

Regularization and Semi-supervised Learning on Large Graphs

Mikhail Belkin, Irina Matveeva, and Partha Niyogi

The University of Chicago, Department of Computer Science
{misha, matveeva, niyogi}@cs.uchicago.edu

Abstract. We consider the problem of labeling a partially labeled graph. This setting may arise in a number of situations from survey sampling to information retrieval to pattern recognition in manifold settings. It is also of potential practical importance, when the data is abundant, but labeling is expensive or requires human assistance.

Our approach develops a framework for regularization on such graphs. The algorithms are very simple and involve solving a single, usually sparse, system of linear equations. Using the notion of algorithmic stability, we derive bounds on the generalization error and relate it to structural invariants of the graph. Some experimental results testing the performance of the regularization algorithm and the usefulness of the generalization bound are presented.

1 Introduction

In pattern recognition problems, there is a probability distribution P according to which labeled and possibly unlabeled examples are drawn and presented to a learner. This P is usually far from uniform and therefore might have some non-trivial geometric structure. We are interested in the design and analysis of learning algorithms that exploit this geometric structure. For example, P may have support on or close to a manifold. In discrete settings, it may have support on a graph. In this paper we consider the problem of predicting the labels on vertices of a partially labeled graph. Our goal is to design algorithms that are adapted to the structure of the graph. Our analysis shows that the generalization ability of such algorithms is controlled by geometric invariants of the graph.

Consider a weighted graph $G = (V, E)$ where $V = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ is the vertex set and E is the edge set. Associated with each edge $e_{ij} \in E$ is a weight W_{ij} . If there is no edge present between \mathbf{x}_i and \mathbf{x}_j , $W_{ij} = 0$. Imagine a situation where a subset of these vertices are labeled with values $y_i \in \mathbb{R}$. We wish to predict the values of the rest of the vertices. In doing so, we would like to exploit the structure of the graph. In particular, in our approach we will assume that the weights are indications of the affinity of nodes with respect to each other and consequently are related to the potential similarity of the y values these nodes are likely to have. Ultimately we propose an algorithm for regularization on graphs.

This general problem arises in a number of different settings. For example, in survey sampling, one has a database of individuals along with their preference profiles that determines a graph structure based on similarity of preferences. One wishes to estimate a survey variable (e.g. hours of TV watched, amount of cheese consumed, etc.). Rather than survey the entire set of individuals every time, which might be impractical, one may sample a subset of the individuals and then attempt to infer the survey variable for the rest of the individuals. In Internet and information retrieval applications, one is often in possession of a database of objects that have a natural graph structure (or more generally affinity matrix). One may wish to categorize the objects into various classes but only a few (object, class) pairs may be obtained by access to a supervised oracle. In the Finite Element Method for solving PDEs, one sometimes evaluates the solution at some of the points of the finite element mesh and one needs to estimate the value of the solution at all other points. A final example arises when data is obtained by sampling an underlying manifold embedded in a high dimensional space. In recent approaches to dimensionality reduction, clustering and classification in this setting, a graph approximation to the underlying manifold is computed. Semi-supervised learning in this manifold setting reduces to a partially labeled classification problem of the graph. This last example is an instantiation of transductive learning where other approaches include the Naive Bayes for text classification in [11], transductive SVM [14, 9], the graph mincut approach in [2], and the random walk on the adjacency graph in [13]. We also note the closely related work [10], which uses kernels and in particular diffusion kernels on graphs for classification.

In the manifold setting the graph is easily seen to be an empirical object. It is worthwhile to note that in all applications of interest, even those unrelated to the manifold setting, the graph reflects pairwise relationships on the data, and hence is an empirical object whenever the data consists of random samples.

We consider this problem in some generality and introduce a framework for regularization on graphs. Two algorithms are derived within this framework. The resulting optima have simple analytical expressions. If the graph is sparse, the algorithms are fast and, in particular, do not require the computation of multiple eigenvectors as is common in many spectral methods (including our previous approach [1]). Another advantage of the current framework is that it is possible to provide theoretical guarantees for generalization error. Using techniques from algorithmic stability we show that generalization error is bounded in terms of the smallest nontrivial eigenvalue (Fiedler number) of the graph. Interestingly, it suggests that generalization performance depends on the geometry of the graph rather than on its size. Finally some experimental evaluation is conducted suggesting that this approach to partially labeled classification is competitive.

Several groups of researchers have been investigating related ideas. In particular, [12] also proposed algorithms for graph regularization. In [16] the authors propose the Label Propagation algorithm for semi-supervised learning, which is similar to our Interpolated Regularization when $S = L$. In [15] a somewhat different regularizer together with the normalized Laplacian is used for semi-

supervised learning. The ideas of spectral clustering motivated the authors of [4] to introduce Cluster Kernels for semi-supervised learning. The authors suggest explicitly manipulating eigenvalues of the kernel matrix.

2 Regression on Graphs

2.1 Regularization and Regression on Graphs

To approximate a function on a graph G , with the weight matrix W_{ij} we need a notion of a “good” function. One way to think about such a function is that is that it does not make too many “jumps”. We formalize that notion (see also our earlier paper [1]), by the smoothness functional

$$\mathcal{S}(f) = \sum_{i \sim j} W_{ij} (f_i - f_j)^2$$

where the sum is taken over the adjacent vertices of G . For “good” functions f the functional \mathcal{S} takes small values.

