

SEMI-SUPERVISED LEARNING ON GRAPHS
– A STATISTICAL APPROACH

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Ya Xu
May 2010

Abstract

Data on graphs are growing tremendously in size and prevalence these days; consider the World Wide Web graph or the Facebook social network. In semi-supervised learning on graphs, features observed at one node are used to estimate missing values at other nodes. Many prediction methods have been proposed in the machine learning community over the past few years. In this thesis we show that several such proposals are equivalent to kriging predictors based on a fixed covariance matrix driven by the link structure of the graph. We then propose a data-driven estimator of the correlation structure that exploits patterns among the observed response values. We also show how we can scale some of the algorithms to large graphs. Finally, we investigate the fundamental smoothness assumption underlying many prediction methods by exploring some normality properties arising from empirical data analysis.

Acknowledgments

Contents

Abstract	v
Acknowledgments	vi
1 Introduction	1
2 Background	3
2.1 Background on graphs	3
2.1.1 Graph structure	3
2.1.2 Graph random walk	4
2.1.3 Graph Laplacian	4
2.1.4 Structural properties of real graphs	7
2.2 Semi-supervised learning graph algorithms	8
2.2.1 Random walk smoothing	9
2.2.2 Tikhonov and interpolated Tikhonov smoothing	10
2.2.3 Undirected random walk smoothing	11
2.2.4 Hub and authority smoothing	12
2.2.5 Manifold smoothing	14
2.2.6 Spectral transformation of Δ	14
3 Semi-supervised learning as kriging	15
3.1 Introduction	15
3.2 Background on kriging	15
3.2.1 Kriging model	16
3.2.2 Covariance functions and estimation	20
3.3 Semi-supervised learning as kriging	25
3.3.1 Random walk smoothing	28
3.3.2 Tikhonov smoothing	30
3.3.3 Interpolated Tikhonov smoothing	31
3.3.4 Undirected random walk smoothing	33
3.3.5 Hub and authority smoothing	33

3.3.6	Manifold smoothing	34
3.3.7	Spectral transformation of Δ	35
4	Empirical stationary correlation	37
4.1	Introduction	37
4.2	A toy example	38
4.3	Stationary correlations	40
4.3.1	Covariance estimation through the variogram	41
4.3.2	Practical issues	42
4.3.3	Relation to Geostatistics	44
4.4	Examples	45
4.4.1	UK university web link dataset	46
4.4.2	WebKB dataset	48
4.5	Variations	50
4.6	Other related literature	52
4.7	Conclusion	53
5	Scale to large graphs	54
5.1	Introduction	54
5.2	Markov chain algorithms	56
5.2.1	Theoretical equivalence	56
5.2.2	Advantages and implications	62
5.2.3	Approximation accuracy	64
5.3	Scale empirical kriging	67
5.3.1	Lanczos method	67
5.3.2	Estimating low rank covariance	70
5.3.3	Practical implementations and PROPACK	71
5.4	Wikipedia dataset	71
5.5	Conclusion and discussion	73
6	Investigating the smoothness measures	75
6.1	Introduction	75
6.2	Motivation: Permutation experiment	76
6.3	Central limit theorems for smoothness measures	78
6.3.1	Preliminaries	80
6.3.2	Maximal degree conditions	81
6.3.3	Degree distribution conditions	84
6.4	Conclusion	88
A	Table of symbols and notation	89

B Proofs	91
B.1 Closest positive semi-definite matrix	91
B.2 Proof of Lemma 6.3.4, (6.7)	92
B.3 Proof of Theorem 6.3.5, (6.11)	93
B.4 Supporting lemmas for Section 6.3.3	94
B.5 Proof of Theorem 6.3.7, (b)	96

List of Tables

2.1	A few examples to show that large-scale, real-world graphs are usually very sparse.	7
3.1	Summary of the semi-supervised learning methods from Section 2.2 in the form of equation (3.11).	26
3.2	Summary of connections between some semi-supervised learning methods and kriging.	36
4.1	Parameters chosen for model (4.1) to obtain the random walk smoothing and the Tikhonov smoothing methods. Both models use the limit $\delta \rightarrow \infty$	40
4.2	The steps we use to estimate the covariance matrix $\Sigma = \sigma^2 V R V$ in model (4.1) via an empirical stationary correlation model.	43
4.3	The relative improvement over baseline when 50 of 107 ARE scores are held out. The baseline methods are simple regressions through the origin on $X = \sqrt{\pi}$ (random walk) and on $X = \mathbf{1}_n$ (Tikhonov).	49
4.4	The relative improvement over baseline when 100 out of 195 webpage labels are held out. The baseline AUC is 0.5.	51
5.1	Summary of Markov chain constructions for computing the predictions \widehat{Y}_i	62
5.2	Markov chain implementation on Wikipedia dataset.	72
5.3	PROPACK implementation of the empirical covariance method on Wikipedia dataset.	72
6.1	Summary of the three datasets used in permutation experiment	78

List of Figures

2.1	The degree distribution (in log-log scale) of the Epinions social network from Richardson et al. [2003]. It decays approximately as a power-law.	8
3.1	Parametric correlation functions (left) and their corresponding one-dimensional realization of Gaussian process (right). Notice that larger κ gives a smoother signal for both Matérn and powered exponential families.	24
4.1	Left: heatmap of signs of a realization of the Gaussian process on 50×50 grid. Right: prediction MSE when 90% (2250) of the nodes have missing Y_i .	39
4.2	Illustration of the empirical Tikhonov method with the UK university data. Left: scatter plot of the naive \hat{R}_{ij} values versus $\log(s_{ij} + 1)$ with the cubic spline smoothing curve (red). Right: final estimates $\hat{\Sigma}_{+ij}/\sigma^2$ versus $\log(s_{ij} + 1)$ with the same smoothing curve (red).	47
4.3	MSEs for the RAE scores at different holdout sizes. Left: the original random walk (red) compared with our empirical random walk (green). Right: the original Tikhonov (red) compared with our empirical Tikhonov (green). Baseline methods (black) are described in the text.	49
4.4	Classification error for webpage labels at different holdout sizes, measured with 1 minus the area under the ROC curve. Left: the original random walk (red) compared with our empirical random walk (green). Right: the original Tikhonov (red) compared with our empirical Tikhonov (green). The baseline method is random guessing.	51
5.1	The approximation error (5.18) as a function of the number of simulations (m). Three types of graphs are considered, together with three prediction algorithms. The dotted line represents a convergence rate of m^{-1} while lines parallel to it indicate a rate of $\mathcal{O}(m^{-1})$. Three λ values are used for the random walk and the Tikhonov methods. The corresponding curves are labeled with their average chain lengths.	66

5.2	(Inverse) rate of convergence of the Lanczos iteration bounds in (5.19) and (5.20).	69
5.3	Prediction performance on the Wikipedia dataset with 50% response values held out.	73
6.1	Permuting labels while fixing the graph structure.	77
6.2	Distributions of the smoothness scores of the randomly permuted labels, compared with that of the ground truth labels (red dots).	79

Chapter 1

Introduction

Data on graphs has long been with us, but the recent explosion of interest in social network data available on the Internet and gene interaction network data from Computational Biology has brought this sort of data to prominence. A typical problem is to predict the value of a feature at one or more nodes in the graph. That feature is assumed to have been measured on some, but not all nodes of the graph. For example, we might want to predict which web pages are spam, after a human expert has labeled a subset of them as spam or not. Similarly, we might want to know on which Facebook profile pages an ad would get a click, although that ad has only been shown on a subset of pages.

The underlying assumption in these prediction problems is that there is some correlation, usually positive, between the values at vertices that are close to each other in the graph. By making predictions that are smooth with respect to a notion of distance in the graph, one is able to define a local average prediction.

This problem is often called semi-supervised learning, because while the entire graph structure is available, the response values are only measured at some of the nodes. Though far from being a matured subject, semi-supervised learning on graphs has attracted much attention over the past few years, partly due to its relevance to problems in practice. Many papers have been written, and to this end, we begin in Chapter 2 with a review of a number of recently proposed methods. This lays a foundation for the entire thesis and hence is referred to frequently in the subsequent chapters. Chapter 2 also includes a brief introduction of graph terminology together with some graph properties that are useful for later discussions.

We find in Chapter 3 that many of these popular graph prediction methods can be unified under a kriging framework. Kriging is a technique that is widely used in Geostatistics to make predictions on random fields. Instead of establishing an optimization criterion, kriging takes a more statistical approach that assumes a Gaussian covariance model with unknown parameters. After an introduction to the kriging

model and its parameter estimation, we show in detail that each of the semi-supervised learning algorithms reviewed in Chapter 2 is equivalent to a kriging predictor based on a fixed covariance matrix driven by the link structure of the graph.

Inspired by the connections between kriging and the graph learning methods explored in Chapter 3, we propose in Chapter 4 our empirical stationary correlation kriging method, where instead of assuming a fixed covariance matrix we use a data-driven estimator of the correlation structure that exploits patterns among the observed response values. By incorporating even a small fraction of observed covariation into the predictions we are able to obtain much improved prediction on two graph datasets.

Chapter 5 follows the development in the earlier chapters and considers a more practical issue: implementing the prediction methods on large scale graphs. Large scale presents different challenges to the existing algorithms and to our empirical kriging. To this end, we devote two (parallel) sections to address the challenges separately. More concretely, we show that the existing algorithms have equivalent Markov chain formulations and hence propose to approximate their solutions using Markov chain simulations. On the other hand, we show that we can efficiently compute the predictions of the empirical covariance method by applying some well-developed tools from numerical analysis. Both proposals are implemented on a Wikipedia graph with about 2.4 million nodes.

Chapter 6 focuses on the smoothness assumption underlying the semi-supervised learning algorithms reviewed in Chapter 2. It is clear from their optimization criterion that these methods rely on smoothness measures to make predictions. We investigate two popular choices of smoothness measures based on the graph Laplacian. We show on three graph datasets the surprising result that these measures can sometimes consider random response values to be much smoother than the ground truth. By applying Stein's central limit theorem for dependent random variables, we are able to develop theoretical justifications for the empirical results, casting doubt on the utility of these measures.

Semi-supervised learning on graphs is a new exciting research area that potentially has important practical impact. A tremendous effort has been made to develop techniques and algorithms, mostly from the machine learning community. This thesis takes a more statistical approach. We show that, through theory and examples, we can offer valuable and helpful insight into the problem, and at the same time provide algorithms that can be successfully applied in practice.

Parts of Chapter 2 through 4 are from Xu et al. [2010]. Chapter 6 is primarily based on Xu [2010].

Chapter 2

Background

Before we can discuss learning on graphs, we need to first introduce the graph notation and some graph related definitions. To this end, Section 2.1 collects the basic information on the graph structure, graph random walk and graph Laplacian. We also think it is important to develop intuition about real-world graphs from the very beginning, and hence have devoted a small subsection to discuss a few common structural properties of real graphs. From there we move to review some recently proposed graph prediction algorithms in Section 2.2. Since semi-supervised learning on graphs has been an exciting and popular new research area, many papers have been written over the past few years. We do not intend to give a full literature review, and only selectively include a few algorithms that are representative. On the other hand, we take one step further to present these algorithms in a somewhat unified fashion. This is to prepare for the relevant discussions in later chapters.

2.1 Background on graphs

2.1.1 Graph structure

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with n vertices, where \mathcal{V} and \mathcal{E} denote the vertex set and the edge set. Unless specified, we consider the general case where \mathcal{G} is a weighted and directed graph. The graph \mathcal{G} is represented by an adjacency matrix W with entries $w_{ij} > 0$ if there is an edge from i to j , and $w_{ij} = 0$ otherwise. We impose $w_{ii} = 0$, so that if the graph contains loops, we do not count them. Node i has out-degree $w_{i+} = \sum_{j=1}^n w_{ij}$ and in-degree $w_{+i} = \sum_{j=1}^n w_{ji}$. The volume of the graph is $\text{vol}(\mathcal{G}) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}$. We assume no isolated nodes such that $w_{+i} > 0$ and $w_{i+} > 0$. In the special case of an unweighted graph, we have $w_{ij} \in \{0, 1\}$. For an undirected graph, the in-degrees and out-degrees are equivalent, and we further introduce $d_i (= w_{i+} = w_{+i})$ to represent both in order to emphasize such a property.

In practice, when the original data come in the form of a graph, the weight w_{ij} usually has a natural interpretation. It could be the number of hyperlinks from one web page to another, or a binary value indicating whether protein i interacts with protein j .

However, when the weights are not readily available from the data, there is usually a two-step process to construct them. First, a symmetric and non-negative function is chosen to quantify the affinity between a pair of nodes. For instance, if each node lives in the Euclidean space \mathbb{R}^d , a popular choice is to use the Gaussian density function $a(i, j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$, where $\mathbf{x}_i \in \mathbb{R}^d$ describes the location of node i . We then need to construct the weights w_{ij} based on the pairwise affinity $a(i, j)$. The ϵ -neighborhood approach sets $w_{ij} = a(i, j)$ if $a(i, j) > \epsilon$ and $w_{ij} = 0$ otherwise. On the other hand, the k -nearest neighbor approach takes $w_{ij} = a(i, j)$ if j is one of the k closest neighbors of i and $w_{ij} = 0$ otherwise. For various other graph construction examples, see Zhu [2005b]. Luxburg [2007] also gives rule-of-thumb recommendations in the context of spectral clustering. In both of their discussions, it is clear that graph construction is crucial, though little is known about its theoretical implications.

From here on, we will assume that the graph is already constructed and the weights are given.

2.1.2 Graph random walk

There is a natural random walk associated with \mathcal{G} in which the probability of transition from i to j is

$$P_{ij} = \frac{w_{ij}}{w_{i+}}.$$

That is, the random walk at node i randomly follows one of i 's out links according to their weights. Very often this walk is irreducible and aperiodic. If not, it may be reasonable to modify W , by for example adding a small probability of a transition uniformly distributed on all nodes. For example, such a modification is incorporated into the PageRank algorithm of Page et al. [1998] to yield an irreducible and aperiodic walk on web pages.

An irreducible and aperiodic walk has a unique stationary distribution, that we call $\boldsymbol{\pi}$, which places probability π_i on vertex i . If the graph is undirected, this stationary distribution has a closed-form expression $\pi_i = d_i/\text{vol}(\mathcal{G})$.

2.1.3 Graph Laplacian

Other than the adjacency matrix W , another representation of the graph \mathcal{G} is the graph Laplacian, which is also the main tool for many semi-supervised graph algorithms. Mathematicians have been studying the properties of Laplacian matrices for

many years in the field of spectral graph theory. See Cvetković et al. [1980] and Chung [1997] for example. In this section, we will only focus on the results that are relevant to our discussions in the thesis.

There are mainly three different definitions of graph Laplacian considered in the literature, all of which require an *undirected* graph \mathcal{G} . For our purpose here, we consider the following unnormalized Δ and normalized $\tilde{\Delta}$:

$$\Delta \equiv D - W, \tag{2.1}$$

$$\tilde{\Delta} \equiv D^{-1/2}(D - W)D^{-1/2} = I - D^{-1/2}WD^{-1/2}, \tag{2.2}$$

where $D = \text{diag}(d_1, \dots, d_n)$. The following two propositions summarize some important facts about the two graph Laplacians that will be useful later. The results are standard and can be found in Mohar [1997] and Luxburg [2007] for instance.

Proposition 2.1.1. *The matrix Δ defined in (2.1) has the following properties:*

(a) *For every vector $\mathbf{y} \in \mathbb{R}^n$, we have*

$$\mathbf{y}^T \Delta \mathbf{y} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} (y_i - y_j)^2.$$

(b) *Δ is symmetric and positive semidefinite.*

(c) *The smallest eigenvalue of Δ is 0, and its corresponding eigenvector is $\mathbf{1}/\sqrt{n}$.*

(d) *The multiplicity k of the eigenvalue 0 of Δ equals the number of connected components in the graph.*

Proof. (b) follows directly from (a). (c) is a trivial result of (a) and the following

$$\Delta \mathbf{1} = D\mathbf{1} - W\mathbf{1} = \mathbf{d} - \mathbf{d} = \mathbf{0}.$$

We refer the readers to Luxburg [2007] for the proof of (d) and only prove (a) below:

$$\begin{aligned}
\mathbf{y}^T \Delta \mathbf{y} &= \mathbf{y}^T D \mathbf{y} - \mathbf{y}^T W \mathbf{y} \\
&= \sum_i d_i y_i^2 - \sum_i \sum_j w_{ij} y_i y_j \\
&= \sum_i \sum_j w_{ij} y_i^2 - \sum_i \sum_j w_{ij} y_i y_j \\
&= \frac{1}{2} \left(\sum_i \sum_j w_{ij} y_i^2 + \sum_i \sum_j w_{ij} y_j^2 - 2 \sum_i \sum_j w_{ij} y_i y_j \right) \\
&= \frac{1}{2} \sum_i \sum_j w_{ij} (y_i - y_j)^2,
\end{aligned}$$

following from $d_i = \sum_j w_{ij}$ and by symmetry. \square

Proposition 2.1.2. *The matrix $\tilde{\Delta}$ defined in (2.2) has the following properties:*

(a) *For every vector $\mathbf{y} \in \mathbb{R}^n$, we have*

$$\mathbf{y}^T \tilde{\Delta} \mathbf{y} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \left(\frac{y_i}{\sqrt{d_i}} - \frac{y_j}{\sqrt{d_j}} \right)^2.$$

(b) *$\tilde{\Delta}$ is symmetric and positive semidefinite.*

(c) *The smallest eigenvalue of $\tilde{\Delta}$ is 0, and its corresponding eigenvector is $\sqrt{\mathbf{d}}/\text{vol}(G)$.*

(d) *The multiplicity k of the eigenvalue 0 of $\tilde{\Delta}$ equals the number of connected components in the graph.*

Proof. (b) follows directly from (a). (c) is a trivial result of (a) and the following

$$\tilde{\Delta} \sqrt{\mathbf{d}} = (D^{-1/2} \Delta D^{-1/2}) \sqrt{\mathbf{d}} = D^{-1/2} (\Delta \mathbf{1}) = \mathbf{0}.$$

Again we only prove (a) below and interested readers are referred to Luxburg [2007] for the proof of part (d).

$$\begin{aligned}
\mathbf{y}^T \tilde{\Delta} \mathbf{y} &= \mathbf{y}^T D^{-1/2} \Delta D^{-1/2} \mathbf{y} \\
&= (D^{-1/2} \mathbf{y})^T \Delta (D^{-1/2} \mathbf{y}) \\
&= \frac{1}{2} \sum_i \sum_j w_{ij} \left(\frac{y_i}{\sqrt{d_i}} - \frac{y_j}{\sqrt{d_j}} \right)^2,
\end{aligned}$$

where the last step follows by applying Proposition 2.1.1 with \mathbf{y} replaced by $D^{-1/2}\mathbf{y}$. \square

2.1.4 Structural properties of real graphs

Even though the focus of this thesis is not to study the link structure of graphs, it is important to include here some structural properties that are common to many real-world graphs. This is not only because real graphs are used in empirical studies throughout this thesis, but also because it helps provide insights and develop theoretical results that are rested upon reasonable/realistic assumptions, as we will see in later chapters.

One of the most essential characteristics is sparsity. The total number of edges in real graphs, particularly in large real graphs, usually grows linearly with n (instead of n^2 as in a fully connected graph). We list a few examples in Table 2.1. As we can see, in all four real graphs the average degrees are less than ten, even though there are thousands of nodes. The sparsity property makes it possible to scale many prediction algorithms to very large graphs as we will discuss in Chapter 5.

Graph	Size	Ave. deg.	Reference
WWW	325,729	4.5	[Albert et al., 1999]
Microsoft IM	1.8×10^8	7.2	[Leskovec and Horvitz, 2008]
Amazon co-purchasing	262,111	4.7	[Leskovec et al., 2007]
Epinions social network	75,879	6.7	[Richardson et al., 2003]

Table 2.1: A few examples to show that large-scale, real-world graphs are usually very sparse.

Another key property is about the degree distribution, which gives the probability that a randomly picked node has a certain degree. The classical Erdős-Rényi random graph model [Erdős and Rényi, 1959; Gilbert, 1959] starts with n nodes and connects every pair of nodes independently with probability p , producing degrees that follow a Poisson distribution. Recently, direct measurement of the degrees of many real graphs show that a Poisson distribution does not usually apply. Rather, the degrees tend to follow a power-law distribution, which means there are usually a small fraction of the nodes that have very large degrees. This is first suggested by Faloutsos et al. [1999] after studying three snapshots of the Internet between 1997 and 1998. Studies on a wide range of other types of real graphs demonstrate such a heavy tailed degree distribution as well [Albert et al., 1999; Jeong et al., 2000]. To further convince ourselves, in Figure 2.1 we plot the degree distribution for the Epinions social network

from Table 2.1. Other than the noisy right tail, we see that the plot is almost linear, indicating an approximately power-law distribution.

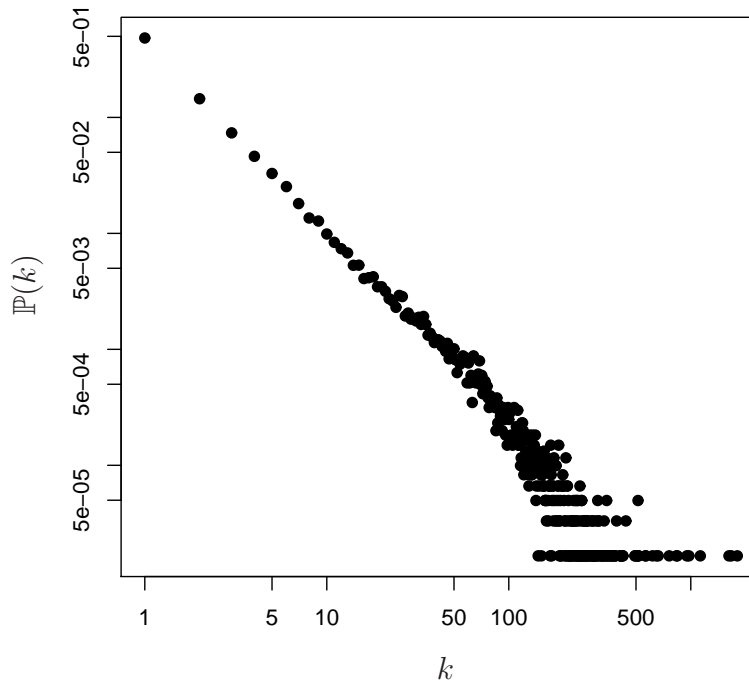


Figure 2.1: The degree distribution (in log-log scale) of the Epinions social network from Richardson et al. [2003]. It decays approximately as a power-law.

There are some other typical structural properties about real graphs. For instance, Watts and Strogatz [1998] noticed the “small-world” phenomenon: real graphs usually have diameters considerably smaller than regularly constructed graphs with the same number of nodes and edges. This is also related to the fact that real graphs tend to have a large number of clusters and triangles. Some other properties, such as the self-similarity, have been noted in the literature as well. See Leskovec and Faloutsos [2007] for example. We skip the details as these properties are not so relevant to this thesis.

2.2 Semi-supervised learning graph algorithms

We are now ready to review some graph prediction algorithms that have been proposed in the recent literature. We will start with the random walk strategy of Zhou

et al. [2005a], and then cover five other methods. Most of these examples are taken from a survey paper by Zhu [2005a].

We suppose that the response random variable at node i of the graph is Y_i . In many applications $Y_i \in \{-1, 1\}$ is binary, though we consider the general case of both discrete and continuous variables.

We partition the entire vector \mathbf{Y} as follows: we let $\mathbf{Y}^{(0)}$ denote the random variables that are observed and the unobserved part is denoted by $\mathbf{Y}^{(1)}$. Without loss of generality, we assume that the vectors are ordered such that $\mathbf{Y}^{(0)}$ comprises of the first $r > 1$ elements of \mathbf{Y} , i.e. $\mathbf{Y}^{(0)} = (Y_1, \dots, Y_r)^T$.

The underlying assumption in these prediction algorithms is that Y_i and Y_j are close in value if nodes i and j are close to each other in the graph. In other words, the vector $\mathbf{Y} \in \mathbb{R}^n$ is smooth with respect to a notion of distance in the graph. By making predictions that are smooth, one is able to define a local average prediction. We will notice that these algorithms all have similar two-component optimization framework. The first component regularizes the solution toward a smoother answer while the second component penalizes lack of fit.

2.2.1 Random walk smoothing

Zhou et al. [2005a] start by constructing a variation functional for vectors $\mathbf{Z} \in \mathbb{R}^n$ defined on the nodes of \mathcal{G} :

$$\Omega(\mathbf{Z}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \pi_i P_{ij} \left(\frac{Z_i}{\sqrt{\pi_i}} - \frac{Z_j}{\sqrt{\pi_j}} \right)^2. \quad (2.3)$$

This variation penalizes vectors \mathbf{Z} that differ too much over similar nodes and hence we refer to it as a smoothness measure. Notice that it contains a scaling of Z_i by $\sqrt{\pi_i}$. One intuitive reason for such a scaling is that a small number of nodes with a large π_i could reasonably have more extreme values of Z_i while the usually much greater number of nodes with small π_i should not ordinarily be allowed to have very large Z_i , and hence should be regularized more strongly. Mathematically, the divisors $\sqrt{\pi_i}$ originate in spectral clustering and graph partitioning algorithms.

For later use, we can write this smoothness measure in matrix notation using the graph Laplacian.

Lemma 2.2.1. *Let $w'_{ij} = (\pi_i P_{ij} + \pi_j P_{ji})/2$. Then the smoothness measure in (2.3) has a matrix form*

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \pi_i P_{ij} \left(\frac{Z_i}{\sqrt{\pi_i}} - \frac{Z_j}{\sqrt{\pi_j}} \right)^2 = \mathbf{Z}^T \tilde{\Delta}' \mathbf{Z},$$

where $\tilde{\Delta}'$ is defined as the normalized graph Laplacian in (2.2) with weights w'_{ij} instead of w_{ij} .

Proof. First note that $w'_{ij} = w'_{ji}$, and by symmetry

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \pi_i P_{ij} \left(\frac{Z_i}{\sqrt{\pi_i}} - \frac{Z_j}{\sqrt{\pi_j}} \right)^2 = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w'_{ij} \left(\frac{Z_i}{\sqrt{\pi_i}} - \frac{Z_j}{\sqrt{\pi_j}} \right)^2.$$

Further, the new degrees are

$$d'_i \equiv \sum_j w'_{ij} = \frac{1}{2} \pi_i \sum_j P_{ij} + \frac{1}{2} \sum_j \pi_j P_{ji} = \pi_i, \quad (2.4)$$

where the last step follows because $\sum_j \pi_j P_{ji} = \pi_i$ by definition of stationary probability. The result now follows by Proposition 2.1.2. \square

The prediction \mathbf{Z} should have a small value of $\Omega(\mathbf{Z})$. But it should also remain close to the observed values. To this end, Zhou et al. [2005a] make a vector \mathbf{Y}^* where $Y_i^* = y_i$ when y_i is observed and $Y_i^* = \mu_i$ when y_i is not observed, where μ_i is a reasonable guess for Y_i . Then the predictions are given by

$$\hat{\mathbf{Y}} = \operatorname{argmin}_{\mathbf{Z} \in \mathbb{R}^n} \mathbf{Z}^T \tilde{\Delta}' \mathbf{Z} + \lambda \|\mathbf{Z} - \mathbf{Y}^*\|^2, \quad (2.5)$$

where $\lambda > 0$ is a parameter governing the trade off between fit and smoothness.

2.2.2 Tikhonov and interpolated Tikhonov smoothing

Belkin et al. [2004] consider undirected graphs with the symmetric edge weights w_{ij} . Their Tikhonov regularization algorithm uses a different smoothness measure

$$\Omega(\mathbf{Z}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} (Z_i - Z_j)^2, \quad (2.6)$$

which is equivalent to $\mathbf{Z}^T \Delta \mathbf{Z}$ by Proposition 2.1.1. Their prediction criterion is thus proportional to

$$\min_{\mathbf{Z} \in \mathbb{R}^n} \mathbf{Z}^T \Delta \mathbf{Z} + \lambda_0 \|\mathbf{Z}^{(0)} - \mathbf{Y}^{(0)}\|^2. \quad (2.7)$$

They also have an option to use the side constraint $\frac{1}{n} \sum_{i=1}^n Z_i = \frac{1}{r} \sum_{i=1}^r Y_i^{(0)}$. That constraint forces the mean prediction to equal the mean observation, and is necessary

for the generalization bound they obtained. We do not include this condition, because the squared error norm on $\mathbf{Z}^{(0)} - \mathbf{Y}^{(0)}$ already forces $\mathbf{Z}^{(0)}$ to be close to $\mathbf{Y}^{(0)}$.

Although the original algorithm is proposed for undirected graphs, it is also well defined when the graph is directed. This is because a directed graph is equivalent to an undirected graph with symmetrized weights $(w_{ij} + w_{ji})/2$ in terms of (2.6).

There are two key differences between this method and the random walk smoothing described in Section 2.2.1. First, the smoothness measure uses weights w_{ij} instead of $\pi_i P_{ij}$ and does not contain a scaling factor. As a result, the corresponding graph Laplacian is unnormalized and constructed based on w_{ij} other than w'_{ij} . Second, this model only penalizes deviations from observed variables $\mathbf{Y}^{(0)}$, while random walk smoothing penalizes departure from the entire initial vector \mathbf{Y}^* . Therefore, this method avoids plugging in a guess for the unobserved $\mathbf{Y}^{(1)}$, and is thus more typical of statistical practice.

Belkin et al. [2004] also propose an interpolating algorithm that leaves all the known values unchanged in the prediction. That is $\hat{\mathbf{Y}}^{(0)} = \mathbf{Y}^{(0)}$. Furthermore, they consider the generalization that replaces Δ by Δ^p for a positive integer power p . They also consider a generalization in which there could be more than one measurement made on the response variable at some of the nodes. We do not consider cases more general than 0 or 1 observed response values per node.

2.2.3 Undirected random walk smoothing

Zhou et al. [2004] present an undirected graph algorithm that is a predecessor to the random walk smoothing of Zhou et al. [2005a].

They minimize

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \left(\frac{Z_i}{\sqrt{d_i}} - \frac{Z_j}{\sqrt{d_j}} \right)^2 + \lambda \|\mathbf{Z} - \mathbf{Y}^*\|^2 \quad (2.8)$$

which is the random walk smoothing criterion (2.5) after replacing $\pi_i P_{ij}$ by the weight w_{ij} and the stationary probability π_i by the degree d_i . Recall that for an irreducible aperiodic random walk on an undirected graph with transitions $P_{ij} = w_{ij}/d_i$, the stationary distribution has $\pi_i = d_i/\text{vol}(\mathcal{G})$. So $\pi_i P_{ij}$ becomes proportional to w_{ij} : $\pi_i P_{ij} = (d_i/\text{vol}(\mathcal{G}))(w_{ij}/d_i) = w_{ij}/\text{vol}(\mathcal{G})$. As a result, (2.5) is equivalent to (2.8) if the graph \mathcal{G} is undirected. In matrix notation, minimizing (2.8) becomes

$$\min_{\mathbf{Z} \in \mathbb{R}^n} \mathbf{Z}^T \tilde{\Delta} \mathbf{Z} + \lambda \|\mathbf{Z} - \mathbf{Y}^*\|^2$$

by Proposition 2.1.2, where $\tilde{\Delta}$ is the normalized graph Laplacian.

2.2.4 Hub and authority smoothing

Zhou et al. [2005b] proposes another random walk based strategy on directed graphs that is motivated by the hub and authority web model introduced by Kleinberg [1999]. For Zhou et al. [2005b], any node with an out-link is a hub and any node with an in-link is an authority. A node can be both a hub and an authority. They use two random walks. Their hub walk transitions between hubs that link to a common authority and their authority walk transitions between authorities linked by a common hub.

The hubs define a walk on the authorities as follows. From authority i we pick a linking hub h with probability w_{hi}/w_{+i} and from there pick an authority j with probability w_{hj}/w_{h+} . The resulting transition probability from i to j is

$$P_{ij}^{(A)} = \sum_h \frac{w_{hi}}{w_{+i}} \cdot \frac{w_{hj}}{w_{h+}}$$

where the sum is over hubs h . Analogous hub transition probabilities are

$$P_{ij}^{(H)} = \sum_a \frac{w_{ia}}{w_{i+}} \cdot \frac{w_{ja}}{w_{+a}},$$

summing over authorities a .

The stationary distributions of these two walks are given in the following lemma.

