RC24005 (W0607-052) July 14, 2006
Computer Science

IBM Research Report

Learning on Graph with Normalized Laplacian Regularization

Rie Kubota Ando
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Tong Zhang
Yahoo! Inc.
New York, NY 10011

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Learning on Graph with Normalized Laplacian Regularization

Rie Kubota Ando
IBM T.J. Watson Research Center
Hawthorne, NY 10532, U.S.A.
rie1@us.ibm.com

Tong Zhang
Yahoo! Inc.
New York City, NY 10011, U.S.A.
tzhang@yahoo-inc.com

Abstract

We consider a general form of transductive learning on graphs with Laplacian regularization for multi-category classification, and obtain margin-based generalization bounds. Using this analysis, we establish the connection between graph cuts and margin, which enables us to express the generalization behavior of graph learning based on appropriate geometric properties of the graph. In particular, using a learning theoretical definition of normalized cut, we show the importance of normalizing the graph Laplacian matrix. Under appropriate assumptions, the optimal normalization factors can be derived. The analysis reveals the limitations of the standard normalization method, and provides a remedy. Experiments confirm the superiority of the learning theoretically motivated normalization scheme on artificial and real-world datasets.

1 Introduction

Graph-based methods, such as spectral embedding, spectral clustering, and semi-supervised learning, have drawn much attention in the machine learning community. While various ideas have been proposed based on different intuitions, only recently have there been theoretical studies trying to understand why these methods work. Such investigations are critical for future progress in the field because the theoretical insights can help us focus on what is important and design more effective algorithms.

In spectral clustering, a traditional starting point is to find a partition of a graph that minimizes a certain definition of "graph cut" that quantifies the quality of the partition. The cut is the objective one attempts to minimize. Spectral methods can then be derived as a certain continuous relaxation that approximately solves the "graph cut" problem. Based on various intuitions and heuristics, various definitions of cuts have been proposed in the literature (for example, [8, 3], among others). In order to understand such methods, we need to ask the following two questions. First what is the quality of the relaxation approach as an approximation method to solve the original "graph cut" problem. Second, and more importantly, we need to understand why one should optimize one definition of "cut" instead of other alternatives. In the literature, different arguments and intuitions have been proposed to justify different choices. However, without a more

universally acceptable criterion, it is difficult to argue that one cut definition is better than another just based on heuristics. If a universally agreeable standard does exist, then one should focus on that criterion instead of an artificially defined cut problem.

A common use of spectral partition is to cluster nodes in a graph (each partition is a cluster). In such applications, there are often pre-defined (but unknown) clusters (classes) that one is interested in. In this setting, the goal is to find such classes either using unsupervised or semi-supervised methods. Therefore for such problems, a universally agreeable standard is to find clusters that overlap significantly with the underlying class labels. In particular, instead of using any artificially defined cut, we should design an algorithm to minimize the classification error. This is the criterion we focus on in this paper.

In order to apply graph methods to statistical clustering or semi-supervised learning, one may construct *similarity graphs* by linking similar data points. For example, one may connect data points that are close in the feature space to form a k-nearest neighbor graph. If the graph is fully connected within each class and disconnected between the classes, then appropriate cut minimization leads to perfect classification. It was proposed in [6] that one may first project these data points into the eigenspace corresponding to the largest eigenvalues of a normalized adjacency matrix of the graph and then use the standard k-means method to perform clustering. The basic motivation is quite similar to that of [8]. It can be shown that in the ideal case (each class forms a connected subgraph, and there is no inter-class edge), points in the same cluster will be mapped into a single point in the reduced eigenspace, while points in different clusters will be mapped to different points. This implies that for clustering the distance in the reduced space is better than the original distance.

A natural question to ask is in general how should one design a distance function that leads to better clustering. While the argument in [6] gives a satisfactory answer in the idealized case, it is far less clear what happens in general. One approach to address this problem is to learn a distance metric that can lead to more desirable clustering results from a set of labeled examples (for example, as in [10]). The inner product associated with a distance metric can be viewed as a kernel, and the kernel fully determines the outcome of the k-means algorithm. Therefore this approach can also be viewed as designing a kernel optimal for clustering. Closely related to clustering, one may also consider kernel design methods in semi-supervised learning using a discriminative method such as SVM (e.g. [5]). In this setting, the change of the distance metric becomes a change of the underlying kernel. If the kernel is induced from a graph, then one may formulate semi-supervised learning directly on the graph; for example, see [1, 9, 13, 14].

In these studies, the kernel is induced from the adjacency matrix $\mathbf{W}$ whose (i, j)-entry is the weight of edge (i, j). $\mathbf{W}$ is often normalized by $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$ [2, 8, 6, 13] where $\mathbf{D}$ is a diagonal matrix whose (j, j)-entry is the degree of the j-th node, but sometimes not [1, 14]. Although such normalization may significantly affect the performance, the issue has not been studied from the learning theory perspective.

The relationship of kernel design and graph learning was investigated in [12], where it was argued that quadratic regularization-based graph learning can be regarded as kernel design in the spectral domain. That is, one keeps the kernel eigenvectors and modifies the corresponding eigenvalues. Moreover if input data are corrupted with noise, then such spectral graph design can help to improve classification performance. The focus there was on graphs with nodes generated from a random distribution (i.e., under the standard assumption of supervised learning), and edges weighted by a fixed kernel function. However, the analysis does not handle general graphs such as web graphs and does not explain why normalization of the adjacency matrix $\mathbf{W}$ is useful for

practical purposes.

Our goals here are twofold. First we present a model for transductive learning on graphs and develop a margin analysis for multi-class graph learning. We use this theory to analyze the performance of graph learning using graph properties such as graph-cut and a concept we call *pure subgraph*. The analysis naturally employs quantities formalizing the standard graph-learning assumption that well connected nodes are likely to have the same label. Second, we use the analysis to obtain a better understanding of the role of normalization of the graph Laplacian matrix ($\mathbf{D}-\mathbf{W}$) as well as dimension reduction in graph learning. The theoretical analysis indicates a limitation of the standard degree-based normalization mentioned above. We propose a remedy based on the learning theory results and use experiments to demonstrate that the remedy leads to improved classification performance.

2 Transductive Learning Model

We consider the following multi-category transductive learning model defined on a graph. Let $V = \{v_1, \ldots, v_m\}$ be a set of m nodes, and let $\mathcal{Y}$ be a set of K possible output values. Assume that each node v_j is associated with an output value $y_j \in \mathcal{Y}$, which we are interested in predicting. In order to do so, we randomly draw a set of n indices $Z_n = \{j_i : 1 \leq i \leq n\}$ from $\{1, \ldots, m\}$ uniformly and without replacement. We manually label the n nodes v_{j_i} with labels $y_{j_i} \in \mathcal{Y}$, and then automatically label the remaining $m - n$ nodes. The goal is to estimate the labels on the remaining $m - n$ nodes as accurately as possible.

In this paper, we shall assume that the labels $y = [y_1, \ldots, y_m]$ are deterministic. However, the analysis can also be applied if we have random labels. In the standard supervised learning model, we want to make a prediction that works well under such randomization of labels. In the transductive learning setting considered in this paper, we may assume that we are given a single random draw $y = [y_1, \ldots, y_m]$, which we fix. With this fixed y vector, we are interested in the performance of reconstructing it from a subset of labels. This formulation is the more appropriate setting for problems such as classification on graphs considered here.

In modern machine learning, instead of estimating the labels y_j directly, y_j is often encoded into a vector in R^K, so that the problem becomes that of generating an estimation vector $f_j = [f_{j,1}, \ldots, f_{j,K}] \in R^K$, which can then be used to recover the label y_j. In multi-category classification with K classes $\mathcal{Y} = \{1, \ldots, K\}$, we encode each $y_j = k \in \mathcal{Y}$ as $e_k \in R^K$, where e_k is a vector of zero entries except for the k-th entry being one. Given a function $f_j = [f_{j,1}, \ldots, f_{j,K}] \in R^K$ (which is intended to approximate e_{y_j}), we decode the corresponding label estimation $\hat{y}_j$ as:

$$\hat{y}_j = \arg\max_k \{f_{j,k} : k = 1, \ldots, K\}.$$

If the true label is y_j, then the corresponding classification error is:

$$\mathbf{err}(f_j, y_j) = I(\hat{y}_j \neq y_j),$$

where we use $I(\cdot)$ to denote the set indicator function.

In order to estimate $f = [f_j] = [f_{j,k}] \in R^{mK}$ from only a subset of labeled nodes, we have to impose restrictions on possible values of f. In this paper, we consider restrictions defined through a quadratic regularizer of the following form:

$$f^T \mathbf{Q}_{\mathbf{K}} f = \sum_{k=1}^{K} f_{\cdot,k}^T \mathbf{K}^{-1} f_{\cdot,k},$$

where $\mathbf{K} \in R^{m \times m}$ is a kernel matrix and $f_{\cdot,k} = [f_{1,k}, \ldots, f_{m,k}] \in R^m$. That is, the predictive vector for each class k is regularized separately. We assume that the kernel matrix $\mathbf{K}$ is full-rank. We will consider the kernel matrix induced by the graph Laplacian, which we shall define later in the paper. Note that we use the bold symbol $\mathbf{K}$ to denote the kernel matrix and the regular capitalized K to denote the number of classes.

Given a vector $f \in R^{mK}$, the accuracy of its component $f_j = [f_{j,1}, \ldots, f_{j,K}] \in R^K$ is measured by a loss function $\phi(f_j, y_j)$. Our learning method attempts to minimize the empirical risk on the set Z_n of n labeled training nodes, subject to $f^T \mathbf{Q}_{\mathbf{K}} f$ being small:

$$\hat{f}(Z_n) = \arg \min_{f \in R^{mK}} \left[\frac{1}{n} \sum_{j \in Z_n} \phi(f_j, y_j) + \lambda f^T \mathbf{Q}_{\mathbf{K}} f \right]. \tag{1}$$

where $\lambda > 0$ is an appropriately chosen regularization parameter.

In this paper, we focus on a special class of loss function that is of the form $\phi(f_j, y_j) = \sum_{k=1}^K \phi_0(f_{j,k}, \delta_{k,y_j})$, where $\delta_{a,b}$ is the delta function defined as: $\delta_{a,b} = 1$ when $a = b$ and $\delta_{a,b} = 0$ otherwise. We are interested in the generalization behavior of (1) compared to a properly defined optimal regularized risk. This type of inequality is often referred to as "oracle inequality" in the learning theory literature and is particularly useful for analyzing the quality of the underlying learning method. The following theorem gives an oracle inequality, and its proof can be found in Appendix A.

Theorem 1 *Let $\phi(f_j, y_j) = \sum_{k=1}^K \phi_0(f_{j,k}, \delta_{k,y_j})$ in (1). Assume that there exist positive constants a, b, and c such that*

- *$\phi_0(x, y)$ is non-negative and convex in x.*
- *$\phi_0(x, y)$ is Lipschitz with constant b when $\phi_0(x, y) \leq a$.*
- *$c = \inf\{x : \phi_0(x, 1) \leq a\} - \sup\{x : \phi_0(x, 0) \leq a\}$.*

Then $\forall p > 0$, the expected generalization error of the learning method (1) over the random training samples Z_n can be bounded by:

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j(Z_n), y_j) \leq \frac{1}{a} \inf_{f \in R^{mK}} \left[\frac{1}{m} \sum_{j=1}^m \phi_0(f_j, y_j) + \lambda f^T \mathbf{Q}_{\mathbf{K}} f \right] + \left(\frac{b \mathbf{tr}_p(\mathbf{K})}{\lambda n c} \right)^p,$$

where $\bar{Z}_n = \{1, \ldots, m\} - Z_n$,

$$\mathbf{tr}_p(\mathbf{K}) = \left(\frac{1}{m} \sum_{j=1}^m \mathbf{K}_{j,j}^p \right)^{1/p},$$

and $\mathbf{K}_{j,j}$ denotes the j-th diagonal entry of matrix $\mathbf{K}$.