It is important to observe that

$$\sum_{i \sim j} W_{ij} (f_i - f_j)^2 = \mathbf{f}^T L \mathbf{f}$$

where L is the Laplacian $L = D - W$, $D = \text{diag}(\sum_i W_{1i}, \dots, \sum_i W_{ni})$. This is a basic identity in the spectral graph theory and provides some intuition for the remarkable properties of the graph Laplacian L .

Other smoothness matrices, such as L^p , $p \in \mathbb{N}$, $\exp(-tL)$, $t \in \mathbb{R}$ are also possible. In particular, L^2 often seems to work well in practice.

2.2 Algorithms for Regression on Graphs

Let $G = (V, E)$ be a graph with n vertices and the weight matrix W_{ij} . For the purposes of this paper we will assume that G is connected and that the vertices of the graph are numbered. We would like to regress a function $f : V \rightarrow \mathbb{R}$. f is defined on vertices of G , however we have only partial information, say for the first k vertices. That is $f(\mathbf{x}_i) = y_i$, $1 \leq i \leq k$. The labels can potentially be noisy. We also allow data points to have multiplicities, i.e. each vertex of the graph may appear more than once with same or different y value.

We precondition the data by mean subtracting first. That is we take

$$\tilde{\mathbf{y}} = (y_1 - \bar{y}, \dots, y_k - \bar{y})$$

where $\bar{y} = \frac{1}{k} \sum y_i$. This is needed for stability of the algorithms as will be seen in the theoretical discussion.

Algorithm 1: Tikhonov regularization (parameter $\gamma \in \mathbb{R}$). The objective is to minimize the square loss function plus the smoothness penalty.

$$\tilde{\mathbf{f}} = \underset{\substack{\mathbf{f}=(f_1, \dots, f_n) \\ \sum f_i=0}}{\text{argmin}} \frac{1}{k} \sum_i (f_i - \tilde{y}_i)^2 + \gamma \mathbf{f}^T L \mathbf{f}$$

S here is a smoothness matrix, e.g. $S = L$ or $S = L^p$, $p \in \mathbb{N}$. The condition $\sum f_i = 0$ is needed to make the algorithm stable. It can be seen by following the proof of Theorem 1 that necessary stability and the corresponding generalization bound cannot be obtained unless the regularization problem is constrained to functions with mean 0.

Without the loss of generality we can assume that the first l points on the graph are labeled. l might be different from the number of sample points k , since we allow vertices to have different labels (or the same label several times).

The solution to the quadratic problem above is not hard to obtain by standard linear algebra considerations. If we denote by $\mathbf{1} = (1, 1, \dots, 1)$ the vector of all ones, the solution can be given in the form

$$\tilde{\mathbf{f}} = (k\gamma S + I_k)^{-1}(\tilde{\mathbf{y}} + \mu \mathbf{1})$$

Here $\tilde{\mathbf{y}}$ is the n -vector $\mathbf{y} = (\sum_i y_{1i}, \sum_i y_{2i}, \dots, \sum_i y_{li}, 0, \dots, 0)$, where we sum the labels corresponding to the same vertex on the graph.

I_k is a diagonal matrix of multiplicities

$$I_k = \text{diag}(n_1, n_2, \dots, n_l, 0, \dots, 0)$$

where n_i is the number of occurrences of vertex i among the labeled point in the sample. μ is chosen so that the resulting vector \mathbf{f} is orthogonal to $\mathbf{1}$. Denote by $s(\mathbf{f})$ the functional

$$s : \mathbf{f} \rightarrow \sum_i f_i$$

Since s is linear, we obtain $0 = s(\tilde{\mathbf{f}}) = s((k\gamma S + I_k)^{-1}\tilde{\mathbf{y}}) + s((k\gamma S + I_k)^{-1}\mathbf{1})$. Therefore we can write

$$\mu = -\frac{s((k\gamma S + I_k)^{-1}\tilde{\mathbf{y}})}{s((k\gamma S + I_k)^{-1}\mathbf{1})}$$

Note that dropping the condition $\mathbf{f} \perp \mathbf{1}$ is equivalent to putting $\mu = 0$.

Algorithm 2: Interpolated Regularization (no parameters).

Here we assume that the values y_1, \dots, y_k have no noise. Thus the optimization problem is to find a function of maximum smoothness satisfying $f(\mathbf{x}_i) = \tilde{y}_i$, $1 \leq i \leq k$:

$$\tilde{\mathbf{f}} = \underset{\substack{\mathbf{f}=(\tilde{y}_1, \dots, \tilde{y}_k, f_{k+1}, \dots, f_n) \\ \sum f_i=0}}{\text{argmin}} \quad \mathbf{f}^t S \mathbf{f}$$

As before S is a smoothness matrix, e.g. L or L^2 . However, here we are not allowing multiple vertices in the sample. We partition S as

$$S = \begin{pmatrix} S_1 & S_2 \\ S_2^T & S_3 \end{pmatrix}$$

where S_1 is a $k \times k$ matrix, S_2 is $k \times n - k$ and S_3 is $(n - k) \times (n - k)$. Let \tilde{f} be the values of f , where the function is unknown, $\tilde{f} = (f_{k+1}, \dots, f_n)$.