Lemma 2.2.2. *The stationary distributions of the hub and authority walks have closed forms*

$$\pi_i^{(H)} = w_{i+}/\text{vol}(\mathcal{G}), \quad \text{and} \quad \pi_i^{(A)} = w_{+i}/\text{vol}(\mathcal{G}). \quad (2.9)$$

Proof. It suffices to show that these probabilities in (2.9) satisfy the following equalities

$$\pi_i^{(H)} = \sum_j \pi_j^{(H)} P_{ji}^{(H)}, \quad \text{and} \quad \pi_i^{(A)} = \sum_j \pi_j^{(A)} P_{ji}^{(A)}.$$

For the hub-walk,

$$\begin{aligned}
\sum_j \pi_j^{(H)} P_{ji}^{(H)} &= \sum_j \frac{w_{j+}}{\text{vol}(\mathcal{G})} \sum_a \frac{w_{ja}}{w_{j+}} \cdot \frac{w_{ia}}{w_{+a}} \\
&= \sum_a \sum_j \frac{w_{j+}}{\text{vol}(\mathcal{G})} \cdot \frac{w_{ja}}{w_{j+}} \cdot \frac{w_{ia}}{w_{+a}} \\
&= \frac{1}{\text{vol}(\mathcal{G})} \sum_a \frac{w_{ia}}{w_{+a}} \sum_j w_{ja} \\
&= \frac{1}{\text{vol}(\mathcal{G})} \sum_a w_{ia} \\
&= \pi_i^{(H)}.
\end{aligned}$$

We can show the same for the authority-walk as follows.

$$\begin{aligned}
\sum_j \pi_j^{(A)} P_{ji}^{(A)} &= \sum_j \frac{w_{+j}}{\text{vol}(\mathcal{G})} \sum_h \frac{w_{hj}}{w_{+j}} \cdot \frac{w_{hi}}{w_{h+}} \\
&= \sum_h \sum_j \frac{w_{+j}}{\text{vol}(\mathcal{G})} \cdot \frac{w_{hj}}{w_{+j}} \cdot \frac{w_{hi}}{w_{h+}} \\
&= \frac{1}{\text{vol}(\mathcal{G})} \sum_h \frac{w_{hi}}{w_{h+}} \sum_j w_{hj} \\
&= \frac{1}{\text{vol}(\mathcal{G})} \sum_h w_{hi} \\
&= \pi_i^{(A)}.
\end{aligned}$$

□

These formulas in (2.9) give appropriate zeros for nodes i that are not hubs or authorities respectively.

Similar as in Lemma 2.2.1, we let $\tilde{\Delta}'_H$ be the normalized Laplacian defined with new weights $w'_{ij}{}^{(H)} = (\pi_i^{(H)} P_{ij}^{(H)} + \pi_j^{(H)} P_{ji}^{(H)})/2$, while $\tilde{\Delta}'_A$ be the normalized Laplacian with weights $w'_{ij}{}^{(A)} = (\pi_i^{(A)} P_{ij}^{(A)} + \pi_j^{(A)} P_{ji}^{(A)})/2$.

The optimization criterion used in the hub and authority regularization of Zhou et al. [2005b] can be written as follows:

$$\min_{\mathbf{Z} \in \mathbb{R}^n} \gamma \mathbf{Z}^T \tilde{\Delta}'_H \mathbf{Z} + (1 - \gamma) \mathbf{Z}^T \tilde{\Delta}'_A \mathbf{Z} + \lambda \|\mathbf{Z} - \mathbf{Y}^*\|^2, \quad (2.10)$$

for some $\gamma \in [0, 1]$. The choice of γ allows the user to weigh the relative importance

of in-links and out-links.

2.2.5 Manifold smoothing

The manifold regularization framework introduced by Belkin et al. [2006] considers undirected graphs. They predict the responses \mathbf{Y} by

$$\hat{\mathbf{Y}} = \underset{\mathbf{Z} \in \mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{Z}\|_{\mathcal{K}}^2 + \gamma \mathbf{Z}^T \Delta \mathbf{Z} + \lambda_0 \|\mathbf{Z}^{(0)} - \mathbf{Y}^{(0)}\|^2, \quad (2.11)$$

where \mathcal{K} is a Mercer kernel [Cristianini and Shawe-Taylor, 2000, Chapter 3], Δ is the graph Laplacian (2.1) and $\gamma > 0$. The term $\|\mathbf{Z}\|_{\mathcal{K}}^2$ controls the smoothness of the predictions in the *ambient* space, while $\mathbf{Z}^T \Delta \mathbf{Z}$ controls the smoothness with respect to the graph. In the special case where \mathcal{K} is a linear kernel, $\|\mathbf{Z}\|_{\mathcal{K}}^2 = \mathbf{Z}^T K \mathbf{Z}$ for a positive semidefinite matrix $K \in \mathbb{R}^{n \times n}$.

2.2.6 Spectral transformation of Δ

A few papers [Kondor and Lafferty, 2002; Smola and Kondor, 2003; Zhu et al., 2003] use the following smoothness measure

$$\Omega(\mathbf{Z}) = \mathbf{Z}^T L \mathbf{Z},$$

where L is the smoothing matrix constructed based on a spectral transformation of the graph Laplacian Δ . They take

$$L = \sum_{i=1}^n f(\tau_i) \mathbf{u}_i \mathbf{u}_i^T, \quad (2.12)$$

where $\tau_i \geq 0$ and \mathbf{u}_i are eigenvalues and eigenvectors of Δ , and $f(\cdot)$ is a non-negative increasing function, such as $f(x) = e^{\alpha^2 x/2}$.

Chapter 3

Semi-supervised learning as kriging

3.1 Introduction

In Chapter 2 we reviewed several semi-supervised learning graph algorithms from the recent literature. These algorithms all make predictions by utilizing an optimization framework. In Section 3.2 of this chapter, we present kriging, a Gaussian covariance model that is widely used in Geostatistics to predict the value of random fields. The two approaches to prediction may seem very different at first, particularly that graphs are far from being like \mathbb{R}^d where kriging is usually applied. However, we show that these recently developed methods for data on graphs can be expressed in terms of kriging. This connection is explored in detail in Section 3.3. In particular, we demonstrate that each of the graph algorithms discussed in Section 2.2 can be obtained by making strategic choices for the parameters of the underlying kriging model. Moreover, the kriging models which yield these smoothing algorithms have covariance assumptions driven by the geometry of the graph. This observation inspires our work in Chapter 4 of the thesis.

3.2 Background on kriging

This section provides the necessary background on kriging for understanding the subsequent sections and chapters. The materials can now be found in many good textbooks, e.g. Cressie [1993]; Diggle and Ribeiro [2006]; Stein [1999]. We first present the kriging model in Section 3.2.1 with notation compatible with the earlier chapters. Two slight variations of the model are also discussed. In the first case, we remove the Gaussian assumption, and show that the kriging predictor is still optimal in certain sense. Secondly, a Bayesian approach is taken where a prior distribution is used to quantify the prior belief of the uncertainty of the predictors. This variation

directly relates to the graph learning algorithms as we will see in Section 3.3.

The covariance matrix plays the most crucial role in the kriging model. To this end, we devote Section 3.2.2 to discuss covariance modeling and estimation techniques. Both the parametric and the nonparametric approaches are considered. We find these discussions useful particularly for Chapter 4.

3.2.1 Kriging model

Kriging is named for the mining engineer Krige, whose paper Krige [1951] introduced the method. For background on kriging see Stein [1999] or Cressie [1993]. Here we present the method and introduce the notation we need later.

The kriging model works as follows. The data $\mathbf{Y} \in \mathbb{R}^n$ are written as

$$\mathbf{Y} = \mathbf{Z} + \boldsymbol{\varepsilon}, \quad (3.1)$$

where $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the noise and $\mathbf{Z} \in \mathbb{R}^n$ is the signal vector where

$$\mathbf{Z} = X\boldsymbol{\beta} + \mathbf{S}.$$

Here $X \in \mathbb{R}^{n \times k}$ is a matrix of known predictors and $\boldsymbol{\beta} \in \mathbb{R}^k$ is a vector of coefficients. The structured part of the signal is $\mathbf{S} \sim \mathcal{N}(0, \Sigma)$ and it is the correlations within Σ that capture how neighbors in the graph are likely to be similar. Finally, $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \Gamma)$ is measurement noise independent of \mathbf{S} . The noise covariance Γ is diagonal.

In this formulation, the values \mathbf{Y} that we have observed are noisy measurements of some underlying quantity \mathbf{Z} that we wish we had observed. We seek to recover \mathbf{Z} from measurements \mathbf{Y} .

Some of the Y_i are observed and some are not. None of the Z_i are observed. Following our notation in Chapter 2, we let $\mathbf{Y}^{(0)}$ denote the random variables that are observed, and $\mathbf{y}^{(0)}$ be the values we saw for them. The unobserved part of \mathbf{Y} is denoted by $\mathbf{Y}^{(1)}$. The kriging predictor is the following natural choice that minimizes mean squared error (MSE)

$$\hat{\mathbf{Z}} = \mathbb{E}(\mathbf{Z} \mid \mathbf{Y}^{(0)} = \mathbf{y}^{(0)}). \quad (3.2)$$

We now give the explicit expression for predictor (3.2) since the conditional distribution of \mathbf{Z} given $\mathbf{Y}^{(0)}$ is known. Without loss of generality, suppose that the vectors are ordered with observed random variables before unobserved ones. We partition Σ as follows

$$\Sigma = \begin{pmatrix} \Sigma_{00} & \Sigma_{01} \\ \Sigma_{10} & \Sigma_{11} \end{pmatrix} = (\Sigma_{\bullet 0} \quad \Sigma_{\bullet 1}),$$

so that, for example, $\Sigma_{00} = \text{cov}(\mathbf{Z}^{(0)}, \mathbf{Z}^{(0)})$ and $\Sigma_{\bullet 0} = \text{cov}(\mathbf{Z}, \mathbf{Z}^{(0)})$. The matrices Σ

and Γ are partitioned the same way.

The joint distribution of \mathbf{Z} and $\mathbf{Y}^{(0)}$ is

$$\begin{pmatrix} \mathbf{Z} \\ \mathbf{Y}^{(0)} \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} X\boldsymbol{\beta} \\ X^{(0)}\boldsymbol{\beta} \end{pmatrix}, \begin{pmatrix} \Sigma & \Sigma_{\bullet 0} \\ \Sigma_{0\bullet} & \Sigma_{00} + \Gamma_{00} \end{pmatrix} \right),$$

where $X^{(0)}$ contains the rows of X corresponding to $\mathbf{Y}^{(0)}$. Therefore we can write the kriging predictor (3.2) explicitly as

$$\widehat{\mathbf{Z}} = \Sigma_{\bullet 0}(\Sigma_{00} + \Gamma_{00})^{-1}(\mathbf{y}^{(0)} - X^{(0)}\boldsymbol{\beta}) + X\boldsymbol{\beta}. \quad (3.3)$$

In the special case where the whole vector $\mathbf{Y} = \mathbf{y}$ is observed, the kriging predictor is

$$\widehat{\mathbf{Z}} = \Sigma(\Sigma + \Gamma)^{-1}(\mathbf{y} - X\boldsymbol{\beta}) + X\boldsymbol{\beta}. \quad (3.4)$$

Notice that the predictions do not necessarily interpolate the known values. That is $\widehat{\mathbf{Z}}^{(0)}$ need not equal $\mathbf{Y}^{(0)}$. Instead some smoothing takes place. The predictions can be forced closer to the data by making Γ_{00} smaller. The kriging approach also gives expressions for the variance of the prediction errors:

$$\text{var}(\mathbf{Z} \mid \mathbf{Y}^{(0)} = \mathbf{y}^{(0)}) = \Sigma - \Sigma_{\bullet 0}(\Sigma_{00} + \Gamma_{00})^{-1}\Sigma_{0\bullet}.$$

We delay the discussions on parameter estimation to Section 3.2.2.

BLUP

We have presented the kriging method under a Gaussian framework, where estimators (3.3) and (3.4) are the conditional expectations and hence are the best predictors in terms of minimizing MSE. However, even without the Gaussian assumption, estimator (3.3) and hence also (3.4) is the best linear unbiased predictor (BLUP) of \mathbf{Z} , because it minimizes the MSE among all linear unbiased predictors. This is a standard result and can be found in, for example, Stein [1999]. We prove it briefly below. For more background on the BLUP, see Robinson [1991].

Following the notation in Chapter 2, let $\mathbf{Y}^{(0)} = (Y_1, Y_2, \dots, Y_r)^T$. Assume that we want to predict Z_i , for $i = 1, 2, \dots, n$, using a linear predictor. That is, the predictor should be of the following form

$$\widehat{Z}_i = \theta_0 + \sum_{j=1}^r \theta_j Y_j = \theta_0 + \boldsymbol{\theta}^T \mathbf{Y}^{(0)}, \quad (3.5)$$

where θ_j 's are scalar coefficients and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_r)^T$. The MSE of this predictor

can be decomposed into squared bias and variance:

$$\begin{aligned} \text{MSE}(\widehat{Z}_i; \boldsymbol{\theta}) &= \mathbb{E}(Z_i - \widehat{Z}_i)^2 \\ &= (X_i^T \boldsymbol{\beta} - \theta_0 - \boldsymbol{\theta}^T X^{(0)} \boldsymbol{\beta})^2 \\ &\quad + \text{var}(Z_i) - 2\boldsymbol{\theta}^T \text{cov}(Z_i, \mathbf{Y}^{(0)}) + \boldsymbol{\theta}^T \text{var}(\mathbf{Y}^{(0)}) \boldsymbol{\theta}, \end{aligned}$$

where X_i is the i th row of the predictor matrix X .

By choosing $\theta_0 = X_i^T \boldsymbol{\beta} - \boldsymbol{\theta}^T X^{(0)} \boldsymbol{\beta}$, the bias term vanishes and we have

$$\text{MSE}(\widehat{Z}_i; \boldsymbol{\theta}) = \Sigma_{ii} - 2\boldsymbol{\theta}^T \Sigma_{0i} + \boldsymbol{\theta}^T (\Sigma_{00} + \Gamma_{00}) \boldsymbol{\theta}, \quad (3.6)$$

which is quadratic in $\boldsymbol{\theta}$. Therefore we obtain $\boldsymbol{\theta}$ that minimizes (3.6) to be

$$\boldsymbol{\theta}^* = (\Sigma_{00} + \Gamma_{00})^{-1} \Sigma_{0i},$$

and hence the corresponding θ_0^*

$$\theta_0^* = X_i^T \boldsymbol{\beta} - \Sigma_{i0} (\Sigma_{00} + \Gamma_{00})^{-1} X^{(0)} \boldsymbol{\beta}.$$

In other words, in view of (3.5), the linear predictor of Z_i that minimizes the MSE is

$$X_i^T \boldsymbol{\beta} + \Sigma_{i0} (\Sigma_{00} + \Gamma_{00})^{-1} (\mathbf{y}^{(0)} - X^{(0)} \boldsymbol{\beta}),$$

which is exactly the estimator in (3.3). We have so far proved that (3.3) is the best linear predictor (BLP). To show that this estimator is BLUP, it is now only left to show that it is unbiased, which clearly follows since $\mathbb{E}(Z_i) = \mathbb{E}(Y_i) = X_i^T \boldsymbol{\beta}$. Similarly, using the full \mathbf{Y} instead of $\mathbf{Y}^{(0)}$, it is readily shown that the estimator (3.4) is also BLUP.

A Bayesian prior on $\boldsymbol{\beta}$

O'Hagan and Kingman [1978] and Koehler [1990] have looked at a slight variation of the kriging model discussed so far, where the coefficient $\boldsymbol{\beta}$ is random with a prior distribution. The effect of the prior distribution is to quantify the prior belief of the uncertainty of the predictors X . For our purpose here, we restrict our attention to the case where the prior distribution of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{b}, \delta I),$$

independent of both \mathbf{S} and $\boldsymbol{\varepsilon}$. As a result, the vector of observations \mathbf{Y} follows

$$\mathbf{Y} \sim \mathcal{N}(X\mathbf{b}, \delta XX^T + \Sigma + \Gamma),$$

with $\delta XX^T + \Sigma$ being the new variance of the signal \mathbf{Z} . It is easy to see that the kriging predictors (3.3) and (3.4) are changed accordingly, to be

$$\widehat{\mathbf{Z}}_\delta = (\delta XX^{(0)T} + \Sigma_{\bullet,0}) (\delta X^{(0)} X^{(0)T} + \Sigma_{00} + \Gamma_{00})^{-1} (\mathbf{y}^{(0)} - X^{(0)}\mathbf{b}) + X\mathbf{b}, \quad (3.3')$$

$$\widehat{\mathbf{Z}}_\delta = (\delta XX^T + \Sigma) (\delta XX^T + \Sigma + \Gamma)^{-1} (\mathbf{y} - X\mathbf{b}) + X\mathbf{b}. \quad (3.4')$$

It is not surprising that (3.4) and (3.4') lead to different solutions in general. However, the following lemma states that if $\boldsymbol{\beta}$ has an improper prior, then (3.4') yields a predictor that is identical to (3.4) with $\boldsymbol{\beta}$ estimated with the generalized least square estimator. The same holds for (3.3) and (3.3').

Lemma 3.2.1. *Let $\hat{\boldsymbol{\beta}}^{GLS}$ be the generalized least square estimator*

$$\hat{\boldsymbol{\beta}}^{GLS} = (X^T(\Sigma + \Gamma)^{-1}X)^{-1} X^T(\Sigma + \Gamma)^{-1}\mathbf{y}. \quad (3.7)$$

Then in (3.4')

$$\lim_{\delta \rightarrow \infty} \widehat{\mathbf{Z}}_\delta = \Sigma (\Sigma + \Gamma)^{-1} (\mathbf{y} - X\hat{\boldsymbol{\beta}}^{GLS}) + X\hat{\boldsymbol{\beta}}^{GLS},$$

which is the kriging predictor (3.4) with $\boldsymbol{\beta}$ estimated using $\hat{\boldsymbol{\beta}}^{GLS}$. The corresponding connection holds between (3.3) and (3.3') as well.

Proof. We only prove for the pair (3.4) and (3.4') here due to notation simplicity.

We first find the posterior mean $\hat{\boldsymbol{\beta}}_\delta^{POST} \equiv \mathbb{E}(\boldsymbol{\beta} \mid \mathbf{Y} = \mathbf{y})$. The joint distribution of $\boldsymbol{\beta}$ and \mathbf{Y} is

$$\begin{pmatrix} \boldsymbol{\beta} \\ \mathbf{Y} \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} \mathbf{b} \\ X\mathbf{b} \end{pmatrix}, \begin{pmatrix} \delta I & \delta X^T \\ \delta X & \delta XX^T + \Sigma + \Gamma \end{pmatrix} \right).$$

Therefore

$$\hat{\boldsymbol{\beta}}_\delta^{POST} = \delta X^T (\delta XX^T + \Sigma + \Gamma)^{-1} (\mathbf{y} - X\mathbf{b}) + \mathbf{b}. \quad (3.8)$$

We are now ready to write (3.4') in terms of $\hat{\boldsymbol{\beta}}_\delta^{POST}$:

$$\begin{aligned}
\hat{\mathbf{Z}}_\delta &= X\hat{\boldsymbol{\beta}}_\delta^{POST} + \Sigma(\delta XX^T + \Sigma + \Gamma)^{-1}(\mathbf{y} - X\mathbf{b}) \\
&= X\hat{\boldsymbol{\beta}}_\delta^{POST} + \Sigma(\Sigma + \Gamma)^{-1}(\Sigma + \Gamma)(\delta XX^T + \Sigma + \Gamma)^{-1}(\mathbf{y} - X\mathbf{b}) \\
&= X\hat{\boldsymbol{\beta}}_\delta^{POST} + \Sigma(\Sigma + \Gamma)^{-1}(I - \delta XX^T(\delta XX^T + \Sigma + \Gamma)^{-1})(\mathbf{y} - X\mathbf{b}) \\
&= X\hat{\boldsymbol{\beta}}_\delta^{POST} + \Sigma(\Sigma + \Gamma)^{-1}(\mathbf{y} - X\hat{\boldsymbol{\beta}}_\delta^{POST}), \tag{3.4''}
\end{aligned}$$

where the first step follows by observing the similarity between (3.8) and (3.4') and the last step by substituting $\hat{\boldsymbol{\beta}}_\delta^{POST}$ with its form in (3.8). Clearly, the predictor in (3.4'') is the same as the predictor in (3.4) after replacing $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}_\delta^{POST}$. It is now only left to show that $\hat{\boldsymbol{\beta}}_\delta^{POST} \rightarrow \hat{\boldsymbol{\beta}}^{GLS}$ as $\delta \rightarrow \infty$.

We can not directly take limit as $\delta \rightarrow \infty$ in (3.8), because the matrix inverted is singular in the limit. However, notice that we can rewrite $\hat{\boldsymbol{\beta}}_\delta^{POST}$ as follows

$$\hat{\boldsymbol{\beta}}_\delta^{POST} = (X^T(\Sigma + \Gamma)^{-1}X + \tau^{-2}I)^{-1}(X^T(\Sigma + \Gamma)^{-1}\mathbf{y} + \tau^{-2}\mathbf{b}). \tag{3.9}$$

This is achieved by multiplying in (3.8) (to the left) $(X^T(\Sigma + \Gamma)^{-1}X + \tau^{-2}I)^{-1}(X^T(\Sigma + \Gamma)^{-1}X + \tau^{-2}I)$ and then noticing that

$$\begin{aligned}
&(X^T(\Sigma + \Gamma)^{-1}X + \tau^{-2}I)\delta XX^T(\delta XX^T + \Sigma + \Gamma)^{-1} \\
&= X^T(\delta(\Sigma + \Gamma)^{-1}XX^T + I)(\delta XX^T + \Sigma + \Gamma)^{-1} \\
&= X^T(\Sigma + \Gamma)^{-1}(\delta XX^T + \Sigma + \Gamma)(\delta XX^T + \Sigma + \Gamma)^{-1} \\
&= X^T(\Sigma + \Gamma)^{-1}.
\end{aligned}$$

We can now safely take $\delta \rightarrow \infty$ in (3.9) and it easily follows that $\hat{\boldsymbol{\beta}}_\delta^{POST} \rightarrow \hat{\boldsymbol{\beta}}^{GLS}$. \square

3.2.2 Covariance functions and estimation

For the plain kriging model, the parameters to be estimated are the coefficient $\boldsymbol{\beta}$, the signal covariance Σ and the noise covariance Γ . We can use the generalized least square estimator in (3.7) to estimate $\boldsymbol{\beta}$. The noises are usually assumed to be i.i.d., so Γ is diagonal as $\lambda^{-1}I$ with only one unknown parameter $\lambda > 0$. On the other hand, the signal covariance Σ is the most complex and plays a crucial role in the model. We devote this section to a brief introduction of common models and estimation techniques for Σ . See Stein [1999] for rigorous theoretical results and Diggle and Ribeiro [2006] for more practical discussions.

For Σ to be a valid covariance matrix, it must be positive semidefinite. Moreover,

it is necessary to make assumptions restricting the class of covariance matrix we consider. This is because estimating a general Σ is impossible given that we at most observe a single realization of each Y_i . To this end, it is common practice to assume stationarity and isotropy. To be more concrete, the signal covariance assumes the form $\Sigma = \sigma^2 R$, where σ^2 is the univariate variance that is the same across all variables and R is the correlation matrix that is stationary and isotropic. In other words, let \mathbf{x}_i be the location associated with the variable Y_i , then the correlation is only a function of distance

$$R_{ij} = \rho(|\mathbf{x}_i - \mathbf{x}_j|),$$

with the correlation function $\rho(\cdot)$. In geostatistical applications, we usually have $\mathbf{x}_i \in \mathbb{R}^d$ for $d = 1, 2, 3$.

Because the covariance is stationary, the *variogram* of the process becomes particularly useful and is defined below:

$$\begin{aligned} \Phi_{ij} &= \frac{1}{2} \mathbb{E}((Y_i - X_i^T \boldsymbol{\beta}) - (Y_j - X_j^T \boldsymbol{\beta}))^2 \\ &= \lambda^{-1} + \sigma^2(1 - \rho(t_{ij})), \end{aligned} \quad (3.10)$$

where $t_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$ is the distance and the second equality follows from the kriging model in (3.1) with $\Sigma_{ij} = \sigma^2 R_{ij} = \sigma^2 \rho(t_{ij})$ and $\Gamma = \lambda^{-1} I$. If we observe $Y_i = y_i$ and $Y_j = y_j$, we have an unbiased point estimate of Φ_{ij}

$$\widehat{\Phi}_{ij} = \frac{1}{2} ((y_i - X_i^T \boldsymbol{\beta}) - (y_j - X_j^T \boldsymbol{\beta}))^2,$$

which then acts as the starting point for estimating $\rho(\cdot)$ in several methods.

It is not easy to check whether a function $\rho(\cdot)$ gives rise to a legitimate covariance matrix. This is particularly hard given that the covariance needs to remain positive semidefinite at any finite list of points in \mathbb{R}^d , including some that are not yet observed. That is,

$$\sum_{i,j=1}^m c_i c_j \rho(|\mathbf{x}_i - \mathbf{x}_j|) \geq 0,$$

for any finite m , all real c_1, \dots, c_m and all $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{R}^d$. Moreover, a correlation function that is valid in \mathbb{R}^d may not be valid in dimensions higher than d , though it is guaranteed to be valid in a lower dimensional space. To remain positive semidefinite, the most negative correlation $\rho(\cdot)$ can achieve is lower bounded and the bound depends on the dimensionality. For instance, $\rho(t) \geq -0.403$ for $d = 2$ and $\rho(t) \geq -0.218$ for $d = 3$. In fact, only nonnegative correlation is allowed when $d \rightarrow \infty$. For more details see Stein [1999], page 45. This highlights a drawback of isotropic correlation

functions in high dimensions.

Generally speaking, there are two approaches to obtain a valid $\rho(\cdot)$. The first is to assume that $\rho(\cdot)$ follows a certain parametric form with parameters to be fitted from data. This is much more popular than the second non-parametric approach, where a “raw” correlation function is first fit to the data, followed by an extra step to turn the “raw” estimate into a valid correlation function. We discuss both approaches below.

Parametric correlation functions

Because it is hard to check whether a function is positive semidefinite, it is convenient and useful to have some standard families of parametric correlation functions that are known to be positive semidefinite and are flexible enough to meet the needs of real data. In the following we describe several such families. See Diggle and Ribeiro [2006] or Koehler [1990] for more details.

The *Matérn family* parameterized by $\alpha > 0$ and $\kappa > 0$ is given by

$$\rho(t) = (2^{\kappa-1}\Gamma(\kappa))^{-1}(\alpha t)^{\kappa}K_{\kappa}(\alpha t), \quad t > 0,$$

where $K_{\kappa}(\cdot)$ is a modified Bessel function of order κ and α is a scale parameter. The parameter κ is crucial as it controls the smoothness: larger κ gives smoother \mathbf{Z} . Figure 3.1(a) plots the function for three values of κ while fixing $\alpha = 4$. It also includes a one-dimensional realization of a Gaussian process corresponding to each of the Matérn correlation functions. We can see that the process becomes smoother as κ increases. It is important to point out that the Matérn family remains positive semidefinite for any dimension $d \geq 1$.

The *powered exponential family* also has two parameters and has the following form

$$\rho(t) = \exp(-(\alpha t)^{\kappa}), \quad t > 0,$$

with $\alpha > 0$ being the scale parameter and $\kappa \in (0, 2]$. These functions are also valid for any dimension $d \geq 1$. Notice that the Matérn correlation function with $\kappa = 0.5$ reduces to the powered exponential with $\kappa = 1$; while the Matérn correlation function with $\kappa \rightarrow \infty$ reduces to the powered exponential with $\kappa = 2$ (also called the Gaussian correlation function). In general, the Matérn family is favored because it requires no more parameters and it provides greater range for the possible local behavior of the signal (see Stein [1999] page 51). Figure 3.1(b) plots both the function and the simulated realizations.

The *spherical family* is widely used in classical Geostatistics, and it only has a

single parameter $\alpha > 0$

$$\rho(t) = \begin{cases} 1 - 1.5\alpha t + 0.5(\alpha t)^3 & 0 \leq t \leq \alpha^{-1} \\ 0 & t > \alpha^{-1} \end{cases}.$$

Notice that this family only has a finite range where two random variables more than α^{-1} distance apart are uncorrelated. It is also less flexible compared to the Matérn family, and is positive semidefinite only when $d \leq 3$. The function and its Gaussian realizations are plotted in Figure 3.1(c).

We refer the readers to Schlather [1999] for some other correlation functions that are less commonly used, and instead briefly discuss methods for fitting these parametric families. By and large, there are curve-fitting methods and maximum likelihood methods. The curve fitting algorithm uses an ordinary least square criterion on the variogram

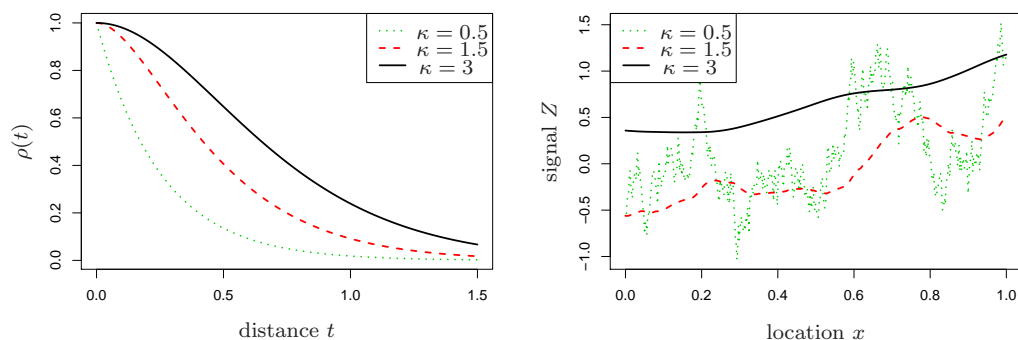
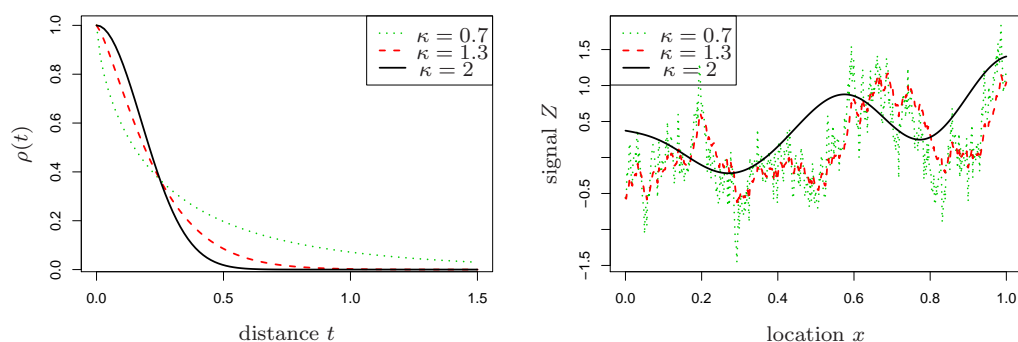
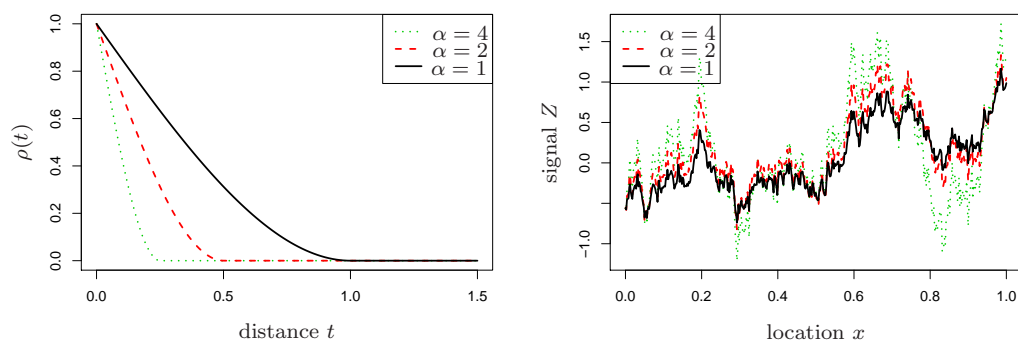
$$\min_{\alpha, \kappa} \sum_{i=1}^r \sum_{j=1}^r (\hat{\Phi}_{ij} - \Phi_{ij}(\alpha, \kappa))^2,$$

where α and κ represent parameters that the correlation function depends on. There are also variations of this criterion. For example, one refinement is to use weighted least square to account for the sampling variance of $\hat{\Phi}_{ij}$. On the other hand, the maximum likelihood method works by directly considering the Gaussian likelihood function of \mathbf{Y} . It maximizes the log-likelihood with respect to the unknown parameters and obtain estimates that are asymptotically optimal (under mild regularity conditions. See Lehmann and Casella [2003]).

