

Spectral Methods for Data Analysis

Frank McSherry

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2004

Program Authorized to Offer Degree: Computer Science & Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Frank McSherry

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

Anna Karlin

Reading Committee:

Dimitris Achlioptas

Paul Beame

Anna Karlin

Date: _____

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Bell and Howell Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Spectral Methods for Data Analysis

by Frank McSherry

Chair of Supervisory Committee:

Professor Anna Karlin
Computer Science & Engineering

“Spectral methods” captures generally the class of algorithms which cast their input as a matrix and employ linear algebraic techniques, typically involving the eigenvectors or singular vectors of the matrix. Spectral techniques have had much success in a variety of data analysis domains, from text classification [26] to website ranking [59, 47]. However, little rigorous analysis has been applied to these algorithms, and we are left without a firm understanding of why these approaches work as well as they do.

In this thesis, we study the application of spectral techniques to data mining, looking specifically at those problems on which spectral techniques have performed well. We will cast each problem into a common mathematical framework, giving a unified theoretical justification for the empirical success of spectral techniques in these domains. Specifically, we present models that justify the prior empirical success of spectral algorithms for tasks such as object classification, web site ranking, and graph partitioning, as well as new algorithms using these techniques for as of yet underdeveloped data mining tasks such as collaborative filtering. We will then take the understanding from this common framework and use it to unify several spectral results in the random graph literature. Finally, we will study several techniques for extending the practical applicability of these techniques, through computational acceleration, support for incremental calculation, and deployment in a completely decentralized environment.

TABLE OF CONTENTS

Chapter 1: Introduction	1
1.1 A Prototypical Case of Spectral Analysis	2
1.2 Data Mining Problems	4
1.3 A General Data Mining Model	10
1.4 Implementation of Spectral Methods	11
1.5 Contributions	13
Chapter 2: Linear Algebra Background	15
2.1 Linear Algebra Introduction	15
2.2 Matrix Perturbation Theory	18
2.3 Random Vectors and Matrices	26
2.4 Conclusions	28
Chapter 3: Data Mining	30
3.1 A Model for Structured Data	30
3.2 Data Cleaning	33
3.3 Information Retrieval	35
3.4 Collaborative Filtering	38
3.5 Conclusions	40
Chapter 4: Web Search	42
4.1 Overview and Discussion	42
4.2 Additional Related Work	43
4.3 The Model	44
4.4 The Algorithm: <i>SmartyPants</i>	47

4.5	Discussion and Extensions	51
4.6	Conclusions	52
Chapter 5:	Graph Partitioning	53
5.1	Spectral Graph Partitioning	59
5.2	A Traditional Spectral Result	60
5.3	Combinatorial Projections	61
5.4	Observations and Extensions	67
Chapter 6:	Eigencomputation	69
6.1	Accelerated Eigencomputation	70
6.2	Incremental Eigencomputation	74
6.3	Decentralized Eigencomputation	76
6.4	Conclusions	84
Chapter 7:	Conclusions	85
7.1	Contributions	85
7.2	Future Research	89
	Bibliography	94

ACKNOWLEDGMENTS

Much of the research in this thesis has been previously published, or has publication pending. The primary areas of overlap are Chapter 3 (from [7]), Chapter 4 (from [1]), Chapter 5 (from [56] and [23]), Section 6.1 (from [2]), and Section 6.3 (from [45]). I give gracious acknowledgment to my coauthors, Dimitris Achlioptas, Yossi Azar, Anirban Dasgupta, Amos Fiat, John Hopcroft, Anna Karlin, David Kempe, Jared Saia, and Bernhard Schoelkopf, both for their technical contributions to this thesis and for being utterly enjoyable collaborators.

Particular acknowledgment needs to be given to a few specific individuals. Dimitris Achlioptas has been an excellent colleague, providing advice and obstinacy on areas both technical and non. John Hopcroft and Jon Kleinberg gave me the opportunity to work with them at Cornell in a postdoctoral capacity, leading to a wealth of new colleagues and research. Finally, Anna Karlin has been beyond incredible as an advisor. I am completely indebted to her for the time she has spent with and positive influence exerted over me. I hope that over time I am able to pass this investment on to others.

Additionally, Paul Beame is due many thanks for reading, proofing, and commenting on this dissertation, which is greatly improved by his efforts.

Chapter 1

INTRODUCTION

World wide communication networks have caused dramatic growth in the size of data corpora, and with it a corresponding, if not greater, increase in the importance of understanding these corpora. Unfortunately, these two trends have been at odds, as most data analyses scale poorly with the input size, thus motivating the design of efficient automated approaches to understanding data. The need for understanding is pervasive; wherever there is data, there is something to be gained by better understanding how that data is structured. Companies seek to understand what drives customers to purchase products. Web users wish to quickly find useful information online. Movie-goers would enjoy useful (or at least provably good) movie recommendations.

These objectives have motivated a great deal of research aimed at answering the following types of questions: How should data be stored and organized so as to allow the most effective retrieval of information? How can “important” structure and “meaningful” patterns be found within a large data set? How and when can this hidden structure be used to help predict missing data or to “clean” data that is imprecise or partially incorrect? Furthermore, can these tasks be performed efficiently enough that we might use them for even the largest data sets?

Spectral techniques have proven, at least empirically, effective in answering some of these questions [26, 48, 47]. The moniker “spectral techniques” refer to approaches which analyze the eigenvectors or singular vectors of a matrix derived from the data. While the data sets themselves vary, many, if not most, can be worked into a matrix form. For example, we can transform a collection of text documents into a *terms* \times *documents* matrix, where entry A_{ij} indicates that term i occurs in document j . Further examples of interest include: i)

the web graph, using A_{ij} to indicate the presence of a link from site i to site j , ii) utility matrices, using entry A_{ij} to describe the utility (value) of object i to person j , and iii) the buddy graph, using entry A_{ij} to indicate whether person i is a friend of person j . There are certainly further examples, as we will see later, but these alone should indicate the capacity of matrices to represent rich and interesting data sets.

The rest of this chapter will proceed as follows: We start in Section 1.1 giving an example of a spectral algorithm applied to a toy problem domain. In so doing, we will highlight, in a casual manner, the key properties that enable spectral methods. In Section 1.2 we consider several domains where spectral algorithms have met with noted success, detailing for each domain the difficulties to overcome, and the impact of spectral methods. In Section 1.3 we lay out a general framework for spectral data analysis, highlighting the important role of latent linear structure and attempting to further characterize those problems that may be solved by spectral methods. Section 1.4 highlights several new algorithmic enhancements resulting from lessons learned about the behavior of spectral algorithms, focusing on extending the application of these techniques to domains that were previously inaccessible. Finally, before heading into the bulk of the text, Section 1.5 details the contributions of this thesis, and provides a roadmap of the chapters to come.

1.1 A Prototypical Case of Spectral Analysis

We start out getting our feet wet by examining the application of spectral analysis to a toy domain, learning mixtures of high dimensional Gaussians distributions. To call this a “toy” domain is a bit unfair, as many researchers are actively interested in this problem; mixtures of Gaussians are used as first approximations to mixtures of arbitrary distributions, which frequently serve as good approximations to real data. Nonetheless, the mathematics of the problem are so properly matched to the needs of spectral analysis as to appear contrived.

A mixture of n dimensional Gaussians defines a distribution on points in R^n : There exist k “mean” vectors μ_i in the n dimensional space, and each data point is produced by choosing a mean vector μ_i uniformly at random, and adding independent, normally distributed noise to each coordinate. We produce a matrix of samples \hat{A} by filling each column of \hat{A} with

a sample from this distribution. Given such a \hat{A} , our goal is to determine the μ_i used to generate each of the samples.

The data produced by a mixture of Gaussians has two key properties that solicit the use of spectral techniques, the two we will ultimately identify as central to spectral analysis.

1. There is a small set of vectors such that every column in the matrix of expectations, denoted A , can be written as a weighted sum of these vectors. In a mixture of Gaussians, the expectation of each column is one of the k vectors μ_i , and these vectors themselves serve as the small set. A matrix whose columns can be written as sums of k vectors is called *rank k* , and such matrices will figure heavily into the understanding of spectral methods.
2. The difference between the observed data \hat{A} and its expectation A is a matrix of independent random variables of modest variance. Such matrices, as we will see in later chapters, are very poorly approximated if you insist that the approximation be described by only a few vectors. Even the very best rank k approximation to $\hat{A} - A$ will only capture an asymptotically small fraction of the entries in $\hat{A} - A$.

On an input matrix \hat{A} , the prototypical spectral algorithm computes a rank k matrix $\hat{A}^{(k)}$ which satisfies

$$\|\hat{A} - \hat{A}^{(k)}\| \leq \min_{\text{rank}(D)=k} \|\hat{A} - D\| \quad (1.1)$$

That is, it computes a rank k matrix which best approximates the input data, using an as of yet undefined norm (the minimization occurs at the same point unique point for most interesting norms, and is unique for most matrices). The algorithm then uses the approximation $\hat{A}^{(k)}$ as a surrogate for the original input A , performing one of many clustering algorithms (k-means, nearest neighbor, or some other clustering technique). As we will see in future chapters, nearly all of the columns of $\hat{A}^{(k)}$ will be concentrated around their means, which makes the clustering rather obvious.

Recall that the input matrix \hat{A} is comprised of two parts: structured data A and unstructured error $\hat{A} - A$. The former may be approximated well by a rank k matrix, while

the latter is poorly approximated by every rank k matrix. The implication, which we will characterize more formally over the course of this thesis, is that any rank k approximation to their sum, \widehat{A} , benefits most by investing in its approximation to the structured data, and gains little from attempting to approximate the unstructured error. Indeed, as we will eventually prove, the optimal rank k approximation will very nearly reflect the actual matrix of expectations. In the case of the mixture of Gaussians we will recover the matrix of the means, but more generally we recover the latent structure of A , through which we can solve many data mining problems.

1.2 Data Mining Problems

We will consider several data mining problems in this thesis, as well as a framework general enough to accommodate a large class of unexplored problems. The primary theme throughout is that of *matrix reconstruction*: “given a randomly perturbed instance \widehat{A} of a matrix A , when is it possible to recover properties of A ?” We will see that this question can capture several data mining problems, and the theoretical results that we achieve for each serve as justification for observed practical success and motivation for the unexplored domains.

We start by looking at several of the problem domains that data miners face, the peculiarities of each domain that makes the task difficult, and characterize the results that our new framework provides.

1.2.1 Information Retrieval

One of the most fundamental data mining tasks is object clustering and classification. A natural example of the object classification problem is embodied in the task of the search engine which must, upon presentation of several key terms, establish a collection of documents which are somehow relevant to the terms presented. Naturally, the notion of *relevance* is ill-defined, typically being redefined as the provable result of a presented algorithm.

Traditionally, web search (and information retrieval in general) have measured relevance or similarity between documents by the cosine of the angle between their vector representations. The similarity of two documents is therefore a function of the number of terms

shared by two documents. Such systems have long suffered from the complementary issues of

- **Synonymy:** Two words may have the same meaning, though to the similarity metric they are distinct. Documents titled “Car Care” and “Automobile Maintenance” are supposed similar, though their diction is disjoint. In the vector model, “Car” is no more similar to “Auto” than it is to “Goat”, and these two titles above are declared dissimilar.
- **Polysemy:** Single words may have multiple meanings, resulting in their presence in documents which are dissimilar. The term “Mouse” occurs both in computer parlance as well as in vermin verbiage, and as such has the undesired effect of indicating relevance between documents on the two different topics.

A technique which has proven empirically effective at overcoming these issues is *latent semantic analysis* (LSA) [26]. LSA uses spectral techniques to reduce the dimensionality of the term-document matrix, computing an optimal rank k approximation to it. It appears that the low dimensional space resulting from applying LSA is strongly identified with the underlying (latent) semantic categories which characterize the data. Once documents are characterized in terms of their actual semantics, rather than overly particular or ambiguous terms, the issues of synonymy and polysemy fade.

The empirical success of LSA has, until recently, been without rigorous prediction and explanation. An important first step towards a theoretical understanding of LSA was taken when Papadimitriou et al [60] introduced a probabilistic model for document generation with latent semantics, and showed that a document corpus generated according to this model yields sharply defined topic clusters in the optimal rank k approximation, with high probability. Limitations of their model include the fact that both documents and terms are assumed to be nearly “pure”, each associated with a single topic. This leaves us with documents that discuss only one topic, and terms that are by assumption not polysemous.

In Section 3.3 We extend their results to a more general probabilistic model for corpus generation which admits both impure documents and polysemous terms. In fact, we allow

each document and term to be associated with arbitrary combinations of topics. We show that even in this very general setting, spectral analysis identifies the underlying topics and is able to determine relevance based on this more meaningful characterization, avoiding synonymy and polysemy. This model, though clearly less complex than actual document generation, gives us an explanation for why LSA addresses synonymy and polysemy, and begins to characterize the settings in which spectral methods succeed.

1.2.2 Collaborative Filtering

A fundamental problem in data mining, typically referred to as “collaborative filtering” or “recommendation systems”, is to use partial information about the preferences of a group of users to make recommendations or predictions regarding the unobserved preferences. For example, a movie recommendation system might recommend “Happiness” to a person who enjoyed “American Beauty” or “Alice in Wonderland” to a person who enjoyed “The Phantom Tollbooth”, based on the general tendency of people who like the latter tend to enjoy the former. More generally, collaborative filtering can be viewed as the problem of using an incomplete data set to determine properties of the complete data (perhaps reconstructing absent entries, though there are other, less ambitious goals).

Intuitively, little can be done to reconstruct missing entries if there is no structure underlying the data. In Section 3.4 we take the previously mentioned notion of *latent semantics* and apply it in this domain. We assume that there is an underlying collection of reasons, analogous to “topics” in LSA, that a person might enjoy a particular product. Specifically, we assume that the utility matrix A , where A_{ij} entry represents the value of object i to person j , is a [nearly] low-rank matrix. Under this assumption, we present an algorithm that given access to a random subset of the entries of A , constructs a matrix whose mean squared error with respect to A vanishes as m and n increase.

To our knowledge, there has been very little prior work on the design, analysis, and evaluation of collaborative filtering algorithms other than the work of Kumar et al [53] who took the important first step of defining an analytic framework for evaluating collaborative filtering. They also presented algorithms for a clustered model of utility where an object’s

utility to an individual is a function only of the individual and the cluster to which the object belongs. In particular, they demonstrate algorithms with provable bounds for the two cases that either (a) the data is clustered into k clusters whose composition is known or (b) each item belongs to one of two clusters, whose compositions are not necessarily known.

The cluster model of Kumar *et al.* is one example of a low-rank utility matrix. Our collaborative filtering algorithms handle significantly more general utility matrices – anything with a good low rank approximation. For such utility matrices, we are able to accurately predict the bulk of the missing utilities. No clustering or *a priori* knowledge of object similarity is required; the assumption of latent semantics imposes sufficient structure on the data.

1.2.3 Web Site Ranking

Kleinberg’s seminal paper on hubs and authorities [47] introduced a natural paradigm for classifying and ranking web pages, setting off an avalanche of subsequent work [17, 18, 16, 27, 55, 19, 10, 21, 39, 7, 13, 6, 25]. Kleinberg’s ideas were implemented in the HITS algorithm as part of the CLEVER project [17, 16]. Around the same time, Brin and Page [14, 59, 38] developed a highly successful search engine, Google [37], which orders search results according to PageRank, a measure of authority of the underlying page. Both approaches rely on linkage information to rank pages while using, chiefly, text matching to determine the candidate set of responses to a query.

Unfortunately, as noted in information retrieval, text matching can run afoul of synonymy and polysemy. While information retrieval techniques like LSA seem to address synonymy and polysemy, they do not make use of the link structure of the web. This form of peer review of web sites simply cannot be overlooked, as it is the primary source of information for distinguishing quality pages from those which merely contain the requisite terms.

In Chapter 4 we present a model and algorithm that simultaneously addresses both the IR problems of synonymy and polysemy, as well as understanding and exploiting the link structure of the web for ranking. Our model is, as before, based on latent semantics.

Web pages can be viewed as documents, as in the information retrieval setting, but their vocabulary is extended from terms alone to include addresses of other pages (representing links). This defines a unified model for the web content, describing how the link structure and text content of documents are created, and connecting the semantics of the two.

Our algorithm combines information retrieval and collaborative filtering, bypassing synonymy and polysemy to predict the pages that are most likely to appear with particular terms. We propose a query generation model in which a searcher presents text that is likely to be used as a link to the desired page, and prove that our algorithm produces search results that are arbitrarily close to the intended results as the number of query terms increases. This algorithm generalizes in scope the approaches of HITS and Google, and our proof gives analytic insight into why these algorithms perform well.

1.2.4 *Graph Partitioning*

For general graphs, the problems of finding noteworthy colorings, cliques, and bisections are well known to be NP-hard. As such, much literature has emerged on the average-case performance of algorithms for each problem, equivalently the performance of a fixed algorithm on a uniformly random graph, the belief being that worst instances are not rarely representative of realistic problem instances, and algorithms that work well “on average” might be of use in practice.

However, the vast majority of graphs do not have small colorings, large cliques, or bisections of merit, and it is ultimately not particularly illuminating to analyze the performance of algorithms on uniformly random graphs. Trivial algorithms perform essentially as well as sophisticated ones. To address this, the input distributions are skewed away from uniform, putting more weight on those graphs that exhibit the desired combinatorial object. Each average-case graph problem is therefore associated with a distribution over graphs, typically defined by a random process for generating graphs. These models produce graphs by independently including each edge with probabilities that are carefully tailored to ensure that the graph includes a particular combinatorial object.

Three problems that have been studied extensively are k-coloring, clique, and min bi-

section, where the associated graph models are:

- **k-Coloring:** Fix a k -coloring. Include each inter-color edge with probability p , and all others with probability 0.
- **Clique:** Fix a set of nodes for inclusion in the clique. Include each clique edge with probability 1, and all other edges with probability p .
- **Bisection:** Fix a bisection. Include each intra-part edge with probability q , and each inter-part edge with probability $p < q$.

For each problem, the goal of the algorithm is the recovery of the latent structure, be it coloring, clique, or bisection. The goal of the algorithm designer is to expand the range of parameters, in the form of p , q , and part sizes, for which such an algorithm exists.

The problem of graph bisection on random graphs has been studied for some time. Bui et al. [15] and Dyer and Frieze [29] have presented algorithms for bisecting dense graphs when $p < (1 - \epsilon)q$. Jerrum and Sorkin [41] consider an approach based on simulated annealing, but using a single constant temperature. Boppana [12] presents a spectral algorithm which succeeds for a large range of parameters (we will see them in Chapter 5), but his approach requires the solution to a convex optimization problem. Condon and Karp [22] analyzed a strictly combinatorial algorithm for partitioning which achieves nearly the same range of parameters as [12], and runs in linear time.

Similarly, many researchers have worked on the problem of coloring random graphs which have k -colorings. Kucera [51], Turner [70], and Dyer and Frieze [29] present algorithms that optimally color uniformly random k -colorable graphs for fixed k , with high probability. However, most k -colorable graphs are dense, and therefore easier to color than sparse graphs (while this may appear counterintuitive, observe that the more dense the graph, the more evidence there is for the predetermined coloring). Blum and Spencer [11] and Alon and Kahale [4] demonstrate algorithms that color random sparse graphs properly with high probability, the latter using a spectral algorithm.

The problem of finding a large clique in a random graph was suggested by Karp in [43]. Kucera observed in [52] that when the size of the clique is $\omega(\sqrt{n \log n})$ and $p = 1/2$, the clique members are simply the vertices of highest degree. Alon, Krivelevich, and Sudakov [5] showed that a planted clique of size $\Omega(\sqrt{n})$ can be found through spectral techniques when $p = 1/2$.

For each of these problems, specialized spectral approaches have proved successful, defining the frontier of feasible parameters. In Chapter 5 we present a generic spectral algorithm which succeeds for each of the problems above, as well as a more general problem of partitioning graphs based on inter-part edge densities. For the most part we are able to replicate the bounds achieved by previous approaches, falling short only in cases where domain specific optimizations are employed. In each case we are able to replicate the bounds generated by the purely spectral aspects of the corresponding approach.

We will perform this generalization by considering the general problem of finding hidden partitions in graphs. In particular, we look at random graphs defined by a partition ψ such that the presence of an edge (or more generally, the weight on an edge) is a random variable whose expectation is defined exactly by the parts of the endpoints. Note that this problem encompasses the problems of finding hidden cliques, k -colorings, and bisections in otherwise random graphs, and the model generalizes the specialized models above. The fundamental observation is that in such a model the expected matrix has low rank, and the graph formation process can be viewed as the addition of independent, mean zero error.

1.3 A General Data Mining Model

We will unify all of the data mining problems discussed thus far into a general framework. Specifically, we assume that there is some pure, meaningful data matrix A that is the target of our analysis. The information we as data miners have access to is a corrupted instance of this data, $\hat{A} = A + E$, where E is some form of problem dependent error. In the case of Object Classification, Web Site Ranking, and Graph Partitioning E captures error due to sampling from the probability distributions modeling the data, whereas in the case of Collaborative Filtering, E captures the omission of some fraction of the entries of A .

Intuitively, some property of A must distinguish it sufficiently from E to have any hope of meaningful reconstruction. In this thesis, data will be considered *structured* if there exists a meaningful low rank approximation to the data, equivalently if A can be described by latent linear semantics. We will argue that if A is structured and E is unstructured we can recover many useful properties of A . Furthermore, we will show that for reasonable definitions of the previous data mining problem problems, E will fit our definition of unstructured.

A pressing question is then “Why should A be structured?” In general data is believed to be structured, otherwise data mining would be a fruitless effort, so perhaps a better question is “Why should A fit this definition of structure?” This question is fundamentally outside the scope of this dissertation, but intuitive arguments abound. The concept of a few latent dimensions which characterize the general trends of data is appealing in its simplicity. Furthermore, *linear* correlation is sufficiently rich to capture many coarse trends in data, notable co-occurrence of items as well as antipathy (mutual non co-occurrence) of items.

Of course, the real justification is empirical. All data sets that we have examined have significant eigenstructure, in that their eigenvalues drop off quickly, implying that most of their *meaning* is to be found in a small subspace. As well, the empirical successes of Latent Semantic Indexing, HITS and Google, and spectral clustering, each of which consider *only* the optimal low rank approximation to the data, are proof positive that such corpora have meaningful low rank representations.

1.4 Implementation of Spectral Methods

As well as crafting models and algorithms for several data analysis domains, we also present three novel approaches to extending the applicability of spectral techniques generally.

1.4.1 Accelerated Computation

Spectral techniques fundamentally rely on the computation of the most significant singular vectors of the input matrix. This task is expensive, requiring time that is super-linear, if only by a logarithmic factor. Nonetheless, when a single pass over the data can not be performed by a single computer in less than a day, logarithmic factors are sufficient to

render an approach infeasible.