If we take $p = 1$ in Theorem 1, then the bound becomes

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j(Z_n), y_j) \leq \frac{1}{a} \inf_{f \in R^{mK}} \left[\frac{1}{m} \sum_{j=1}^m \phi_0(f_j, y_j) + \lambda f^T \mathbf{Q}_{\mathbf{K}} f \right] + \frac{b \mathbf{tr}(\mathbf{K})}{\lambda n m c},$$

where $\mathbf{tr}(\mathbf{K}) = m\mathbf{tr}_1(\mathbf{K})$ is the trace of matrix $\mathbf{K}$. The trace of a kernel matrix has been employed in a number of previous studies to characterize generalization ability of kernel methods. The generalized quantity in Theorem 1 with $p \neq 1$ has non-trivial consequences which we will investigate in the paper. The formulation used here corresponds to the one-versus-all method for multi-category classification, and standard binary loss functions such as least squares, logistic regression, and SVMs can be used. For the SVM loss function $\phi_0(x, y) = \max(0, 1-(2x-1)(2y-1))$, we may take $a = 0.5$, $b = 2$, and $c = 0.5$. In the experiments reported here, we shall employ the least squares function $\phi_0(x, y) = (x - y)^2$ which is widely used in the graph learning literature. With this formulation, we may choose $a = 1/16$, $b = 0.5$, $c = 0.5$, and obtain the following result.

Corollary 1 *Consider the least squares one-versus-all method for graph learning:*

$$\hat{f}(Z_n) = \arg\min_{f \in R^{mK}} \left[\frac{1}{n} \sum_{j \in Z_n} \sum_{k=1}^{K} (f_j - \delta_{k,y_j})^2 + \lambda f^T \mathbf{Q}_{\mathbf{K}} f \right].$$

Then $\forall p > 0$, the expected generalization error of the learning method over the random training samples Z_n can be bounded by:

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j(Z_n), y_j) \leq 16 \inf_{f \in R^{mK}} \left[\frac{1}{m} \sum_{j=1}^{m} \sum_{k=1}^{K} (f_j - \delta_{k,y_j})^2 + \lambda f^T \mathbf{Q}_{\mathbf{K}} f \right] + \left(\frac{\mathbf{tr}_p(\mathbf{K})}{\lambda n} \right)^p.$$

Note that one may also use other forms of loss function such as $\phi(f_j, y_j) = \sup_{k \neq y_j} \phi_0(f_{j,y_j} - f_{j,k})$ and obtain similar bounds. Moreover, it is possible to derive other types of (non-oracle) generalization bounds, such as probability inequalities in which the generalization error is bounded using the observed training error plus a complexity term. We shall not include them in this paper since Theorem 1 is sufficient for our purposes here. What is important in our analysis are the two quantities $f^T \mathbf{Q}_{\mathbf{K}} f$ and $\mathbf{tr}_p(\mathbf{K})$ that determine the generalization performance. We will focus on the interpretation of these quantities.

3 Margin and Graph Cut

Consider an undirected graph $G = (V, E)$ defined on the nodes $V = \{v_j : j = 1, \ldots, m\}$, with edges $E \subset \{1, \ldots, m\} \times \{1, \ldots, m\}$, and weights $w_{j,j'} \geq 0$ associated with edges $(j, j') \in E$. For simplicity, we assume that $(j, j) \notin E$ and $w_{j,j'} = 0$ when $(j, j') \notin E$. Let $\deg_j(G) = \sum_{j'=1}^{m} w_{j,j'}$ be the degree of node j of graph G. We consider the following definition of normalized Laplacian.

Definition 1 *Consider a graph $G = (V, E)$ of m nodes with weights $w_{j,j'}$ $(j, j' = 1, \ldots, m)$. The unnormalized Laplacian matrix $\mathcal{L}(G) \in R^{m \times m}$ is defined as: $\mathcal{L}_{j,j'}(G) = -w_{j,j'}$ if $j \neq j'$; $\deg_j(G)$ otherwise. Given m scaling factors $\mathbf{S}_j$ $(j = 1, \ldots, m)$, let $\mathbf{S} = \mathrm{diag}(\{\mathbf{S}_j\})$. The $\mathbf{S}$-normalized Laplacian matrix is defined as: $\mathcal{L}_{\mathbf{S}}(G) = \mathbf{S}^{-1/2} \mathcal{L}(G) \mathbf{S}^{-1/2}$. The corresponding regularization is based on:*

$$f_{\cdot,k}^T \mathcal{L}_{\mathbf{S}}(G) f_{\cdot,k} = \frac{1}{2} \sum_{j,j'=1}^{m} w_{j,j'} \left(\frac{f_{j,k}}{\sqrt{\mathbf{S}_j}} - \frac{f_{j',k}}{\sqrt{\mathbf{S}_{j'}}} \right)^2.$$

A common choice of $\mathbf{S}$ is $\mathbf{S} = \mathbf{I}$, corresponding to regularizing with the unnormalized Laplacian $\mathcal{L}$. The idea is natural: we assume that the predictive values $f_{j,k}$ and $f_{j',k}$ should be close when $(j, j') \in E$ with a strong link. Another common choice is to normalize by $\mathbf{S}_j = \deg_j(G)$, as in [6, 8, 13, 2], which we refer to as degree-based normalization. At first sight, the need for normalization is not immediately clear. However, as we will show later, normalization using appropriate scaling factors can improve performance.

In this section we focus on learning on graph with the Laplacian regularizer. Another important issue, dimension reduction, will be considered in the next section.

3.1 Generalization analysis using graph-cut

We will adapt the margin style generalization analysis in Section 2 to analyze graph learning using graph properties such as graph-cut. We now introduce a learning theoretical definition of $\mathbf{S}$-normalized graph cut as follows.

Definition 2 *Given label $y = \{y_j\}_{j=1,\ldots,m}$ on V, we define the cut for the $\mathbf{S}$-normalized Laplacian $\mathcal{L}_{\mathbf{S}}$ in Definition 1 as:*

$$\mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y) = \sum_{j,j':y_j \neq y_{j'}} \frac{w_{j,j'}}{2}\left(\frac{1}{\mathbf{S}_j} + \frac{1}{\mathbf{S}_{j'}}\right) + \sum_{j,j':y_j = y_{j'}} \frac{w_{j,j'}}{2}\left(\frac{1}{\sqrt{\mathbf{S}_j}} - \frac{1}{\sqrt{\mathbf{S}_{j'}}}\right)^2 .$$

Note that unlike typical graph-theoretical definitions of graph-cut in the literature, the learning theoretical definition of cut not only penalizes a normalized version of between-class edge weights, but also penalizes within-class edge weights when such an edge connects two nodes with different scaling factors. This difference has important consequences, which we will investigate later in the paper. For unnormalized Laplacian, the second term on the right hand side of Definition 2 vanishes, which means that it only penalizes weights corresponding to edges connecting nodes with different labels. In this case, the learning theoretical definition is identical to the standard graph-theoretical definition:

$$\mathbf{cut}(\mathcal{L}, y) = \sum_{j,j':y_j \neq y_{j'}} w_{j,j'}.$$

Using the learning theoretical graph-cut definition, we can obtain a generalization result for the estimator in (1) with $\mathbf{K}$ defined as follows:

$$\mathbf{K}^{-1} = \alpha\mathbf{S}^{-1} + \mathcal{L}_{\mathbf{S}}(G) = \mathbf{S}^{-1/2}(\alpha\mathbf{I} + \mathcal{L}(G))\mathbf{S}^{-1/2} , \tag{2}$$

where $\mathbf{I}$ is the identity matrix. Note that $\alpha > 0$ is a tuning parameter to ensure that $\mathbf{K}$ is strictly positive definite. As we will see later, this parameter is important. The corresponding regularization condition is

$$f^T\mathbf{Q}_{\mathbf{K}}f = \sum_{k=1}^{K}\left(\alpha\sum_{j=1}^{m}\frac{f_{k,j}^2}{\mathbf{S}_j} + \frac{1}{2}\sum_{j,j'=1}^{m}\left(\frac{f_{j,k}}{\sqrt{\mathbf{S}_j}} - \frac{f_{j',k}}{\sqrt{\mathbf{S}_{j'}}}\right)^2 w_{j,j'}\right).$$

Another possibility is to let $\mathbf{K}^{-1} = \alpha\mathbf{I} + \mathcal{L}_{\mathbf{S}}(G)$. The conclusions, which we will not include in this paper, are similar to that of (2).

For simplicity, we state the generalization bound based on Theorem 1 with optimal λ. Note that in applications, λ is usually tuned through cross validation. Therefore assuming optimal λ will simplify the bound so that we can focus on the more essential characteristics of generalization performance.

Theorem 2 *Assume that the conditions in Theorem 1 hold with the regularization condition (2). Moreover, assume that $\phi_0(0,0) = \phi_0(1,1) = 0$, then $\forall p > 0$, there exists a sample independent regularization parameter λ in (1) such that the expected generalization error is bounded by:*

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j(Z_n), y_j) \leq \frac{C_p(a,b,c)}{n^{p/(p+1)}} (\alpha s + \mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y))^{p/(p+1)} \mathbf{tr}_p(\mathbf{K})^{p/(p+1)},$$

where $C_p(a,b,c) = (b/ac)^{p/(p+1)}(p^{1/(p+1)} + p^{-p/(p+1)})$ and $s = \sum_{j=1}^{m} \mathbf{S}_j^{-1}$.

Proof Let $f_{j,k} = \delta_{y_j,k}$. It can be easily verified that

$$\frac{1}{m} \sum_{j=1}^{m} \phi(f_j, y_j) + \lambda f^T \mathbf{Q}_{\mathbf{K}} f = \lambda(\alpha s + \mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y)).$$

Now, using this expression in Theorem 1, and then optimizing over λ, we obtain the desired inequality. ■

It is easy to check that the conditions on the loss function in Theorem 2 hold for the least squares method in Corollary 1 with $b/ac = 16$. The conditions also hold for some other standard loss functions such as SVM.

With a fixed p, the generalization error decreases at the rate $O(n^{-p/(p+1)})$ when the sample size n increases. This rate of convergence is faster when p increases. However in general, $\mathbf{tr}_p(\mathbf{K})$ is an increasing function of p. Therefore we have a trade-off between the two terms, and without appropriate normalization (which we will consider later in the paper), one may prefer a smaller p in order to optimize the bound. An analysis will be provided in the next section. The bound also suggests that if we normalize $\mathbf{K}$ so that its diagonal entries $\mathbf{K}_{j,j}$ become a constant, then $\mathbf{tr}_p(\mathbf{K})$ is independent of p, and thus a larger p can be used in the bound. This motivates the idea of normalizing the diagonals of $\mathbf{K}$, which we will further investigate later in the paper. The generalization bound in Theorem 2 is closely related to the margin analysis for binary linear classification. Here we relate it to the concept of graph cut. Our goal is to better understand the quantity $(\alpha s + \mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y))^{p/(p+1)} \mathbf{tr}_p(\mathbf{K})^{p/(p+1)}$ using graph properties, which gives better understanding of graph based learning.

In the following, we will give example applications of Theorem 2. They illustrate that theoretically it is important to tune the parameter α to achieve good performance, which is also empirically observed in our experiments.

3.2 Zero-cut and Geometric Margin Separation

We consider an application of Theorem 2 for the unnormalized Laplacian under the zero-cut assumption that each connected component of the graph has a single label. With this assumption, the task is simply to estimate what label each connected component has.