Nonparametric correlation functions

The reason that nonparametric approaches are not widely used is because it is difficult to preserve positive semidefiniteness enjoyed by a true covariance function. However, parametric approaches can be inflexible and inadequate. Several nonparametric estimators of correlation function have been proposed in the literature. For example, Shapiro and Botha [1991] propose a constrained curve-fitting approach that produces correlation values at a discrete set of distances. We describe here the method suggested by Hall et al. [1994]. Even though their approach applies to \mathbb{R}^1 only, it shares some intuition behind part of our graph kriging algorithm to be discussed in Chapter 4.

Before we start, recall that by Bochner's theorem, a function is continuous and positive semidefinite if and only if it is the Fourier transform of a nonnegative bounded

(a) Matérn family, $\alpha = 4$ (b) Powered exponential family, $\alpha = 4$ 

(c) Spherical family

Figure 3.1: Parametric correlation functions (left) and their corresponding one-dimensional realization of Gaussian process (right). Notice that larger κ gives a smoother signal for both Matérn and powered exponential families.

Borel measure. Define the Fourier transform of $\rho(\cdot)$ to be

$$\rho^\dagger(\theta) \equiv \int_{-\infty}^{\infty} \rho(t)e^{i\theta t} dt = 2 \int_0^{\infty} \rho(t)\cos(\theta t)dt,$$

then we know that it must satisfy

$$\rho^\dagger(\theta) \geq 0, \quad \forall \theta,$$

for any legitimate correlation function $\rho(\cdot)$.

We first obtain an initial estimator of $\rho(\cdot)$ and denote it $\hat{\rho}(\cdot)$. This can be done, for instance, by back solving for $\rho(t_{ij})$ in (3.10) based on $\hat{\Phi}_{ij}$ and then using kernel smoothing to get a continuous function $\hat{\rho}(\cdot)$. Clearly, $\hat{\rho}(\cdot)$ is not necessarily positive semidefinite, and thus the following steps are taken to obtain the final estimate:

- 1) Compute the Fourier transform, $\hat{\rho}^\dagger(\cdot)$, of $\hat{\rho}(\cdot)$.
- 2) Render $\hat{\rho}^\dagger(\cdot)$ nonnegative, for example by deleting any negative lobes and perhaps doing a little additional smoothing. Denote the resulting function $\tilde{\rho}^\dagger(\cdot)$.
- 3) Fourier-invert $\tilde{\rho}^\dagger(\cdot)$ to obtain

$$\tilde{\rho}(t) = (2\pi)^{-1} \int_{-\infty}^{\infty} \tilde{\rho}^\dagger(\theta)e^{-i\theta t} d\theta,$$

which is positive semidefinite by construction.

3.3 Semi-supervised learning as kriging

In Section 2.2, we reviewed several graph prediction algorithms in semi-supervised learning literature. In this section, we will show that these methods can be expressed in terms of kriging with covariance assumptions driven by the geometry of the graph.

As a starting point, notice that we can unify these algorithms in the following sense. In each case there is a quadratic variation $\Omega(\mathbf{Z})$ and a quadratic error norm on $\mathbf{Z} - \mathbf{Y}^*$, each of which should ideally be small subject to a trade-off between them. We take $\Omega(\mathbf{Z}) = \mathbf{Z}^T L \mathbf{Z}$ for a smoothing matrix L and measure the error between \mathbf{Z} and \mathbf{Y}^* by $(\mathbf{Z} - \mathbf{Y}^*)^T \Lambda (\mathbf{Z} - \mathbf{Y}^*)$. The smoothing matrix L is positive semidefinite and Λ is a diagonal matrix with $\Lambda_{ii} \geq 0$, while the sum $L + \Lambda$ is invertible. The algorithm then picks the minimizer of

$$Q(\mathbf{Z}) = \mathbf{Z}^T L \mathbf{Z} + (\mathbf{Z} - \mathbf{Y}^*)^T \Lambda (\mathbf{Z} - \mathbf{Y}^*). \quad (3.11)$$

In Table 3.1, we summarize the methods reviewed in Section 2.2 in the form above. Note that a few of these methods are only defined for undirected graphs. To apply one

Method	L	Λ
Random walk smoothing [Zhou et al., 2005a]	$\tilde{\Delta}'$	λI
Tikhonov smoothing [Belkin et al., 2004]	Δ	$\begin{pmatrix} \lambda_0 I_r & 0 \\ 0 & 0 I_{n-r} \end{pmatrix}$
Undirected RW smoothing [Zhou et al., 2004]	$\tilde{\Delta}$	λI
Hub & authority smoothing [Zhou et al., 2005b]	$(1-\gamma)\tilde{\Delta}'_A + \gamma\tilde{\Delta}'_H$	λI
Manifold smoothing [Belkin et al., 2006]	$K + \gamma\Delta$	$\begin{pmatrix} \lambda_0 I_r & 0 \\ 0 & 0 I_{n-r} \end{pmatrix}$
Spectral transform	$\sum_{i=1}^n f(\tau_i)^{-1} \mathbf{u}_i \mathbf{u}_i^T$	$\begin{pmatrix} \lambda_0 I_r & 0 \\ 0 & \lambda_1 I_{n-r} \end{pmatrix}$

Table 3.1: Summary of the semi-supervised learning methods from Section 2.2 in the form of equation (3.11).

of them to a given directed graph, the standard technique is to work with $W + W^T$.

The following result is trivial, but it makes clear the necessary conditions to obtain the predictor (3.12) below.

Lemma 3.3.1. *Let $L \in \mathbb{R}^{n \times n}$ be a positive semidefinite matrix and $\Lambda \in \mathbb{R}^{n \times n}$ be a diagonal matrix with $\Lambda_{ii} \geq 0$, and assume that $L + \Lambda$ is invertible. Let $\mathbf{Z}, \mathbf{Y}^* \in \mathbb{R}^n$, then the minimizer of (3.11) is*

$$\hat{\mathbf{Y}} \equiv \arg \min_{\mathbf{Z} \in \mathbb{R}^n} Q(\mathbf{Z}) = (L + \Lambda)^{-1} \Lambda \mathbf{Y}^*. \quad (3.12)$$

Proof. The first derivative of $Q(\mathbf{Z})$ is

$$\frac{\partial Q(\mathbf{Z})}{\partial \mathbf{Z}} = 2L\mathbf{Z} + 2\Lambda(\mathbf{Z} - \mathbf{Y}^*),$$

using the fact that both L and Λ are symmetric. Setting the above equation to zero and solving for \mathbf{Z} , we get the desired result, noting that $L + \Lambda$ is invertible. \square

Notice that predictors in the form of (3.12) are linear in \mathbf{Y}^* . Therefore, we might expect these semi-supervised learning methods to have a representation as a minimum mean squared error linear prediction under a Gaussian process model for \mathbf{Z} . That is, they might be a form of kriging.

In the rest of this section, we will show that, for each algorithm considered in Section 2.2, there is a corresponding kriging estimator based on the full data Bayesian formulation in (3.4') that we recall below:

$$\hat{\mathbf{Z}}_\delta = (\delta X X^T + \Sigma) (\delta X X^T + \Sigma + \Gamma)^{-1} (\mathbf{y} - X\mathbf{b}) + X\mathbf{b}.$$

Particularly, to get a semi-supervised learning predictor, we

- 1) make strategic choices for Γ , Σ , and X ,
- 2) treat the missing parts of \mathbf{Y} as observed,
- 3) use the full data kriging estimator (3.4'), and then
- 4) take necessary limits.

The detailed recipes are given below. Then, to allow easy comparison of the methods, we present a summary in Table 3.2 at the end of this section.

3.3.1 Random walk smoothing

It easily follows from Lemma 3.3.1 that the random walk smoother in (2.5) is

$$\begin{aligned}\widehat{\mathbf{Y}} &= \lambda(\widetilde{\Delta}' + \lambda I)^{-1} \mathbf{Y}^* \\ &= (I + \lambda^{-1} \widetilde{\Delta}')^{-1} \mathbf{Y}^*,\end{aligned}\tag{3.13}$$

where we recall that $\widetilde{\Delta}'$ is the normalized graph Laplacian of the graph \mathcal{G}' with weights $w'_{ij} = (\pi_i P_{ij} + \pi_j P_{ji})/2$. The following theorem shows that the random walk predictor above can be cast in terms of a sequence of kriging estimators.

Theorem 3.3.2. *Let $\mathbf{Y} = \mathbf{Z} + \boldsymbol{\varepsilon} \in \mathbb{R}^n$. Suppose that $\mathbf{Z} = X\beta + \mathbf{S}$ where $X \in \mathbb{R}^n$, $\beta \sim \mathcal{N}(b, \delta)$ and $\mathbf{S} \sim \mathcal{N}(0, \Sigma)$. Let $\boldsymbol{\varepsilon} \sim N(0, \Gamma)$ and assume that \mathbf{S} , β , and $\boldsymbol{\varepsilon}$ are mutually independent. Suppose that $\mathbf{Y}^{(0)}$ comprising the first $r > 1$ elements of \mathbf{Y} is observed. Let $\mathbf{Y}^* \in \mathbb{R}^n$ with $Y_i^* = Y_i^{(0)}$ for $i = 1, \dots, r$ and $Y_i^* = bX_i$ for $i = r + 1, \dots, n$. Let $\widehat{\mathbf{Z}}_\delta^*$ be the kriging estimator (3.4') applied with $\mathbf{y} = \mathbf{Y}^*$. Assume that the graph \mathcal{G}' has only one connected component. We now choose*

$$\begin{aligned}\Gamma &= \lambda^{-1} I, \\ \Sigma &= \widetilde{\Delta}'^+, \quad \text{and} \\ X &= (\sqrt{\pi_1}, \dots, \sqrt{\pi_n})^T,\end{aligned}$$

where $\widetilde{\Delta}'^+$ is the Moore-Penrose inverse of $\widetilde{\Delta}'$. Then

$$\lim_{\delta \rightarrow \infty} \widehat{\mathbf{Z}}_\delta^* = (I + \lambda^{-1} \widetilde{\Delta}')^{-1} \mathbf{Y}^*,$$

which is the random walk predictor given by (3.13).

Proof. First we notice that since \mathcal{G}' is connected, the eigen decomposition of $\widetilde{\Delta}'$ is

$$\widetilde{\Delta}' = U \text{diag}(\tau_1, \tau_2, \dots, \tau_n) U^T,\tag{3.14}$$

where $U^T U = I_n$, with $\tau_i > 0$ for $i < n$ and $\tau_n = 0$ by Proposition 2.1.2. Further, since the new degrees $d'_i = \pi_i$ from (2.4) we have that the last column of U is $\pm \sqrt{\boldsymbol{\pi}}$.

The kriging estimator is $\widehat{\mathbf{Z}}_\delta^* = M_\delta(\mathbf{Y}^* - bX) + bX$, where

$$\begin{aligned}M_\delta &= \left(\delta X X^T + \Sigma \right) \left(\delta X X^T + \Sigma + \Gamma \right)^{-1} \\ &= \left(\delta \sqrt{\boldsymbol{\pi}} \sqrt{\boldsymbol{\pi}}^T + \widetilde{\Delta}'^+ \right) \left(\delta \sqrt{\boldsymbol{\pi}} \sqrt{\boldsymbol{\pi}}^T + \widetilde{\Delta}'^+ + \lambda^{-1} I \right)^{-1}.\end{aligned}\tag{3.15}$$

Using (3.14), we can write

$$\delta\sqrt{\boldsymbol{\pi}}\sqrt{\boldsymbol{\pi}}^T + \tilde{\Delta}'^+ = U \operatorname{diag}\left(\frac{1}{\tau_1}, \frac{1}{\tau_2}, \dots, \frac{1}{\tau_{n-1}}, \delta\right) U^T, \quad (3.16)$$

which is invertible. Moving its inverse inside the matrix inverse in (3.15), we get

$$\begin{aligned} M_\delta &= \left(I + \lambda^{-1} \left(\delta\sqrt{\boldsymbol{\pi}}\sqrt{\boldsymbol{\pi}}^T + \tilde{\Delta}'^+ \right)^{-1} \right)^{-1} \\ &= \left(I + \lambda^{-1} U \operatorname{diag}\left(\tau_1, \tau_2, \dots, \tau_{n-1}, \delta^{-1}\right) U^T \right)^{-1}, \end{aligned}$$

where the second equality follows by applying (3.16).

Letting $\delta \rightarrow \infty$

$$M_\delta \rightarrow M_\infty = (I + \lambda^{-1} \tilde{\Delta}')^{-1}.$$

This limit exists because the matrix being inverted is positive definite.

Finally, the terms related to the mean bX in $\hat{\mathbf{Z}}_\delta = M_\delta(\mathbf{Y}^* - bX) + bX$ vanish because

$$\begin{aligned} (M_0 X - X) &= (I + \lambda^{-1} \tilde{\Delta}')^{-1} \sqrt{\boldsymbol{\pi}} - \sqrt{\boldsymbol{\pi}} \\ &= (I + \lambda^{-1} \tilde{\Delta}')^{-1} (\lambda^{-1} \tilde{\Delta}' \sqrt{\boldsymbol{\pi}} + \sqrt{\boldsymbol{\pi}}) - \sqrt{\boldsymbol{\pi}} \\ &= (I + \lambda^{-1} \tilde{\Delta}')^{-1} (\lambda^{-1} \tilde{\Delta}' + I) \sqrt{\boldsymbol{\pi}} - \sqrt{\boldsymbol{\pi}} \\ &= \mathbf{0}. \end{aligned}$$

The second equality follows because $\tilde{\Delta}' \sqrt{\boldsymbol{\pi}} = \mathbf{0}$. Therefore, in view of (3.13), $\hat{\mathbf{Z}}_\delta^* \rightarrow \hat{\mathbf{Y}}$ as $\delta \rightarrow \infty$. \square

One thing that stands out from the kriging analysis is the vector $X = \sqrt{\boldsymbol{\pi}}$ interpreted component-wise. The equivalent prior on \mathbf{Y} in the direction parallel to X is improper. Thus the method anticipates that \mathbf{Y} could reasonably be a large multiple of X . When $Y \doteq \beta X$ for some value $\beta \neq 0$ the similar nodes are the ones with comparable values of $\sqrt{\pi_i}$. These are not necessarily close together in the graph.

The next thing that stands out is that the correlation strength between nodes is a fixed property of W , the graph adjacency matrix. If some response variables have stronger local correlations, others weaker, and still others negative local correlations, that is not reflected in this choice of Σ .

Finally, notice that the predictions do not interpolate the known values. One reason not to interpolate, is that when the graph correlations are strong, it may be possible to detect erroneous labels as cases where $|\hat{Z}_i^{(0)} - Y_i^{(0)}|$ is large.

3.3.2 Tikhonov smoothing

Introduce

$$\Lambda_{\lambda_0, \lambda_1} = \text{diag}(\lambda_0 I_r, \lambda_1 I_{n-r}),$$

then by Lemma 3.3.1 the Tikhonov smoother in (2.7) is

$$\widehat{\mathbf{Y}} = (\Delta + \Lambda_{\lambda_0, 0})^{-1} \Lambda_{\lambda_0, 0} \mathbf{Y}^*. \quad (3.17)$$

where Y_i^* for $i > r$ can be arbitrarily chosen as they will be zeroed out by $\Lambda_{\lambda_0, 0}$. The following theorem shows that this predictor can be written as limit of a sequence of kriging estimators.

Theorem 3.3.3. *Let $\mathbf{Y} = \mathbf{Z} + \boldsymbol{\varepsilon} \in \mathbb{R}^n$. Suppose that $\mathbf{Z} = X\boldsymbol{\beta} + \mathbf{S}$ where $X \in \mathbb{R}^n$, $\boldsymbol{\beta} \sim \mathcal{N}(b, \delta)$ and $\mathbf{S} \sim \mathcal{N}(0, \Sigma)$. Let $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \Gamma)$ and assume that \mathbf{S} , $\boldsymbol{\beta}$, and $\boldsymbol{\varepsilon}$ are mutually independent. Suppose that $\mathbf{Y}^{(0)}$ comprising the first $r > 1$ elements of \mathbf{Y} is observed. Let $\mathbf{Y}^* \in \mathbb{R}^n$ with $Y_i^* = Y_i^{(0)}$ for $i = 1, \dots, r$ and Y_i^* arbitrarily chosen for $i = r + 1, \dots, n$. Let $\widehat{\mathbf{Z}}_\delta^*$ be the kriging estimator (3.4') applied with $\mathbf{y} = \mathbf{Y}^*$. Assume that the graph \mathcal{G} has only one connected component. We now choose*

$$\begin{aligned} \Gamma &= \text{diag}(\lambda_0^{-1} I_r, \lambda_1^{-1} I_{n-r}), \\ \Sigma &= \Delta^+ \quad \text{and,} \\ X &= \mathbf{1}_n, \end{aligned}$$

where Δ^+ is the Moore-Penrose inverse of Δ . Then

$$\lim_{\lambda_1 \rightarrow 0^+} \lim_{\delta \rightarrow \infty} \widehat{\mathbf{Z}}_\delta^* = (\Delta + \Lambda_{\lambda_0, 0})^{-1} \Lambda_{\lambda_0, 0} \mathbf{Y}^*$$

is the Tikhonov smoother given by (3.17).

Proof. First we notice that since \mathcal{G} is connected, the eigen decomposition of Δ is

$$\Delta = U \text{diag}(\tau_1, \tau_2, \dots, \tau_n) U^T, \quad (3.18)$$

where $U^T U = I_n$, with $\tau_i > 0$ for $i < n$ and $\tau_n = 0$ by Proposition 2.1.1. Further, we know the last column of U is $\mathbf{1}_n / \sqrt{n}$.

The kriging estimator is $\widehat{\mathbf{Z}}_\delta^* = M_{\delta, \lambda_1} (\mathbf{Y}^* - bX) + bX$, where

$$\begin{aligned} M_{\delta, \lambda_1} &= (\delta X X^T + \Sigma) (\delta X X^T + \Sigma + \Gamma)^{-1} \\ &= (\delta \mathbf{1}_n \mathbf{1}_n^T + \Delta^+) (\delta \mathbf{1}_n \mathbf{1}_n^T + \Delta^+ + \Lambda_{\lambda_0, \lambda_1}^{-1})^{-1}. \end{aligned} \quad (3.19)$$

Using (3.18), we can write

$$\delta \mathbf{1}_n \mathbf{1}_n^T + \Delta^+ = U \text{diag} \left(\frac{1}{\tau_1}, \frac{1}{\tau_2}, \dots, \frac{1}{\tau_{n-1}}, n\delta \right) U^T, \quad (3.20)$$

which is invertible. Moving its inverse inside the matrix inverse in (3.19), we get

$$\begin{aligned} M_{\delta, \lambda_1} &= \left(I + \Lambda_{\lambda_0, \lambda_1}^{-1} \left(\delta \mathbf{1}_n \mathbf{1}_n^T + \Delta^+ \right)^{-1} \right)^{-1} \\ &= \left(\Lambda_{\lambda_0, \lambda_1} + U \text{diag} \left(\tau_1, \tau_2, \dots, \tau_{n-1}, \frac{1}{n\delta} \right) U^T \right)^{-1} \Lambda_{\lambda_0, \lambda_1}, \end{aligned}$$

where the second equality follows from (3.20).

Letting $\delta \rightarrow \infty$

$$M_{\delta, \lambda_1} \rightarrow M_{\infty, \lambda_1} = (\Lambda_{\lambda_0, \lambda_1} + \Delta)^{-1} \Lambda_{\lambda_0, \lambda_1}.$$

This limit exists because the matrix being inverted is positive definite. Then let $\lambda_1 \rightarrow 0^+$

$$M_{\infty, \lambda_1} \rightarrow M_{\infty, 0} = (\Lambda_{\lambda_0, 0} + \Delta)^{-1} \Lambda_{\lambda_0, 0},$$

noting that $\Lambda_{\lambda_0, 0} + \Delta$ is also positive definite.

Finally, the terms related to the mean bX in $\widehat{\mathbf{Z}}_\delta = M_{\delta, \lambda_1} (\mathbf{Y}^* - bX) + bX$ vanish because

$$\begin{aligned} M_{\infty, 0} X - X &= (\Lambda_{\lambda_0, 0} + \Delta)^{-1} \Lambda_{\lambda_0, 0} \mathbf{1}_n - \mathbf{1}_n \\ &= (\Lambda_{\lambda_0, 0} + \Delta)^{-1} (\Lambda_{\lambda_0, 0} \mathbf{1}_n + \Delta \mathbf{1}_n) - \mathbf{1}_n \\ &= (\Lambda_{\lambda_0, 0} + \Delta)^{-1} (\Lambda_{\lambda_0, 0} + \Delta) \mathbf{1}_n - \mathbf{1}_n \\ &= \mathbf{0}. \end{aligned}$$

The second equality follows because $\Delta \mathbf{1}_n = \mathbf{0}$. Therefore, in view of (3.17), $\widehat{\mathbf{Z}}_\delta^* \rightarrow \widehat{\mathbf{Y}}$ as $\delta \rightarrow \infty$ and $\lambda_1 \rightarrow 0^+$. \square

3.3.3 Interpolated Tikhonov smoothing

The interpolating algorithm minimizes

$$Q(\mathbf{Z}) = \mathbf{Z}^T \Delta \mathbf{Z},$$

subject to $Z_i^{(0)} = \mathbf{Y}_i^{(0)}$. We partition Δ as follows

$$\Delta = \begin{pmatrix} \Delta_{00} & \Delta_{01} \\ \Delta_{10} & \Delta_{11} \end{pmatrix}, \quad (3.21)$$

and get

$$Q(\mathbf{Z}) = \mathbf{Z}^{(0)T} \Delta_{00} \mathbf{Z}^{(0)} + \mathbf{Z}^{(1)T} \Delta_{11} \mathbf{Z}^{(1)} + 2\mathbf{Z}^{(1)T} \Delta_{10} \mathbf{Z}^{(0)}.$$

Letting $\mathbf{Z}^{(0)} = \mathbf{Y}^{(0)}$, the first derivative w.r.t. $\mathbf{Z}^{(1)}$ is

$$\frac{\partial Q(\mathbf{Z})}{\partial \mathbf{Z}^{(1)}} = 2\Delta_{10} \mathbf{Y}^{(0)} + 2\Delta_{11} \mathbf{Z}^{(1)}.$$

Setting the derivative to zero and solving for $\mathbf{Z}^{(1)}$, we get

$$\widehat{\mathbf{Y}}^{(1)} = -\Delta_{11}^{-1} \Delta_{10} \mathbf{Y}^{(0)}. \quad (3.22)$$

We now show that this predictor is the kriging estimator $\widehat{\mathbf{Z}}_\delta^*$ after making the same choices for Γ , Σ and X as the Tikhonov predictor and taking limits

$$\lim_{\lambda_0 \rightarrow \infty} \lim_{\lambda_1 \rightarrow 0^+} \lim_{\delta \rightarrow \infty} \widehat{\mathbf{Z}}_\delta^*.$$

In view of Theorem 3.3.3, it is sufficient to show that the interpolated predictor (3.22) can be obtained by taking $\lambda_0 \rightarrow \infty$ in the Tikhonov predictor (3.17).

Using the same partition as in (3.21), the Tikhonov predictor (3.17) becomes

$$\begin{aligned} \widehat{\mathbf{Y}}_{\lambda_0} &= \begin{pmatrix} \Delta_{00} + \lambda_0 I_r & \Delta_{01} \\ \Delta_{10} & \Delta_{11} \end{pmatrix}^{-1} \begin{pmatrix} \lambda_0 I_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{Y}^{(0)} \\ \mathbf{Y}^{*(1)} \end{pmatrix} \\ &= \begin{pmatrix} \Delta_{00} + \lambda_0 I_r & \Delta_{01} \\ \Delta_{10} & \Delta_{11} \end{pmatrix}^{-1} \begin{pmatrix} \lambda_0 \mathbf{Y}^{(0)} \\ \mathbf{0}_{n-r} \end{pmatrix}. \end{aligned} \quad (3.23)$$

Let A be the matrix inverse in (3.23). We have

$$\begin{aligned} A_{00} &= (\Delta_{00} + \lambda_0 I_r - \Delta_{01} \Delta_{11}^{-1} \Delta_{10})^{-1}, \\ A_{10} &= -\Delta_{11}^{-1} \Delta_{10} (\Delta_{00} + \lambda_0 I_r - \Delta_{01} \Delta_{11}^{-1} \Delta_{10})^{-1}. \end{aligned}$$

Plugging back into (3.23), we get

$$\begin{aligned} \widehat{\mathbf{Y}}_{\lambda_0}^{(0)} &= (\lambda_0^{-1} \Delta_{00} + I_r - \lambda_0^{-1} \Delta_{01} \Delta_{11}^{-1} \Delta_{10})^{-1} \mathbf{Y}^{(0)}, \\ \widehat{\mathbf{Y}}_{\lambda_0}^{(1)} &= -\Delta_{11}^{-1} \Delta_{10} (\lambda_0^{-1} \Delta_{00} + I_r - \lambda_0^{-1} \Delta_{01} \Delta_{11}^{-1} \Delta_{10})^{-1} \mathbf{Y}^{(0)}. \end{aligned}$$

Now taking $\lambda_0 \rightarrow \infty$

$$\begin{aligned}\widehat{\mathbf{Y}}_{\lambda_0}^{(0)} &\rightarrow \mathbf{Y}^{(0)}, \\ \widehat{\mathbf{Y}}_{\lambda_0}^{(1)} &\rightarrow -\Delta_{11}^{-1}\Delta_{10}\mathbf{Y}^{(0)},\end{aligned}$$

which give the interpolated predictor in (3.22). The limits exist because the matrices being inverted are positive definite.

3.3.4 Undirected random walk smoothing

By Lemma 3.3.1, the undirected random walk smoother in (2.8) is

$$\widehat{\mathbf{Y}} = \lambda(\widetilde{\Delta} + \lambda I)^{-1}\mathbf{Y}^*. \quad (3.24)$$

This estimate reduces to the kriging estimator $\widehat{\mathbf{Z}}_{\delta}^*$ in (3.4') with the following choices:

$$\begin{aligned}\Gamma &= \lambda^{-1}I, \\ \Sigma &= \widetilde{\Delta}^+, \quad \text{and,} \\ X &= (\sqrt{d_1}, \dots, \sqrt{d_n})^T,\end{aligned}$$

in the limit as $\delta \rightarrow \infty$.

Because undirected random walk smoothing is a special case of the directed random walk, proof for this example follows the same as the proof for Theorem 3.3.2 and hence is omitted.

3.3.5 Hub and authority smoothing

By Lemma 3.3.1 the hub-and-authority predictor in (2.10) is

$$\widehat{\mathbf{Y}} = \lambda(\gamma\widetilde{\Delta}'_H + (1 - \gamma)\widetilde{\Delta}'_A + \lambda I)^{-1}\mathbf{Y}^*. \quad (3.25)$$

Ordinarily, $L = \gamma\widetilde{\Delta}'_H + (1 - \gamma)\widetilde{\Delta}'_A$ is positive definite for $0 < \gamma < 1$. The two terms each have one eigenvector with eigenvalue 0, but those two eigenvectors are, in general, linearly independent. We can construct exceptions. For example if \mathcal{G} is the complete graph then the hub and authority walks coincide and L reduces to the random walk case which has one zero eigenvalue. More generally if every node has $w_{i+} = w_{+i}$ the same thing happens. Outside of such pathological examples, L is positive definite.

For later, we state and prove a general result in the lemma below. It connects the kriging estimator with the general semi-supervised learning predictor in (3.12) for the

special case where both L and Λ are positive definite.

Lemma 3.3.4. *In Lemma 3.3.1, we further assume that L is positive definite and $\Lambda_{ii} > 0$. Let $\widehat{\mathbf{Z}}_\delta^*$ be the kriging estimator (3.4') applied with $\mathbf{y} = \mathbf{Y}^*$, and choose*

$$\begin{aligned}\Gamma &= \Lambda^{-1}, \\ \Sigma &= L^{-1}, \quad \text{and} \\ X &= \mathbf{0}_n.\end{aligned}$$

Then

$$\widehat{\mathbf{Z}}_\delta^* = (L + \Lambda)^{-1} \Lambda \mathbf{Y}^*$$

is the general semi-supervised learning predictor in (3.12).

Proof. With the assumed choices for Γ , Σ and X , the kriging estimator becomes

$$\begin{aligned}\widehat{\mathbf{Z}}_\delta &= (\delta X X^T + \Sigma) (\delta X X^T + \Sigma + \Gamma)^{-1} (\mathbf{y} - bX) + bX \\ &= L^{-1} (L^{-1} + \Lambda^{-1})^{-1} \mathbf{Y}^* \\ &= (L + \Lambda)^{-1} \Lambda \mathbf{Y}^*,\end{aligned}$$

where the last step follows because L and Λ are both invertible. \square

It now follows easily from the lemma that the hub-and-authority smoother in (2.10) matches the kriging estimator $\widehat{\mathbf{Z}}_\delta^*$ in (3.4') with the following choices:

$$\begin{aligned}\Gamma &= \lambda^{-1} I, \\ \Sigma &= (\gamma \widetilde{\Delta}'_H + (1 - \gamma) \widetilde{\Delta}'_A)^{-1}, \quad \text{and} \\ X &= \mathbf{0}_n.\end{aligned}$$

3.3.6 Manifold smoothing

The manifold regularization in (2.11) with a linear kernel K is

$$\widehat{\mathbf{Y}} = (K + \gamma \Delta + \Lambda_{\lambda_0, 0})^{-1} \Lambda_{\lambda_0, 0} \mathbf{Y}^*, \quad (3.26)$$

by Lemma 3.3.1.

We have two cases to consider. The matrix $\gamma \Delta$ has $n - 1$ positive eigenvalues and an eigenvalue of 0 for the eigenvector $\mathbf{1}_n$. If $K \mathbf{1}_n = \mathbf{0}_n$ then $L = K + \gamma \Delta$ is singular but otherwise L is positive definite.

When L is positive definite the implied prior is not improper in any direction so we take $X = \mathbf{0}_n$. In this case, the manifold regularization predictions (3.26) are from

the kriging estimator (3.4') with the following choices:

$$\begin{aligned}\Gamma &= \text{diag}(\lambda_0^{-1}I_r, \lambda_1^{-1}I_{n-r}), \\ \Sigma &= (K + \gamma\Delta)^{-1}, \quad \text{and} \\ X &= \mathbf{0}_n,\end{aligned}$$

in the limit $\lambda_1 \rightarrow 0^+$. This follows by first applying Lemma 3.3.4 and then taking the limit.

Now suppose that $K + \gamma\Delta$ fails to be invertible because K has eigenvector $\mathbf{1}_n$ with eigenvalue 0. In this case, we replace $(K + \gamma\Delta)^{-1}$ by the corresponding Moore-Penrose inverse and use $X = \mathbf{1}_n$, taking the limit $\delta \rightarrow \infty$ and then $\lambda_1 \rightarrow 0^+$. The proof is identical to that of the Tikhonov smoothing in Theorem 3.3.3.

Our condition that the Mercer kernel be linear is necessary. For a general Mercer Kernel \mathcal{K} , the prediction $\hat{\mathbf{Y}}$ need not be linear in \mathbf{Y}^* , and so for such kernels, manifold regularization does not reduce to kriging.

3.3.7 Spectral transformation of Δ

In the spectral transform examples, we have two cases to consider as well. This is because $\tau_n = 0$ in (2.12) so $f(\tau_n)$ may equal to 0.

When $f(\tau_n) > 0$, the connection to kriging can be written as

$$\begin{aligned}\Gamma &= \text{diag}(\lambda_0^{-1}I_r, \lambda_1^{-1}I_{n-r}) \\ \Sigma &= \sum_{i=1}^n f(\tau_i)^{-1} \mathbf{u}_i \mathbf{u}_i^T, \quad \text{and} \\ X &= \mathbf{0}_n,\end{aligned}$$

follows from Lemma 3.3.4

When $f(d_n) = 0$, Γ remains the same but now

$$\begin{aligned}\Sigma &= \sum_{i=1}^{n-1} f(\tau_i)^{-1} \mathbf{u}_i \mathbf{u}_i^T, \quad \text{and} \\ X &= \mathbf{1}_n,\end{aligned}$$

with $\delta \rightarrow \infty$. The proof is then identical to that of the Tikhonov smoothing in Theorem 3.3.3.