Straightforward linear algebra yields the solution:

$$\tilde{f} = S_3^{-1} S_2^T ((\tilde{y}_1, \dots, \tilde{y}_k)^T + \mu \mathbf{1})$$

$$\mu = -\frac{s(S_3^{-1} S_2^T \tilde{\mathbf{y}})}{s(S_3^{-1} S_2^T \mathbf{1})}$$

The regression formula is very simple and has no free parameters. However, the quality of the results depends on whether S_3 is well conditioned.

It can be shown that Interpolated Regularization is the limit case of Tikhonov regularization when γ tends to 0. That is, given a function f , and denoting by $\mathcal{R}eg_\gamma$ and $\mathcal{R}eg_{int}$, Tikhonov regularization and Interpolated regularization, respectively, we have

$$\lim_{\gamma \rightarrow 0} \mathcal{R}eg_\gamma(f) = \mathcal{R}eg_{int}(f)$$

That correspondence suggests using the condition $f \perp \mathbf{1}$ for interpolated regularization as well, even though no stability-based bounds are available in that case.

It is interesting to note that this condition, imposed for purely theoretical reasons, seems similar to class mass normalization step in [16].

3 Theoretical Analysis

In this section we investigate some theoretical guarantees for the generalization error of regularization on graphs. We use the notion of algorithmic stability, first introduced by Devroye and Wagner in [6] and later used by Bousquet and Elisseeff in [3] to prove generalization bounds for regularization networks.

The goal of a learning algorithm is to learn a function on some space V from examples. Given a set of examples T the learning algorithm produces a function $f_T : V \rightarrow \mathbb{R}$. Therefore a learning rule is a map from data sets into functions on V . We will be interested in the case where V is a graph.

The empirical risk $R_k(f)$ (with the square loss function) is a measure of how well we do on the training set:

$$R_k(f) = \frac{1}{k} \sum_{i=1}^k (f(\mathbf{x}_i) - y_i)^2$$

The generalization error $R(f)$ is the expectation of how well we do on all points, labeled or unlabeled.

$$R(f) = E_\mu (f(\mathbf{x}) - y(\mathbf{x}))^2$$

where the expectation is taken over an underlying distribution μ on $V \times \mathbb{R}$ according to which the labeled examples are drawn.

As before denote the smallest nontrivial eigenvalue of the smoothness matrix S by λ_1 . If S is the Laplacian of the graph, this value, first introduced by Fiedler

in [7] as algebraic connectivity and is sometimes known as the Fiedler constant, plays a key role in spectral graph theory. One interpretation of λ_1 is that it gives an estimate of how well V can be partitioned. We expect λ_1 to be relatively large, say $\lambda_1 > O(\frac{1}{n^r})$, $0 \leq r \ll 1$. For example for an n -dimensional hypercube $\lambda_1 = 2$. If λ_1 is very small, a sensible possibility would be to cut the graph in two, using the eigenvector corresponding to λ_1 and proceed with regularization separately for the two parts.

The theorem below states that as long as k is large and the values of the solution to the regularization problem are bounded, we get good generalization results. We note that the constant K can be bounded using the properties of the graph. See the propositions below for the details. We did not make these estimates a part of the Theorem 1 as it would make the formulas even more cumbersome.

Theorem 1 (Generalization Performance of Graph Regularization).

Let γ be the regularization parameter, T be a set of $k \geq 4$ vertices $\mathbf{x}_1, \dots, \mathbf{x}_k$, where each vertex occurs no more than t times, together with values y_1, \dots, y_k , $|y_i| \leq M$. Let f_T be the regularization solution using the smoothness functional S with the second smallest eigenvalue λ_1 . Assuming that $\forall \mathbf{x} |f_T(\mathbf{x})| \leq K$ we have with probability $1 - \delta$ (conditional on the multiplicity being no greater than t):

$$|R_k(f_T) - R(f_T)| \leq \beta + \sqrt{\frac{2 \log(2/\delta)}{k}} (k\beta + (K + M)^2)$$

where

$$\beta = \frac{3M\sqrt{tk}}{(k\gamma\lambda_1 - t)^2} + \frac{4M}{k\gamma\lambda_1 - t}$$

Proof. The theorem is obtained by rewriting the formula in the Theorem 4 in terms of k and then applying the Theorem 5.

We see that as usual in the estimates of the generalization error it decreases at a rate $\frac{1}{\sqrt{k}}$. It is important to note that the estimate is nearly independent of the total number of vertices n in the graph. We say “nearly” since the probability of having multiple points increases as k becomes close to n and since the value of λ_1 may (or may not) implicitly depend on the number of vertices.

The only thing that is missing is an estimate for K . Below we give two such estimates, one for the case of general S and the other, possibly sharper, when the smoothness matrix is the Laplacian $S = L$.

