Learning from Partially Labeled Data:

Unsupervised and Semi-supervised Learning on Graphs and Learning with Distribution Shifting

by

Jiayuan Huang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Scince

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis focuses on two fundamental machine learning problems: unsupervised learning, where no label information is available, and semi-supervised learning, where a small amount of labels are given in addition to unlabeled data. These problems arise in many real word applications, such as Web analysis and bioinformatics, where a large amount of data is available, but no or only a small amount of labeled data exists. Obtaining classification labels in these domains is usually quite difficult because it involves either manual labeling or physical experimentation. This thesis approaches these problems from two perspectives: graph based and distribution based.

First, I investigate a series of graph based learning algorithms that are able to exploit information embedded in different types of graph structures. These algorithms allow label information to be shared between nodes in the graph—ultimately communicating information globally to yield effective unsupervised and semi-supervised learning. In particular, I extend existing graph based learning algorithms, currently based on undirected graphs, to more general graph types, including directed graphs, hypergraphs and complex networks. These richer graph representations allow one to more naturally capture the intrinsic data relationships that exist, for example, in Web data, relational data, bioinformatics and social networks. For each of these generalized graph structures I show how information propagation can be characterized by distinct random walk models, and then use this characterization to develop new unsupervised and semi-supervised learning algorithms.

Second, I investigate a more statistically oriented approach that explicitly models a learning scenario where the training and test examples come from different distributions. This is a difficult situation for standard statistical learning approaches, since they typically incorporate an assumption that the distributions for training and test sets are similar, if not identical. To achieve good performance in this scenario, I utilize unlabeled data to correct the bias between the training and test distributions. A key idea is to produce resampling weights for bias correction by working directly in a feature space and bypassing the problem of explicit density estimation. The technique can be easily applied to many different supervised learning algorithms, automatically adapting their behavior to cope with distribution shifting between training and test data.

Acknowledgements

This thesis is the final prize at the end of an incredible journey full of bumps and turns. Without the company and support of many people for the past four and half years, I could never make it to the end. It is a great pleasure that I now have an opportunity to thank all of them.

I would like to express my first and foremost gratitude to my PhD supervisor — Professor Dale Schuurmans. He had nurtured me with his wide and deep knowledge, his overly enthusiasm and integral view on research, and his encouragement; that had helped me build up great interests, confidence, and determination in pursuing scientific research over these years. He also gave me the freedom to choose topics that I was passionate about and always supported me through the tough road of research development with optimism and encouragement.

I also had opportunities to work with other researchers in different labs during my PhD study. While I was a research intern in the Department of Empirical Inference for Machine Learning and Perception at Max Planck Insitute for Biological Cybernetics(MPI) in Germany, I had collaborated closely with my mentor — Dengyong Zhou. Not only had his teaching and mentoring boosted my ability and confidence in scientific research; but, most importantly, his never diminishing enthusiasm in working on new and challenging problems had been a constant inspiration for me to pursue more challenging goals in life. I would also like to thank Bernhard and other members in MPI for their invaluable mentorship, discussions, and comments.

I also had opportunities to work with Professor Ali Ghodsi and Dr. Finnegan in my first two years of study, and with Dr. Tingshao and Professor Russell Greiner since 2005 after I joined Alberta Ingenuity Centre of Machine Learning(AICML). I leared much from them and enjoyed the collaborations. I also had many informative research conversations with other students when writing the thesis: Linli, Yuxi, Baochun, Tao, Li, Shaojun and Feng.

The visiting research experience in National ICT, Australia(NICTA) also had substantial contribution to this work. I would express the gratitude to Professor Alexander J. Smola and Dr. Tiberio for their invitation. I am grateful to Alex, Arthur, Bernhard and Karsten, team members of the kernel mean matching project, who had taught me such

much about efficient research collaboration. I also appreciate the discussions and therefore established friendship with Jin, Quac, Doug, Justin, Nic, Vish and Steven during my stay in Canberra.

Thanks to the members of my committee, Shai Ben-David, Ali Ghodsi, Pascal Poupart, Dale Schuurmans, Hongyuan Zhang, Mu Zhu, who provided valuable feedback that greatly improved this thesis.

Beyond study, the personal support from my dear friends during these years is significant and priceless. In particular, I would like to thank Xiye, Nick, Xianjie, Eric, Even, Wendy, Baochun, Yuxi, Yang, Robin and Steven for bringing cheers and happiness, and offering ongoing social support.

Finally, I wish to express my love and gratitude to all my family: my parents Wenqi and Wenfei for encouraging me with their persistent inspiration and faith in me. I am grateful for my brother Jiachun, my sister-in-law Xiaofang and my lovely nephew Yurui, for being part of family that take care of my life. Also, a special thanks goes to my husband Qiang, who disclosed a fantastic and romantic page in my life. Thank you, my dearest Qiang — you let me know the truth of love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Two of the most prominent areas of machine learning research are supervised and unsupervised learning respectively. In *supervised learning* a learner attempts to acquire a predictive model from explicitly labeled training examples, while in *unsupervised learning* a learner attempts to extract a descriptive model from unlabeled training examples. Recently, interest has increased in the hybrid problem of learning a predictive model given a combination of both labeled and unlabeled examples. This modified learning problem, generally referred to as *semi-supervised learning*, arises in many real world applications, such as text and gene classification, since unlabeled data is usually freely available, whereas explicitly labeled data is expensive and requires manual effort to obtain. For example, in text classification, great effort is required to manually label a set of documents for supervised training, while at the same time, unlabeled documents are available in abundance. It is natural in this case to attempt to exploit the existence of a large set of unlabeled documents to reduce the number of labeled documents required to learn a good document classifier. Similarly, in the problem of predicting gene function from microarray data and sequence information, the experiments needed to label a subset of the genes are typically very expensive to conduct. As a result, there exist only a few hundred labeled genes out of the population of thousands.

Although it is a challenging problem, semi-supervised learning offers sufficient promise

in practice that many algorithms have been proposed for this type of problem in the past few years. Among the proposals are *graph based* learning algorithms, which have become popular due to their computational efficiency and their effectiveness at semi-supervised learning. Some of these graph based learning algorithms generate predictions directly for a target set of unlabeled data without creating a model that can be used for out-of-sample predictions; a process referred to as *transductive learning*. Such algorithms bypass many of the requirements of traditional supervised learning and can be much simpler as a result. However, other approaches to semi-supervised learning still generate a model that can be used to make predictions unseen test data.

The thesis focuses on learning from partially labeled data—that is, unsupervised learning and semi-supervised learning—approaching them mainly from two perspectives: graph based and distribution based.

In the graph based approach, training examples are represented by vertices in a graph where edges convey the *local* similarity between data examples. However, local information is usually not sufficient to yield accurate predictions, and current graph based learning methods obtain an advantage by propagating information *globally* to obtain more accurate predictions of missing labels. One shortcoming with current work on graph based learning is that it is based on *undirected* graphs. Undirected graphs are appropriate for capturing useful symmetric relations that exist in many real-world applications; for example, the correlations between documents are calculated by symmetric bag of word similarities. However, undirected graphs are not effective at capturing every type of data relationship. For example, there are many challenging problems where the relationship between data items follows a much more complex arrangement. I give some examples to illustrate.

- *Web clustering and classification.* Many challenging problems have been raised due to the success of search engines. Among the fundamental problems of improving search engine services are Web clustering and Web page classification. Web clustering is motivated by the fact that if one could identify Web communities, this could facilitate improved browsing and personalized retrieval services. Web page classification is motivated by the fact that if an intrinsic categorization of Web pages was available, then vertical search services could be enabled. To attack these problems, one might attempt to represent the data in an undirected graph, as suggested above. However,

the structure of the Web graph can not be completely preserved with an undirected graph: directionality information is lost. A more natural choice is to represent the relationships in a *directed* graph, since, intuitively, asymmetric relationships encode important information in the Web, such as whether a page is a Hub or an Authority on some topic.

- *Relational database analysis.* Learning in relational databases involves analyzing data items that are represented by vectors of categorical attributes. One way to determine the similarity of two data items is whether they share any common attributes. If we consider using a graph to represent similarities between items in a relational dataset, one might consider placing an undirected edge between any two vertices (data items) that share a common attribute value; i.e., we could represent the data similarities by an undirected graph. However, a simple binary edge does not indicate whether there exist other vertices sharing the same attribute value. The pairwise relationships encoded in an undirected graph will lead to an information loss in relational data. A more natural choice in this setting is to represent the data relationships by a *hypergraph* where an edge may contain multiple vertices.

- *Citation network analysis.* In a citation network, as well as other natural bipartite graphs, one is often interested in performing a simultaneous clustering of the different object types, e.g., authors and papers. However, in reality one also has additional relevant data items, such as publication venues, that might also be worth clustering, and can moreover contribute to improving the clustering of papers and authors. In general, there may be multiple types of relationships present in a data network; for example, author-paper relationships and paper-conference relationships in a citation graph. For citation networks, note that the author-paper relationship can be represented in a bipartite graph, and the additional paper-conference relationship will turn the bipartite graph into a tripartite graph. In some other applications, one may even encounter higher order of k-partite graphs, if observing multiple object types and relationships. Thus, using a bipartite or tripartite graph representation is still not sufficient in general. A complex, *heterogeneous* network of multiple object types and multiple relations is a more natural representation to consider in these scenarios.

In a nutshell, one would like to be able to use different types of graph representations that preserve the natural data information as much as possible, rather than simply using undirected graphs. Learning will not yield accurate results if the graphical representation loses the natural relationships between the data items. This observation raises an obvious question: Given that important data sets are naturally represented in more general structures than undirected graphs, how can one adapt the existing unsupervised and semi-supervised learning algorithms for undirected graphs to these more general cases? The first part of the thesis answers this question. I extend the existing unsupervised and semi-supervised learning algorithms that have been developed for undirected graphs to directed graphs, hypergraphs and complex heterogeneous networks. In addition to proposing new learning algorithms, I also draw connections between unsupervised, semi-supervised and supervised learning by presenting them in a common regularization framework.

The second part of the thesis investigate a more statistically oriented approach that models a different learning scenario from the traditional one. In most classical research on supervised and semi-supervised learning, there is a hidden assumption that the distribution of the training and test data should be the same, or at least very close. Intuitively, it would be hard to make good predictions on test data if one had to use a model inferred from training drawn from a different distribution. Although not been well recognized, this type of problem is common in real world applications. For example, if one were to analyze data generated from a Brain Computer Interface, it is known that the distribution over incoming signals changes as experiments continue because of subject fatigue and the sensor setup changes and so on. Here we wish to obtain a model based on the earlier experimental data that is also able to make accurate predictions on signals obtained from the later period of the experiments, but obviously these two distributions are quite different. Another example comes from survey analysis. If, for example, one would like to collect some features of customers that are interested in a certain product. A survey form will be given to the customers for data collection. However, it is more likely that those people who are willing to fill the forms are those who are interested in the product. If we generate a model based on the profiles of these customers, it would be less accurate to make judgment on a random person in the whole general population. The problem is that the training set is collected in a biased manner and we do not notice the bias. Therefore a natural question

is whether it is possible to still learn a good model when a bias exists between training and test sets? In the second part of the thesis, I present an approach that accommodates the difference between the two distributions by using unlabeled data drawn from the test distribution. Thus, I present a semi-supervised learning method that is able to handle the above challenging scenario.

1.2 Thesis Contributions

The contributions of the thesis are outlined below:

- First, in the background chapter, I provide a common view of unsupervised and semi-supervised learning on undirected graphs. Various graph based unsupervised and semi-supervised algorithms have been separately proposed in literature recently, while little attention has been paid to the connection between the algorithms developed for these problems. In the background chapter, I demonstrate a novel unified perspective of unsupervised, semi-supervised and supervised learning on graphs, based on a common regularization framework. The chapter also discusses the differences between different regularizers in graph based semi-supervised learning to further validate the connection.

- Then, I propose new unsupervised and semi-supervised learning algorithms for directed graphs. Directed graphs are useful when the underlying data relationship is not symmetric. For example, if a Web page A has a hyperlink to Web page B, it does not necessarily mean that B also has a link pointing to A. I present experimental results on Web classification and protein function prediction which demonstrate that the proposed directed graph algorithms obtain much better performance than their undirected counterparts. Recalling the unified framework I develop for undirected graphs, I also develop a corresponding analysis for directed graphs that relates unsupervised, semi-supervised and supervised learning with this representation.

- The unsupervised algorithm on directed graphs is very useful for solving the Web community identification problem, a challenging problem in Web search. Noticing that the random walk model is a free parameter in learning on directed graphs, I

propose variations of the random walk models raised from different Web topologies and investigate their effects for finding Web communities. The analysis shows that the hyperlink structure of the Web provides very useful information for identifying Web communities, and that random walks are able to capture different relationships based on various hyperlink topologies. This analysis provides a practical characterization of distinct random walks for unsupervised learning on directed graphs.

- Next, I propose unsupervised and semi-supervised algorithms for hypergraphs. Hypergraphs are useful when the underlying data relationships are not naturally pairwise. The algorithms I develop here involve new extensions to the original undirected variants. This extension also has a useful random walk interpretation, as in the directed case. I present experiments in unsupervised and semi-supervised learning on various real world relational datasets to illustrate the advantage of preserving non-pairwise relationships.

- Furthermore, I develop a simple, unified mechanism for incorporating information from multiple object types and relations on a target subgraph—achieving a general approach to learning problems in heterogeneous networks that involve multiple object types and relations. I define a marginalized random walk that effectively propagates all sources of relevant information onto a target subgraph. I present experimental evaluations in challenging real world problems, including Web classification with both text and hyperlink information, and ranking in citation networks.

- Finally, I develop a new kernel method for solving sample selection bias in learning problems. The sampling bias, which arises from differences between the training and test distributions, usually causes significant inaccuracies in standard learning algorithms. I propose a method that uses a reweighting scheme to correct for the bias. The resampling weights are inferred directly by distribution matching between training and testing sets, where the matching is performed implicitly in a feature space. I show that the matching error and the estimated risk error are bounded in terms of the support of the distribution and the sample sizes. I also demonstrate empirically that the new method yields significant improvements over standard learning algorithms in the presence of distribution shifting.

1.3 Thesis Outline

Below is a summary of the thesis.

- **Chapter 2 Background: Learning with undirected graphs** I provide the relevant background on learning with undirected graphs, including unsupervised and semi-supervised learning algorithms. The chapter attempts to illustrate the key connection between graph based unsupervised and semi-supervised learning algorithms through a unified regularization interpretation. This unification requires some additional background on graph Laplacians, discrete analysis on undirected graphs, and regularization theory.

- **Chapter 3 Beyond symmetry: Learning with directed graphs** I show how directionality can be efficiently used in unsupervised and semi-supervised learning algorithms by representing data as a directed graph. The chapter also extends the unified regularization analysis to directed graphs.

- **Chapter 4 Beyond pairs: Learning with hypergraphs** This chapter presents unsupervised and semi-supervised learning algorithms based on hypergraphs.

- **Chapter 5 Beyond homogeneity: Learning with complex networks** This chapter extends the graph based learning framework to complex heterogeneous networks that involve multiple types of data objects and relations.

- **Chapter 6 Learning under distribution shifting with unlabeled data** I present the new de-biasing procedure I propose to reweight training data to account for distributional shifting between the training and test distributions. I show that this procedure can be used to modify many standard learning algorithms, by incorporating additional unlabeled test data, to achieve more accurate results in these circumstances.

- **Chapter 7 Conclusions** Finally, Chapter 7 concludes the thesis with a summary of the main contributions and a discussion of several directions for future research.

1.4 Publication Notes

Most of the work presented in this thesis has already been published. Some of the background material presented in Chapter 2 is based on a technical report published at MPI in 2005 (Huang, 2005). Material presented in Chapter 3 appeared in (Zhou et al., 2005a) and (Huang et al., 2006d). Material presented in Chapter 4 has been published in (Zhou et al., 2006); earlier it is published as a technical report at MPI when I was working there as a research intern. Material presented in Chapter 5 has been published in (Huang et al., 2006c). The work in Chapter 6 was completed when I was working at NICTA and has been published in (Huang et al., 2006a); earlier it is published as a technical report (Huang et al., 2006b) at University of Waterloo.