In Section 6.1 we will develop and analyze algorithms for computing low rank approximations to a matrix, which, while not optimal, are nearly as good as the optimal approximation. Our approach is based on our observation that unstructured error does not much affect the optimal low rank approximation: by introducing cleverly sculpted random (unstructured) error to the data, we can greatly accelerate the computation of the optimal approximation. Sparsifying the input by randomly discarding entries of A accelerates the computation, as well as reducing the memory footprint of the process. Randomly quantizing the entries of the input can reduce the representation of each to a single bit, again compressing the data as well as simplifying the arithmetic operations that need to be performed. In each case, the analysis we have developed argues that the optimal approximations to these perturbed matrices are not dissimilar to the optimal approximation to the input matrix.

1.4.2 Incremental Computation

A redeeming quality of most large data sets is that they are somewhat stable; changes tend to be cosmetic and it is rare for the latent structure to change completely in a short period of time. Our analyses indicates that happenstance changes, those not due to a systematic shift in the data's semantics, exert little influence on the optimal rank k approximation. We might therefore hope for incremental implementations of our algorithms which make good use of prior results, and whose performance is characterized by the degree of change in the latent structure since the previous application.

Indeed this is possible. In Section 6.2 we will analyze the simplest of algorithms and see that it accommodates incremental updates quite naturally. As before, the heart of our analysis will be that slight changes to the data (viewed as error) do not much affect the optimal low rank approximations. After a slight perturbation, the new optimal approximation is not far from our old approximation, and we are able to find it far more quickly than if we had started from scratch.

1.4.3 *Decentralized Computation*

One of the most daunting aspects of analyzing large data sets is the need to collect the data to a central location and there analyze it. Considering that the web graph has several billions of pages, simply collecting and storing these pages requires a tremendous investment. Were this not enough of an issue, there are many interesting networks that cannot be as easily collected as the web graph, in which one simply requests the edges (links) associated with each page. Consider a peer to peer network, where each node is connected only to other nodes that it trusts. In this domain one can not simply ask each node to transmit its list of neighbors; even if it were possible to directly communicate with a given node, each values its privacy and only intends to communicate with other nodes that it knows and trusts.

In Section 6.3 we will present an algorithm that solicits the participation of the nodes in the graph to be analyzed, and by their communication and computation establishes the optimal low rank approximation to the adjacency matrix defined by the network. By fortuitous coincidence, the data flow in the traditional method for computing low rank approximations to a graph's adjacency matrix exactly mirrors the communication links present in the graph. Our algorithm simply prompts the initiation of a very simple data dissemination process, occasionally interrupting to redirect the nodes. After a polylogarithmic number of rounds, the participants collectively converge to the optimal approximation. As the result is now stored in a decentralized fashion, the data can be harvested if needed or, as we will see is often the case, the data can simply be used locally, obviating any need for centralization.

1.5 *Contributions*

The contributions of this thesis are threefold in the extension of the domains to which spectral techniques can be soundly applied. Through modeling data and problems, we introduce several new domains which when viewed properly become natural targets for spectral analysis. Through rigorous proof we further the understanding for why such techniques succeed, enabling more successful formulation of problems and extending the confidence with which one might apply these techniques. Finally, through algorithmic enhancements we extend the possibility of applying spectral techniques to impractically large, incrementally updated,

and inconveniently distributed data sets.

Chapter 2

LINEAR ALGEBRA BACKGROUND

This thesis is largely about linear trends in data, exhibited when the data is cast as a matrix. As such, we now cover some linear algebra basics and develop the tools we will use in subsequent chapter. The text of Golub and Van Loan [36] serves as a superior resource for algorithmic linear algebra, while the excellent text of Stewart and Sun [68] focuses specifically on matrix perturbation theory. Both are ideal choices for those with a deeper interest in perturbation theory than is presented here, and would serve as excellent companions for this chapter.

This chapter will be broken into three sections. We start by covering some notational and introductory material about linear algebra, focusing on spectral features of matrices. We then develop several perturbation theoretic tools that we will use throughout the text to bound the degree to which perturbations may influence spectral structure. Finally, we study the behavior of random vectors and matrices, specifically in the context of their relation to the perturbation theory just developed, concluding that random matrices of a particular flavor have a nominal impact on many spectral properties.

2.1 Linear Algebra Introduction

We refer to matrices frequently in this thesis, as they are the primary objects of interest, and so we use capital letters (A) to denote them. Single subscripts (A_j) denote column j in the matrix A , and double subscripts (A_{ij}) denote the entry in row i and column j of A . All matrices we consider will be of dimension $m \times n$, with m rows and n columns. In general m need not equal n , though in some special cases (which will be noted) the matrix may be square, and even symmetric.

The reader is probably familiar with the standard Euclidean norm for vectors:

$$\|v\|_2 = \left(\sum_j v_j^2 \right)^{1/2}$$

We will typically drop the subscript, as this is the only vector norm we will use. We also use two matrix norms: the L_2 norm and the Frobenius norm, defined respectively as

$$\|A\|_2 = \max_{|v|=1} \|Av\| \quad \text{and} \quad \|A\|_F = \left(\sum_{i,j} A_{ij}^2 \right)^{1/2}$$

Proposition 1 *For any matrices A and B ,*

$$\|AB\|_2 \leq \|A\|_2 \|B\|_2 \quad \text{and} \quad \|AB\|_F \leq \|A\|_2 \|B\|_F$$

The proofs of these inequalities are fairly elementary linear algebra exercises, and would serve as an excellent warm up to this section.

2.1.1 The Singular Value Decomposition

Spectral methods as we will consider them are based upon computing the *singular value decomposition* of a matrix. This is a manner of representing a matrix that reveals which subspaces are most important to a matrix, and to what degree.

Theorem 2 (Jordan) *Every matrix A can be written as*

$$A = U\Sigma V^T$$

where U and V are orthonormal matrices, and Σ is a diagonal matrix whose entries σ_i are non-negative and non-increasing with i .

The elements on the diagonal of Σ are called the singular values (σ_i), and each has an associated pair of left and right singular vectors, U_i and V_i respectively. The *rank* of a matrix is the number of non-zero singular values.

An alternate way to write the decomposition, which reveals the association between the singular values and singular vectors is

$$A = \sum_i \sigma_i U_i V_i^T \tag{2.1}$$

This formulation describes the matrix as a weighted sum of rank 1 matrices. The associated weights are the corresponding singular values.

Proposition 3 For any matrix A , $\|A\|_2 = \sigma_1$ and $\|A\|_F = (\sum_i \sigma_i^2)^{1/2}$.

Proof. The L_2 norm and Frobenius norm are “unitarily invariant”; they do not change if a change of basis is applied to rows or columns of the matrix. By Proposition 1,

$$\begin{aligned}\|\Sigma\| &= \|U^T A V\| \leq \|U\|_2 \|A\| \|V\|_2 = \|A\| \\ \|A\| &= \|U \Sigma V^T\| \leq \|U\|_2 \|\Sigma\| \|V\|_2 = \|\Sigma\|\end{aligned}$$

We conclude that both $\|A\|_2 = \|\Sigma\|_2$ and $\|A\|_F = \|\Sigma\|_F$, and the proposition follows. \blacksquare

One important consequent of Proposition 3 that we will use frequently is the bound the L_2 norm imposes on the Frobenius norm.

Proposition 4 For any rank k matrix A , $\|A\|_2 \leq \|A\|_F \leq \sqrt{k}\|A\|_2$

This proposition follows directly from Proposition 3, noting that a rank k matrix has only k non-zero singular values.

2.1.2 Low Rank Approximations

The central linear algebraic approach throughout this thesis is the construction of the *optimal rank k approximation* to a matrix. The singular value decomposition above has organized subspaces for us, sorting them by their importances, the σ_i . It is perhaps natural that the approximation that we seek is simply the truncated summation of equation 2.1.

$$A^{(k)} = \sum_{i \leq k} \sigma_i U_i V_i^T \tag{2.2}$$

$A^{(k)}$ is the *best* rank k approximation to A in the following precise sense.

Theorem 5 (Optimality) For any matrix A and any rank k matrix D

$$\|A - A^{(k)}\|_2 \leq \|A - D\|_2 \quad \text{and} \quad \|A - A^{(k)}\|_F \leq \|A - D\|_F .$$

The proof of this theorem can be found in the proof of Theorem 2.5.3 in Golub and Van Loan [36].

We now look at an alternate characterization of $A^{(k)}$, as the projection of A onto an optimal subspace. If U is the matrix of left singular vectors of A , let us define the *optimal k dimensional projection* for each of A and A^T as

$$P_A^{(k)} = \sum_{i \leq k} U_i U_i^T \quad \text{and} \quad P_{A^T}^{(k)} = \sum_{i \leq k} V_i V_i^T$$

The matrix $P_A^{(k)}$ projects any vector onto the space spanned by the first k columns of U . Its operation can be viewed as first rewriting the vector in terms of the basis vectors U_i , and then discarding all but the first k coordinates, which are then transformed back to the original space.

As suggested above, one alternate characterization of $A^{(k)}$ is as the projection of the columns of A onto the space spanned by the first k left singular vectors.

$$A^{(k)} = P_A^{(k)} A = A P_{A^T}^{(k)}$$

To verify the equivalence, notice that the orthonormality of U causes $P_A^{(k)}$ to act as the identity on the first k terms of the sum of Equation 2.1, and causes it to zero out all other terms, resulting in Equation 2.2. The same holds for $P_{A^T}^{(k)}$, which projects rows onto the space spanned by the first k columns of V .

2.2 Matrix Perturbation Theory

Matrix perturbation theory is the study of the change in matrix properties as a function of additive error. Given two matrices A and B and a characterization of their difference $A - B$, how might we characterize the change in properties of the matrices? We are specifically interested in two properties of a matrix A : the optimal rank k approximation $A^{(k)}$ and the optimal k dimensional projection $P_A^{(k)}$.

We will see in this section that we are generally able to characterize the degree of change as a simple function of $\|A - B\|_2$, the L_2 norm of the perturbation matrix. We consider spectral perturbation bounds for three different types of matrices: matrices of low rank,

matrices with a gap in their singular values, and general matrices. As our assumptions become weaker and weaker, our ability to reconstruct $A^{(k)}$ and $P_A^{(k)}$ will diminish, giving us a broad spectrum of bounds to work with.

2.2.1 Perturbation of Low Rank Matrices

We start out with bounds for the case where the initial matrix A is rank k itself. In this setting A is already its own optimal approximation, and we would like to recover a good approximation to it.

Theorem 6 *Let A be a rank k matrix. For any matrix B of like dimensions,*

$$\|A - B^{(k)}\|_2 \leq 2\|A - B\|_2 \quad \text{and} \quad \|A - B^{(k)}\|_F \leq \sqrt{8k}\|A - B\|_2$$

Proof. We start by proving the L_2 bound. We apply the triangle inequality first, and then observe that as A is a rank k matrix, Theorem 5 bounds $\|B - B^{(k)}\|_2 \leq \|B - A\|_2$.

$$\begin{aligned} \|A - B^{(k)}\|_2 &\leq \|A - B\|_2 + \|B - B^{(k)}\|_2 \\ &\leq \|A - B\|_2 + \|B - A\|_2 \end{aligned}$$

These final two terms are equal, and we collect them into the stated bound.

To prove the Frobenius bound, we observe that as each of A and $B^{(k)}$ are rank k , the rank of their sum is at most $2k$. Proposition 4 gives us the inequality

$$\|A - B^{(k)}\|_F \leq \sqrt{2k}\|A - B^{(k)}\|_2$$

to which we apply our just proven L_2 bound. ■

2.2.2 Perturbation of Matrices with a Spectral Gap

We now move on to a less constrained class of matrices. Rather than assume that the matrix is rank deficient, in this section we will prove results for matrices which have reasonable gaps in their sequence of singular values. Throughout this section, we will use the notation

$$\delta_k(A) = \sigma_k(A) - \sigma_{k+1}(A)$$

to reference a gap in the spectrum of A . A reasonable gap between adjacent singular values corresponds to a sudden jump in the significance of the corresponding subspaces, distinguishing those above the gap from those below.

Perhaps the first spectral perturbation bound is that of Davis and Kahan [24]. Their approach is quite simple in spirit, and we will recreate the core of it here.

Theorem 7 (Davis & Kahan) *For any matrices A and B of like dimensions, for which $\sigma_k(A) > \sigma_{k+1}(B)$*

$$\|P_A^{(k)}(I - P_B^{(k)})\|_2 \leq \|A - B\|_2 / (\sigma_k(A) - \sigma_{k+1}(B))$$

Recall that $(I - P_B^{(k)})$ is the projection onto the space spanned by the bottom $n - k$ singular vectors of B , and so the theorem bounds the interplay between the most significant structure of A and the least significant structure of B .

Proof. We will start by proving the bound for symmetric matrices A and B , where $P_A^{(k)} = P_{A^T}^{(k)}$ and $P_B^{(k)} = P_{B^T}^{(k)}$. Notice that in this case

$$\begin{aligned} P_A^{(k)}(A - B)(I - P_B^{(k)}) &= P_A^{(k)}A(I - P_B^{(k)}) - P_A^{(k)}B(I - P_B^{(k)}) \\ &= AP_{A^T}^{(k)}(I - P_B^{(k)}) - P_A^{(k)}B(I - P_{B^T}^{(k)}) \\ &= AP_A^{(k)}(I - P_B^{(k)}) - P_A^{(k)}(I - P_B^{(k)})B \end{aligned}$$

At this point, we begin to consider $\|P_A^{(k)}(I - P_B^{(k)})\|_2$. Recall from its definition that the L_2 norm is defined by a unit vector which undergoes maximum stretch when multiplied by the matrix. Let x be a unit vector such that $|P_A^{(k)}(I - P_B^{(k)})x| = \|P_A^{(k)}(I - P_B^{(k)})\|_2$. Applying the triangle inequality to the equation above

$$|P_A^{(k)}(A - B)(I - P_B^{(k)})x| \geq |AP_A^{(k)}(I - P_B^{(k)})x| - |P_A^{(k)}(I - P_B^{(k)})Bx|$$

Note that $P_A^{(k)}(I - P_B^{(k)})x$ lies in the space spanned by the first k left singular vectors of A , and so when multiplied by A its norm increases by *at least* a factor of $\sigma_k(A)$. Likewise, $P_A^{(k)}(I - P_B^{(k)})$ annihilates any aspect of Bx that emerges on the top k left singular vectors

of B , and so $|P_A^{(k)}(I - P_B^{(k)})Bx| = |P_A^{(k)}(I - P_B^{(k)})(B - B^{(k)})x|$. Noting that $\|B - B^{(k)}\|_2 = \sigma_{k+1}(B)$, and with a norm bound on the left hand side, we arrive at

$$\|A - B\|_2 \geq \sigma_k(A)\|P_A^{(k)}(I - P_B^{(k)})\|_2 - \sigma_{k+1}(B)\|P_A^{(k)}(I - P_B^{(k)})\|_2$$

Some simple rearrangement of terms yields the bound

$$\|P_A^{(k)}(I - P_B^{(k)})\|_2 \leq \|A - B\|_2 / (\sigma_k(A) - \sigma_{k+1}(B))$$

which completes the proof for the case of symmetric A and B .

With the result proved for symmetric matrices, we now generalize it to arbitrary matrices. The generalization makes use of a construction typically credited to Jordan:

$$J(M) = \begin{bmatrix} 0 & M^T \\ M & 0 \end{bmatrix}$$

This construction is interesting for several reasons, mainly that while $J(M)$ is now a symmetric matrix, its singular value decomposition is essentially a reformulation of the singular value decomposition of M . As the reader may delight in verifying,

$$P_{J(M)}^{(2k)} = \begin{bmatrix} P_M^{(k)} & 0 \\ 0 & P_{M^T}^{(k)} \end{bmatrix} \quad (2.3)$$

Additionally, the non-zero singular values of $J(M)$ consist of exactly two occurrences of each singular value of M . Among other things, this implies that $\|J(M)\|_2 = \|M\|_2$.

As $J(A)$ and $J(B)$ are symmetric, we apply the bound we have proven thus far, yielding

$$\begin{aligned} \|P_{J(A)}^{(2k)}(I - P_{J(B)}^{(2k)})\|_2 &\leq \|J(A) - J(B)\|_2 / (\sigma_{2k}(J(A)) - \sigma_{2k+1}(J(B))) \\ &= \|A - B\|_2 / (\sigma_k(A) - \sigma_{k+1}(B)) \end{aligned}$$

By expanding $P_{J(A)}^{(2k)}$ and $P_{J(B)}^{(2k)}$ using Equation 2.3 we get the bound

$$\left\| \begin{bmatrix} P_A^{(k)}(I - P_B^{(k)}) & 0 \\ 0 & P_{A^T}^{(k)}(I - P_{B^T}^{(k)}) \end{bmatrix} \right\|_2 \leq \|A - B\|_2 / (\sigma_k(A) - \sigma_{k+1}(B))$$

from which we conclude the statement of the theorem for general matrices A and B . \blacksquare

While we will not use it directly in this thesis, the Davis and Kahan result can be generalized to handle subspaces of non-adjacent index.

Corollary 8 *For any matrices A and B of like dimensions, for i, j such that $\sigma_i(A) > \sigma_j(B)$,*

$$\|P_A^{(i)}(I - P_B^{(j-1)})\|_2 \leq \|A - B\|_2 / (\sigma_i(A) - \sigma_j(B))$$

Proof. The proof is conducted in a manner identical to the proof of Theorem 7 above. The only modification is that we lower bound the application of $A^{(i)}$ by $\sigma_i(A)$ and upper bound $\|B - B^{(j-1)}\|_2$ by $\sigma_j(B)$. ■

This generalization allows us to discuss the effect of a perturbation on a subspace by subspace basis. Those subspaces associated with larger singular values will be more accurately preserved, which is crucial as they contribute comparatively more to the Frobenius norm of A . While we do not use this result explicitly in this thesis, it forms the basis of many of the results we will see, though through reformulation its application is concealed from the reader.

Stewart's Theorem

Many papers on spectral methods invoke “Stewart’s theorem”, a fairly popular result about the stability of invariant subspaces. In fact, the generality of Stewart’s theorem is rarely necessary. The theorem involves a good deal of complexity to accommodate invariant subspaces of non-symmetric matrices. Nearly all spectral techniques either do not apply to non-symmetric matrices, or do so through the singular value decomposition, which is not covered by Stewart’s theorem.

The results of Davis and Kahan as stated in Theorem 7 will suffice for all of the results we detail in this thesis, and in fact are sufficient for all spectral techniques cited throughout the paper. We will restate their result in a form that Stewart’s theorem commonly takes, as this formulation is typically easier to apply.

Theorem 9 *For any matrices A and B of like dimensions for which $\delta_k(A) \geq \|A - B\|_2$,*

$$\|P_A^{(k)} - P_B^{(k)}\|_2 \leq 2\|A - B\|_2 / (\delta_k(A) - \|A - B\|_2)$$

While we will refer to this theorem as “Stewart’s Theorem”, this is due to its structure only. Stewart’s oft cited theorem addresses invariant subspaces of general matrices, not the spaces spanned by singular vectors, as are bounded here.

Proof. Note that we can write

$$P_A^{(k)} - P_B^{(k)} = P_A^{(k)}(I - P_B^{(k)}) - (I - P_A^{(k)})P_B^{(k)}$$

Theorem 7 bounds the norms of these two terms as

$$\begin{aligned} \|P_A^{(k)}(I - P_B^{(k)})\|_2 &\leq \|A - B\|_2 / (\sigma_k(A) - \sigma_{k+1}(B)) \\ \|P_B^{(k)}(I - P_A^{(k)})\|_2 &\leq \|A - B\|_2 / (\sigma_k(B) - \sigma_{k+1}(A)) \end{aligned}$$

We now apply a bound on the perturbation of singular values, drawn from the minimax definition of singular values (see Corollary 8.6.2 of Golub and Van Loan [36]).

$$|\sigma_k(A) - \sigma_k(B)| \leq \|A - B\|_2$$

which bounds each of the above terms by $\|A - B\|_2 / (\delta_k(A) - \|A - B\|_2)$. ■

Notice that this bound is more or less useless when $\delta_k(A) \leq \|A - B\|_2$. This is a very clear instance where the error we can tolerate is defined by the spectral gap in the data. To tolerate an arbitrary perturbation of size $\|A - B\|_2$ a gap twice as large needs to exist. We will frequently use this bound with the stronger assumption that $\delta_k(A) \geq 2\|A - B\|_2$, which allows us to remove the $\|A - B\|_2$ from the denominator at the expense of a factor of 2 in the numerator.

2.2.3 Perturbation of General Matrices

In this section we consider the perturbation of general matrices: those without any specific properties or constraining spectral structure. These results are somewhat different in flavor from the previous ones, as it is generally not possible to reconstruct $A^{(k)}$ after a slight perturbation. Instead, our bounds will describe the difference in *quality of approximation*, arguing that while the optimal rank k approximation to a perturbed input is not necessarily close to $A^{(k)}$, it is nearly as good at approximating A .

Theorem 10 (General L2) *For any matrices A and B of like dimensions,*

$$\|A - B^{(k)}\|_2 \leq \|A - A^{(k)}\|_2 + 2\|A - B\|_2$$

Proof. We start with a simple application of the triangle inequality, followed by an application of Theorem 5, observing that $\|B - B^{(k)}\|_2 \leq \|B - A^{(k)}\|_2$, followed by a final application of the triangle inequality:

$$\begin{aligned} \|A - B^{(k)}\|_2 &\leq \|A - B\|_2 + \|B - B^{(k)}\|_2 \\ &\leq \|A - B\|_2 + \|B - A^{(k)}\|_2 \\ &\leq \|A - B\|_2 + \|B - A\|_2 + \|A - A^{(k)}\|_2 \end{aligned}$$

Observing that $\|B - A\|_2 = \|A - B\|_2$, we collect the terms into the stated bound. \blacksquare

We now prove a similar result for the Frobenius bound. Lemma 12 will capture the spirit of the bound, and has a proof which parallels that of Theorem 10. We then do some symbol manipulation to put it in the more useable form of Theorem 13. Both Lemma 12 and Theorem 13 will make use of the following reinterpretation of Theorem 5.

Lemma 11 *For any matrices A and B of like dimensions*

$$\|P_A^{(k)} A\|_F \geq \|P_B^{(k)} A\|_F$$

Recalling that $P_A^{(k)} A = A^{(k)}$, this lemma asserts that $A^{(k)}$ captures as much Frobenius norm as any other k dimensional projection of A .

Proof. The Pythagorean equality states that for orthogonal vectors a, b $|a+b|^2 = |a|^2 + |b|^2$. As for any B and vector v , $P_B^{(k)} v$ and $(I - P_B^{(k)})v$ are orthogonal, we apply the Pythagorean equality to each column of A , yielding

$$\|A\|_F^2 = \|P_B^{(k)} A\|_F^2 + \|A - P_B^{(k)} A\|_F^2. \quad (2.4)$$

The second term of the right hand side is minimized at $B = A$, by the optimality of $A^{(k)}$ (Theorem 5). As the left hand side does not vary with B , the first term of the right hand side, $\|P_B^{(k)} A\|_F^2$, is maximized at $B = A$. \blacksquare

With this alternate optimality characterization in hand, we now describe the Frobenius difference that results from applying a suboptimal projection to A .