Method	Γ	Σ	X	Limits
Random walk [Zhou et al., 2005a]	$\lambda^{-1}I$	$\tilde{\Delta}'^+$	$\Pi^{1/2}\mathbf{1}_n$	$\delta \rightarrow \infty$
Tikhonov [Belkin et al., 2004]	$\begin{pmatrix} \lambda_0^{-1}I_r & 0 \\ 0 & \lambda_1^{-1}I_{n-r} \end{pmatrix}$	Δ^+	$\mathbf{1}_n$	$\delta \rightarrow \infty$ $\lambda_1 \rightarrow 0$
Interpolated Tik [Belkin et al., 2004]	$\begin{pmatrix} \lambda_0^{-1}I_r & 0 \\ 0 & \lambda_1^{-1}I_{n-r} \end{pmatrix}$	Δ^+	$\mathbf{1}_n$	$\delta \rightarrow \infty$ $\lambda_1 \rightarrow 0$ $\lambda_0 \rightarrow \infty$
Undirected RW [Zhou et al., 2004]	$\lambda^{-1}I$	$\tilde{\Delta}^+$	$D^{1/2}\mathbf{1}_n$	$\delta \rightarrow \infty$
Hub & authority [Zhou et al., 2005b]	$\lambda^{-1}I$	$((1-\gamma)\tilde{\Delta}'_A + \gamma\tilde{\Delta}'_H)^{-1}$	$\mathbf{0}_n$	—
Manifold, $K\mathbf{1}_n \neq \mathbf{0}_n$ [Belkin et al., 2006]	$\begin{pmatrix} \lambda_0^{-1}I_r & 0 \\ 0 & \lambda_1^{-1}I_{n-r} \end{pmatrix}$	$(K + \gamma\Delta)^{-1}$	$\mathbf{0}_n$	$\lambda_1 \rightarrow 0$
Manifold, $K\mathbf{1}_n = \mathbf{0}_n$ [Belkin et al., 2006]	$\begin{pmatrix} \lambda_0^{-1}I_r & 0 \\ 0 & \lambda_1^{-1}I_{n-r} \end{pmatrix}$	$(K + \gamma\Delta)^+$	$\mathbf{1}_n$	$\delta \rightarrow \infty$ $\lambda_1 \rightarrow 0$
Spectral transform $f(d_n) > 0$	$\begin{pmatrix} \lambda_0^{-1}I_r & 0 \\ 0 & \lambda_1^{-1}I_{n-r} \end{pmatrix}$	$\sum_{i=1}^n f(\tau_i)^{-1}\mathbf{u}_i\mathbf{u}_i^T$	$\mathbf{0}_n$	—
Spectral transform $f(d_n) = 0$	$\begin{pmatrix} \lambda_0^{-1}I_r & 0 \\ 0 & \lambda_1^{-1}I_{n-r} \end{pmatrix}$	$\sum_{i=1}^{n-1} f(\tau_i)^{-1}\mathbf{u}_i\mathbf{u}_i^T$	$\mathbf{1}_n$	$\delta \rightarrow \infty$

Table 3.2: Summary of connections between some semi-supervised learning methods and kriging.

Chapter 4

Empirical stationary correlation

4.1 Introduction

In Chapter 3, we have established connections between kriging and several semi-supervised learning models for prediction on graphs. Such relationships are themselves interesting, but what is more striking to us is that, the connections to kriging reveal a unanimous assumption by all these models: the signal covariance is a given function of the graph adjacency matrix W . This is clearly not an effective way to capture the various correlation properties that different response variables may present. For instance, even on the same social network (i.e. the same W), friends may correlate differently for age, than for gender, school attended, or opinions about music, movies or restaurants.

This troubling feature motivates us to propose a different model for the signal covariance that can adapt to the nature of the response variable. Similar to the graph prediction methods discussed thus far, we keep the kriging framework as presented in Section 3.2.1, but now we show how to adapt the covariance to the dependency pattern seen among the non-missing \mathbf{Y} values.

The outline of this chapter is as follows. We start with a simple example in Section 4.2, where we demonstrate that the fixed correlation structure assumed under these learning models can lead to very poor performance in terms of prediction. In Section 4.3 we derive another kriging method incorporating the empirical variogram of the observed Y_i values into an estimate of the covariance. That method uses a full rank covariance, which is therefore computationally expensive for large n . We also present a lower rank version more suitable to large scale problems.

Section 4.4 presents two numerical examples. In Section 4.4.1, Y_i is a numerical measure of the research quality of 107 universities in the UK and w_{ij} measures the number of links from university i to j . In holdout comparisons our kriging method is more accurate than the random walk smoother, which ends up being quite similar to a

linear regression on $\sqrt{\pi_i}$ values without an intercept. Section 4.4.2 presents a binary example, the WebKB dataset, where the response is 1 for student web pages and -1 otherwise. Incorporating empirical correlations into the semi-supervised learning methods brings a large improvement in the area under the ROC curve.

Section 4.5 describes some simple adaptations of the approach presented here. Section 4.6 discusses some related literature in fields other than machine learning. Section 4.7 has our conclusions.

4.2 A toy example

We start with a simple simulation. Consider a two dimensional 50×50 grid, where each node is connected to its four immediate neighbors (opposite boundaries are glued together so the boundary nodes have four neighbors as well). Assume a Gaussian process $\{Y_i\}_{i=1}^{2500}$ on the grid, where $\text{var}(Y_i) = 1$ and $\text{cov}(Y_i, Y_j) = -1/4$ if i and j are neighbors (denoting $i \sim j$) and 0 otherwise. In other words, we have

$$\mathbf{Y} \sim \mathcal{N}(\mathbf{0}, \Sigma)$$

where

$$\Sigma_{ij} = \begin{cases} 1 & i = j \\ -1/4 & i \sim j \\ 0 & \text{otherwise.} \end{cases}$$

The same construction is considered in Diaconis and Evans [2002]. Figure 4.1 (left) includes the heatmap of one realization of this process, where the cell is colored black if its corresponding Y_i is positive and white otherwise.

We now evaluate on this toy example the performance of the semi-supervised learning methods presented in Section 2.2. We do not consider the manifold smoothing and the spectral transformation methods since they require knowledge of the kernel matrix K and the transformation function $f(\cdot)$. All the other methods have equivalent smoothness measures because the graph is regular with $d_i = 4$ for all nodes. The difference lies in whether the lack of fit is penalized on the entire vector \mathbf{Y}^* (random walk smoothing), only on the observed $\mathbf{Y}^{(0)}$ (Tikhonov smoothing) or infinitely much on the observed $\mathbf{Y}^{(0)}$ (interpolated Tikhonov smoothing).

Figure 4.1 (right) plots the MSE against the tuning parameter λ when 90% of the Y_i 's are held out. The three prediction methods just mentioned are compared against the naive method which predicts using the average of the observed 250 Y_i 's, and the oracle method which is kriging with the true covariance. The results are averaged over 10 trials. As we expect, the oracle kriging has the smallest MSE. However, it

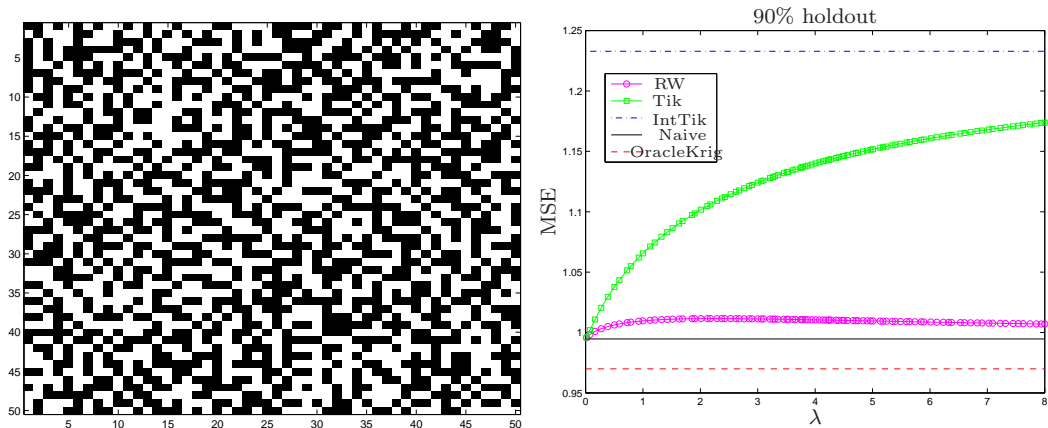


Figure 4.1: Left: heatmap of signs of a realization of the Gaussian process on 50×50 grid. Right: prediction MSE when 90% (2250) of the nodes have missing Y_i .

is the most striking that even the naive method performs better than all three semi-supervised learning methods at all $\lambda > 0$. Moreover, we can compare the gap between the naive and the oracle to that between the naive and the three learning algorithms. This gives a sense of the scale of the errors these algorithms can make by assuming wrong covariance.

In fact, this seemingly strange phenomenon is a result of the fixed kriging covariance these methods implicitly assume. These methods fail in this toy example because the neighboring Y_i 's are negatively correlated, while the models assume only non-negative correlations, following from the lemma below.

Lemma 4.2.1. *Let $L \in \mathbb{R}^{n \times n}$ be any of the following matrices defined in Chapter 2:*

$$\Delta, \quad \tilde{\Delta}, \quad \tilde{\Delta}', \quad \text{or} \quad \gamma \tilde{\Delta}'_H + (1 - \gamma) \tilde{\Delta}'_A.$$

Let $\Lambda \in \mathbb{R}^{n \times n}$ be any diagonal matrix with $\Lambda_{ii} \geq 0$ such that $L + \Lambda$ is positive definite. Then $(L + \Lambda)^{-1} \geq 0$, where \geq denotes element-wise inequality.

Proof. Clearly, all the matrices considered here have non-positive off-diagonals, i.e. $L_{ij} \leq 0$ for $i \neq j$. The desired result now follows directly from Berman and Plemmons [1994] Theorem 2.3. \square

Notice that when $L = \gamma \tilde{\Delta}'_H + (1 - \gamma) \tilde{\Delta}'_A$ is itself positive definite, it follows from the above lemma that L^{-1} , the implied signal covariance for hub-and-authority smoothing, is non-negative element-wise and hence the correlations are all non-negative. In the other cases where L is not invertible, the implied kriging covariances involve taking limit as $\delta \rightarrow \infty$. This makes it harder to see the implied positive correlations. However, if we turn to the predicting equation in (3.12), we can notice that if we write

the prediction $\widehat{Y}_i = \sum_{j=1}^n \alpha_{ij} Y_j^*$, then Lemma 4.2.1 above immediately gives that the α_{ij} 's are all non-negative.

Even in this simple example, we see the potential problem of assuming *a priori* a fixed covariance matrix. In the following sections, we will propose a data driven estimator of the correlation structure and show that we are able to obtain much improved prediction on some real datasets.

4.3 Stationary correlations

We start by recalling the kriging framework presented in Section 3.2.1. The observation \mathbf{Y} is comprised of the underlying signal \mathbf{Z} and the noise $\boldsymbol{\varepsilon}$, both of which are normally distributed. Under both the plain kriging model and the variation with a prior on the coefficient, the model for the signal can be written as

$$\mathbf{Z} \sim \mathcal{N}(bX, \sigma^2VRV), \quad (4.1)$$

where the covariance is decomposed into a correlation matrix $R \in \mathbb{R}^{n \times n}$, a diagonal matrix $V = \text{diag}(v_1, \dots, v_n)$ containing known relative standard deviations $v_i > 0$, and a scale parameter $\sigma > 0$. The design matrix X has one row for each node of the graph. It can include a column of ones for an intercept, columns for other predictors constructed from the graph adjacency matrix W , and columns for other covariates measured at the nodes. To emphasize the role of the graph structure, we only use covariates derived from W . Here we take $X \in \mathbb{R}^n$, so b is a scalar.

Model (4.1) includes both random walk regularization (Section 3.3.1) and the Tikhonov regularization (Section 3.3.2) as special cases. The connections are made in Table 4.1, where we use the fact that $d'_i = \sum_j w'_{ij} = \pi_i$ to write $\widetilde{\Delta}'^+ = \Pi^{1/2} \Delta'^+ \Pi^{1/2}$ with $\Pi = \text{diag}(\pi_1, \dots, \pi_n)$.

	X	\mathbf{v}	$\sigma^2 R_{ij}$
Random walk:	$\sqrt{\boldsymbol{\pi}}$	$\sqrt{\boldsymbol{\pi}}$	$\Delta'_{ij}{}^+ + \delta$
Tikhonov:	$\mathbf{1}_n$	$\mathbf{1}_n$	$\Delta'_{ij}{}^+ + \delta$

Table 4.1: Parameters chosen for model (4.1) to obtain the random walk smoothing and the Tikhonov smoothing methods. Both models use the limit $\delta \rightarrow \infty$.

The key element in (4.1) is the correlation matrix R . All of the methods summarized in Table 3.2 take R to be a fixed matrix given by the graph adjacency matrix, via Δ , Δ' and related quantities. Our model for R_{ij} is $\rho(s_{ij})$ where ρ is a smooth function to be estimated using the response values, and s_{ij} is a measure of graph similarity

between nodes i and j . For instance, the random walk smoothing defines similarity to be $s_{ij} = w'_{ij} = (\pi_i P_{ij} + \pi_j P_{ji})/2$, while the Tikhonov smoothing uses $s_{ij} = w_{ij}$. These correlations are stationary in s , by which we mean that two node pairs ij and $i'j'$ from different parts of the graph are thought to have the same correlation, if $s_{ij} = s_{i'j'}$. The standard deviations σv_i , by contrast, are proportional to given numbers that need not be stationary with respect to any feature of the nodes. The signal means bX need not be stationary either. The estimation procedure, including measures to make R positive semidefinite, is described in detail in Section 4.3.1 below.

Like these regularization methods, we assume in model (4.1) that X , \mathbf{v} and s_{ij} are prespecified based on domain knowledge. We take the noise variance to be $\lambda^{-1}I$, like the random walk does, but unlike the Tikhonov method, which uses effectively infinite variance for the unmeasured responses.

In matrix notation, our prediction is

$$\widehat{\mathbf{Z}} = \Sigma_{\bullet 0}(\Sigma_{00} + \lambda^{-1}I)^{-1}(\mathbf{y}^{(0)} - bX^{(0)}) + bX, \quad (4.2)$$

where $\Sigma = \sigma^2 V R V$ with the estimate R described in next section. Two things we would like to point out. First, our predictor (4.2) is based on the kriging model without the Bayesian prior. As established in Lemma 3.2.1, the kriging predictor with an improper prior is equivalent to having the coefficient estimated with the generalized least square estimator. Second, we do not plug in a guess for the unobserved $\mathbf{Y}^{(1)}$ to accord with common statistical practice.

Finally, looking ahead to the empirical experiments in Section 4.4, when we compare to the random walk method, we will use $s_{ij} = w'_{ij}$. Similarly, when we compare to the Tikhonov regularized method, we will use $s_{ij} = w_{ij}$, or symmetrized to $w_{ij} + w_{ji}$ if the graph is directed. In this way we will use the exact same similarity measures as those methods do. There is one additional subtlety. We found it more natural to make the correlation a smooth function of similarity. For the other methods it is the inverse of the correlation that is smooth in s_{ij} .

4.3.1 Covariance estimation through the variogram

Here we adapt the variogram-based approach from Geostatistics (see for example Cressie [1993]) to estimate the matrix R . The approach we consider here has a similar flavor as the nonparametric correlation estimation briefly discussed in Section 3.2.2.

With noise variance $\lambda^{-1}I$, recall that the variogram of the model (4.1) is

$$\begin{aligned} \Phi_{ij} &\equiv \frac{1}{2} \mathbb{E}((Y_i - bX_i) - (Y_j - bX_j))^2 \\ &= \lambda^{-1} + \frac{1}{2} \sigma^2 (v_i^2 + v_j^2 - 2v_i v_j R_{ij}). \end{aligned} \quad (4.3)$$

For $1 \leq i, j \leq r$ both $Y_i = y_i$ and $Y_j = y_j$ are observed and so we have the naive estimator

$$\widehat{\Phi}_{ij} = \frac{1}{2}((y_i - bX_i) - (y_j - bX_j))^2. \quad (4.4)$$

The naive variogram is our starting point. We translate it into a naive value \widehat{R}_{ij} by solving equation (4.3). This requires the prespecified values of v_i and v_j . We also need values for λ and σ , which we consider fixed for now and will discuss how to choose them later.

Once we have the naive correlation estimates \widehat{R}_{ij} we use a spline smoother to fit the smooth function $\widehat{R}_{ij} \doteq \widehat{\rho}(s_{ij})$. Smoothing serves two purposes. It yields correlation as a function of similarity s_{ij} , and it reduces sampling fluctuations. Next we use $\widehat{\rho}$ to estimate the entire correlation matrix via $\widetilde{R}_{ij} = \widehat{\rho}(s_{ij})$ for $i \neq j$ with of course $\widetilde{R}_{ii} = 1$. To complete our estimation of the signal variance we take $\widehat{\Sigma} = \sigma^2 V \widetilde{R} V$, and then if necessary modify $\widehat{\Sigma}$ to be positive semi-definite. Two versions of the last step are considered. One is to use $\widehat{\Sigma}_+$, the positive semi-definite matrix that is closest to $\widehat{\Sigma}$ in Frobenius norm (see Appendix B.1). The other method is to use a low rank version of $\widehat{\Sigma}_+$ to save computational cost.

The step-by-step procedure to estimate the signal covariance is listed in Table 4.2. The output is the estimated Σ , which we use through equation (4.2) to make predictions.

We choose σ and λ by cross-validation. This is the same technique used by the semi-supervised methods discussed in Section 2.2. It is also similar to treatment of the shrinkage parameter used in ridge regression.

In our cross-validation, some known labels are temporarily treated as unknown and then predicted after fitting to the other labels. The entire graph structure is retained, as that mimics the original prediction problem. When estimating error rates we use training, test, and validation subsets.

4.3.2 Practical issues

As we have discussed, we need to make choices for X , \mathbf{v} and s_{ij} that go into our model (4.1). These prespecified values should come from domain knowledge about the response variable of interest. They may depend on the graph adjacency matrix W , but not on the realization of the variable \mathbf{Y} . The single predictor X corresponds to the direction that \mathbf{Y} varies along and \mathbf{v} the amount of univariate variations. The similarity s_{ij} defines the closeness of nodes i and j . There are clearly many possible choices for these parameter values, and we don't yet have any guidance on how best to select them for a specific problem.

Variance estimation with stationary correlations

Given $\lambda > 0$ and $\sigma > 0$:

1. For every pair of observed nodes $i, j = 1, \dots, r$ and $i \neq j$, estimate R_{ij} by solving (4.3) with Φ_{ij} estimated using (4.4):

$$\widehat{R}_{ij} = \frac{\sigma^2(v_i^2 + v_j^2)/2 + \lambda^{-1} - \widehat{\Phi}_{ij}}{\sigma^2 v_i v_j}. \quad (4.5)$$

2. Smooth the pairs $\{(\widehat{R}_{ij}, s_{ij}) : i, j = 1, \dots, r\}$ to obtain the estimated correlation function $\widehat{\rho}(\cdot)$.
3. Compute $\widetilde{R}_{ij} = \widehat{\rho}(s_{ij})$ for $i \neq j$ and $\widetilde{R}_{ii} = 1$.
4. Set $\widehat{\Sigma} = \sigma^2 V \widetilde{R} V$.
5. Use one of the following two methods to make $\widehat{\Sigma}$ positive semi-definite. Let $\widehat{\Sigma} = U H U^T$ be the eigen-decomposition of $\widehat{\Sigma}$. Then
 - (a) use $\widehat{\Sigma}_+ = U H_+ U^T$, where $H_+ = \max(H, 0)$, or,
 - (b) use $\widehat{\Sigma}_+^{(k)} = U H_+^{(k)} U^T$, where $H_+^{(k)}$ consists of the first k diagonal elements of H_+ and the rest are set to be zero.

Choice (a) gives the positive semi-definite matrix that is closest to $\widehat{\Sigma}$ in Frobenius norm. Choice (b) is used when computational cost is a concern or the true covariance Σ is believed to be low-rank.

Table 4.2: The steps we use to estimate the covariance matrix $\Sigma = \sigma^2 V R V$ in model (4.1) via an empirical stationary correlation model.

The connections to the random walk and the Tikhonov smoothing methods present two sets of example choices, as listed in Table 4.1. While \mathbf{v} and X can be set separately, both methods take $\mathbf{v} = X$, and so signals Z_i with a larger absolute mean $|bX_i|$ also have a larger variance $\sigma^2 X_i^2$. This seems reasonable but of course some data will be better fit by other relationships between mean and variance. One appealing feature of choosing $\mathbf{v} = X$ is that (4.1) has a simple interpretation that the scaled signals Z_i/X_i are Gaussian with constant mean, constant variance, and stationary correlation R . We will compare the two sets of choices using real examples in Section 4.4 below.

It is worthwhile to point out that for unweighted graphs, the Tikhonov smoothing method leads to very few distinct values of s_{ij} . In this case, we simply use the average of \widehat{R}_{ij} for each distinct s_{ij} to approximate $\hat{\rho}(\cdot)$ without smoothing. For choices that lead to many distinct values of s_{ij} , cubic splines with ten knots are used to get $\hat{\rho}$ out of convenience. Better choices of smoothing method could probably be made, but we expect the differences among adaptive correlation methods to be smaller than those between methods with fixed correlations and methods with simple adaptive correlations.

Finally, all measurement errors have the same variance λ^{-1} in our model. It is unnecessary to assume a different noise variance for the unobserved Y because their variance does not affect our predictor (4.2).

4.3.3 Relation to Geostatistics

We use a nonparametric estimate $\rho(\cdot)$ to avoid forcing a parametric shape on the correlation function. The parametric curves used in the Geostatistics literature for \mathbb{R}^d with small d may not extend naturally to graphs, even if they could be properly embedded in Euclidean space. We have also seen in Section 3.2.2 that when the dimension is high, which is usually the case for graphs, the parametric families essentially allow only non-negative correlations.

Both Hall et al. [1994] and Shapiro and Botha [1991] have discussed ways to fit a nonparametric variogram while ensuring a positive semi-definite covariance (see Section 3.2.2). Their techniques apply when the predictor space is \mathbb{R}^d . The usual definition of the similarity measure on a graph is far from being a metric in \mathbb{R}^d . Our approach ensures that the estimate for Σ is positive semi-definite.

When there are n observations, Hall et al. [1994] find convergence rates for the smoother $\hat{\rho}(\cdot)$ that are comparable to that using n^2 observations. The reason is that we get $O(n^2)$ pairs (Y_i, Y_j) in the empirical variogram. In our application there are only $r(r-1)/2$ observed pairs to use.

In the spatial smoothing problems where kriging originates, it is often necessary for the covariance to remain semi-definite at any finite list of points in \mathbb{R}^d , including

some that are not yet observed. Our setting does not require us to construct an extension of the covariance function to Y_i for nodes i that are not in the graph. Even in cross-validation, we know the positions in the graph for the points at which no response values have been observed, and so we can still compute s_{ij} for all data pairs. This aspect of the semi-supervised setting makes the problem much simpler than that faced in Geostatistics. It does however mean that when the graph changes, the covariance model may have to be recomputed.

4.4 Examples

In this section, we compare our empirical covariance approach to the random walk and the Tikhonov regularization methods. We use two extremely different real datasets. The first one has a continuous response on a dense, weighted graph, and the second one has a binary response on a sparse, unweighted graph. Because both graphs are directed, we construct an undirected graph for the Tikhonov approach using $W + W^T$ as the adjacency matrix. Our empirical-based method, together with its low rank variations, brings substantial improvements for both methods on both datasets.

Recall that we need to prespecify the values of X , \mathbf{v} and s_{ij} for our empirical covariance approach. Following the discussion in Section 4.3.2, we consider two versions of our model. One follows the choices of the random walk method and the other follows the Tikhonov method, which we call “empirical random walk” and “empirical Tikhonov” respectively, compared to the “original random walk” and the “original Tikhonov”. Therefore, the comparisons using real data serve two purposes. First, the comparison between the empirical and the original shows how performance changes when we incorporate empirical stationary correlations. Second, the comparison between the two original (or empirical) methods shows how the choices of the prespecified parameters affect the predictions.

We also need an estimate of the coefficient b in the mean. For binary problems with $Y_i \in \{-1, 1\}$ we take $b = 0$, as is done in the machine learning literature. For continuous responses we use

$$\hat{b} = \frac{1}{r} \sum_{i=1}^r \frac{y_i}{X_i}. \quad (4.6)$$

We also investigated estimating b by generalized least squares regression of $\mathbf{Y}^{(0)}$ on $X^{(0)}$ taking account of estimated correlations among the first r response values. This made only a very small difference even on the small problems we are about to report, and so we see no reason to prefer it to the very simple estimate (4.6). We do want to point out that estimating b is necessary for the empirical methods and the original

random walk method, but not necessary for the original Tikhonov method. This is because even though b is used in the construction of \mathbf{Y}^* , it disappears from \mathbf{Y}^* in the $\lambda_1 \rightarrow 0$ limit for the original Tikhonov method.

4.4.1 UK university web link dataset

Data description

The university dataset contains the number of web links between UK universities in 2002. Each university is associated with a research score (RAE), which measures the quality of the university's research¹. After removing four universities that have missing RAE scores, or that have no in-link or out-link, there are 107 universities.

The response variable, RAE score, is continuous and ranges from 0.4 to 6.5 with a mean of 3.0 and a variance of 3.5. The number of links from one university to another forms the (asymmetric) weighted adjacency matrix W . The distribution of the weights w_{ij} is heavily right tailed and approximately follows a power law. About 15% of the weights are zero, and 50% of them are less than 7, while the maximum is 2130.

Illustration of the empirical covariance method

We first use the entire dataset to illustrate the empirical variance estimation procedure as given in Table 4.2. For simplicity, we illustrate only the empirical Tikhonov method and hence use $\mathbf{v} = X = \mathbf{1}_n$ and $s_{ij} = w_{ij} + w_{ji}$. These similarity scores take many values, and so we use correlation smoothing. The empirical random walk method is similar. In practice, σ^2 and λ are chosen by cross-validation, but we fix $\sigma^2 = 5$ and $\lambda^{-1} = 0.01$ here to show one iteration of the estimation procedure.

Figure 4.2 (left) plots the naive estimates \widehat{R}_{ij} , as computed in (4.5), against (log transformed) similarity s_{ij} values. The logarithm is used because the s_{ij} are skewed. The scatter plot is very noisy, but we can nonetheless extract a non-trivial $\hat{\rho}(\cdot)$ with cubic spline smoothing (ten knots), as shown by the red curve. The same curve is also included on the right plot at a larger scale.

It is striking that $\hat{\rho}(\cdot)$ is not monotonically increasing in s_{ij} . The greatest correlations arise for very similar nodes, but the very least similar node pairs also have somewhat more correlation than do pairs with intermediate similarity. Recall that a similarity of 0 means that the pair of universities are not linked. Universities without many links are over represented in such pairs, and those universities tend to have similar (low) RAE scores.

¹The data are at <http://cybermetrics.wlv.ac.uk/database/stats/data/>. We use the link counts at the directory level.

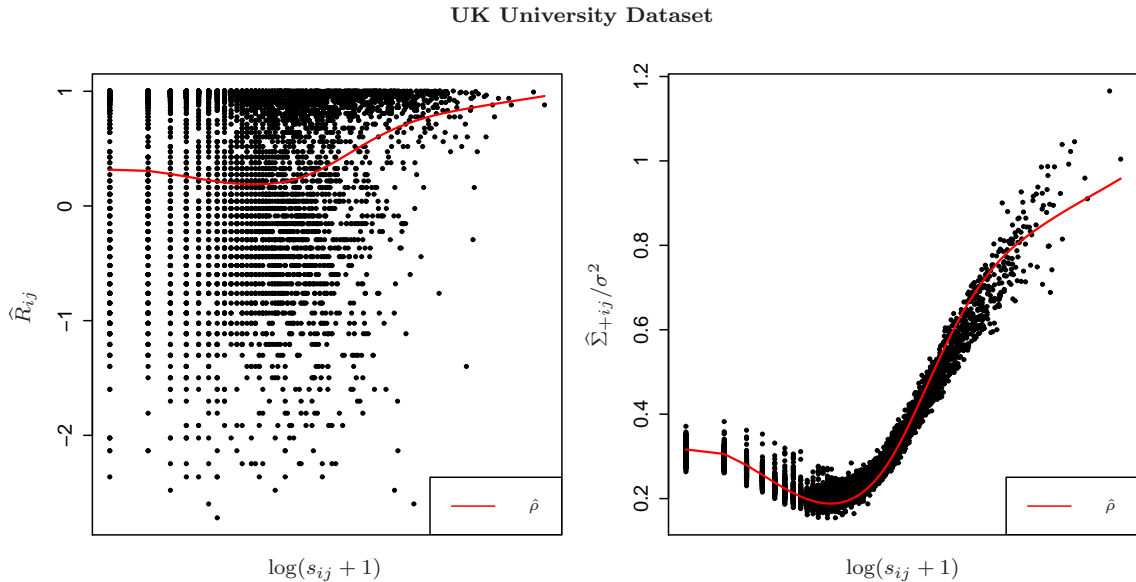


Figure 4.2: Illustration of the empirical Tikhonov method with the UK university data. Left: scatter plot of the naive \widehat{R}_{ij} values versus $\log(s_{ij} + 1)$ with the cubic spline smoothing curve (red). Right: final estimates $\widehat{\Sigma}_{+ij}/\sigma^2$ versus $\log(s_{ij} + 1)$ with the same smoothing curve (red).

The final step in Table 4.2, is to make the covariance matrix $\widehat{\Sigma}$ that directly results from $\widehat{\rho}(\cdot)$ positive semi-definite. For the full rank version $\widehat{\Sigma}_+$, we plot points $\widehat{\Sigma}_+/\sigma^2$ on the right side of Figure 4.2. These scatter around the red curve which shows $\widehat{\rho}(\cdot)$. We saw similar patterns (not shown here) with some low rank estimates $\widehat{\Sigma}_+^{(k)}$. During this final step, we saw in Figure 4.2 (right) that, a small number of highly similar node pairs got the greatest change in model correlation. That pattern did not always arise in other examples we looked at.

Performance comparisons

Now we turn to performance comparisons. For this, we hold out the RAE scores of some universities and measure each prediction method by mean squared error (MSE) on the held out scores. The size of the holdout set ranges from approximately 10% to 90% of the entire dataset, and 50 trials are done at each holdout level.

Our empirical methods have two tuning parameters λ and σ while the original random walk and Tikhonov methods have only one. Nevertheless, the comparison is fair because it is based on hold out sets. For each set of held-out data we used ten-fold cross-validation within the held-in data to pick λ and σ for empirical stationary correlation kriging. For the original random walk and Tikhonov methods we use the

best tuning parameter (λ), and so our comparisons are to somewhat better versions of the random walk and Tikhonov method than one could actually get in practice.

We define a baseline method that considers no signal covariance, and simply regresses the responses Y_i on X_i . With the random walk choice of $X_i = \sqrt{\pi_i}$, the baseline prediction is $\hat{b}\sqrt{\pi_i}$, while with the Tikhonov choice of $X_i = 1$, it is simply \hat{b} .

The results are shown in Figure 4.3. The random walk method performs quite well compared to the Tikhonov method, but neither of them outperform their corresponding baseline methods by much, even with the *best* tuning parameters. The black and red curves track each other closely over a wide range of data holdout sizes, with the red (graph-based) curve just slightly lower than the black (baseline) curve.

The results show that the random walk choices $\mathbf{v} = X = \sqrt{\boldsymbol{\pi}}$ and $s_{ij} = w'_{ij}$ are clearly better than the Tikhonov choices $\mathbf{v} = X = \mathbf{1}_n$ and $s_{ij} = w_{ij} + w_{ji}$ for the UK university data. Another difference between the methods is that the Tikhonov method symmetrizes the graph. As such, it does not distinguish between links from University i to j and links in the other direction. Even the baseline for the random walk method, which does regression on $\sqrt{\boldsymbol{\pi}}$, makes use of the directionality because that directionality is reflected within $\boldsymbol{\pi}$.

The green curves in Figure 4.3 show the error rates for the two versions of the empirical stationary correlation method. They generally bring large performance improvements, except at the very highest holdout levels for the Tikhonov case. Then as few as 17 University scores are being used and while this is probably too few to estimate a good covariance, it does not do much harm either. All the methods do better when less data are held out. The methods with data driven correlations have slightly steeper performance curves.

We make a numerical summary of the curves from Figure 4.3 in Table 4.3. We compare performance for the setting where about half of the data are held out. For both cases, kriging with empirical stationary correlations typically brings quite large improvements over the original methods. Low rank variations of empirical stationary correlation kriging perform similarly to the full rank empirical method, except for the rank 1 case in the random walk setting. There we still see a large improvement but not as much as for the full rank or rank 5 cases. The good performance of the low rank versions could reflect a small number of latent effects, or the benefits of regularization.