Proposition 1. *With λ_1 , M and γ as above we have the following inequality:*

$$\|f\|_\infty \leq \frac{M}{\sqrt{\lambda_1 \gamma}}$$

Proof. Let's first denote the quantity we are trying to minimize by $P(\mathbf{f})$:

$$P(\mathbf{f}) = \frac{1}{k} \sum_i (f_i - y_i)^2 + \gamma \mathbf{f}^t L \mathbf{f}^t$$

The first observation we make is that when $\mathbf{f} = 0$, $P(\mathbf{f}) = \frac{1}{k} \sum_i y_i^2 \leq M^2$. Thus, if $\tilde{\mathbf{f}}$ minimizes $P(\mathbf{f})$, we have $0 \leq \gamma \tilde{\mathbf{f}}^t L \tilde{\mathbf{f}} \leq M^2$. Recall that $f \in H$, where H is the linear space of vectors with mean 0 and that the smallest eigenvalue of S restricted to H is λ_1 . Therefore, recalling that $\|f\|_2 \geq \|f\|_\infty$, we obtain

$$\tilde{\mathbf{f}}^t L \tilde{\mathbf{f}} \geq \lambda_1 \|f\|^2 \geq \lambda_1 \|f\|_\infty^2$$

Thus

$$\|f\|_\infty \leq \sqrt{\frac{\tilde{\mathbf{f}}^t L \tilde{\mathbf{f}}}{\lambda_1}} \leq \frac{M}{\sqrt{\lambda_1 \gamma}}$$

A different inequality can be obtained when $S = L$. Note the the diameter of the graph is typically far smaller than the number of vertices. For example, when G is a n -cube, the number of vertices is 2^n , while the diameter is n .

Proposition 2. *Let $W = \min_{i \sim j} w_{ij}$ be the smallest nonzero weight of the graph G . Assume G is connected. Let D be the unweighted diameter of the graph, i.e. the maximum length of the shortest path between two points on the graph. Then the maximum entry K of the solution to the γ -regularization problem with y 's bounded by M satisfies the following inequality:*

$$K \leq M \sqrt{\frac{D}{\gamma W}}$$

A useful special case is

Corollary 2 *If all weights of G are either 0 or 1, then*

$$K \leq M \sqrt{\frac{D}{\gamma}}$$

Proof. Using the same notation as above, we see by substituting the 0 vector that if $\tilde{\mathbf{f}}$ minimizes $P(\mathbf{f})$, then $P\tilde{\mathbf{f}} \leq M^2$.

Let K be the biggest entry of $\tilde{\mathbf{f}}$ with the corresponding vertex v_1 . Take any vertex v_2 for which there is a $y \leq 0$. Such vertex exists, since the data has mean 0. Now let e_1, e_2, \dots, e_m be a sequence of edges on the graph connecting the vertices v_1 and v_2 . We put w_1, \dots, w_m to be the corresponding weights and let g_0, g_1, \dots, g_m be the values of $\tilde{\mathbf{f}}$ corresponding to the consecutive vertices of that sequence. Now let $h_i = g_i - g_{i-1}$ be the differences of values of $\tilde{\mathbf{f}}$ along that path. We have $\sum_i h_i = g_m - g_0 \geq K$.

Consider the minimum value Z of $\sum_i w_i h_i^2$, given that $\sum_i h_i \geq K$. Using Lagrangian multipliers, we see that the solution is given by $h_i = \frac{\alpha}{w_i}$. We find α using the condition $\sum_i h_i = \alpha \sum_i \frac{1}{w_i} = K$. Therefore

$$\sum_i w_i h_i^2 = \sum_i \frac{\alpha^2}{w_i} = \frac{K^2}{\sum_i \frac{1}{w_i}}$$

Recall that $\frac{m}{\sum_{i=1}^m \frac{1}{w_i}}$ is the harmonic mean of numbers w_i and is therefore greater than $\min(w_1, \dots, w_m)$. Thus we obtain

$$\sum_i w_i h_i^2 \geq \frac{K^2}{m} \min(w_1, \dots, w_m)$$

On the other hand, we see that

$$\tilde{\mathbf{f}}^t L \tilde{\mathbf{f}}^t = \sum_{i < j, i \sim j} w_{ij} (\tilde{f}_i - \tilde{f}_j)^2 \geq \sum_i w_i h_i^2$$

since the right-hand side of the inequality is a partial sum of the terms of the left-hand side.

Hence

$$P(\tilde{\mathbf{f}}) \geq \frac{K^2}{m} \min(w_1, \dots, w_m)$$

Recalling that $P(\tilde{\mathbf{f}}) \leq M^2$, we finally obtain:

$$K \leq \frac{M \sqrt{m}}{\sqrt{\gamma \min(w_1, \dots, w_m)}}$$

Since the path between those points can be chosen arbitrarily, we can choose it so that the length of the path m does not exceed the unweighted diameter D of the graph, which proves the theorem.

In particular, if all weights of G are either zero or one, we have:

$$K \leq \frac{M \sqrt{D}}{\sqrt{\gamma}}$$

assuming, of course, that G is connected.

To prove the main theorem we will use a result of Bousquet and Elisseeff ([3]). First we need the following

Definition 3 *A learning algorithm is said to be uniformly (or algorithmically) β -stable, if for any two training sets T_1, T_2 different at no more than one point,*

$$\forall \mathbf{x} \quad |f_{T_1}(\mathbf{x}) - f_{T_2}(\mathbf{x})| \leq \beta$$

The stability condition can be thought of as the Lipschitz property for maps from the set of training samples endowed with the Hamming distance into $L^\infty(V)$.

Theorem 4 (Bousquet, Elisseeff). *For a β -stable algorithm $T \rightarrow f_T$ we have:*

$$\forall \epsilon > 0 \quad \text{Prob}(|R_k(f_T) - R(f_T)| > \epsilon + \beta) \leq 2 \exp \left(- \frac{k \epsilon^2}{2(k\beta + (K + M))^2} \right)$$

The above theorem¹ together with the appropriate stability of graph regularization algorithm yields Theorem 1. We now proceed to show that regularization on graphs using the smoothness functional S is β -stable, with β as in Theorem 1.