Theorem 3 *Assume that* $\mathbf{cut}(\mathcal{L}, y) = 0$*, and the graph has* q *connected components of sizes* $m_1 \leq \cdots \leq m_q$ $(\sum_\ell m_\ell = m)$*. For all* $p > 0$*, let* $\alpha \to 0$*, and with optimal* λ*, the learning method (1) with* $\mathbf{K}^{-1} = \alpha\mathbf{I} + \mathcal{L}$ *under the assumptions of Theorem 2 has generalization error*

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq \frac{C_p(a,b,c)}{n^{p/(p+1)}} \left(\sum_{\ell=1}^{q} (m/m_\ell)^{p-1} \right)^{1/(p+1)} + O(\alpha).$$

In particular, we have

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq 2\sqrt{\frac{b}{ac} \cdot \frac{q}{n}} + O(\alpha).$$

and

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq \frac{b}{ac} \cdot \frac{m}{nm_1} + O(\alpha).$$

Proof Since the graph has q connected components, $\mathcal{L}$ has q eigenvectors $\mathbf{v}_\ell$ $(\ell = 1, \ldots, q)$ with zero-eigenvalues, where each eigenvector $\mathbf{v}_\ell$ is the indicator function of the ℓ-th connected component in the graph, i.e. the j-th entry of vector $\mathbf{v}_\ell$ is 1 for $j \in V_\ell$ and 0 otherwise. It is not hard to check that as $\alpha \to 0$, $\alpha\mathbf{K} \to \sum_{\ell=1}^{q} \frac{1}{m_\ell} \mathbf{v}_\ell \mathbf{v}_\ell^T + O(\alpha)$. Therefore $\alpha \mathbf{tr}_p(\mathbf{K}) \to m^{-1/p} (\sum_{\ell=1}^{q} m_\ell^{1-p})^{1/p}$. Now, we can use Theorem 2 to obtain the first inequality. The second inequality is obtained by setting $p = 1$, and the third inequality is obtained by letting $p \to \infty$. ■

Under the zero-cut assumption, the generalization performance can be bounded as $O(\sqrt{q/n})$ when $\alpha \to 0$. However, we can also achieve a faster convergence rate of $O(1/n)$, although the generalization performance depends on the inverse of the smallest component size through $m/m_1 \geq q$. This implies that we will achieve better convergence at the $O(1/n)$ level if the sizes of the components are balanced. If the component sizes are significantly different, the convergence may behave like $O(\sqrt{q/n})$.

We discuss a concrete example in which Theorem 3 is applicable. Assume that each node v_j is associated with a data point x_j that belongs to the d-dimensional unit ball $B = \{x \in R^d : \|x\|_2 \leq 1\}$. We form a graph by connecting all nodes v_j to their nearest neighbors. In particular, we may consider an ϵ-ball centered at each v_j: $B_j(\epsilon) = \{x : \|x - x_j\|_2 \leq \epsilon\}$. We then form a graph by connecting each j with all points within the ball $B_j(\epsilon)$ and with unit weights.

We say that the data points are separable with geometric margin γ if for each node v_j the ball $B_j(\gamma)$ only contains points in class y_j. Now assume we use a ball of size $\epsilon \leq \gamma$. In this case, $\mathbf{cut}(\mathcal{L}, y) = 0$, and there is a constant $q \leq \epsilon^{-d}$ such that the graph has at most q connected components, and we have:

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq 2\sqrt{\frac{b}{ac} \cdot \frac{q}{n}} + O(\alpha).$$

This bound does not depend on margin γ but depends only on q, the number of connected components. So even if the margin γ is small, the bound can still be good as long as q is small.

The results obtained here are novel and critical for understanding why graph based semi-supervised learning may work better than standard kernel learning. In fact, it is not possible to derive similar generalization bounds for supervised learning because one needs unlabeled data (in addition to labeled data) to define such connected components. This means that graph semi-supervised learning can take advantage of the new quantity q to characterize its generalization performance, and this quantity cannot be utilized by standard supervised learning.

Note that we have assumed a very specific generative model for the data. In particular, if the data are generated in a way such that the number of connected components q is small, and each connected component belongs to a single class, then graph based semi-supervised learning can work better than supervised kernel learning. If this assumption does not hold (at least approximately), then graph based learning methods may fail. However, for many practical applications, the geometric margin separation assumption does appear quite reasonable. Therefore for such problems, graph based semi-supervised learning, which can take advantage of the underlying data generation model, may become helpful.

The analysis given here is related to that of [12], where the benefit of graph learning was examined through the kernel design point of view, and a data generation model with noisy input was considered. [12] showed that unsupervised kernel design will generally be beneficial under that assumption, using a generalization bound that depends on the trace of the kernel matrix (corresponding to $p = 1$ in this paper). Here we consider the specific kernel induced from graph Laplacian and characterize the generalization behavior of graph based semi-supervised learning with respect to properties of the underlying graph. We seek to give useful insights into the effectiveness of semi-supervised learning from a different (but related) point of view from that of [12]. The statistical analysis presented here is more general since the characterization uses a generalized trace in which we can take $p \neq 1$. This is important to obtain good understanding of the role of normalization (which was not considered in [12]), as we will demonstrate later in the paper.

3.3 Non-zero cut and Pure Components

It is often too restrictive to assume that each connected component has only one label (that is, the cut is zero). In this section, we show that similar bounds can be obtained when this data generation assumption is relaxed. We are still interested in giving a characterization of the performance of (1) in terms of properties of the graph and introduce the following definition.

Definition 3 *A subgraph $G_0 = (V_0, E_0)$ of $G = (V, E)$ is called a pure component if G_0 is connected, E_0 is induced by restricting E on V_0, and the labels y have identical values on V_0. A pure subgraph $G' = \cup_{\ell=1}^{q} G_\ell$ of G divides V into q disjoint sets $V = \cup_{\ell=1}^{q} V_\ell$ such that each subgraph $G_\ell = (V_\ell, E_\ell)$ is a pure component. Denote by $\lambda_i(G_\ell) = \lambda_i(\mathcal{L}(G_\ell))$ the i-th smallest eigenvalue of $\mathcal{L}(G_\ell)$.*

For instance, if we remove all edges of G that connect nodes with different labels, then the resulting subgraph is a pure subgraph (though it may not be the only one). For each pure component G_ℓ, its first eigenvalue $\lambda_1(G_\ell)$ is always zero. The second eigenvalue $\lambda_2(G_\ell) > 0$ because G_ℓ is connected. This $\lambda_2(G_\ell)$ can be regarded as a measurement of how well G_ℓ is connected. We use it together with graph cut to specify our generalization bound.

Theorem 4 *Let the assumptions of Theorem 2 hold with regularization condition (2). Let $G' = \cup_{\ell=1}^{q} G_\ell$ ($G_\ell = (V_\ell, E_\ell)$) be a pure subgraph of G. For all $p \geq 1$, there exist sample-independent*

regularization parameter λ and a fixed tuning parameter α, such that the learning method (1) under the assumptions of Theorem 2 has generalization error

$$\begin{aligned}&\mathbf{E}_{Z_n}\frac{1}{m-n}\sum_{j\in\bar{Z}_n}\mathbf{err}(\hat{f}_j,y_j)\\ \leq&\frac{C_p(a,b,c)}{n^{p/(p+1)}}\left(s^{1/2}\left(\sum_{\ell=1}^{q}\frac{s_\ell(p)/m}{m_\ell^p}\right)^{1/2p}+\mathbf{cut}(\mathcal{L}_{\mathbf{S}},y)^{1/2}\left(\sum_{\ell=1}^{q}\frac{s_\ell(p)/m}{\lambda_2(G_\ell)^p}\right)^{1/2p}\right)^{2p/(p+1)},\end{aligned}$$

where $m_\ell=|V_\ell|$, $s=\sum_{j=1}^m\mathbf{S}_j^{-1}$, and $s_\ell(p)=\sum_{j\in V_\ell}\mathbf{S}_j^p$.

Proof The idea is similar to that of Theorem 3. We use the same notation, and let $\mathbf{v}_\ell$ be the indicator function of V_ℓ in V. Let $\mathbf{I}_\ell$ be the diagonal matrix with value ones for nodes corresponding to V_ℓ and zeros elsewhere. By the definitions, it is not hard to verify that $\mathcal{L}-\sum_{\ell=1}^q\lambda_2(G_\ell)(\mathbf{I}_\ell-\mathbf{v}_\ell\mathbf{v}_\ell^T/m_\ell)$ is positive semi-definite. Therefore

$$\left(\alpha\mathbf{I}+\sum_{\ell=1}^{q}\lambda_2(G_\ell)(\mathbf{I}_\ell-\mathbf{v}_\ell\mathbf{v}_\ell^T/m_\ell)\right)^{-1}-(\alpha\mathbf{I}+\mathcal{L})^{-1}$$

is positive-semi-definite. Also, from (2) we have $\mathbf{S}^{-1/2}\mathbf{K}\mathbf{S}^{-1/2}=(\alpha\mathbf{I}+\mathcal{L}(G))^{-1}$, so we know that the diagonal entries of $\mathbf{S}^{-1/2}\mathbf{K}\mathbf{S}^{-1/2}$ can be upper-bounded by those of

$$\left(\alpha\mathbf{I}+\sum_{\ell=1}^{q}\lambda_2(G_\ell)(\mathbf{I}_\ell-\mathbf{v}_\ell\mathbf{v}_\ell^T/m_\ell)\right)^{-1}=\sum_{\ell=1}^{q}(\alpha+\lambda_2(G_\ell))^{-1}\left(\mathbf{I}_\ell+\alpha^{-1}\lambda_2(G_\ell)\mathbf{v}_\ell\mathbf{v}_\ell^T/m_\ell\right).$$

For the latter, its m_ℓ diagonal entries for each pure component ℓ can be upper bounded by $\lambda_2(G_\ell)^{-1}+(\alpha m_\ell)^{-1}$. Therefore:

$$\begin{aligned}m^{1/p}\mathbf{tr}_p(\mathbf{K})&\leq\left(\sum_{\ell=1}^{q}s_\ell(p)(\alpha^{-1}m_\ell^{-1}+\lambda_2(G_\ell)^{-1})^p\right)^{1/p}\\&\leq\left[\alpha^{-1}\left(\sum_{\ell=1}^{q}s_\ell(p)m_\ell^{-p}\right)^{1/p}+\left(\sum_{\ell=1}^{q}s_\ell(p)\lambda_2(G_\ell)^{-p}\right)^{1/p}\right].\end{aligned}$$

Substitute this estimate into Theorem 2, we have

$$\mathbf{E}_{Z_n}\frac{1}{m-n}\sum_{j\in\bar{Z}_n}\mathbf{err}(\hat{f}_j(Z_n),y_j)\leq\frac{C_p(a,b,c)}{n^{p/(p+1)}}\left[m^{-1/p}(\alpha s+C)(\alpha^{-1}A+B)\right]^{p/(p+1)},$$

where $A=(\sum_{\ell=1}^q s_\ell(p)m_\ell^{-p})^{1/p}$, $B=(\sum_{\ell=1}^q s_\ell(p)\lambda_2(G_\ell)^{-p})^{1/p}$, and $C=\mathbf{cut}(\mathcal{L}_{\mathbf{S}},y)$. Now optimize over α (let $\alpha=\sqrt{AC/(sB)}$), and simplify, we obtain the desired inequality. ■

Theorem 4 is a natural generalization of Theorem 3 when $p\geq 1$. It quantitatively illustrates the importance of analyzing graph learning using a partition of the original graph into well-connected

pure components. The second eigenvalue $\lambda_2(G_i)$ measures how well-connected G_i is. A more intuitive quantity that measures the connectedness of graph $G = (V, E)$ is the *isoperimetric number* h_G defined as

$$h_G = \inf_{S \subset V} \sum_{j \in S, j' \in V - S} w_{j,j'} / \min(|S|, |V - S|).$$

It is well-known that $\lambda_2(G_i) \geq h_{G_i}^2 / (2 \max_j \deg_j(G_i))$ [2]. The isoperimetric number of a graph is large when the nodes are well-connected everywhere. In particular, if $\deg_j(G)$ is of the order $|V|$, and $w_{i,j} = 1$ when $(i, j) \in E$, then for a well-connected graph, $\sum_{j \in S, j' \in V - S} w_{j,j'}$ is of the order $|S||V - S|$, and $h_G = O(|V|)$. We thus have the condition $\lambda_2(G_\ell) \geq u(G')|V_\ell|$ for some constant $u(G')$ that does not depend on the size of the pure components. Under this condition, we may replace $\sum_{\ell=1}^q m_\ell \lambda_2(G_\ell)^{-p}$ by $u(G')^{-p} \sum_{\ell=1}^q m_\ell^{1-p}$ in Theorem 4 and obtain a simplified bound:

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq \frac{C_p(a,b,c)}{n^{p/(p+1)}} \left(\sum_{\ell=1}^q \frac{s_\ell(p)/m}{(m_\ell/m)^p} \right)^{1/(p+1)} \left(\sqrt{\frac{s}{m}} + \sqrt{\frac{\mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y)}{u(G')m}} \right)^{2p/(p+1)},$$

where we define $u(G') = \min_\ell(\lambda_2(G_\ell)/m_\ell)$. We consider two special cases: $p = 1$ and $p \to \infty$:

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq 2 \sqrt{\frac{b}{ac} \cdot \frac{\sum_{\ell=1}^q (s_\ell(1)/m_\ell)}{n}} \left(\sqrt{\frac{s}{m}} + \sqrt{\frac{\mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y)}{u(G')m}} \right), \tag{3}$$

$$\mathbf{E}_{Z_n} \frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq \frac{b}{ac} \cdot \frac{\max_\ell \max_{j \in V_\ell} (\mathbf{S}_j / m_\ell)}{n} \left(\sqrt{s} + \sqrt{\frac{\mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y)}{u(G')}} \right)^2. \tag{4}$$

These bounds are generalizations of those in Theorem 3. Suppose that we take $\mathbf{S} = \mathbf{I}$. Then the number of pure components q affects the $O(1/\sqrt{n})$ convergence rate in (3) as $\sum_{\ell=1}^q s_\ell(1)/m_\ell = q$. If the sizes of the components are balanced, we can achieve better convergence at the $O(1/n)$ level as in (4); otherwise, the convergence may behave like $O(\sqrt{q/n})$. This observation motivates a scaling matrix $\mathbf{S}$ that compensates for the unbalanced pure component sizes, which we will investigate next.