Lemma 12 *For any matrices A and B of like dimensions*

$$\|P_B^{(k)} A\|_F \geq \|P_A^{(k)} A\|_F - 2\|(A - B)^{(k)}\|_F .$$

Proof. In direct analogy to the proof of Theorem 10, we start by applying the triangle inequality, followed by an application of Lemma 11 to bound $\|P_B^{(k)} B\|_F \geq \|P_A^{(k)} B\|_F$, followed by a final application of the triangle inequality.

$$\begin{aligned} \|P_B^{(k)} A\|_F &\geq \|P_B^{(k)} B\|_F - \|P_B^{(k)}(A - B)\|_F \\ &\geq \|P_A^{(k)} B\|_F - \|P_B^{(k)}(A - B)\|_F \\ &\geq \|P_A^{(k)} A\|_F - \|P_A^{(k)}(B - A)\|_F - \|P_B^{(k)}(A - B)\|_F \end{aligned}$$

Lemma 11 bounds the $\|P_X^{(k)}(A - B)\|_F$ terms by $\|P_{(A-B)}^{(k)}(A - B)\|_F = \|(A - B)^{(k)}\|_F$. ■

We now use Lemma 12 to prove that if $\|(A - B)^{(k)}\|_F$ is small, then $\|A - B^{(k)}\|_F$ is not much larger than $\|A - A^{(k)}\|_F$. This theorem does little more than arrange the bound of Lemma 12 into a statement whose structure mirrors that of Theorem 10.

Theorem 13 (General Frobenius) *For any matrices A and B of like dimensions,*

$$\|A - B^{(k)}\|_F \leq \|A - A^{(k)}\|_F + 2\sqrt{\|(A - B)^{(k)}\|_F \|A^{(k)}\|_F} + \|(A - B)^{(k)}\|_F .$$

Proof. We start by first applying the triangle inequality, followed by an application of the matrix Pythagorean equality of Equation 2.4 to get

$$\begin{aligned} \|A - B^{(k)}\|_F &\leq \|A - P_B^{(k)} A\|_F + \|P_B^{(k)}(A - B)\|_F \\ &\leq (\|A\|_F^2 - \|P_B^{(k)} A\|_F^2)^{1/2} + \|P_B^{(k)}(A - B)\|_F \end{aligned}$$

To bound the right hand side, we invoke the lower bound for $\|P_B^{(k)} A\|_F$ provided by Lemma 12, dropping a $-4\|(A - B)^{(k)}\|_F^2$ term that we will not need.

$$\|A - B^{(k)}\|_F \leq \left(\|A\|_F^2 - \|P_A^{(k)} A\|_F^2 + 4\|(A - B)^{(k)}\|_F \|P_A^{(k)} A\|_F \right)^{1/2} + \|P_B^{(k)}(A - B)\|_F$$

We clean up this bound, applying the matrix Pythagorean equality again to reconstitute $\|A - A^{(k)}\|_F^2$ and finally the inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$

$$\begin{aligned} \|A - B^{(k)}\|_F &\leq \left(\|A - A^{(k)}\|_F^2 + 4 \|(A - B)^{(k)}\|_F \|A^{(k)}\|_F \right)^{1/2} + \|P_B^{(k)}(A - B)\|_F \\ &\leq \|A - A^{(k)}\|_F + \left(4 \|(A - B)^{(k)}\|_F \|A^{(k)}\|_F \right)^{1/2} + \|P_B^{(k)}(A - B)\|_F \end{aligned}$$

Lemma 11 then bounds $\|P_B^{(k)}(A - B)\|_F$ by $\|P_{(A-B)}^{(k)}(A - B)\|_F = \|(A - B)^{(k)}\|_F$. ■

2.3 Random Vectors and Matrices

Our one approach to random matrices is to project them to a lower dimensional subspace in an attempt to filter out the error introduced by the randomness. As such, we are naturally interested in how random matrices behave when projected to a fixed subspace, as well as how random matrices may influence the choice of optimal projection. In this section we will study these behaviors and bound the impact of each.

2.3.1 Random Vectors in Fixed Subspaces

We will frequently project random matrices onto fixed low dimensional subspaces, and we will need to characterize the result. As a matrix projection simply amounts to projecting each of its columns, we will study the projection of random vectors. Projecting a random vector onto a fixed vector amounts to an inner product, which in this setting results in a sum of many independent random variables. Standard concentration results can be used to bound the variance of these sums from their expectation. We will use a convenient vector reformulation of Azuma's inequality which appears as Theorem 4.16 in the text of Motwani and Raghavan [58].

Theorem 14 (Azuma's Inequality) *Let \hat{v} be a vector of independent random variables whose entries are bounded in magnitude by 1, and let $v = E[\hat{v}]$. For any unit vector u ,*

$$Pr[|\langle \hat{v}, u \rangle - \langle v, u \rangle| \geq \lambda] \leq 2e^{-\lambda^2/4}$$

We can easily extend this to multidimensional subspaces by bounding the projection onto each of the basis vectors:

Corollary 15 *Let \hat{v} be a vector of independent random variables whose entries are bounded in magnitude by 1, and let $v = E[\hat{v}]$. For any k dimensional projection $P_A^{(k)}$,*

$$\Pr[|P_A^{(k)}\hat{v} - P_A^{(k)}v| \geq \lambda\sqrt{k}] \leq 2ke^{-\lambda^2/4}$$

Proof. If the columns U_i form the basis of $P_A^{(k)}$, then for any vector x we may write

$$P_A^{(k)}x = \sum_{i \leq k} U_i U_i^T x$$

As each of the U_i are orthogonal,

$$|P_A^{(k)}x|^2 = \sum_{i \leq k} |U_i^T x|^2$$

Substituting $\hat{v} - v$ for x , we apply Theorem 14 to each inner product, and then a union bound on the probabilities to bound the sum above by $k\lambda^2$ with probability $2ke^{-\lambda^2/4}$. Taking square roots gives the stated bound. ■

These results are particularly compelling when we consider the distance from the random vector to its expectation before and after it is projected. A vector whose entries are independent random variables of constant magnitude may easily have squared distance to its expectation proportional to the dimension in which it lies. However, after projecting such a vector to a fixed k dimension space, we have reduced the squared distance to something which is now proportional to k , a tremendous dampening for small values of k and especially significant if the expected vector lies in the projected space, as its length decreases not at all.

2.3.2 The Influence of Random Matrices on Optimal Subspaces

We now study the other negative influence that randomness can have on our analyses: perturbing the k dimensional subspace that is optimal for the matrix. As observed in the previous section, we can bound this change by the L_2 norm of the perturbation. We now see a result of Furedi and Komlos which bounds the L_2 norm of certain random matrices.

Theorem 16 (Furedi and Komlos) *Let \widehat{A} be a $m \times n$ matrix whose entries are independent random variables with variance bounded by σ^2 , and absolute magnitude bounded by $\sigma(m+n)^{1/2}/8\log^3(m+n)$. Let $A = E[\widehat{A}]$. With probability at least $1 - 2\exp(-\log^6 n/2)$*

$$\|A - \widehat{A}\|_2 \leq 4\sigma\sqrt{m+n}$$

Proof. This proof is the combination of two results: chiefly the work of Furedi and Komlos [35], but enhanced by an elegant concentration result of Alon, Krivelevich, and Vu [49]. ■

It is worth noting that *any* matrix whose entries are σ in magnitude will have L_2 norm at least $\sigma\sqrt{\max(m,n)}$. This means that the matrix $A - \widehat{A}$ has, to within constant factors, the *smallest possible* L_2 norm of matrices with comparable entry magnitudes.

While the L_2 norm is central to our perturbation theory, it will be useful to have results in the Frobenius norm as well. We now extend the above result cosmetically, restating it as a bound on $\|(A - \widehat{A})^{(k)}\|_F$.

Corollary 17 *Using the notation and assumptions of Theorem 16, with high probability*

$$\|(A - \widehat{A})^{(k)}\|_F \leq 4\sigma\sqrt{k(m+n)}$$

Proof. We use the bound just proved in Theorem 16 combined with Proposition 4. We also adopt the convention, used throughout the remainder of the text, that $1 - 2\exp(-\log^6 n/2)$ represents a sufficiently high probability as to no longer concern us with its particular value. In situations where there are several sources of failure probability, this term will be dropped, as it is asymptotically smaller than any other probabilities that we consider. ■

Note that since $\|M\|_2 = \|M^{(1)}\|_F$, this result serves as a more general form of Theorem 16.

2.4 Conclusions

We have developed two very important classes of tools for our spectral analysis toolkit. First, we have established several perturbation bounds, noting that the L_2 norm consistently figures prominently in our perturbation bounds. Perturbations which exhibit small L_2

norm are unable to significantly perturb our optimal approximations, for various classes of matrices.

Second, we have noted that a very broad class of perturbations, random perturbations whose entries are independent, zero mean, and of moderate variance, have very limited L_2 norm. This will lead us to show, in various forms throughout the remaining chapters, the degree to which spectral methods succeed in the presence of random perturbations, bypassing an apparently large perturbation to retrieve a very accurate estimate of the optimal approximation to the original data.

Chapter 3

DATA MINING

In this chapter we look at the application of spectral techniques to various problems that fall under the general heading of “data mining”. We start by examining a model for producing data sets with rich spectral structure, as an introduction to the type of structure we hope to extract from data. We then tackle the problems of Data Cleaning, Information Retrieval, and Collaborative Filtering, for each defining a random process to model the data corruption process, as well as analyzing a spectral resolution to each.

3.1 A Model for Structured Data

We now discuss the issue of modeling the pure data matrix A . The reality of data mining is invariably that an observed matrix \hat{A} is simply presented, with little or no knowledge of what phenomena generated it. Nonetheless, it is interesting and important to study various models on which spectral techniques succeed, as they offer insight into why they succeed generally. Through understanding what is important in each of the settings we study, we learn about the capabilities of spectral methods and can more carefully frame problems and design algorithms to take advantage of them.

This said, we now consider a model that lends itself quite well to spectral analysis. It is important to note that at this point we only aim to model a matrix which describes the *actual* correspondence between rows and columns. We are not yet in the business of producing observed data; such problem specific models will come later in this chapter. Instead, we are interested in the unavailable matrix of complete, unperturbed entries which perfectly describe the relation of each row to each column.

Inherent in our model (and in nearly all models for structured data) is the idea that while there are some mn observable matrix entries, there is a set of significantly fewer *latent*

variables which describe the data. In our model, we start by associating a k -tuple u_i with row i and a k -tuple v_j with column j . As our first approximation to the true data, each entry A_{ij} will be set equal to the inner product of the k -tuples associated with row i and column j .

Low Rank Data($\{u_i\}, \{v_j\}$):

For each i, j , A_{ij} is defined as

$$A_{ij} = \langle u_i, v_j \rangle$$

At an intuitive level, the reader can imagine a set of k attributes, with each k -tuple describing the row or column's relation to the k attributes. To establish a value in the matrix, the relevant row and column establish their relationship to each other vis-a-vis each attribute, and accumulate the total.

To give a slightly more concrete feel to this model, let us consider three instantiations of this model in practical domains.

- **Movie Reviews:** The rows and columns of A correspond to movies and moviegoers, and entry A_{ij} represents the entertainment value of movie i to moviegoer j . We imagine that there are but a few facets of movie entertainment, perhaps in the forms of comedy, drama, action, romance, etc... Movies have an associated amount of content for each of these facets, measured in abstract “content” units. Viewers likewise have an appreciation for each facet, measured by the entertainment derived from each unit of content. A natural estimate of the aggregate entertainment drawn from a movie would be the sum of the entertainment from each facet, determined by the content provided times the appreciation for it.
- **Text Documents:** The rows and columns of A correspond to terms and documents, and entry A_{ij} represents the relevance of term i to document j . Again, imagine that in the document corpus there are only a few topics (perhaps our collection represents the union of Computer Science papers, of which there are arguably a small number of sub-disciplines). Each document may be characterized by the amount of space spent

discussing each topic, and each term by its relevance to each topic. The relevance of a term to the document as a whole might then be the sum of its relevances, weighted by the amount of content in the document.

- **Product Awareness:** The rows and columns of A correspond to products and consumers, and entry A_{ij} represents the number of times that consumer j is made aware of product i . Imagine that there are only a few channels of advertising: television, radio, billboards, etc.. Each product will have a certain amount of exposure in each of these channels. Likewise, each consumer will have a particular exposure to each channel. The number of times that the consumer hears about the product is then the sum of probabilities of hearing about it from each channel, which is proportional to the exposure of the product times the exposure of the person.

These instances are not designed to convince the reader that these models precisely capture what happens in constructing these data sets, but rather to demonstrate phenomena that lead to latent linear structure. In any realistic data, we expect that such correlations are at best guidelines, and while they may roughly describe the data, any individual entry is likely to vary from the intended value.

As such, we permit a certain degree of arbitrary modifications to our low rank matrix A , under the constraint that the change have small L_2 norm. Theorem 18 bounds the influence that error of small L_2 norm can exert on the spectral gap.

Theorem 18 *For matrices A and B of like dimension*

$$|\delta_k(A) - \delta_k(B)| \leq 2\|A - B\|_2$$

Proof. The proof is a simple application of the minimax definition of singular values, as used in the proof of Theorem 9. ■

This theorem intends to accommodate the possibility that while there is latent linear structure underlying our model, the matrix A does not need to be rank k . The data may still exhibit a strong spectral gap even after fleshing it out to a more reasonable matrix B . The constraint on $\|A - B\|_2$ reflects the intent that the “fleshing out” reflect local, arbitrary

decisions, rather than the global systematic changes that we captured in A . Recall from our discussion of random matrices that it is possible for a matrix to effect significant changes in the data entries without a large L_2 norm.

3.2 Data Cleaning

To ease our way into data mining results, we will start with a problem that fits very naturally into our framework. The use of spectral techniques to clean (or filter) data has a long history. In the signal processing domain, Fourier analysis projects a input signal onto a collection of basis vectors and retains those basis vectors on which there is significant projection. The discarded basis vectors typically correspond to high frequency noise, whose omission results in a simpler and, it is hoped, more accurate signal. We consider an analogous process in the matrix realm. By computing $\widehat{A}^{(k)}$ we extract from \widehat{A} only the most significant set of basis vectors, simplifying and, as we will see, clarifying the data.

We examine the situation where there is a true data matrix A which has become randomly corrupted to become the matrix \widehat{A} . Many reasons may exist for this corruption: inaccuracy of sensing devices, error due to limited numerical precision, or simple fluctuations in the data itself. The precise nature of the error does not concern us at this point, but we will impose three conditions: the errors should be i) independent, ii) zero mean, and iii) of bounded variance.

Data Cleaning Error(A):

Add independent, mean zero, error of at most unit variance and magnitude bounded independent of n to each entry of A .

The goal in data cleaning is to take such a perturbed matrix and produce a “cleaned” version of it; one that removes as much of the error as possible while retaining the important features of the original. For our purposes, the “important features” of a matrix A are captured by its optimal low rank approximation $A^{(k)}$, and recovering it is our goal.

Notice that the error introduced by this error process fits the requirements of Corollary 17 and has bounded L_2 norm, and ultimately a bounded influence on the optimal low rank

approximation.

Theorem 19 *Let \widehat{A} result from **Data Cleaning Error**(A). If $\epsilon = 16\sqrt{m+n}/\delta_k(A)$ is at most 2, then with high probability*

$$\|A^{(k)} - \widehat{A}^{(k)}\|_F \leq \epsilon \|A\|_F + 4\sqrt{k(m+n)}$$

Proof. Notice that we can write

$$\begin{aligned} \|A^{(k)} - \widehat{A}^{(k)}\|_F &= \|P_A^{(k)} A - P_{\widehat{A}}^{(k)} \widehat{A}\|_F \\ &\leq \|(P_A^{(k)} - P_{\widehat{A}}^{(k)})A\|_F + \|P_{\widehat{A}}^{(k)}(A - \widehat{A})\|_F \end{aligned}$$

Lemma 11 bounds $\|P_{\widehat{A}}^{(k)}(A - \widehat{A})\|_F \leq \|(A - \widehat{A})^{(k)}\|_F$, which gives us

$$\|A^{(k)} - \widehat{A}^{(k)}\|_F \leq \|P_A^{(k)} - P_{\widehat{A}}^{(k)}\|_2 \|A\|_F + \|(A - \widehat{A})^{(k)}\|_F$$

We now apply Corollary 17 with $\sigma = 1$ to bound with high probability

$$\|(A - \widehat{A})^{(k)}\|_F \leq 4\sqrt{k(m+n)}$$

With this bound and our assumption on the size of ϵ , we may apply Stewart's theorem (Theorem 9), bounding

$$\begin{aligned} \|A^{(k)} - \widehat{A}^{(k)}\|_F &\leq (4\|A - \widehat{A}\|_2/\delta_k(A))\|A\|_F + \|(A - \widehat{A})^{(k)}\|_F \\ &\leq (16\sqrt{k(m+n)}/\delta_k(A))\|A\|_F + 4\sqrt{k(m+n)} \end{aligned}$$

Substituting in for the definition of ϵ , we achieve the stated bound. ■

Corollary 20 *Using the notation and assumptions of Theorem 19, with high probability*

$$\|A^{(k)} - \widehat{A}^{(k)}\|_F \leq 1.25 \epsilon \|A\|_F$$

Proof. We show that the second term in the bound of Theorem 19 is bounded by $\epsilon\|A\|_F/4$. By Proposition 3, $\|A\|_F$ is at least $\sqrt{k}\sigma_k(A)$, and in turn $\sigma_k(A)$ is at least $\delta_k(A)$. We conclude that $\|A\|_F/\delta_k(A)$ is at least \sqrt{k} , which lower bounds $\epsilon\|A\|_F$ by $16\sqrt{k(m+n)}$. ■

Notice that these results are the most meaningful when $\|A\|_F$ is not much larger than $\|A^{(k)}\|_F$. This is natural; if $A^{(k)}$ itself does not stand out as a good approximation to A we should not expect to recover it easily. However, a large spectral gap can compensate in cases when $\|A^{(k)}\|_F \ll \|A\|_F$.

The result that we have proved is somewhat coarse, bounding only the aggregate effect of error, but it does show that with a significant spectral gap we can hope to recover the optimal rank k approximation despite substantial random perturbation. In the coming sections we will see how to more carefully characterize the error experienced and bound the influence on individual rows and columns.

3.3 Information Retrieval

Salton [64] introduced the now traditional framework of the term-document matrix combined with the cosine similarity measure. While this approach worked well, it was noted by Berry et al. [9] that one could improve the performance of this approach by replacing the term-document incidence matrix A with $A^{(k)}$. This was justified by arguing that $A^{(k)}$ recovers the latent linear structure of the data set, and by removing all else the cosine similarity more accurately measures the intended similarity. Papadimitriou et al. [60] present a theoretical model which argues analytically for the success of this approach. Their model introduces synonymy, but, in the context of our modelling framework, they require that each document and term have at most one non-zero element in their k -tuple. The result is documents that must be focused on one topic, and terms which are not permitted to be polysemous. We will see a much more general result, using only the assumption of a spectral gap in A .

Following the lead of Papadimitriou et al., we will define a generative model for such graphs, based on latent linear structure. We have seen how we can model the correspondence of terms to documents, and we simply imagine that the probability that a term occurs in a document is proportional to its relevance to that document.

Information Retrieval Error(A):

For each entry A_{ij} , independently set

$$\widehat{A}_{ij} = \begin{cases} \lfloor A_{ij} \rfloor & \text{with probability } A_{ij} - \lfloor A_{ij} \rfloor \\ \lceil A_{ij} \rceil & \text{with probability } \lceil A_{ij} \rceil - A_{ij} \end{cases}$$

In the information retrieval domain, the matrix \widehat{A} represents the observed term-document incidence matrix. Our goal is to evaluate the similarity of rows and columns *in terms of the original matrix* A . That is, we would like to produce for i, j good approximations to $\langle A_i, A_j \rangle$ which will imply good approximations to the cosines between each A_i and A_j .

Theorem 21 *Let \widehat{A} result from **Information Retrieval Error**(A). If $\epsilon = 16\sqrt{m+n}/\delta_k(A)$ is at most 2, then with high probability for all but at most c columns i it is the case that*

$$Pr[|A_i^{(k)} - \widehat{A}_i^{(k)}| \geq \epsilon|A_i| + \epsilon\sqrt{32k(m+n)/c} + \sqrt{k}\lambda] \leq 2ke^{-\lambda^2/4}$$

Notice that this theorem argues that, aside from a set of c columns, it is unlikely that many columns will vary greatly from their intended value.

Proof. This result is similar to that just proven in Theorem 19, but we will be interested in the error each column experiences. For each column, we can write

$$\begin{aligned} A_i^{(k)} - \widehat{A}_i^{(k)} &= P_A^{(k)} A_i - P_{\widehat{A}}^{(k)} \widehat{A}_i \\ &= (P_A^{(k)} - P_{\widehat{A}}^{(k)}) A_i + P_{\widehat{A}}^{(k)} (A_i - \widehat{A}_i) \\ &= (P_A^{(k)} - P_{\widehat{A}}^{(k)}) A_i + P_A^{(k)} (A_i - \widehat{A}_i) - (P_A^{(k)} - P_{\widehat{A}}^{(k)}) (A_i - \widehat{A}_i) \end{aligned}$$

We will bound the norms of these three error terms separately, each with different techniques. The first term is handled by considering the bound that Corollary 17 provides on $\|(A - \widehat{A})^{(k)}\|_2$ using $\sigma = 1$, and then applying Stewart's Theorem to place an ϵ upper bound on $\|P_A^{(k)} - P_{\widehat{A}}^{(k)}\|_2$. The second term is the projection of a vector of independent random variables, $A_i - \widehat{A}_i$, onto a fixed subspace, $P_A^{(k)}$. Corollary 15 bounds the probability that any one vector exceeds $\sqrt{k}\lambda$ by $2ke^{-\lambda^2/4}$, the probability noted in the theorem statement.

We now argue that the third term is bounded for all but at most c columns. Observe that the matrix $(P_A^{(k)} - P_{\widehat{A}}^{(k)})(A - \widehat{A})$ is not only rank at most $2k$, but involves the matrix

$P_A^{(k)} - P_{\widehat{A}}^{(k)}$, which has L_2 norm bounded by ϵ . As such, by Proposition 4 we may bound

$$\begin{aligned} \|(P_A^{(k)} - P_{\widehat{A}}^{(k)})(A - \widehat{A})\|_F^2 &\leq 2k\|(P_A^{(k)} - P_{\widehat{A}}^{(k)})\|_2^2\|A - \widehat{A}\|_2^2 \\ &\leq 2k\epsilon^2\|A - \widehat{A}\|_2^2 \end{aligned}$$

Recalling the definition of the Frobenius norm squared as the sum of squared entries, we associate the terms in the summation by column, to get the bound

$$\sum_i |(P_A^{(k)} - P_{\widehat{A}}^{(k)})(A_i - \widehat{A}_i)|^2 \leq 2k\epsilon^2\|A - \widehat{A}\|_2^2$$

By a counting argument, for at most c columns i

$$|(P_A^{(k)} - P_{\widehat{A}}^{(k)})(A_i - \widehat{A}_i)|^2 > 2k\epsilon^2\|A - \widehat{A}\|_2^2/c$$

With all error terms now accounted for, the proof is complete. ■

While this theorem does not directly bound the error associated with inner products, it bounds the distance that each column may move, which imposes a bound on the change in inner products associated with columns of sufficient norm. We now consider a corollary which instantiates several parameters, working with the assumptions that each column has non-trivial norm and that there are at least as many documents as terms.