4.4.2 WebKB dataset

The WebKB dataset² contains webpages collected from computer science departments of various universities in January 1997. The pages were manually classified into seven categories: student, faculty, staff, department, course, project and other. The dataset

²The data are at <http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html>.

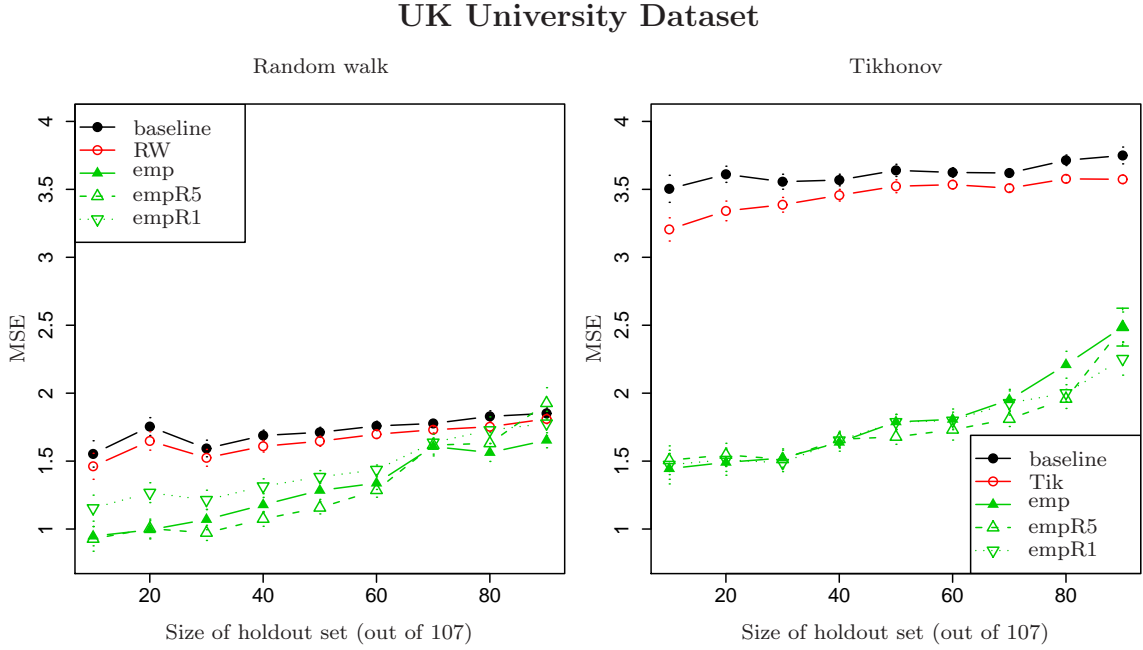


Figure 4.3: MSEs for the RAE scores at different holdout sizes. Left: the original random walk (red) compared with our empirical random walk (green). Right: the original Tikhonov (red) compared with our empirical Tikhonov (green). Baseline methods (black) are described in the text.

Improvement over baseline		
	Random walk	Tikhonov
Baseline MSE	1.71	3.64
Original random walk	3.8%	-
Original Tikhonov	-	3.2%
Empirical	25.0%	50.9%
Empirical R5	32.4%	53.9%
Empirical R1	19.1%	50.9%

Table 4.3: The relative improvement over baseline when 50 of 107 ARE scores are held out. The baseline methods are simple regressions through the origin on $X = \sqrt{\pi}$ (random walk) and on $X = \mathbf{1}_n$ (Tikhonov).

we have is a subset, where the webpages belonging to the “other” class are removed. We will only use the data for Cornell University, which has 195 webpages and 301 links, after removing the three self loops. We further reduce the webpage labels to be “student” (1) and “non-student” (−1). There are 83 student pages in total. The adjacency matrix is unweighted, i.e., w_{ij} is 1 if there is a link from page i to j and 0 otherwise. Again, the links are directed and hence W is asymmetric, with 99.2% of the w_{ij} being zero.

The kriging models make continuous predictions of the binary response. We use the area under the ROC curve (AUC) to measure performance on the holdout sets. The AUC is equivalent to the probability that a positive label will get a higher prediction than a negative label. To estimate the correlation function in the empirical based method, we again use cubic splines with ten knots for the random walk s_{ij} . However, for the Tikhonov s_{ij} , which has only three possible values 0, 1 and 2 in an unweighted directed graph, we simply use the average at each s_{ij} without smoothing. The tuning parameters are picked in the same way as for the university dataset.

The results are plotted in Figure 4.4 and summarized in Table 4.4. As a baseline, we consider a model which sorts the web pages in random order. It would have an AUC of 0.5. For the WebKB data, the Tikhonov method has better accuracy than the random walk method which actually has trouble getting an AUC below 0.5. It is interesting that in this case the method which ignores link directionality does better. In both cases empirical stationary correlations bring large improvements. As before we see that larger amounts of missing data make for harder prediction problems.

4.5 Variations

In many applications, we may want to use more nuanced error variance measures, such as $\Gamma = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$ and this fits easily into the kriging framework. For example, web pages determined to be spam after a careful examination could be given a smaller σ_i^2 than those given less scrutiny, and those not investigated at all can be given a still higher σ_i^2 .

Sometimes we can make use of an asymmetry in the labels. For example, positive determinations, e.g. 1s, may have intrinsically higher confidence than negative determinations, −1s, and we can vary σ_i to account for this. Similarly, when one binary label in ± 1 is relatively rare, we could use a value other than 0 as our default guess.

We have only considered $X \in \mathbb{R}^n$ constructed from the graph adjacency matrix W . It is likely that there are other covariates measured at the nodes. In that case, a simple fix would be to regress \mathbf{Y} on these covariates and then treat the residuals as the “new” responses to feed to the algorithms.

Finally, it is not necessary to have $\mathbf{v} = X$, where \mathbf{v} appears in the variance model through $\sigma^2 V R V$ with $V = \text{diag}(\mathbf{v})$ and X in the model for the mean through bX .

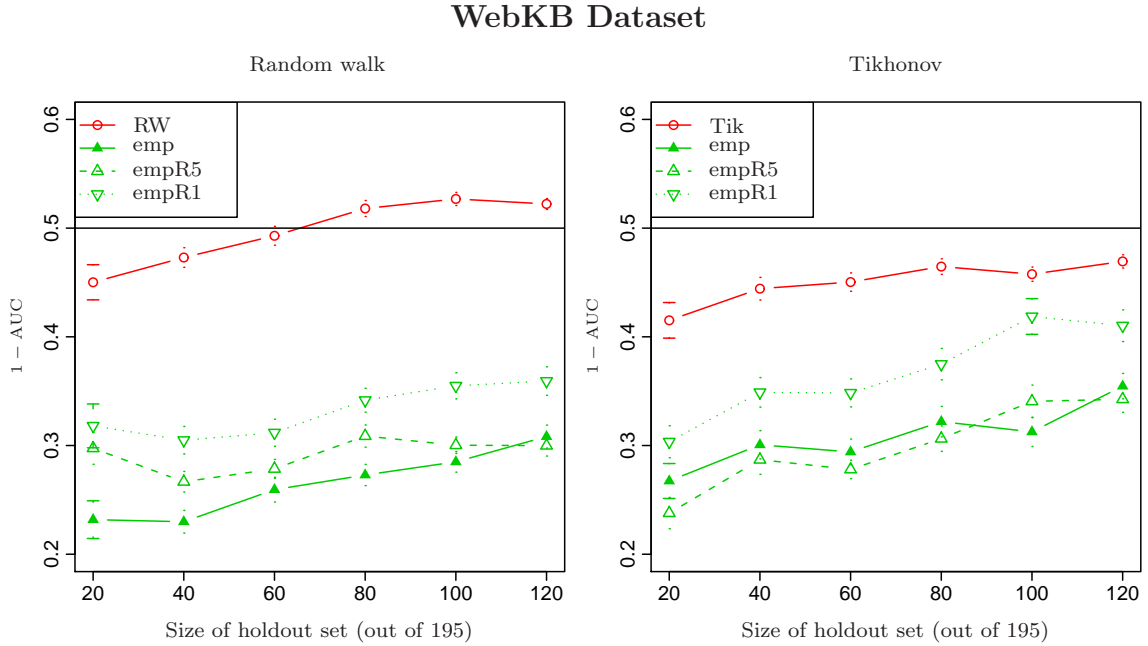


Figure 4.4: Classification error for webpage labels at different holdout sizes, measured with 1 minus the area under the ROC curve. Left: the original random walk (red) compared with our empirical random walk (green). Right: the original Tikhonov (red) compared with our empirical Tikhonov (green). The baseline method is random guessing.

Improvement over baseline		
	Random walk	Tikhonov
Baseline (1-AUC)	0.5	0.5
Original random walk	-5.4%	-
Original Tikhonov	-	8.5%
Empirical	43.0%	37.5%
Empirical R5	40.0%	31.9%
Empirical R1	29.0%	16.3%

Table 4.4: The relative improvement over baseline when 100 out of 195 webpage labels are held out. The baseline AUC is 0.5.

We use $\mathbf{v} = X$ in the examples in Section 4.4 because this is the choice of the random walk and the Tikhonov methods. Also, we could hybridize the Tikhonov and random walk models, using $\mathbf{v} = X = \mathbf{1}_n$ from the former inside the regression model with the edge directionality respecting covariance of the latter.

4.6 Other related literature

We have so far focused on the graph-based prediction methods from the machine learning literature. We would like to point out a few related works in some other fields as well.

In the social network literature, researchers have built network autocorrelation models to examine social influence process. For more details see, for example, Leenders [2002] and Marsden and Friedkin [1993]. A typical model is as follows:

$$\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\omega}, \quad \boldsymbol{\omega} = \alpha B\boldsymbol{\omega} + \boldsymbol{\varepsilon}, \quad (4.7)$$

where α is the network autocorrelation parameter, B is a weight matrix and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I)$. This model is mainly used for estimating or testing α and $\boldsymbol{\beta}$, but we could of course use it for prediction purpose as well. Notice that we can write model (4.7) as

$$\mathbf{Y} \sim \mathcal{N}(X\boldsymbol{\beta}, \sigma^2 AA^T),$$

where $A = (I - \alpha B)^{-1}$. Comparing to the other models we have discussed so far, \mathbf{Y} here is no longer a noisy measurements of some underlying quantity \mathbf{Z} . The covariance $\sigma^2 AA^T$ depends on a scaled weight matrix αB . Leenders [2002] discusses a few ways to construct the weight matrix B , but all of them involve only the graph adjacency matrix and some a priori quantities. Nevertheless, the autocorrelation scale α , which is estimated from data, can incorporate some empirical dependence from the observed Y .

Heaton and Silverman [2008] consider prediction at unobserved sites in \mathbb{R}^1 or \mathbb{R}^2 . The underlying function \mathbf{Z} is assumed to have a sparse wavelet expansion, which they utilize within an MCMC framework to generate a posterior distribution for the unobserved \mathbf{Y} . Their method is shown to have better performance in neighborhoods containing discontinuities where other methods, e.g. kriging, would smooth. While this method applies to data in Euclidean space with the regular wavelet transform, Jansen et al. [2009] discuss a potential extension to data arising on graphs using the wavelet-like transform they introduce.

Finally, Hoff et al. [2002] model the relational tie between a pair of nodes in a social network by introducing a latent position for each node in a low dimensional Euclidean space. Handcock et al. [2007] then propose a(n) (unsupervised) clustering

method by assuming these latent positions arise from a mixture of distributions, each corresponding to a cluster. Of course, we can also see the potential to utilize these latent positions in Euclidean space kriging methods to make predictions. Along the same line, we could also use multidimensional scaling to map the graph to a low dimensional Euclidean space where kriging naturally applies.

4.7 Conclusion

We have shown in Chapter 3 that several recently developed semi-supervised learning methods for data on graphs can be expressed in terms of kriging. This chapter is then motivated by the observation that those kriging models use implied correlations that derive from the graph structure but do not take account of sample correlations among the observed values.

Our proposed empirical stationary correlation model uses correlation patterns seen among the observed values to estimate a covariance matrix over the entire graph. In two numerical examples we saw that using empirical correlations brought large improvements in performance. Even when there were large differences between the performance levels of different semi-supervised methods, the use of empirical correlations narrowed the gap. This reduces the penalty for the user who makes a suboptimal choice for X , \mathbf{v} and s_{ij} .

The stationary correlation model was motivated by the idea that the correlations should be some unknown monotone function of similarity, and that given enough data, we could approximate that function. We were mildly surprised to see a non-monotone relationship emerge in our first example, though it was interpretable with hindsight. We do not have a way to test models of this kind, beyond using cross-validation to choose between two of them.

We have not considered large scale problems in this chapter. We defer this topic to Chapter 5 coming up next. In our examples covariance estimates derived from quite small numbers of observation pairs still performed well. We finish by pointing out that there are a good many smaller datasets to which semi-supervised learning on graphs may be applied.

Chapter 5

Scale to large graphs

5.1 Introduction

We have described several recent semi-supervised learning graph methods in Chapter 2 and proposed our own prediction method in Chapter 4 that uses kriging with empirical stationary correlations. In this chapter, we focus on implementation of these algorithms on large scale graphs. It is natural to consider massive graphs, as they appear in many real-world applications. For instance, Facebook social network has more than 400 million users¹ while the World Wide Web (WWW) is estimated to have more than 10 billion webpages², both of which are still rapidly growing. Even on a “smaller” scale, the entire human gene-gene interaction network contains more than 20,000 nodes [International Human Genome Sequencing Consortium, 2004].

In order to scale up these prediction algorithms, we need to understand where the computational challenges are for each of them. The optimization framework shared by the methods reviewed in Chapter 2 is different from the statistical modeling approach we take in our empirical correlation method (Chapter 4), even though both can be framed under kriging. As a result, they require very different strategies when it comes to scaling. We will treat them separately in Section 5.2 and 5.3 of this Chapter with a discussion of connections in Section 5.5.

Recall that in Chapter 3 we summarized the reviewed graph-based learning algorithms into a linear system with the following solution

$$\hat{\mathbf{Y}} = (L + \Lambda)^{-1} \Lambda \mathbf{Y}^*, \quad (5.1)$$

where L is the smoothing matrix and Λ is the diagonal matrix penalizing lack-of-fit.

¹<http://www.facebook.com/press/info.php?statistics>

²<http://www.worldwidewebsize.com/>

Solving this system requires inverting an $n \times n$ matrix, which usually has a computational complexity of $\mathcal{O}(n^3)$. Fortunately, the matrix to be inverted, $L + \Lambda$, is sparse in most cases. This is because many real graphs, particularly real large graphs, are very sparse, where the total number of edges in the graph grows linearly with n . Therefore, the smoothing matrix L , when comprised of only the graph Laplacian, retains the same sparsity. The two exceptions, the manifold smoothing and the spectral transformation methods, are not discussed here as they require extra knowledge of the kernel and the transformation function.

In general, there are two families of strategies toward solving (5.1). The first is to devise a linear system solver that produces a numerical solution within ϵ distance of the exact solution. Methods in this category are usually proposed for a more general context. For instance, Spielman and Teng [2003] presented a solver for positive semidefinite and diagonally dominant matrix (PSDDD), in which case the solution can be found in $\mathcal{O}(|\mathcal{E}|^{1.31})$ steps where $|\mathcal{E}|$ is the total number of non-zero entries in the matrix (equivalent to the size of edge set in the graph setting). Because the matrix $L + \Lambda$ is PSDDD, we can achieve the same complexity for solving (5.1).

The second strategy is specific to the problem of graph-based semi-supervised learning, where an approximation of the original problem is used to speed up the algorithm. For example, Tsang and Kwok [2006] introduce the ϵ -insensitive loss to “sparsify” the regularization, by replacing $\sum_{i,j} w_{ij}(Z_i - Z_j)^2$ with $\sum_{i,j} w_{ij}|Z_i - Z_j|_\epsilon^2$, where $|x|_\epsilon = \max(|x| - \epsilon, 0)$. On the other hand, Delalleau et al. [2006] keep the original regularization intact, but approximate many of the unobserved responses with a (prespecified) linear combination of other responses. This reduces the total number of optimization variables and hence reduces the complexity. One drawback of these approximating proposals is that, the loss of prediction accuracy due to the approximation is usually not well studied or understood.

We now turn to our empirical kriging method proposed in Chapter 4, where the predictions are

$$\widehat{\mathbf{Z}} = \Sigma_{\bullet,0}(\Sigma_{00} + \lambda^{-1}I)^{-1}(\mathbf{y}^{(0)} - bX^{(0)}) + bX. \quad (5.2)$$

Recall that $\Sigma_{00} \in \mathbb{R}^{r \times r}$ is the covariance for the r observed responses with r reasonably small in many real applications. Therefore, the computational challenge here is not the matrix inversion, but the estimation of a positive semi-definite covariance Σ . To obtain a valid covariance matrix, our method truncates the SVD of a naive estimate of Σ , and SVD usually takes $\mathcal{O}(n^3)$.

Having presented the challenges to scale these prediction algorithms for large graphs and reviewed some existing solutions, we propose in Section 5.2 a way to approximate (5.1) using Markov chain simulations which works naturally on graphs of very large scale. In Section 5.3 we show it is possible to obtain a $\mathcal{O}(n)$ linear time

estimate of the covariance matrix for our empirical kriging method by applying some well-developed tools from numerical analysis. Section 5.4 then runs these algorithms on a Wikipedia graph with about 2.4 million nodes. Finally, we conclude in Section 5.5 with a discussion on possible extensions.

5.2 Markov chain algorithms

We propose here a different strategy to solve the predictors in the form of (5.1). First, notice that because (5.1) is a linear predictor, we can write the prediction at node i as a weighted sum of all the imputed responses

$$\hat{Y}_i = \sum_{j=1}^n \alpha_{ij} Y_j^*. \quad (5.3)$$

In this section, we show that we can obtain these weights α_{ij} by constructing a Markov chain for each of the random walk smoothing, Tikhonov smoothing, interpolated Tikhonov smoothing and undirected random walk smoothing methods presented in Section 2.2. We start by establishing the theoretical equivalence between these methods and their corresponding Markov chain constructions, and then at the end of this section, we include a discussion of their implications and also the advantages of approximating using Markov chain simulations.

5.2.1 Theoretical equivalence

The following lemma states that we can construct a Markov chain such that the random walk weight α_{ij}^{rw} is proportional to the expected number of times that the chain starting at node i visits node j .

Lemma 5.2.1 (Random walk smoothing). *Let $\{\tilde{X}_t\}_{t \geq 0}$ be a discrete time Markov chain on the vertex set \mathcal{V} with the following transitional probability*

$$\tilde{P}_{ij} = \frac{w'_{ij}}{d'_i + \lambda d'_i}, \quad (5.4)$$

where recall that $w'_{ij} = (\pi_i P_{ij} + \pi_j P_{ji})/2$, $d'_i = \sum_j w'_{ij}$ and π_i is the stationary distribution on the graph \mathcal{G} as defined in Section 2.1.2. That is, with probability

$$\gamma = \frac{\lambda}{1 + \lambda},$$

this Markov chain jumps into an absorbing state (i.e. killed) from node i . The random

walk smoothing predicts with weights

$$\alpha_{ij}^{rw} = \frac{\lambda}{1 + \lambda} \frac{\sqrt{d'_i}}{\sqrt{d'_j}} \mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right). \quad (5.5)$$

Proof. Recall from (3.13) that the random walk predictions have weights in the following matrix form

$$\begin{aligned} \alpha^{rw} &= (L + \Lambda)^{-1} \Lambda = \lambda (\tilde{\Delta}' + \lambda I)^{-1} \\ &= \lambda D^{1/2} (D' - W' + \lambda D)^{-1} D^{1/2} \\ &= \lambda D^{1/2} (I - (D' + \lambda D')^{-1} W')^{-1} (D' + \lambda D')^{-1} D^{1/2} \\ &= \frac{\lambda}{1 + \lambda} D^{1/2} (I - \tilde{P})^{-1} D'^{-1} D^{1/2} \\ &= \frac{\lambda}{1 + \lambda} D^{1/2} \left(\sum_{t=0}^{\infty} \tilde{P}^t \right) D'^{-1/2}. \end{aligned} \quad (5.6)$$

The third equality uses the fact that $\tilde{\Delta}' = D'^{-1/2} (D' - W') D'^{-1/2}$. The second to last equality follows from the definition of \tilde{P} in (5.4). The last equality follows because $\{\tilde{X}_t\}_{t \geq 0}$ is transient and hence $\|\tilde{P}\|_2 < 1$ with $\|\cdot\|_2$ denoting the operator norm.

Therefore, for individual terms (5.6) gives

$$\begin{aligned} \alpha_{ij}^{rw} &= \frac{\lambda}{1 + \lambda} \frac{\sqrt{d'_i}}{\sqrt{d'_j}} \sum_{t=0}^{\infty} (\tilde{P}^t)_{ij} \\ &= \frac{\lambda}{1 + \lambda} \frac{\sqrt{d'_i}}{\sqrt{d'_j}} \mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right), \end{aligned}$$

as desired. \square

It easily follows the above lemma that the random walk predictions have the following form:

$$\begin{aligned} \hat{Y}_i^{rw} &= \sum_{j=1}^n \alpha_{ij}^{rw} Y_j^* = \sum_{j=1}^n \frac{\lambda}{1 + \lambda} \frac{\sqrt{d'_i}}{\sqrt{d'_j}} \mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right) Y_j^* \\ &= \mathbb{E}_i \left(\sum_{t=0}^{\infty} \frac{\lambda}{1 + \lambda} \frac{\sqrt{d'_i}}{\sqrt{d'_{\tilde{X}_t}}} Y_{\tilde{X}_t}^* \right). \end{aligned} \quad (5.7)$$

In the lemma below, we show that the Tikhonov weight α_{ij}^{Tik} can also be obtained

in a similar way using a different Markov chain.

Lemma 5.2.2 (Tikhonov smoothing). *Let $\{\tilde{X}_t\}_{t \geq 0}$ be a discrete time Markov chain on the vertex set \mathcal{V} with the following transitional probability*

$$\tilde{P}_{ij} = \frac{w_{ij}}{d_i + \lambda_i}, \quad (5.8)$$

where

$$\lambda_i = \begin{cases} \lambda_0 & i \leq r \\ 0 & \text{otherwise.} \end{cases}$$

In other words, with probability

$$\gamma_i = \frac{\lambda_i}{d_i + \lambda_i},$$

this Markov chain jumps into an absorbing state (i.e. killed) from node i . The Tikhonov smoothing predicts with weights

$$\alpha_{ij}^{Tik} = \frac{\lambda_j}{d_j + \lambda_j} \mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbf{1}_{\{\tilde{X}_t=j\}} \right). \quad (5.9)$$

Proof. Recall from (3.17) that the random walk predictions have weights in the following matrix form

$$\begin{aligned} \alpha^{Tik} &= (L + \Lambda)^{-1} \Lambda = (\Delta + \Lambda_{\lambda_0,0})^{-1} \Lambda_{\lambda_0,0} \\ &= (D - W + \Lambda_{\lambda_0,0})^{-1} \Lambda_{\lambda_0,0} \\ &= (I - (D + \Lambda_{\lambda_0,0})^{-1} W)^{-1} (D + \Lambda_{\lambda_0,0})^{-1} \Lambda_{\lambda_0,0} \\ &= (I - \tilde{P})^{-1} (D + \Lambda_{\lambda_0,0})^{-1} \Lambda_{\lambda_0,0} \\ &= \left(\sum_{t=0}^{\infty} \tilde{P}^t \right) (D + \Lambda_{\lambda_0,0})^{-1} \Lambda_{\lambda_0,0}. \end{aligned} \quad (5.10)$$

The third equality uses the fact that $\Delta = D - W$. The second to last equality follows from the definition of \tilde{P} in (5.8). The last equality follows because $\{\tilde{X}_t\}_{t \geq 0}$ is transient and hence $\|\tilde{P}\|_2 < 1$ with $\|\cdot\|_2$ denoting the operator norm.

Therefore, for individual terms (5.10) gives

$$\begin{aligned}\alpha_{ij}^{Tik} &= \frac{\lambda_j}{d_j + \lambda_j} \sum_{t=0}^{\infty} (\tilde{P}^t)_{ij} \\ &= \frac{\lambda_j}{d_j + \lambda_j} \mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right),\end{aligned}$$

as desired. \square

It easily follows the above lemma that the Tikhonov predictions have the following form:

$$\begin{aligned}\hat{Y}_i^{Tik} &= \sum_{j=1}^n \alpha_{ij}^{Tik} Y_j^* = \sum_{j=1}^n \frac{\lambda_j}{d_j + \lambda_j} \mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right) Y_j^* \\ &= \mathbb{E}_i \left(\sum_{t=0}^{\infty} \frac{\lambda_{\tilde{X}_t}}{d_{\tilde{X}_t} + \lambda_{\tilde{X}_t}} Y_{\tilde{X}_t}^* \right).\end{aligned}\tag{5.11}$$

So far we have shown that we can compute the predictions for the random walk smoothing and Tikhonov smoothing algorithms using Markov chain occupation time. We will see in the following lemma that the Markov chain construction for the interpolated Tikhonov algorithm is very different. Instead of being absorbed randomly, this Markov chain stops when it first hits a known node. The weight α_{ij}^{int} turns out to be the probability that the chain starting at node i stops at node j .

Lemma 5.2.3 (Interpolated Tikhonov smoothing). *Let $\{\tilde{X}_t\}_{t \geq 0}$ be a discrete time Markov chain on the vertex set \mathcal{V} with the following transitional probability*

$$\tilde{P}_{ij} = \frac{w_{ij}}{d_i},\tag{5.12}$$

i.e. $\tilde{P}_{ij} = P_{ij}$ defined in Section 2.1.2 for undirected graphs. Let

$$\tau = \inf\{t \geq 0, \tilde{X}_t \leq r\}$$

be the first time the chain hits a node with known response. The interpolated Tikhonov method predicts with weights

$$\alpha_{ij}^{int} = \mathbb{P}_i \left(\tilde{X}_\tau = j \right).\tag{5.13}$$

Proof. Because the interpolated algorithm leaves the known values unchanged, we

know from Section 3.3.3 that

$$\alpha_{ij}^{int} = \begin{cases} 1 & i \leq r, j = i \\ 0 & i \leq r, j \neq i \\ 0 & i > r, j > r \\ \alpha_{10}^{int}[i - r, j] & i > r, j \leq r, \end{cases}$$

where the $(n - r) \times r$ matrix

$$\alpha_{10}^{int} = -\Delta_{11}^{-1}\Delta_{10},$$

as shown in (3.22) and $[i, j]$ represents the (i, j) th entry of the matrix.

It is easy to check that the weights proposed in equation (5.13) satisfies the first three simple cases. We now check the case when $i > r$ and $j \leq r$. First note that in matrix form

$$\begin{aligned} \alpha_{10}^{int} &= -\Delta_{11}^{-1}\Delta_{10} \\ &= -(D_{11} - W_{11})^{-1}(-W_{10}) \\ &= (I - \tilde{P}_{11})^{-1}\tilde{P}_{10} \\ &= \left(\sum_{t=0}^{\infty} \tilde{P}_{11}^t\right)\tilde{P}_{10}, \end{aligned} \tag{5.14}$$

where we partition D, W and \tilde{P} the same way as Δ to get D_{11}, W_{11} and \tilde{P}_{11} . The second to last equality follows from the definition of the transitional probability in (5.12). The last equality follows because $\|\tilde{P}_{11}\|_2 < \|\tilde{P}\|_2 = 1$.

Define $\{\tilde{X}_t^{(1)}\}_{t \geq 0}$ to be $\{\tilde{X}_t\}_{t \geq 0}$ restricted to nodes $\{i : i > r\}$. In other words, it is the same as $\{\tilde{X}_t\}_{t \geq 0}$ but is killed whenever the chain leaves the unknown nodes. Note that \tilde{P}_{11} is the transitional probability for $\{\tilde{X}_t^{(1)}\}_{t \geq 0}$.

We now examine the following probability for $i > r$ and $j \leq r$.

$$\begin{aligned}
\mathbb{P}_i(\tilde{X}_\tau = j) &= \sum_{t=0}^{\infty} \mathbb{P}_i(\tilde{X}_t = j, \tau = t) \\
&= \sum_{t=1}^{\infty} \mathbb{P}_i(\tilde{X}_t = j, \tau = t) \quad (\text{because } i \neq j) \\
&= \sum_{t=1}^{\infty} \mathbb{P}_i(\tilde{X}_{t'} > r, \forall t' < t \text{ and } \tilde{X}_t = j) \\
&= \sum_{t=1}^{\infty} \sum_{k>r} \mathbb{P}_i(\tilde{X}_{t'} > r, \forall t' < t-1 \text{ and } \tilde{X}_{t-1} = k \text{ and } \tilde{X}_t = j) \\
&= \sum_{t=1}^{\infty} \sum_{k>r} \mathbb{P}_i(\tilde{X}_{t'} > r, \forall t' < t-1 \text{ and } \tilde{X}_{t-1} = k) \mathbb{P}_k(\tilde{X}_1 = j) \\
&= \sum_{t=1}^{\infty} \sum_{k>r} \mathbb{P}_i(\tilde{X}_{t-1}^{(1)} = k) \mathbb{P}_k(\tilde{X}_1 = j) \quad (\text{by definition of } \tilde{X}_t^{(1)}) \\
&= \sum_{t=0}^{\infty} \sum_{k>r} \mathbb{P}_i(\tilde{X}_t^{(1)} = k) \mathbb{P}_k(\tilde{X}_1 = j) \\
&= \sum_{t=0}^{\infty} \tilde{P}_{11}^t \tilde{P}_{10},
\end{aligned}$$

which is the desired result as in (5.14). \square

It easily follows the above lemma that the interpolated Tikhonov predictions have the following form:

$$\hat{Y}_i^{int} = \sum_{j=1}^n \alpha_{ij}^{int} Y_j^* = \sum_{j=1}^n \mathbb{P}_i(\tilde{X}_\tau = j) Y_j^* = \mathbb{E}_i \left(Y_{\tilde{X}_\tau}^* \right) \quad (5.15)$$

As we expect, the undirected random walk smoothing is similar to its directed version as shown below.

Lemma 5.2.4 (Undirected random walk smoothing). *Let $\{\tilde{X}_t\}_{t \geq 0}$ be a discrete time Markov chain on the vertex set \mathcal{V} with the following transitional probability*

$$\tilde{P}_{ij} = \frac{w_{ij}}{d_i + \lambda d_i}. \quad (5.16)$$

That is, with probability

$$\gamma = \frac{\lambda}{1 + \lambda},$$

this Markov chain jumps into an absorbing state (i.e. killed) at every step. The undirected random walk smoothing predicts with weights

$$\alpha_{ij}^{urw} = \frac{\lambda}{1 + \lambda} \frac{\sqrt{d_i}}{\sqrt{d_j}} \mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right). \quad (5.17)$$

Proof. The proof follows from Lemma 5.2.1 after noticing that $w'_{ij} = w_{ij}/\text{vol}(\mathcal{G})$ and $d'_i = d_i/\text{vol}(\mathcal{G})$ for undirected graph. \square

5.2.2 Advantages and implications

The Markov chain constructions discussed so far offer an alternative way to approach these semi-supervised graph learning problems. As we can see from the summary in Table 5.1, for each of the four methods considered, the prediction at node i is the expected value of some function of the chain that starts from node i . Therefore, we can simulate the chain and use the sample averages to estimate the corresponding mean. In many applications, it may be sufficient to provide a rough estimate, in which case we only need to use a small number of simulations.

Reference	\tilde{P}_{ij}	γ_i	\hat{Y}_i
Zhou et al. [2005a] (Random walk)	$\frac{w'_{ij}}{d'_i + \lambda d'_i}$	$\frac{\lambda}{1 + \lambda}$	$\mathbb{E}_i \left(\sum_{t=0}^{\infty} \frac{\lambda}{1 + \lambda} \frac{\sqrt{d_i}}{\sqrt{d_{\tilde{X}_t}}} Y_{\tilde{X}_t}^* \right)$
Belkin et al. [2004] (Tikhonov)	$\frac{w_{ij}}{d_i + \lambda_i}$	$\frac{\lambda_i}{d_i + \lambda_i}$	$\mathbb{E}_i \left(\sum_{t=0}^{\infty} \frac{\lambda_{\tilde{X}_t}}{d_{\tilde{X}_t} + \lambda_{\tilde{X}_t}} Y_{\tilde{X}_t}^* \right)$
Belkin et al. [2004] (Interpolated Tik)	$\frac{w_{ij}}{d_i}$	—	$\mathbb{E}_i \left(Y_{\tilde{X}_\tau}^* \right)$
Zhou et al. [2004] (Undirected rw)	$\frac{w_{ij}}{d_i + \lambda d_i}$	$\frac{\lambda}{1 + \lambda}$	$\mathbb{E}_i \left(\sum_{t=0}^{\infty} \frac{\lambda}{1 + \lambda} \frac{\sqrt{d_i}}{\sqrt{d_{\tilde{X}_t}}} Y_{\tilde{X}_t}^* \right)$

Table 5.1: Summary of Markov chain constructions for computing the predictions \hat{Y}_i .