3.4 Optimal Normalization for Near-zero-cut Partition

As discussed in the introduction, the common practice of the normalization of the adjacency matrix ($\mathbf{W}$) or the graph Laplacian ($\mathbf{D} - \mathbf{W}$) is based on degrees, which corresponds to setting $\mathbf{S} = \mathbf{D}$. Although such normalization may significantly affect the performance, to our knowledge, there is no learning theory analysis on the effect of normalization. The purpose of this section is to fill this gap using the theoretical tools developed earlier. We shall focus on a near ideal situation to gain intuition.

Consider a pure subgraph $G' = \cup_{\ell=1}^q G_\ell$ ($G_\ell = (V_\ell, E_\ell)$) of G. From Definition 1, we know that good scaling factors $\mathbf{S}_j$ should be approximately constant within each class. In the following, we are interested in finding an optimal scaling matrix $\mathbf{S}$ such that $\mathbf{S}_j$ is constant within each pure component V_ℓ. Therefore in the following, we will assume that $\mathbf{S}$ is quantified by q numbers $[\bar{s}_\ell]_{\ell=1,\ldots,q}$, such that $\mathbf{S}_j = \bar{s}_\ell$ when $j \in V_\ell$.

Consider the following quantity:

$$\mathbf{cut}(G', y) = \sum_{j,j':y_j \neq y_{j'}} w_{j,j'} + \sum_{\ell \neq \ell'} \sum_{j \in V_\ell, j' \in V_{\ell'}} \frac{w_{j,j'}}{2}.$$

It is easy to check that

$$\mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y) \leq \mathbf{cut}(G', y) / \min_\ell \bar{s}_\ell.$$

Assume that weights are small between pure components, and therefore, $\mathbf{cut}(G', y)$ is small.

With the $O(1/n)$ convergence rate, we obtain from (4) that

$$\frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq \frac{b}{ac} \cdot \frac{\max_\ell(\bar{s}_\ell/m_\ell)}{n} \left(\sqrt{\sum_{\ell=1}^{q} m_\ell/\bar{s}_\ell} + \sqrt{\frac{\mathbf{cut}(G', y)}{u(G') \min_\ell \bar{s}_\ell}} \right)^2.$$

If we assume that $\mathbf{cut}(G', y)/(u(G') \min_\ell m_\ell) \ll q$, then the dominating term on the right hand side is

$$\frac{\max_\ell(\bar{s}_\ell/m_\ell)}{n} \sum_{\ell=1}^{q} \frac{m_\ell}{\bar{s}_\ell},$$

which can be optimized with the choice $\bar{s}_\ell = m_\ell$, and the resulting bound becomes:

$$\frac{1}{m-n} \sum_{j \in \bar{Z}_n} \mathbf{err}(\hat{f}_j, y_j) \leq \frac{b}{ac} \cdot \frac{1}{n} \left(\sqrt{q} + \sqrt{\frac{\mathbf{cut}(G', y)}{u(G') \min_\ell m_\ell}} \right)^2.$$

That is, if $\mathbf{cut}(G', y)$ is small, then we can choose scaling factor $\bar{s}_\ell \propto m_\ell$ for each pure component ℓ so that the generalization performance is approximately $(ac)^{-1} b \cdot q/n$, which is of the order $O(1/n)$.

The analysis provided here not only proves the importance of normalization under the learning theoretical framework, but also suggests that the good normalization factor for each node j is approximately the size of the well-connected pure component that contains node j (assuming that nodes belonging to different pure components are only weakly connected). Our analysis focused on the case that the scaling factors are a constant within each pure component. This condition is quite natural if we look at the normalized Laplacian regularization condition in Definition 1, where $f_{j,k}/\sqrt{\mathbf{S}_j}$ should be similar to $f_{j',k}/\sqrt{\mathbf{S}_{j'}}$ when $w_{j,j'}$ is large. If j and j' belongs to the same class, then $f_{j,k}$ should be similar to $f_{j',k}$. Therefore for such a pair (j, j'), we want to have $\mathbf{S}_j \approx \mathbf{S}_{j'}$ if $w_{j,j'}$ is large. Note that this requirement is not enforced by the standard degree-based normalization method $\mathbf{S}_j = \deg_j(G)$ because a well-connected pure component may contain nodes with quite different degrees. The assumption is satisfied under a simplified "box model", which is related to the models used by some previous researchers to derive the standard normalization method (e.g. [8]). In this model, a pure component is completely connected, and each node connects to all other nodes and itself with edge weight $w_{j,j'} = 1$. The degree is thus $\deg_j(G_\ell) = |V_\ell| = m_\ell$, which gives the optimal scaling in our analysis.

In general, the box model may not be a good approximation for practical problems. A more realistic approximation, which we call core-satellite model, will be introduced in the experimental section. For such a model, the degree-based normalization can fail because the $\deg_j(G_\ell)$ within each pure component G_ℓ is not approximately constant, and it may not be proportional to m_ℓ. In

general, this approximation using degrees causes $\mathbf{S}_j$ to potentially vary significantly within a pure component because each $\mathbf{S}_j$ is only determined by its local neighborhoods.

Our analysis suggests that it is necessary to modify the degree-based scaling method $\mathbf{S}_j = \deg_j(G)$ so that the scaling factor is approximately a constant within each pure component, which should be proportional to m_ℓ. Our remedy is to look for connected components at a larger distance scale. Although there could be various methods to achieve this effect, we shall focus on a specific method motivated by the proofs of Theorem 3 and Theorem 4. Let $\bar{\mathbf{K}} = (\alpha\mathbf{I} + \mathcal{L})^{-1}$ be the kernel matrix corresponding to the unnormalized Laplacian. Using the terminology in the proofs, we observe that for small α:

$$\alpha\bar{\mathbf{K}} = \sum_{\ell=1}^{q} \mathbf{v}_\ell \mathbf{v}_\ell^T / m_\ell + O(1),$$

and thus $\bar{\mathbf{K}}_{j,j} \propto m_\ell^{-1}$ for each $j \in V_\ell$. Therefore with small α, the scaling factor $\mathbf{S}_j = 1/\bar{\mathbf{K}}_{j,j}$ is near optimal for all j. For $\alpha > 0$, the effect of this scaling factor is essentially equivalent to looking for connected components at a scale of at most $O(1/\alpha)$ number of nodes. We call this method of normalization $\mathbf{K}$-*scaling* in this paper. It is equivalent to a normalization of the kernel matrix $\mathbf{K}$, so that each $\mathbf{K}_{j,j} = 1$. Although this method coincides with a common practice in standard kernel learning, it is important to notice that to show this method behaves well in the graph learning setting is highly non-trivial and novel. To our best knowledge, no one has proposed this normalization method in the graph learning setting before. In fact, without learning theoretical results developed in this paper, it is not obvious to observe or argue that this method should work better than the more standard degree-based normalization method. In our framework, the main advantage of $\mathbf{K}$-scaling (compared to the standard degree-scaling, which we call $\mathbf{L}$-scaling) is twofold:

- The resulting $\mathbf{S}_j$ does not vary significantly within a well-connected pure component.
- The resulting scaling is approximately m_ℓ (at a scale of $1/\alpha$), which is predicted by our theory to be desirable.

The superiority of this method will be demonstrated in our experiments. The main drawback of this method is the computational cost of directly inverting $(\alpha\mathbf{I} + \mathcal{L})$. For large scale problems, approximation methods are required.

4 Dimension Reduction

Normalization and dimension reduction have been commonly used in spectral clustering such as [6, 8]. For semi-supervised learning, dimension reduction (without normalization) is known to improve performance [1, 12] while the degree-based normalization (without dimension reduction) has also been explored [13]. In this section, we show that an appropriate combination of normalization and dimension reduction can improve classification performance.

We shall first introduce dimension reduction with normalized Laplacian $\mathcal{L}_{\mathbf{S}}(G)$. Denote by $\mathbf{P}^r_{\mathbf{S}}(G)$ the projection operator onto the eigenspace of $\alpha\mathbf{S}^{-1} + \mathcal{L}_{\mathbf{S}}(G)$ corresponding to the r smallest eigenvalues. Now, we may define the following regularizer on the reduced subspace:

$$f_{\cdot,k}^T \mathbf{K}^{-1} f_{\cdot,k} = \begin{cases} f_{\cdot,k}^T \mathbf{K}_0^{-1} f_{\cdot,k} & \text{if } \mathbf{P}^r_{\mathbf{S}}(G) f_{\cdot,k} = f_{\cdot,k}, \\ +\infty & \text{otherwise.} \end{cases} \tag{5}$$

Note that in the following, we will focus on bounding the generalization complexity using the reduced dimensionality r. In such context, the choice of $\mathbf{K}_0$ is not important as far as our analysis is concerned. We may simply choose $\mathbf{K}_0 = \mathbf{I}$ (or we may let $\mathbf{K}_0^{-1} = \alpha\mathbf{S}^{-1} + \mathcal{L}_{\mathbf{S}}(G)$).

The benefit of dimension reduction in graph learning has been investigated in [12], under the spectral kernel design framework. The idea is to modify the kernel eigenvalues so that the target spectral coefficient matches the kernel coefficients. Note that the normalization issue, which will change the eigenvectors and their ordering, wasn't investigated there. However, with a fixed scaling matrix $\mathbf{S}$, the reasoning given in [12] can also be applied here. It was shown there that if noise is added into the kernel matrix, then in general kernel eigenvalues will decay slower than the target spectral coefficients. Because of this, dimension reduction, which makes kernel eigenvalues better match the decay of target spectral coefficients, will become helpful. For Laplacian regularization investigated here, we may regard noise as edges connecting different pure components that increase the cut in Definition 2. Such noise can be significantly reduced if we project it into a low-dimensional space, and if the target functions approximately lie in this low-dimensional space. In this context, the effect of modification of eigenspaces through appropriate Laplacian normalization is to achieve faster decay of the target spectral coefficients in the first few eigenvectors of the kernel.

The following theorem shows that the target vectors can be well approximated by their projection via $\mathbf{P}_{\mathbf{S}}^q(G)$.