Corollary 22 *Using the notation and assumptions of Theorem 21, if $n \geq m$ and for every i , $|A_i| \geq 16\sqrt{k \log k}/\epsilon$, then with high probability for at least $n/4$ columns it is the case that*

$$\Pr[|A_i^{(k)} - \widehat{A}_i^{(k)}| \geq 3\epsilon|A_i|] \leq 2e^{-64}$$

and consequently, with high probability for at least $n^2/4$ pairs i, j of columns

$$|\langle A_i^{(k)}, A_j^{(k)} \rangle - \langle \widehat{A}_i^{(k)}, \widehat{A}_j^{(k)} \rangle| \leq 9\epsilon^2|A_i||A_j|$$

Proof. We instantiate Theorem 21 using $c = n/4$, and $\lambda = 16\sqrt{k \log k}$. The failure probability of $2e^{-64}$ is sufficiently small that through tail inequalities we can confidently state that fewer than $n/4$ nodes will fail, giving the stated number of accurate inner products.

■

Clearly this result is most exciting when $|A_i^{(k)}| \gg \epsilon|A_i|$ and $|A_j^{(k)}| \gg \epsilon|A_j|$, as it is in this setting that angles will be best preserved.

The implication of this theorem is that in a text corpus model based on latent semantics we are able to overcome the vagaries of the random rounding that **Information Retrieval Error** introduces. We overcome synonymy by learning a measure of term similarity more accurate than simple co-occurrence. This ameliorates much of the polymemy problem, which resulted from a need to emphasize any term co-occurrence as an indication of similarity.

3.3.1 Discussion of Kleinberg's Link Analysis

Analogous results can be produced for the domain of web hyperlinks. If we draw out the connection between pages containing links and documents containing terms, we see that we can produce an interesting model for web data. Indeed, we will do this more thoroughly in Chapter 4, but at this point we have enough mathematical foundation to give a rigorous justification of Kleinberg's HITS algorithm [47], which is based on a rank 1 approximation to various subgraphs of the web's linkage graph. By positing that various pages have an intrinsic "authority" which attracts incoming links, while others have an intrinsic "awareness" that leads them to authoritative pages, we can produce a rank 1 probability matrix equal to the outer product of these two vectors. Theorem 21 argues that in this model Kleinberg's approach will accurately reconstruct the majority of posited values.

3.4 Collaborative Filtering

We next consider the problem of analyzing an incomplete data set. We model the production of incomplete data as omission from a complete data set:

Collaborative Filtering Error(A, P):

For each entry A_{ij} , independently set

$$A_{ij}^* = \begin{cases} A_{ij} & \text{w.p. } P_{ij} \\ \text{"?"} & \text{w.p. } 1 - P_{ij} \end{cases}$$

Our goal is to reconstruct the entries of A . For the first time, we have an instance where we do not simply wish to compute the optimal rank k approximation to the data set at hand. This should come as no surprise, as the matrix A^* contains entries (the “?”) that can not be readily added or multiplied.

We propose the following approach for recovering the values obscured by the “?”s assuming that the omission probabilities P_{ij} are known. We will describe a technique for estimating the P_{ij} values in certain cases later, but we have no scheme for addressing this problem for general, unknown P_{ij} .

Algorithm **CF**(A^*, P)

1. Return the matrix \widehat{A} , defined as:

$$\widehat{A}_{ij} = \begin{cases} A_{ij}/P_{ij} & \text{if } A_{ij}^* \neq \text{“?”} \\ 0 & \text{if } A_{ij}^* = \text{“?”} \end{cases}$$

While the algorithm may seem arbitrary at first, its key feature is that the matrix \widehat{A} has expectation A . Again we will argue that the difference $A - \widehat{A}$ has small L_2 norm and therefore $\widehat{A}^{(k)}$ is an excellent approximation to $A^{(k)}$.

Theorem 23 *Let A^* result from **Collaborative Filtering Error**(A, P), where each $P_{ij} > p \geq 64 \log^6(m+n)/(m+n)$ and each $|A_{ij}| \leq 1$. Let $\widehat{A} = \mathbf{CF}(A^*, P)$. If $\epsilon = 16\sqrt{m+n}/\delta_k(A)$ is at most 2, then with high probability*

$$\|A^{(k)} - \widehat{A}^{(k)}\|_F \leq \epsilon \|A\|_F + 4\sqrt{k(m+n)/p}$$

Proof. The proof proceeds exactly as in Theorem 19, with the additional observation that the variances are now bounded by $1/p$, rather than 1. Our lower bound on p ensures that the range constraint of Corollary 17 is met. \blacksquare

Theorem 23 bounds the total squared error between $A^{(k)}$ and $\widehat{A}^{(k)}$, which implies a bound on the mean squared error, using the relation $(a+b)^2 \leq 2a^2 + 2b^2$:

$$\begin{aligned} \text{avg}_{i,j} (A_{ij}^{(k)} - \widehat{A}_{ij}^{(k)})^2 &\leq \frac{2\epsilon^2 \|A\|_F^2 + 32k(m+n)/p}{mn} \\ &= 2\epsilon^2 \frac{\|A\|_F^2}{mn} + \frac{32k/p}{\min(m,n)} \end{aligned}$$

The MSE between $A^{(k)}$ and $\hat{A}^{(k)}$ is thus an $2\epsilon^2$ fraction of the mean squared entry size of A , plus a term which goes to zero provided both m and n grow faster than $1/p$. The larger the gap $\delta_k(A)$, the greater the accuracy of our estimate.

3.4.1 Estimating The Omission Probabilities

The algorithm **CF** assumes that the probability of retaining a particular entry is known. Such information is unlikely to be available in practice, as the only evidence of the probabilities lies in the sampled values. However, if we believe that these probabilities exhibit latent linear structure themselves, not unlike the assumption made of A , we should be able to recover a good approximation to them, using the techniques of Section 3.3:

Algorithm **ExtendedCF**(A^*)

1. Let \hat{P} be the matrix defined as

$$\hat{P}_{ij} = \begin{cases} 1 & \text{if } A_{ij}^* \neq \text{"?"} \\ 0 & \text{if } A_{ij}^* = \text{"?"} \end{cases}$$

2. Compute and return **CF**(A^* , $\hat{P}^{(k)}$).

It is important to note that our proof of the performance of **CF** relies on having the precise omission probabilities. At this time, we do not know what performance is guaranteed if **CF** takes as input only a very good approximation to the omission probabilities. Resolving this question is a key problem left open in this work.

3.5 Conclusions

In this chapter we have worked through several data mining scenarios, and observed that spectral methods are capable of overcoming the apparent impediment of significant random perturbation. The two critical properties that we used throughout this section were that

1. The original data A had strong spectral structure, here in the form of a spectral gap much larger than $\sqrt{m+n}$. Such a gap can result from a data set guided by a latent linear structure, as offered in the model of Section 3.1.

2. The perturbation that was applied to the data was random, with each entry zero mean and independent. The variance of the perturbation was also important, as its magnitude appears in the bound we produced on $\|A - \hat{A}\|_2$ and $\|(A - \hat{A})^{(k)}\|_F$.

In the case of collaborative filtering we had to massage the input data to bring it to a form where these properties both held. It was by understanding which properties were required of the observed data that we arrived at the correct transformation. Throughout the rest of this thesis we will see this theme revisited, frequently requiring us to tailor probability distributions to properly align the expected value with the intended value.

Chapter 4

WEB SEARCH

Sing, clear voiced Muse, of Hephaistos (Vulcan), renowned for his inventive skill, who with grey-eyed Athene, taught to men upon earth arts of great splendor, men who in former days lived like wild beasts in mountain caves. But having learned skills from Hephaestus, famed for his work and craftsmanship, they now, free from care, peacefully live year by year in their houses. Be gracious, Hephaestus, and grant me excellence and prosperity!

–Homeric Hymn to Hephaestus

4.1 Overview and Discussion

In this chapter, we define a mathematical framework for evaluating web search algorithms based on a simple, yet rich model for generating web documents and queries. We also present a new web search algorithm based on spectral techniques, dubbed *SmartyPants*, that is guaranteed to give near-optimal results in this framework (Theorem 24). Our algorithm is entirely motivated by the model, and indeed may not seem intuitive *unless* one considers our generative model and its implications.

We feel that the process of defining such a model is useful for a number of reasons. First, the mathematical framework enables rigorous comparison of search algorithms. More generally, casting search as a mathematical problem brings a beneficial separation of two intertwined concepts: (a) an abstraction (model) that describes the correlations that make search possible and (b) an algorithm that exploits those correlations. At the very least, such a separation is beneficial in the following sense: if an algorithm is proven to be “good” with respect to a model, yet “no good” in practice, then we will be motivated to further understand in what way the model (and hence the algorithm) is lacking. Finally, in our

case, since the algorithm we propose is based on spectral techniques, the model also serves as a testbed for understanding when and why such techniques work.

The basic idea of our model, described in detail in Section 4.3, is that there exist some k latent concepts underlying the web, and that every topic can be represented as a combination of these concepts. Web pages, terms, and queries are each associated with some topic, and the linkage that occurs from hub pages to authorities is based on the similarity of their topics.

We like to think of the task at hand for *SmartyPants* as being:

1. Take the human generated query and determine the topic to which the query refers.
2. Synthesize a “perfect hub page” for this topic. The perfect hub is a fictional page; it needn’t exist in the data set. A perfect hub has links to those pages most authoritative on the specified topic.

This task breakdown is explicit in Kleinberg’s HITS algorithm [47] and seems desirable for any search algorithm. Unlike other algorithms *SmartyPants* uses link information for both the second *and* the first subtask. In particular, we combine latent semantic analysis of term content and link structure, finding the identifying the topic of the query and establishing the authorities on this topic. This is made possible because of a model in which a unified semantic space of underlies the generation of link structure, term content, and query generation. Thus, we obtain a principled mechanism for avoiding the difficulties other search algorithms have experienced in connecting text with authoritative sites.

4.2 Additional Related Work

There have been previous attempts to fuse term content and link structure (e.g., the work of Cohn [21], Chakrabarti et al [18], and Hofmann [39]). Previous literature makes use of either spectral techniques or the EM algorithm of Dempster. However, whether these algorithms perform well or why remains unclear. The main distinguishing characteristic of our algorithm is that it is provably correct in the context of the model.

Our model is inspired by many previous works, including the term-document model used to rigorously analyze LSI in [61, 7], PLSI [39], the web-link generation model used to study Kleinberg’s algorithm [7], PHITS [39] and the combined link structure and term content model of Cohn [21]. All of these models, as well as some of the models described in Tsaparas et al [13] can be viewed as special cases of our model.

4.3 The Model

The fundamental assumption in our model is that there exists a set of k unknown (latent) *concepts* whose combinations capture every topic considered in the web. How large k is, and what each of these concepts means is unknown. In fact, our algorithm need not (and can not) identify the underlying concepts.

Given a set of k concepts, a *topic* is a k -dimensional vector w , describing the contribution of each of the basic concepts to this topic; the ratio between the i -th and j -th coordinates of w reflects the relative contributions of the underlying i -th and j -th concepts to this subject. In order to ascribe a probabilistic interpretation to the various quantities in the model, we assume that the coordinates of each topic vector are non-negative, though the assumption is not needed from a mathematical point of view.

Associated with each web page p are two vectors.

- The first vector associated with p is a k -tuple A_p – reflecting the topic on which p is perceived as an *authority*. This topic models the content on this page and, therefore, the topics of pages likely to link to this page. The magnitude of a web page’s authority vector, $|A_p|$, determines how *strong* an authority it is on that particular topic.
- The second vector associated with p is a k -tuple H_p – reflecting the topic on which p is a *hub*, i.e., the topic that defines the set of links that p will create to other pages. Intuitively, the hub topic of the page is usually pretty well captured by the anchor text for the links from that page. As before, the magnitude $|H_p|$ captures how strong of a hub it is.

Remark: These topics can be used to represent the union of many “real-life” topics on the page. Hence, if for example page p is the union/concatenation of pages p_1, p_2 then each of the two vectors for p is the sum of the corresponding vectors of p_1, p_2 .

4.3.1 Link Generation

In our model the number of links from page p to page q is a random variable with expected value equal to $L_{qp} = \langle A_q, H_p \rangle$. The more closely aligned the hub topic of page p is with the authority topic of page q , the more likely it is that there will be a link from p to q . In addition, the stronger a hub p is (as measured by $|H_p|$), and/or the stronger an authority q is (as measured by $|A_q|$), the more likely it is that there will be a link from p to q . Our model allows the distribution of this random variable to be arbitrary, so long as its range is bounded by a constant independent of the number of web documents.

Collecting these entries L_{pq} , we describe the link generation model’s expectation as an n by n matrix L , where n is the number of documents on the web. L is the product of two matrices

$$L = A^T H$$

We denote an instance of the web’s link structure by \hat{L} , where \hat{L}_{ij} is the number of links from page j to page i . \hat{L} is an instantiation of the random web model defined by the matrix L . \hat{L}_{ij} is obtained by sampling from the distribution with expectation L_{ij} of bounded range.

4.3.2 Term Generation

We now introduce terms to the web pages. Associated with each term is a topic vector.

- The vector is a k -tuple S_u that describes the use of term u as a hub term (i.e., as anchor text). The i th entry of this tuple is the expected number of occurrences of the term u in a hub document on concept i .

Our model then assumes that terms on a page p with hub topic H_p are generated from a distribution of bounded range where the expected number of occurrences of term u is

$T_{up} = \langle S_u, H_p \rangle$. Again, the greater the magnitude of $|S_u|$, the more common the term is, and the more frequently it appears.

Thus, we can describe the term generation model's expectation as an n by t matrix T , where n is the number of documents on the web and t is the total number of terms,

$$T = S^T H$$

We denote an instance of the web's document-term matrix by \hat{T} . \hat{T}_{ij} is the number of occurrences of term j in page i , and is assumed by the model to be an instantiation of the term model matrix T . \hat{T}_{ij} is obtained by sampling from a distribution with expectation T_{ij} of bounded range.

4.3.3 Query Generation

We assume that the search process begins with the searcher conceiving a query topic, on which he wishes to find the most authoritative pages. The terms that the searcher presents to the search engine will be the terms that a perfect hub on this topic would use; arguably the best terms to describe the topic. An intuitive way to think about this is that the searcher presents the search engine with a portion of anchor text, and expects the pages most likely to be linked to using that text. The search engine must then produce the pages that are likely linked to by this text.

This motivates our model for the query generation process. In order to generate the search terms of a query:

- The searcher chooses a query topic, represented by a k -tuple v .
- The searcher mentally computes the vector $q = S^T v$. Observe that q_u is the expected number of occurrences of the term u in a pure hub page on topic v .
- The searcher then decides whether to include each term u among his search terms by sampling independently from a distribution with expectation q_u . We denote the instantiated vector by \hat{q}_u .

The input to the search engine consists of the terms with non-zero values in the vector \hat{q} .

By choosing the magnitude of v to be very small, the searcher can guarantee that only a small number of search terms will be sampled. In this case, \hat{q} will be largely zero and have very few non-zero entries.

4.3.4 The Correct Answer to a Search Query

Given our model as stated above, the searcher is looking for the most authoritative pages on topic v . For a page p , this is scored precisely as $\langle A_p, v \rangle$. Thus, *the correct answer to the search query* is given by presenting the user with pages sorted by their corresponding entries of $A^T v$.

We will take this moment to note also that $A^T v = LT^{-1}q$, as can be seen through the following equalities:

$$\begin{aligned} LT^{-1}q &= (A^T H)(S^T H)^{-1}(S^T v) \\ &= A^T H H^{-1} S^{-T} S^T v \end{aligned}$$

The terms $H H^{-1}$ and $S^{-T} S^T$ collapse to $k \times k$ identity matrices, leaving us $A^T v$, a fortunate event, as the same is not true of $H^{-1} H$ and $S^T S^{-T}$.

4.4 The Algorithm: *SmartyPants*

The algorithm takes as input a search query \hat{q} . Under the model, this query is generated by a human searcher by instantiating $q = S^T v$ for some topic v . The goal of *SmartyPants* is to compute the order of authority implied by $A^T v$. To do so, *SmartyPants* makes use of the web graph \hat{L} and the web term matrix \hat{T} , both of which are derived by web crawling. An interesting feature of *SmartyPants* is that it does not compute either v or A . In fact, one can show that it is not possible to explicitly derive those matrices given \hat{L} and \hat{T} only. Nonetheless, as we will see, *SmartyPants* does extract a good approximation to $A^T v$.

4.4.1 The Algorithm

- Before queries are presented, *SmartyPants* computes $\hat{L}^{(k)}$ and $\hat{T}^{(k)}$.

- When a query vector \hat{q} is presented, *SmartyPants* computes

$$w = \hat{L}^{(k)} \hat{T}^{(k)-1} \hat{q}$$

and returns the highest scoring entries of w .

Note that this algorithm is quite efficient. The preprocessing of \hat{L} and \hat{T} can be done efficiently as both \hat{L} and \hat{T} are sparse, and in any case it need only be done once. At query time, \hat{q} is likely sparse, which makes the computation of w very easy. As $\hat{L}^{(k)}$ and $\hat{T}^{(k)-1}$ are rank k we may efficiently multiply vectors by them, requiring $k(|q| + 3)$ flops to generate a given entry of w . We can efficiently generate the largest entries of w without computing the entire vector using techniques of Fagin et al [30].

Remark: The clever reader will note that as $\hat{T}^{(k)}$ is rank k , it does not actually have an inverse. As such, we abuse notation and let X^{-1} be the *pseudo inverse*, which is the inverse of the matrix on the range that it spans, and the zero matrix elsewhere. Whereas the inverse of $A = UDV^T$ is equal to $VD^{-1}U^T$, the pseudoinverse equals VCU^T , where the matrix C equals D^{-1} for those entries where D is non-zero, and is 0 otherwise.

4.4.2 The Main Theorem

For any matrix B , let $r_i(B) = \sigma_1(B)/\sigma_i(B)$. If $r(B) = 1$ then this means that the singular values do not drop at all, the larger $r_i(B)$ is the larger the drop in singular values. We use n for the number of pages and t for the number of terms.

Theorem 24 *Let L and T be matrices of probabilities generated according to our model, and let \hat{L} and \hat{T} result from randomly rounding the corresponding matrices. For any $\epsilon < 1$, if both $\sigma_k(L)$ and $\sigma_k(T)$ are at least $8r_k(L)r_k(T)\sqrt{n+t}/\epsilon$, then with high probability:*

*For each query \hat{q} generated by rounding entries of a query vector q generated according to our model, *SmartyPants* computes a vector of authorities w , such that:*

$$Pr \left[\frac{|w - A^T v|}{|A^T v|} \geq 2\epsilon + \delta \right] \leq 2ke^{-\left(\frac{\delta|q|}{2\sqrt{k}r_k(L)r_k(T)}\right)^2}$$

In essence, Theorem 24 claims that with sufficiently structured L and T , we can expect that most queries q with length much greater than $\sqrt{k}r_k(L)r_k(T)$ will result in quality results.

Proof. The correct answer to a query is $LT^{-1}q$, whereas we compute $\widehat{L}^{(k)}\widehat{T}^{(k)-1}\widehat{q}$. We break the proof into two steps, first proving that with high probability

$$\|LT^{-1} - \widehat{L}^{(k)}\widehat{T}^{(k)-1}\|_2 \leq 2\epsilon\sigma_k(L)/\sigma_1(T)$$

And then proving that projecting \widehat{q} instead of q has bounded negative impact.

To start on the first bound, we apply the triangle inequality and then Proposition 1

$$\begin{aligned} \|LT^{-1} - \widehat{L}^{(k)}\widehat{T}^{(k)-1}\|_2 &\leq \|LT^{-1} - L\widehat{T}^{(k)-1}\|_2 + \|(L - \widehat{L}^{(k)})\widehat{T}^{(k)-1}\|_2 \\ &\leq \|LT^{-1} - L\widehat{T}^{(k)-1}\|_2 + \|L - \widehat{L}^{(k)}\|_2\|\widehat{T}^{(k)-1}\|_2 \end{aligned}$$

Theorem 16 taking $\sigma = 1$ bounds $\|L - \widehat{L}^{(k)}\|_2 \leq 4\sqrt{2n}$ with high probability.

$$\|LT^{-1} - \widehat{L}^{(k)}\widehat{T}^{(k)-1}\|_2 \leq \|LT^{-1} - L\widehat{T}^{(k)-1}\|_2 + \frac{4\sqrt{2n}}{\sigma_k(\widehat{T})} \quad (4.1)$$

To address the remaining term, we write

$$\|LT^{-1} - L\widehat{T}^{(k)-1}\|_2 \leq \|L\|_2\|T^{-1} - \widehat{T}^{(k)-1}\|_2$$

We now use a theorem of Wedin [71], which bounds for arbitrary A and B

$$\|A^{-1} - B^{-1}\|_2 \leq \frac{2\|A - B\|_2}{\min(\sigma_k(A)^2, \sigma_k(B)^2)}$$

Applying this theorem with $A = T$ and $B = \widehat{T}^{(k)}$, and applying Theorem 16 with $\sigma = 1$ to bound $\|T - \widehat{T}\|_2 = \|T - \widehat{T}^{(k)}\|_2$

$$\|LT^{-1} - L\widehat{T}^{(k)-1}\|_2 \leq \frac{\sigma_1(L)8\sqrt{n+t}}{\min(\sigma_k(T)^2, \sigma_k(\widehat{T})^2)} \quad (4.2)$$

Our lower bounds on $\sigma_k(L)$ and $\sigma_k(T)$ ensure that $\sigma_k(\widehat{L})$ and $\sigma_k(\widehat{T})$ are effectively equal to their unperturbed counterparts. We will equate the pairs of terms for now, and incorporate a factor of 2 to account for the slop.

Combining Equations 4.1 and 4.2, and using our lower bounds on $\sigma_k(L)$ and $\sigma_k(T)$,

$$\begin{aligned} \|LT^{-1} - \widehat{L}^{(k)}\widehat{T}^{(k)-1}\|_2 &\leq \frac{\sigma_1(L)8\sqrt{n+t}}{\min(\sigma_k(T)^2, \sigma_k(\widehat{T})^2)} + \frac{4\sqrt{2n}}{\sigma_k(\widehat{T})} \\ &\leq \epsilon \left(\frac{\sigma_k(L)}{\sigma_1(T)} + \frac{\sigma_k(L)}{r_k(L)r_k(T)\sigma_k(T)} \right) \\ &\leq 2\epsilon\sigma_k(L)/\sigma_1(T) \end{aligned}$$

In the second half of the proof, we bound

$$Pr \left[\frac{|w - A^T v|}{|A^T v|} \geq 2\epsilon + \delta \right] \leq 2ke^{-\left(\frac{\delta|q|}{2\sqrt{k}r_k(L)r_k(T)}\right)^2}$$

where $w = \widehat{L}^{(k)}\widehat{T}^{(k)-1}\widehat{q}$ is the output *SmartyPants* produces.