Chapter 2

Background: Learning with Undirected Graphs

Graphs are a very useful representation of data. Often, data relationships are naturally obtained from original connectivity information between individual data items. This chapter provides background on learning algorithms for problems where the input data is represented as a graph. In particular, in this setting, each vertex in a graph denotes an observation, and we wish to learn a function that assigns a label to each vertex. Beyond the basic connectivity information, I will also assume that we have some prior knowledge about the similarities between the vertices; given for example, by a Gaussian response to the Euclidean distances between the observations at each vertex. These similarities characterize the *local* connection strengths in the graph.

A number of machine learning methods for unsupervised and semi-supervised learning can be formulated in terms of optimizing a labeling function over vertices in a given graph. However, in current research, the algorithms for different types of learning problems—unsupervised versus semi-supervised learning—have been developed individually, with very few direct connections established between them. Here, in addition to reviewing current techniques in graph based unsupervised and semi-supervised learning, I demonstrate a novel unified understanding of the connections between unsupervised, semi-supervised and supervised learning on graphs. I begin with the observation that graph based unsupervised learning algorithms optimize a labeling function over vertices that minimizes some cut cost

objective; whereas semi-supervised and supervised learning algorithms minimize a combination of a loss between the function and a set of target labels with a regularization penalty. The connection is achieved by observing that the cut cost objective used by unsupervised learning algorithms can also serve as a valid regularizer for labeling functions defined on the graph vertices. The connection is further reinforced by a direct analogy between the functional analysis of the cut cost regularizer applied to functions on graphs, with the the functional analysis of standard regularizers applied to functions over continuous spaces. In this way, one can see that the cut cost regularizer forces the function values change "smoothly" over the graph. Once a regularizer has been properly formulated from an unsupervised learning objective on the graph, it is then straightforward to formulate semi-supervised learning principles from unsupervised criteria by combining the regularizer with an empirical loss of the observed data. Thus, we obtain a learning mechanism on graphs that is based on the same general learning principles as traditional supervised learning, while avoiding the need to invent new principles. Understanding the behavior of different cut cost objectives also explains the effectiveness of different graph based semi-supervised learning methods.

2.1 Preliminaries

Throughout this chapter, I assume data is represented in a simple undirected graph. An undirected graph $G = (V, E)$ consists of a set of vertices V and a set of edges between the vertices E. I denote by (u, v) the edge e that joins vertices u and v. Hence I refer to u and v as neighbors. A self-loop is an edge which starts and ends at the same vertex. The assumption that the graph is simple means that it has no self-loops and at most one edge connects between any two vertices. The assumption that the graph is undirected means that the each edge is a unordered pair of distinct vertices; i.e., $(u, v) \in E$ denotes the same edge as $(v, u) \in E$. Thus, I denote $u \sim v$ when u and v are neighbors.

An undirected graph is weighted if its edges are associated with a symmetric weight function $w : E \to \mathbb{R}^+$. Such graph is also called as *similarity graph*. The weight function indicates the similarity between vertices. For example, the weight function could be constructed via a k-nearest neighbor rule that models the local k nearest neighborhood

relationships; or alternatively, we can simply use fully connected graph where a Gaussian function is used to calculate similarity weights based on features associated with each vertex.

I will also make use of the following definitions. The degree function $d : V \to \mathbb{R}^+$ is defined as

$$d(u) = \sum_{v \sim u} w(u, v) \tag{2.1}$$

The volume of a set of vertices $S \subset G$ is defined as

$$\operatorname{vol} S = \sum_{u \in S} d(u) \tag{2.2}$$

The volume of the graph is therefore given by

$$\operatorname{vol} G = \sum_{v \in V} d(v) \tag{2.3}$$

In addition, the cardinality of S is defined as $|S|$.

2.2 Unsupervised Learning on Undirected Graphs

First, I review the classical problem in unsupervised learning—clustering—where one paritions the data in an attempt to uncover the intrinsic class structure. Clustering is a fundamental problem in machine learning that has been widely applied in many application areas, ranging from statistics, computer vision, and data mining, to biology and physics. When data is represented as a weighted graph, the goal of clustering is to partition a connected graph into homogeneous and well separated subsets such that the vertices within the same subset are similar and the vertices in different subsets are dissimilar. For a binary graph clustering, the problem is to find an integer assignment function $f : V \to \{-1, 1\}$ that achieves this intuitive criterion.

Mathematically, a binary vertex partition on a graph separates the graph into two disjoint vertex sets S and S^c, where S^c is the compliment of S, by removing edges connecting the two sets. This partition is denoted as $\Pi(S, S^c)$. In general, the sets $S_1, \ldots, S_k$ form a partition of the graph if $S_i \bigcap S_j = \emptyset$ and $S_1 \bigcup \cdots \bigcup S_k = V$. In this section I will consider work that exploits partitioning as a means to *cluster* the vertices in the graph.

Define the *out-boundary* , ∂S, of S to be $\partial S = \{(u, v)|u \in S, v \in S^c\}$ which is the cut set. The quantity $\mathrm{vol}\,\partial S$ is also referred to as the edge cut cost, $\mathrm{cut}(S, S^c)$. That is

$$\mathrm{cut}(S, S^c) = \sum_{u \in S, v \in S^c} w(u, v) = \mathrm{vol}\,\partial S \tag{2.4}$$

One intuitive way to compute a partition $\Pi(S, S^c)$ is to minimize the edge cut cost. By convention I will label

$$f(u) = \begin{cases} 1 & u \in S \\ -1 & u \in S^c \end{cases} \tag{2.5}$$

The problem of undirected graph clustering is well studied and the literature on the subject is very rich (Everitt, 1980). Binary graph clustering is primarily related to combinatorial problems that involve partitioning vertices in two equal subsets with the minimum number of edges cutting across the partition (Garey and Johnson, 1979). Although there is a simple polynomial time minimum cut algorithm for this problem, richer partition objectives require an exponential search time for finding the exact optimum. Many heuristics have been developed over the years for these richer criteria. Recently, spectral partitioning methods emerged as a particularly effective and principled approach that often outperform the traditional techniques, such as k-means or single linkage. Spectral graph clustering methods relax the combinatorial problem into a real valued problem that can be solved efficiently. Interestingly, these relaxed cut cost objectives can usually be expressed in terms of a graph Laplacian, which facilitates the analysis and derivation of the clustering techniques, and later, as we will see, provides a connection to continuous regularization theory. In the following, I review some typical spectral clustering methods on undirected graphs and compare some of their properties.

2.2.1 Minimum Cut

Consider the simple objective of minimizing $\mathrm{cut}(S, S^c)$ defined in (2.4), referred to as Minimum Cut. It is well-known that the dual of the minimum cut problem is the well known max flow problem (Cormen et al., 2001), which has a polynomial time solution via linear programming.

Even though there is a polynomial time solution, below I will develop an efficient spectral approximation to the exact algorithm. An earlier paper by (Pothen et al., 1990) has further discussions of this spectral approximation. Although the approximate solution is not useful in practice, I would like to explain the derivation here in order to make other spectral methods easier to understand. The spectral approximation for Minimum Cut is achieved by relaxing the integer constraint on the integer assignment function f. First note that the objective of minimizing $\text{cut}(S, S^c)$ can be expressed directly in terms of the unnormalized graph Laplacian (also referred to as the *combinatorial Laplacian*) L, which is defined as

$$L(u,v) = \begin{cases} d(u) \text{ if } u = v \\ -w(u,v) \text{ if } u \sim v \\ 0 \text{ otherwise} \end{cases}$$

It is convenient to view the Laplacian as a linear operator such that for any function $f : V \to \mathbb{R}$, we have

$$Lf(u) = \sum_{v \sim u} w(u,v)(f(u) - f(v))$$

In matrix form, L is represented as

$$L = D - W$$

where D is a diagonal matrix that $D(u,u) = d(u)$, and W is the matrix for the weight function where $W(u,v) = w(u,v)$. It is known that L is a symmetric positive semidefinite operator, so its eigenvalues are real and non-negative (Chung, 1997). Obviously, its first eigenvector is $e = (1, 1, ..., 1)$ with the eigenvalue of 0. Mohar (1991, 1997) has an overview of its many other properties.

We have the following proposition for the Minimum Cut objective.

Proposition 2.2.1. *Minimizing* $\text{cut}(S, S^c)$ *is equivalent to minimizing* $\frac{1}{4} f^T L f$*, where* f *is defined as in (2.5).*

Proof. Consider an edge $(u, v) \in E$. Note that for any proposed partition $\Pi(S, S^c)$, given the definition of f, then $(f(u) - f(v))^2 = 4$ if $u \in S$ and $v \in S^c$. But if u, v are both in S

or both in S^c then $(f(u) - f(v))^2 = 0$. Therefore, we obtain

$$\begin{aligned} \text{cut}(S, S^c) &= \frac{1}{4} \sum_{u \in S, v \in S^c} w(u,v)(f(u) - f(v))^2 \\ &= \frac{1}{8} \sum_{u \sim v} w(u,v)(f(u) - f(v))^2 \\ &= \frac{1}{4} f^T L f \end{aligned} \tag{2.6}$$

□

Thus, solving for the Minimum Cut is equivalent to solving for a $\{+1, -1\}$-valued function, f, on vertices that minimizes (2.6). If one were to drop the integer constraint on f, this becomes a straightforward convex quadratic optimization with a closed form solution: $Lf = 0$. This equation explicitly demonstrates that the solution f requires the harmonic property—the function value of vertex u is the weighted linear combination of the values of its neighbors. Interestingly, in the continuous case, this is exactly Laplace's equation, which is a partial differential equation that occurs in many fields of science, such as electromagnetism, astronomy and fluid dynamics. A function f that satisfies Laplace's equation is said to be harmonic. A solution to Laplace's equation has the property that the average value over a spherical surface is equal to the value at the center of the sphere. The minimization problem has no local maxima or minima because the solution equation is linear; any superposition of any two solutions is also a solution. In the discrete case, as in a graph, the equation has similar meaning and properties. I will explore a deeper connection between the graph Laplacian with the continuous standard Laplacian defined in terms of differential operators in Section 2.3.2.

Unfortunately, without any further constraints, the solution of the equation $Lf = 0$ is not uniquely determined so that the solution can still be undesirable. For example, there is nothing to prevent the solution from cutting off a single vertex that happens to have a small total weight in its connections to the rest of the graph (Shi and Malik, 2000). To overcome this problem, the objective should be modified to take into account the sizes of S and S^c in the partition, which leads to the next clustering method.

2.2.2 Ratio Cut

To improve the quality of the clustering results, it is natural to impose an additional constraint on partition size to ensure balance. However, this results in an NP-complete problem (Matula and Shahrokhi, 1990). Although many heuristics have been proposed to solve this problem, a significant advance was made by the Ratio Cut method, which incorporates partition balance in the cut cost criterion rather than imposing the constraints explicitly (Hagen and Kahng, 1992b). This allows one to achieve a convenient spectral approximation.

Ratio Cut attempts to minimize the cut cost while simultaneously balancing the cardinality of the partitions. The Ratio Cut criterion is defined as

$$\mathrm{Rcut}(S, S^c) = \frac{\mathrm{cut}(S, S^c)}{|S|} + \frac{\mathrm{cut}(S^c, S)}{|S^c|} \tag{2.7}$$

Proposition 2.2.2. *Let* $\alpha = |S|/|G|$. *Then*

$$\mathrm{Rcut}(S, S^c) \;=\; \frac{g^T L g}{g^T g}, \quad \textit{where} \quad g(u) = \begin{cases} 2(1-\alpha) & u \in S \\ -2\alpha & u \in S^c \end{cases} \tag{2.8}$$

Moreover, g *satisfies*

$$g^T e \;=\; 0 \tag{2.9}$$

where e *is a column vector with all elements equal* 1.

Proof. First to establish (2.8), note that from the definition of f given in (2.5) and from Proposition 2.2.1 we have

$$\begin{aligned} \mathrm{Rcut}(S, S^c) &= \left(\frac{1}{|S|} + \frac{1}{|S^c|}\right) \mathrm{cut}(S, S^c) \\ &= \left(\frac{1}{|S|} + \frac{1}{|S^c|}\right) \left(\frac{1}{8} \sum_{u \sim v} w(u,v)(f(u) - f(v))^2\right) \\ &= \frac{\sum_{u \sim v} w(u,v)(f(u) - f(v))^2}{8\alpha(1-\alpha)\sum_u f^2(u)} \end{aligned} \tag{2.10}$$

where the last step follows from the fact that $\frac{1}{|S|} + \frac{1}{|S^c|} = \frac{|G|}{|S||S^c|} = \frac{1}{\alpha(1-\alpha)|G|}$ and $|G| = \sum_u f^2(u)$. Now by the definition of g, we have that $f(u) - f(v) = g(u) - g(v)$, and hence

$$\sum_{u \sim v} w(u,v)(f(u) - f(v))^2 \;=\; \sum_{u \sim v} w(u,v)(g(u) - g(v))^2 \;=\; 2g^T L g \tag{2.11}$$

which establishes the numerator in (2.8). It remains only to derive the denominator. To do so, first notice that $g(u) = f(u) + (1 - 2\alpha)$, and therefore

$$\sum_u f(u) \;=\; \sum_{u \in S} f(u) + \sum_{u \in S^c} f(u) \;=\; (2\alpha - 1)|G|$$

Thus, we obtain

$$\begin{aligned} g^T g \;&=\; \sum_u g^2(u) \;=\; \sum_u (f(u) + (1 - 2\alpha))^2 \\ &=\; |G| + (1-2\alpha)^2|G| + 2(1-2\alpha)\sum_u f(u) \\ &=\; |G| + (1-2\alpha)^2|G| + 2(1-2\alpha)(2\alpha - 1)|G| \\ &=\; 4\alpha(1-\alpha)|G| \;=\; 4\alpha(1-\alpha)\sum_u f^2(u) \end{aligned} \tag{2.12}$$

Plugging (2.11) and (2.12) into (2.10) yields (2.8).

Finally, to show that the constraint (2.9) holds, note that

$$\sum_u g(u) \;=\; \sum_{u \in S} g(u) + \sum_{u \in S^c} g(u) \;=\; \alpha|G|(2 - 2\alpha) + (1-\alpha)|G|(-2\alpha) \;=\; 0$$

□

Therefore, an approximate solution to minimizing the Ratio Cut objective can be obtained by relaxing the discrete constraint on g. In this case, minimizing the objective (2.8), which is the Rayleigh Quotient, can be obtained by solving

$$\min g^T L g \quad \text{s.t.} \quad \| g \| = 1,\; g^T e = 0$$

Since there are boundary conditions, the Rayleigh-Ritz theorem (Lutkepohl, 1997) can be used to show that the second smallest eigenvector of L is the unique solution.

Although the Ratio Cut criterion is a significant improvement over Minimum Cut, in that it attempts to balance the size of the partitions, it still does not necessarily give reasonable results if the edge weights are not approximately uniform, which is common in reality.

2.2.3 Normalized Cut

A more sophisticated method has recently been proposed by Shi and Malik (2000). In this method, the volumes of the partitions are considered, not just their cardinalities, which takes into account the distribution of weights in the graph. The Normalized Cut cost objective is given by

$$\mathrm{Ncut}(S, S^c) = \frac{\mathrm{cut}(S, S^c)}{\mathrm{vol}\, S} + \frac{\mathrm{cut}(S^c, S)}{\mathrm{vol}\, S^c} \tag{2.13}$$

Computing the minimum Normalized cut is also a combinatorial optimization problem that is NP hard (Shi and Malik, 2000). However, the spectral approach, again, allows the problem be approximately solved by relaxing the integer contraints. The spectral solution involves another form of the graph Laplacian—the *normalized graph Laplacian* Δ which is defined as

$$\Delta(u, v) = \begin{cases} 1 - \frac{w(v,v)}{d(v)} \text{ if } u = v \\ -\frac{w(u,v)}{\sqrt{d(u)d(v)}} \text{ if } u \sim v \\ 0 \text{ otherwise} \end{cases} \tag{2.14}$$

As with the combinatorial Laplacian, L, the normalized Laplacian Δ can also be interpreted as a linear operator on vertex functions f

$$\Delta f(u) = \frac{1}{\sqrt{d(u)}} \sum_{v \sim u} w(u, v) \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right) \tag{2.15}$$

Note that if $d(u)$, defined in (2.1), is uniformly distributed, then $\Delta f(u)$ reduces to $Lf(u)$ up to a constant factor $\frac{1}{d(u)}$.