Theorem 5 *Let $G' = \cup_{\ell=1}^q G_\ell$ ($G_\ell = (V_\ell, E_\ell)$) be a pure subgraph of G. Consider $r \geq q$. Then we have:*

$$\lambda_{r+1}(\mathcal{L}_{\mathbf{S}}(G)) \geq \lambda_{r+1}(\mathcal{L}_{\mathbf{S}}(G')) \geq \min_\ell \lambda_2(\mathcal{L}_{\mathbf{S}}(G_\ell)).$$

For each k, let $\bar{f}_{j,k} = \delta_{y_j,k}$ be the target (encoding of the true labels) for class k ($j = 1, \ldots, m$). Then $\|\mathbf{P}_{\mathbf{S}}^r(G)\bar{f}_{\cdot,k} - \bar{f}_{\cdot,k}\|_2^2 \leq \delta_r(\mathbf{S})\|\bar{f}_{\cdot,k}\|_2^2$, where

$$\delta_r(\mathbf{S}) = \frac{\|\mathcal{L}_{\mathbf{S}}(G) - \mathcal{L}_{\mathbf{S}}(G')\|_2 + d(\mathbf{S})}{\lambda_{r+1}(\mathcal{L}_{\mathbf{S}}(G))}, \quad d(\mathbf{S}) = \max_\ell \frac{1}{2|V_\ell|} \sum_{j,j' \in V_\ell} (\mathbf{S}_j^{-1/2} - \mathbf{S}_{j'}^{-1/2})^2.$$

Proof Let $\mathbf{E} = \mathcal{L}_{\mathbf{S}}(G) - \mathcal{L}_{\mathbf{S}}(G')$, then $\mathbf{E}$ is a positive semi-definite matrix. Therefore, we know that the eigenvalues of $\mathcal{L}_{\mathbf{S}}(G)$ are no less than the corresponding eigenvalues of $\mathcal{L}_{\mathbf{S}}(G')$. Since the $(q+1)$-th smallest eigenvalue of $\mathcal{L}_{\mathbf{S}}(G')$ is $\min_\ell \lambda_2(\mathcal{L}(G_\ell))$, we obtain the first displayed inequalities. Moreover,

$$\bar{f}_{\cdot,k}^T \mathcal{L}_{\mathbf{S}}(G)\bar{f}_{\cdot,k} = \bar{f}_{\cdot,k}^T \mathbf{E}\bar{f}_{\cdot,k} + \bar{f}_{\cdot,k}^T \mathcal{L}_{\mathbf{S}}(G')\bar{f}_{\cdot,k} \leq (\|\mathbf{E}\|_2 + d(\mathbf{S}))\bar{f}_{\cdot,k}^T \bar{f}_{\cdot,k}.$$

Therefore

$$\bar{f}_{\cdot,k}^T(\mathbf{I} - \mathbf{P}_{\mathbf{S}}^r(G))\bar{f}_{\cdot,k} \leq \frac{1}{\lambda_{r+1}(\mathcal{L}_{\mathbf{S}}(G))} \bar{f}_{\cdot,k}^T \mathcal{L}_{\mathbf{S}}\bar{f}_{\cdot,k} \leq \frac{\|\mathcal{L}_{\mathbf{S}}(G) - \mathcal{L}_{\mathbf{S}}(G'))\|_2 + d(\mathbf{S})}{\lambda_{r+1}(\mathcal{L}_{\mathbf{S}}(G))} \|\bar{f}_{\cdot,k}\|_2^2.$$

The result follows from the observation that $\bar{f}_{\cdot,k}^T(\mathbf{I} - \mathbf{P}_{\mathbf{S}}^r(G))\bar{f}_{\cdot,k} = \|\bar{f}_{\cdot,k} - \mathbf{P}_{\mathbf{S}}^r(G)\bar{f}_{\cdot,k}\|_2^2$. ■

In Theorem 5, normalization plays a direct role because $\mathbf{S}$ affects $\delta_r(\mathbf{S})$. The analysis is analogous to that of Section 3.4 and the conclusions there hold. Similar to Theorem 3, we can prove the following generalization bound using Theorem 5. For simplicity, we only consider a simple kernel $\mathbf{K}_0 = \mathbf{I}$, and take $p = 1$.

Theorem 6 *Let the assumptions of Theorem 5 hold. Consider the least squares loss* $\phi(f_j, y_j) = \sum_{k=1}^{K}(f_{j,k} - \delta_{k,y_j})^2$ *in (1) using the regularization condition (5) and* $\mathbf{K}_0 = \mathbf{I}$*. The generalization error with optimal* λ *can be bounded as:*

$$\mathbf{E}_{Z_n}\frac{1}{m-n}\sum_{j\in\bar{Z}_n}\mathbf{err}(\hat{f}_j, y_j) \leq 16\delta_r(\mathbf{S}) + 32\sqrt{r/n}.$$

Proof Using Theorem 5, it can be easily verified that

$$\frac{1}{m}\sum_{j=1}^{m}\phi(\bar{f}_j, y_j) + \lambda\sum_{k=1}^{K} f_{\cdot,k}^T\mathbf{K}^{-1}f_{\cdot,k} \leq \delta_r(\mathbf{S}) + \lambda m.$$

Since regularizing with $\mathbf{K}_0 = \mathbf{I}$ is equivalent with regularizing with $\mathbf{K}_0 = \mathbf{P}_{\mathbf{S}}^r(G)$, we can use $\mathbf{tr}(\mathbf{K}) = r$. Now using this estimate in Corollary 1, we have

$$\mathbf{E}_{Z_n}\frac{1}{m-n}\sum_{j\in\bar{Z}_n}\mathbf{err}(\hat{f}_j, y_j) \leq 16(\delta_r(\mathbf{S}) + \lambda m) + \frac{r}{\lambda nm}.$$

Optimizing over λ gives the desired bound. ■

Similar to Theorem 4, it is possible to prove a bound for general p in Theorem 6, but the estimation of $\mathbf{tr}_p(\mathbf{K})$ is more complicated than that of $\mathbf{tr}(\mathbf{K})$. We skip the derivation because the extra complication is not important for the purpose of this paper. Compared to Theorem 4, the advantage of dimension reduction in Theorem 6 is that the quantity $\mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y)$ is replaced by $\|\mathcal{L}_{\mathbf{S}}(G) - \mathcal{L}_{\mathbf{S}}(G')\|_2$, which is typically much smaller. Instead of a rigorous analysis, we shall just give a brief intuition. For simplicity we take $\mathbf{S} = \mathbf{I}$ so that we can ignore the variations caused by $\mathbf{S}$. The 2-norm of the symmetric error matrix $\mathcal{L}_{\mathbf{S}}(G) - \mathcal{L}_{\mathbf{S}}(G')$ is its largest eigenvalue, which is no more than the largest 1-norm of one of its row vectors. In contrast, $\mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y)$ behaves similar to the absolute sum of entries of the error matrix, which is m times more than the averaged 1-norm of its row vectors. Therefore if error is relatively uniform across rows, then $\mathbf{cut}(\mathcal{L}_{\mathbf{S}}, y)$ can be at an order of m times more than $\|\mathcal{L}_{\mathbf{S}}(G) - \mathcal{L}_{\mathbf{S}}(G')\|_2$.

5 Experiments

We experiment with the Laplacian regularization with the normalization methods discussed above, on synthesized data sets generated by controlling graph properties as well as three real-world data sets.

5.1 Experimental framework

The Laplacian matrix $\mathcal{L}$ is generated from a graph G so that $\mathcal{L}_{j,j'} = -w_{j,j'}$ for $j \neq j'$ and $\mathcal{L}_{j,j} = \deg_j(G)$. Using $\mathcal{L}$, we define matrix $\mathbf{K}$ as follows:

- *Unnormalized:* $\mathbf{K} = (\alpha\mathbf{I} + \mathcal{L})^{-1}$. That is, $\mathbf{S} = \mathbf{I}$. No scaling.

	classes #1, #2	classes #3–#10
graph1	(4,2)	(2,1)
graph2	(6,3)	(2,1)
graph3	(8,4)	(2,1)

Figure 1: Generation of graph 1–5. (c, e) in the table indicates that for each node, we randomly chose c nodes of the same class and connect it to them, and we randomly chose e nodes of other classes (introducing errors) and connect it to them. Edge weights are fixed to 1.

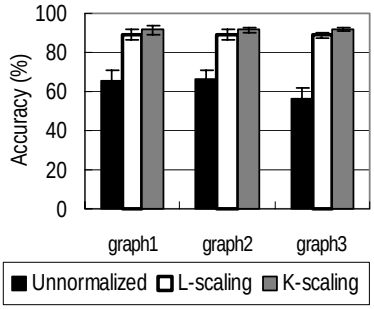

Figure 2: Classification accuracy (%) on the graphs where degrees are nearly constant within the class. $n = 40, m = 2000$. With dimension reduction (dim $\leq$ 20; chosen by cross validation). Average over 10 random splits with one standard deviation.

- $\mathbf{K}$-*scaling:* $\mathbf{K} = (\mathbf{S}^{-1/2}(\alpha\mathbf{I} + \mathcal{L})\mathbf{S}^{-1/2})^{-1}$ where $\mathbf{S} = \mathrm{diag}_j(\bar{\mathbf{K}}_{j,j}^{-1})$ with $\bar{\mathbf{K}} = (\alpha\mathbf{I} + \mathcal{L})^{-1}$. The diagonal entries of $\mathbf{K}$ are all ones.
- $\mathbf{L}$-*scaling:* $\mathbf{K} = (\alpha\mathbf{I} + \mathbf{S}^{-1/2}\mathcal{L}\mathbf{S}^{-1/2})^{-1}$ where $\mathbf{S} = \mathrm{diag}_j(\deg_j(G))$. The diagonal entries of $\mathbf{K}^{-1}$ are constant $(\alpha + 1)$. This is the standard degree-based scaling.

Using these three types of matrix $\mathbf{K}$, we test the following two types of regularization. One regularizes by $f^T\mathbf{K}^{-1}f$ using $\mathbf{K}$ without dimension reduction, as in Section 3. The other reduces the dimension of $\mathbf{K}^{-1}$ to r by leaving out all but several eigenvectors corresponding to the smallest r eigenvalues to obtain the eigenspace projector $\mathbf{P}_{\mathbf{S}}^r(G)$ and regularizes by:

$$\begin{cases} f^T\mathbf{K}^{-1}f & \text{if } \mathbf{P}_{\mathbf{S}}^r(G)f = f \\ +\infty & \text{otherwise} \end{cases}$$

as in Section 4. We use the one-versus-all strategy and use least squares as our loss function: $\phi_k(a, b) = (a - \delta_{k,b})^2$.

In related studies, the Laplacian or the adjacency matrix is either normalized using degrees like $\mathbf{L}$-scaling or unnormalized, e.g. [6, 13, 1]. We are interested in how well $\mathbf{K}$-scaling performs.

From m data points, n training labeled examples are randomly chosen while ensuring that at least one training example is chosen from each class. The remaining $m - n$ data points serve as test data. The regularization parameter λ is chosen by cross validation on the n training labeled examples. We will show performance when the rest of the parameters (α and dimensionality r) are also chosen by cross validation on the training labeled examples and when they are set to the

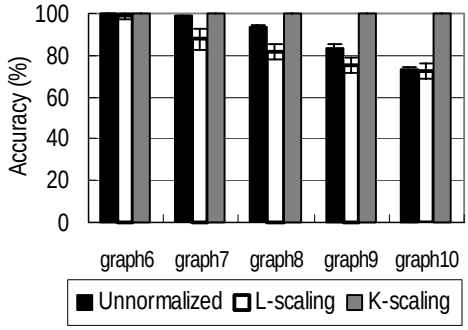

Figure 3: Classification accuracy on the core-satellite graphs. $n = 40, m = 2000$. With dimension reduction (dim ≤ 20; chosen by cross validation). Average over 10 random splits with one standard deviation.

optimum. The dimensionality r is chosen from $K, K+5, K+10, \cdots, 100$ where K is the number of classes unless otherwise specified. Our focus is on small n close to the number of classes. Throughout this section, we conduct 10 runs with random training/test splits and report the average accuracy.

5.2 Controlled data experiments

The purpose of the controlled data experiments is to observe the correlation of the effectiveness of the normalization methods with graph properties. The graphs we generate contain 2000 nodes, each of which is assigned one of 10 classes.

First, we show the results when dimension reduction is applied to the three types of matrix **K**. Figure 2 shows classification accuracy on three graphs that were generated so that the node degrees (of either correct edges or erroneous edges) are close to constant within each class but vary across classes. Details of their generation are described in Figure 1. We observe that on these graphs, both **K**-scaling and **L**-scaling significantly improve classification accuracy over the unnormalized baseline. There is no prominent difference between **K**-scaling's and **L**-scaling's performance.

