLS

[1] http://yann.lecun.com/exdb/mnist/



(a) Binary classification on digit 8 and 9

(b) Multi-class classification on all 10 digits

Figure 3: Error rate as a function of number of labeled samples on MNIST dataset.

will use the one against rest strategy.

A random subset with $l$ $(= 1, 2, 5, \cdots, 40, 50)$ samples per digit from the training set were labeled and the rest were left unlabeled. For each given $l$, we average the results over 20 random selection. In both LapRLS and SR, we first need to construct a $p$-nearest neighbor graph over all the training data. We set $p = 5$ in all our experiments. In SR, the parameter $\delta$ was set to 0.05. We use cross validation on the training set to select all the other parameters ($\alpha$ in all the algorithms, $\beta$ in LapRLS and $\gamma$ in SR).

In Figure 3(a), we compare three algorithms on the binary classification task (classify digit 8 and 9). The following comments can be made: (a) All these algorithms can take advantage of more labeled samples, which is important to the real-world classification task. (b) Both LapRLS and SR utilized the unlabeled samples and achieved significant improvement over the purely supervised algorithm RLS, especially when the number of labeled examples is small. (c) SR achieved better performance than LapRLS. It suggests that incorporating both label and neighborhood information in the same graph might be a better way for semi-supervised learning.

Figure 3(b) shows the performance of these algorithms on multi-class classification task. It is not surprising that LapRLS dose not show any improvement over RLS. In multi-class situation, the neighbor graph is usually multi-class structured. LapRLS takes the graph as a regularization term for decision function thus can not fully utilize the multi-class structure in the neighbor graph. Figure (4) shows the test errors of LapRLS as a function of $\beta$. The results failed to suggest the usefulness of the manifold regularizer in LapRLS for multi-class classification. In contrast to LapRLS, SR uses the $c-1$ eigenvectors of the graph which can fully captured the multi-class information.

### 5.3 Spoken Letter Recognition

This experiment was performed on the Isolet database (available from the UCI machine learning repository). The data set contains 150 subjects who spoke the name of each letter of the alphabet twice.

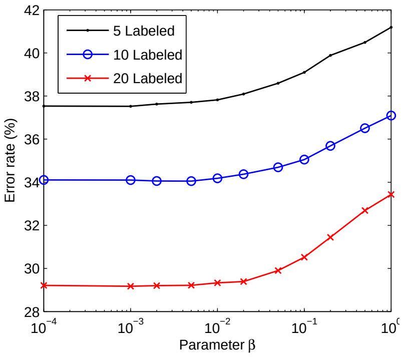
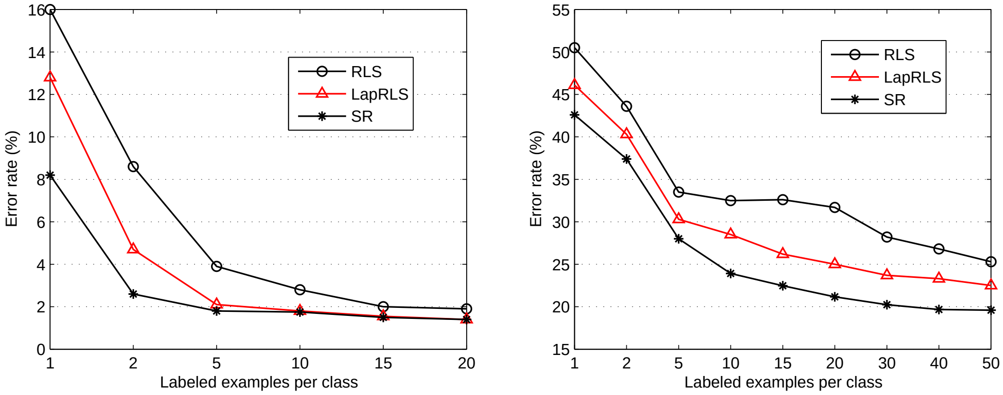


Figure 4: Test errors for the MNIST multi-class classification with 5, 10 and 20 labeled examples per digit, using LapRLS with a series of values of $\beta$. The results failed to suggest the usefulness of the manifold regularizer in LapRLS for multi-class classification on MNIST.