We can write the normalized Laplacian as a matrix

$$\Delta = I - D^{-1/2} W D^{-1/2} \tag{2.16}$$

In general, the following relation holds between L and Δ:

$$\Delta = D^{-1/2} L D^{-1/2}$$

Δ is also semidefinite. However, its first eigenvector is $D^{1/2}e$ which is not constant anymore. Chung (1997) introduces other properties for the normalized graph Laplacian.

Proposition 2.2.3. *Let* $\gamma = \operatorname{vol} S / \operatorname{vol} G$ *where* $\operatorname{vol} S$ *and* $\operatorname{vol} G$ *are defined as in (2.2) and (2.3), and* d *is defined as in (2.1). Then*

$$\operatorname{Ncut}(S, S^c) = \frac{h^T \Delta h}{h^T h} \tag{2.17}$$

where $h = \sqrt{d} \circ r^1$ *such that*

$$r(u) = \begin{cases} 2(1-\gamma) & u \in S \\ -2\gamma & u \in S^c \end{cases}$$

Moreover, h *satisfies*

$$h^T D^{-1/2} e = 0 \tag{2.18}$$

Proof. Similar to the proof of (2.10) in Proposition 2.2.2, the Normalized Cut cost criterion can be written as

$$\operatorname{Ncut}(S, S^c) = \frac{\sum_{u \sim v} w(u,v) \left(f(u) - f(v)\right)^2}{8\gamma(1-\gamma) \sum_{v \in V} f^2(v) d(v)} \tag{2.19}$$

Clearly for all $u, v \in V$,

$$\operatorname{sign}(r(v)) = \operatorname{sign}(f(v)) \text{and} r(u) - r(v) = f(u) - f(v) \tag{2.20}$$

Following a similar proof to (2.12) and (2.9), we have

$$4\gamma(1-\gamma) \sum_{v \in V} f^2(v) d(v) = 2 \sum_{v \in V} r^2(v) d(v) \tag{2.21}$$

and

$$\sum_{v \in V} r(v) d(v) = 0 \tag{2.22}$$

(2.22) shows the constraint (2.18) holds given $h = \sqrt{d} \circ r$. Therefore, combining (2.20) and

[1] Operator $\circ$ denotes componentwise multplication.

(2.21) into (2.19) we have

$$\begin{aligned} \mathrm{Ncut}(S, S^c) &= \frac{\sum_{u\sim v} w(u,v)\left(r(u)-r(v)\right)^2}{2\sum_{v\in V} r^2(v)d(v)} \\ &= \frac{\sum_{u\sim v} w(u,v)\left(\frac{h(u)}{\sqrt{d(u)}}-\frac{h(v)}{\sqrt{d(u)}}\right)^2}{2\sum_{v\in V} h^2(v)} \qquad (2.23) \\ &= \frac{h^T\Delta h}{h^T h} \qquad (2.24) \end{aligned}$$

□

As in the Ratio Cut case, an approximate solution to minimizing the Normalized Cut objective, can be obtained by relaxing the discrete constraint on h by solving

$$\min h^T\Delta h \quad \text{s.t.} \quad \| h \| = 1,\ h^T D^{-1/2} e = 0$$

The solution is the second smallest eigenvector of normalized Laplacian Δ, or equivalently the generalized eigenvector of $Lv = \lambda Dv$. This gives an efficient computational technique for finding an approximate solution by relaxing the discrete constraint on h. Another closely related work is by Ng et al. (2002), which directly finds the eigenvector with the second largest eigenvalue of $I - \Delta$, obtaining the same solution as above.

2.2.4 Random Walk Interpretation

A nice advantage of the normalized cut criterion in particular is that it can also be understood in terms of stationary random walks on the undirected graph. Random walks can be used to elegantly model how local information is naturally propagated over the entire graph. This interpretation is related to low conductivity sets in a Markov random walk (Meila and Shi, 2001). The random walk model is also related by analogy to a model of electrical flow in a network (Lovasz, 1996). Some basic properties of random walks on a graph (e.g. the mixing time that measures how fast the random walk reaches its stationary status) are determined by the spectrum of the graph. To demonstrate the interpretation, I will use some basic definitions in random walks.

A random walk is determined by the transition probability $P(u, v)$ indicating the probability of traversing from vertex u to v. Clearly for each vertex u

$$\sum_{u \sim v} P(u, v) = 1$$

A random walk is said to be *ergodic* if there is a unique stationary distribution π satisfying the following *balance equation*

$$\sum_{v \sim u} \pi(u) P(u, v) = \pi(v), \forall v \in V \tag{2.25}$$

A natural random walk on a weighted undirected graph can be defined by the transition probabilities

$$p(u, v) = \frac{w(u, v)}{d(u)} \tag{2.26}$$

where $d(u)$ is defined in (2.1). It is easy to verify that the stationary distribution of this particular random walk model on a undirected graph satisfies

$$\pi(u) = d(u) / \operatorname{vol} G$$

where $\operatorname{vol} G$ is defined in (2.3). The conductance of a set $S \subset V$ is defined as

$$\Phi(S) = \frac{\sum_{u \in S, v \in S^c} \pi(u) P(u, v)}{\pi(S)}$$

where $\pi(S) = \sum_u \pi(u)$.

A natural way to measure the quality of a partition $\Pi(S, S^c)$ is the frequency with which a stationary random walk goes from S to S^c in proportion to the frequency with which the walk remains in the set S. It can be shown that that minimizing the Normalized Cut in undirected graph spectral clustering is in fact equivalent to minimizing the following (Shi and Malik, 2000):

$$\begin{aligned} \operatorname{Ncut}(S, S^c) &= \Phi(S) + \Phi(S^c) \\ &= Pr[S \to S^c | S] + Pr[S^c \to S | S] \end{aligned}$$

Here, $Pr[S \to S^c | S]$ is the probability that the random walker goes from set S to set S^c in one step, given that the current state is in S and the random walk has reached its

stationary distribution. Intuitively, this makes sense that the normalized minimum cut cost corresponds to having the probability of jumping between different clusters be small while the probability of staying within the same cluster be large. This random walk interpretation makes the normalized spectral clustering method more special and interesting compared to other spectral methods. This interpretation will be exploited again in my research below, and generalized to other types of graphs.

2.2.5 k-way Spectral Partitioning

The spectral clustering methods discussed up to now focus on binary clustering. For the case of finding k clusters where $k > 2$, there are various approaches. For example, one can recursively perform binary clustering. Another approach is to modify the cut criteria to consider k clusters simultaneously. I briefly present the derivation for k-way Normalized Cut in the following. The method for Ratio Cut can be obtained in a similar way.

Let $\Pi = (V_i)_{i=1}^k$ be a k-way disjoint partition of V such that $V = \bigcup_{i=1}^k V_i$, where $V_i \bigcap V_j = \emptyset$ for all $1 \leq i, j \leq k, i \neq j$. The k-way Normalized Cut given by Gu et al. (2001) is

$$\text{Ncut}(\Pi) = \frac{W(V_1, V_1^c)}{W(V_1, V)} + \frac{W(V_2, V_2^c)}{W(V_2, V)} + \cdots + \frac{W(V_k, V_k^c)}{W(V_k, V)} \tag{2.27}$$

Let $x_i = [0, \ldots 0, 1 \ldots 1, 0, \ldots 0]^T$ be an indicator vector with respect to all other vertices such that 1 indicates two vertices belong to the same cluster. Then, (2.27) can be written as

$$\text{Ncut}(\Pi) = \frac{x_1^T(D-W)x_1}{x_1^T D x_1} + \frac{x_2^T(D-W)x_2}{x_2^T D x_2} + \cdots + \frac{x_k^T(D-W)x_k}{x_k^T D x_k} \tag{2.28}$$

Let $y_i = D^{1/2}x_i / \|D^{1/2}x_i\|^2$ and $Y_k = [y_1, y_2, \ldots y_k]$. Then $Y_k^T Y_k = I_k$. Relaxing the discreteness condition and substituting y_i to x_i in (2.28), we obtain the relaxed optimization problem

$$\min_{Y_k \in \mathbb{R}^{n \times k}} \text{Tr}(Y_k^T(I - \Delta)Y_k) \quad \text{s.t.} \quad Y_k^T Y_k = I_k$$

Again, this is a standard problem which can be solved by choosing the first k eigenvectors of Δ as the real valued solution.

To use the first k eigenvectors to obtain a discrete k-way partition, many heuristics have been proposed. Perhaps the most popular one among them is the following (Ng

et al., 2002): First form a matrix $X = [\Phi_1 \dots \Phi_k]$, consisting of the k smallest eigenvectors of Δ. Then each row vector in X is regarded as the representation of one of vertex in a k-dimensional Euclidean space. The vectors corresponding to vertices in distinct classes are generally expected to be well separated, and consequently we can obtain a good partition simply by running k-means on the rows of X.

A general practical issue for multi-class clustering problem is how to choose the number of clusters k. A variety of methods have been proposed for this problem. One is to use the eigengap heuristic that seeks a k such that the eigenvalues $\lambda_1, \dots \lambda_k$ are very small but $\lambda_k + 1$ is relatively large (Tibshirani et al., 2001). BenHur et al. (2002) and Lange et al. (2004) propose to apply the stability measurement to decide k. Interestingly, Ben-David et al. (2006b) argue that the stability is not a suitable criterion to determine k. So far, there is no justification of an optimal solution for this problem.

2.2.6 Comparison of Different Cut Criteria

It is worth noting that there is significant difference between the solutions of different combinatorial problems arising from the different cut criteria. For example, Ratio Cut only considers the cardinalities of the partition sets, whereas Normalized Cut considers the weight-volumes of the partition sets. Thus, Ratio Cut essentially makes the assumption that the node degree distribution over the graph is approximately uniform, which is not generally true in applications. This explains why Normalized Cut outperforms both Ratio Cut and Minimum Cut in most cases. The performance differences in clustering will have impact on the semi-supervised learning methods that I will discuss in Section 2.4. In addition, Normalized Cut is the only cut criterion that has the natural random walk interpretation I gave in Section 2.2.4, which implies more powerful interpretations and extensions that I will exploit in the next few chapters.

2.3 Supervised Learning on Undirected Graphs

One of my goals in this thesis is to demonstrate connections between different types of learning problems on graphs; specifically unsupervised, semi-supervised and supervised learning. To begin to establish these connections, I now consider a different learning

problem: supervised learning. In supervised learning, all the training data (i.e. vertices) are labeled $y(v) \in \{1, -1\}$, and we attempt to acquire a predictive model for unobserved test data. If we were to temporarily restrict attention to classification behavior just on the graph, intuitively, one could imagine balancing a tradeoff between minimizing a cut cost while also trying to minimize the loss with respect to the given vertex labels. In this way, it might be natural to consider learning a function by optimizing the combination

$$\min_f \sum_{v \in V} \text{loss}(f(v), y(v)) + \lambda \Omega(f) \tag{2.29}$$

where $\Omega(f) = \text{cut}(f)$ and λ is a parameter. Clearly, the role that the cut cost plays in (2.29) is that of a regularizer. For example, if one uses the Ratio Cut criterion, the regularizer would be

$$\Omega_R(f) = f^T L f$$

on the other hand, if one uses the Normalized Cut criterion, the regularizer would be

$$\Omega_N(f) = f^T \Delta f$$

(Note that the denominator $f^T f$ in the cut costs is not considered here.)

Equation (2.29) provides a tentative general principle for graph based supervised learning that raises many interesting questions: What is the relationship between a cut cost and a regularizer? Is the cut cost equivalent to a regularizer in terms of making the smoothness assumption? Can one combines a cut cost regularizer with traditional regularizers in supervised learning? To answer these questions more concretely, I need to first review classical regularization theory for supervised learning in continuous spaces.

2.3.1 Regularization over a Continuous Domain

In the usual supervised learning setting, we are given a set of input vectors $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^n$, and a corresponding set of target labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$. We seek a function $f : \mathcal{X} \to \mathcal{Y}$ such that given a new input x^* the inferred label $f(x^*)$ approximates the true label y^*. Referring to $\mathcal{X}$ and $\mathcal{Y}$ as the training data and the target labels respectively, note that it is easy to arbitrarily construct some function that fits the training data exactly. However, such an arbitrarily constructed function will predict poorly on unseen test data,

particularly if there was noise present in the training data. This effect is usually referred to as *overfitting*. To overcome the problem of overfitting when designing learning algorithms, one typically uses a regularizer that smooths the function in a way that hopefully estimates the true outputs more accurately for the given and unseen inputs. In other words, we impose a bias towards smoothness in an attempt to reduce generalization error.

Thus, one normally optimizes a combined objective in this case, just as in the graph case above

$$\min_f \sum_{i=1}^{n} \text{loss}\left(y_i, f(\mathbf{x}_i)\right) + \lambda\Omega(f) \tag{2.30}$$

where $\text{loss}(y_i, f(\mathbf{x}_i))$ is a loss function, Ω is a regularization operator on f, and λ is a tradeoff parameter. A difference from before is that here we assume the domain is $\mathbb{R}^n$, not a discrete vertex set, and that f is defined on all of $\mathbb{R}^n$. A key part of understanding (2.30) is understanding the regularization operator Ω on f. The study of regularizers originally arose from research on multivariate function estimation in statistics, such as polynomial spline analysis. Regularization theory provides a framework for restoring well-posedness by adding an appropriate constraint on the solution in (2.30) (Tikhonov and Arsenin, 1977; Morozov, 1984; Wahba, 1979). The choice of regularizer is understood as looking for a *smoothness functional* that (hopefully) facilitates generalization to future inputs. For example, a classical regularization functional is

$$\Omega(f) = \int \|Df\|^2 dx \tag{2.31}$$

where D is a linear differential operator, e.g., $D = \frac{\partial}{\partial x}$ or $D = \frac{\partial^2}{\partial^2 x}$ (or even higher order). The smoothness prior implicated in D makes the solution stable and insensitive to noise. Intuitively, the higher the order of the derivative we consider, the greater the prior preference will be for function smoothness.

An important special case is to choose $D = \frac{\partial}{\partial x}$, which gives the regularization functional

$$\Omega(f) = \int \|Df\|^2 dx = \|\Delta^{1/2} f\|^2 = f^T \Delta f \tag{2.32}$$

where Δ is the Laplacian operator. The Laplacian operator is a second order differential operator, defined as the divergence (div) of the gradient (∇) as

$$\Delta f = \text{div}(\nabla f)$$

This definition can be applied to functions defined on $\mathbb{R}^n$, but can even be extended to functions defined on a Riemannian manifold. The divergence operator on a manifold is adjoint to the gradient operator that is implied by the classical Stoke's theorem

$$\int_M \langle \nabla f, g \rangle = - \int_M (\operatorname{div} g) f$$

Below I demonstrate that the graph based Normalized Laplacian that was derived above can be interpreted as a direct analogue of this classical continuous Laplacian. In particular, I will establish this connection by showing that the important properties of the Laplacian still hold.