There are several advantages of using Markov chains to approximate the predictions, especially when the graph is in large scale. First, these Markov chain algorithms

are online and local. In other words, to compute the prediction at node i , we only need to feed information piece by piece to the chain and such information is mostly local to node i . This is because the probability that the chain wanders far away from i before getting killed or stopped is very small. This feature is particularly useful if we are dealing with Internet-size problems, where even storing the entire graph is impossible. Second, every chain is independently run. These Markov chain algorithms are therefore easily parallelizable, which offers a cost-effective way to drastically speed up the computation as it is inexpensive to access clusters of workstations in the modern computer age. Finally, these Markov chain algorithms also make it possible to compute the predictions only for a subset of chosen nodes. If for some reason we only want to learn about a few nodes, it is only necessary to start the chains from these nodes and none from the rest of the graph.

Moreover, the Markov chain formulations provide two natural ways to reduce computation by further approximation. The first is to impose a “global” stopping criterion, where we make sure each chain does not run for too long by chance. For instance, a simple modification is to kill a chain if $t > T$ for some fixed $T > 0$. Secondly, we can design an adaptive method that is able to account for different variances at different nodes, and choose to simulate more chains from nodes with larger variability.

Other than computational advantages, these Markov chain constructions also open the door to alternative interpretations of the original algorithms and hence offer insights into how they work. First, it is easy to see that $\alpha_{ij} \geq 0$ across all methods, matching the observations we made in Section 4.2. It is also fairly easy to notice that the random walk prediction at node i has a factor of $\sqrt{d'_i}$, and the Tikhonov smoothing does not rely on the imputed Y_j^* if j is unknown since $\lambda_j = 0$ in that case. Both random walk methods choose to use the same killing probability γ for all nodes, while the Tikhonov smoothing is more likely to kill the chain at low-degree nodes and never kills at unobserved nodes. In terms of the tuning parameter λ , we saw in Chapter 3 that λ can be interpreted as the inverse of the noise variance, while the Markov chain formulations here shed light on a different understanding. Notice that the killing probability γ_i increases as λ increases for all methods. Therefore the size of λ effectively controls the length of the chain, where larger λ tends to give shorter chains. In fact, for the two random walk methods, the length of the chain follows a geometric distribution with probability $\gamma = \lambda/(1 + \lambda)$. The expected length of the chain is then $\gamma^{-1} = \lambda^{-1} + 1$ (including the starting node).

So far we have always assumed a perfect knowledge of the graph structure and never doubted its validity. This is partly because examining the graph links is a big research field of its own. See Taskar et al. [2003] for example. However, with the help of the Markov chain construction, we can touch on the robustness of these prediction methods against errors in the graph structure. We demonstrate using a toy example

as follows. Consider a node i that has only one neighbor k in an undirected graph, where k (a hub/authority) has many neighbors. That is, $w_{ik} = w_{ki} = d_i \ll d_k$. We would like to investigate the impact on the prediction \widehat{Y}_i when w_{ik} is increased to $4w_{ik}$ (while the rest of the graph stays the same). For simplicity, we focus only on the undirected random walk method. Recall that

$$\alpha_{ij}^{urw} = \frac{\lambda}{1 + \lambda} \frac{\sqrt{d_i}}{\sqrt{d_j}} \mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right).$$

We first argue that the change to the expected occupation time is negligible. Depending on whether the chain is killed on the first step, we can write

$$\mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right) = \begin{cases} \mathbb{E}_i \left(\sum_{t=2}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \mid \tilde{X}_1 = k \right) \mathbb{P}(\tilde{X}_1 = k) & j \neq i \\ \mathbb{E}_i \left(\sum_{t=2}^{\infty} \mathbb{1}_{\{\tilde{X}_t=i\}} \mid \tilde{X}_1 = k \right) \mathbb{P}(\tilde{X}_1 = k) + 1 & j = i, \end{cases}$$

where $\mathbb{P}(\tilde{X}_1 = k) = 1/(1 + \lambda)$ does not depend on w_{ik} . Because $4w_{ik} \ll d_k$, the conditional expectation in the case $j = i$ can be ignored compared to 1, and for $j \neq i$, there is hardly any impact on the conditional expectation when w_{ik} is changed to $4w_{ik}$. Therefore, we can consider $\mathbb{E}_i \left(\sum_{t=0}^{\infty} \mathbb{1}_{\{\tilde{X}_t=j\}} \right)$ to be approximately unchanged. On the other hand, the new prediction weights α_{ij}^{urw} double for $j \neq i$ due to the change in d_i , and so does the prediction \widehat{Y}_i (if the imputed $Y_i^* = 0$). Even though hardly rigorous, this toy example presents a simple scenario where the undirected random walk prediction is highly affected by errors in the graph weights.

Finally, we want to point out that we have not been able to exhibit a similar Markov chain construction for the hub-and-authority smoothing. The hub random walk and the authority random walk have different stationary distributions in general, and hence their corresponding graph Laplacians $\tilde{\Delta}'_H$ and $\tilde{\Delta}'_A$ use different normalizations. As a result, we are unable to construct a simple transitional probability \tilde{P}_{ij} from a convex combination of these two Laplacians.

5.2.3 Approximation accuracy

We have seen that the solutions of some graph learning algorithms have equivalent formulations based Markov chains. We have therefore proposed to use Markov chain simulations to approximate these solutions in large scale problems. The approximation uses the sample averages to estimate the mean. In this section, we briefly study the rate that these sample averages converge. A faster convergence rate indicates a smaller number of simulations necessary to achieve the same accuracy and hence has practical importance.

We defer any theoretical analysis to future work, and only provide some simulation results here.

We consider three types of undirected graphs. The first one is the two dimensional grid considered in Section 4.2 where every node had degree 4. The second one is an Erdős-Rényi random graph with $p = 0.02$. The last one is a scale free random graph generated using the Barabási-Albert model.

For the purpose here, we use small graphs with $n = 400$ so we can compute the predictions exactly. The true responses Y_i are generated as independent $\mathcal{N}(0, 1)$, though the choice of Y_i does not matter in general. We then hold out 25% of the Y_i 's and predict using the (undirected) random walk, Tikhonov and interpolated Tikhonov methods, and their corresponding Markov chain approximations.

To quantify the distance between the approximated and the “exact” predictions, we use the following relative squared error:

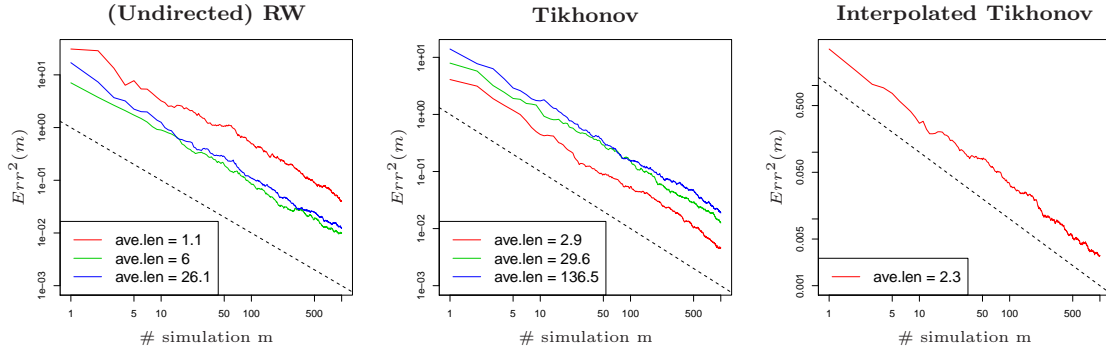
$$Err^2(m) = \frac{\sum_{i=r+1}^n (\hat{Y}_i^{(m)} - \hat{Y}_i)^2}{\sum_{i=r+1}^n \hat{Y}_i^2}, \quad (5.18)$$

where $\hat{Y}_i^{(m)}$ is the sample average from m independent chains and \hat{Y}_i is the exact prediction of the corresponding original algorithms (computed using matrix inversion).

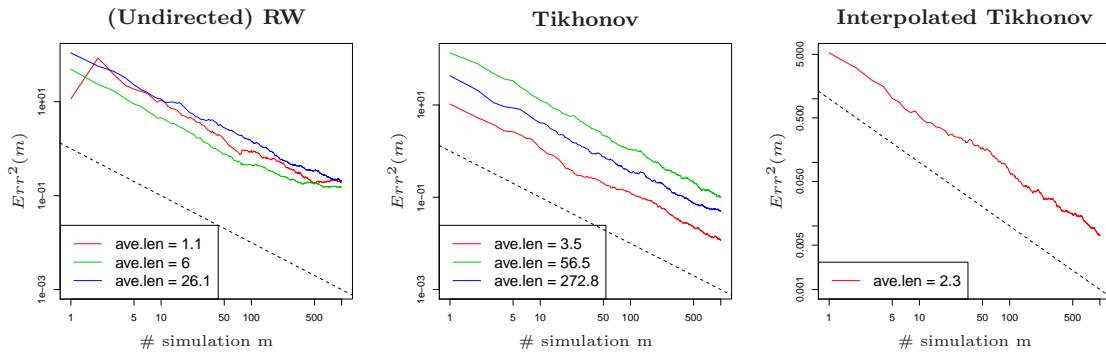
Figure 5.1 plots the relative error as a function of m in log-log scale. There are nine plots in total, one for each graph-method pair. For the random walk and the Tikhonov method, we present the results for three choices of λ : 10, 0.2 and 0.04. The corresponding curves are labeled with their average chain lengths in the legend. We also include a 45 degree reference line (dotted) in each plot, noting that it represents a convergence rate of exactly m^{-1} .

The first thing we notice is that the squared error, $Err^2(m)$, decreases as $\mathcal{O}(m^{-1})$ in most plots. The choice of the methods or the tuning parameter λ seems to only affect the rate by a constant. However, the exception is when applying the random walk algorithm on a Barabási-Albert random graph. The error converges very slowly in that case. It turns out that $\hat{Y}_i^{(1)}$, the estimate from a single chain, has a very skewed distribution due to the heavily skewed degree distribution of the Barabási-Albert graph. This presents a potential disadvantage of approximating the random walk algorithms using Markov chains in practice, as many real large graphs are shown to have power-law degree distributions (see Section 2.1.4). One possible way to mitigate this problem is to use importance sampling, though implementing it for a graph Markov chain seems non-trivial.

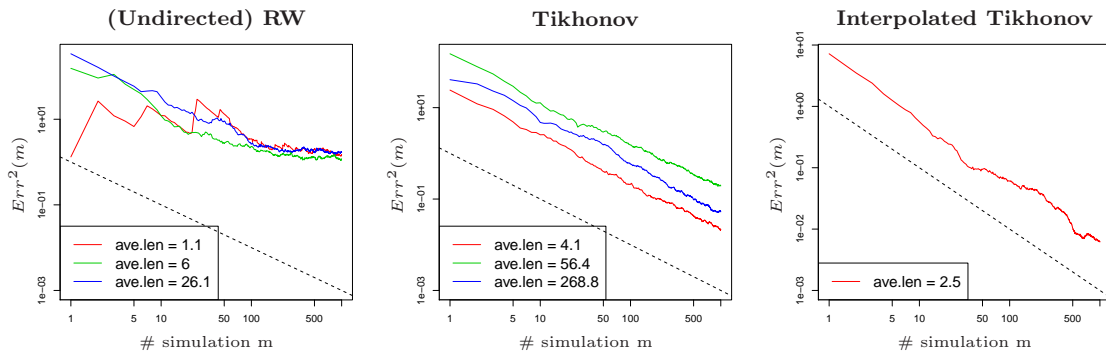
Note also the average chain lengths included in the legend. The Markov chain for the random walk method indeed has length $\lambda^{-1} + 1$ on average while the Tikhonov method has much longer chains, as discussed in the last section.



(a) Two dimensional grid



(b) Erdős-Rényi random graph



(c) Barabási-Albert random graph

Figure 5.1: The approximation error (5.18) as a function of the number of simulations (m). Three types of graphs are considered, together with three prediction algorithms. The dotted line represents a convergence rate of m^{-1} while lines parallel to it indicate a rate of $\mathcal{O}(m^{-1})$. Three λ values are used for the random walk and the Tikhonov methods. The corresponding curves are labeled with their average chain lengths.

5.3 Scale empirical kriging

As mentioned in Section 5.1, compared to the existing learning methods, our empirical kriging approach faces different challenges when it comes to scaling. First, the number of correlation pairs $(\widehat{R}_{ij}, s_{ij})$ to smooth is large. Second, we need to compute SVD of an $n \times n$ matrix to get a valid covariance estimation, which is expensive when n is large. The first problem only arises when the number r of labeled cases is large. Large r is much rarer than large n , and in any case can be mitigated by down-sampling the correlation pairs before smoothing. Therefore, we devote this section to focus on how to efficiently compute SVD during the covariance estimation.

We have shown in Chapter 4 that reduced rank covariance matrices can be used instead of the full rank version to reduce the cost of SVD computation, and it does not seem to hurt the prediction performance in some empirical experiments. Without any structural assumption about the matrix, an efficient SVD implementation takes approximately $\mathcal{O}(kn^2)$ to get the first $k \ll n$ eigen pairs of an $n \times n$ matrix. However, it is well studied that this can be largely improved if the matrix is sparse. In fact, using the Lanczos method described next in Section 5.3.1, the computation complexity is reduced to about $\mathcal{O}(kN)$ with N being the total number of non-zero elements of the matrix. We will then see in Section 5.3.2 that, with careful implementation, applying the Lanczos method on our *non-sparse* matrix $\widehat{\Sigma}$ can also be very efficient, thanks to the sparsity of the similarities s_{ij} .

5.3.1 Lanczos method

In this subsection we use notation that is common in the field of matrix computation but may be a departure from the conventions adopted in the rest of the thesis. The results we include here are standard, and we mostly follow the presentation of Chapter 9.1 in Golub et al. [1996].

Let $A \in \mathbb{R}^{n \times n}$ be a large, sparse and symmetric matrix. The Lanczos method, attributed to Lanczos [1950], is particularly useful when a few of A 's largest or smallest eigen pairs are desired. Denote $\lambda_i(\cdot)$ the i th largest eigenvalue and $\mathbf{u}_i(\cdot)$ the corresponding eigenvector of a given matrix. We suppress the notation for matrix A and use $\lambda_i = \lambda_i(A)$ and $\mathbf{u}_i = \mathbf{u}_i(A)$ for simplicity.

We start with a lemma that states the equivalence between the eigen pairs of A and that of its tridiagonalization.

Lemma 5.3.1. *Let $Q \in \mathbb{R}^{n \times n}$ be an orthogonal matrix such that $T = Q^T A Q$ is tridiagonal. Then*

$$\lambda_i = \lambda_i(T) \quad \text{and} \quad \mathbf{u}_i = Q \mathbf{u}_i(T).$$

Clearly, computing SVD of the tridiagonal matrix T is much easier and takes only $\mathcal{O}(n)$. However, two challenges present. First, we need an efficient algorithm to

find Q and hence T . Second, even with an algorithm that is computationally efficient, getting the full matrix Q (and T) is still too expensive when we only want the extreme eigen pairs.

Householder tridiagonalization can be adapted to solve the first challenge. However, it utilizes a transformation that destroys sparsity and hence is inefficient if A is large and sparse. This suggests direct computation using the Lanczos iteration in Algorithm 1 which produces $Q_m = [\mathbf{q}_1, \dots, \mathbf{q}_m]$ and

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & \dots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & \beta_{m-1} \\ 0 & \dots & & \beta_{m-1} & \alpha_m \end{bmatrix}$$

Obviously, with $m = n$ we get Q and T of the full tridiagonalization.

Algorithm 1: Lanczos iteration

Input: $A, \mathbf{q}_1, m; \mathbf{b}_0 = \mathbf{q}_1, \beta_0 = 1, \mathbf{q}_0 = 0$

Output: Q_m, T_m

```

1 for  $j = 0, \dots, m$  do
2    $\mathbf{q}_{j+1} = \mathbf{b}_j / \beta_j; j = j + 1; \alpha_j = \mathbf{q}_j^T A \mathbf{q}_j$ 
3    $\mathbf{b}_j = (A - \alpha_j I) \mathbf{q}_j - \beta_{j-1} \mathbf{q}_{j-1}; \beta_j = \|\mathbf{b}_j\|_2$ 
4 end
```

It turns out that the tridiagonal matrices T_m have a remarkable property that $\lambda_1(T_m)$ and $\lambda_m(T_m)$ are increasingly better estimates of λ_1 and λ_n as m increases. This provides solution to the second challenge and indicates that we may only need a small m to obtain the first or last eigen pair. In fact the following theorem from Golub et al. [1996] (Theorem 9.1.3 and Corollary 9.1.4) presents a result on the rate of convergence.

Theorem 5.3.2. *Let T_m be the tridiagonal matrix obtained after m steps of the Lanczos iteration, then*

$$\lambda_1 \geq \lambda_1(T_m) \geq \lambda_1 - \frac{(\lambda_1 - \lambda_n) \tan(\phi_1)^2}{(c_{m-1}(1 + 2\rho_1))^2} \quad (5.19)$$

$$\lambda_n \leq \lambda_m(T_m) \leq \lambda_n + \frac{(\lambda_1 - \lambda_n) \tan(\phi_n)^2}{(c_{m-1}(1 + 2\rho_n))^2}, \quad (5.20)$$

where $\cos(\phi_1) = |\mathbf{q}_1^T \mathbf{u}_1|$, $\cos(\phi_n) = \mathbf{q}_n^T \mathbf{u}_n$, $\rho_1 = (\lambda_1 - \lambda_2)/(\lambda_2 - \lambda_n)$, $\rho_n = (\lambda_{n-1} - \lambda_n)/(\lambda_1 - \lambda_{n-1})$, and $c_{m-1}(x)$ is the Chebyshev polynomial of degree $m - 1$.

To get a sense of how fast the bounds converge, we plot the denominator $(c_{m-1}(1 + 2\rho))^2$ as a function of the iteration m in Figure 5.2. Clearly, when $\rho = 0$, the denominator stays at 1 and the bounds never converge. When ρ increases to 0.1, we immediately observe convergence by the 50th iteration. In contrast, if $\rho = 10^{-10}$ (not shown in the plot), it takes more than 10^6 iterations to reach the same level of convergence. The rate of convergence is extremely sensitive to the value of ρ , or equivalently, to the size of the relevant eigen gap.

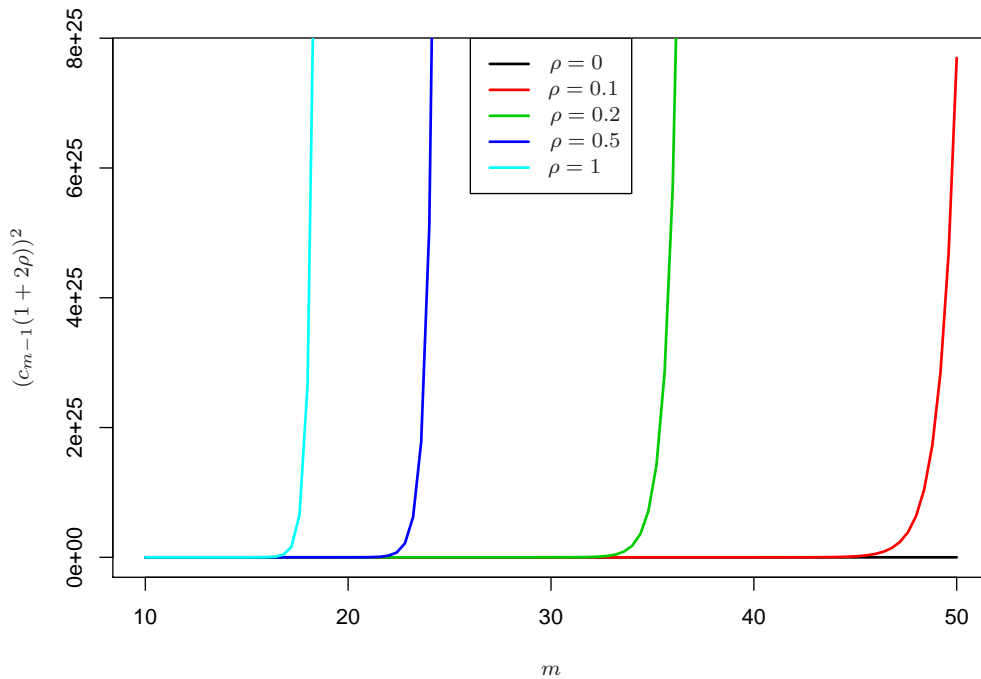


Figure 5.2: (Inverse) rate of convergence of the Lanczos iteration bounds in (5.19) and (5.20).

In summary, if we want to compute the largest eigen pair $(\lambda_1, \mathbf{u}_1)$ of the matrix A , we

- 1) compute Q_m and T_m using the Lanczos iteration in Algorithm 1;
- 2) compute $(\lambda_1(T_m), \mathbf{u}_1(T_m))$;
- 3) set $\lambda_1 \approx \lambda_1(T_m)$ and $\mathbf{u}_1 \approx \mathbf{u}_1(T_m)$.

The total computation for a given m is then $\mathcal{O}(mN)$ where N is the total number of non-zero elements in A . Of course, the number m of iterations needed to reach a certain precision depends on the eigen gaps and the choice of \mathbf{q}_1 . Similar procedure follows if $(\lambda_n, \mathbf{u}_n)$ is desired instead.

5.3.2 Estimating low rank covariance

We are now ready to see how we can apply the Lanczos method to get a low rank estimate of the covariance matrix Σ .

Recall that the rank- k covariance estimate $\widehat{\Sigma}_+^{(k)}$ is the rank- k positive semi-definite matrix that is closest to the initial estimate $\widehat{\Sigma}$. That is, the estimate keeps only the top k eigen pairs of $\widehat{\Sigma}$. The Lanczos algorithm just described would have been the exact tool we need, if $\widehat{\Sigma}$ were sparse. Fortunately, we will see that with careful implementation, the same algorithm still works for our dense matrix $\widehat{\Sigma}$.

We start with the decomposition

$$\widehat{\Sigma} = \sigma^2 V \widetilde{R} V,$$

where $\widetilde{R}_{ij} = \hat{\rho}(s_{ij})$ and V is $\text{diag}(\mathbf{v})$. Let $\hat{r}_0 = \hat{\rho}(0)$, then

$$\begin{aligned} \widehat{\Sigma} &= \sigma^2 V \left(\widetilde{R} - \hat{r}_0 \mathbf{1}\mathbf{1}^T + \hat{r}_0 \mathbf{1}\mathbf{1}^T \right) V \\ &= \sigma^2 V \left(\widetilde{R} - \hat{r}_0 \mathbf{1}\mathbf{1}^T \right) V + \sigma^2 \hat{r}_0 \mathbf{v}\mathbf{v}^T, \end{aligned} \quad (5.21)$$

where the first part $A \equiv \sigma^2 V \left(\widetilde{R} - \hat{r}_0 \mathbf{1}\mathbf{1}^T \right) V$ is an $n \times n$ matrix with the same sparsity as the matrix of similarities. In both the random walk smoothing and the Tikhonov smoothing, for example, the similarities s_{ij} share the same sparsity as the graph, in which case A will have the same number of non-zero elements as there are edges in the graph. More importantly, notice that we have now decomposed $\widehat{\Sigma}$ into the sum of a sparse, symmetric matrix A and a rank one matrix $\sigma^2 \hat{r}_0 \mathbf{v}\mathbf{v}^T$.

To see why and how the Lanczos algorithm still applies, first notice that the sparsity plays a role only in the tridiagonalization process. In both Step 2 and 3 of Algorithm 1, $A\mathbf{q}_j$ is computed which takes only $\mathcal{O}(N)$ because A has N non-zero elements. However, replacing A with our non-sparse $\widehat{\Sigma}$ does not increase the computation complexity because

$$\widehat{\Sigma}\mathbf{q}_j = A\mathbf{q}_j + \sigma^2 \hat{r}_0 \mathbf{v}(\mathbf{v}^T \mathbf{q}_j),$$

where the second part takes only $n < N$ flops.

5.3.3 Practical implementations and PROPACK

We have shown that the Lanczos algorithm can be used to scale our covariance estimation in exact arithmetic notation. However, in practice, the Lanczos method as described in Section 5.3.1 can not be directly applied because it is largely affected by rounding errors, which then leads to the loss of orthogonality in Q . Many ideas have been proposed to mitigate this problem. We refer the interested readers to Chapter 9.2 of Golub et al. [1996], which outlines several of these practical implementations.

Our implementation relies on PROPACK, a package for computing SVD of sparse matrices based on the Lanczos method [Larsen, 1998]. PROPACK uses the Lanczos bidiagonalization with partial reorthogonalization technique to overcome the loss of orthogonality problem. The Matlab implementation is well documented and for our purpose we only need to make a small modification inside the Lanczos iteration. Furthermore, the convergence of the Lanczos iteration is usually fast as the first few eigen gaps are usually far from zero.

5.4 Wikipedia dataset

In this section, we will predict on a Wikipedia graph using the algorithms we have discussed so far. The dataset contains a snapshot of <http://en.wikipedia.org> on February 6th, 2007. The original dataset includes about 3.5 million pages together with the links among them and is provided by David Gleich at <http://www.cise.ufl.edu/research/sparse/matrices/Gleich/index.html>.

This dataset originally does not include labels for each node. We construct the labels as follows: (1) Set $Y_i = 1$ if node i points to “United States” (the node with the highest in-degree) and -1 otherwise. (2) Remove the “United States” node and all its edges. (3) Remove (sequentially) any node that has zero in-degree or out-degree. The resulting graph has 2,371,607 nodes with 41,109,211 edges. Among all the nodes, 7.56% of them (179,333) are labeled with 1 and the rest are -1 s. The graph is directed. In fact, only about 12.8% of the out/in links have a reciprocal in/out link.

The four semi-supervised learning methods are implemented based on their Markov chain equivalence as discussed in Section 5.2, compared with the PROPACK implementation of our empirical covariance method (Section 5.3). For all the experiments, we hold out the same 50% of randomly chosen labels. For methods that require an undirected graph, we symmetrize the graph by the convention that uses $W + W^T$ as the adjacency matrix.

Table 5.2 includes the summary information from one simulation of the Markov chain algorithm, where one chain is started from each of the unknown node. When it applies, we choose λ so that the average chain lengths are comparable across methods.

We also cutoff the chains at length 1000 to avoid extremely long chains, which may happen for the two Tikhonov methods as they only kill the chain at observed nodes. In fact, if $\Delta + \Lambda_{\lambda_0,0}$ or Δ_{11} is singular, which is likely in real data, some chains will never die. The singularity makes it impossible to compute the exact solutions to the Tikhonov methods, and the early cutoff in the Markov chain implementation offers an easy and reasonable approximation.

Method	λ	ave. len.	# long chains	CPU time
Random Walk	1/30	30.95	0	16 mins
Tikhonov	3	31.58	10	19 mins
Interpolated Tik.	–	2.04	10	42 secs
Undirected RW	1/30	30.95	0	18 mins

Table 5.2: Summary information of one simulation on the Wikipedia dataset, where one chain is started from each of the 1.2 million unknown node. The length of the chain includes the starting node. The long chains are marked as length greater than 1000. All the timings are carried out on an Intel Xeon 2.66GHz processor.

Table 5.3 includes the information for the PROPACK implementation of the empirical covariance method. For simplicity, we illustrate using only the Tikhonov choice of s_{ij} and v_i (see Section 4.3.2 for discussion).

Method	λ	σ^2	s_{ij}	v_i	CPU time
Empirical R1	3	5/3	$w_{ij} + w_{ji}$	1	63 secs
Empirical R5	3	5/3	$w_{ij} + w_{ji}$	1	7 mins

Table 5.3: Summary information of the PROPACK implementation of the empirical covariance method on the Wikipedia dataset. All the timings are carried out on an Intel Xeon 2.66GHz processor.

Figure 5.3 plots the prediction errors measured based on the area under the ROC curve (AUC). We ran 200 simulations in total for each Markov chain algorithm, and the errors are then computed using the predictions averaged over the first m of them. We observe a nice decay of the error as the number of simulations increase for the two Tikhonov methods, and a particularly faster convergence for the interpolated version. In contrast, the two random walk methods do not seem to do better with more simulations. It is possible that, as we observed in Section 5.2.3, the random walk methods converge a lot slower due to the heavily skewed degree distribution. The empirical covariance method, on the other hand, does not require multiple simulations, so their

corresponding error curves are constant. The rank 5 version performs slightly better than the rank 1 version in this example.



Figure 5.3: Prediction performance on the Wikipedia dataset with 50% response values held out.

5.5 Conclusion and discussion

We have shown that several recently developed semi-supervised prediction algorithms can be scaled to large graphs using Markov chains. The equivalence between these algorithms and their corresponding Markov chain constructions not only offer a different perspective to understand these algorithms, but also a way to approximate their predictions that have computational and storage advantages. We have also shown that we can scale our empirical covariance method by carefully implementing the sparse matrix SVD techniques. In particular, our implementation involves decomposing a dense matrix into the sum of a sparse matrix and a low rank matrix, and then applying PROPACK with a small modification. We saw in the Wikipedia example that we can easily scale to predict on a graph with 2.3 million nodes.

The strategies we have proposed to scale the existing graph prediction algorithms and our empirical covariance method are completely different. A natural question is whether these two scaling approaches can be used interchangeably. In other words, we would like to know whether it is possible to construct a Markov chain to solve our empirical method and whether PROPACK can be used to get a low rank approximation for the existing algorithms.

As far as we understand, in order to have a Markov chain equivalence, an algorithm must have an implicit Markovian structure. The underlying assumption of the empirical covariance model is that the correlations are stationary with respect to a certain distance in the graph. This does not automatically entail any Markovian property. In fact, the Markovian and the stationary assumptions seem quite different intuitively. Consider a toy example where the whole graph is comprised of several smaller disjoint components with identical link structures. A Markov chain will be trapped in a single component and cannot borrow strength from labels known in other components. On the other hand, under the stationarity assumption, all the observations are used to learn the correlation function and we are not limited to “local” information only.

The other direction, i.e. applying PROPACK to get a low rank approximation of $(L + \Lambda)^{-1}$, may seem easier at first. However, notice that because the approximation requires the smallest eigen pairs of $L + \Lambda$, the convergence rate of the Lanczos iterations is highly sensitive to the eigen gap between the smallest and the second smallest eigenvalues (see (5.20) and Figure 5.2). In particular, the convergence is very slow when the gap is extremely close to zero, which is often the case in practice. In fact, if L is unnormalized or normalized graph Laplacian and there are more than one connected component in the graph, the eigen gap is exactly zero (see Proposition 2.1.1 and 2.1.2), which means the bounds never converge.

Even though the two scaling strategies may not be interchangeable, the Markov chain interpretation of the existing algorithms opens door to a variety of possible extensions where we may borrow some intuition from the empirical stationary correlation method. For instance, we may modify the Tikhonov chain to occasionally jump to distant nodes that share the *same* label as the current node. This encourages the chain to explore regions that are far away but hopefully similar at the same time. We have not studied simple modifications like this, however we do hope that what we have done can provide a suitable starting point for future work in this direction.

Chapter 6

Investigating the smoothness measures

6.1 Introduction

We have presented several semi-supervised learning graph algorithms in Chapter 2 and then studied them from a kriging framework (Chapter 3) and a Markov chain framework (Chapter 5) in the subsequent chapters. However, in this chapter, we will return to their original formulations and look into the fundamental assumption underlying these methods.

We have seen that all these methods assume the response should vary smoothly in the graph. Such smoothness is usually achieved by regularization, and a popular choice is to use the graph Laplacian. The smoothness assumption plays an essential role in prediction. However, to the best of our knowledge, there has not been any systematic approach proposed to investigate whether a smoothness measure is reasonable. Many graph-based learning algorithms attempt to validate their choices by demonstrating good empirical performance. Our approach starts by examining how much smoother the true labels are compared to random labels under a given measure. To our surprise, we found that randomizing the labels can sometimes lead to a smoother answer in real examples. This casts doubt on the utility of the corresponding smoothness measures, and motivated us to further investigate into the phenomenon.