Theorem 5 (Stability of Regularization on Graphs). *For data samples of size $k \geq 4$ with multiplicity of at most t , γ -regularization using the smoothness functional S is a $\left(\frac{3M\sqrt{tk}}{(k\gamma\lambda_1-t)^2} + \frac{4M}{k\gamma\lambda_1-t}\right)$ -stable algorithm, assuming that the denominator $k\gamma\lambda_1 - t$ is positive.*

Proof. Let H be the hyperplane orthogonal to the vector $\mathbf{1} = (1, \dots, 1)$. We will denote by P_H the operator corresponding to the orthogonal projection on H . Recall that the solution to the regularization problem is given by

$$(k\gamma S + I_k)\mathbf{f} = \tilde{\mathbf{y}} + \mu\mathbf{1}$$

where μ is chosen so that \mathbf{f} belongs to H . We order the graph so that the labeled points come first. Then the diagonal matrix I_k can be written as

$$I_k = \text{diag}(n_1, \dots, n_l, 0, \dots, 0)$$

where l is the number of distinct labeled vertices of the graph and $n_i \leq t$ is the multiplicity of the i th data point. The spectral radius of I_k is $\max(n_1, \dots, n_l)$ and is therefore no greater than t . Note that $l \leq k$.

On the other hand, the smallest eigenvalue of S restricted to H is λ_1 . Noticing that H is invariant under S and that for any vector \mathbf{v} , $\|P_H(\mathbf{v})\| \leq \|\mathbf{v}\|$, since P_H is an orthogonal projection operator, and using the triangle inequality, we immediately obtain that for any $\mathbf{f} \in H$

$$\|P_H(k\gamma S + I_k)\mathbf{f}\| \geq \|P_H k\gamma S\mathbf{f}\| - \|P_H I_k\mathbf{f}\| \geq (\lambda_1 \gamma k - t)\|\mathbf{f}\|$$

It follows that the spectral radius of the inverse operator $(P_H(k\gamma S + I_k))^{-1}$ does not exceed $\frac{1}{\lambda_1 \gamma k - t}$, when restricted to H (of course, the inverse is not even defined outside of H).

To demonstrate stability we need to show that the output of the algorithm does not change much when we change the input at exactly one data point. Suppose that \mathbf{y} , \mathbf{y}' are the data vectors different in at most one entry. We can assume that \mathbf{y}' contains a new point. The other case, when only the multiplicities differ, follows easily from the same considerations. Thus we write:

$$\mathbf{y} = \left(\sum_i y_{i1}, \sum_i y_{i2}, \dots, \sum_i y_{il}, 0, \dots, 0\right)$$

$$\mathbf{y}' = \left(\sum_i y_{i1}, \sum_i y_{i2}, \dots, \sum_i' y_{il}, y_{l+1}, 0, \dots, 0\right)$$

¹ Which is, actually, a special case of the original theorem, when the cost function is quadratic.

The sums are taken over all values of y corresponding to a node on a graph. The last sum \sum' contains one fewer term than the corresponding sum for \mathbf{y} .

Put \bar{y}, \bar{y}' to be the averages for \mathbf{y}, \mathbf{y}' respectively. We note that $|\bar{y} - \bar{y}'| \leq \frac{2M}{k}$ and that the entries of $\tilde{\mathbf{y}}, \tilde{\mathbf{y}}'$ differ by no more than that except for the last two entries, which differ by at most $2M + \frac{2M}{k}$. Of course, the last $n - l - 1$ entries of both vectors are equal to zero. Therefore

$$\|\tilde{\mathbf{y}} - \tilde{\mathbf{y}}'\| \leq \sqrt{2 \left(2M + \frac{2M}{k}\right)^2 + k \left(\frac{2M}{k}\right)^2} < 4M$$

assuming that $k \geq 4$.

The solutions to the regularization problem \mathbf{f}, \mathbf{f}' are given by the equations

$$\mathbf{f} = (P_H(\gamma k S + I_k))^{-1} \tilde{\mathbf{y}}$$

$$\mathbf{f}' = (P_H(\gamma k S + I'_k))^{-1} \tilde{\mathbf{y}}'$$

where I_k and I'_k are $n \times n$ diagonal matrices, $I_k = \text{diag}(n_1, n_2, \dots, n_l, 0, \dots, 0)$, $I'_k = \text{diag}(n_1, n_2, \dots, n_l - 1, 1, 0, \dots, 0)$ and the operators are restricted to the hyperplane H .

In order to ascertain stability, we need to estimate the maximum difference between the entries of \mathbf{f} and \mathbf{f}' , $\|\mathbf{f} - \mathbf{f}'\|_\infty$. We will use the fact that $\|\cdot\|_\infty \leq \|\cdot\|$.