2.3.2 Regularization over a Discrete Domain

To show that the cut cost obtained from spectral clustering is really a regularizer in the same sense in a continuous space, I will show the normalized cut cost can be viewed as a discrete form of the Laplacian differential operator.[2]

Let $\mathcal{H}(V)$ denote the space of functions, in which $f : V \to \mathbb{R}$ assigns a real value $f(u)$ to each vertex u. A function in $\mathcal{H}(V)$ can be thought of as a column vector in $\mathbb{R}^{|V|}$, where $|V|$ is the number of vertices in V. The function space then can be endowed with the standard inner product in $\mathbb{R}^{|V|}$ as

$$\langle f, g \rangle_{\mathcal{H}(V)} = \sum_{u \in V} f(u) g(u)$$

for any two functions f and g in $\mathcal{H}(V)$. Similarly define $\mathcal{H}(E)$ consisting of the real-valued functions on edges.

Remember that, regularization is achieved by enforcing smoothness in f via (2.31) or via the more particular form (2.32). Such a smoothness objective has the same form as the

[2]Recently a family of analogous "differential operators" have been developed that connects to those in continuous spaces studied in differential geometry (Zhou and Schölkopf, 2005). The paper attempts to study the regularizer in a semi-supervised learning method(Zhou et al., 2004). Here I further explain why the cut cost can be understood as a regularizer and present this unification of unsupervised, supervised and semi-supervised learning on graphs in a way that facilitates solving problems on more complicated graphs.

spectral graph clustering objectives, e.g. $\langle f, \Delta f \rangle$, thus indicating that the cut cost may be interpretable as penalizing the second order "derivatives".

For continuous spaces, the divergence of the gradient of a scalar valued function is the Laplacian. Therefore, we would like to explore if the graph Laplacian operator also satisfies this definition as stated in the following definitions and theorem (Zhou and Schölkopf, 2005).

Definition 2.3.1. *Let the graph gradient on an undirected graph be an operator* $\nabla : \mathcal{H}(V) \to \mathcal{H}(E)$ *defined by*

$$(\nabla f)(u, v) = \sqrt{\frac{w(u, v)}{d(v)}} f(v) - \sqrt{\frac{w(u, v)}{d(u)}} f(u), \text{ for all } (u, v) \in E \tag{2.33}$$

In the definition, clearly, the gradient measures the change of a function on each edge. Moreover

$$(\nabla f)(u, v) = -(\nabla f)(v, u)$$

Definition 2.3.2. *Let the graph divergence* $\operatorname{div} : \mathcal{H}(E) \to \mathcal{H}(V)$ *be an operator defined by*

$$(\operatorname{div} g)(v) = \sum_{u \sim v} \sqrt{\frac{w(u, v)}{d(v)}} \left(g(v, u) - g(u, v)\right) \tag{2.34}$$

Then we have the following proposition.

Proposition 2.3.3. *The graph divergence* div *satisifies*

$$\langle \nabla f, g \rangle = \langle f, -\operatorname{div} g \rangle \text{ for all } f \in \mathcal{H}(V) \text{ and } g \in \mathcal{H}(E) \tag{2.35}$$

The proof is obtained simply by applying the definitions of the gradient and divergence to both sides of (2.35) which quickly yields the result.

The discrete divergence operator can be thought of discrete analogue of the classical Stoke's theorem that $\int_M \langle \nabla f, g \rangle = -\int_M (\operatorname{div} g) f$. Intuitively, the divergence measures the net outflow of function f at each vertex.

Definition 2.3.4. *Let the normalized graph Laplacian operator* $\Delta : \mathcal{H}(V) \to \mathcal{H}(V)$ *on an undirected graph defined by*

$$(\Delta f)(v) = f(v) - \sum_{u \sim v} \frac{w(u, v)}{\sqrt{g(u) g(v)}} f(u)$$

Theorem 2.3.5. *The normalized graph Laplacian on an undirected graph satisfies*

$$\Delta f = -\frac{1}{2}\operatorname{div}(\nabla f)$$

Proof. Substituting Equation (2.33) and (2.34) into (2.32), we have

$$\begin{aligned}(\Delta f)(v) &= \frac{1}{2}\sum_{u\sim v}\sqrt{\frac{w(u,v)}{g(v)}}\left((\nabla f)(u,v) - (\nabla f)(v,u)\right)\\ &= \sum_{u\sim v}\sqrt{\frac{w(u,v)}{g(v)}}\left(\sqrt{\frac{w(u,v)}{g(v)}}f(v) - \sqrt{\frac{w(u,v)}{g(u)}}f(u)\right)\\ &= f(v) - \sum_{u\sim v}\frac{w(u,v)}{\sqrt{g(u)g(v)}}f(u)\end{aligned}$$

It is not hard to see that in matrix notation, Δ can be written as $\Delta = I - D^{-1/2}WD^{-1/2}$, which is just the normalized Laplace matrix in Equation (2.16).[3] □

Thus the theorem shows the graph Laplacian is a discrete version of a classical differential operator.

Further justification for the analogy is given by considering the *spectrum* of the operator Δ, which is defined by its eigenvalues $0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{n-1}$. To analyze the spectrum, first note that the Rayleigh quotient of Δ associated with an arbitrary function $f \in \mathcal{H}(V)$ is given by Chung (1997),

$$\frac{\langle f, \Delta f\rangle}{\langle f, f\rangle} = \frac{\sum_{u\sim v}(g(u) - g(v))^2 w(u,v)}{\sum_v g(v)^2 d(v)}$$

where $g(u) = d(u)^{-1/2}f(u)$. Then, letting e be the constant 1 function, note that $e_0(u) = d(u)^{-1/2}e$ is an eigenfunction of Δ with eigenvalue 0. This immediately yields the result

[3] Similarily, we can also derive the combinatorial Laplacian by defining the gradient as

$$(\nabla f)(u,v) = \sqrt{w(u,v)}f(v) - \sqrt{w(u,v)}f(u), \text{ for all } (u,v) \in E$$

and the divergence as

$$(\operatorname{div} g)(v) = \sum_{u\sim v}\sqrt{w(u,v)}\,(g(v,u) - g(u,v))$$

that

$$\lambda_1 = \min_{g:\sum g(u)d(u)=0} \frac{\sum_{u\sim v}(g(u)-g(v))^2 w(u,v)}{\sum_v g(v)^2 d(v)} \tag{2.36}$$

This shows that the definitions leading to the normalized graph Laplacian are not arbitrary, since (2.36) closely corresponds to the first nontrivial eigenfunction of the *Laplacian-Beltrami* operator on functions defined over a Riemannian manifold.[4] The investigation of the connection between continuous Laplacian operator and the graph Laplacians has also been studied by Belkin and Niyogi (2005); Hein et al. (2005); Gine and Koltchinskii (2005) and Hein (2006).

2.4 Semi-supervised Learning on Undirected Graphs

Interestingly, this connection between unsupervised cut criteria and supervised regularizations yields its greatest benefits when applied to semi-supervised learning. Semi-supervised learning attempts to learn a predictive model given a combination of both labeled and unlabeled data. Formally, the problem is defined as given a set of data $X = \{x_1, x_2, \ldots, x_l, x_{l+1}, \ldots, x_n\}$ on domain $\mathcal{X}$ where the first l are labeled $Y = \{y_1, y_2, \ldots, y_l\}$ and the rest data are not labeled. There are two common variants of this problem: out of sample and transduction. In the out of sample problem, one attempts to generate a model that can be used for out-of-sample predictions. The transductive problem typically considers a discrete assignment function that only labels the given set of data without creating a model.

Either way, the intuition in semi-supervised learning is the same as graph based supervised learning. The goal is to minimize

$$\min_f \sum_{i=1}^{l} \text{loss}(f(x_i), y_i) + \lambda \, \text{cut}(f) \tag{2.37}$$

[4]For functions defined over a Riemannian manifold M, the eigenvalue of the first nontrivial eigenfunction, f, of the *Laplacian-Beltrami* operator satisfies

$$\lambda_1 = \inf_{f:\int_M f=0} \frac{\int_M |\nabla f|^2}{\int_M |f|^2}$$

where cut(f) is one of the natural graph cut criteria discussed above. The combined objective encourages the learner to minimize the cut cost over both labeled and unlabeled data, while trying to preserve the original class labels on labeled data.

Section 2.3.2 demonstrated that the cut cost associated with the graph Laplacian is a valid regularizer defined on functions over graphs. Therefore, a general objective for semi-supervised learning can be rewritten as

$$\min_f \sum_{i=1}^{l} \text{loss}(f(x_i), y_i) + \lambda\Omega(f) \tag{2.38}$$

The difference between this objective and the graph based supervised discussed earlier is that here the regularizer is applied to the function on both labeled and unlabeled data.

Not surprisingly, objective (2.38) serves as a general principle in many graph based semi-supervised learning algorithms. Here I list some of them.

- Zhou et al. (2004) proposes a framework for learning from labeled and unlabeled data on graphs by using the Normalized Cut objective as the regularizer. The framework solves the optimization problem

  $$f^* = \arg\min_{f \in \mathcal{H}(V)} \mu\|f - y\|^2 + \Omega_N(f) \tag{2.39}$$

  where the regularizer $\Omega_N(f)$ is derived from Normalized Cut objective. Initially, $f_i = y_i$ for all labeled data and $f_i = 0$ for all unlabeled data. Conveniently, (2.39) has a closed form solution

  $$f^* = (1 - \alpha)(I - \alpha S)^{-1} y$$

  where $\alpha = 1/(1 + \mu)$ and $S = D^{-1/2} W D^{-1/2} = I - \Delta$.

- The method in (Zhu et al., 2003) is equivalent to using the Ratio Cut objective as the regularizer in (2.38) with a hard labeling constraint—the labels on the labeled vertices are all fixed to $f = y$. Both (Zhou et al., 2004) and (Zhu et al., 2003) propose transductive learning algorithms over graphs. Interestingly, they both have interesting random walk interpretations, which I include in the thesis Appendix. Notably, I present an interpretation for (Zhu et al., 2003) that has not been explicitly outlined in literature before.

- Belkin and Niyogi (2004) propose to solve the out of sample problem by combining the regularizer derived from Ratio Cut with a classical regularizer in supervised learning. The objective is

$$f^* = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{l} \text{loss}(x_i, y_i, f) + \gamma_A \|f\|_{\mathcal{H}}^2 + \gamma_I \Omega_R(f)$$

  where $\|f\|_{\mathcal{H}}$ denotes the norm of f in a reporducing kernel Hilbert space. The solution for the optimization problem can be derived by the Representer theorem. The discrete analysis in Section 2.3.2 motivates why the Ratio Cut cost can be used as a regularizer and therefore can be properly combined with other regularizers.

- Another related algorithm for out of sample prediction is the transductive SVM algorithm of Vapnik (1998), which involves finding a separating hyperplane for a labeled data set that is also maximally distant from a given set of unlabeled test points. The problem can be written as a optimization problem as,

$$\min_{\mathbf{w}} C \sum_{i=1}^{l} |1 - \big(y_i(\mathbf{w}\mathbf{x}_i + b)\big)|_+ + C^* \sum_{i=l+1}^{n} |1 - (\mathbf{w}\mathbf{x}_i + b)|_+ + \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

  The drawback is that the objective is non-convex and thus it is difficult to minimize.

A broader literature review in semi-supervised learning can be found in (Zhu, 2006). Though there are various methods, many of them follow the general principle (2.38). I will also utilize this framework to solve more involved graph based learning problems in the next few chapters.

2.4.1 Empirical Comparison of Regularizers

Section 2.2 introduces three distinct cut cost objectives which we have seen can all be interpreted as regularizers. A natural question is: Which of these alternative regularizers gives better semi-supervised learning performance in practice?

To illustrate the performance differences when using the different regularizers, I consider a semi-supervised classification task using the USPS handwritten 16×16 digits dataset,

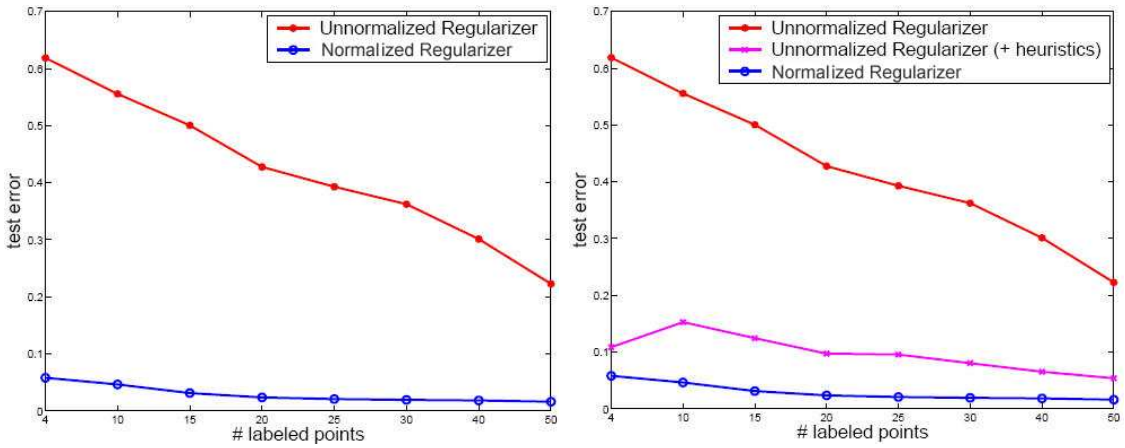


Figure 2.1: test errors in classification using different cut costs as regularizers

providing both labeled and unlabeled data to the learning algorithm (2.39) but with different regularizations that are from Normalized Cut and Ratio Cut respectively. I use the digits $1, 2, 3$ and 4 as the four classes in the experiment. There are 1269, 929, 824 and 852 examples for each class respectively, for a total of 3874. I construct a fully connected graph by using a RBF kernel where the width is set to 1.25. I sequentially increase the number of labeled points. The test errors are averaged over 100 trials and are summerized in Figure 2.1–left. It is clear that the method using Normalized Cut regularization significantly improved the accuracy over Ratio Cut (which I refer to as unnormalized regularizer). The right side of Figure 2.1 shows the performance of these two methods in comparison to a heuristic approach used in (Zhu et al., 2003) that approximates class proportions as prior knowledge. The method using Normalized Cut regularization still exhibits the best performance.

The reason for the performance difference is that the Normalized Cut objective is typically better than Ratio Cut in unsupervised learning, and this advantage carries over to the semi-supervised case. The methods proposed in (Zhu et al., 2003) and (Belkin and Niyogi, 2004) amount to adding label constraints to the Ratio Cut objective, which explains why these methods require additional heuristics to maintain the class proportions. The experiment shows that the selection of unsupervised cut criteria has significant influence on the performance of the resulting semi-supervised learning algorithms.

2.5 Summary

This background provides reviews for graph based unsupervised, supervised learning and semi-supervised learning. I have demonstrated a unified connection for unsupervised, semi-supervised and supervised learning on undirected graphs. This connection was established by noticing that the cut costs derived from unsupervised learning are equivalent to regularizers that impose smoothness assumptions in semi-supervised learning. Different learning algorithms on graphs can be unified in the same regularization framework.

I would like to note that a further generalization of regularization operators can be obtained via reproducing kernel Hilbert space(RKHS). For example, the radial basis function (RBF) kernel can be shown to correspond to regularization based on higher order differential operators (see thesis Appendix for a brief discussion). In Chapter 7, I discuss a future work the possibility that regularization based on such higher order differential operators might also be possible over graphs.

An obvious limitation of the algorithms so far is that they only work for undirected graphs. However, undirected graphs are not effective at capturing every type of data relationship. This leads to my next few chapters that solve learning problems where the relationship between data items follows a much more complex arrangement.

Chapter 3

Beyond Symmetry: Learning with Directed Graphs

In this chapter, I investigate the question of how to exploit the directionality and connectivity structure of a directed graph, rather than just exploit symmetric relationships encoded in an undirected graph in unsupervised and semi-supervised learning.