We start by applying the triangle inequality to the left hand side

$$\begin{aligned} |\widehat{L}^{(k)}\widehat{T}^{(k)-1}\widehat{q} - LT^{-1}q| &\leq |(\widehat{L}^{(k)}\widehat{T}^{(k)-1} - LT^{-1})q| + |\widehat{L}^{(k)}\widehat{T}^{(k)-1}(q - \widehat{q})| \\ &\leq \|\widehat{L}^{(k)}\widehat{T}^{(k)-1} - LT^{-1}\|_2|q| + |\widehat{L}^{(k)}\widehat{T}^{(k)-1}(q - \widehat{q})| \end{aligned}$$

Using the first part of our proof, the first term has norm bounded as

$$\|\widehat{L}^{(k)}\widehat{T}^{(k)-1} - LT^{-1}\|_2|q| \leq |q|2\epsilon\sigma_k(L)/\sigma_1(T)$$

Recalling that $A^T v = LT^{-1}q$,

$$\begin{aligned} |q|2\epsilon\sigma_k(L)/\sigma_1(T) &= |TL^{-1}A^T v|2\epsilon\sigma_k(L)/\sigma_1(T) \\ &\leq \|T\|_2\|L^{-1}\|_2|A^T v|2\epsilon\sigma_k(L)/\sigma_1(T) \\ &= 2\epsilon|A^T v| \end{aligned}$$

Piecing this train of inequalities together

$$\|\widehat{L}^{(k)}\widehat{T}^{(k)-1} - LT^{-1}\|_2|q| \leq 2\epsilon|A^T v| \tag{4.3}$$

which is good, as we ultimately intend to divide by $|A^T v|$.

The second term we must bound, $|\widehat{L}^{(k)}\widehat{T}^{(k)-1}(q - \widehat{q})|$, is the projection of a random vector, $q - \widehat{q}$, onto a fixed low dimensional subspace, in the form of $\widehat{L}^{(k)}\widehat{T}^{(k)-1}$. Applying Corollary 15, we establish that

$$Pr \left[|\widehat{L}^{(k)}\widehat{T}^{(k)-1}(q - \widehat{q})| > \|\widehat{L}^{(k)}\widehat{T}^{(k)-1}\|_2\lambda\sqrt{k} \right] \leq 2ke^{-\lambda^2/4}$$

We will perform a substitution for λ , using

$$\begin{aligned} \lambda &= \delta|A^T v|/\|\widehat{L}^{(k)}\widehat{T}^{(k)-1}\|_2\sqrt{k} \\ &\geq \delta|q|(\sigma_k(L)/\sigma_1(T))/(\sigma_1(L)\sigma_k(T))\sqrt{k} \\ &= \delta|q|/(kr_k(L)r_k(T))\sqrt{k} \end{aligned}$$

With this substitution made, our bound becomes

$$\Pr \left[\left| \widehat{L}^{(k)} \widehat{T}^{(k)-1} (q - \widehat{q}) \right| > \delta |A^T v| \right] \leq 2ke^{\frac{-\delta^2 |q|^2}{4kr_k^2(L)r_k^2(T)}} \quad (4.4)$$

Collecting Equations 4.3 and 4.4, and dividing by $|A^T v|$ concludes the proof. \blacksquare

It is worth noting that the $r_k(X)$ terms are necessary due to worst case assumptions. In particular, we need to cover the case where the strongest linear trends in L actually correspond to the weakest linear trends in T . This would be odd, but possible. One could imagine a topic that defined itself by links, and used no text to speak of. In the event that the singular values line up properly, formally stated by bounding $\|LT^{-1}\|_2$ and $\|TL^{-1}\|_2$, we have the following best-case corollary:

Corollary 25 *Let L and T be matrices of probabilities generated according to our model, and let \widehat{L} and \widehat{T} result from randomly rounding the corresponding matrices. For any $\epsilon < 1$, if both $\sigma_k(L)$ and $\sigma_k(T)$ are at least $8\sqrt{n+t}/\epsilon$, and both $\|LT^{-1}\|_2$ and $\|TL^{-1}\|_2$ are bounded by 1, then with high probability:*

*For each query \widehat{q} generated by rounding entries of a query vector q generated according to our model, *SmartyPants* computes a vector of authorities w , such that:*

$$\Pr \left[\frac{|w - A^T v|}{|A^T v|} \geq 2\epsilon + \delta \right] \leq 2ke^{-\left(\frac{\delta|q|}{2\sqrt{k}}\right)^2}$$

We see that the r_k terms have vanished, owing to our assumption that the strengths of corresponding trends in L and T are commensurate.

4.5 Discussion and Extensions

One problem with the approach taken by *SmartyPants* is that the k fundamental concepts in the overall web will be quite different from the k fundamental concepts when we restrict attention to computer science sites. What we would like to do is to focus onto the relevant subset of documents so that we indeed have sufficient resolution to identify the topic at hand, and the topic does not vanish as one of the insignificant singular values.

A natural recursive approach would be to apply *SmartyPants*, sort the sites by both authority value and hub value, and take the top 1/3 of the most authoritative sites along

with the top 1/3 of the hub sites and recur. This process clearly converges, and we suspect (but don't have any evidence) that this may work well in practice. If the search term "singular value decomposition" (say) has *any* significance in the topic "Engineering" (say) then the first round will post Engineering authorities and hubs high up in the list and Theology authorities and hubs low in the list. The second round will start with Theology and other unrelated topics omitted, and a possibly higher resolution set of concepts for Engineering. In fact, since we don't know the resolution at which the user wants to perform the query, it may be useful to provide answers at various resolutions.

4.6 Conclusions

In this chapter we have looked at a fairly detailed example of spectral analysis. Whereas in previous problems we were able to simply associate the problem with matrix reconstruction, here we must reconstruct the matrices and then leverage them to compute a more complex function. In our case, we used the matrix T^{-1} to rewrite the query in terms of hub pages, and then fed these web pages to L , who was able to convert hub pages to authority pages.

It is not difficult to imagine other settings which involve translation of data, operating off of multiple data sets with unified underlying semantics. The techniques used here show that such is possible, and that if the operations rely principally on the strong latent structure, then even significant random perturbations may have little effect on the result.

Chapter 5

GRAPH PARTITIONING

In the introduction we posed the problems of finding bisections, cliques, and colorings in random graphs that were known to contain interesting solutions. In this chapter we will investigate these problems further, generalizing the techniques that have been used in previous literature into a common algorithmic framework, and observing that the framework can be used to solve a far more general class of graph partitioning problems.

5.0.1 *Graph Partition Model*

The graph models that have been proposed for planting combinatorial objects in random graphs, described in section 1.2.4, each generate a graph by including each edge independently with an associated probability. These probabilities are carefully chosen so that the desired combinatorial object exists, be it bisection, clique or coloring. Importantly, in each model the nodes can be partitioned into a few parts so that the probability that an edge occurs depends only on the parts to which its endpoints belong. In the planted bisection model edges occur with a probability p if they lie within the same part and q otherwise, in the clique model edge probabilities are 1 if both endpoints are in the clique and p otherwise, and in the planted coloring model, edge probabilities are zero if endpoints belong to the same color class and p otherwise.

Based on this observation we introduce the following general model of “structured” random graphs.

$\mathcal{G}(\psi, P)$: Let $\psi : \{1, \dots, n\} \rightarrow \{1, \dots, k\}$ be a partition of n nodes into k classes. Let P be a $k \times k$ matrix with $P_{ij} \in [0, 1]$ for all i, j . Include edge (u, v) with probability $P_{\psi(u)\psi(v)}$.

For a particular distribution $\mathcal{G}(\psi, P)$ we let \widehat{G} refer to the matrix of random variables

corresponding to the adjacency matrix of the random graph. We also let G refer to the matrix of expectations, where $G_{uv} = P_{\psi(u)\psi(v)}$.

We now specialize this model into three models which are equivalent to the models for planted bisection/multisection, coloring, and clique presented in the literature.

- **Planted Multisection** (ψ, p, q) : ψ is the multisection. P is p everywhere, except the diagonal where it is q .
- **Planted k-Coloring** (ψ, p) : ψ is the coloring. P is p everywhere except the diagonal, where it is 0.
- **Planted Clique** (ψ, p) : Let $\psi(v) = 1$ iff v is in the clique. P is p everywhere, except P_{11} , which is 1.

Remark: Undirected graphs usually do not permit self-edges, and so we are likely interested in forcing the diagonal entries to 0. We will deal with this complication in Section 5.3.1, where we examine the implications of clamping diagonal entries, and conclude doing so does not much affect our algorithms. Until then, we consider the model as stated.

Our model of a random graph with a planted partition leads naturally to the following graph partitioning problem.

Planted Partition Problem: Given a graph \widehat{G} drawn from the distribution $\mathcal{G}(\psi, P)$, produce a partition $\widehat{\psi}$ so that

$$\widehat{\psi}(u) = \widehat{\psi}(v) \quad \text{iff} \quad \psi(u) = \psi(v)$$

As ψ encodes the solution to each of the problems above, recovering a partition equivalent to ψ generalizes the problems of finding planted multisections, cliques, and colorings in their respective models.

Remark: It is important to disassociate this problem from related traditional optimization problems. The goal is not to find the largest clique or min cost bisection, but rather to recover the planted object. In many cases these two will be the same, but if the optimal

solution is not equivalent to the planted object our goal is to find the latter. There will be a fairly small range of parameters where we can prove that our approach finds the planted solution, but can not prove that it is optimal.

In some ways, this is a very interesting distinction. First, one can always find the optimal object, be it clique, coloring, or partition, it may just take a while. On the other hand, there is a point at which it is simply not possible to confidently state that a particular partition was the planted one. Recovering the planted partition is something that can only be done for certain parameter ranges.

5.0.2 Our Approach

Given the matrix G , it is easy to reconstruct ψ by clustering the columns G_u of G . Unfortunately, we have instead a matrix \widehat{G} which is a highly perturbed version of G , and the columns \widehat{G}_u are nowhere near the G_u . It is perhaps natural to ask: “Why can we hope to recover ψ at all?” The answer to this question, and our approach to this problem, is based on the following series of observations:

1. For any ψ and P , the matrix G has rank k .
2. If $P_G^{(k)}$ is the projection on the column space of G
 - $|P_G^{(k)}(G_u) - G_u|$ is zero.
 - $|P_G^{(k)}(G_u - \widehat{G}_u)|$ is small.
3. By the triangle inequality, $P_G^{(k)}(\widehat{G}_u)$ equals G_u , plus a “small” error.

Of course, we do not have direct access to $P_G^{(k)}$ either. Matrix perturbation theory comes to our rescue, in the spirit of Stewart’s theorem, in that the equivalent projection for \widehat{G} is very similar to $P_G^{(k)}$.

Our approach is now to find a projection $P_X^{(k)}$, perhaps based on $P_{\widehat{G}}^{(k)}$, such that

$$\begin{aligned} \text{Data is Preserved:} & \quad |P_X^{(k)}(G_u) - G_u| \text{ is small} \\ \text{Noise is Removed:} & \quad |P_X^{(k)}(G_u - \widehat{G}_u)| \text{ is small} \end{aligned}$$

If so, then $P_X^{(k)}(\widehat{G}_u)$ equals G_u up to a “small” error. If when $\psi(u) \neq \psi(v)$, $|G_u - G_v|$ is much larger than this error, we can apply a simple greedy clustering process to the $P_X^{(k)}(\widehat{G}_u)$, recovering the partition ψ .

With a minor modification, this is the algorithm we will use. Let τ be a threshold parameter, let s_m capture the size of the smallest part, and let **CProj** be a function which computes an “appropriate” projection matrix.

Algorithm 1 Partition($\widehat{G}, k, s_m, \tau$)

- 1: Randomly divide $\{1, \dots, n\}$ into two parts, dividing the columns of \widehat{G} as $[\widehat{A}|\widehat{B}]$
 - 2: Let $P_1 = \mathbf{CProj}(\widehat{B}^{(k)}, s_m, \tau)$; let $P_2 = \mathbf{CProj}(\widehat{A}^{(k)}, s_m, \tau)$
 - 3: Let $\widehat{H} = [P_1(\widehat{A})|P_2(\widehat{B})]$
 - 4: While there are unpartitioned nodes
 - 5: **loop**
 - 6: Choose an unpartitioned node u_i arbitrarily
 - 7: For each unpartitioned v , set $\widehat{\psi}(v) = i$ if $|\widehat{H}_{u_i} - \widehat{H}_v| \leq \tau$
 - 8: **end loop**
 - 9: Return the partition $\widehat{\psi}$.
-

Notice that the main difference from the outlined scheme is that we split the matrix \widehat{G} into two parts. This is done to avoid the conditioning that would otherwise exist between the error $G - \widehat{G}$ and the computed projection, a function of \widehat{G} . This is done for the sake of analysis, and it is still unclear if it is required in practice.

5.0.3 Results

The main result we will see is an analysis of the algorithm **Partition**. Appropriate choices of **CProj** and τ result in perfect classification for a large range of (ψ, P) . To describe the performance of **Partition** in the general context of the Planted Partition Problem we must describe the range of (ψ, P) for which the algorithm succeeds. This range is best described by a requisite lower bound on $|G_u - G_v|$ when $\psi(u) \neq \psi(v)$:

Theorem 26 *Let (ψ, P) be an instance of the planted partition problem. Let $\sigma^2 \gg \log^6 n/n$ be an upper bound on the variance of the entries in G , let s_m be the size of the smallest part of ψ , and let $\tau = \min_{u,v:\psi(u) \neq \psi(v)} |G_u - G_v|$. With probability $1 - \delta$, there is a constant c such that for sufficiently large n if*

$$\tau > ck\sigma^2 \left(\frac{n}{s_m} + \log \left(\frac{n}{\delta} \right) \right) ,$$

*each application of **Partition** $(\widehat{G}, k, s_m, \tau)$ returns ψ with probability at least $1/2 + 1/16$.*

The proof of this theorem will follow from Claim 30, which presents a characterization of when this algorithm will succeed, combined with Theorem 32, which argues that these conditions hold with sufficient probability.

Notice that this theorem implies that with high probability, we may apply **Partition** repeatedly, and use the majority answer. As such, if the antecedents of Theorem 26 are satisfied, then with high probability we can efficiently recover the partition ψ .

Before proving Theorem 26, we consider its consequences for the three problems we have mentioned: Bisection, Coloring, and Clique. For these corollaries we insist that the failure probability δ is not required to be smaller than $\exp(-\log^6 n)$. The proofs amount to instantiation of each model's associated parameters, and are not presented here in detail.

Corollary 27 (Bisection) *Let (ψ, p, q) be an instance of the planted bisection problem with k parts. There is a constant c so that for sufficiently large n if*

$$\frac{q - p}{q} > c \sqrt{\frac{\log(n/\delta)}{qn}}$$

then we can efficiently recover ψ with probability $1 - \delta$.

This range of parameters is equivalent to the range Boppana produces in [12], up to constant factors. It is worth emphasizing that Boppana produces the optimal bisection for graph generated according to our model, whereas we recover the planted bisection, and can generalize to multisections. Condon and Karp [22] succeed with multisections for a nearly identical (though strictly smaller) range of parameters using a very elegant linear time algorithm.

Corollary 28 (Coloring) *Let (ψ, p) be an instance of the planted k -coloring problem, where the size of each color class is linear in n . There is a constant c such that for sufficiently large n if*

$$p > c \log^3(n/\delta)/n$$

then we can efficiently recover ψ with probability $1 - \delta$.

This result is simultaneously weaker and more general than that of Alon et al in [4]. Here we admit color classes of differing sizes, and can further generalize to cover the case where the sizes of the color classes are asymptotically different. On the other hand, this result covers a smaller range of p than [4] who show that the problem can be solved even when $p = c/n$, for some large constant c . Their improved range is due to combinatorial preprocessing wherein they remove all nodes whose degrees are much larger than their expectation, trimming the variance of the entries of \widehat{G} . It appears that this pre-processing can also be applied to our algorithm, with similar benefit, but this requires more investigation before stating it as fact.

Corollary 29 (Clique) *Let (ψ, p) be an instance of the planted clique problem, where the clique size is s . There is a constant c such that for sufficiently large n if*

$$\frac{1-p}{p} > c \left(\frac{n}{s^2} + \frac{\log(n/\delta)}{s} \right)$$

then we can efficiently recover ψ with probability $1 - \delta$.

This result subsumes the spectral result of Alon et al in [5], where they show that cliques of size $\Omega(\sqrt{n})$ can be found when $p = 1/2$. Note that this theorem allows for variable p and s . The restriction to a single clique is also not necessary. The general theorem addresses graphs with several hidden cliques and hidden independent sets, each of varying size. For fair comparison, Alon et al also show how the constant c may be reduced to any constant d , though the technique incorporates a factor of $n^{c/d}$ into the running time.

5.0.4 Additional Related Work

As well as theoretical success in average case analysis, spectral algorithms have been successfully used in practice as a heuristic for data partitioning. While there is no single

spectral approach, most examine the eigenvectors of the adjacency matrix of a graph (or of the Laplacian of this matrix). In particular, the second eigenvector is typically used as a classifier, partitioning nodes based on the sign of their coordinate. The special cases of bounded degree planar graphs and d-dimensional meshes (cases which occur frequently in practice) were analyzed successfully by Spielman and Teng in [66]. Recently, Kannan, Vempala, and Vetta [42] gave a compelling clustering bi-criteria and a spectral algorithm which produces clusterings of quality similar to the optimal clustering.

Feige and Kilian [31] consider an alternate model for describing the performance of “empirical” algorithms. For the problems of bisection, coloring, and clique, a random graph is produced as before. However, they now allow an adversary to “help” the algorithm, perhaps by including additional edges between color classes, or removing non-clique edges. [31] give algorithms that address these problems when the objects are of linear size. While this model is interesting in these three domains, it does not seem to generalize to planted partition problem.

5.1 Spectral Graph Partitioning

It is now time to identify projections and delve into their particulars. Recall that **Partition** works when the projected columns $P(\widehat{G}_u)$ are close to the original columns G_u , and the columns in G from different parts are distant. We can codify this in the following observation.

Claim 30 *Let P_1 , and P_2 be defined as in **Partition** $(\widehat{G}, k, s_m, \tau)$, and let A and B represent the division of the columns of G analogous to the division of the columns of \widehat{G} . Assume that for all u*

$$\begin{aligned} |P_1(A_u) - A_u| &\leq \gamma_1 & \text{and} & & |P_1(A_u - \widehat{A}_u)| &\leq \gamma_2 \\ |P_2(B_u) - B_u| &\leq \gamma_1 & \text{and} & & |P_2(B_u - \widehat{B}_u)| &\leq \gamma_2 \end{aligned}$$

If when $\psi(u) \neq \psi(v)$

$$|G_u - G_v| \geq 4(\gamma_1 + \gamma_2)$$

*then **Partition** $(\widehat{G}, k, s_m, 2(\gamma_1 + \gamma_2))$ is equivalent to ψ .*

Proof. The main gist of the proof is that we may place a bound on

$$||H_u - H_v| - |G_u - G_v|| \leq 2(\gamma_1 + \gamma_2)$$

Therefore, provided that $|G_u - G_v|$ is at least $4(\gamma_1 + \gamma_2)$, then $\psi(u) = \psi(v)$ iff

$$|H_u - H_v| \leq 2(\gamma_1 + \gamma_2)$$

By using $\tau = 2(\gamma_1 + \gamma_2)$, **Partition** $(\widehat{G}, k, s_m, \tau)$ recovers the correct partition. ■

The challenging question is now: “Given \widehat{A} and \widehat{B} , can we compute projections P_1 and P_2 with small values of γ_1 and γ_2 ?” Theorems 31 and 32 bound these values for two different computable projections. Theorem 32 involves **CProj** and largely surpasses the results of Theorem 31, but the latter is pleasingly simple and (in the author’s opinion) more natural than Theorem 32.

Before we barge into any proofs or other discussions, we need to define a few terms. With high probability, the size of each part of ψ when restricted to A (or B) is close to half the original size of the part. Let $s_i(A)$ and $s_i(B)$ denote the sizes of part ψ_i in each of A and B . Let s_m be a lower bound on these sizes. The variables i, j will lie in the range $\{1, \dots, k\}$, whereas the variables u, v will lie in the range $\{1, \dots, n\}$. We will occasionally use notation such as G_i to refer to the column of G that corresponds to nodes in part i . Likewise, we will use \widehat{s}_u to refer to the size of the part containing u .

5.2 A Traditional Spectral Result

A natural projection to consider for P_2 is the projection onto the first k left singular vectors of \widehat{A} . As they are the best basis to describe \widehat{A} , we might imagine that they will capture the structure of B as well (the columns of A and B are the same).

Theorem 31 *For $\sigma \geq 8 \log^3(n)/n^{1/2}$, with probability at least $1 - 2\delta$, for all u*

$$\begin{aligned} |P_{\widehat{A}}^{(k)}(B_u) - B_u| &\leq 5\sigma\sqrt{n/s_u(A)} \\ |P_{\widehat{A}}^{(k)}(B_u - \widehat{B}_u)| &\leq 2\sqrt{k \log(2kn/\delta)} \end{aligned}$$

Proof. We address the first inequality first. Note that the column B_u appears $s_u(A)$ times in A . Consider the vector x which has a 1 in entry v if $A_v = B_u$ and 0 otherwise. Notice that $Ax = |x|^2 B_u$, and that $|x|^2 = s_u(A)$. The definition of the L_2 norm requires that

$$\begin{aligned} |(I - P_{\hat{A}}^{(k)})Ax|/|x| &\leq \|(I - P_{\hat{A}}^{(k)})A\|_2 \\ |(I - P_{\hat{A}}^{(k)})B_u| &\leq \|(I - P_{\hat{A}}^{(k)})A\|_2/|x| \end{aligned}$$

As such, we will now work to bound $\|(I - P_{\hat{A}}^{(k)})A\|_2$. Notice that by the triangle inequality:

$$\|(I - P_{\hat{A}}^{(k)})A\|_2 \leq \|(I - P_{\hat{A}}^{(k)})\hat{A}\|_2 + \|(I - P_{\hat{A}}^{(k)})(\hat{A} - A)\|_2$$

We now bound each of these terms by $\|\hat{A} - A\|_2$. The first term equals $\|\hat{A} - \hat{A}^{(k)}\|_2$, and by the optimality of $\hat{A}^{(k)}$ is only made larger if we replace $\hat{A}^{(k)}$ with A , itself a rank k matrix. In the second term, $(I - P_{\hat{A}}^{(k)})$ is a projection matrix, and has norm 1. Applying Corollary 17 bounds, with high probability

$$\|A - \hat{A}\|_2 \leq 4\sigma\sqrt{1.5n}$$

The second inequality to be proved is a direct application of Azuma's inequality (as Corollary 15), as $P_{\hat{A}}^{(k)}(B_u - \hat{B}_u)$ is the projection of a vector of independent, zero mean random variables onto a fixed, k -dimensional subspace.