Put $A = P_H(\gamma k S + I_k)$, $B = P_H(\gamma k S + I'_k)$ restricted to the hyperplane H . We have

$$\mathbf{f} - \mathbf{f}' = A^{-1} \tilde{\mathbf{y}} - B^{-1} \tilde{\mathbf{y}}' = A^{-1}(\tilde{\mathbf{y}} - \tilde{\mathbf{y}}') + A^{-1} \tilde{\mathbf{y}}' - B^{-1} \tilde{\mathbf{y}}'$$

Therefore

$$\|\mathbf{f} - \mathbf{f}'\|_\infty \leq \|\mathbf{f} - \mathbf{f}'\| \leq \|A^{-1}(\tilde{\mathbf{y}} - \tilde{\mathbf{y}}')\| + \|A^{-1} \tilde{\mathbf{y}}' - B^{-1} \tilde{\mathbf{y}}'\|$$

Since the spectral radius of A^{-1} and B^{-1} is at most $\frac{1}{k\gamma\lambda_1 - t}$ and $\|\tilde{\mathbf{y}} - \tilde{\mathbf{y}}'\| \leq 4M$,

$$\|A^{-1}(\tilde{\mathbf{y}} - \tilde{\mathbf{y}}')\| \leq \frac{4M}{k\gamma\lambda_1 - t}$$

On the other hand, it can be checked that $\|\tilde{\mathbf{y}}'\| \leq 2\sqrt{tk}M$. Indeed, it can be easily seen that the length is maximized, when the multiplicity of each point is exactly t . Noticing that the spectral radius of $P_H(I_k - I'_k)$ cannot exceed $\sqrt{2} < 1.5$, we obtain:

$$\begin{aligned} \|A^{-1} \tilde{\mathbf{y}}' - B^{-1} \tilde{\mathbf{y}}'\| &= \|B^{-1}(B - A)A^{-1} \tilde{\mathbf{y}}'\| = \|B^{-1}P_H(I_k - I'_k)A^{-1} \tilde{\mathbf{y}}'\| \leq \\ &\leq \frac{3M\sqrt{tk}}{(k\gamma\lambda_1 - t)^2} \end{aligned}$$

Putting it all together

$$\|\mathbf{f} - \mathbf{f}'\|_\infty \leq \frac{3M\sqrt{tk}}{(k\gamma\lambda_1 - t)^2} + \frac{4M}{k\gamma\lambda_1 - t}$$

Of course, we would typically expect $\frac{2M\sqrt{tk}}{(k\gamma\lambda_1-t)^2} \ll \frac{4M}{k\gamma\lambda_1-t}$.

However one issue still remains unresolved. Just how likely are we to have multiple points in a sample. Having high multiplicities is quite unlikely as long as $k \ll n$ and the distribution is reasonably close to the uniform.

We make a step in the direction with the following simple combinatorial estimate to show that for the uniform distribution on the graph, data samples, where point occur with high multiplicities (and, in fact, with any multiplicity greater than 1) are unlikely as long as k is relatively small compared to n .

It would be easy to give a similar estimate for a more general distribution, where probability of each point is bounded from below by, say, $\frac{\alpha}{n}$, $0 < \alpha \leq 1$.

Proposition 3. *Assuming the uniform distribution on the graph, the probability P of a sample that contains some data point with multiplicity more than t can be estimated as follows:*

$$P < \frac{2n}{(t+1)!} \left(\frac{k}{n}\right)^{t+1}$$

Proof. Let us first estimate the probability P_l that the l th point will occur more than t times, when choosing k points at random from a dataset of n points with replacement.

$$P_l = \sum_{i=t+1}^k \binom{k}{i} \frac{1}{n^i} \left(1 - \frac{1}{n}\right)^{k-i} < \sum_{i=t+1}^k \binom{k}{i} \frac{1}{n^i}$$

Writing out the binomial coefficients and using an estimate via the sum of a geometric progression yields:

$$\sum_{i=t+1}^k \binom{k}{i} \frac{1}{n^i} < \frac{1}{(t+1)!} \sum_{i=t+1}^k \left(\frac{k}{n}\right)^i = \frac{1}{(t+1)!} \left(\frac{k}{n}\right)^{t+1} \frac{1}{1 - \frac{k}{n}}$$

Assuming that $k \leq \frac{n}{2}$, we finally obtain

$$P_l < \frac{2}{(t+1)!} \left(\frac{k}{n}\right)^{t+1}$$

Applying the union bound, we see that the probability P of some point being chosen more than t times is bounded as follows:

$$P \leq \sum_{i=1}^n P_i < \frac{2n}{(t+1)!} \left(\frac{k}{n}\right)^{t+1}$$

By rewriting k in terms of the probability, we immediately obtain the following

Corollary 6 *With probability at least $1 - \epsilon$ the multiplicity of the sample does not exceed t , given that $k \leq \sqrt[t+1]{\epsilon \frac{(t+1)!}{2}} n^{t-\frac{1}{t+1}}$. In particular, the multiplicity of the sample is exactly 1 with probability at least $1 - \epsilon$, as long as $k \leq \sqrt{\epsilon n}$.*

4 Experiments and Discussion

An interesting aspect of the generalization bound derived in the previous section is that it depends on certain geometric aspects of the graph. The size of the graph seems relatively unimportant. For example consider the edge graph of a d -dimensional hypercube. Such a graph has $n = 2^d$ vertices. However, the spectral gap is always $\lambda_1 = 2$. Thus the generalization bound on such graphs is *independent* of the size n . For other kinds of graphs, it may be the case that λ_1 depends weakly on n . For such graphs, we may hope for good generalization from a small number of labeled examples relative to the size of the graph.