Many types of relationships between data can be better modelled by directed as opposed to undirected edges; for example, consider the hyperlink structure of the World Wide Web (Figure 3.1), the citation and reference links in bibliographic data, and the relationships in social networks. In these domains directionality is important and encodes useful information that is not adequately captured by undirected edges. In particular, there has been a significant research on exploiting the link structure of the Web for many purposes, including ranking Web pages, detecting Web communities, finding Web pages similar to a given web page, and finding Web pages of interest to a given geographical region; see Henzinger (2001) for a comprehensive survey. Unfortunately, few of the previous research efforts have worked directly with directed graphs; instead they have alternatively resorted to transforming a directed graph to an undirected one and applied undirected methods like those discussed in the previous chapter. However, by doing so, important information is lost—the asymmetric relationship encoded by edge direction. The shortcoming in current research raises the question of how to explore the directionality information naturally encoded in directed graphs to achieve better results.

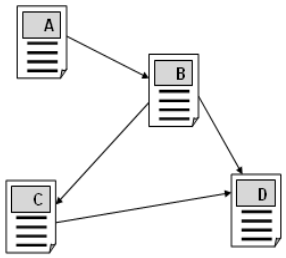


Figure 3.1: The World Wide Web can be considered as a directed graph, where vertices correspond to web pages and directed edges represent hyperlinks between them.

The major contribution of this chapter is to propose new unsupervised and semi-supervised learning algorithms for directed graphs. The unsupervised learning approach will be based on a generalization of undirected spectral clustering to directed graphs. Interestingly, the question of how eigenvectors partition a directed graph has been listed as one of six algorithmic challenges in Web search engines by Henzinger (2003). I have made progress on this question, and developed effective new unsupervised and semi-supervised learning algorithms for directed graphs as a result. I apply these new algorithms to classification and clustering problems on different types of networks, including Web and protein interaction networks. In particular, I study the problem of Web community identification, investigating how the *directed* hyperlink information conveyed via random walks can help one efficiently identify latent Web communities from the hyperlink topology alone. The analysis contributes a practical characterization of distinct random walks for unsupervised learning on the Web.

3.1 Preliminaries

A directed graph $G = (V, E)$ consists of a finite set of vertices V, together with a subset of directed edges E. An edge of a directed graph is an ordered pair (u, v) where u and v are the vertices of the graph. Given an edge (u, v), I say that the vertex v is *adjacent from* the

vertex u, and the the vertex u is *adjacent to* the vertex v, and the edge (u, v) is *incident from* the vertex u and *incident to* the vertex v.

A *path* in a directed graph is a tuple of vertices $(v_1, v_2, \ldots, v_p)$ with the property that $(v_i, v_{i+1}) \in E$ for $1 \leq i \leq p-1$. A directed graph is *strongly connected* if for every pair of vertices u and v there is a path in which $v_1 = u$ and $v_p = v$. For a strongly connected graph, there is an integer $k \geq 1$ and a unique partition $V = V_0 \cup V_1 \cup \cdots \cup V_{k-1}$ such that for all $0 \leq r \leq k-1$ each edge $(u, v) \in E$ with $u \in V_r$ has $v \in V_{r+1}$, where $V_k = V_0$, and k is maximal, that is, there is no other such partition $V = V_0^{'} \cup \cdots \cup V_{k^{'}-1}^{'}$ with $k^{'} > k$. When $k = 1$, we say that the graph is *aperiodic*; otherwise we say that the graph is *periodic*.

A *weighted* directed graph incorporates an associated weight function $w : E \to \mathbb{R}^+$ that assigns a weight to each edge. Given a weighted directed graph and a vertex v of this graph, the *in-degree function* $d^- : V \to \mathbb{R}^+$ and *out-degree function* $d^+ : V \to \mathbb{R}^+$ are defined by[1]

$$d^-(v) := \sum_{v \leftarrow u} w(u, v) \tag{3.1}$$

and

$$d^+(v) := \sum_{v \rightarrow u} w(v, u) \tag{3.2}$$

where $u \to v$ denotes the set of vertices adjacent to the vertex v, and $u \leftarrow v$ the set of vertices adjacent from the vertex v.

3.2 Unsupervised Learning on Directed Graphs

To solve unsupervised learning on directed graphs, I will need to give the following definitions.

Given weighted directed graph, there is a natural random walk on the graph with the transition probability function $p : V \times V \to \mathbb{R}^+$ defined by $p(u, v) = w(u, v)/d^+(u)$ for all $(u, v) \in E$, and 0 otherwise. The random walk on a strongly connected and aperiodic directed graph has a unique *stationary distribution* π; i.e. a unique probability distribution satisfying the balance equation $\pi(v) = \sum_{u \to v} \pi(u) p(u, v)$, for all $v \in V$. Moreover, $\pi(v) > 0$ for all $v \in V$.

[1]Note that $\sum_{u \to v}$ is the same as $\sum_{u : u \to v}$, where I omit 'u :' for short.

Given a subset S of the vertices from a directed graph G, define the volume of S to be

$$\operatorname{vol} S = \sum_{v \in S} \pi(v) \tag{3.3}$$

Clearly, $\operatorname{vol} S$ is the probability that the random walk occupies some vertex in S and consequently $\operatorname{vol} G = 1$.

Define the volume of the out-boundary of the subset S to be

$$\operatorname{vol} \partial S = \sum_{\partial S} \pi(u) p(u, v) \tag{3.4}$$

Note that $\operatorname{vol} \partial S$ is the probability with which one sees a transition from the subset S to its complement S^c. The definition clearly takes the directionality into account by considering one step move after the walk in the graph has reached stationary status.

3.2.1 Normalized Cut on Directed Graphs

Now we may partition a directed graph into two parts S and S^c by minimizing

$$\operatorname{Ncut}(S, S^c) = \left(\frac{\operatorname{vol} \partial S}{\operatorname{vol} S} + \frac{\operatorname{vol} \partial S^c}{\operatorname{vol} S^c} \right) \tag{3.5}$$

which is a directed generalization of the Normalized Cut criterion for undirected graphs in Section 2.2.3.

The key observation for generalizing the Normalized Cut criterion for directed graphs is

Proposition 3.2.1. $\operatorname{vol} \partial S = \operatorname{vol} \partial S^c$.

Proof. Obviously the probability of a transition from the subset S to its complement S^c must be compensated by the probability of an opposite transition. Formally, for each vertex v in V, it is easy to see that

$$\sum_{u \leftarrow v} \pi(v) p(v, u) = \pi(v) \sum_{u \leftarrow v} p(v, u) = \pi(v)$$

and

$$\sum_{u \rightarrow v} \pi(u) p(u, v) = \pi(v)$$

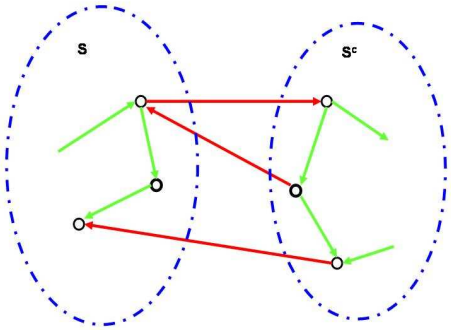


Figure 3.2: A subset S and its complement S^c. Note that there is only one edge in the out-boundary of S.

Therefore,

$$\sum_{u \to v} \pi(u)p(u,v) - \sum_{u \leftarrow v} \pi(v)p(v,u) = 0 \tag{3.6}$$

This is consistent with the law of flow conservation in electrical networks stating that the amount of flow entering a vertex equals the amount of flow that leaves the vertex.

Summing (3.6) over the vertices of S (see also Figure 3.2) then

$$\begin{aligned} &\sum_{v \in S} \left(\sum_{u \to v} \pi(u)p(u,v) - \sum_{u \leftarrow v} \pi(v)p(v,u) \right) \\ = &\sum_{(u,v) \in \partial S^c} \pi(u)p(u,v) - \sum_{(u,v) \in \partial S} \pi(u)p(u,v) = 0, \end{aligned}$$

which completes the proof. □

Therefore, The directed normalized cut criterion (3.5) can be further written as

$$\mathrm{Ncut}(S, S^c) = \mathrm{vol}\, \partial S \left(\frac{1}{\mathrm{vol}\, S} + \frac{1}{\mathrm{vol}\, S^c} \right)$$

I next consider a spectral relaxation of the Normalized Cut criterion that permits an efficient algorithmic approach.

Proposition 3.2.2. *Let ν denote* $\operatorname{vol} S$ *as defined in (3.3). Then*

$$\operatorname{Ncut}(S, S^c) = \frac{\sum_{(u,v)\in E} \pi(u)p(u,v)\left(\frac{f(u)}{\sqrt{\pi(u)}} - \frac{f(v)}{\sqrt{\pi(v)}}\right)^2}{2\langle f, f\rangle} \tag{3.7}$$

where $f = \sqrt{\pi} \circ g$ such that

$$g(u) = \begin{cases} 2(1-\nu) & u \in S \\ -2\nu & u \in S^c \end{cases}$$

Moreover,

$$\sum_{v\in V} \sqrt{\pi(v)} f(v) = 0 \tag{3.8}$$

Proof. Define an indicator function $h \in \mathbb{R}^{|V|}$ by $h(v) = 1$ if $v \in S$, and -1 if $v \in S^c$. Clearly, we have $0 < \nu < 1$ due to $S \subset G$. Then (3.5) may be written

$$\operatorname{Ncut}(S, S^c) = \frac{\sum_{(u,v)\in E} \pi(u)p(u,v)(h(u) - h(v))^2}{8\nu(1-\nu)}$$

Based on the definition of g, clearly, $\operatorname{sign} g(v) = \operatorname{sign} h(v)$ for all $v \in V$ and $h(u) - h(v) = g(u) - g(v)$ for all $u, v \in V$. Moreover, similar to the proof in Proposition 2.2.3, it is not hard to see that

$$\sum_{v\in V} \pi(v) g(v) = 0$$

which shows the constraint (3.8) holds given $f = \sqrt{\pi} \circ g$. Also we have

$$\sum_{v\in V} \pi(v) g^2(v) = 4\nu(1-\nu)$$

Therefore

$$\operatorname{Ncut}(S, S^c) = \frac{\sum_{(u,v)\in E} \pi(u)p(u,v)(g(u) - g(v))^2}{2\sum_{v\in V} \pi(v) g^2(v)}$$

The above equation may be further transformed into (3.7) when $f = \sqrt{\pi} \circ g$. □

Efficient computation Define the numerator of $\mathrm{Ncut}(S)$ as $\Omega(f)$

$$\Omega(f) = \frac{1}{2}\sum_{(u,v)\in E} \pi(u)p(u,v)\left(\frac{f(u)}{\sqrt{\pi(u)}} - \frac{f(v)}{\sqrt{\pi(v)}}\right)^2$$

For solving the minimization of the directed Normalized Cut objective, I introduce an operator $\Theta : \mathbb{R}^{|V|} \to \mathbb{R}^{|V|}$ defined by

$$(\Theta f)(v) = \frac{1}{2}\left(\sum_{u\to v} \frac{\pi(u)p(u,v)f(u)}{\sqrt{\pi(u)\pi(v)}} + \sum_{u\leftarrow v} \frac{\pi(v)p(v,u)f(u)}{\sqrt{\pi(v)\pi(u)}}\right) \tag{3.9}$$

Let Π denote the diagonal matrix with $\Pi(v,v) = \pi(v)$ for all $v \in V$; let P denote the matrix with $P(u,v) = p(u,v)$ if $(u,v) \in E$ and 0 otherwise; and let P^T denote the transpose of P. Then the operator Θ may then be written in matrix form as

$$\Theta = \frac{\Pi^{1/2}P\Pi^{-1/2} + \Pi^{-1/2}P^T\Pi^{1/2}}{2} \tag{3.10}$$

The following lemma allow us to rewrite $\Omega(f)$ in terms of a inner product that facilitates minimizing (3.7)

Lemma 3.2.3. *Let I denote the identity matrix. Then*

$$\Omega(f) = 2\langle f, (I-\Theta)f\rangle$$

Proof. The idea is to use summation by parts, a discrete analogue of the more common integration by parts.

$$\begin{aligned}
&\sum_{(u,v)\in E} \pi(u)p(u,v)\left(\frac{f(u)}{\sqrt{\pi(u)}} - \frac{f(v)}{\sqrt{\pi(v)}}\right)^2 \\
&= \frac{1}{2}\sum_{v\in V}\left\{\sum_{u\to v} \pi(u)p(u,v)\left(\frac{f(u)}{\sqrt{\pi(u)}} - \frac{f(v)}{\sqrt{\pi(v)}}\right)^2 + \sum_{u\leftarrow v} \pi(v)p(v,u)\left(\frac{f(v)}{\sqrt{\pi(v)}} - \frac{f(u)}{\sqrt{\pi(u)}}\right)^2\right\} \\
&= \frac{1}{2}\sum_{v\in V}\left\{\sum_{u\to v} p(u,v)f^2(u) + \sum_{u\to v}\frac{\pi(u)p(u,v)}{\pi(v)}f^2(v) - 2\sum_{u\to v}\frac{\pi(u)p(u,v)f(u)f(v)}{\sqrt{\pi(u)\pi(v)}}\right\} \\
&\quad + \frac{1}{2}\sum_{v\in V}\left\{\sum_{u\leftarrow v} p(v,u)f^2(v) + \sum_{u\leftarrow v}\frac{\pi(v)p(v,u)}{\pi(u)}f^2(u) - 2\sum_{u\leftarrow v}\frac{\pi(v)p(v,u)f(v)f(u)}{\sqrt{\pi(v)\pi(u)}}\right\}
\end{aligned}$$

The first term on the right-hand side may be written

$$\sum_{(u,v)\in E} p(u,v)f^2(u) = \sum_{u\in V}\sum_{v\leftarrow u} p(u,v)f^2(u)$$
$$= \sum_{u\in V}\left(\sum_{v\leftarrow u} p(u,v)\right) f^2(u) = \sum_{u\in V} f^2(u) = \sum_{v\in V} f^2(v)$$

and the second term

$$\sum_{v\in V}\left(\sum_{u\rightarrow v} \frac{\pi(u)p(u,v)}{\pi(v)}\right) f^2(v) = \sum_{v\in V} f^2(v)$$

Similarly, for the fourth and fifth terms, one can show that

$$\sum_{v\in V}\sum_{u\leftarrow v} p(v,u)f^2(v) = \sum_{v\in V} f^2(v)$$

and

$$\sum_{v\in V}\sum_{u\leftarrow v} \frac{\pi(v)p(v,u)}{\pi(u)} f^2(u) = \sum_{v\in V} f^2(v)$$

respectively. Therefore

$$\Omega(f) = \sum_{v\in V}\left\{ f^2(v) - \frac{1}{2}\left(\sum_{u\rightarrow v} \frac{\pi(u)p(u,v)f(u)f(v)}{\sqrt{\pi(u)\pi(v)}} + \sum_{u\leftarrow v} \frac{\pi(v)p(v,u)f(v)f(u)}{\sqrt{\pi(v)\pi(u)}}\right)\right\}$$

which completes the proof. □

Lemma 3.2.4. *The eigenvalues of the operator* Θ *are in* $[-1,1]$*, and the eigenvector with the eigenvalue equal to 1 is* $\sqrt{\pi}$*.*

Proof. It is easy to see that Θ is *similar* to the operator $\Psi : \mathbb{R}^{|V|} \rightarrow \mathbb{R}^{|V|}$ defined by $\Psi = \left(P + \Pi^{-1}P^T\Pi\right)/2$. Hence Θ and Ψ have the same set of eigenvalues. Assume that f is the eigenvector of Ψ with eigenvalue λ. Choose a vertex v such that $|f(v)| = \max_{u\in V}|f(u)|$.