Observe that **K**-scaling and **L**-scaling perform differently on the graphs used in Figure 3. These graphs have the following properties. Each class consists of *core nodes* and *satellite nodes*. Core nodes of the same class are tightly connected with each other and do not have any erroneous edges. Satellite nodes are relatively weakly connected to core nodes of the same class. The satellite nodes are also connected to some other classes' satellite nodes (i.e., introducing errors). This core-satellite model is intended to simulate real-world data in which some data points are close to the class boundaries (satellite nodes). More precisely, graphs 6–10 were generated as follows. Each graph consists of 2000 nodes ($m = 2000$) uniformly distributed over 10 classes ($K = 10$). 10% of the nodes are the core nodes. For every core node, we randomly choose 10 other core nodes of the same class and connect it to them with edge weight 1. For every satellite node, we randomly choose one core node of the same class and connect them with edge weight 0.01. Also, for each satellite node, we randomly choose one satellite node of some other class (i.e., introducing error) and connect them with edge weight w_e. We set the error edge weight $w_e = 0.002, 0.004, \cdots, 0.01$ for graphs $6, 7, \cdots, 10$, respectively. Note that although classes are uniformly distributed, pure components that optimize the generalization bound may be non-uniform in size. For graphs generated in this manner, degrees vary within the same class since the satellite nodes have smaller degrees than the core nodes. Our analysis suggests that **L**-scaling will do poorly. Figure 3 shows that on the five

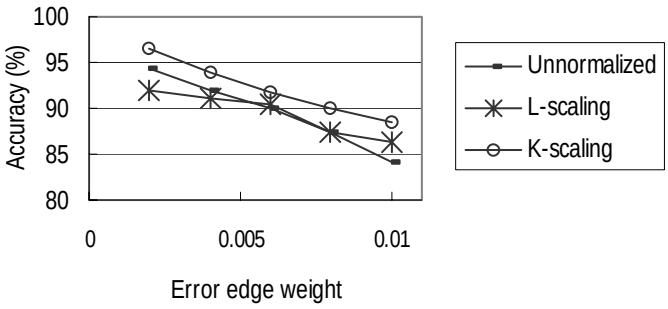

Figure 4: Classification accuracy on the core-satellite graphs. x-axis: error edge weight w_e. $n = 40, m = 2000$. Without dimension reduction. Average over 10 random splits.

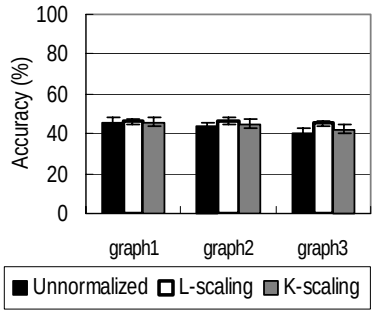

Figure 5: Classification accuracy (%) on the graphs where degrees are nearly constant within the class. Average over 10 random splits. $n = 40, m = 2000$. Without dimension reduction.

core-satellite graphs, **K**-scaling indeed produces higher performance than **L**-scaling. In particular, **K**-scaling does well even when **L**-scaling rather underperforms the unnormalized baseline.

Our analysis suggests that **K**-scaling should work well when the graph has relatively small error. This trend is more clearly observed on these core-satellite graphs without dimension reduction. As shown in Figure 4, the advantage of **K**-scaling over **L**-scaling is more prominent on the graphs with smaller error edge weights. On the other hand, the theory suggests that when the graph has large error (large cut), the benefit of normalization is less clear (since the derivation of **K**-scaling assumes near-zero cut). This is especially so when dimension reduction is not applied because as pointed out in Section 4, dimension reduction reduces error. This trend can be observed in Figure 5, which shows that on graphs 1–3 (having larger errors than the core-satellite graphs), neither **L**-scaling nor **K**-scaling prominently improves performance over the unnormalized Laplacian without dimension reduction though **L**-scaling seems to perform slightly better. Note that the performance without dimension reduction (Figure 5) is significantly worse than the performance with dimension reduction (Figure 2). This means that dimension reduction, which reduces error, is important when we try to apply graph based methods.

GPOL	Domestic politics	486
GSPO	Sports	407
GDIP	International relations	299
GCRIM	Crime, low enforcement	224
GJOB	Labor issues	206
GVIO	War, civil war	142
GDIS	Disasters and accidents	89
GHEA	Health	57
GENT	Arts, culture, entertainment	47
GENV	Environments	43
Total		2000

Figure 6: 10 RCV1 categories and their populations used in our experiments.

5.3 Real-world data experiments

5.3.1 Data and baseline

Our real-world data experiments use two image data sets (MNIST and UMIST) and one text data set (RCV1). The MNIST data set, downloadable from http://yann.lecun.com/exdb/mnist/, consists of hand-written digit image data (representing 10 classes, from digit "0" to "9"). For our experiments, we randomly choose 2000 images (i.e., $m = 2000$). The UMIST data set, downloadable from http://images.ee.umist.ac.uk/danny/database.html, consists of 575 face images taken from several angles of 20 people (representing 20 classes). The details of this data are described in [4]. We use all the images (i.e., $m = 575$). Reuters Corpus Version 1 (RCV1) consists of news articles labeled with topics. For our experiments, we chose 10 topics (representing 10 classes) that have relatively large populations and randomly chose 2000 articles that are labeled with exactly one of those 10 topics. The class distribution over these 2000 articles is non-uniform as shown in Figure 6.

To generate graphs from the image data, as is commonly done, we first generate the vectors of the gray-scale values of the pixels, and produce the edge weight between the i-th and the j-th data points $\mathbf{X}_i$ and $\mathbf{X}_j$ by $w_{i,j} = \exp(-||\mathbf{X}_i - \mathbf{X}_j||^2/t)$ where $t > 0$ is a parameter (RBF kernels). To generate graphs from the text data, we first create the bag-of-word vectors (without stemming and removing common stopwords) and then set $w_{i,j}$ based on RBF as above or set $w_{i,j}$ to the inner product of $\mathbf{X}_i$ and $\mathbf{X}_j$ (linear kernels). Optionally, we zero out all $w_{i,j}$ but k nearest neighbors (i.e., i is j's k nearest neighbors or j is i's k nearest neighbors) to reduce error in graphs and refer to it as the RBF (or linear) kernel with kNN.

As our baseline, we also test the supervised configuration by letting $\mathbf{W} + \beta\mathbf{I}$ (where $\mathbf{W}$ is a weight matrix whose (i, j)-entry is $w_{i,j}$) be the kernel matrix and using the same least squares loss function. We set β to the optimum, which was 0.001 for the RBF kernel for RCV1 and 1 for the other graphs.

5.3.2 Results

Figure 7 shows performance in relation to the number of labeled examples (n) on the MNIST data set. The comparison of the three bold lines (representing the methods with dimension reduction)

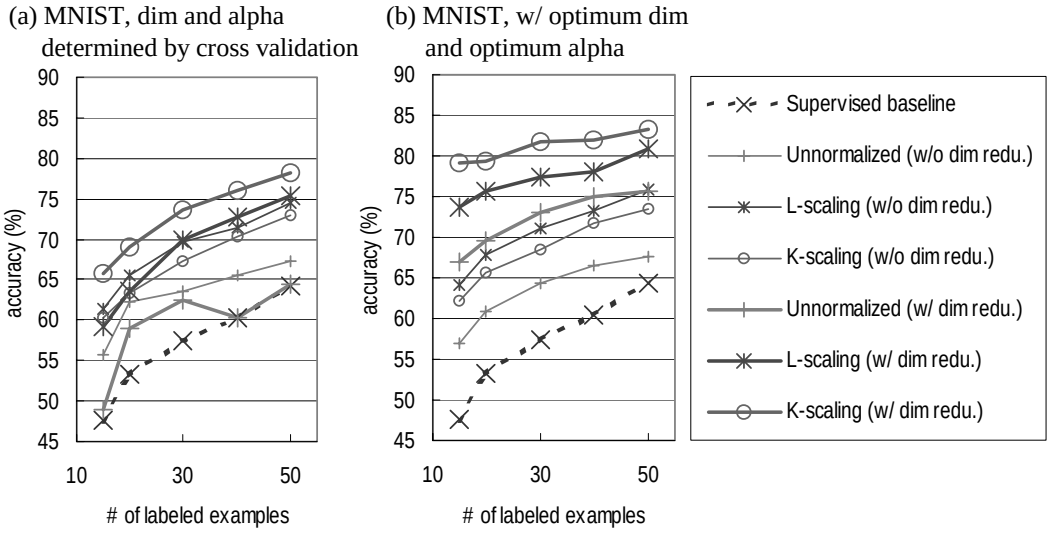

Figure 7: Classification accuracy (%) in relation to the number of labeled examples (n) on MNIST. $m = 2000$. (a) Dimensionality and α were determined by cross validation. (b) Dimensionality and α were set to the optimum. Average over 10 random splits.

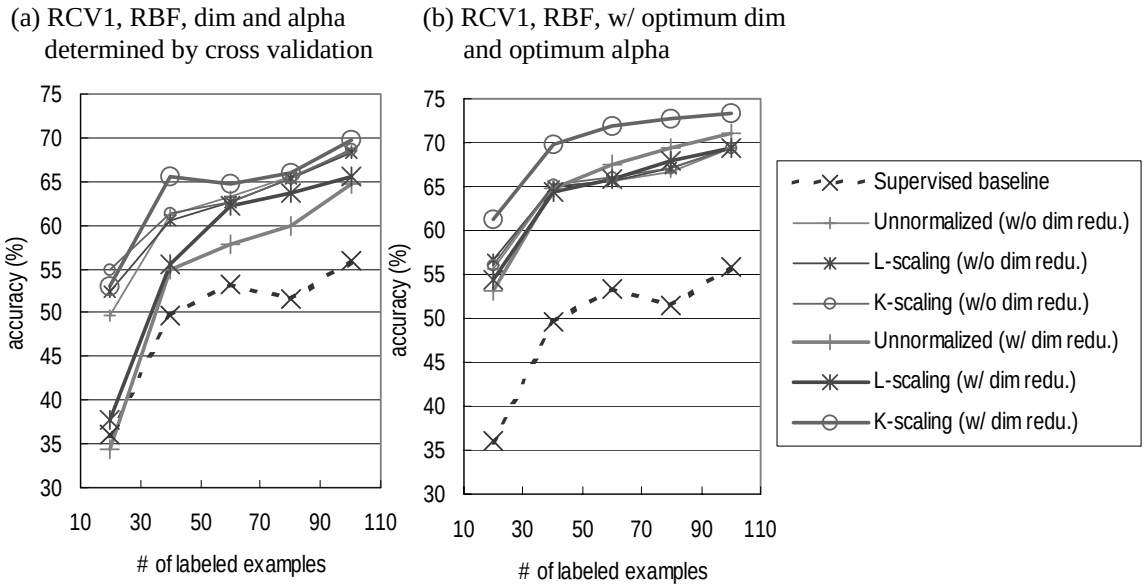

Figure 8: Classification accuracy (%) in relation to the number of labeled examples (n) on RCV1. RBF kernel (with $t = 0.25$). $m = 2000$. (a) Dimensionality and α were determined by cross validation. (b) Dimensionality and α were set to the optimum. Performance differences of the best performing method '**K**-scaling (w/ dim redu.)' from '**L**-scaling (w/ dim redu.)' and 'Unnormalized (w/ dim redu.)' are statistically significant ($p \leq 0.01$) in both the settings (a) and (b).