To evaluate the performance of our regularization algorithms and the insights from our theoretical analysis, we conducted a number of experiments. For example, our experimental results indicate that both Tikhonov and interpolated regularization schemes are generally competitive and often better than other semi-supervised algorithms. However, in this paper we do not discuss these performance comparisons. Instead, we focus on the performance of our algorithm and the usefulness of our bounds.

We present results on two data sets of different sizes.

4.1 Ionosphere Data Set

The Ionosphere data set has 351 examples of two classes in a 34 dimensional space. A graph is made by connecting nearby (6) points to each other. This graph therefore has 351 vertices. We computed the value of the spectral gap of this graph and the corresponding bound using different values of γ for different numbers of labelled points (see table 4). We also computed the training error (see table 2), the test error (see table 1), and the generalization gap (see table 3), to compare it with the value of the bound.

For $\gamma \geq 1$, the value of the bound is reasonable and the difference between the training and the test error is small, as can be seen in the last columns of these tables. However, both the training and the test error for $\gamma = 1$ were high. In regimes where training and test errors were smaller, we find that our bound becomes vacuous.

4.2 Mnist Data Set

We also tested the performance of the regularization algorithm on the MNIST data set. We used a training set with 11,800 examples corresponding to a two class problem with digits 8 and 9.

We computed the training and the test error as well as the bound for this two-class problem. We report the results for the digits 8 and 9, averaged over 10 random splits. Table 5 and table 6 show the error on the test and on the training set, respectively. The regularization algorithm achieves a very low error rate on this data set even with a small number of labelled points. The difference between the training and the test error is shown in table 7 and can be compared to the value of the bound in table 8.

#L	$\gamma=0.001$	$\gamma=0.01$	$\gamma=0.1$	$\gamma=1$
10	0.36	0.40	0.38	0.36
20	0.29	0.35	0.38	0.36
40	0.22	0.36	0.37	0.36
60	0.20	0.36	0.36	0.36
80	0.17	0.35	0.39	0.36
100	0.18	0.30	0.36	0.36
200	0.20	0.36	0.35	0.36
300	0.13	0.40	0.36	0.34

Table 1. Ionosphere data set. Classification error rates on the test set. #L is the number of labelled examples.

#L	$\gamma=0.001$	$\gamma=0.01$	$\gamma=0.1$	$\gamma=1$
10	0.00	0.09	0.26	0.30
20	0.01	0.22	0.29	0.33
40	0.01	0.25	0.31	0.35
60	0.08	0.28	0.36	0.34
80	0.09	0.30	0.35	0.36
100	0.10	0.31	0.36	0.37
200	0.14	0.35	0.36	0.36
300	0.15	0.35	0.36	0.36

Table 2. Ionosphere data set. Classification error rates on the training set. #L is the number of labelled examples.

#L	$\gamma=0.001$	$\gamma=0.01$	$\gamma=0.1$	$\gamma=1$
10	0.36	0.31	0.12	0.06
20	0.28	0.13	0.09	0.03
40	0.21	0.11	0.06	0.01
60	0.12	0.08	0.00	0.02
80	0.08	0.05	0.04	0.00
100	0.08	0.01	0.00	0.01
200	0.06	0.01	0.01	0.00
300	0.02	0.05	0.00	0.02

Table 3. Ionosphere data set. Difference between error rates on the test set and on the training set.

#L	$\gamma=0.001$	$\gamma=0.01$	$\gamma=0.1$	$\gamma=1$
10	173.59	32.87	2.92	1.16
20	1641.55	16.38	2.02	0.82
40	2138.57	9.73	1.40	0.58
60	469.07	7.44	1.14	0.47
80	251.67	6.22	0.98	0.41
100	173.02	5.43	0.87	0.36
200	72.72	3.64	0.61	0.26
300	48.97	2.90	0.50	0.21

Table 4. Ionosphere data set, $\lambda_1 = 34.9907$. Generalization bound for confidence $(1 - \delta)$, $\delta = 0.1$.

Here again, we observe that the value of the bound is reasonable for $\gamma = 0.1$ and $\gamma = 1$ but the test and training errors for these values of γ are rather high. Note, however, that with 2000 labelled points, the error rate for $\gamma = 0.1$ is very similar to the error rates achieved with smaller values of γ .

Interestingly, the regularization algorithm has very similar gaps between the training and the test error for these two data sets although the number of points in their graphs is very different (351 for the Ionosphere and 11,800 for the MNIST two-class problem). The value of the smallest non-zero eigenvalue for these two graphs is, however, similar. Therefore the similarity in the generalization gaps is consistent with our analysis.

5 Conclusions

In a number of different settings, the need arises to fill in the labels (values) of a partially labeled graph. We have provided a principled framework within which one can meaningfully formulate regularization for regression and classification on such graphs. Two different algorithms were then derived within this framework and have been shown to perform well on different data sets.

#L	$\gamma=0.001$	$\gamma=0.01$	$\gamma=0.1$	$\gamma=1$
20	0.04	0.03	0.45	0.50
40	0.02	0.03	0.42	0.40
100	0.02	0.03	0.37	0.40
200	0.02	0.02	0.28	0.41
400	0.02	0.02	0.09	0.46
800	0.02	0.02	0.11	0.44
2000	0.02	0.02	0.03	0.41

Table 5. Mnist data set, two-class classification problem for digits 8 and 9. Classification error rates on the test set.