Then we can show that $|\lambda| \leq 1$ by

$$\begin{aligned} |\lambda||f(v)| &= \left|\sum_{u\in V}\Psi(v,u)f(u)\right| \leq \sum_{u\in V}\Psi(v,u)|f(v)| \\ &= \frac{|f(v)|}{2}\left(\sum_{u\leftarrow v}p(v,u)+\sum_{u\rightarrow v}\frac{\pi(u)p(u,v)}{\pi(v)}\right) \\ &= |f(v)| \end{aligned}$$

In addition, we can show that $\Theta\sqrt{\pi} = \sqrt{\pi}$ by

$$\begin{aligned} &\frac{1}{2}\left(\sum_{u\rightarrow v}\frac{\pi(u)p(u,v)\sqrt{\pi(u)}}{\sqrt{\pi(u)\pi(v)}}+\sum_{u\leftarrow v}\frac{\pi(v)p(v,u)\sqrt{\pi(u)}}{\sqrt{\pi(v)\pi(u)}}\right) \\ = &\frac{1}{2}\left(\sum_{u\rightarrow v}\frac{\pi(u)p(u,v)}{\sqrt{\pi(v)}}+\sum_{u\leftarrow v}\frac{\pi(v)p(v,u)}{\sqrt{\pi(v)}}\right) \\ = &\frac{1}{2}\left(\frac{1}{\sqrt{\pi(v)}}\sum_{u\rightarrow v}\pi(u)p(u,v)+\sqrt{\pi(v)}\sum_{u\leftarrow v}p(v,u)\right) \\ = &\sqrt{\pi(v)} \end{aligned}$$

□

According to the previous lemmas, if function f is allowed to take arbitrary real values, then the directed graph partition problem (3.5) becomes

$$\begin{aligned} &\arg\min_{f\in\mathbb{R}^{|V|}}\langle f,(I-\Theta)f\rangle \\ &\text{subject to} \quad \|f\|=1,\ \langle f,\sqrt{\pi}\rangle = 0 \end{aligned} \tag{3.11}$$

Therefore, similar to the relaxed solution for undirected spectral clustering, the approximate solution for (3.11) is the second smallest eigenvector of $\Delta = I - \Theta$. Δ shall be called the *directed graph Laplacian*. I name this new clustering method as *directed spectral clustering*.

In addition, one can define the Cheeger's bound to indicate the quality of the approximation for directed spectral clustering. In undirected graphs, the Cheeger's constant

is defined as $h(G) = \min_S h(S)$, where $h(S) = \frac{\text{vol}(\partial S)}{\min\{\text{vol}(S), \text{vol}(S^c)\}}$. Similarly to undirected graphs, one can define the Cheeger's constant by replacing $\text{vol}(S), \text{vol}(S^c)$ and $\text{vol}(\partial S)$ to be our new definitions on directed graphs. The Cheeger's constant is bounded by the expression with the second smallest eigenvalue λ of Δ as $2h(G) \geq \lambda \geq \frac{h^2(G)}{2}$ (Chung, 2005).

3.2.2 Random Walk Interpretation

Clearly, in the directed normalized cut criterion (3.5), the ratio of $\text{vol}\,\partial S$ to $\text{vol}\,S$ is the probability that the random walk leaves S in the next step given that it is currently in S. A similar property holds for the ratio of $\text{vol}\,\partial S^c$ to $\text{vol}\,S^c$. Therefore, the directed normalized cut corresponds to finding a cut such that the probability of jumping between different clusters is small while the probability of staying within the same cluster is large, given that the current state is in stationary distribution. Clearly, the random walk model is a free parameter in the directed spectral clustering. Technically, the only requirement is that the transition probabilities of the random walk satisfies the balance equation $\pi(v) = \sum_{u \to v} \pi(u) p(u, v)$.

So far, I have assumed that the graph is strongly connected and aperiodic, which ensures that the natural random walk over the graph converges to a unique stationary distribution. Obviously this assumption cannot be guaranteed for a general directed graph. If the graph is not aperiodic but is strongly connected, one remedy is to introduce a *lazy random walk*. The lazy random walk has a transition probability as $P = (I + P_0)/2$, where P_0 is the original natural random walk defined in Section 3.2. Since it has self-loops, so clearly it is aperiodic. This modification is suggested by Chung (2005). Another situation which happens more often in real-world applications is that the graph is not strongly connected. If so, to remedy this problem, one can introduce the so-called *teleporting random walk* (Page et al., 1998) to replace the natural one.

Practical Analysis of Random Walks

In practice, the random walk should be defined according to specific context of problems. In this section, I analyze the behavior of different random walk models in the problem of Web communities identification (Huang et al., 2006d). Before I analyze the specific models,

I briefly review related work on identifying Web communities. The overview will help us understand the advantages of the new directed spectral clustering.

As we know, the Web is comprised of multiple communities (Flake et al., 2002) created by different groups of people having common interests. However, the sheer heterogeneity of Web users and authors—given diverse backgrounds and interests—hampers traditional information retrieval approaches that rely on content analysis alone. The identification of Web communities can help users with their information retrieval goals, by allowing the construction of pre-classified directories and the creation of more effective recommendation services.

The problem of identifying Web communities is clearly related to the more fundamental problem of graph partitioning. For general graph partitioning, one can often resort to straightforward principles such as unrestricted minimal cut. However, the graphs used by most such techniques are undirected, and therefore they ignore the directionality information encoded in Web hyperlinks (Flake et al., 2000; Ino et al., 2005). Another simple approach is to extract similarity measurements between neighboring vertices (Web pages) directly from the link structure to perform a generic clustering method (Kessler, 1963). However, the similarity should be measured from the *global* structure of the graph. A more global approach to Web graph clustering suggests, therefore, that some sort of aggregate similarity measure be used, such as those based on the spectrum of the connectivity matrix. For *undirected* graph clustering, a common suggestion is to partition by performing a singular value decomposition (SVD) on W (Perona and Freeman, 1998). However again, the connectivity matrix W is *not* symmetric.

By considering the directed links of Web pages, Kleinberg showed that the HITS ranking algorithm (Kleinberg, 1999) converges to a spectral method that uses the principle eigenvectors of W^TW and WW^T—the final weight scores for the authorities and hubs.[2] Later, it was observed that this technique can in fact be used to identify web communities, where Web pages with highest authority and hub scores are used to define the core of a community (Gibson et al., 1998). However, one can see that this approach reduces to SVD on an *undirected* graph weight matrices WW^T and W^TW. In fact, this approach suffers from two drawbacks: first, a straightforward graph partition method based on simply com-

[2]Vertices that have in-links are called *authorities*, vertices that have out-links are called *hubs*.

puting the principle eigenvectors is not very effective in general; and second, the directed hyperlink information is significantly diminished through the symmetric transformations. Regarding the first drawback, a more appropriate way to solve the graph partitioning problem is to consider it as a *balanced* minimum cut problem, which usually results in more accurate clusters being obtained. Although most versions of the balanced minimum cut are NP-complete, the eigenvectors of graph Laplacians (Chung, 1997) provide a good approximation to this NP-hard problem. Unfortunately, these methods have only been developed for undirected graphs, and do not consider directionality information.

To address these shortcomings, one requires a balanced spectral clustering principle that can take the directionality of Web hyperlinks into account. My new unsupervised method in directed graphs offers a mathematically clean solution to this problem. It minimizes a balanced cut criterion for directed graphs that has a very natural interpretation in a random walk framework.

In this section, I am addressing the specific role of random walks in Web clustering. It is critical to formulate a proper random walk model that ensures similar pages are grouped into coherent Web communities. Therefore, I analyze two random walk models with their variants that are sufficiently flexible to capture important aspects of Web graph topology, and disclose how walk connectivity is related to page similarity in directed spectral clustering. I will also investigate the performance of these random walk models in comparison with standard models of spectral clustering on undirected graphs (Gibson et al., 1998) in Section 3.2.4.

One-Step Random Walk The one-step random walk model I examine initially is the *teleporting random walk* model of Page et al. (1998). Given that the random surfer is currently at a vertex u: (a) with probability ϵ it chooses an outlink uniformly at random and follows the link to the next page; or (b) with probability $1 - \epsilon$ it jumps to a Web page uniformly at random over the entire Web (excluding itself). Here, a damping factor ϵ ($0 < \epsilon < 1$) is introduced in the case where the current page has no outlink. Such a random walk is guaranteed to converge to a unique stationary distribution.The transition probability $p_{tele}(u, v)$ between u and v under this model can be written as

$$p_{tele}(u, v) = \epsilon \frac{w(u, v)}{d^+(u)} + p_\epsilon(u, v),$$

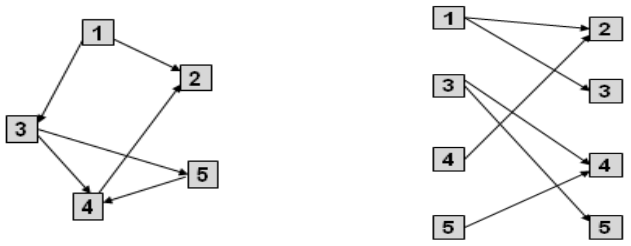


Figure 3.3: Constructing a bipartite graph from a directed graph. Left: directed graph. Right: bipartite graph. The hub set $H = \{1, 3, 4, 5\}$, and the authority set $A = \{2, 3, 4, 5\}$. Notice that the vertex indexed by $3, 4, 5$ are simultaneously in the hub and authority set.

where $p_\epsilon(u, v) = w(u, v)/\operatorname{vol} G$ if $d^+(u) = 0$ and $p_\epsilon(u, v) = (1-\epsilon)w(u, v)/\operatorname{vol} G$ if $d^+(u) > 0$; $\operatorname{vol} G = \sum_u (d^+(u) + d^-(u))$ in which d^- and d^+ are defined in (3.1) and (3.2).

This random walk makes the simple assumption that similar pages are directly linked. The stationary probability of a Web page corresponds to the frequency that a surfer visits the page following forward links. This can be viewed as an authority effect in the Web page ranking. I refer to this random walk as the one-step authority model (***OneStepA***). Conversely, one can consider another random walk that traverses *backward* along the hyperlinks (Ding et al., 2002). This is equivalent to the hub effect, since a good hub page should be able to visit many other related pages. Therefore, I refer to this random walk as the one-step hub model (***OneStepH***).

Two-Step Random Walk Web pages are "connected" by more than their direct hyperlinks. Intuitively, commonality between two Web pages is revealed by the presence of common co-citation or co-reference pages. The random walk should therefore also consider these implicit connections in Web community identification.

I now consider a *two-step random walk* model motivated by the Hubs and Authorities model by Kleinberg (1999) on a bipartite graph. A directed graph can naturally converted into a bipartite graph representing the hub and authority subsets of the data (Zhou et al., 2005b). Figure 3.3 depicts the construction of the bipartite graph. Assume

temporarily that each Web page has inlinks and outlinks. Then, starting from a page u, the random surfer first jumps backward to an adjacent hub vertex h with probability $p^-(u,h) = w(h,u)/d^-(u)$, then it jumps forward to a page v adjacent from h with probability $p^+(h,v) = w(h,v)/d^+(h)$. Then the two-step transition probability $p^A(u,v)$ between two authorities u and v is given by

$$p^A(u,v) = \sum_h p^-(u,h)p^+(h,v) \tag{3.12}$$

Proposition 3.2.5. *The stationary distribution π^A of p^A is*

$$\pi^A(u) = d^-(u)/\operatorname{vol} G^-$$

where $\operatorname{vol} G^- = \sum_{u\in V} d^-(u)$.

Proof. To show π^A satisfies the balance equation

$$\begin{aligned}\sum_{u\in V} \pi^A(u)p^A(u,v) &= \sum_{u\in V} \frac{d^-(u)}{\operatorname{vol} G^-} \sum_{h\in V} \frac{w(h,u)w(h,v)}{d^-(u)d^+(h)} \\ &= \frac{1}{\operatorname{vol} G^-} \sum_{h\in V} \frac{w(h,v)}{d^+(h)} \sum_{u\in V} w(h,u) = \frac{d^-(v)}{\operatorname{vol} G^-} = \pi^A(v)\end{aligned}$$

□

This random walk is performed by treating pages as authorities.

Using the same argument, one can define a two-step random walk by treating pages as hubs. The random walk performs among hubs u and v by first taking a forward step and then a backward step along the edges $u \to a$ and $a \leftarrow v$, yielding the transition probability between hubs

$$p^H(u,v) = \sum_a p^+(u,a)p^-(a,v) \tag{3.13}$$

Similarly, this random walk between hubs has the stationary distribution

$$\pi^H(u) = d^+(u)/\operatorname{vol} G^+$$

The two-step random walk exploits the co-citation and co-reference effects in the high level Web link topology. The assumption here is that two similar pages should share more common hubs or authorities.[3]

The above two-step random walks require that each Web page has inlinks and outlinks, but this is not always true for real Web graphs. To be able to handle the general case, I propose to combine the two-step random walk with a teleporting step, so that each forward and backward step through an outlink and a inlink has a damping factor. Therefore, to obtain the mixed two-step random walk, simply plug the modified transition probabilities p^- and p^+ into formulas (5.3) and (3.13) to modify p^A and p^H among authorities and hubs. In my experiments below I only use the mixed version of the two-step random walks, but for simplicity I just refer to them as ***TwoStepA*** and ***TwoStepH*** respectively. Finally, I consider a convex combination of the two types of two-step random walks that address the hyperlink structure in a more flexible manner $P = \beta P^A + (1-\beta)P^H$, where β is a tuning parameter that controls the different weights of co-citation and co-reference effects. The advantage of this combination is that it can help us determine which effect is dominant in the link structure, based on the results. Or conversely, given some prior knowledge about the levels of link structure, we can set a proper value for β that consistently matches the hyperlink topology.

Comparison of Different Random Walks To partition a directed Web graph, one can simply use the adjacency matrix A with unit weights (i.e., $a(u,v) = 1$ when $u \to v$). It is interesting to compare the results of the different random walk models and the symmetrized transformation models in this case. To demonstrate the differences in a simple toy example, I compute the second eigenvectors of Θ formulated as in (3.10) for both the one-step and two-step random walks on the graph in Figure 3.4-left. I set $\epsilon = 0.95$. I also obtain the principal eigenvectors of A^TA and AA^T, corresponding to the symmetrized authority and hub scores (Kleinberg, 1999). I refer to these symmetrized (undirected) methods as ***Auth*** and ***Hub*** respectively.

One can partition the directed graph into two clusters by examining the values in the

[3]I briefly note that Lempel and Moran (2000) uses the stationary distribution proportional to vertex in-degrees to perform a simple ranking method and showed similar derivations of stationary distributions.

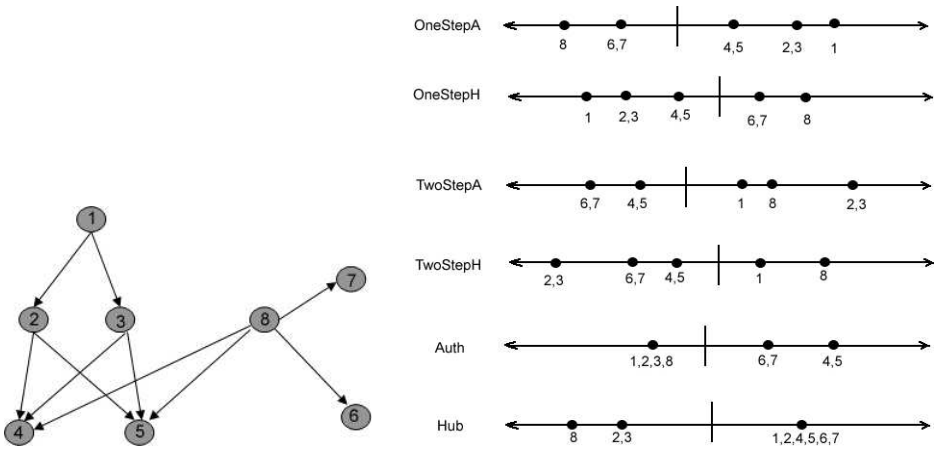


Figure 3.4: Left: A toy example of a directed graph. Right: Illustrating partitioning by sorted values. Here, "|" indicates the threshold value (zero) such that vertices on each sides are grouped into separate clusters.

eigenvector thresholding at zero. Pages within an initial grouping can then be partitioned further after the first partitioning (Chakrabarti et al., 1998), and so on. In addition to just partitioning the vertices, however, the eigenvector values can also be used to assign a *weight* or *confidence* that each Web page belongs to its assigned cluster. That is, the greater the eigenvector value at a page, the more likely the page is to belong to the given cluster. I will therefore refer to these values as the *weights* of pages below. The visualization of the partitioning by assigning each vertex on a solid line is as shown in Figure 3.4-right.