The outline of this chapter is as follows. Section 6.2 starts with a brief review of the relevant notation and background. We then present the experimental results from three real datasets, where randomly permuted labels are sometimes found to be much smoother than the ground truth. We therefore propose a measure to quantify the validity of the smoothness assumption. Section 6.3 includes some theoretical results based on Stein's central limit theorem for dependent random variables [Stein,

1986]. It serves two purposes: first, it provides insight and theoretical justifications for the empirical results and our proposed measure; second, it shows that the surprising findings in Section 6.2 are reproducible.

6.2 Motivation: Permutation experiment

We restrict our attention to graph \mathcal{G} that is undirected with weights $\{w_{ij}\}_{i,j=1}^n$ and degrees $\{d_i\}_{i=1}^n$. Recall that responses Y_i and Y_j are assumed to be close in value if nodes i and j are close to each other in the graph. In other words, the vector $\mathbf{Y} \in \mathbb{R}^n$ is smooth with respect to a notion of distance in the graph. Mathematically, this smoothness assumption is reflected as a regularization term that penalizes large values of a variation functional. Two popular choices for the functional correspond to the unnormalized and the normalized graph Laplacians (see Chapter 2):

$$\Omega(\mathbf{Y}) \equiv \mathbf{Y}^T \Delta \mathbf{Y} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} (Y_i - Y_j)^2 \quad (6.1)$$

$$\tilde{\Omega}(\mathbf{Y}) \equiv \mathbf{Y}^T \tilde{\Delta} \mathbf{Y} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \left(\frac{Y_i}{\sqrt{d_i}} - \frac{Y_j}{\sqrt{d_j}} \right)^2. \quad (6.2)$$

Both smoothness measures penalize vectors \mathbf{Y} that differ too much over similar nodes. However, by using the normalized graph Laplacian $\tilde{\Delta}$, $\tilde{\Omega}(\mathbf{Y})$ effectively scales the variables by $\sqrt{d_i}$.

Note the importance of the smoothness assumption in the semi-supervised learning algorithms. Through smoothing, the information from the observed nodes is “propagated” to the unobserved nodes. In fact, the smoothness measure such as the ones in (6.1) and (6.2) is usually the only component in the algorithms that connects the graph structure and the feature of interest. Clearly, if that measure fails, these algorithms are likely to perform poorly.

To make the problem more difficult, the feature \mathbf{Y} may vary smoothly in the graph with respect to one measure, but less so to another. This is because there are many choices to make when defining a smoothness measure, and some may be better than others for a particular dataset. First of all, there is no unique way to construct a graph when the edge weights w_{ij} are not readily available. Readers are referred to a recent paper by Jebara et al. [2009] for a nice empirical study of several construction algorithms, while Argyriou et al. [2005] proposes to use an optimal combination of a number of differently constructed graphs. Secondly, the theoretical implications of Laplacian normalization are not well understood, thus it is unclear whether one should use the smoothness measure $\Omega(\mathbf{Y})$ or $\tilde{\Omega}(\mathbf{Y})$ in the learning algorithm. Johnson and Zhang [2007] derives near-optimal normalization factors under certain assumptions

using a worst-case generalization bound.

The approach we take to investigate the smoothness measures starts with the following question: how much smoother are the true labels compared to a random set of labels? To be more concrete, for a fixed graph structure, we randomly permute the labels on the nodes and then compute the smoothness score. We then compare the score of the true labels with that of the permuted labels. This idea is demonstrated in Figure 6.1. Intuitively, permutation destroys the connection between the labels and the graph structure, and hence the resulted labels should be rougher with respect to the graph, resulting in a larger score.

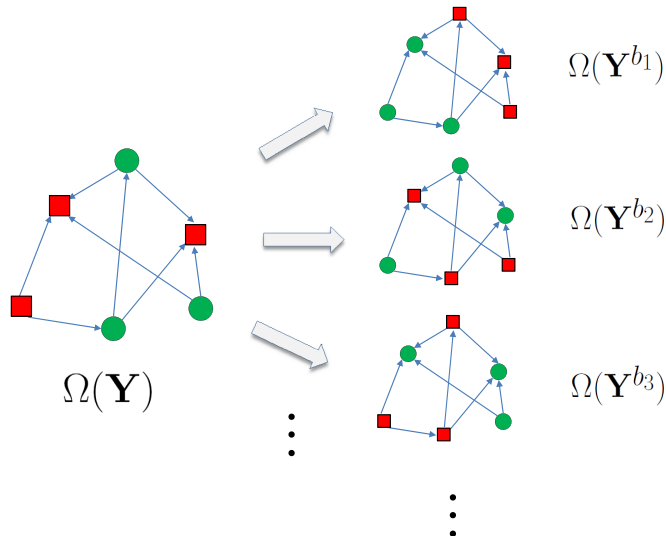


Figure 6.1: Permuting labels while fixing the graph structure.

As an attempt to validate our intuition, we perform this permutation procedure on three real datasets. The first dataset is a protein interaction network from Jeong et al. [2001], where some proteins are labeled as lethal and others are non-lethal. The second is a webspam dataset available at <http://barcelona.research.yahoo.net/webspam/datasets/uk2007/>. The third dataset is the Wikipedia dataset we used in Chapter 5. Table 6.1 includes a summary of all three datasets.

Figure 6.2 shows the results for the two smoothness measures defined in (6.1) (first row) and (6.2) (second row). The histograms represent the distributions of scores for the randomly permuted labels, with the red dashed lines being the medians of the empirical distributions. The red dots are the scores of the true labels. Under the smoothness assumption, we expect the ground truth to give smaller scores, however, five out of the six plots tell us otherwise. Except for plot (1b) where the red dot is

Name	Nodes	Edges	Response Y
Protein	2114	4406	401 lethal (20.0%)
Webspam	3014	5578	225 spam (7.5%)
Wikipedia	2,371,607	41,109,211	179,333 points to “US” (7.5%)

Table 6.1: Summary of the three datasets used in permutation experiment

in the left tail of the distribution, all the other red dots are either near the median line or deep in the right tail. This implies that the smoothness measures defined in (6.1) or (6.2) are sometimes inadequate to differentiate the true set of labels from a random permutation. Moreover, random shuffling could even result in labels that are much smoother as shown in (1c) and (2c)!

These results are quite striking. They not only reveal some potential problems with the smoothness assumption or some particular smoothness measures, but also inspire us to consider the following z -score to quantify how well the assumption holds:

$$z(\Omega, \mathbf{Y}) \equiv \frac{\Omega(\mathbf{Y}) - \mathbb{E}(\Omega(\mathbf{Y}^b))}{\text{sd}(\Omega(\mathbf{Y}^b))}, \quad (6.3)$$

where \mathbf{Y}^b is a bootstrap copy of the original response \mathbf{Y} . The z -score quantifies the difference between the red dot and the mean of the distribution, taking into account of the spread. Clearly, a large negative z -score supports the smoothness assumption, while a large positive one suggests otherwise.

In the next section, we will present some theoretical results that will justify the empirical findings in Figure 6.2 as well as the z -score defined in (6.3).

6.3 Central limit theorems for smoothness measures

We notice that among the examples in Figure 6.2, almost all of the histograms look normally distributed. If we can well approximate the score distributions of the permuted labels with a normal distribution, the z -score defined in (6.3) is equivalent to a p -value in hypothesis testing, and therefore is a more interpretable measure that is readily comparable across different cases. However, the normal approximation clearly does not hold in general, as demonstrated in Figure 6.2 (2a). In this section, we will develop some theoretical results that give conditions under which normality holds.

Before we investigate how the smoothness measures behave asymptotically, we want to emphasize two properties of real graphs we have discussed in Section 2.1.4.

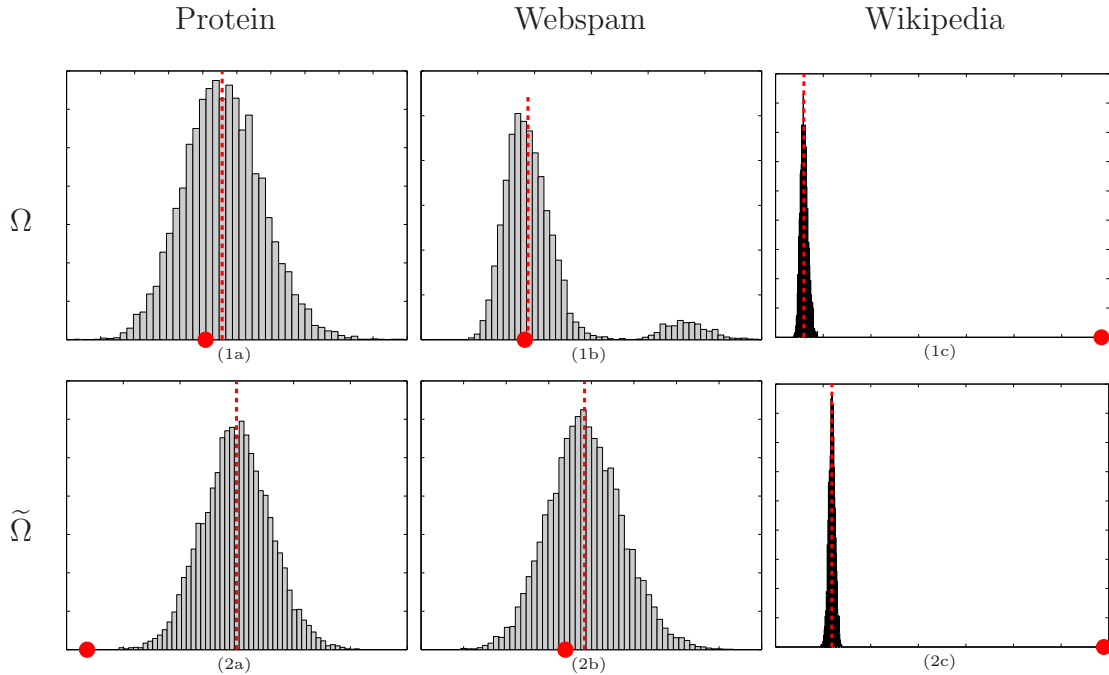


Figure 6.2: Distributions of the smoothness scores of the randomly permuted labels, compared with that of the ground truth labels (red dots).

First, real large graphs are usually very sparse, where the total number of edges in the graph grows linearly with n . We can notice this phenomenon in the examples presented in Table 6.1. Secondly, many studies have shown that real graphs tend to have a heavy tailed degree distribution. Even though far from capturing the full structure of a graph, the degrees are fairly informative, and in fact, many efforts have been devoted to generating random graphs that match the degree distributions of those of real graphs.

To this end, we make two structural assumptions about the graph \mathcal{G} in our analysis. First, we assume \mathcal{G} is sparse such that $\text{vol}(\mathcal{G}) = \mathcal{O}(n)$. Second, the degrees of \mathcal{G} have certain nice properties, and two different aspects are considered. In Section 6.3.2, we investigate the case where the maximal degree of \mathcal{G} may grow with n while all other nodes have bounded degrees. We are able to give sufficient conditions where a normal approximation works or fails, which turn out to help explain our observations in Figure 6.2. In Section 6.3.3, the degrees are assumed to follow a probability distribution and the conditions for a CLT involve moments of that distribution. In the context of power-law degree distributions, these conditions can be readily translated into conditions on the exponent.

We simplify the analysis by assuming the graph is unweighted, i.e. $w_{ij} \in \{0, 1\}$.

All the results can be generalized to weighted graphs with nonzero weights bounded from above and away from zero. We further assume bounded feature values.

6.3.1 Preliminaries

We start with two preliminary theorems. Stein's central limit theorem for dependent random variables is the most fundamental result of all those to be discussed in this section. It is quoted below:

Theorem 6.3.1 (Stein [1986], p.110). *Let X_1, \dots, X_n be random variables, and let $\mathcal{S}_1, \dots, \mathcal{S}_n$ be subsets of $\{1, \dots, n\}$ such that*

$$\mathbb{E}X_i = 0, \quad \mathbb{E}X_i^4 < \infty, \quad \text{and} \quad \mathbb{E} \sum_{i=1}^n X_i \sum_{j \in \mathcal{S}_i} X_j = 1.$$

Let $T = \sum_{i=1}^n X_i$. Then for any $t \in \mathbb{R}$,

$$\begin{aligned} \left| \mathbb{P}(T \leq t) - \Phi(t) \right| &\leq 2 \sqrt{\mathbb{E} \left\{ \sum_{i=1}^n \sum_{j \in \mathcal{S}_i} (X_i X_j - \mathbb{E}X_i X_j) \right\}^2} \\ &\quad + \sqrt{\frac{\pi}{2}} \mathbb{E} \sum_{i=1}^n |\mathbb{E}(X_i | X_j : j \notin \mathcal{S}_i)| \\ &\quad + 2^{3/4} \pi^{-1/4} \sqrt{\mathbb{E} \sum_{i=1}^n |X_i| \left(\sum_{j \in \mathcal{S}_i} X_j \right)^2}, \end{aligned} \quad (6.4)$$

where $\Phi(\cdot)$ is the standard normal cdf.

Stein's theorem allows dependence among the variables, and in many applications, \mathcal{S}_i contains all the variables X_i depends on. Intuitively, the theorem states that if the overall dependence of the random variables is weak, a normal approximation of their sum still works well.

Rinott [1994] refines the result in Theorem 6.3.1 assuming that the random variables are bounded. Even though his intention was to improve the convergence rate in (6.4), we include this result here because it is directly applicable for the problems we consider in Section 6.3.2. We quote a necessary definition and then the theorem itself.

Definition Let $\{X_i : i \in \mathcal{V}^*\}$ be a collection of random variables. The graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$, where \mathcal{V}^* and \mathcal{E}^* denote the vertex set and the edge set, respectively, is said to be a *dependency graph* for the collection if for any pair of disjoint subsets of \mathcal{V}^* ,

\mathcal{S}_1 and \mathcal{S}_2 such that no edge in \mathcal{E}^* has one endpoint in \mathcal{S}_1 and the other in \mathcal{S}_2 , the sets of random variables $\{X_i : i \in \mathcal{S}_1\}$ and $\{X_i : i \in \mathcal{S}_2\}$ are independent.

Note that we distinguish a dependency graph \mathcal{G}^* from a general graph \mathcal{G} . \mathcal{G}^* is undirected and unweighted by definition. Moreover, it contains selfloops since a variable always depends on itself.

Theorem 6.3.2 (Rinott [1994]). *Let X_1, \dots, X_n be random variables having a dependency graph \mathcal{G}^* whose maximal degree is less than d^* , satisfying $|X_i - \mathbb{E}X_i| \leq c$ a.s. for $i = 1, \dots, n$. Let $T = \sum_{i=1}^n X_i$ and assume $\text{var}(T) = \sigma^2 > 0$. Then*

$$\left| \mathbb{P}\left(\frac{T - \mathbb{E}T}{\sigma} \leq t\right) - \Phi(t) \right| \leq \frac{1}{\sigma} \left(\sqrt{\frac{1}{2\pi}} d^* c + 16 \frac{\sqrt{n}}{\sigma} d^{*3/2} c^2 + 10 \frac{n}{\sigma^2} d^{*2} c^3 \right). \quad (6.5)$$

Note that when d^* and c are bounded, the right hand side of (6.5) converges to zero at the rate of $n^{-1/2}$ if $\sigma^2 = \mathcal{O}(n)$.

6.3.2 Maximal degree conditions

As a starting point, we consider a simple case where all the degrees in the graph are bounded. Intuitively, a real person (i.e. not a robot) can only have so many real friends on Facebook even though the entire social network has millions of users. We then extend the results to a more general framework where the maximal degree is allowed to grow with n . This is a more realistic condition in many applications. For instance, on Internet graphs, there may be a few hub/authority websites that many other sites connect to, and it is clearly no longer appropriate to assume bounded degrees as $n \rightarrow \infty$.

We will first quote a result on variance of quadratic forms from Seber and Lee [2003]. We will find it useful when computing the variance of the smoothness scores.

Lemma 6.3.3 (Seber and Lee [2003], p.10). *Let Y_1, \dots, Y_n be i.i.d. random variables with mean μ_1 . Denote $\mu_r = \mathbb{E}(Y_i - \mu_1)^r$. If $A \in \mathbb{R}^{n \times n}$ is symmetric and $\mathbf{a} = \text{diag}(A)$, then*

$$\text{var}(\mathbf{Y}^T A \mathbf{Y}) = (\mu_4 - 3\mu_2^2) \mathbf{a}^T \mathbf{a} + 2\mu_2^2 \text{tr}(A^2) + 4\mu_2 \mu_1^2 \mathbf{1}^T A^2 \mathbf{1} + 4\mu_3 \mu_1 \mathbf{1}^T A \mathbf{a}.$$

The following lemma states that the central limit theorem holds for both smoothness measures (6.1) and (6.2), when the maximal degree of a graph is bounded.

Lemma 6.3.4 (Bounded maximal degree). *Let Y_1, \dots, Y_n be i.i.d. random variables corresponding to vertices of the graph \mathcal{G} , satisfying $|Y_i| \leq c < \infty$ a.s. $\forall i$. Assume*

$\text{vol}(\mathcal{G}) = \mathcal{O}(n)$ and $\max_i d_i \leq d < \infty$. Then, as $n \rightarrow \infty$

$$\frac{\Omega(\mathbf{Y}) - \mathbb{E}\Omega(\mathbf{Y})}{\text{sd}(\Omega(\mathbf{Y}))} \implies \mathcal{N}(0, 1), \quad \text{and} \quad (6.6)$$

$$\frac{\tilde{\Omega}(\mathbf{Y}) - \mathbb{E}\tilde{\Omega}(\mathbf{Y})}{\text{sd}(\tilde{\Omega}(\mathbf{Y}))} \implies \mathcal{N}(0, 1), \quad (6.7)$$

with $\Omega(\cdot)$ and $\tilde{\Omega}(\cdot)$ defined in (6.1) and (6.2), and $\mathcal{N}(0, 1)$ being the standard normal distribution.

Proof. We prove the result in (6.6) and the similar proof for (6.7) can be found in Appendix B.2. Notice that $\Omega(\mathbf{Y})$ can be decomposed into sum of dependent variables:

$$\Omega(\mathbf{Y}) = \mathbf{Y}^T \Delta \mathbf{Y} = \sum_{i=1}^n d_i Y_i^2 - \sum_{i=1}^n \sum_{j=1}^n w_{ij} Y_i Y_j \equiv \sum_{i=1}^n X_i,$$

where $X_i = d_i Y_i^2 - \sum_{j=1}^n w_{ij} Y_i Y_j$. From here on, we will drop the summation limits when they go from 1 to n .

The random variables X_1, \dots, X_n are almost independent. In fact X_i and X_j are dependent if and only if $w_{ij} > 0$ or $\sum_k w_{ik} w_{kj} > 0$. In other words, X_i 's have a dependency graph \mathcal{G}^* where i and j are connected if and only if they are first or second order neighbors in the original graph \mathcal{G} . The maximal degree of this dependency graph is at most $d + d^2$, so is clearly bounded. Therefore, (6.6) readily follows from Theorem 6.3.2 after we show that $|X_i|$ is bounded a.s. for all i and $\text{var}(\Omega(\mathbf{Y})) = \mathcal{O}(n)$.

First,

$$|X_i| \leq d_i Y_i^2 + |Y_i \sum_j w_{ij} Y_j| \leq 2c^2 d_i \leq 2c^2 d. \quad (6.8)$$

Applying Lemma 6.3.3 with $A = \Delta$ and noting that $\Delta_{ii} = d_i$ and $\Delta \mathbf{1} = \mathbf{0}$, we have

$$\begin{aligned} \text{var}(\Omega(\mathbf{Y})) &= (\mu_4 - 3\mu_2^2) \sum_i d_i^2 + 2\mu_2^2 \left(\sum_i d_i^2 + \sum_i \sum_j w_{ij}^2 \right) \\ &= (\mu_4 - \mu_2^2) \sum_i d_i^2 + 2\mu_2^2 \sum_i \sum_j w_{ij}^2 \\ &\leq d(\mu_4 + \mu_2^2) \sum_i d_i \\ &= \mathcal{O}(n), \end{aligned} \quad (6.9)$$

where μ_r 's are the centered moments of Y_i as defined in Lemma 6.3.3. \square

We are now ready to state a more general theorem where the maximal degree grows as $\mathcal{O}(n^\epsilon)$. It turns out that the asymptotic normality breaks down for $\Omega(\mathbf{Y})$ when $\epsilon \geq 1/2$, while it holds for $\tilde{\Omega}(\mathbf{Y})$ when $0 \leq \epsilon < 1$. The theorem reduces to Lemma 6.3.4 when $\epsilon = 0$.

Theorem 6.3.5. *Let Y_1, \dots, Y_n be i.i.d. random variables corresponding to vertices of the graph \mathcal{G} , satisfying $|Y_i| \leq c < \infty$ a.s. $\forall i$. Assume $\text{vol}(\mathcal{G}) = \mathcal{O}(n)$. Let $d_k = \max_i d_i = \mathcal{O}(n^\epsilon)$ with $0 \leq \epsilon < 1$, while $d_i \leq d < \infty \forall i \neq k$. Then, as $n \rightarrow \infty$,*

$$\frac{\Omega(\mathbf{Y}) - \mathbb{E}\Omega(\mathbf{Y})}{\text{sd}(\Omega(\mathbf{Y}))} \implies \begin{cases} \mathcal{N}(0, 1) & 0 \leq \epsilon < 1/2 \\ \mathcal{O}_p(1) \cdot \mathcal{N}(0, 1) + \mathcal{O}_p(1) \cdot g(Y_k) & \epsilon = 1/2 \\ \mathcal{O}_p(1) \cdot g(Y_k) & 1/2 < \epsilon < 1, \end{cases} \quad (6.10)$$

$$\frac{\tilde{\Omega}(\mathbf{Y}) - \mathbb{E}\tilde{\Omega}(\mathbf{Y})}{\text{sd}(\tilde{\Omega}(\mathbf{Y}))} \implies \mathcal{N}(0, 1) \quad \forall 0 \leq \epsilon < 1, \quad (6.11)$$

where $g(Y_k) = Y_k^2 - 2\mu_1 Y_k - \mu_2 + \mu_1^2$.

Proof. We prove the result in (6.10) and the proof for (6.11) can be found in Appendix B.3. Following equation (6.9), we have $\text{var}(\Omega(\mathbf{Y})) = \mathcal{O}(\sum_i d_i^2) = \mathcal{O}(n + n^{2\epsilon})$, where the last equality follows because $d_k = \mathcal{O}(n^\epsilon)$ is the only unbounded degree.

We can separate out the terms involving node k :

$$\Omega(\mathbf{Y}) = \Omega_{-k}(\mathbf{Y}_{-k}) + \sum_i w_{ki} (Y_k - Y_i)^2, \quad (6.12)$$

where $\Omega_{-k}(\mathbf{Y}_{-k}) = \sum_{i \neq k} \sum_{j \neq k} w_{ij} (Y_i - Y_j)^2 / 2$. From here on, we will drop the subscript on Ω for clarity. Applying this trick to the standardized score, we get

$$\begin{aligned} & \frac{\Omega(\mathbf{Y}) - \mathbb{E}\Omega(\mathbf{Y})}{\sqrt{n + n^{2\epsilon}}} \\ &= \frac{\Omega(\mathbf{Y}_{-k}) - \mathbb{E}\Omega(\mathbf{Y}_{-k})}{\sqrt{n + n^{2\epsilon}}} + \frac{\sum_i w_{ki} \left((Y_k - Y_i)^2 - \mathbb{E}(Y_k - Y_i)^2 \right)}{\sqrt{n + n^{2\epsilon}}}. \end{aligned} \quad (6.13)$$

By Lemma 6.3.4 the first term in (6.13) converges to a normal distribution when

$\epsilon \leq 1/2$ and is $o_p(1)$ when $\epsilon > 1/2$. On the other hand,

$$\begin{aligned} \frac{1}{\sqrt{n+n^{2\epsilon}}} \sum_i w_{ki} (Y_k - Y_i)^2 &= \frac{1}{\sqrt{n+n^{2\epsilon}}} \sum_i w_{ki} (Y_k^2 - 2Y_k Y_i + Y_i^2) \\ &= \frac{1}{\sqrt{n+n^{2\epsilon}}} (d_k Y_k^2 - 2Y_k \sum_i w_{ki} Y_i + \sum_i w_{ki} Y_i^2) \\ &\Rightarrow \begin{cases} o_p(1) & 0 \leq \epsilon < 1/2 \\ Y_k^2 - 2\mu_1 Y_k + (\mu_2 + \mu_1^2) & 1/2 \leq \epsilon < 1, \end{cases} \end{aligned}$$

by Slutsky's Theorem. The second term in (6.13) is this sum centered by its expectation, and hence goes to zero if $\epsilon < 1/2$ and to

$$g(Y_k) = Y_k^2 - 2\mu_1 Y_k - (\mu_2 - \mu_1^2)$$

when $\epsilon \geq 1/2$. Combining the two parts, together with $\text{var}(\Omega(\mathbf{Y})) = \mathcal{O}(n + n^{2\epsilon})$, we get the desired result. \square

We can show that, by repeatedly applying (6.12), Theorem 6.3.5 generalizes to the case where there are a bounded number of nodes with degrees growing as $\mathcal{O}(n^\epsilon)$.

It also follows from Theorem 6.3.5 that the smoothness measure $\tilde{\Omega}(\mathbf{Y})$ tends to be more robust than $\Omega(\mathbf{Y})$ against the maximal degree in terms of asymptotic normality. This is in fact well supported by the observations in Figure 6.2. The only non-normality occurs when applying $\Omega(\mathbf{Y})$ to the webspam dataset. A closer look at the webspam data shows that the maximal degree node is connected to about 25% of the entire graph. This is probably an artifact from web crawling, but nonetheless causes the maximal degree to be non-negligible compared to n . Even the two humps are well explained by the theorem. This is because for binary Y_k , the asymptotic distribution in (6.10) when $\epsilon = 1/2$ turns out to be a two-component Gaussian mixture.

6.3.3 Degree distribution conditions

Other than focusing on the maximal degree, we now consider the entire degree sequence. This is mainly motivated by the important role that degree distribution plays in the field of random graphs. The classical Erdős-Rényi model starts with n nodes and connects every pair of nodes with probability p , producing degrees that follow a Poisson distribution. Recently, direct measurement of the degrees of many real graphs show that a Poisson distribution does not usually apply. Rather, the degrees tend to follow a power-law distribution. This finding then inspired the study of scale-free graph models that lead to a power-law degree distribution. Nevertheless, when graphs are characterized by their degree distributions, the results in Section 6.3.2

developed around the maximal degree are no longer sufficient. In the main theorem (Theorem 6.3.7) of this section, we will give conditions for a normal approximation that involve only the moments of the degree distribution.

For clarity and continuity, we include several supporting lemmas in Appendix B.4. The following lemma is an application of Stein's result in Theorem 6.3.1, and is directly relevant to the main theorem to be introduced. It applies Stein's theorem to the situation where the variables form a dependency graph with a prescribed degree sequence. Some of the proof techniques are borrowed from Baldi and Rinott [1989].

Lemma 6.3.6. *Let $\{X_i, i \in \mathcal{V}^*\}$ be random variables having a dependency graph \mathcal{G}^* with degrees $\{d_1^*, d_2^*, \dots, d_n^*\}$. Let $T = \sum_i X_i$ and $\mathcal{S}_i^* = \{j \in \mathcal{V}^* : j \sim i\}$ be the set of nodes connected to node i . Assume that $\mathbb{E}X_i = 0$ and $\mathbb{E}(\sum_i \sum_{j \in \mathcal{S}_i^*} X_i X_j) = 1$. Then for any $t \in \mathbb{R}$,*

$$\begin{aligned} \left| \mathbb{P}(T \leq t) - \Phi(t) \right| &\leq 2^{3/4} \pi^{-1/4} \sqrt{\frac{1}{3} \sum_i \mathbb{E}|X_i|^3 d_i^{*2} + \frac{2}{3} \sum_i \sum_{j \in \mathcal{S}_i^*} \mathbb{E}|X_i|^3 d_j^*} \\ &\quad + 2 \sqrt{2 \sum_i \mathbb{E}X_i^4 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* + \sum_{k \in \mathcal{S}_i^*} d_i^* d_k^* \right)} \end{aligned} \quad (6.14)$$

Proof. Following Theorem 6.3.1, it suffices to show that the RHS of (6.4) is bounded from above by the RHS of (6.14).

First,

$$\begin{aligned} \mathbb{E} \sum_i |X_i| \left(\sum_{j \in \mathcal{S}_i^*} X_j \right)^2 &\leq \sum_i \sum_{j, k \in \mathcal{S}_i^*} \mathbb{E}|X_i| |X_j| |X_k| \\ &\leq \frac{1}{3} \sum_i \sum_{j, k \in \mathcal{S}_i^*} \mathbb{E} (|X_i|^3 + |X_j|^3 + |X_k|^3) \\ &= \frac{1}{3} \sum_i \mathbb{E}|X_i|^3 d_i^{*2} + \frac{2}{3} \sum_i \sum_{j \in \mathcal{S}_i^*} \mathbb{E}|X_i|^3 d_j^*, \end{aligned}$$

where the inequality follows from (B.7) and the equality follows from equations (B.9) and (B.8) in the Appendix. On the other hand,

$$\begin{aligned} \mathbb{E} \left(\sum_i \sum_{j \in \mathcal{S}_i^*} (X_i X_j - \mathbb{E}X_i X_j) \right)^2 &= \mathbb{E} \left(\sum_i \sum_{j \in \mathcal{S}_i^*} X_i X_j \right)^2 - 1 \\ &\leq 2 \sum_i \mathbb{E}X_i^4 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* + \sum_{k \in \mathcal{S}_i^*} d_i^* d_k^* \right), \end{aligned}$$

where the last step follows from (B.12) in the Appendix.

Finally, since X_i is independent of $\{X_j : j \notin \mathcal{S}_i^*\}$, the second term on the RHS of (6.4) vanishes. \square

The following theorem gives conditions on the degree distribution in order to achieve normality. These conditions may seem stringent at first. However, we should keep in mind that the assumptions are on the degrees only, which is one of the simplest properties of a graph. The theorem needs to take into consideration the “worst” graph for a given degree sequence. It is possible to achieve sharper results by assuming more about the link structure of the graph.

Theorem 6.3.7. *Let $\{Y_i, i \in \mathcal{V}\}$ be i.i.d. random variables on graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose degrees are $\{d_1, \dots, d_n\}$. Assume $|Y_i| \leq c < \infty$ a.s. $\forall i$, $\text{vol}(\mathcal{G}) = \mathcal{O}(n)$ and $\sum_i d_i^2 = \mathcal{O}(n)$. As $n \rightarrow \infty$,*

(a) *If $n^{-3/2} \sum_i d_i^7 \rightarrow 0$ and $n^{-2} \sum_i d_i^{10} \rightarrow 0$, then*

$$\frac{\Omega(\mathbf{Y}) - \mathbb{E}\Omega(\mathbf{Y})}{\text{sd}(\Omega(\mathbf{Y}))} \implies \mathcal{N}(0, 1);$$

(b) *If $n^{-3/2} \sum_i d_i^{11/2} \rightarrow 0$ and $n^{-2} \sum_i d_i^8 \rightarrow 0$, then*

$$\frac{\tilde{\Omega}(\mathbf{Y}) - \mathbb{E}\tilde{\Omega}(\mathbf{Y})}{\text{sd}(\tilde{\Omega}(\mathbf{Y}))} \implies \mathcal{N}(0, 1).$$

Proof. We prove the result in (a) and the similar proof for (b) can be found in Appendix B.5.

Following the proof of Lemma 6.3.4, we write $\Omega(\mathbf{Y}) = \sum_i X_i$, where $|X_i| \leq 2c^2 d_i$ and $\sigma^2 \equiv \text{var}(\Omega(\mathbf{Y})) = \mathcal{O}(\sum_i d_i^2)$ as shown in (6.8) and (6.9). The dependency graph formed by X_i has degrees $d_i^* = d_i + \sum_{j \in \mathcal{S}_i} d_j$, where $\mathcal{S}_i = \{j \in \mathcal{V} : j \sim i\}$. We are now ready to apply Lemma 6.3.6 with the centered and scaled variables $(X_i - \mathbb{E}X_i)/\sigma$.