(a) Binary classification on alphabet A and B

(b) Multi-class classification on all 26 alphabets

Figure 5: Error rate as a function of number of labeled samples on Isolet dataset.

The speakers are grouped into sets of 30 speakers each, and are referred to as isolet1 through isolet5. For the purposes of this experiment, we chose isolet1 which contains 1560 examples (60 examples per class) as the training set, and test on isolet5 which contains 1559 examples (1 example is missing due to the difficulties in recording). The experimental setting is the same as before. A random subset with $l$ $(= 1, 2, 5, \cdots, 40, 50)$ examples per letter from the training set were labeled and the rest were left unlabeled. And for each given $l$, we average the results over 20 random selection.

Figure 5(a) shows the results on the binary classification task (classify alphabet A and B) and Figure 5(b) shows the results on the multi-class classification task. Both two semi-supervised algorithms LapRLS and SR utilized the unlabeled samples and achieved significant improvement over supervised RLS. SR achieves better performance than LapRLS indicates the effectiveness of our algorithm for semi-supervised learning.

# 6 Conclusion

In this paper, we propose a new semi-supervised learning algorithm called *Spectral Regression.* It combined the ordinary regression with spectral graph analysis to provides a uniform approach for learning from examples (labeled or unlabeled). Compared to most recent graph based semi-supervised learning algorithms, our approach has natural out-of-sample extension and can well handle multi-class problems.

Another major advantage of our proposed algorithm over previous graph based semi-supervised learning algorithm [4] is that the memory requirement of our algorithm is significantly less. In many application involving large scale data sets, like web page classification, previous graph based algorithms may fail due to the large scale dense matrix computation.

# References

[1] C. J. Alpert, A. B. Kahng, and S.-Z. Yao. Spectral partitioning with multiple eigenvectors. *Discrete Applied Mathematics*, 90:3–26, 1999.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, Cambridge, MA, 2001.

[3] M. Belkin and P. Niyogi. Using manifold structure for partially labeled classification. In *Advances in Neural Information Processing Systems 15.* MIT Press, Cambridge, MA, 2002.

[4] M. Belkin, P. Niyogi, and V. Sindwani. On manifold regularization. In *Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

[5] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*, 2003.

[6] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conf. on Machine Learning*, pages 19–26, 2001.

[7] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

[8] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems 16*, 2003.

[9] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics.* AMS, 1997.

[10] T. Hastie, A. Buja, and R. Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23:73–102, 1995.

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

[12] T. Joachims. Transductive learning via spectral graph partitioning. In *Proceeding of The Twentieth International Conference on Machine Learning*, 2003.

[13] A. Kapoor, Y. Qi, H. Ahn, and R. Picard. Hyperparameter and kernel learning for graph based semi-supervised classification. In *Advances in Neural Information Processing Systems 18*, Vancouver, Canada, 2005.

[14] F. Li and Y. Yang. A loss function analysis for classification methods in text categorization. In *Proc. 2003 Int. Conf. Machine Learning (ICML'03)*, 2003.

[15] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, Cambridge, MA, 2001.

[16] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, March 1982.

[17] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[18] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[19] A. N. Tikhonov. Regularization of incorrectly posed problems. *Soviet Math.*, (4), 1963 (English Translation).

[20] V. N. Vapnik. *Statistical learning theory.* John Wiley & Sons, 1998.

[21] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *Proc. Int. Conf. Computer Vision (ICCV'99)*, pages 975–982, Kerkyra, Greece, Sept. 1999.

[22] T. Zhang and R. Ando. Analysis of spectral kernel design based semi-supervised learning. In *Advances in Neural Information Processing Systems 18*, 2005.

[23] T. Zhang and F. J. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 4:5–31, 2001.

[24] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2003.

[25] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of the twentieth Internation Conference on Machine Learning*, 2003.