We increase the failure probability δ by a factor of 2 to account for the failure probability of Corollary 17. ■

While this result is indeed quite strong, it is not quite strong enough to get all the results that we would like. The problem is the absence of a σ term in the second bound, which we will work to introduce in the next section. While this will prevent us from analyzing low variance situations such as very sparse graphs, it suffices for clique bounds, proving Corollary 29, and the dense graph bisection cases.

5.3 Combinatorial Projections

We now present the algorithm **CProj** and analyze its performance. In this algorithm, and in its discussion, we use the notation \hat{A}_v^T . This is the v th column of the transpose of \hat{A} ,

equivalently the v th row of \widehat{A} , which is a vector of roughly $n/2$ coordinates. At a high level, **CProj** attempts to cluster the *rows* of A , and uses these clusters to define a subspace for the columns.

Algorithm 2 **CProj**(X, s_m, τ)

- 1: **while** there are at least $s_m/2$ unclassified nodes **do**
- 2: Choose an unclassified node v_i randomly.
- 3: Let $\widehat{\psi}_i = \{u : |X_{v_i}^T - X_u^T| \leq \tau\}$
- 4: Mark each $u \in \widehat{\psi}_i$ as classified.
- 5: **end while**
- 6: Assign each remaining node u to the $\widehat{\psi}_i$ minimizing $|X_{v_i}^T - X_u^T|$.
- 7: Let \widehat{c}_i be the characteristic vector of $\widehat{\psi}_i$
- 8: Return the projection onto the space spanned by the \widehat{c}_i , equal to

$$P_{\widehat{c}} = \sum_i \frac{\widehat{c}_i \widehat{c}_i^T}{|\widehat{c}_i|^2}$$

If the \widehat{c}_i were the characteristic vectors of ψ , this projection would be exactly $P_A^{(k)}$. Instead, we will see that the \widehat{c}_i are not unlike the characteristic vectors of ψ , differing in at most a few positions.

Theorem 32 *Let \widehat{G} be generated according to the Planted Partition Problem, and let \widehat{A} and \widehat{B} represent a random partitioning of the columns of \widehat{G} . For $\sigma \geq 8 \log^3(n)/n^{1/2}$, if when $\psi(u) \neq \psi(v)$*

$$|G_u - G_v| > 64\sigma \sqrt{nk \log(k)/s_m}$$

then with high probability, letting $P_{\widehat{c}}^{(k)} = \mathbf{CProj}(A, k, s_m, 32\sigma \sqrt{nk \log(k)/s_m})$,

$$\begin{aligned} |P_{\widehat{c}}^{(k)}(B_u) - B_u| &\leq 128\sigma \sqrt{nk/s_m} \\ |P_{\widehat{c}}^{(k)}(B_u - \widehat{B}_u)| &\leq 4\sigma \sqrt{k \log(kn)} + 4 \log(nk) \sqrt{k/s_m} \end{aligned}$$

with probability at least 3/4.

Proof. The proof is conducted in two parts: Lemmas 34 and 35 bounding the first and second term, respectively. Theorem 6 combined with Theorem 16 provides the bound:

$$\|G - \widehat{G}^{(k)}\|_F \leq 16\sigma\sqrt{kn}$$

which in turn bounds $\|A - \widehat{A}^{(k)}\|_F$, as is needed by Lemma 34. As well, with high probability, the sizes $s_i(A)$ and $s_i(B)$ are all at least $s_i/4$. Note that we multiply the bound of Lemma 35 by 2 to lower the failure probability below $1/n$. ■

We start our analysis by proving a helper lemma.

Lemma 33 *Under the assumptions of Theorem 32, if τ is such that when $\psi(u) \neq \psi(v)$*

$$|A_u^T - A_v^T|/2 \geq \tau \geq \|A - X\|_F \sqrt{\log(k)/s_m}$$

*then with probability at least $3/4$ each of the k nodes v_i we select in **CProj** will be drawn from different parts of ψ , and will each satisfy*

$$|A_u^T - X_u^T| \leq \tau/2$$

Proof. Let us call any node u that satisfies $|A_u^T - X_u^T| \leq \tau/2$ “good”; others nodes will be called “bad”. Notice that if at each step we do choose a good node, then by using radius τ we will mark all the good nodes from the same part as u , and no good nodes from any other part (as the A_i^T are assumed to be separated by 2τ). If we only ever chose good nodes, the proof would be complete.

So, let us look at the probability of choosing a bad node at a particular step. For every bad node u , by definition $1 < 2|A_u^T - X_u^T|/\tau$, and so

$$\begin{aligned} |BAD| &< \sum_{u \in BAD} 4|A_u^T - X_u^T|^2/\tau^2 \\ &\leq 4\|A - X\|_F^2/\tau^2 \end{aligned}$$

By assumption, $\tau^2 \geq \|A - X\|_F^2 \log k/s_m$, and we establish that

$$|BAD| < s_m/4 \log k$$

However, notice that in the first iteration of **CProj** we have at least ks_m nodes to choose from, and so the probability of choosing a bad node is at most $(4k \log k)^{-1}$. Indeed, at any step i at which we have not yet chosen a bad node, there are still $(k - i + 1)s_m$ nodes to choose from, as we have only marked nodes from $i - 1$ parts and all good nodes from the remaining parts still remain.

The probability of selecting a bad node at the i th step is therefore at most

$$\Pr[v_i \in \text{BAD}] \leq ((k - i + 1)4 \log k)^{-1}$$

If we now take a union bound, we see that

$$\begin{aligned} \Pr[\vee_i (v_i \in \text{BAD})] &\leq \frac{\sum_{i \leq k} (k - i + 1)^{-1}}{4 \log k} \\ &\approx 1/4 \end{aligned}$$

As we choose only good nodes, and at each step mark all good nodes associated with a particular part, each selection must be from a different part. ■

Lemma 34 (Systematic Error) *Under the assumptions of Lemma 33, with probability at least $3/4$, for all u*

$$|G_u - P_{\hat{c}}^{(k)}(G_u)|^2 < 16 \|A - X\|_F^2 / s_u(A)$$

Proof. We start the proof by defining the matrix E , of dimensions equivalent to A , whose entries are set as

$$E_{vu} = G_{\hat{\psi}(v)\psi(u)}$$

We can view E as what the matrix of probabilities A would look like if its columns obeyed the partition ψ but its rows obeyed the partition $\hat{\psi}$.

Notice that the columns E_u lie in the space spanned by the \hat{c}_i , the basis vectors for the projection $P_{\hat{c}}^{(k)}$. On the other hand, $P_{\hat{c}}^{(k)}(G_u)$ is the vector in the space spanned by the \hat{c}_i that minimizes the distance to G_u . In other words:

$$\begin{aligned} |G_u - P_{\hat{c}}^{(k)}(G_u)| &\leq |G_u - E_u| \\ &= |A_u - E_u| \end{aligned}$$

For all columns v such that $\psi(v) = \psi(u)$, both $A_v = A_u$ and $E_v = E_u$, and so

$$\begin{aligned} |A_u - E_u|^2 &= \sum_{v:\psi(v)=\psi(u)} |A_v - E_v|^2 / s_u(A) \\ &\leq \|A - E\|_F^2 / s_u(A) \end{aligned}$$

To make the transition to $\|A - X\|_F^2$ in the numerator we will consider the rows of $A - E$, or equivalently, the columns of $A^T - E^T$. If vertex u is correctly classified in $\hat{\psi}$, then E_u^T equals A_u^T and their difference is 0. Alternatively, consider a node u that should have been associated with a node w , but was instead associated with v . As u was associated with v instead of w , it must be the case that X_u^T was closer to X_v^T than it was to X_w^T :

$$|X_u^T - X_v^T| \leq |X_u^T - X_w^T|$$

In the next two steps, we first rewrite the terms above, noting that $A_w^T = A_u^T$, and second apply the triangle inequality

$$\begin{aligned} |(X_u^T - A_u^T) + (A_u^T - A_v^T) + (A_v^T - X_v^T)| &\leq |(X_u^T - A_w^T) + (A_w^T - X_w^T)| \\ |A_w^T - A_v^T| - |X_u^T - A_u^T| - |A_v^T - X_v^T| &\leq |X_u^T - A_u^T| + |A_w^T - X_w^T| \end{aligned}$$

We rearrange terms and apply Lemma 33 to bound $|A_v^T - X_v^T|$ and $|A_w^T - X_w^T|$ by $\tau/2$.

$$|A_w^T - A_v^T| - \tau \leq 2|X_u^T - A_u^T|$$

Our assumed lower bound $|A_w^T - A_v^T| \geq 2\tau$ from Lemma 33 leads us to

$$|A_w^T - A_v^T|/2 \leq 2|X_u^T - A_u^T|$$

Recalling that $A_w^T = A_u^T$ and $A_v^T = E_u^T$:

$$\begin{aligned} \sum_u |A_u^T - E_u^T|^2 / 4 &\leq \sum_u 4|X_u^T - A_u^T|^2 \\ \|A - E\|_F^2 &\leq 16\|A - X\|_F^2 \end{aligned}$$

which, when substituted above, concludes the proof. ■

Next, we prove that the projection of the rounding error in the subspace $P_{\hat{c}}^{(k)}$ is small. This argument is little more than a dressed up Chernoff bound, although some care must

be taken to get it into this form. It is also important to note that, while the notation does not reveal that $P_{\hat{c}}^{(k)}$ and \hat{G}_u are independent, the former is based on a set of columns which do not contain \hat{G}_u .

Lemma 35 (Random Error) *Let σ^2 be an upper bound on the entries of G . With probability $1 - \delta$*

$$|P_{\hat{c}}^{(k)}(G_u - \hat{G}_u)|^2 \leq 4k\sigma^2 \log(k/\delta) + 8k \log^2(k/\delta)/s_m$$

Proof. Notice that the vectors \hat{c}_i used to define our projection are disjoint, and therefore orthogonal. As such, we can easily decompose $|P_{\hat{c}}^{(k)}(G_u - \hat{G}_u)|$ into a sum of k parts, defined by the vector's projection onto each of the basis vectors $\hat{c}_i/|\hat{c}_i|$.

$$|P_{\hat{c}}^{(k)}(G_u - \hat{G}_u)|^2 = \sum_{i \leq k} (\hat{c}_i^T (G_u - \hat{G}_u))^2 / |\hat{c}_i|^2$$

We consider each of the k terms separately, observing that each is a sum of independent random variables with mean zero. Noting that the entries of \hat{G} are independent 0/1 random variables, we apply a form of the Chernoff bound from Motwani and Raghavan [58], which says that for a sum of 0/1 random variables, X ,

$$Pr[|E[X] - X| \geq t] \leq \max\{\exp(-t^2/4\mu), \exp(-t/2)\}$$

If we apply this to our sum, we see that

$$Pr[\hat{c}_i^T (G_u - \hat{G}_u) > t^{1/2}] \leq \max\{\exp(-t/4\mu), \exp(-t^{1/2}/2)\}$$

where,

$$\mu = E[\hat{c}_i^T \hat{G}_u] \leq \sum_{v \in \hat{\psi}_i} \max_{i,j} G_{ij}$$

$\max_{i,j} G_{ij}$ is bounded by σ , and the number of terms in this sum is at most $\hat{s}_m > s_m/2$.

We therefore instantiate

$$t = 4\sigma^2 \log(k/\delta) + 8 \log^2(k/\delta)/s_m$$

and apply a union bound, concluding that all of the k terms are bounded by t with probability at least $1 - \delta$. ■

5.3.1 Undirected Graphs: Excluding Self-Loops

We noted in the introduction to this chapter that most undirected graph models do not admit self-loops. In its present form, our analysis does not apply to such graphs. We now go through the details required to extend the analysis to such undirected graphs.

There are two sources of error that we bounded in Lemmas 34 and 35: error in the computed subspace, and error in the projection of the random vectors, respectively. For the first lemma, notice that the main property of $A - \hat{A}$ used was that its norm is bounded. If we consider the squashing of diagonal entries to be additional, non-random error, we see that its norm is bounded by 1, as it is a diagonal matrix whose entries are either -1 or 0 .

The second lemma argued that the projection of the $[0/1]$ vector \hat{G}_u onto the provided basis behaved much like its expected vector, G_u . This projection is analyzed by considering the sum of the $[0/1]$ terms corresponding to each of the parts defined by $\hat{\psi}$. If we clamp one of these values to 0, we decrease the number of terms in the sum, and therefore decrease its variability. However, the expectation of this sum is now off by G_{uu}/\hat{s}_u for column \hat{c}_u ; the other sums stay the same, as \hat{G}_{uu} is only accumulated into one sum. If u was a good node, there are at least $s_u/2$ other terms of equivalent magnitude, and this clamping can be viewed much like the event that any one of these terms resulted in a 0. If u was bad, we have avoided contributing G_{uu}/\hat{s}_u to the wrong sum, actually bringing $P_{\hat{c}}\hat{G}_u$ closer to G_u .

5.4 Observations and Extensions

For the most part, the question of average case multisection, coloring, and clique is: “for what range of parameters can we find a solution in poly-time?” The algorithm presented in this paper also has the desirable property that it is not slow. Aside from the computation of the matrix $P_{\hat{A}}^{(k)}$, we require $O(nk^2 + mk)$ time to partition and classify nodes.

Several recent papers have begun to address the problem of efficiently computing approximations to $P_{\hat{A}}^{(k)}\hat{A}$. The optimality of $P_{\hat{A}}^{(k)}\hat{A}$ is not truly required; if one were to produce a rank k matrix X instead of $P_{\hat{A}}^{(k)}\hat{A}_k$, the term $|\hat{A} - X|$ could be introduced into the bound, replacing occurrences of $|A - \hat{A}|$. In the next chapter, we will examine algorithms which quickly compute nearly optimal approximations. These approaches sample and scale en-

tries, maintaining the expectation, but increasing the variance. Our analysis can be applied trivially with the slightly larger variance, resulting in sub-linear time partitioning for a certain (non-trivial) range of parameters.

5.4.1 General Graph Partitioning

Our restriction to unweighted symmetric graphs is purely artificial. The bound of Theorem 16 applies equally well to weighted, non-symmetric matrices if we apply the $J(M)$ transformation of Jordan (at the expense of a $\sqrt{2}$ term). Perhaps even more interesting, at no point have we actually required that our matrices be square. Our analyses have tacitly assumed this, but they can easily be rewritten in terms of n_1 and n_2 , should the input matrix be $n_1 \times n_2$ dimensional.

5.4.2 Parameterless Partitioning

The principal open question of Condon and Karp [22] involves the problem of partitioning a graph when either the part sizes or number of parts is unknown. For our result we need only a lower bound on the sizes, and an upper bound on the number of parts. These two bounds do occur in the requisite lower bound on $|G_u - G_v|$, and if they are too loose we risk not satisfying this bound. Otherwise, the algorithm performs properly, even without precise information about the size and number of parts.

Chapter 6

EIGENCOMPUTATION

In this chapter we will look at improved methods for computing low rank approximations, and the associated singular subspaces. Before doing so, we briefly visit a common technique for computing them. We will look at the simplest approach, both because it is commonly used, and because its simplicity will allow us to easily make several beneficial modifications. The approach is called Orthogonal Iteration, and operates as follows:

Algorithm 3 Orthogonal Iteration (A, X, i)

- 1: **for** $1 \dots i$ **do**
 - 2: Let $X = \text{Orthonormalize}(AA^T X)$.
 - 3: **end for**
 - 4: Return X .
-

We then use $P_X^{(k)} A$ as $P_A^{(k)} A$. Notice that this algorithm does little other than repeatedly multiply vectors (the columns of X) by AA^T . Orthonormalization is generally performed by Gram-Schmidt, where one orthogonalizes each vector only against those that precede it, but other approaches also make sense. For example, Raleigh-Ritz acceleration results from setting X to the left singular vectors of $AA^T X$.

While this algorithm is quite simple, it is not obvious that such an approach should result in anything useful. Indeed, it does, and the convergence is rather quick.

Theorem 36 For any A and orthonormal X , let $Y = \text{OrthogonalIteration}(A, X, i)$.

$$\|(I - P_A^{(k)})P_Y^{(k)}\|_2 \leq \left(\frac{\sigma_{k+1}(A)}{\sigma_k(A)}\right)^{2i} \cdot \frac{\|(I - P_A^{(k)})P_X^{(k)}\|_2}{\sigma_k(P_A^{(k)} P_X^{(k)})}$$

Proof. The proof, while short and elegant, requires a few concepts we have not covered. It is omitted here but can be found as Theorem 8.2.2 in Golub and Van Loan [36]. ■

6.1 Accelerated Eigencomputation

In this section we will examine two specific techniques within a broad framework of acceleration techniques for eigencomputation. Both techniques are based on the introduction of computationally friendly error into the matrices. The error will accelerate orthogonal iteration, and we will be able to prove that the produced result is not too far from the desired matrix.

The two techniques we examine are random sparsification and random quantization of entries. Both perform independent random operations to every entry in A , resulting in a random matrix \widehat{A} , whose expectation is equal to A , and whose variance is modest.

6.1.1 Random Sparsification

This technique is simple in description: we randomly zero out entries in the input matrix.

Random Sparsification(A, p):

Compute and return \widehat{A} , where for each \widehat{A}_{ij} we independently set

$$\widehat{A}_{ij} = \begin{cases} A_{ij}/p & \text{with probability } p \\ 0 & \text{otherwise} \end{cases}$$

Notice that $E[\widehat{A}_{ij}] = A_{ij}$, and the variance of each term is bounded by $|A_{ij}|^2/p$, which means we can apply Corollary 17 and get

Lemma 37 *For any matrix A whose entries are bounded in magnitude by b , we can produce a matrix \widehat{A} with p times as many non-zero entries such that*

$$\|(A - \widehat{A})^{(k)}\|_F \leq 4b\sqrt{k(m+n)/p}$$

As the matrix \widehat{A} has p times as many non-zero entries, the multiplication step of Orthogonal Iteration takes p times as long. Furthermore, we have only p times as much data to store, enabling the use of main memory in place of disk for small p . It is also possible to avoid flipping a coin for each entry: instead we produce a random variable from a geometric distribution indicating the number of entries we should skip over. This has the advantages

of producing \hat{A} with fewer random bits and only probing A at the locations we intend to retain. This last feature is very important in many machine learning problems where each entry of A is the result of extensive computation, and avoiding such computation leads to large performance gains.

6.1.2 Random Quantization

Quantization is similar in spirit, though perhaps slightly more complicated. We wish to quantize each entry to one of two values. Given that the expectation must work out to A_{ij} , we tailor the probabilities appropriately.

Random Quantization(A):

Let $b = \max_{i,j} |A_{ij}|$. Compute and return \hat{A} , where for each \hat{A}_{ij} we independently set

$$\hat{A}_{ij} = \begin{cases} +b & \text{with probability } 1/2 + A_{ij}/2b \\ -b & \text{with probability } 1/2 - A_{ij}/2b \end{cases}$$

Notice that as before, $E[\hat{A}_{ij}] = A_{ij}$, and but here the variance of each term is bounded by b^2 . Intuitively, there is some sense to the probabilities. They are roughly equal probability ($1/2$ apiece) weighted to favor the appropriate sign. Again, Corollary 17 yields

Lemma 38 *For any matrix A whose entries are bounded in magnitude by b , we can produce a matrix \hat{A} whose entries are each one of $\pm b$, such that with high probability*

$$\|(A - \hat{A})^{(k)}\|_F \leq 4b\sqrt{k(m+n)}$$

As each entry is one of two possible values, we can represent each by a single bit. While this has nominal complexity implications, the degree of compression this will achieve in typical modern systems using double precision floats is a factor of 64. Indeed this scheme gives a solid analytic foundation for the use of limited precision numbers, provided the rounding is performed randomly.

6.1.3 Error Bounds

In general, spectral techniques are applied because the data is thought to have strong spectral structure. It would therefore not be unreasonable to posit that our input data is a

perturbation of either low rank data or data with a spectral gap. To keep our discussion as general as possible, we will make neither of these assumptions, and instead work with our results for general matrix perturbation, Theorems 10 and 13.

Rather than restate these theorems here with our now traditional instantiation of $\|(A - \widehat{A})^{(k)}\|_F$, we give a high level argument for why, independent of any specific perturbation bounds, these acceleration techniques should be acceptable. Firstly, one assumes that the data miner is interested in the optimal rank k approximation for the purposes of removing noise from the input, and has therefore tacitly agreed that such noise may exist and that whatever algorithm they intend to employ will operate properly despite the presence of random error. The techniques of random sparsification and quantization that we proposed in this section involve only the addition of independent, mean zero random error to the input matrix, perhaps the most benign class of random error.

6.1.4 Combining Sparsification and Quantization

While each approach above is useful in its own right, combining sparsification and quantization is not trivial. Much of the representation cost of a sparse matrix lies in describing its sparsity structure, not in storing its entries. A typical representation stores the non-zero elements as a list of (row, col, val) triples. Compressing the val field to a single bit does not result in substantial compression overall.

We may address this problem in practice by noting that the sparsification process is the result of a pseudo-random process that we control. Given the seed pseudorandom number generator, we could re-run the sampling process, generating indices on the fly. Assuming that we have quantized and recorded the corresponding entries, regenerating the list of indices and reading the associated entry data reconstructs the matrix for us, allowing us to perform fast matrix multiplication without explicitly transcribing any more than a few quantized entries.

6.1.5 Enhanced Random Sparsification

While we have defined sparsification in terms of a single omission probability p , there is no reason each entry could not be omitted with a different probability. Consider

General Random Sparsification(A, P):

Compute and return \widehat{A} , where for each \widehat{A}_{ij} we independently set

$$\widehat{A}_{ij} = \begin{cases} A_{ij}/P_{ij} & \text{with probability } P_{ij} \\ 0 & \text{otherwise} \end{cases}$$

In light of Theorem 16, whose bound relies only on the *maximum* variance of the entries in \widehat{A} , we might choose (again using $b = \max_{i,j} |A_{ij}|$)

$$P_{ij} = p|A_{ij}|^2/b^2$$

With such omission probabilities, the variances of the entries \widehat{A}_{ij} are bounded by b^2/p . This is the same variance bound we established when using the global omission probability p . On the other hand, we have lowered the number of retained entries from pmn to

$$\begin{aligned} E[\text{entries}] &= \sum_{i,j} p|A_{ij}|^2/b^2 \\ &= p\|A\|_F^2/b^2 \end{aligned}$$

This quantity is at most pmn , and equal to pmn only when all entries in the matrix have identical magnitude. If there is any variability in the magnitude of the matrix entries, we have retained fewer entries without increasing the error bound provided by Theorem 16.