In the toy example, the partitions are the same for OneStepA and OneStepH, which tend to extract highly correlated clusters via direct connections. Moreover the vertices that have large values (e.g., 1 and 8) are also the vertices that have the highest stationary distributions under the random walks. It is known that PageRank ranks Web pages by their stationary distribution, but pages with high stationary probabilities might be of dissimilar topics. However, besides clustering, this method can provide rearranged rankings within each cluster which is very useful to current search engines.

TwoStepA tends to group strong authorities (4, 5, 6, 7) together that are linked by common pages. TwoStepH extracts the hub vertices (1, 8) that link to similar vertices directly and/or indirectly, e.g., vertex 1 points to vertices 4 and 5 after passing 2 and

3. Vertex 8 points to vertices 4 and 5 directly. This random walk tends to group good hubs that link to common pages either implicitly or explicitly. The partition using the symmetrized authority score is similar to TwoStepA, but it does not distinguish among the vertices 1, 2, 3 and 8. The partition using the symmetrized hub score also ignores any differences among the vertices in each group, and is thereby less meaningful.

Random walks are able to effectively capture the differences between direct hyperlink and indirect second order hyperlink topologies that have different co-citation and co-reference patterns in directed spectral clustering. All of these can be exploited to efficiently identify vertex communities via directed spectral clustering. Section 3.2.4 demonstrates more extensive experiments in Web communities identification.

3.2.3 k-way Directed Spectral Partioning

It is easy to extend the directed spectral clustering technique to computing k-way partitions instead of just 2-partitions. Define a k-way partition to be $V = V_1 \cup V_2 \cup \cdots \cup V_k$, where $V_i \cap V_j = \emptyset$ for all $1 \leq i, j \leq k$, $i \neq j$. Let P_k denote a k-partition. Then we may obtain a k-way partition by minimizing

$$\mathrm{Ncut}(P_k) = \sum_{1 \leq i \leq k} \frac{\mathrm{vol}\, \partial V_i}{\mathrm{vol}\, V_i} \tag{3.14}$$

Using similar derivation in Section 2.2.5, it is easy to see that the solution of the corresponding relaxed optimization problem of (3.14) can be any orthonormal basis for the linear space spanned by the eigenvectors of Θ corresponding to the k largest eigenvalues.

3.2.4 Evaluation and Comparison of Directed Spectral Clustering

Now I show real world evaluations of the new directed spectral clustering algorithm in terms of using different random walk models in the practical problem of Web communities identification. I examine various random walk models that capture different level aspects of hyperlink connectivity. In addition, I examine the performance of different random walk models and damping factors in identifying Web communities from pure graph topology.

The empirical results provide practical insights in applying directed spectral clustering in real-world Web clustering.

I also perform comparison experiments to see the different performance between directed and undirected methods in the problem of Web clustering. This demonstrates the directionality does play an important role in Web clustering.

Experimental Design

Root queries	vertex num	edge num
1. "waterloo"	2130	4688
2. "movies"+"olympics"	6634	65536
3. "risk analysis"+"bussiness optimization"	3357	10490
4. "differential geometry"+ "parallel computing"	2575	6844
5. "data mining"+"computer vision"	3907	12416
6. "body arts"+"fashion design"	3091	4122

Table 3.1: Web graphs statistics

I construct Web graphs of varying degrees of difficulty by either building the graph from a single topic query, which results in multiple topics that can be hard to distinguish, or building the graph from multiple queries, which results in a few more easily distinguishable topics. To obtain Web graphs, I first chose some *root queries*, submited these to Google, and retrieved the first t html pages (not including pdf or ps files). For a given query or set of queries, I then combined the retrieved pages as *roots* and perform a one level expansion by adding pages that are linked from or link to the root pages. Finally, I filtered out non-informative links that exist among Web pages as follows. I restrict the number of pages that link to or are pointed to by every root URL to be at most d pages. This operation was first proposed by Kleinberg (1999). I also filter out all *cgi scripts* links. I set t and d equal to 100 and 50 respectively. The collections I finally obtain are relatively sparse graphs. In the experiments, I use several groups of root queries that focus on a variety of interests. Pages retrieved from queries that have significant overlap intuitively should increase the difficulty of Web page clustering. The query statistics are listed in Table 3.1.

Results

Choosing Parameters Practically, two parameters need to be selected when defining the random walks on Web graphs: the damping factor ϵ in the one-step and two-step random walks, and the tuning parameter β in the two-step random walks. I test with 2 root queries using the damping factor ϵ set to 0.75, 0.85 and 0.95. Clustering performance is evaluated by counting the correctly classified pages that have the 30 greatest weights among those ranked within top 100 by Google.

Figures 3.5 plot the confusion matrix[4] values corresponding to the numbers of pages among the 30 with the greatest weight that are classified as "movie" (class 1) and "olympics" (class 2). Ideally, the best result should have corresponding numbers of 30, 0, 0, 30. Since OneStepA and OneStepH give very similar results in this experiment, I only show the results of OneStepA.

One can see from these figures that the directed spectral method with OneStepA obtains the best performance when ϵ equals 0.85. Thus, we fix this value for OneStepA in later experiments. For TwoStepA, the results are competitive when ϵ takes value 0.85 and 0.95. Since each result has a better performance for one of the communities, I choose $\epsilon = 0.90$ as an compromise value in the following experiments.

Next, I consider the tuning parameter β that balances between P^A and P^H in the two-step random walk. Figure 3.6 (left) shows the results when β changes from 1 to 0 in the "movies+olympic" Web graph. Instead of reporting the confusion matrix values in detail, I summarize it by the *F measure*, which can be derived from the confusion matrix as $\frac{2(precision \times recall)}{(precision+recall)}$ where $precision = C_{11}/(C_{11} + C_{21})$ and $recall = C_{11}/(C_{11} + C_{12})$. The Figure shows that the best performance is obtained when $\beta = 1$. This means that the Web page similarities are most correctly assessed when the transition matrix is P^A for this Web graph. Not surprisingly, this result is consistent with the ranking methods that consider inlink degree and authority scores from $A^T A$ (Gibson et al., 1998; Lempel and Moran,

[4]A confusion matrix C contains information about actual and predicted classifications. The elements in a confusion matrix for a two class classifier are: C_{11} is the number of correct predictions that an item is positive; C_{12} is the number of incorrect predictions that an item is positive; C_{21} is the number of incorrect of predictions that an item is negative; and C_{22} is the number of correct predictions that an item is negative.

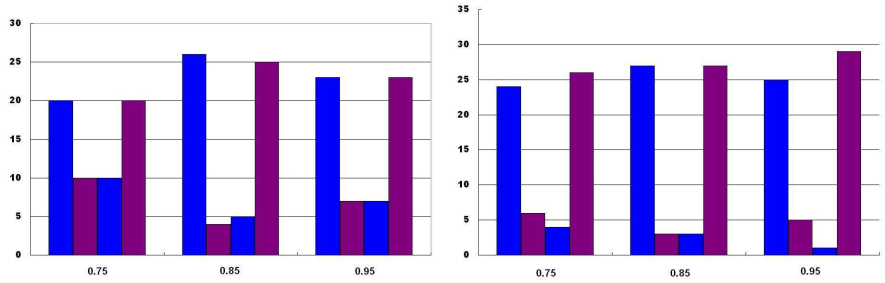


Figure 3.5: OneStepA results(left) and TwoStepA results(right). Plot of confusion matrix values C_{11}, C_{12}, C_{21}, C_{22}(from left to right of each column block) for $\epsilon = 0.75, 0.85, 0.95$.

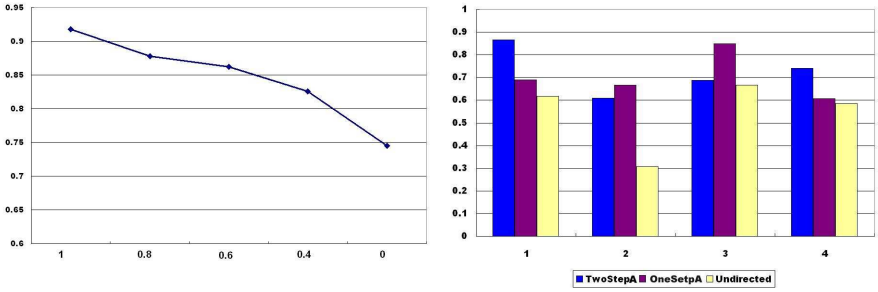


Figure 3.6: Left: F scores when β changes in two-step random walk, $\epsilon = 0.90$. Right: F score for 4 binary clustering tasks. Blue: TwoStepA, Red: OneStepA, Yellow: Undirected

2000). These studies have already shown that important pages can be found by evaluating their authority scores only. Thus, I set $\beta = 1$ in my following experiments, although I should point out that this is not a globally optimal choice.

Single Broad-topic Query Table 3.2 lists the communities detected by the directed spectral method using OneStepA. For each community, I list the URLs with significant PageRanks.

It is not hard to see that using only hyperlink structure, one can still identify reasonable communities from a Web graph constructed by a single broad topic query. The weights of pages in clusters 1 to 4 are closer to each other than to the pages in other clusters. This discloses that the first 4 clusters are related within a broader scope: they are mainly

Table 3.2: Communities from query "waterloo"

Cluster 1: Pages from universities and schools at Waterloo, Canada	Cluster 2: Pages for the public community service in Waterloo, Canada
www.uwaterloo.ca/ www.wlu.ca/ www.lib.uwaterloo.ca/ www.math.uwaterloo.ca/ www.cs.uwaterloo.ca/ www.wcdsb.edu.on.ca/	www.city.waterloo.on.ca/ www.waterloorecords.com/ www.therecord.com/ www.wpl.ca/ www.wrps.on.ca/ www.oktoberfest.ca/
Cluster 3: Pages for living at Waterloo, Canada	Cluster 4: Pages for life at Waterloo, Canada
www.waterlooinn.com/ www.waterloochamber.org/ www.kwhumane.com/ www.kwsymphony.on.ca/	www.kwymca.org/ www.waterloo.ca/ www.kwag.on.ca/ www.uptownwaterloojazz.ca/ www.kwsc.org/ www.waterloo-biofilter.com/ www.wnhydro.com/
Cluster 5: Pages for Waterloo, Iowa, USA	Cluster 6: Pages for Waterloo in the USA
www.wplwloo.lib.ia.us/waterloo/ www.wcfsymphony.org/ www.waterloocvb.org/ www.waterlooindustries.com/	www.waterloobucks.com/ www.waterloo.k12.ia.us/ www.waterloo.il.us/ www.waterlooindustries.com/
Cluster 7: Pages for Waterloo in Europe	Clusters 8 and 9: Pages for the history of Waterloo from public pages and from wiki
www.trabel.com/waterloo/ waterloo-thebattle.htm/ www.waterloo.org.uk/ www.trabel.com/waterloo/waterloo.htm/ www.napoleonguide.com/ battle_waterloo.htm/ www.waterloo.co.uk/	www.garywill.com/waterloo/ history.htm/ www.bbc.co.uk/history/war/ trafalgar_waterloo/ en.wikipedia.org/wiki/ Battle_of_Waterloo/ en.wikipedia.org/wiki/Waterloo_station/

pages from Waterloo, Canada, including academic institutions, social communities and living. The observation that the weights of clusters 5 and 6 are closer to each other than to the others identifies they are the pages of Waterloo locales in the US. The sub-topics are generalized upward to larger common topics. Cluster 9 identifies the pages from Wikipedia, even though I eliminate links among pages from the same domain.

Multiple Topic Related Queries I also evaluate clustering performance for 4 Web graphs that are obtained from multiple root queries. I compare the directed spectral methods using one-step random walk and two-step random walks to the undirected method that uses the symmetrized authority scores from $A^T A$ (referred to as the undirected method

Table 3.3: Pages with the top 10 significant weights for Queries of "computer vision" + "data mining"

Directed spectral method with OneStepA		**Undirected method.**	
URL	Cat	URL	Cat
cmp.felk.cvut.cz/eccv2004/	1	dms.irb.hr/index.php	2
iris.usc.edu/Vision-Notes/bibliography/contents.html	1	www.comp.leeds.ac.uk/nlp/	2
www.intel.com/research/mrl/research/opencv/	1	www.comp.leeds.ac.uk/vision/	1
marathon.csee.usf.edu/	1	www.statsoft.com/textbook/stdatmin.html	2
vis-www.cs.umass.edu/	1	lear.inrialpes.fr/people/triggs/events/iccv03/	1
www.cs.cmu.edu/ cil/vision.html	1	dir.groups.yahoo.com/group/datamining2/	2
www.sciencedirect.com/science/journal/10773142	1	www.acv.ac.at/	1
www.cs.cmu.edu/ cil/v-source.html	1	www-ai.ijs.si/SasoDzeroski/RDMBook/	2
iris.usc.edu/Information/Iris-Conferences.html	1	www.autonlab.org/tutorials/	2
homepages.inf.ed.ac.uk/rbf/CVonline/	1	www.cs.columbia.edu/ sal/hpapers/USENIX/ usenix.html	2
itmanagement.webopedia.com/TERM/D/ data_mining.html	2	www.scd.ucar.edu/hps/GROUPS/dm/dm.html	2
www.ncdm.uic.edu/	2	www.kdnuggets.com/	2
www.kdnuggets.com/	2	www.spss.com/	2
www.dmg.org/	2	www.eco.utexas.edu/ norman/BUS.FOR/course.mat/ Alex/	2
www.salforddatamining.com/	2	www.acm.org/sigkdd/	2
www.spss.com/	2	www.infogoal.com/dmc/dmcdwh.htm	2
www.acm.org/sigkdd/	2	www.the-data-mine.com/	2
www.megaputer.com/	2	www.thearling.com/text/dmwhite/dmwhite.htm	2
www.cacs.louisiana.edu/ icdm05/	2	www.ncdm.uic.edu/	2

in the results) as used by Gibson et al. (1998).

Figure 3.6–Right shows the clustering results for 4 Web graphs obtained from root queries 3, 4, 5 and 6. Not surprisingly, both of the directed spectral methods outperformed the undirected method in all cases.

I also show some of the clustering results by listing the highly ranked URLs with the most significant weights in corresponding communities in Tables 3.3 and 3.4."Cat" denotes the ture category for each URL. Once again, we can see that the directed spectral methods work better than the undirected method by tending to group pages more correctly. For example, in Table 3.3, the pages correctly clustered in the data mining community are about major conferences, term explanations, and companies in data mining. In Table 3.4, we see in the olympics community, multiple homepages from the olympic game hosts were obtained. Although these pages do not have hyperlinks between them, they all are pointed

Table 3.4: Pages with top 15 significant weights for Queries "movies"+ "olympics"

Directed spectral method with TwoStepA		**Undirected method**	
URL	Cat	URL	Cat
www.saltlake2002.com/	1	www.fhw.gr/olympics/ancient/	1
www.specialolympics.org/	1	cityguide.aol.com/main.adp	1
www.olympic.org/	1	www.dallasnews.com/sharedcontent/dws/spt/olympics/vitindex.html	1
www.torino2006.it	1	www.baltimoresun.com/sports/olympics/	1
sports.espn.go.com/oly/index	1	www.latimes.com/sports/olympics/	1
www.athens2004.com/athens2004/	1	diveintomark.org/howto/ipod-dvd-ripping-guide/	2
www.perseus.tufts.edu/Olympics/	1	movies.nytimes.com/pages/movies/	2
www.perseus.tufts.edu/Olympics/sports.html	1	news.bbc.co.uk/sport1/hi/other_sports/olympics_2012/default.stm	1
news.bbc.co.uk/sport1/hi/olympics_2004/default.stm	1	www.austin360.com/movies/content/movies/	2
www.nbcolympics.com/	1	www.musicfromthemovies.com/default.asp	2
www.olympics.com.au/	1	sports.yahoo.com/olympics	1
www.fhw.gr/projects/olympics/	1	movies.yahoo.com/mv/upcoming/	2
www.london2012.org/	1	www.fairolympics.org/en/	2
en.beijing-2008.org/	1	cbs.sportsline.com/u/olympics/2002/	1
www.imdb.com/	2	www.imdb.com/	2
us.imdb.com/	2	us.imdb.com/	2
www.imdb.com/search	2	rogerebert.suntimes.com/	2
movies.go.com/	2	www.lordoftherings.net/	2
www.usatoday.com/life/movies/front.htm	2	www.allmovie.com/	2
movies.aol.com/	2	www.rottentomatoes.com/	2
movies.yahoo.com/	2	www.infonegocio.com/xeron/bruno/olympics.html	1
movies.guide.real.com	2	www.brainpop.com/	2
www.rottentomatoes.com/	2	www.foxmovies.com/	2
www.hollywood.com/	2	www.hollywood.com/	2
www.boxofficemojo.com/	2	www.reel.com/	2
www.movieflix.com/	2	www.perseus.tufts.edu/Olympics/	1
www.ifilm.com/	2	www.ucmp.berkeley.edu/geology/tectonics.html	1

to by the Olympic Games organization (olympic.org). Thus, the two-step random walk was able to detect their similarity by identifying a common hub. Similar observations can be made about the pages classified in the movies community. In each of these tasks, the undirected method failed to identify pages from same communities, and tended to mix pages from the different communities.