First,

$$\begin{aligned} \sum_i \sigma^{-3} \mathbb{E} |X_i - \mathbb{E}X_i|^3 d_i^{*2} &\leq \sigma^{-3} (4c^2)^3 \sum_i d_i^3 (d_i + \sum_{j \in \mathcal{S}_i} d_j)^2 \\ &= 64c^6 \sigma^{-3} \sum_i \left(d_i^5 + 2 \sum_{j \in \mathcal{S}_i} d_i^4 d_j + \sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{S}_i} d_i^3 d_j d_k \right) \\ &= \mathcal{O}(\sigma^{-3} \sum_i d_i^7), \end{aligned}$$

where the last step follows from lemmas in B.4.1 and B.4.2. Similarly,

$$\begin{aligned} \sum_i \sum_{j \in \mathcal{S}_i} \sigma^{-3} \mathbb{E} |X_i - \mathbb{E}X_i|^3 d_j^* &= 64c^6 \sigma^{-3} \sum_i \left(\sum_{j \in \mathcal{S}_i} d_i^3 d_j + \sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{S}_j} d_i^3 d_k \right) \\ &= \mathcal{O}(\sigma^{-3} \sum_i d_i^6), \end{aligned}$$

and

$$\begin{aligned} &\sum_i \sigma^{-4} \mathbb{E} |X_i - \mathbb{E}X_i|^4 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* + \sum_{k \in \mathcal{S}_i^*} d_i^* d_k^* \right) \\ &\leq \sigma^{-4} (4c^2)^4 \sum_i d_i^4 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* + \sum_{k \in \mathcal{S}_i^*} d_i^* d_k^* \right) \\ &= \sigma^{-4} \mathcal{O} \left(\sum_i d_i^4 \sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* \right) \\ &= \sigma^{-4} \mathcal{O} \left(\sum_i d_i^4 \sum_{l \in \mathcal{S}_i} \sum_{j \in \mathcal{S}_l} \sum_{m \in \mathcal{S}_j} \sum_{k \in \mathcal{S}_m} \sum_{r \in \mathcal{S}_k} d_r \right) \\ &= \mathcal{O}(\sigma^{-4} \sum_i d_i^{10}), \end{aligned}$$

where the last step follows by applying (B.7) from Appendix repeatedly.

Putting together the three pieces and the assumption that $\sigma^2 = \mathcal{O}(n)$, we have shown that the upper bound in (6.14) goes to 0, and hence the result in (a) follows. \square

Since the Erdős-Rényi random graphs have Poisson degree distributions with finite moments, the conditions in Theorem 6.3.7 are always satisfied. Unfortunately, this is usually not the case when the degrees follow the power-law

$$\mathbb{P}(x) \propto x^{-\gamma}. \tag{6.15}$$

Clearly, the asymptotic normality holds if $\gamma > 11$ or $\gamma > 9$ for (a) and (b) respectively. However, real graphs tend to have a smaller γ as shown in Bornholdt and Schuster [2003]. The following corollary obtains a better condition on γ using Theorem 8.9 in Durrett [2004] page 66.

Corollary 6.3.8. *If $\{d_i, i \in \mathcal{V}\}$ are i.i.d. and follow the power-law distribution in (6.15), then the conditions in Theorem 6.3.7 are equivalent to*

$$(a') \quad \gamma > 6,$$

$$(b') \quad \gamma > 5.$$

Proof. Denote $d_{i,m} = d_i^m$ for $m > 1$. Then

$$\mathbb{P}(d_{i,m} = x) \propto x^{-\frac{m+\gamma-1}{m}}.$$

Clearly, $\mathbb{E}(d_{i,m})$ is finite if and only if $\gamma > m + 1$. When $\mathbb{E}(d_{i,m}) = \infty$, it follows from Theorem 8.9 in Durrett [2004] that $\lim_n n^{-q} \sum_{i=1}^n d_{i,m} = 0$ if and only if $\sum_{k=1}^{\infty} \mathbb{P}(d_{i,m} \geq k^q) < \infty$ for $q > 1$. We can compute

$$\sum_{k=1}^{\infty} \mathbb{P}(d_{i,m} \geq k^q) \propto \sum_{k=1}^{\infty} k^{\frac{q(1-\gamma)}{m}}.$$

It is well known that the sum of this infinite series converges if and only if $q(1-\gamma)/m < -1$, i.e. $\gamma > 1 + m/q$. Plugging in the appropriate m and q , we get the desired result. \square

6.4 Conclusion

Many recently developed semi-supervised learning methods for data on graphs assume that the feature of interest varies smoothly along the edges of the graph, and such smoothness has been used to motivate the Laplacian based regularization. Surprisingly, we have found that by randomly permuting the true labels on the graph, we can consistently get smoother answers in some real datasets. These examples strongly suggest inadequacy of the smoothness measures in many situations. To this end, we proposed the z -score in (6.3) to quantify how well the smoothness assumption holds. In an attempt to provide theoretical justifications for our proposal and also for the empirical observations, we have shown that both the unnormalized and normalized smoothness measures obey a central limit theorem under certain conditions on the node degrees.

If it is shocking that randomizing labels can give smoother results, then it is equally unsettling that how well the smoothness assumption stands is not indicative of performance. In some further empirical experiments, we compared the two smoothness measures from the following perspectives: the z -scores and the prediction errors. To our surprise, we found that the measure with a better z -score does not always perform better in terms of prediction. We have not yet been able to find a satisfactory explanation. We finish by pointing out that it is important to understand the smoothness properties of the data and to start with an assumption that truly reflects these properties.

Appendix A

Table of symbols and notation

As a general rule, we use capital letters to denote random variables or matrices. Vectors are always column vectors and are in bold, e.g. $\mathbf{1} = (1, 1, \dots, 1)^T$. We sometimes suppress the limits on summation signs if it is clear in the context, e.g. \sum_i often means $\sum_{i=1}^n$.

Here is a non-exclusive list of symbols and notation frequently used in the thesis:

Notation	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	graph with vertex set \mathcal{V} and edge set \mathcal{E}
$n = \mathcal{V} $	size of \mathcal{V}
$W = (w_{ij})_{i,j=1}^n$	adjacency matrix
$\text{vol}(\mathcal{G})$	$= \sum_{i=1}^n \sum_{j=1}^n w_{ij}$, total weights
w_{+i}	in-degree of i
w_{i+}	out-degree of i
d_i	in/out-degree of i for undirected \mathcal{G}
P_{ij}	transition probability
π_i	stationary distribution
Π	$\text{diag}(\boldsymbol{\pi})$
Δ	unnormalized graph Laplacian
$\tilde{\Delta}$	normalized graph Laplacian
s_{ij}	similarity between i and j
w'_{ij}	$= (\pi_i P_{ij} + \pi_j P_{ji})/2$, new graph weights assumed by random walk smoothing; \mathcal{G}' , d'_i , Δ' and $\tilde{\Delta}'$ are defined accordingly using w'_{ij} instead of w_{ij} .
A^+	Moore-Penrose generalized inverse of matrix A
A^T	transpose of matrix A

Y_i	response variable at node i , observed at $i = 1, \dots, r$
Z_i	prediction at i (or signal)
Y_i^*	$= Y_i$ if observed; $= \mu_i$ (default) otherwise
$\Omega(\mathbf{Y}), \tilde{\Omega}(\mathbf{Y})$	roughness measure
λ	tuning parameter (or inverse of noise variance)
X	predictor, e.g. $\sqrt{\boldsymbol{\pi}}$ or $\mathbf{1}$
Σ	signal covariance matrix
R	correlation matrix
$V = \text{diag}(\mathbf{v})$	relative univariate standard deviations
$\rho(\cdot)$	correlation function

Appendix B

Proofs

B.1 Closest positive semi-definite matrix

Lemma B.1.1. *For any arbitrary symmetric matrix A with eigen decomposition $A = UHU^T$, the matrix $A_+ = UH_+U^T$ is the closest positive semidefinite matrix to A in Frobenius norm, where $H_+ = \max(H, 0)$.*

Proof. First for any square matrix B , since $U^TU = I$, we have

$$\begin{aligned}\|UBU^T\|_F^2 &= \text{Tr}((UBU^T)(UBU^T)^T) \\ &= \text{Tr}(UBB^TU^T) \\ &= \text{Tr}(BB^T) \\ &= \|B\|_F^2.\end{aligned}\tag{B.1}$$

Let X be any positive semidefinite matrix. Denote $\tilde{X} = U^TXU$, then $X = U\tilde{X}U^T$.

$$\begin{aligned}\|A - X\|_F^2 &= \|UHU^T - X\|_F^2 \\ &= \|UHU^T - U\tilde{X}U^T\|_F^2 \\ &= \|H - \tilde{X}\|_F^2 \\ &= \sum_{i \neq j} \tilde{X}_{ij}^2 + \sum_i (H_{ii} - \tilde{X}_{ii})^2,\end{aligned}\tag{B.2}$$

where the third equality follows from (B.1). Since X is positive semidefinite, \tilde{X} is also positive semidefinite. Therefore $\tilde{X}_{ii} \geq 0$ for all i . It's now easy to see that the \tilde{X} that minimizes (B.2) is

$$\tilde{X}_{ij}^* = \delta_{(i=j)} \cdot \max(H_{ii}, 0) = \max(H_{ij}, 0) = H_+,$$

since H is diagonal. Correspondingly, $X^* = \operatorname{argmin}_{X \geq 0} \|A - X\|_F^2$, where $X^* = UH_+U^T$ is the defined A_+ . \square

B.2 Proof of Lemma 6.3.4, (6.7)

Proof. The proof follows similarly as that for (6.6). We first decompose $\tilde{\Omega}(\mathbf{Y})$ into sum of dependent variables:

$$\tilde{\Omega}(\mathbf{Y}) = Y^T \tilde{\Delta} Y = \sum_i Y_i^2 - \sum_i \sum_j \frac{w_{ij}}{\sqrt{d_i d_j}} Y_i Y_j \equiv \sum_i \tilde{X}_i,$$

where $\tilde{X}_i = Y_i^2 - \sum_j w_{ij} / \sqrt{d_i d_j} Y_i Y_j$. Note that \tilde{X}_i 's have the same dependency graph as the X_i 's defined for $\Omega(\mathbf{Y})$. Again, the result readily follows from Theorem 6.3.2 after we show that $|\tilde{X}_i|$ is bounded a.s. $\forall i$ and $\operatorname{var}(\tilde{\Omega}(\mathbf{Y})) = \mathcal{O}(n)$.

Notice that

$$\sum_j \frac{w_{ij}}{\sqrt{d_i d_j}} = \frac{1}{\sqrt{d_i}} \sum_j \frac{w_{ij}}{\sqrt{d_j}} \leq \frac{1}{\sqrt{d_i}} d_i = \sqrt{d_i}. \quad (\text{B.3})$$

Therefore

$$|\tilde{X}_i| \leq |Y_i^2| + \left| Y_i \sum_j \frac{w_{ij}}{\sqrt{d_i d_j}} Y_j \right| \leq (\sqrt{d_i} + 1)c^2. \quad (\text{B.4})$$

Applying Lemma 6.3.3 with $A = \tilde{\Delta}$ and noting that $\mathbf{a} = \mathbf{1}$, we have

$$\begin{aligned} \operatorname{var}(\tilde{\Omega}(\mathbf{Y})) &= (\mu_4 - 3\mu_2^2)n + 2\mu_2^2 \left(n + \sum_i \sum_j \frac{w_{ij}^2}{d_i d_j} \right) \\ &\quad + 4\mu_2 \mu_1^2 \sum_i \left(1 - \sum_j \frac{w_{ij}}{\sqrt{d_i d_j}} \right)^2 + 4\mu_3 \mu_1 \sum_i \left(1 - \sum_j \frac{w_{ij}}{\sqrt{d_i d_j}} \right) \\ &= (\mu_4 - \mu_2^2 + 4\mu_2 \mu_1^2 + 4\mu_3 \mu_1)n + 2\mu_2^2 \sum_i \sum_j \frac{w_{ij}^2}{d_i d_j} \\ &\quad + 4\mu_2 \mu_1^2 \sum_i \left(\sum_j \frac{w_{ij}}{\sqrt{d_i d_j}} \right)^2 - (8\mu_2 \mu_1^2 + 4\mu_3 \mu_1) \sum_i \sum_j \frac{w_{ij}}{\sqrt{d_i d_j}} \\ &= \mathcal{O}(n). \end{aligned}$$

The last equality follows from (B.3) and

$$\sum_i \sum_j \frac{w_{ij}^2}{d_i d_j} = \sum_i \frac{1}{d_i} \sum_j \frac{w_{ij}^2}{d_j} \leq \sum_i \frac{1}{d_i} d_i = n.$$

□

B.3 Proof of Theorem 6.3.5, (6.11)

Proof. The proof follows similarly as that for (6.10).

From (B.4), $\text{var}(\tilde{\Omega}(\mathbf{Y})) = \mathcal{O}(\sum_i d_i) = \mathcal{O}(n + n^\epsilon) = \mathcal{O}(n)$. We separate out the terms involving node k ,

$$\begin{aligned} \frac{\tilde{\Omega}(\mathbf{Y}) - \mathbb{E}\tilde{\Omega}(\mathbf{Y})}{\sqrt{n}} &= \frac{\tilde{\Omega}(\mathbf{Y}_{-k}) - \mathbb{E}\tilde{\Omega}(\mathbf{Y}_{-k})}{\sqrt{n}} \\ &+ \frac{1}{\sqrt{n}} \sum_i w_{ki} \left(\left(\frac{Y_k}{\sqrt{d_k}} - \frac{Y_i}{\sqrt{d_i}} \right)^2 - \mathbb{E} \left(\frac{Y_k}{\sqrt{d_k}} - \frac{Y_i}{\sqrt{d_i}} \right)^2 \right) \end{aligned} \quad (\text{B.5})$$

By Lemma 6.3.4 the first term converges to a normal distribution. We now show that the second term in (B.5) is $o_p(1)$. First,

$$\begin{aligned} \frac{1}{\sqrt{n}} \sum_i w_{ki} \left(\frac{Y_k}{\sqrt{d_k}} - \frac{Y_i}{\sqrt{d_i}} \right)^2 &= \frac{1}{\sqrt{n}} \sum_i w_{ki} (Y_k^2/d_k - 2Y_k Y_i/\sqrt{d_k d_i} + Y_i^2/d_i) \\ &= \frac{Y_k^2}{\sqrt{n}} - \frac{2Y_k}{\sqrt{n}\sqrt{d_k}} \sum_i \frac{w_{ki}}{\sqrt{d_i}} Y_i + \frac{1}{\sqrt{n}} \sum_i \frac{w_{ki}}{d_i} Y_i^2. \end{aligned} \quad (\text{B.6})$$

After centering, the third term in (B.6) becomes

$$\frac{1}{\sqrt{n}} \sum_i \frac{w_{ki}}{d_i} (Y_i^2 - \mathbb{E}Y_i^2) = \sqrt{\frac{s_k}{n}} \cdot \frac{1}{\sqrt{s_k}} \sum_i \frac{w_{ki}}{d_i} (Y_i^2 - \mathbb{E}Y_i^2) = o_p(1).$$

where $s_k = \sum_i w_{ki}^2/d_i^2 \leq d_k$. The last equality follows because $s_k/n \rightarrow 0$ and

$$\frac{1}{\sqrt{s_k}} \sum_i \frac{w_{ki}}{d_i} (Y_i^2 - \mathbb{E}Y_i^2) \implies \mathcal{N}(0, \text{var}(Y_i^2))$$

by Billingsley [1995], Theorem 27.2. The second term in (B.6) goes to 0 because

$$\frac{Y_k}{\sqrt{n}\sqrt{d_k}} \sum_i \frac{w_{ki}}{\sqrt{d_i}} Y_i = \frac{\sqrt{d_k} Y_k}{\sqrt{n}} \left(\frac{1}{d_k} \sum_i \frac{w_{ki}}{\sqrt{d_i}} Y_i \right) = \frac{\sqrt{d_k} Y_k}{\sqrt{n}} \cdot O_p(1) = o_p(1).$$

The first term in (B.6) goes to 0 trivially because Y_i is bounded.

Collecting the three parts, we have proved that the second term in (B.5) is $o_p(1)$, and hence the desired result. \square

B.4 Supporting lemmas for Section 6.3.3

Lemma B.4.1. For $x_1, \dots, x_m > 0$, $a_1, \dots, a_m > 0$ and $a = \sum_{i=1}^m a_i$,

$$\prod_{i=1}^m x_i^{a_i} \leq \sum_{i=1}^m \frac{a_i}{a} x_i^a \quad (\text{B.7})$$

Proof. Applying Jensen's inequality with the concave function $\log(\cdot)$, we have

$$\log\left(\sum_{i=1}^m \frac{a_i}{a} x_i^a\right) \geq \sum_{i=1}^m \frac{a_i}{a} \log(x_i^a) = \sum_{i=1}^m \log(x_i^{a_i}) = \log\left(\prod_{i=1}^m x_i^{a_i}\right).$$

\square

Lemma B.4.2. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with degree sequence $\{d_1, d_2, \dots, d_n\}$. Let $\mathcal{S}_i = \{j \in \mathcal{V} : j \sim i\}$ be the set of nodes connected to node i . Then,

$$\sum_{j \in \mathcal{S}_i} 1 = d_i, \quad (\text{B.8})$$

$$\sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{S}_i} 1 = d_i^2, \quad (\text{B.9})$$

$$\sum_i \sum_{j \in \mathcal{S}_i} d_i d_j \leq \sum_i d_i^3, \quad (\text{B.10})$$

$$\sum_i \sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{S}_j} d_k = \sum_i \sum_{j \in \mathcal{S}_i} d_i d_j \leq \sum_i d_i^3. \quad (\text{B.11})$$

Proof. The first two equalities are trivial. We now show (B.10) holds by applying

(B.7) and then by symmetry,

$$\sum_i \sum_{j \in \mathcal{S}_i} d_i d_j \leq \frac{1}{2} \sum_i \sum_{j \in \mathcal{S}_i} d_i^2 + d_j^2 = \sum_i \sum_{j \in \mathcal{S}_i} d_i^2 = \sum_i d_i^3.$$

We show the first half of (B.11), and the second half follows from (B.10).

$$\begin{aligned} \sum_i \sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{S}_j} d_k &= \sum_{j \in \mathcal{V}} \sum_{i: j \in \mathcal{S}_i} \sum_{k \in \mathcal{S}_j} d_k \\ &= \sum_{j \in \mathcal{V}} \sum_{i: i \in \mathcal{S}_j} \sum_{k \in \mathcal{S}_j} d_k \\ &= \sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{S}_j} d_k \sum_{i \in \mathcal{S}_j} 1 \\ &= \sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{S}_j} d_k d_j. \end{aligned}$$

□

Lemma B.4.3. *Let $\{X_i, i \in \mathcal{V}^*\}$ be random variables having a dependency graph \mathcal{G}^* with degrees $\{d_1^*, d_2^*, \dots, d_n^*\}$. Let $\mathcal{S}_i^* = \{j \in \mathcal{V}^* : j \sim i\}$ be the set of nodes connected to node i . Assume $\mathbb{E}X_i = 0$ and $\mathbb{E}(\sum_i \sum_{j \in \mathcal{S}_i^*} X_i X_j) = 1$. Then,*

$$\mathbb{E} \left(\sum_i \sum_{j \in \mathcal{S}_i^*} X_i X_j \right)^2 \leq 1 + 2 \sum_i \mathbb{E}X_i^4 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* + \sum_{k \in \mathcal{S}_i^*} d_i^* d_k^* \right) \quad (\text{B.12})$$

Proof. First,

$$\mathbb{E} \left(\sum_i \sum_{j \in \mathcal{S}_i^*} X_i X_j \right)^2 = \mathbb{E} \sum_{\{i,j\}\{k,l\}} X_i X_j X_k X_l + \mathbb{E} \sum_{\{i,j,k,l\}} X_i X_j X_k X_l,$$

where $j \in \mathcal{S}_i^*$ and $l \in \mathcal{S}_k^*$ in both sums, but $\{i, j, k, l\}$ form a connected set whereas $\{i, j\}$ and $\{k, l\}$ are disconnected. Notice that

$$\mathbb{E} \sum_{\{i,j\}\{k,l\}} X_i X_j X_k X_l = \mathbb{E} \left(\sum_i \sum_{j \in \mathcal{S}_i^*} X_i X_j \right) \mathbb{E} \left(\sum_k \sum_{l \in \mathcal{S}_k^*} X_k X_l \right) = 1,$$

and

$$\mathbb{E} \sum_{\{i,j,k,l\}} X_i X_j X_k X_l \leq \frac{1}{4} \mathbb{E} \sum_{\{i,j,k,l\}} X_i^4 + X_j^4 + X_k^4 + X_l^4 = \sum_{\{i,j,k,l\}} \mathbb{E}X_i^4,$$

where the last step follows by symmetry.

For a fixed node i , $\{i, j\}$ and $\{k, l\}$ are connected only if the four nodes have at least one of the following connections: $i \sim j \sim k \sim l$ or $j \sim i \sim k \sim l$, where k and l are interchangeable. This notation includes the case where the four tuple includes only 2 or 3 distinct nodes. Therefore,

$$\begin{aligned} \sum_{\{i,j,k,l\}} \mathbb{E}X_i^4 &\leq 2 \sum_i \mathbb{E}X_i^4 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} \sum_{l \in \mathcal{S}_k^*} 1 + \sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_i^*} \sum_{l \in \mathcal{S}_k^*} 1 \right) \\ &= 2 \sum_i \mathbb{E}X_i^4 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* + \sum_{k \in \mathcal{S}_i^*} d_i^* d_k^* \right) \end{aligned}$$

Collecting the two pieces we get (B.12). \square

B.5 Proof of Theorem 6.3.7, (b)

Proof. Following the proof of Lemma 6.3.4, we write $\tilde{\Omega}(\mathbf{Y}) = \sum_i \tilde{X}_i$, where $|\tilde{X}_i| \leq (\sqrt{d_i} + 1)c^2 \leq 2c^2\sqrt{d_i}$ and $\sigma^2 \equiv \text{var}(\tilde{\Omega}(\mathbf{Y})) = \mathcal{O}(n)$. The dependency graph formed by \tilde{X}_i has degrees $d_i^* = d_i + \sum_{j \in \mathcal{S}_i} d_j$. We are now ready to apply Lemma 6.3.6 with the centered and scaled variables $(\tilde{X}_i - \mathbb{E}\tilde{X}_i)/\sigma$.

First,

$$\begin{aligned} \sum_i \sigma^{-3} \mathbb{E} \left| \tilde{X}_i - \mathbb{E}\tilde{X}_i \right|^3 d_i^{*2} &\leq \sigma^{-3} (4c^2)^3 \sum_i d_i^{3/2} (d_i + \sum_{j \in \mathcal{S}_i} d_j)^2 \\ &= 64c^6 \sigma^{-3} \sum_i d_i^{3/2} \left(d_i^2 + 2 \sum_{j \in \mathcal{S}_i} d_i d_j + \sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{S}_i} d_j d_k \right) \\ &= \mathcal{O}(\sigma^{-3} \sum_i d_i^{11/2}), \end{aligned}$$

where the last step follows from Lemma B.4.1 and B.4.2. Similarly,

$$\begin{aligned} \sum_i \sum_{j \in \mathcal{S}_i} \sigma^{-3} \mathbb{E} \left| \tilde{X}_i - \mathbb{E}\tilde{X}_i \right|^3 d_j^* &= 64c^6 \sigma^{-3} \sum_i \left(\sum_{j \in \mathcal{S}_i} d_i^{3/2} d_j + \sum_{j \in \mathcal{S}_i} \sum_{k \in \mathcal{S}_j} d_i^{3/2} d_k \right) \\ &= \mathcal{O}(\sigma^{-3} \sum_i d_i^{9/2}), \end{aligned}$$

and

$$\begin{aligned}
& \sum_i \sigma^{-4} \mathbb{E} |X_i - \mathbb{E} X_i|^4 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* + \sum_{k \in \mathcal{S}_i^*} d_i^* d_k^* \right) \\
& \leq \sigma^{-4} (4c^2)^4 \sum_i d_i^2 \left(\sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* + \sum_{k \in \mathcal{S}_i^*} d_i^* d_k^* \right) \\
& = \sigma^{-4} \mathcal{O} \left(\sum_i d_i^2 \sum_{j \in \mathcal{S}_i^*} \sum_{k \in \mathcal{S}_j^*} d_k^* \right) \\
& = \sigma^{-4} \mathcal{O} \left(\sum_i d_i^2 \sum_{l \in \mathcal{S}_i} \sum_{j \in \mathcal{S}_l} \sum_{m \in \mathcal{S}_j} \sum_{k \in \mathcal{S}_m} \sum_{r \in \mathcal{S}_k} d_r \right) \\
& = \mathcal{O}(\sigma^{-4} \sum_i d_i^8),
\end{aligned}$$

where the last step follows by applying (B.7) repeatedly.

Putting together the three pieces and the assumption that $\sigma^2 = \mathcal{O}(n)$, we have shown that the upper bound in (6.14) goes to 0, and hence the result in (b) follows. \square

Bibliography

- Albert, R., Jeong, H., and Barabási, A. L. (1999). Diameter of the World-Wide Web. *Nature*, 400:130–131.
- Argyriou, A., Herbster, M., and Pontil, M. (2005). Combining graph Laplacians for semi-supervised learning. In *Advances in Neural Information Processing Systems 18*, pages 67–74. MIT Press.
- Baldi, P. and Rinott, Y. (1989). On normal approximations of distributions in terms of dependency graphs. *The Annals of Probability*, 17(4):1646–1650.
- Belkin, M., Matveeva, I., and Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. In Shawe-Taylor, J. and Singer, Y., editors, *COLT*, volume 3120 of *Lecture Notes in Computer Science*, pages 624–638. Springer.
- Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434.
- Berman, A. and Plemmons, R. J. (1994). *Nonnegative matrices in the mathematical sciences*. Academic Press, New York .:
- Billingsley, P. (1995). *Probability and Measure*. Wiley-Interscience, 3 edition.
- Bornholdt, S. and Schuster, H. G., editors (2003). *Handbook of Graphs and Networks: From the Genome to the Internet*. John Wiley & Sons, Inc., New York, NY, USA.
- Chung, F. R. K. (1997). *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society.
- Cressie, N. (1993). *Statistics for spatial data*. Wiley, New York.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, UK.

- Cvetković, D. M., Doob, M., and Sachs, H. (1980). *Spectra of graphs : theory and application*. Academic Press, New York ; London .
- Delalleau, O., Bengio, Y., and Le Roux, N. (2006). Large-scale algorithms. In Chapelle, O., Schölkopf, B., and Zien, A., editors, *Semi-Supervised Learning*, pages 333–341. MIT Press.
- Diaconis, P. and Evans, S. N. (2002). A different construction of gaussian fields from markov chains: Dirichlet covariances. *Annales de l'Institut Henri Poincaré (B) Probability and Statistics*, 38(6):863 – 878.
- Diggle, P. J. and Ribeiro, P. J. (2006). *Model-based Geostatistics*. Springer.
- Durrett, R. (2004). *Probability: Theory and Examples*. Duxbury Press, 3 edition.
- Erdős, P. and Rényi, A. (1959). On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297.
- Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the Internet topology. In *SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251–262, New York, NY, USA. ACM.
- Gilbert, E. N. (1959). Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144.
- Golub, G. H., F., C., and Loan, V. (1996). *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.
- Hall, P., Fisher, N. I., and Hoffmann, B. (1994). On the nonparametric estimation of covariance functions. *The Annals of Statistics*, 22(4):2115–2134.
- Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354.
- Heaton, T. J. and Silverman, B. W. (2008). A wavelet- or lifting-scheme-based imputation method. *Journal Of The Royal Statistical Society Series B*, 70(3):567–587.
- Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, pages 1090–1098.
- International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945.

- Jansen, M., Nason, G. P., and Silverman, B. W. (2009). Multiscale methods for data on graphs and irregular multidimensional situations. *Journal Of The Royal Statistical Society Series B*, 71(1):97–125.
- Jebara, T., Wang, J., and Chang, S.-F. (2009). Graph construction and b-matching for semi-supervised learning. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 441–448, New York, NY, USA. ACM.
- Jeong, H., Mason, S. P., Barabasi, A. L., and Oltvai, Z. N. (2001). Lethality and centrality in protein networks. *Nature*, 411(6833):41–42.
- Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabasi, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654.
- Johnson, R. and Zhang, T. (2007). On the effectiveness of Laplacian normalization for graph semi-supervised learning. *J. Mach. Learn. Res.*, 8:1489–1517.
- Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632.
- Koehler, J. (1990). *Design and estimation issues in computer experiments*. PhD thesis, Stanford University.
- Kondor, R. I. and Lafferty, J. (2002). Diffusion kernels on graphs and other discrete structures. In *In Proceedings of the ICML*, pages 315–322.
- Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society of South Africa*, 52:119–139.
- Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282.
- Larsen, R. M. (1998). Lanczos bidiagonalization with partial reorthogonalization. Technical report, Department of Computer Science, Aarhus University.
- Leenders, R. T. A. J. (2002). Modeling social influence through network autocorrelation: constructing the weight matrix. *Social Networks*, 24(1):21 – 47.
- Lehmann, E. L. and Casella, G. (2003). *Theory of Point Estimation (Springer Texts in Statistics)*. Springer.

- Leskovec, J., Adamic, L. A., and Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Trans. Web*, 1(1):5.
- Leskovec, J. and Faloutsos, C. (2007). Scalable modeling of real graphs using kronecker multiplication. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 497–504, New York, NY, USA. ACM.
- Leskovec, J. and Horvitz, E. (2008). Worldwide buzz: Planetary-scale views on an instant messaging network. In *Proc. 17th International World Wide Web Conference*.
- Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.
- Marsden, P. and Friedkin, N. (1993). Network studies of social influence. *Sociological Methods and Research*, 22:1:127–151.
- Mohar, B. (1997). Some applications of Laplace eigenvalues of graphs. In *Graph Symmetry: Algebraic Methods and Applications, volume 497 of NATO ASI Series C*, pages 227–275. Kluwer.
- O’Hagan, A. and Kingman, J. F. C. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42.
- Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.
- Richardson, M., Agrawal, R., and Domingos, P. (2003). Trust management for the semantic web. In *In proceedings of the second international semantic web conference*, pages 351–368. Springer Berlin / Heidelberg.
- Rinott, Y. (1994). On normal approximation rates for certain sums of dependent random variables. *J. Comput. Appl. Math.*, 55(2):135–143.
- Robinson, G. K. (1991). That BLUP is a good thing: the estimation of random effects. *Statistical Science*, 6:15–32.
- Schlather, M. (1999). Introduction to positive definite functions and to unconditional simulation of random fields. Technical report, Dept. Maths and Stats, Lancaster University, Lancaster, UK.
- Seber, G. A. F. and Lee, A. J. (2003). *Linear Regression Analysis*. Wiley Series in Probability and Statistics. Wiley-interscience.

- Shapiro, A. and Botha, J. D. (1991). Variogram fitting with a general class of conditionally nonnegative definite functions. *Comput. Stat. Data Anal.*, 11(1):87–96.
- Smola, A. J. and Kondor, I. R. (2003). Kernels and regularization on graphs. In *Proceedings of the Annual Conference on Computational Learning Theory*.
- Spielman, D. A. and Teng, S.-H. (2003). Solving sparse, symmetric, diagonally-dominant linear systems in time $O(m^{1.31})$. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 416, Washington, DC, USA. IEEE Computer Society.
- Stein, C. (1986). *Approximate Computation of Expectations*, volume 7 of *Lecture Notes–Monograph Series*. IMS.
- Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York.
- Taskar, B., Wang, M., Abbeel, P., and Koller, D. (2003). Link prediction in relational data. In *in Neural Information Processing Systems (NIPS)*.
- Tsang, I. W. and Kwok, J. T. (2006). Large-scale sparsified manifold regularization. In *Advances in Neural Information Processing Systems (NIPS) 19*.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442.
- Xu, Y. (2010). An investigation of smoothness measures for semi-supervised learning on graphs. Technical report, Department of Statistics, Stanford University.
- Xu, Y., Dyer, J. S., and Owen, A. (2010). Empirical stationary correlations for semi-supervised learning on graphs. *Annals of Applied Statistics, to appear*.
- Zhou, D., Bousquet, O., Lal, T., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. In *NIPS*, volume 16, pages 321–328, Cambridge, MA, USA. MIT Press.
- Zhou, D., Huang, J., and Schölkopf, B. (2005a). Learning from labeled and unlabeled data on a directed graph. In *the 22nd ICML*, pages 1041 – 1048.
- Zhou, D., Schölkopf, B., and Hofmann, T. (2005b). Semi-supervised learning on directed graphs. In *NIPS*, volume 17, pages 1633–1640, Cambridge, MA, USA. MIT Press.
- Zhu, X. (2005a). Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

- Zhu, X. (2005b). *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA. Chair-Lafferty, John and Chair-Rosenfeld, Ronald.
- Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. In *In ICML*, pages 912–919.