Remark: We must be careful in discounting the probabilities too low, as there is a range constraint in Theorem 16 which may be violated. For very small entries, we run the unlikely but possible risk of a very large entry when we divide by an exceedingly small p_{ij} . We must therefore, for technical reasons, constrain p_{ij} to be at least $|A_{ij}|2 \log^3 n / \sqrt{n/p}$. Viewed differently, at a particular threshold p_{ij} begins to decrease proportionally to $|A_{ij}|$ as opposed to $|A_{ij}|^2$. The number of additional entries we expect to see due to this thresholding is bounded by $4n \log^6 n$.

6.2 Incremental Eigencomputation

The majority of large data sets, perhaps by virtue of their size, experience little relative change on a daily basis. A vast amount of data does change, but compared to the total amount of data, the change is not overly large. Furthermore, when change does occur it is likely to be cosmetic, the latent structure remaining constant.

In this section we will examine an incremental algorithm for computing singular vectors, using Orthogonal Iteration as its base. Orthogonal Iteration is an iterative algorithm, meaning that it takes an approximation to the solution (in this case, an approximation to the left singular vectors) and improves them. If they are good to begin with, fewer improvement steps will be required to return to a desired accuracy. Orthogonal Iteration, combined with information about the number of steps required, becomes an incremental algorithm for eigencomputation.

Analysis of singular vectors can often be made much easier by assuming that there is a gap in the singular values. That is, there is an index k such that $\sigma_k - \sigma_{k+1}$ is large. We made this assumption in Chapter 3, and it is certainly the case in many matrices (the pagerank transition matrix, for example). This assumption will be central to the result we now prove, though the possibility of a gapless bound remains high.

In the following theorem, A and B are two data matrices, B resulting from an incremental change to A . X spans a subspace that approximates A well, and Y will result from $OrthogonalIteration(B, X, i)$.

Theorem 39 *For any two matrices A and B of like dimensions, let $\delta = \|A - B\|_2 / \delta_k(A)$. If X is an orthonormal matrix such that*

$$\|(I - P_A^{(k)})P_X^{(k)}\|_2 \leq \epsilon$$

then, letting $Y = OrthogonalIteration(B, X, i)$

$$\|(I - P_B^{(k)})P_Y^{(k)}\|_2 \leq \left(\frac{\hat{\sigma}_{k+1}(B)}{\hat{\sigma}_k(B)} \right)^{2i} \cdot \frac{\epsilon + \delta}{1 - (\epsilon + \delta)^2}$$

Proof. Recall from our discussion of Orthogonal Iteration that

$$\|(I - P_B^{(k)})P_Y^{(k)}\|_2 \leq \left(\frac{\hat{\sigma}_{k+1}(B)}{\hat{\sigma}_k(B)} \right)^{2i} \cdot \frac{\|(I - P_B^{(k)})P_X^{(k)}\|_2}{\|P_B^{(k)}P_X^{(k)}\|_2}$$

Stewart's theorem (Theorem 9) tells us that $\|(I - P_B^{(k)})P_X^{(k)}\|_2$ is close to $\|(I - P_A^{(k)})P_X^{(k)}\|_2$.

$$\begin{aligned} \|(I - P_B^{(k)})P_X^{(k)}\|_2 &\leq \|(I - P_A^{(k)})P_X^{(k)}\|_2 + \delta \\ &\leq \epsilon + \delta \end{aligned}$$

Recall now that $P_X^{(k)} = P_B^{(k)}P_X^{(k)} + (I - P_B^{(k)})P_X^{(k)}$. Let V_k be the k th right singular vector of $P_B^{(k)}P_X^{(k)}$. As V_k lies in the preimage of $P_X^{(k)}$, $|P_X^{(k)}V_k| = |V_k| = 1$. However, the Pythagorean equality tells us that

$$\begin{aligned} |P_X^{(k)}V_k|^2 &= |P_B^{(k)}P_X^{(k)}V_k|^2 + |(I - P_B^{(k)})P_X^{(k)}V_k|^2 \\ &\leq |P_B^{(k)}P_X^{(k)}V_k|^2 + (\epsilon + \delta)^2 \end{aligned}$$

From this, we conclude that

$$\begin{aligned} \sigma_k(P_B^{(k)}P_X^{(k)}) &= |P_B^{(k)}P_X^{(k)}V_k| \\ &\geq 1 - (\epsilon + \delta)^2 \end{aligned}$$

With these bounds inserted into Theorem 36, the proof is completed. ■

We are presumably interested in the number of iterations required to return to the bound of ϵ on the accuracy of our eigenvector estimate. We can calculate the number of iterations from the formula, but instead let's look at two cases of interest:

1. δ is much larger than ϵ : This is the common case when very accurate estimates are required, or when one can not iterate the eigencomputation frequently. In this case, we imagine that $\epsilon + \delta$ is effectively δ , and as such

$$\|(I - P_B^{(k)})P_X^{(k)}\|_2 \leq \left(\frac{\sigma_{k+1}(B)}{\sigma_k(B)} \right)^{2i} \cdot \frac{\delta}{1 - \delta^2}$$

Assuming that δ is closer to zero than to one, we see that the number of iterations required to return this bound to ϵ is

$$i \geq \frac{\log 2\delta/\epsilon}{2 \log(\sigma_{k+1}(B)/\sigma_k(B))}$$

The relative sizes of δ and ϵ drive this bound.

2. ϵ is much larger than δ : This can occur when only rough estimates of the eigenvectors are required, or when one is diligent in performing iterations frequently enough that large change does not occur at once. In this case, we imagine that $\epsilon + \delta$ is effectively ϵ , and as such

$$\|(I - P_B^{(k)})P_Y^{(k)}\|_2 \leq \left(\frac{\sigma_{k+1}(B)}{\sigma_k(B)}\right)^{2i} \cdot \frac{\epsilon}{1 - \epsilon^2}$$

We see that the number of iterations required is

$$\text{iterations} \geq \frac{-\log(1 - \epsilon^2)}{2\log(\sigma_{k+1}(B)/\sigma_k(B))}$$

Here the number of iterations is determined only by the magnitude of ϵ . Typically $1 - \epsilon$ will be bounded away from zero, and so the number of iterations can be thought of as independent of ϵ, δ in this case.

6.3 Decentralized Eigencomputation

There are a many of settings in which it is difficult, either from a computational or social perspective, to collect and process the graphs that we intend to analyze. In the setting of the web graph the data is so massive that it may take many months with several dedicated links to simply collect the data; processing it once collected then requires a dedicated cluster of fully equipped machines. In the setting of the instant messenger graph, the link information is not publicly available, and in many cases the participants have reason to prefer that this information remain unavailable.

In this section we will observe a technique which pushes the computation of the SVD out to the nodes that comprise the graph, using the assumption that a non-zero matrix entry A_{ij} implies that nodes i and j can communicate. The nodes will communicate along their links, and each perform a trivial amount of computation. Their participation will allow us to compute the singular value decomposition of the graph more effectively than in a central setting:

1. The time taken to compute the singular value decomposition can be much less in the decentralized setting. As each node contributes computing power, we can compute the

SVD in time which is poly-logarithmic in the graph size. The aggregate computation used will not be much larger than that used by the centralized approach.

2. We will require *no* central control. No single participant needs to provide a significant investment, allowing a collection of simple peers to perform the operation. By the same token no node can stop the others from performing the computation, and concerns of authoritarian control (via censorship, for example) are assuaged.
3. We do not collect the link data. In fact, all that each node sees is aggregate information collected by its neighbors. We will see that there are both efficiency and privacy implications of this feature.

6.3.1 Decentral Orthogonal Iteration

Our approach is to adapt Orthogonal Iteration to a decentralized environment. Each node i will take full responsibility for the rows of X associated with it, denoted X_i . Establishing the vector X from $P_X^{(k)}$ is not difficult, as $P_X^{(k)}$ is typically specified as the outer product of two $n \times k$ orthogonal matrices, the first being an excellent choice for X . As well, it is not difficult for each node to compute $(AA^T X)_j$: Note first that $(A^T X)_i$ is computed by

$$(A^T X)_i = \sum_j A_{ij} X_j$$

As each non-zero A_{ij} implies a communication link between i and j , this value may be computed by having each node i share X_i with its neighbors. Likewise,

$$(AA^T X)_i = \sum_j A_{ji} (A^T X)_j$$

which may be computed by having each node i share $(A^T X)_i$ with its neighbors.

The central difficulty with applying Orthogonal Iteration in a decentral setting is the orthonormalization step, and it is this operation that we now tackle. There are two steps to our approach: first, we show that the process of orthonormalization can be performed locally using a matrix which is small enough to be held by each node. This matrix is the $k \times k$ matrix of inner products of the columns of $AA^T X$. Second, we show a decentral

process which produces an arbitrarily good approximation this matrix, based on decentral summation. Each node is able to compute its contribution to each of the inner products, and the sum of these contributions results in the desired matrix.

We first describe how to orthonormalize a matrix using only the matrix of inner products. To orthonormalize a collection of columns $V = AA^T X$ in a central setting, one typically uses the QR factorization of V , i.e. matrices Q, R such that $V = QR$, the k columns of Q are orthonormal, and the $k \times k$ matrix R is upper triangular. Orthonormalization is performed by computing R and applying R^{-1} to V , yielding Q . If each node had access to R , it could locally compute $V_i R^{-1} = Q_i$.

To compute R without explicitly collecting the columns of V , note that if we define $K = V^T V$,

$$\begin{aligned} K &= R^T Q^T Q R \\ &= R^T R \end{aligned}$$

The decomposition of any symmetric positive definite matrix, such as K , into LL^T for a lower triangular matrix L is called the Cholesky decomposition, and is both unique and easily computed. With a copy of K in hand, each node could thus compute $R = L^T$, and by way of R^{-1} conduct the orthonormalization.

Unfortunately, it is unclear how to provide each node with the precise matrix K . Instead, we will have each node compute an approximation to K of arbitrarily high accuracy. To see how, observe that $K = \sum_i V_i^T V_i$. Each node i is capable of producing $V_i^T V_i$ locally, and if we can sum these matrices decentrally, each node can thereby obtain a copy of K .

To compute this sum of matrices in a decentralized fashion, we employ a technique proposed by Kempe et al in [44]: Each node maintains a value, initialized by the node. Nodes then update their values by repeated application of a stochastic matrix, setting their value equal to the sum of the contributions of its neighbors. Under certain assumptions on the stochastic matrix, these values will converge to sum of the initial values, weighted by the nodes stationary probability.

We will use for M the transition matrix defined by a random walk on the communication

Algorithm 4 PushSum (M, val_i, t)

- 1: All nodes synchronously perform
 - 2: **for** $1 \dots t$ **do**
 - 3: Set $val_i = \sum_j M_{ij}val_j$
 - 4: **end for**
 - 5: Return val_i .
-

graph. This has the advantage that all terms in the sum $\sum_j M_{ij}val_j$ that are non-zero represent a communication link. This sum is therefore easily performed by having each node send its value down each of its incident links.

To approximate K , we start each node with their contribution to K , $V_i^T V_i$. After many iterations, this converges to $K\pi_i$ for each node, where π is the stationary distribution for M . To extract K from this quantity, we conduct a parallel execution of PushSum, where each node begins with a value $w_i = 0$, except for one node which has a value of $w_i = 1$. This process will converge to π_i at each node, and we may divide the result of the first PushSum by that of the second to get our approximation to K .

At each node, this ratio converges to K at essentially the same speed as the random walk on the communication network converges to its stationary distribution, as described by the following theorem

Theorem 40 Let K_i^t be the $k \times k$ matrix held by node i resulting from $PushSum(V_i^T V_i, t)$, and let w_i^t be the scalar held by node i resulting from $PushSum(w_i, t)$. For any ϵ , after $t \gg \tau_{\text{mix}} \cdot \log(1/\epsilon)$ rounds

$$\|K_i^t/w_i^t - K\|_F \leq \epsilon k \|V\|_F^2$$

Proof. The proof of this theorem appears in Kempe et al [45]. ■

Combining this orthonormalization process with the decentral computation of $AA^T X$, we obtain the following decentral algorithm for eigencomputation, as executed at each node:

Algorithm 5 Decentralized Orthogonal Iteration ($A, X, iter, t$)

- 1: Each node i synchronously performs:
 - 2: **for** $1 \dots iter$ **do**
 - 3: Set $V_i = (AA^T X)_i = \sum_{j,k} A_{ij} A_{jk} X_k$.
 - 4: Set $K = \text{PushSum}(V_i^T V_i, t) / \text{PushSum}(w_i, t)$.
 - 5: Compute the Cholesky factorization $K = R^T R$.
 - 6: Set $X_i = V_i R^{-1}$.
 - 7: **end for**
 - 8: Return X
-

We have been fairly casual about the number of iterations that should occur, and how a common consensus on this number is achieved by the nodes. One simplistic approach is to simply have the initiator specify a number of iterations, and keep this amount fixed throughout the execution. More sophisticated approaches exist, but are not discussed here.

6.3.2 Analysis

We now analyze the convergence properties of Decentralized Orthogonal Iteration, and prove the following main theorem:

Theorem 41 *For any matrix A and orthonormal X , let $Y = \text{OrthogonalIteration}(A, X, i)$. For X such that $\|R^{-1}\|_2$ is consistently upper bounded by c , and for $t \gg 4i\tau_{\text{mix}} \log(\|A\|_2^2 c / \epsilon)$, letting $\hat{Y} = \text{DecentralizedOrthogonalIteration}(A, X, i, t)$ it is the case that*

$$\|P_Y^{(k)} - P_{\hat{Y}}^{(k)}\|_F \leq 4\epsilon^{4t}$$

Proof. Using the triangle inequality, and the fact that $\|Y\|_2 = \|\hat{Y}\|_2 = 1$,

$$\begin{aligned} \|P_Y^{(k)} - P_{\hat{Y}}^{(k)}\|_F &= \|YY^T - \hat{Y}\hat{Y}^T\|_F \\ &\leq \|(Y - \hat{Y})Y^T\|_F + \|\hat{Y}(Y^T - \hat{Y}^T)\|_F \\ &\leq 2\|Y - \hat{Y}\|_F \end{aligned}$$

Lemma 42 bounds $\|Y - \hat{Y}\|_F$ for $i = 1$, and repeated application of this lemma gives us the desired bound for general i . ■

The main focus of our analysis is to deal with the approximation errors introduced by the PushSum algorithm. The error $\|K - K_i^t\|_F$ drops exponentially in t , but after any finite number of steps each node is still using different approximations to K , and thus to R^{-1} . We must analyze the number of iterations of PushSum required to keep the error sufficiently small even after the accumulation over multiple iterations of Orthogonal Iteration.

Lemma 42 *Let X and \widehat{X} be arbitrary $n \times k$ matrices, where X is orthonormal. Let $X' = \text{OrthogonalIteration}(A, X, 1)$ and let $\widehat{X}' = \text{DecentralizedOrthogonalIteration}(A, \widehat{X}, 1, t)$. If*

$$\|X - \widehat{X}\|_F + \epsilon k^2 \leq (2\|A\|_2^2 \|R^{-1}\|_2)^{-3}$$

and $t \gg \tau_{mix} \log(1/\epsilon)$ then

$$\|X' - \widehat{X}'\|_F \leq (2\|A\|_2^2 \|R^{-1}\|_2)^4 (\|X - \widehat{X}\|_F + \epsilon k^2).$$

Proof. The proof consists of two parts: First, we apply perturbation results for the Cholesky decomposition and matrix inversion to bound $\|R^{-1} - \widehat{R}_i^{-1}\|_2$ for all i . Second, we analyze the effect of applying the matrices \widehat{R}_i^{-1} to the rows \widehat{V}_i instead of R^{-1} to the rows V_i .

Notationally, we let $V = AA^T X$ and $\widehat{V} = AA^T \widehat{X}$ as well as $K = V^T V$ and $\widehat{K} = \widehat{V}^T \widehat{V}$. Each node i will compute a matrix \widehat{K}_i which approximates \widehat{K} , and from it compute \widehat{R}_i^{-1} .

First off, notice that $\|V\|_2 = \|AA^T X\|_2 \leq \|A\|_2^2$, and the definitions $V = AA^T X$ and $\widehat{V} = AA^T \widehat{X}$ give us that $\|V - \widehat{V}\|_F \leq \|A\|_2^2 \|X - \widehat{X}\|_F$. Combining this bound with the triangle inequality, we bound

$$\begin{aligned} \|K - \widehat{K}\|_F &= \|V^T V - \widehat{V}^T \widehat{V}\|_F \\ &\leq \|V^T (V - \widehat{V})\|_F + \|(V^T - \widehat{V}^T) \widehat{V}\|_F \\ &\leq 2\|A\|_2^4 \|X - \widehat{X}\|_F \end{aligned}$$

Next, to bound $\|K - \widehat{K}_i\|_F$, we use the triangle inequality, noting that our choice of t implies that $\|\widehat{K}_i - \widehat{K}\|_F \leq \epsilon k \|V\|_F^2 \leq \epsilon k^2 \|A\|_2^4$ for all i .

$$\|K - \widehat{K}_i\|_F \leq \|K - \widehat{K}\|_F + \|\widehat{K} - \widehat{K}_i\|_F$$

$$\begin{aligned}
&\leq 2\|A\|_2^4\|X - \widehat{X}\|_F + \epsilon k^2\|A\|_2^4 \\
&\leq 2\|A\|_2^4(\|X - \widehat{X}\|_F + \epsilon k^2)
\end{aligned}$$

We apply two well-known theorems to bound the propagation of errors in the Cholesky factorization and matrix inversion steps. First, a theorem by Stewart [67] states that if $K = R^T R$ and $\widehat{K} = \widehat{R}^T \widehat{R}$ are the Cholesky factorizations of K and \widehat{K} , then $\|R - \widehat{R}\|_F \leq \|K^{-1}\|_2\|\widehat{K} - K\|_F\|R\|_2$, which, applied to our setting, yields (using $\|K^{-1}\|_F \leq \|R^{-1}\|_F^2$)

$$\begin{aligned}
\|R - \widehat{R}_i\|_F &\leq \|K^{-1}\|_2\|K - \widehat{K}_i\|_F\|R\|_2 \\
&\leq 2\|A\|_2^6\|R^{-1}\|_2^2(\|X - \widehat{X}\|_F + \epsilon k^2)
\end{aligned}$$

Before continuing, we should note that our assumption on the size of $\|X - \widehat{X}\|_F + \epsilon k^2$ bounds $\|R - \widehat{R}_i\|_F \leq \|R^{-1}\|_2^{-1}/2$.

Next, we apply Wedin's Theorem [71], used earlier in chapter 4, which bounds

$$\|R^{-1} - \widehat{R}_i^{-1}\|_2 \leq \|R^{-1}\|_2\|\widehat{R}_i^{-1}\|_2\|R - \widehat{R}_i\|_2$$

To bound $\|\widehat{R}_i^{-1}\|_2$, recall that the singular values of \widehat{R}_i are perturbed from those of R by at most $\|R - \widehat{R}_i\|_2 \leq \|R^{-1}\|_2^{-1}/2$. This imposes a relative perturbation of at most a factor of 2 in the singular values of $\|\widehat{R}_i^{-1}\|$. Putting this all together,

$$\begin{aligned}
\|R^{-1} - \widehat{R}_i^{-1}\|_2 &\leq 2\|R^{-1}\|_2^2\|R - \widehat{R}_i\|_2 \\
&\leq 4\|R^{-1}\|_2^4\|A\|_2^6(\|X - \widehat{X}\|_F + \epsilon k^2),
\end{aligned}$$

concluding the first half of the proof.

In the second half of the proof, we analyze the error incurred by each node i applying its own matrix \widehat{R}_i^{-1} to \widehat{V}_i . This is a non-linear operation applied to \widehat{V} , and so instead of arguing in terms of matrix products, we must perform the analysis on a row-by-row basis. We can write the i th row of $\widehat{X}' - X'$ as

$$\begin{aligned}
\widehat{X}'_i - X'_i &= \widehat{V}_i\widehat{R}_i^{-1} - V_iR^{-1} \\
&= (\widehat{V}_i - V_i)\widehat{R}_i^{-1} + V_i(\widehat{R}_i^{-1} - R^{-1}).
\end{aligned}$$

We will let $C_i = (\widehat{V}_i - V_i)\widehat{R}_i^{-1}$, and $D_i = V_i(\widehat{R}_i^{-1} - R^{-1})$ and go on to bound $\|C\|_F$ and $\|D\|_F$ separately, thereby bounding $\|X' - \widehat{X}'\|_F$. To bound $\|C\|_F$, observe that

$$\begin{aligned}\|C\|_F^2 &= \sum_i \|(\widehat{V}_i - V_i)\widehat{R}_i^{-1}\|_2^2 \\ &\leq \sum_i \|\widehat{V}_i - V_i\|_2^2 \|\widehat{R}_i^{-1}\|_2^2 \\ &\leq \|\widehat{V} - V\|_F^2 \cdot \max_i \|\widehat{R}_i^{-1}\|_2^2.\end{aligned}$$

Similarly, to bound the Frobenius norm of D we write

$$\begin{aligned}\|D\|_F^2 &= \sum_i \|V_i(\widehat{R}_i^{-1} - R^{-1})\|_2^2 \\ &\leq \|V\|_F^2 \cdot \max_i \|\widehat{R}_i^{-1} - R^{-1}\|_2^2.\end{aligned}$$

We combine these two bound (taking square roots first), recalling our established bounds on $\|\widehat{R}_i^{-1}\|_2$ and $\|R^{-1} - \widehat{R}_i^{-1}\|_2$, and our bounds on $\|V - \widehat{V}\|_F$ and $\|V\|_F$.

$$\begin{aligned}\|X' - \widehat{X}'\|_F &\leq \|\widehat{V} - V\|_F \cdot \max_i \|\widehat{R}_i^{-1}\|_2 + \|V\|_F \cdot \max_i \|\widehat{R}_i^{-1} - R^{-1}\|_2 \\ &\leq 2\|A\|_2^2 \|R^{-1}\|_2 \|\widehat{X} - X\|_F + 4\|A\|_2^8 \|R^{-1}\|_2^4 (\|X - \widehat{X}\|_F + \epsilon k^2) \\ &\leq (2\|A\|_2^2 \|R^{-1}\|_2)^4 (\|X - \widehat{X}\|_F + \epsilon k^2)\end{aligned}$$

completing the proof. ■

Lemma 42 indicates that $\|X - \widehat{X}\|_F$ effectively grows by a factor of up to $(2\|R^{-1}\|_2 \|A\|_2^2)^4$ with each iteration, starting at ϵk^2 . While this exponential growth is worrisome, Theorem 40 shows that the error ϵ diminishes at a rate exponential in the number of PushSum steps. Choosing t sufficiently large, we make ϵ sufficiently small to counteract the growth of $\|X - \widehat{X}\|_F$.

6.3.3 Discussion

The main assumption of Theorem 41, that $\|R^{-1}\|_2$ is bounded, raises an interesting point. $\|R^{-1}\|_2$ becoming unbounded corresponds to the columns of X becoming linearly dependent, an event that is unlikely to happen outside of matrices A of rank less than k . Should it happen, the decentralized algorithm will deal with this in the same manner that the central

algorithm does: The final degenerate columns of X will be filled uniformly with garbage. This garbage will then serve as the beginning of a new attempt at convergence; these columns are kept orthogonal to the previous non-degenerate ones, and should converge to lesser eigenvectors. The difference between the centralized and decentralized approaches is precisely which garbage is used, and we are unable to prove that it will be the same.