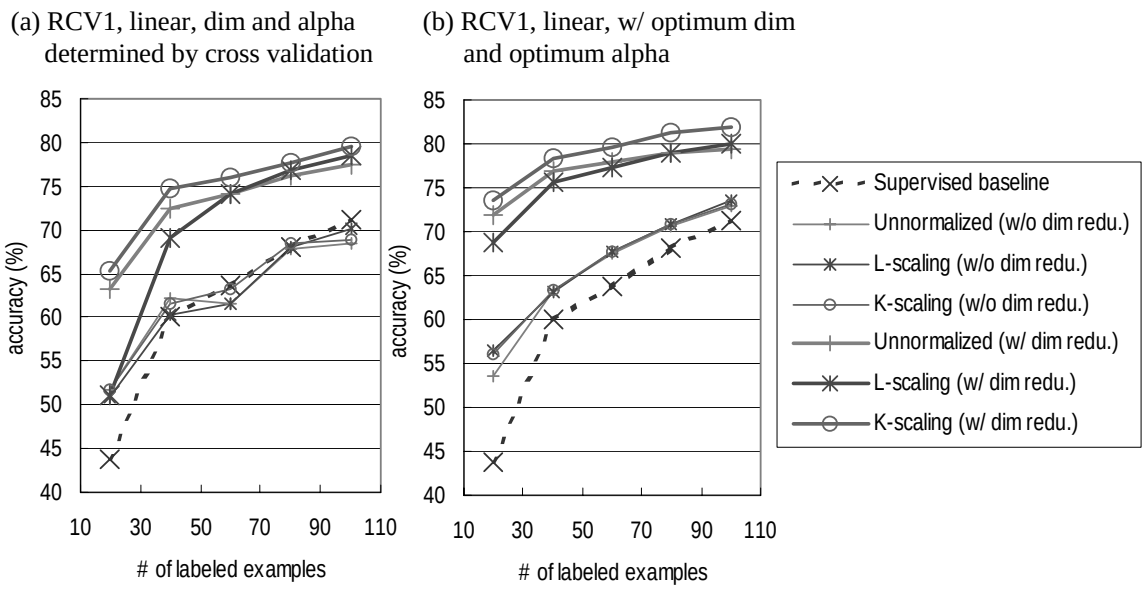

Figure 9: Classification accuracy (%) in relation to the number of labeled examples (n) on RCV1. Linear kernel. $m = 2000$. (a) Dimensionality and α were determined by cross validation. (b) Dimensionality and α were set to the optimum. Performance differences of the best performing method '**K**-scaling (w/ dim redu.)' from the second and third best '**L**-scaling (w/ dim redu.)' and 'Unnormalized (w/ dim redu.)' are statistically significant ($p \leq 0.01$) in both the settings (a) and (b).

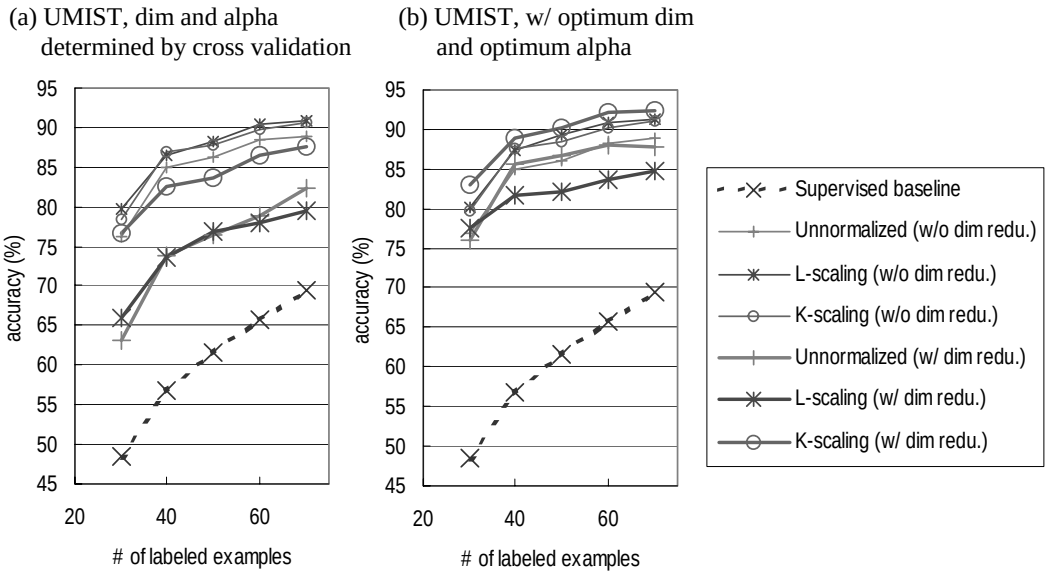

Figure 10: Classification accuracy (%) in relation to the number of labeled examples (n) on UMIST. $m = 575$. (a) Dimensionality and α were determined by cross validation. (b) Dimensionality and α were set to the optimum. In (b), performance differences of the best performing method '**K**-scaling (w/ dim redu.)' from the second and third best '**K**-scaling (w/o dim redu.)' and '**L**-scaling (w/o dim redu.)' are statistically significant ($p \leq 0.01$).

in Figure 7 (a) shows that when the dimensionality and α are determined by cross validation, **K**-scaling outperforms **L**-scaling, and **L**-scaling outperforms the unnormalized Laplacian. These performance differences are statistically significant ($p \leq 0.01$) based on the paired t test. The performance of the unnormalized Laplacian (with dimension reduction) is roughly consistent with the performance with similar (m, n) with heuristic dimension selection in [1]. Although without dimension reduction, **L**-scaling and **K**-scaling still improve performance over the unnormalized Laplacian, the best performance is always obtained by **K**-scaling with dimension reduction (the bold line with circles).

In Figure 7 (a), the unnormalized Laplacian with dimension reduction underperforms the unnormalized Laplacian without dimension reduction, indicating that dimension reduction rather degrades performance in this case. By comparing Figure 7 (a) and (b), we observe that this seemingly conter-intuitive performance trend is caused by the difficulty in choosing the right dimensionality by cross validation. Figure 7 (b) shows the performance at the optimum dimensionality and the optimum α. As observed, if the optimum dimensionality is known (as in (b)), dimension reduction improves performance either with or without normalization by **K**-scaling and **L**-scaling, and that all the transductive configurations outperform the supervised baseline. We also note that the comparison of Figure 7 (a) and (b) shows that choosing good dimensionality by cross validation is much harder than choosing α by cross validation especially when the number of labeled examples is small. This trend was observed also on the other data sets we experimented.

On the RCV1 data set, the performance trend is essentially similar to that of MNIST. Figure 8 shows the performance on RCV1 using the RBF kernel ($t = 0.25$, 100NN). In the setting of Figure 8 (a) where the dimensionality and α were determined by cross validation, **K**-scaling with dimension reduction generally performs the best. By setting the dimensionality and α to the optimum, the benefit of **K**-scaling with dimension reduction is even clearer (Figure 8 (b)).

On text data like RCV1, linear kernels (instead of RBF) are often used. Figure 9 shows the performance with linear kernels with 100NN. Again, **K**-scaling with dimension reduction performs the best. Its performance differences from the second and third best '**L**-scaling (w/ dim redu.)' and 'Unnormalized (w/ dim redu.)' are statistically significant ($p \leq 0.01$) in both Figure 9 (a) and (b).

In Figure 10, we observe that dimension reduction seems less useful on the UMIST data set. We conjecture that this may be because UMIST differs from our other data sets in that it is much more 'sparse'; UMIST has a smaller number of data points ($m = 575$ vs. $m = 2000$) while it has more classes ($K = 20$ vs. $K = 10$). Nevertheless, when the dimensionality and α are set to the optimum (Figure 10 (b)), again, **K**-scaling with dimension reduction performs the best. Its differences from the second and the third best methods (**K**-scaling without dimension reduction and **L**-scaling without dimension reduction) are statistically significant ($p \leq 0.01$).

Overall, on these graphs generated from image and text data sets, **K**-scaling with dimension reduction consistently outperformed the others. But without dimension reduction, **K**-scaling and **L**-scaling were not always effective. Transductive learning (either with or without normalization) generally improved performance.

5.4 Approximation of K-scaling

Although **K**-scaling consistently improves performance as shown above, its drawback is the relatively large runtime as it involves the computation of the inverse of an m-by-m matrix. We propose a less computationally-intensive approximation method using a known fact that $(\mathbf{I}-\mathbf{A})^{-1} = \sum_{k=0}^{\infty} \mathbf{A}^k$

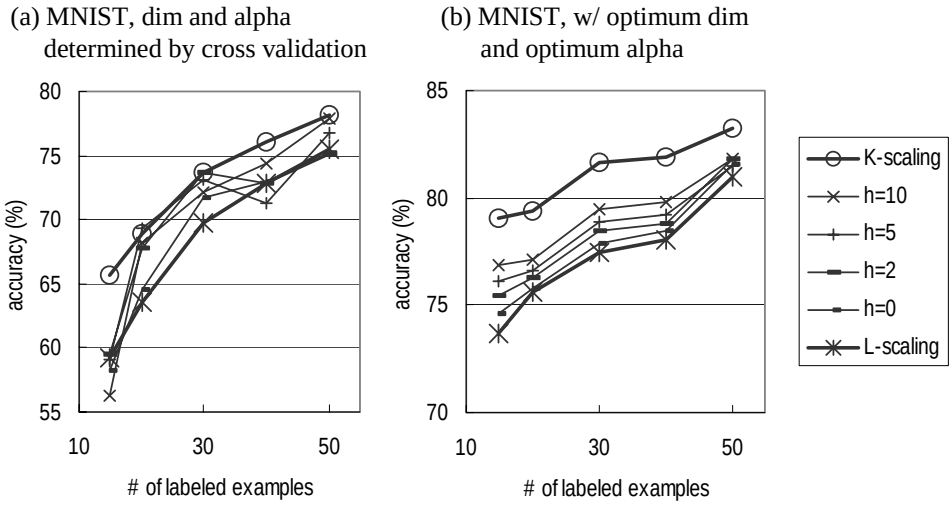

Figure 11: Classification accuracy (%) of the approximation method using $\hat{\mathbf{K}}(h)$. MNIST. (a) Dimensionality and α were determined by cross validation. (b) Dimensionality and α were set to the optimum.

if $||\mathbf{A}||_2 < 1$. As in the introduction, let $\mathbf{D} = \mathrm{diag}_i(\deg_i(G))$, and let $\mathbf{W}$ be a weight matrix such that $\mathbf{W}_{i,j} = w_{i,j}$ so that we can write $\mathcal{L} = \mathbf{D} - \mathbf{W}$. Let $\hat{\mathbf{D}} = \mathbf{D} + \alpha\mathbf{I}$. We define $\hat{\mathbf{K}}(h)$ to be the h-th order approximation of $\bar{\mathbf{K}} = (\mathcal{L} + \alpha\mathbf{I})^{-1}$ as follows:

$$\hat{\mathbf{K}}(h) = \hat{\mathbf{D}}^{-1/2} \left(\sum_{k=0}^{h} \left(\hat{\mathbf{D}}^{-1/2} \mathbf{W} \hat{\mathbf{D}}^{-1/2} \right)^k \right) \hat{\mathbf{D}}^{-1/2} .$$

We then set the i-th scaling factor $\mathbf{S}_i$ so that:

$$\mathbf{S}_i = \hat{\mathbf{K}}(h)_{i,i}^{-1} .$$

Since $\lim_{h\to\infty} \hat{\mathbf{K}}(h) = \bar{\mathbf{K}}$, the scaling factors produced with a sufficiently large h closely approximate $\mathbf{K}$-scaling. On the other hand, since $\hat{\mathbf{K}}(0) = \hat{\mathbf{D}}^{-1} = (\mathbf{D} + \alpha\mathbf{I})^{-1}$, the scaling factors produced by $\hat{\mathbf{K}}(0)$ with $\alpha = 0$ are exactly the same as $\mathbf{L}$-scaling (or the standard degree-scaling).

Figure 11 shows the performance of this approximation method with $h = 0, 2, 5, 10$ with dimension reduction in comparison with corresponding $\mathbf{K}$-scaling and $\mathbf{L}$-scaling on MNIST. In Figure 11 (b), we observe that at the optimum dimensionality and α, the performance of the approximation method lies exactly between that of $\mathbf{L}$-scaling and $\mathbf{K}$-scaling, and it approaches to $\mathbf{K}$-scaling as the order h increases. Intuitively, with a larger h, this approximation method takes more and more global connections into account and improves performance.