#L	$\gamma=0.001$	$\gamma=0.01$	$\gamma=0.1$	$\gamma=1$
20	0.00	0.01	0.33	0.40
40	0.00	0.01	0.36	0.36
100	0.01	0.02	0.32	0.38
200	0.02	0.02	0.24	0.39
400	0.02	0.02	0.09	0.45
800	0.02	0.02	0.10	0.42
2000	0.02	0.02	0.03	0.40

Table 6. Mnist data set, two-class classification problem for digits 8 and 9. Classification error rates on the training set.

#L	$\gamma=0.001$	$\gamma=0.01$	$\gamma=0.1$	$\gamma=1$
20	0.04	0.02	0.12	0.10
40	0.02	0.02	0.06	0.04
100	0.01	0.01	0.05	0.02
200	0.00	0.00	0.04	0.02
400	0.00	0.00	0.00	0.01
800	0.00	0.00	0.01	0.02
2000	0.00	0.00	0.00	0.01

Table 7. Mnist data set, two-class classification problem for digits 8 and 9. Difference between error rates on the test set and the on the training set.

#L	$\gamma=0.001$	$\gamma=0.01$	$\gamma=0.1$	$\gamma=1$
20	1774.43	16.04	2.00	0.81
40	1928.94	9.55	1.39	0.57
100	166.74	5.34	0.87	0.36
200	70.69	3.58	0.61	0.26
400	37.13	2.44	0.43	0.18
800	21.60	1.69	0.30	0.13
2000	11.50	1.04	0.19	0.08

Table 8. Mnist data set, two-class classification problem for digits 8 and 9, $\lambda_1=35.5460$. Generalization bound for confidence $(1-\delta)$, $\delta=0.1$.

The regularization framework offers several advantages.

1. It eliminates the need for computing multiple eigenvectors or complicated graph invariants (min cut, max flow etc.). Unlike some previously proposed algorithms, we obtain a simple closed form solution for the optimal regressor. The problem is reduced to a single, usually sparse, linear system of equations whose solution can be computed efficiently. One of the algorithms proposed (interpolated regularization) is extremely simple with no free parameters.
2. We are able to bound the generalization error and relate it to properties of the underlying graph using arguments from algorithmic stability.
3. If the graph arises from the local connectivity of data obtained from sampling an underlying manifold, then the approach has natural connections to regularization on that manifold.

The experimental results presented here suggest that the approach has empirical promise. Our future plans include more extensive experimental comparisons and investigating potential applications to survey sampling and other areas.

Acknowledgements

We would like to thank Dengyong Zhou, Olivier Chapelle and Bernard Schoelkopf for numerous conversations and, in particular, for pointing out that Interpolated Regularization is the limit case of Tikhonov regularization, which motivated us to modify the Interpolated Regularization algorithm by introducing $f \perp \mathbf{1}$ condition.

References

1. M. Belkin, P. Niyogi, *Using Manifold Structure for Partially Labeled Classification* Advances in Neural Information Processing Systems 15, MIT Press, 2003,
2. A. Blum, S. Chawla, *Learning from Labeled and Unlabeled Data using Graph Min-cuts*, ICML, 2001,
3. Bousquet, O., A. Elisseeff, *Algorithmic Stability and Generalization Performance*. Advances in Neural Information Processing Systems 13, 196-202, MIT Press, 2001,
4. Chapelle, O., J. Weston and B. Scholkopf, *Cluster Kernels for Semi-Supervised Learning*, Advances in Neural Information Processing Systems 15. (Eds.) S. Becker, S. Thrun and K. Obermayer,
5. Fan R. K. Chung, *Spectral Graph Theory*, Regional Conference Series in Mathematics, number 92, 1997
6. L.P. Devroye, T. J. Wagner, *Distribution-free Performance Bounds for Potential Function Rules*, IEEE Trans. on Information Theory, 25(5): 202-207, 1979.
7. M. Fiedler, *Algebraic connectivity of graphs*, Czechoslovak Mathematical Journal, 23(98):298-305, 1973.
8. D. Harville, *Matrix Algebra From A Statistician's Perspective*, Springer, 1997.
9. T. Joachims, *Transductive Inference for Text Classification using Support Vector Machines*, Proceedings of ICML-99, pps 200-209, 1999.
10. I.R. Kondor, J. Lafferty, *Diffusion Kernels on Graphs and Other Discrete Input Spaces*, Proceedings of ICML, 2002.
11. K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, *Text Classification from Labeled in Unlabeled Data*, Machine Learning 39(2/3), 2000,
12. A. Smola and R. Kondor, *Kernels and Regularization on Graphs*, COLT/KW 2003,
13. Martin Szummer, Tommi Jaakkola, *Partially labeled classification with Markov random walks*, Neural Information Processing Systems (NIPS) 2001, vol 14.,
14. V. Vapnik, *Statistical Learning Theory*, Wiley, 1998,
15. D. Zhou, O. Bousquet, T.N. Lal, J. Weston and B. Schoelkopf, *Learning with Local and Global Consistency*, Max Planck Institute for Biological Cybernetics Technical Report, June 2003,
16. X. Zhu, J. Lafferty and Z. Ghahramani, *Semi-supervised learning using Gaussian fields and harmonic functions*, Machine Learning: Proceedings of the Twentieth International Conference, 2003.