Notice that even if $\|R^{-1}\|$ is large for some value of k , it may be bounded for smaller values of k . $\|R^{-1}\|_2$ becomes large when a degenerate column occurs, and if the preceding columns are still non-degenerate we may characterize their behavior by performing our analysis with a smaller value of k . In fact, to analyze the accuracy of column j it is best to perform the analysis $k = j$, as any larger k will provide a strictly less tight bound.

6.4 Conclusions

In this chapter we examined several pitfalls commonly associated with eigencomputation: its tendency to be large, its tendency to change, and its tendency to be distributed remotely. For each of these problems we have seen remedies, and analyzed their efficacy. Our techniques revolve around the now frequently observed fact that many spectral properties of matrices are robust in the presence of noise. We used random perturbation to simplify the data, accelerating the computation of a low rank approximation which approximates the original well. We observed that incremental changes to the matrix could be accommodated by continuing Orthogonal Iteration from the previous singular vectors, as they would be close to the new solution. We finally observed that one can compute singular vectors in a manner that is resilient to error, even when different participants see different results, allowing us to decentralize the computation of singular vectors.

Chapter 7

CONCLUSIONS

7.1 Contributions

In this thesis we have analyzed many results surrounding the application of spectral methods. They can be divided roughly into three classes. First, the modeling of input data and problem dependent error, which characterizes the phenomena in data that we might hope to recover, as well as the phenomena we may be able to filter out. Second, the design of algorithms to solve several of the data mining problems posed; information retrieval, collaborative filtering, and web search being three examples. Finally, the improvement of general spectral methods, through accelerated, incremental, and decentral eigencomputation.

7.1.1 Data Modeling

As an integral part of understanding when spectral analysis can be used, and what type of structure it recovers, this thesis examined several important models for data. We have modeled both the structure underlying “meaningful data” as well as random phenomena that can be used to describe the discrepancy between the meaningful data and what is observed. When the observed data can be described as meaningful data plus random error, where each entry is independent, mean zero, and of limited variance, then we have general results that indicate that spectral analysis will perform well.

We have looked at several models of “structured data” that suits spectral analysis. While there are an abundance of technical characterizations, the one that seems both technically precise and intelligible is that of *latent semantics*. The term was introduced by Dumais et al in [26], and corresponds to the existence of a small number (k) of latent attributes, where each entity, be it document and term, page and link, or person and product, is described by a k -vector giving a value to each of these attributes. These attributes are such that the

correspondence between any two entities is equal to the inner product of their k -vectors.

While much data and many correspondences may be structured as above, we rarely observe any data that is. The data is almost certainly constrained by many practical phenomena: terms exist in documents in integral capacity, people do not have the time to rate their utility for every product. It is therefore important to understand which phenomena can have a significant effect on spectral methods, and which phenomena are relatively harmless. Perhaps more importantly, in understanding which phenomena are benign, we can look for ways to transform the data so that significant error becomes benign error, as done for collaborative filtering in Section 3.4.

The characterization of benign random phenomena that we achieve centers around three properties of the difference between the observed data and the expected data:

1. The entries should be independently distributed.
2. The entries should have mean zero.
3. The entries should have moderate variance.

We have seen several phenomena that fit this characterization, the most interesting being quantization, as would occur when correspondences occur or not with some probability, and omission, where entries are simply replaced by zero (or some “absent” placeholder). As noted above, omission actually requires a transformation to the data to become benign, and it was only through understanding what makes a phenomena benign that we determined the correct transformation (namely, the scaling of observed data by each’s observation probability).

7.1.2 Algorithm Design

We have examined many data mining problems, producing varied solutions based on spectral methods. For the problems of data cleaning, information retrieval, and collaborative filtering we have seen that the optimal low rank approximation to the input matrix can serve as a good approximation to the original data, from which we establish the correct answer. We

examined a detailed algorithm for web search, which takes two sources of input data, text and links, and learns the correlation between the two, allowing us to transform search queries into linked pages. Finally, we considered a class of algorithms for random graphs which seek out planted structure in these graphs, in the form of cliques, colorings, and bisections, and we then unified these results into a common framework, simplifying the analysis in the process.

In Chapter 3 we studied several problems that have fallen under the heading of data mining. Data cleaning, information retrieval, and collaborative filtering are each tasks with a rich history and a large collection of approaches. One technique that had seen much unexplained success was LSA, or *latent semantic analysis*, whereby one uses a low rank approximation to the document term matrix to compute similarities instead of the document term matrix itself. Papadimitriou et al provide a preliminary justification for this success in [61], and we are much indebted to this work in motivating the formal study of spectral methods in the data mining setting. We generalized their results to a broader class of input matrices, admitting polysemous terms, documents of mixed topic composition, and much greater deviation from the expected matrix in the form of random error. In understanding the requirements for this analysis, it became clear that collaborative filtering could be addressed in a similar manner, so long as one normalized each entry by its observation probability.

Chapter 4 conducted an in depth analysis of the role of spectral methods in web search. While there have been several previous spectral efforts in web search, HITS [47], Google [37], Salsa [54], and Manjara [55] describing a representative sample, none provided an end to end analysis of web search; each relies on an external component, typically a method for producing good candidates for ranking. We describe an algorithm that unifies the text retrieval benefits of LSA with the spectral ranking of most web search algorithms. It operates on the belief that the same latent semantics underlie the generation of text and links, and by conducting a unified analysis of the data we may learn the connection. In the context of such a model, outlined explicitly in Section 4.3, our algorithm provides results that approach optimal as the number of search terms increases.

Finally, in Chapter 5 we observed a common thread between many algorithms for mining

structured random graphs. Such graphs have an interesting combinatorial object embedded in them, such as a large clique, small coloring, or noteworthy bisection, but are otherwise random. Many approaches have emerged for recovering these objects, and the algorithms that uniformly outdistanced the others were each spectral in nature. However, each approach was specialized, and the analysis largely opaque. We unified these results through the simple observation that for each of these models, the random graph is defined by a $k \times k$ block matrix of probabilities. We then present an algorithm which on such an input is capable of recovering the latent partition, corresponding in the special cases to recovering the clique, coloring, and bisection. The generalization admits solution of more general problems not discussed before, such as the identification of abnormally sparse/dense subgraphs.

7.1.3 Improved Eigencomputation

There are three techniques that we have considered to improve and extend traditional eigencomputation. Each technique is simple enough that we only need to apply a modified form of orthogonal iteration, and each can be combined with the others and thereby enjoy all the beneficial properties. All approaches are based around the perturbation theory of random matrices, leveraging the observation that we may efficiently compute nearly optimal solutions, and are much more efficient. All approaches provide sharply stated high probability bounds on their performance.

Traditional eigencomputation methods such as orthogonal iteration and Lanczos iteration operate by repeated matrix vector multiplication. We considered the approach of applying random sparsification and quantization to entries of a matrix, and observed that the optimal approximation to the resulting matrix is nearly optimal for the original matrix as well. For both the Frobenius and L2 norm the amount of error that is introduced into the approximation is proportional to the L2 norm of the random matrix that is the difference between the initial and perturbed matrices. We have argued in previous chapters that this amount is relatively small, and if the data can not withstand this degree of perturbation one should not expect spectral methods to work in the first place.

Most, if not all of the large data sets that present challenges to data miners undergo only

incremental changes. While individual rows and columns may come and go, the fundamental trends that underlie the data set do not change rapidly. We have shown an incremental approach to eigencomputation which makes use of previous results. The main observation is that if the input matrix is perturbed slightly, then only a few iterations of orthogonal iteration are required to take the old eigenvectors to the new. The number of steps required is proportional to the change that has been effected, and so the more frequently the recomputation is performed, the less time it takes, leading to a positive feedback cycle.

Finally, we examined orthogonal iteration in a purely decentral setting, where each node is only capable of limited computation, communication, and storage. In such a setting, no small group of nodes can perform the eigencomputation without the cooperation of the other nodes in the network. We have seen a decentralized implementation of orthogonal iteration which maps precisely onto the communication network of the graph. This algorithm naturally and efficiently distributes the computation amongst all nodes, resulting an an implementation where, if we seek k singular vectors over i iterations, each node performs $O(k^3i^2)$ computation, transmits $O(k^2i^2)$ messages, and uses $O(k^2)$ space. What is more, the entire computation concludes in $O(k^3i^2)$ steps; far sooner than any distribution of the $\Theta(|E|i)$ computation over any small set of nodes. Finally, the approach has several beneficial implications, notably enabling both natural freshness and a form of privacy in the eigencomputation; issues which have arisen in traditional eigencomputation.

7.2 *Future Research*

We now highlight several areas of research which are suggested by this thesis.

7.2.1 *Experimental Evaluation*

Much as experimental results, in the form of the unexplained success of LSA [26], initially gave rise to this line research, the time appears right to complete the circle and bring the theories and understanding formed herein back to the practical realm. Several of the theoretical results have stumbled on certain issues that have largely been ignored in practice. These issues are resolved theoretically through algorithmic alteration, and it is worth

evaluating these changes in practice to see what gain can be had.

Degree Normalization

A significant problem, first noted by Mihail et al in [57], lies in sparse graphs with non-uniform degree distributions. These graphs, including the ubiquitous graphs with power law degree distributions, will have their spectrum dominated by the nodes of highest degree, resulting in essentially meaningless principal eigenvectors. The issue, it appears, lies in the fact that the variances in the entries of such a random graph model are not uniform; in a simple model, the variance of matrix entry A_{ij} is proportional to $\sqrt{d_i d_j}$, the geometric mean of the degrees. This non-uniformity in variance means that when applying Theorem 16 we must use as the bound the largest variance in the matrix, which has little to do with the variance of the typical entry.

Work has begun on an algorithmic modification to address this issue. In particular, given a matrix A and a diagonal matrix D where D_{ii} is the degree of node i , we construct the matrix

$$L = D^{-1/2} A D^{-1/2}$$

Notice that if the edges from a node are the result of sampling d_i times from a given distribution, the variance of each row will scale with d_i . This normalization factor is chosen so as to unify each of the variances, and the bound we attain using Theorem 16 is now representative of the average variance.

Initial empirical results in graph partitioning suggest that this algorithmic modification is crucial to good performance. Indeed, the deficiencies observed in [57] do appear, centering parts trivially around the nodes of highest degree. Using the matrix L instead, the partitioning results improve dramatically.

Cross Training

Much trouble is caused theoretically by the projection of a random data set onto its own eigenvectors. This summons the bugaboo of conditioning of random variables, and it becomes impossible to argue about the concentration of any one row or column near its

expectation. In Theorem 21 we argued that such projection yields a bounded total squared error, but few guarantees could be made about any one point. Far preferable is to show that the error is spread out, and that each node is unlikely to deviate far from its expectation.

This issue was addressed theoretically by cross training; the data set is split in two pieces, each producing the subspace on to which the other is projected. The conditioning between subspace and random vectors is now gone, and the belief is that each of the produced subspaces should be nearly as good as the original. There is now an empirical tradeoff to examine: is it more valuable to remove the conditioning at the expense of fewer examples in each eigencomputation, or is the conditioning harmless in contrast with the instability that may result from only training on half the examples? Further, is there merit in performing more computationally expensive cross-training, for example projecting each column onto the subspace produced by the other $n - 1$ columns?

7.2.2 Data Compression

A recent algorithm of Kleinberg and Sandler for collaborative filtering [46] gives us a marvelous example of how to reformulate the interface to a problem to produce data that is substantially more useful. Collaborative filtering as described in chapter 3 suffers in practice from the general reluctance of users to provide arbitrarily large amounts of data. Kleinberg and Sandler address this issue by not studying the *person* \times *product* utility matrix U , but rather the *product* \times *product* covariance matrix $U^T U$. By asking each user to choose two products i, j at random and evaluate the product of their utility, adding the result to entry $[U^T U]_{ij}$, they produce a matrix which approaches the covariance matrix as the user base increases, and yet no one user must provide more than two pieces of data.

This approach provides even more benefits than simply enabling data mining for easily distracted users. First the problems of learning and recommendation are decoupled. In the approach of Chapter 3 the user provides one set of data, from which the system must both learn and provide recommendations. In the approach of [46], there is an initial learning stage which requires very little commitment from the users. Once the learning is done, users may ask queries of arbitrary precision, by presenting item sets of arbitrary complexity. The

quality of recommendations is therefore controller at query time, not established *a priori* in the data collection phase

Second, there appear to be very solid privacy results about this new formulation. As each user need only provide information about two entries in his vector, there is little that can be said about the entirety of the user’s data. Chawla et al, in [20], argue that privacy of a user is preserved so long as an adversary is unable to produce a point which is much closer to one data point than it is to any other data points. This “isolation” indicates that the adversary was able to learn most of the attributes of a data point to an alarming accuracy. In the setting of [46] each individual only presents information about at most two of their attributes, leaving the adversary with literally no information about the others.

Finally, it appears that this approach has the ability to overcome certain theoretical shortcomings of the analysis of the *person* \times *product* utility matrix. Namely, the random perturbation theory that we have established does not fare terribly well in the presence of unshapely matrices; the less square the matrix, the less useful the perturbation bounds. By considering the covariance matrix, one is guaranteed a square matrix, and the number of users growing large only benefits the accuracy of the matrix.

7.2.3 Spectral Methods and Data Fusion

Data Fusion is a name applied to techniques which take multiple facets of the same data set and attempt to analyze them in concert. For example, the web contains pages, links, and terms, and it would not be hard to imagine that would could analyze the three of them together. We began this in Chapter 4, learning the term-link correlation by way of their correlations with pages. However, it appears that there is a more rich and less ad-hoc approach to this problem.

A tensor is a high dimensional analog of a matrix. Whereas a matrix can be viewed as data indexed by two coordinates, row and column, a d dimensional tensor is indexed by d attributes. Web data, for example, could be viewed as a collection of triples (p, t, l) indicating that page p uses term t to link to page l . Much as we were able to reconstruct matrices and thereby predict entries and rank pages, we might like to do the same with

tensors.

BIBLIOGRAPHY

- [1] Dimitris Achlioptas, Amos Fiat, Anna Karlin, and Frank McSherry. Web search via hub synthesis. *Foundations of Computer Science*, pages 500 – 509, 2001.
- [2] Dimitris Achlioptas and Frank McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the thirty-third annual ACM symposium on the theory of computing*, pages 611 – 618, 2001.
- [3] N. Alon. Eigenvalues and expanders, 1986.
- [4] Noga Alon and Nabil Kahale. A spectral technique for coloring random 3-colorable graphs. *SIAM Journal on Computing*, 26(6):1733–1748, 1997.
- [5] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13:457–466, 1998.
- [6] Brian Amento, Loren G. Torveen, and Willuam C. Hill. Does authority mean quality? predicting expert ratings of web documents. In *Research and Development in Information Retrieval*, pages 296–303, 2000.
- [7] Yossi Azar, Amos Fiat, Anna Karlin, Frank McSherry, and Jared Saia. Data mining through spectral analysis. *Symposium on the Theory of Computation*, pages 619 – 626, 2001.
- [8] Michael W. Berry, Zlatko Drmač, and Elizabeth R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Rev.*, 41(2):335–362 (electronic), 1999.
- [9] Michael W. Berry, Susan T. Dumais, and Gavin W. O’Brien. Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4):573–595, 1995.

- [10] Krishna Bharat and Monika Rauch Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Research and Development in Information Retrieval*, pages 104–111, 1998.
- [11] Avrim Blum and Joel Spencer. Coloring random and semi-random k -colorable graphs. *Journal of Algorithm*, 19:204 – 234, 1995.
- [12] Ravi B. Boppana. Eigenvalues and graph bisection: An average-case analysis. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 280 – 285, 1985.
- [13] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Finding authorities and hubs from link structure on the world wide web.
- [14] Sergei Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 30(1–7):107–117, 1998.
- [15] Thang N. Bui, Soma Chaudhuri, F. Thomas Leighton, and Michael Sipser. Graph bisection algorithms with good average case behavior. In *Proceedings of the 25th IEEE Symposium on the Foundations of Computer Science*, pages 181–192, 1984.
- [16] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Mining the web’s link structure. In *Computer*, pages 32(8):60–67, 1999.
- [17] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S.R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Hyper-searching the web. In *Scientific American*, June 1999.
- [18] S. Chakrabarti, B. Dom, D. Gibson, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Spectral filtering for resource discovery. 1998.

- [19] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Computer Networks and ISDN Systems*, pages 30(1–7):65–74, 1998.
- [20] Shuchi Chawla, Cynthia Dwork, Frank McSherry, Adam Smith, Larry Stockmeyer, and Hoeteck Wee. From idiosyncratic to stereotypical: Preserving privacy in public databases. *In preparation*, 2003.
- [21] David Cohn. The missing link - a probabilistic model of document content and hyper-text connectivity.
- [22] Anne Condon and Richard Karp. Algorithms for graph partitioning on the planted partition model. *Random Structure and Algorithms*, 8(2):116 – 140, 1999.
- [23] Anirban Dasgupta, John Hopcroft, and Frank McSherry. Spectral partitioning in graphs with skewed degree distributions. *In submission*, 2003.
- [24] Chandler Davis and William Kahan. The rotation of eigenvectors by a perturbation 3. *SIAM Journal on Numerical Analysis*, 7:1–46, 1970.
- [25] Jeffrey Dean and Monika Rauch Henzinger. Finding related pages in the world wide web. In *WWW8 / Computer Networks*, pages 31(11–16):1467–1479, 1999.
- [26] Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.
- [27] Petros Drineas, Alan Frieze, Ravi Kannan, Santosh Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, MD, 1999)*, pages 291–299, 1999.
- [28] Petros Drineas, Ravi Kannan, Alan Frieze, Santosh Vempala, and V. Vinay. Clustering in large graphs and matrices, 1999.

- [29] Martin E. Dyer and Alan M. Frieze. Fast solution of some random NP-hard problems. In *IEEE Symposium on Foundations of Computer Science*, pages 331–336, 1986.
- [30] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. In *Symposium on Principles of Database Systems*, 2001.
- [31] Uriel Feige and Joe Kilian. Heuristics for finding large independent sets, with applications to coloring semi-random graphs. *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 674 – 683, 1998.
- [32] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *39th Annual Symposium on Foundations of Computer Science (Palo Alto, CA, 1998)*, pages 370–378, 1998.
- [33] Alan Frieze and Colin McDiarmid. Algorithmic theory of random graphs. *RSA: Random Structures & Algorithms*, 10, 1997.
- [34] Alan M. Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. In *IEEE Symposium on Foundations of Computer Science*, pages 370–378, 1998.
- [35] Zoltán Füredi and János Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.
- [36] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.
- [37] google. Google search engine, 1999.
- [38] Monika Rauch Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. Measuring index quality using random walks on the web. In *WWW8 / Computer Networks*, pages 31(11–16):1291–1303, 1999.

- [39] Thomas Hofmann. Probabilistic latent semantic analysis. In *Research and Development in Information Retrieval*, pages 50–57, 1999.
- [40] Mark Jerrum. Large cliques elude the Metropolis process. *Random Structures and Algorithms*, 3(4):347–359, 1992.
- [41] Mark Jerrum and Gregory B. Sorkin. Simulated annealing for graph bisection. In *IEEE Symposium on Foundations of Computer Science*, pages 94–103, 1993.
- [42] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: good, bad and spectral. In *IEEE Symposium on Foundations of Computer Science*, 2000.
- [43] Richard Karp. The probabilistic analysis of some combinatorial search algorithms. In J. F. Traub, editor, *Algorithms and Complexity, New Directions and Recent Results*, pages 1–20, New York, 1976. Academic Press.
- [44] D. Kempe, A. Dobra, and J. Gehrke. Computing aggregate information using gossip. In *FOCS 2003*, 2003.
- [45] David Kempe and Frank McSherry. A decentral algorithm for spectral analysis. *In submission*, 2003.
- [46] Jon Kleinberg and Mark Sandler. Convergent algorithms for collaborative filtering.
- [47] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
- [48] Flip Korn, Alexandros Labrinidis, Yannis Kotidis, and Christos Faloutsos. Ratio rules: A new paradigm for fast, quantifiable data mining. In *24th International Conference on Very Large Databases (New York, NY, 1998)*, pages 582–593, 1998.
- [49] M. Krivelevich and V. Vu. the concentration of eigenvalues of random symmetric matrices, 2000.

- [50] Michael Krivelevich and Van H. Vu. On the concentration of eigenvalues of random symmetric matrices. In *Microsoft Technical Report*, number 60, 2000.
- [51] Ludek Kucera. Expected behavior of graph colouring algorithms. In *Lecture Notes Comput. Sci 56*, pages 447 – 451, 1977.
- [52] Ludek Kucera. Expected complexity of graph partitioning problems. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 57, 1995.
- [53] S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Recommendation systems: A probabilistic analysis. In *IEEE Symposium on Foundations of Computer Science*, pages 664–673, 1998.
- [54] R. Lempel and S. Moran. Salsa: Stochastic approach for link-structure analysis. In *ACM Transactions on Information Systems*, pages 19:131–160, 2001.
- [55] manjara. Manjara search engine, 1999.
- [56] Frank McSherry. Spectral partitioning of random graphs. *Foundations of Computer Science*, pages 529 – 537, 2001.
- [57] Milena Mihail and Christos Papadimitriou. On the eigenvalue power law. *RANDOM 02*.
- [58] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, Philadelphia, PA, 1995.
- [59] L. Page and S. Brin. Pagerank, an eigenvector based ranking approach for hypertext, 1998.
- [60] Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *17th Annual Symposium on Principles of Database Systems (Seattle, WA, 1998)*, pages 159–168, 1998.

- [61] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. pages 159–168, 1998.
- [62] Beresford N. Parlett. *The symmetric eigenvalue problem*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Corrected reprint of the 1980 original.
- [63] L. Saloff-Coste. Lectures on finite markov chains. In *Lecture Notes in Mathematics 1665*, pages 301–408. Springer, 1997. École d’été de St. Flour 1996.
- [64] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. In *Communications of the ACM*, pages 18:613–620, 1975.
- [65] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains, 1989.
- [66] Daniel Spielman and Shang Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. In *Proc. 37th Conf. on Foundations of Computer Science*, pages 96–105, 1996.
- [67] G. Stewart. On the perturbation of lu and cholesky factors. *IMA Journal of Numerical Analysis*, 1997.
- [68] G. W. Stewart and Ji Guang Sun. *Matrix perturbation theory*. Academic Press Inc., Boston, MA, 1990.
- [69] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [70] Jonathan Turner. Almost all k-colorable graphs are easy to color. *Journal of Algorithms*, 9(1):63–82, 1988.
- [71] P. Wedin. Perturbation theory for pseudo-inverses. *BIT*, 13:217–232, 1973.

VITA

Frank McSherry was born in Burlington, VT, on October 4, 1976. After being fed through the Vermont public school system, Frank went to high school in Concord, Mass, and then went on to acquire a Bachelor's degree from Cornell University in Computer Science and Mathematics.