6 Conclusion

We derived generalization bounds for multi-category classification on graphs with Laplacian regularization, using geometric properties of the graph. In particular, we used this analysis to obtain a better understanding of the role of normalization of the graph Laplacian matrix as well as the effect of dimension reduction. We argued that the standard $\mathbf{L}$-scaling normalization method has

the undesirable property that the normalization factors can vary significantly within a pure component. An alternate normalization method, which we call **K**-scaling, is proposed to remedy the problem. Experiments confirm the superiority of **K**-scaling combined with dimension reduction.

A Proof of Theorem 1

The proof employs the stability analysis of [11], and is similar to the proof of a related bound for binary-classification in [12]. We shall introduce the following notation. let $i_{n+1} \neq i_1, \ldots, i_n$ be an integer randomly drawn from $\bar{Z}_n$, and let $Z_{n+1} = Z_n \cup \{i_{n+1}\}$. Let $\hat{f}(Z_{n+1})$ be the semi-supervised learning method (1) using training data in Z_{n+1}:

$$\hat{f}(Z_{n+1}) = \arg \inf_{f \in R^{mK}} \left[\frac{1}{n} \sum_{j \in Z_{n+1}} \phi(f_j, y_j) + \lambda f^T \mathbf{Q}_{\mathbf{K}} f \right].$$

We have the following stability lemma (a related result can be found in [11]);

Lemma 1 *The following inequality holds for each $k = 1, \ldots, K$:*

$$|\hat{f}_{i_{n+1},k}(Z_{n+1}) - \hat{f}_{i_{n+1},k}(Z_n)| \leq |\nabla_{1,k}\phi(\hat{f}_{i_{n+1}}(Z_{n+1}), y_{i_{n+1}})| \mathbf{K}_{i_{n+1},i_{n+1}}/(2\lambda n),$$

where $\nabla_{1,k}\phi(f_i, y)$ denotes a sub-gradient of $\phi(f_i, y)$ with respect to $f_{i,k}$, where $f_i = [f_{i,1}, \ldots, f_{i,K}]$.

Proof From [7], we know that there exist sub-gradients of $\nabla_{1,k}\phi$ such that the following first-order condition for the optimization problem (1) holds:

$$-2\lambda n \mathbf{K}^{-1} \hat{f}_{\cdot,k}(Z_n) = \sum_{j \in Z_n} \nabla_{1,k}\phi(\hat{f}_j(Z_n), y_j) e_j,$$

where e_j is the m-dimensional vector with all zeros except for the j-component with value one. Similarly, we have

$$-2\lambda n \mathbf{K}^{-1} \hat{f}_{\cdot,k}(Z_{n+1}) = \sum_{j \in Z_{n+1}} \nabla_{1,k}\phi(\hat{f}_j(Z_{n+1}), y_j) e_j.$$

Now, for simplicity, let $g = \hat{f}(Z_n)$ and $h = \hat{f}(Z_{n+1})$. By subtracting the above two equations, and then taking the inner product with $h_{\cdot,k} - g_{\cdot,k}$, we obtain

$$\begin{aligned} -2\lambda n (h_{\cdot,k} - g_{\cdot,k})^T \mathbf{K}^{-1} (h_{\cdot,k} - g_{\cdot,k}) =& \nabla_{1,k}\phi(h_{i_{n+1}}, y_{i_{n+1}})(h_{i_{n+1},k} - g_{i_{n+1},k}) \\ &+ \sum_{j \in Z_n} (\nabla_{1,k}\phi(h_j, y_j) - \nabla_{1,k}\phi(g_j, y_j))(h_{j,k} - g_{j,k}). \end{aligned}$$

Note that if $c(s)$ is a convex function of s, then it is easy to verify that $(\nabla c(s_1) - \nabla c(s_2))(s_1 - s_2) \geq 0$. Therefore we have $\sum_{j \in Z_n} (\nabla_{1,k}\phi(h_j, y_j) - \nabla_{1,k}\phi(g_j, y_j))(h_{j,k} - g_{j,k}) \geq 0$. This implies that

$$2\lambda n (h_{\cdot,k} - g_{\cdot,k})^T \mathbf{K}^{-1} (h_{\cdot,k} - g_{\cdot,k}) \leq -\nabla_{1,k}\phi(h_{i_{n+1}}, y_{i_{n+1}})(h_{i_{n+1},k} - g_{i_{n+1},k}).$$

Using Cauchy-Schwartz inequality, we have

$$\begin{aligned}2\lambda n(h_{i_{n+1},k}-g_{i_{n+1},k})^2 =& 2\lambda n((h_{\cdot,k}-g_{\cdot,k})^T e_{i_{n+1}})^2\\ \leq& 2\lambda n(h_{\cdot,k}-g_{\cdot,k})^T\mathbf{K}^{-1}(h_{\cdot,k}-g_{\cdot,k})e_{i_{n+1}}^T\mathbf{K}e_{i_{n+1}}\\ \leq& |\nabla_{1,k}\phi(h_{i_{n+1}},y_{i_{n+1}})|\cdot|h_{i_{n+1},k}-g_{i_{n+1},k}|\mathbf{K}_{i_{n+1},i_{n+1}}.\end{aligned}$$

Therefore we have $|h_{i_{n+1},k}-g_{i_{n+1},k}|\leq|\nabla_{1,k}\phi(h_{i_{n+1}},y_{i_{n+1}})|\mathbf{K}_{i_{n+1},i_{n+1}}/(2\lambda n)$. ■

Lemma 2 *The following inequality holds*

$$\mathbf{err}(\hat{f}_{i_{n+1}}(Z_n),y_{i_{n+1}})\leq\sup_{k=k_0,i_{n+1}}\left[\frac{1}{a}\phi_0(\hat{f}_{i_{n+1},k}(Z_{n+1}),\delta_{i_{n+1},k})+\left(\frac{b}{c\lambda n}\mathbf{K}_{i_{n+1},i_{n+1}}\right)^p\right].$$

Proof If $\hat{f}(Z_n)$ does not make an error on the i_{n+1}-th example. That is, if $\mathbf{err}(\hat{f}_{i_{n+1}}(Z_n),y_{i_{n+1}})=0$, then the inequality automatically holds.

Now, assume that $\hat{f}(Z_n)$ makes an error on the i_{n+1}-th example, that is, $\mathbf{err}(\hat{f}_{i_{n+1}}(Z_n),y_{i_{n+1}})=1$. Then there exists $k_0\neq y_{i_{n+1}}$ such that $\hat{f}_{i_{n+1},y_{i_{n+1}}}(Z_n)\leq\hat{f}_{i_{n+1},k_0}(Z_n)$. If we let $d=(\inf\{x:\phi_0(x,1)\leq a\}+\sup\{x:\phi_0(x,0)\leq a\})/2$, then either $\hat{f}_{i_{n+1},y_{i_{n+1}}}(Z_n)\leq d$ or $\hat{f}_{i_{n+1},k_0}(Z_n)\geq d$. By the definition of c, either we have $\inf\{x:\phi_0(x,1)\leq a\}-\hat{f}_{i_{n+1},y_{i_{n+1}}}(Z_n)\geq c/2$ or we have $\hat{f}_{i_{n+1},k_0}(Z_n)-\sup\{x:\phi_0(x,0)\leq a\}\geq c/2$. It follows that there exists $k=k_0$ or $k=y_{i_{n+1}}$ such that either $\phi_0(\hat{f}_{i_{n+1},k}(Z_{n+1}),\delta_{y_{i_{n+1}},k})\geq a$ or $\left|\hat{f}_{i_{n+1},k}(Z_{n+1})-\hat{f}_{i_{n+1},k}(Z_n)\right|\geq c/2$. Using Lemma 1, we have either $\phi_0(\hat{f}_{i_{n+1},k}(Z_{n+1}),\delta_{y_{i_{n+1}},k})\geq a$ or $b\mathbf{K}_{i_{n+1},i_{n+1}}/(2\lambda n)\geq c/2$, implying that

$$\frac{1}{a}\phi_0(\hat{f}_{i_{n+1},k}(Z_{n+1}),\delta_{y_{i_{n+1}},k})+\left(\frac{b\mathbf{K}_{i_{n+1},i_{n+1}}}{c\lambda n}\right)^p\geq 1=\mathbf{err}(\hat{f}_{i_{n+1}}(Z_n),y_{i_{n+1}}).$$

■

We are now ready to prove Theorem 1. For every $j\in Z_{n+1}$, denote by $Z_{n+1}^{(j)}$ the subset of n samples in Z_{n+1} with the j-th data point left out. From Lemma 2, we have

$$\mathbf{err}(\hat{f}_j(Z_n^{(j)}),y_j)\leq\frac{1}{a}\phi(\hat{f}_j(Z_{n+1}),y_j)+\left(\frac{b}{c\lambda n}\mathbf{K}_{j,j}\right)^p.$$

We thus obtain for all $f\in R^{mK}$:

$$\begin{aligned}\sum_{j\in Z_{n+1}}\mathbf{err}(\hat{f}_j(Z_n^{(j)}),y_j)\leq&\frac{1}{a}\sum_{j\in Z_{n+1}}\phi(\hat{f}_j(Z_{n+1}),y_j)+\sum_{j\in Z_{n+1}}\left(\frac{b}{c\lambda n}\mathbf{K}_{j,j}\right)^p\\ \leq&\frac{1}{a}\left[\sum_{j\in Z_{n+1}}\phi(f_j,y_j)+\lambda f^T\mathbf{Q}_{\mathbf{K}}f\right]+\sum_{j\in Z_{n+1}}\left(\frac{b}{c\lambda n}\mathbf{K}_{j,j}\right)^p.\end{aligned}$$

Therefore

$$
\begin{aligned}
&\mathbf{E}_{Z_n}\frac{1}{m-n}\sum_{j\in\bar{Z}_n}\mathbf{err}(\hat{f}_j(Z_n),y_j)\\
\leq&\frac{1}{n+1}\mathbf{E}_{Z_{n+1}}\sum_{j\in Z_{n+1}}\mathbf{err}(\hat{f}_j(Z_n^{(j)}),y_j)\\
\leq&\frac{n}{a(n+1)}\mathbf{E}_{Z_{n+1}}\left[\frac{1}{n}\sum_{j\in Z_{n+1}}\phi(f_j,y_j)+\lambda f^T\mathbf{Q}_{\mathbf{K}}f\right]+\frac{1}{n+1}\mathbf{E}_{Z_{n+1}}\sum_{j\in Z_{n+1}}\left(\frac{b}{c\lambda n}\mathbf{K}_{j,j}\right)^p\\
=&\frac{1}{a}\left[\frac{1}{m}\sum_{j=1}^{m}\phi(f_j,y_j)+\frac{\lambda n}{n+1}f^T\mathbf{Q}_{\mathbf{K}}f\right]+\frac{1}{m}\sum_{j=1}^{m}\left(\frac{b\mathbf{K}_{j,j}}{c\lambda n}\right)^p.
\end{aligned}
$$

References

[1] Mikhail Belkin and Partha Niyogi. Semi-supervised learning on Riemannian manifolds. *Machine Learning*, Special Issue on Clustering:209–239, 2004.

[2] Fan R.K. Chung. *Spectral Graph Theory*. Regional Conference Series in Mathematics. American Mathematical Society, Rhode Island, 1998.

[3] Chris Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst Simon. A min-max cut algorithm for graph partitioning and data clustering. In *IEEE Int'l Conf. Data Mining*, pages 107–114, 2001.

[4] Daniel B. Graham and Nigel M. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences*, 163:446–456, 1998.

[5] G.R.G. Lanckriet, N. Cristianini, L.El Ghaoui, P.L. Bartlett, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[6] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[7] R. Tyrrell Rockafellar. *Convex analysis*. Princeton University Press, Princeton, NJ, 1970.

[8] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell*, 22:888–905, 2000.

[9] M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. In *NIPS 2001*, 2002.

[10] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *NIPS*, 2003.

[11] Tong Zhang. Leave-one-out bounds for kernel methods. *Neural Computation*, 15:1397–1437, 2003.

[12] Tong Zhang and Rie K Ando. Analysis of spectral kernel design based semi-supervised learning. In *NIPS*, 2006.

[13] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schlkopf. Learning with local and global consistency. In *NIPS 2003*, pages 321–328, 2004.

[14] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML 2003*, 2003.