The experiments again demonstrate that the directionality contains important information that significantly improves the unsupervised performance.

3.3 Supervised Learning on Directed Graphs

I consider learning a function for supervised learning on a directed graph by optimizing the combination

$$\min_{f\in\mathbb{R}^{|V|}} \sum_{v\in V} \text{loss}(f(v), y_v) + \lambda\Omega(f)$$

where $\Omega : V \to \mathbb{R}^+$ is the cut cost derived from directed Normalized Cut. As in undirected graphs, One can interpret the Normalized Cut cost of directed graph clustering to be a regularizer, which is a smoothness functional.

$$\Omega(f) = \frac{1}{2} \sum_{(u,v)\in E} \pi(u)p(u,v)\left(\frac{f(u)}{\sqrt{\pi(u)}} - \frac{f(v)}{\sqrt{\pi(v)}}\right)^2 \tag{3.15}$$

For an undirected graph, it is well-known that the stationary distribution of the natural random walk has a closed form expression $\pi(v) = d(v)/\sum_{u\in V} d(u)$. Substituting the closed form expression into (3.15), we have

$$\Omega(f) = \frac{1}{2\,\text{vol}\,G} \sum_{(u,v)\in E} w(u,v)\left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}}\right)^2$$

which is exactly the cut cost for the normalized spectral clustering in (2.23) and the regularizer on undirected graphs up to a constant of $\frac{1}{\text{vol}\,G}$. This validates the generalization of the new regularizer.

Below I will verify the cut cost for directed spectral clustering plays the same role as a standard regularizer.

3.3.1 Regularization over a Discrete Domain

I briefly develop a discrete analysis on directed graphs that extends the previous work on undirected graphs in Section 2.3.2. This allows the regularizer derived from directed spectral clustering to be reconstructed and generalized as a discrete analogue of classic regularization theory (Tikhonov and Arsenin, 1977; Wahba, 1990).

Let $\mathcal{H}(V)$ and $\mathcal{H}(E)$ be the space of functions over vertices and edges, as defined in Section 2.3.2.

Definition 3.3.1. *The graph gradient on a directed graph is an operator* $\nabla : \mathcal{H}(V) \to \mathcal{H}(E)$ *defined by*

$$(\nabla f)(u,v) = \sqrt{\pi(u)}\left(\sqrt{\frac{p(u,v)}{\pi(v)}}f(v) - \sqrt{\frac{p(u,v)}{\pi(u)}}f(u)\right) \tag{3.16}$$

Note that for an undirected graph, equation (3.16) reduces to

$$(\nabla f)(u,v) = \sqrt{\frac{w(u,v)}{d(v)}}f(v) - \sqrt{\frac{w(u,v)}{d(u)}}f(u)$$

which is exactly the same as the gradient operator defined in Section 2.3.2. Therefore, in the same way as in Section 2.3.2, one can recover natural representations of the divergence and the graph Laplacian for directed graphs as well.

Definition 3.3.2. *Let the graph divergence* $\operatorname{div} : \mathcal{H}(E) \to \mathcal{H}(V)$ *be an operator defined by*

$$(\operatorname{div} g)(v) = \frac{1}{\sqrt{\pi(v)}}\left(\sum_{u \leftarrow v}\sqrt{\pi(v)p(v,u)}g(v,u) - \sum_{u \to v}\sqrt{\pi(u)p(u,v)}g(u,v)\right)$$

Then we have the same proposition as in proposition 2.3.3: $\langle \nabla f, g\rangle_{\mathcal{H}(E)} = \langle f, -\operatorname{div} g\rangle_{\mathcal{H}(V)}$ with the same proof.

Definition 3.3.3. *Let the directed graph Laplacian* $\Delta : \mathcal{H}(V) \to \mathcal{H}(V)$ *on a directed graph defined by*

$$(\Delta f)(v) = f(v) - \frac{1}{2}\left(\sum_{u \to v}\frac{\pi(u)p(u,v)f(u)}{\sqrt{\pi(u)\pi(v)}} + \sum_{u \leftarrow v}\frac{\pi(v)p(v,u)f(u)}{\sqrt{\pi(v)\pi(u)}}\right) \tag{3.17}$$

Then we again have the same theorem as in Theorem 2.3.5: $\Delta f = -\frac{1}{2}\operatorname{div}(\nabla f)$, with a similar proof. It is not hard to see that in matrix notation, Δ can be written as

$$\Delta = I - \frac{\Pi^{1/2}P\Pi^{-1/2} + \Pi^{-1/2}P^T\Pi^{1/2}}{2}, \tag{3.18}$$

which is just the Laplace matrix for directed graphs appearing in $\Omega(f)$ in directed spectral clustering. It is also the same matrix that proposed by Chung (2005). For an undirected graph, (3.18) clearly reduces to the Laplacian for undirected graphs (Chung, 1997).

For well understanding the regularizer, I may compare it with an alternative approach which defines the gradient as

$$(\nabla f)(u,v) = \sqrt{\frac{w(u,v)}{d^-(v)}} f(v) - \sqrt{\frac{w(u,v)}{d^+(u)}} f(u), \text{for all} (u,v) \in E$$

The corresponding regularizer is

$$\Omega(f) = \sum_{(u,v)\in E} w(u,v) \left(\frac{f(u)}{\sqrt{d^+(u)}} - \frac{f(v)}{\sqrt{d^-(v)}} \right)^2 \tag{3.19}$$

A similar solution can be obtained from the corresponding optimization problem for unsupervised learning and later in semi-supervised learning. Clearly, this function also reduces to the regularizer for undirected graphs. At first glance, this function may look natural, but in the later experiments I will show that the algorithm based on this functional does not work as well as the previous one. This is because the directionality is only slightly taken into account via the degree normalization such that much valuable information for classification conveyed by the directionality is ignored by the corresponding algorithm. Once I remove the degree normalization from this functional, the resulted functional is totally insensitive to the directionality.

3.4 Semi-supervised Learning on Directed Graphs

I focus on a transducive problem in semi-supevised learning on a directed graph. Given a directed graph $G = (V, E)$ and a label set $\mathcal{Y} = \{1, -1\}$, assume that a subset $S \subset V$ of the vertices have been labeled. The problem is to classify the vertices in the complement of S. The graph G is assumed to be strongly connected and aperiodic.

The goal is to solve the optimization problem

$$\arg \min_{f \in \mathcal{H}(V)} \Omega(f) + \mu \|f - y\|^2 \tag{3.20}$$

where $\mu > 0$ is the regularization parameter. On one hand, one wants to keep a good partition on the graph which consists both labeled and unlabeled vertices, and on the

other hand one wants to minimize the loss over the labeled vertices. Section 3.3.1 shows that $\Omega(f)$ is a valid regularizer.

From Lemma 3.2.3, now differentiate 3.20 with respect to function f, we get $(I-\Theta)f^*+\mu(f^*-y)=0$. Define $\alpha=1/(1+\mu)$. This system may be written $(I-\alpha\Theta)f^*=(1-\alpha)y$. From Lemma 3.2.4, we easily know that $(I-\alpha\Theta)$ is positive definite and thus invertible.

It is worth mentioning that the approach of Zhou et al. (2005b) can also be derived from this algorithmic framework by defining a two-step random walk P^A which is the same as being discussed in Section 3.2.2.

3.4.1 Empirical Evaluation

To evaluate the regularization principle for directed graphs, I conduct semi-supervised classification experiments on two sets of data where directionality of the edges encodes meaningful information. Thus, the directed framework proposed above is expected to demonstrate an advantage in these cases.

Web Page Classification

The first data set I consider is the WebKB data set (Craven et al., 1998)—using a subset of the data set containing the pages from the four universities: Cornell, Texas, Washington and Wisconsin. I remove the pages that have no incoming nor outgoing links, reducing the number of pages to 858, 825, 1195 and 1238 respectively, for a total of 4116 web pages. All of these pages are manually classified into one of the seven categories: student, faculty, staff, department, course, project and other.

I compare the directed graph approaches to the alternate directed graph approach using a different regularizer in (3.19). I also compare this method to the schemes proposed in (Zhou et al., 2005b, 2004). Interestingly, it is not hard to show that the method in (Zhou et al., 2005b) is equivalent to my new directed method by using the TwoStepA. To distinguish among these approaches, I refer to them as *distribution regularization, degree regularization, second-order regularization*, and *undirected regularization* respectively. The distribution regularization method developed in this section uses teleporting random walks. Second-order regularization uses two-step random walks. Undirected regularization is the

method based on undirected graphs.

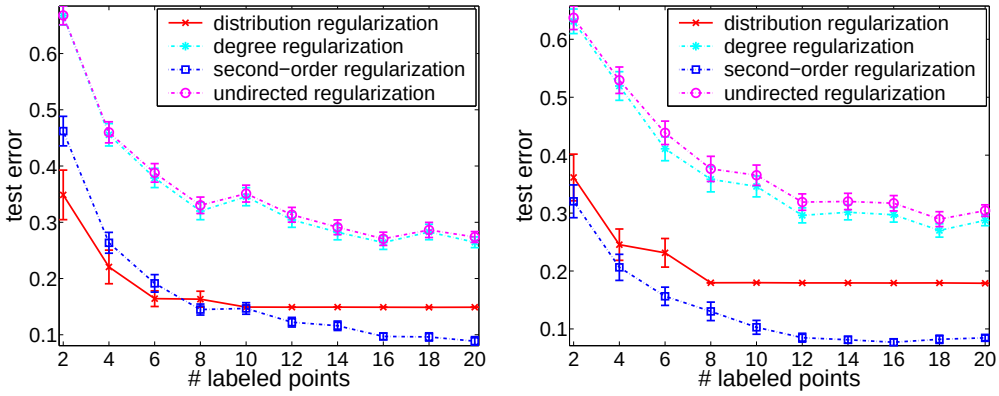
Although one can assign a weight to each hyperlink in this data that depends on the textual content or anchor text, here I am only interested in investigating what can be inferred solely from the link structure, and hence adopt that canonical (uniform 1) weight function. For every method, the regularization parameter is set to $\alpha = 0.1$. For the distribution regularization approach, I adopt the teleporting random walk with jump probability $\eta = 0.01$. Figure 3.7 reports the results. Each test error is averaged over 50 random repeats, where each repeat is guaranteed to have at least one labeled point in each class (otherwise the sampling is repeated).

These results show that the distribution regularization approach obtains superior results to the degree regularization method. Furthermore, the distribution regularization approach is competitive with second-order regularization as they both are directed methods. By contrast, the degree regularization approach shows weaker performance that is only comparable to the undirected regularization method. This shows that degree regularization only considers directionality slightly.

Protein Function Prediction

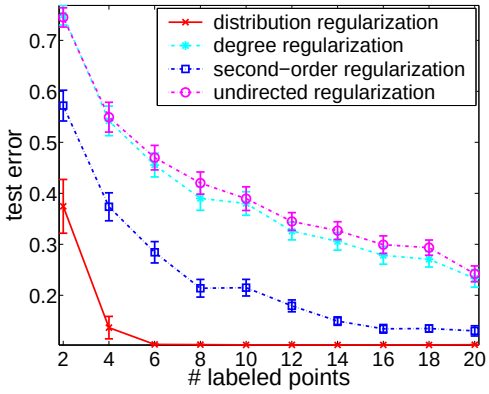
The next dataset I consider is a protein-protein interaction network constructed from yeast two-hybrid screens. Large-scale yeast two-hybrid screens are usually used to identify protein-protein interactions between full-length open reading frames (ORFs) predicted from the *saccharomyces cerevisiae* genome sequences (Uetz and et al., 2000; Ito and et al., 2001). I focus on the assignment of proteins to functional classes on the basis of the physical interaction network in yeast saccharomyces cerevisiae (Schwikowski et al., 2000).

The search for reliable methods for assigning protein function is the most challenging problem of the post-genomic era. Many approaches have been proposed for protein function prediction from protein-protein interaction networks using the information derived from sequence similarity, phylogenetic profiles, protein-protein interactions, and protein complexes (Vazquez et al., 2003). A map of protein-peortein interactions typically provides valuable insight into the cellular function and machinery of a proteome. A common approach is the majority vote, which involves assigning a function to an unclassified protein based on the most common function of its neighbors (Hishigaki et al., 2001; Schwikowski

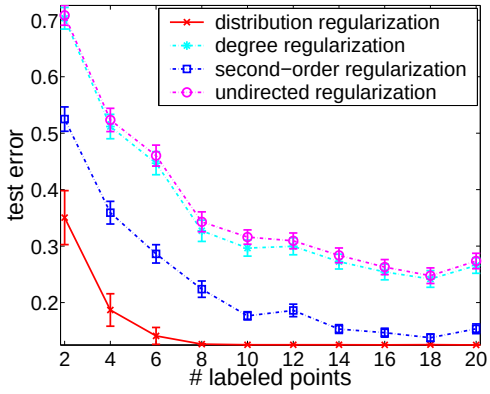


(a) Cornell (student)

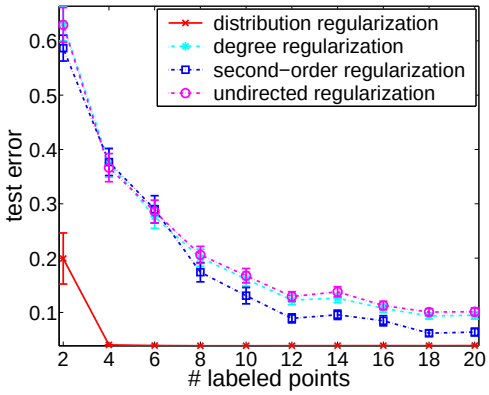
(b) Texas (student)



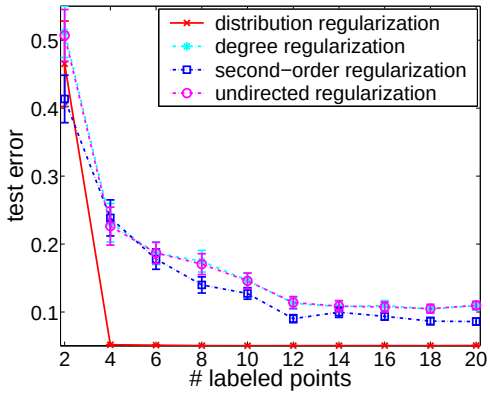
(c) Washington (student)



(d) Wisconsin (student)



(e) Cornell (faculty)



(f) Cornell (course)

Figure 3.7: Classification on the WebKB data set. Figures (a)-(d) depict the test errors of the regularization approaches on the classification problem of student vs. non-student in each university. Figures (e)-(f) illustrate the test errors of these methods on the classification problems of faculty vs. non-faculty and course vs. non-course in Cornell University.

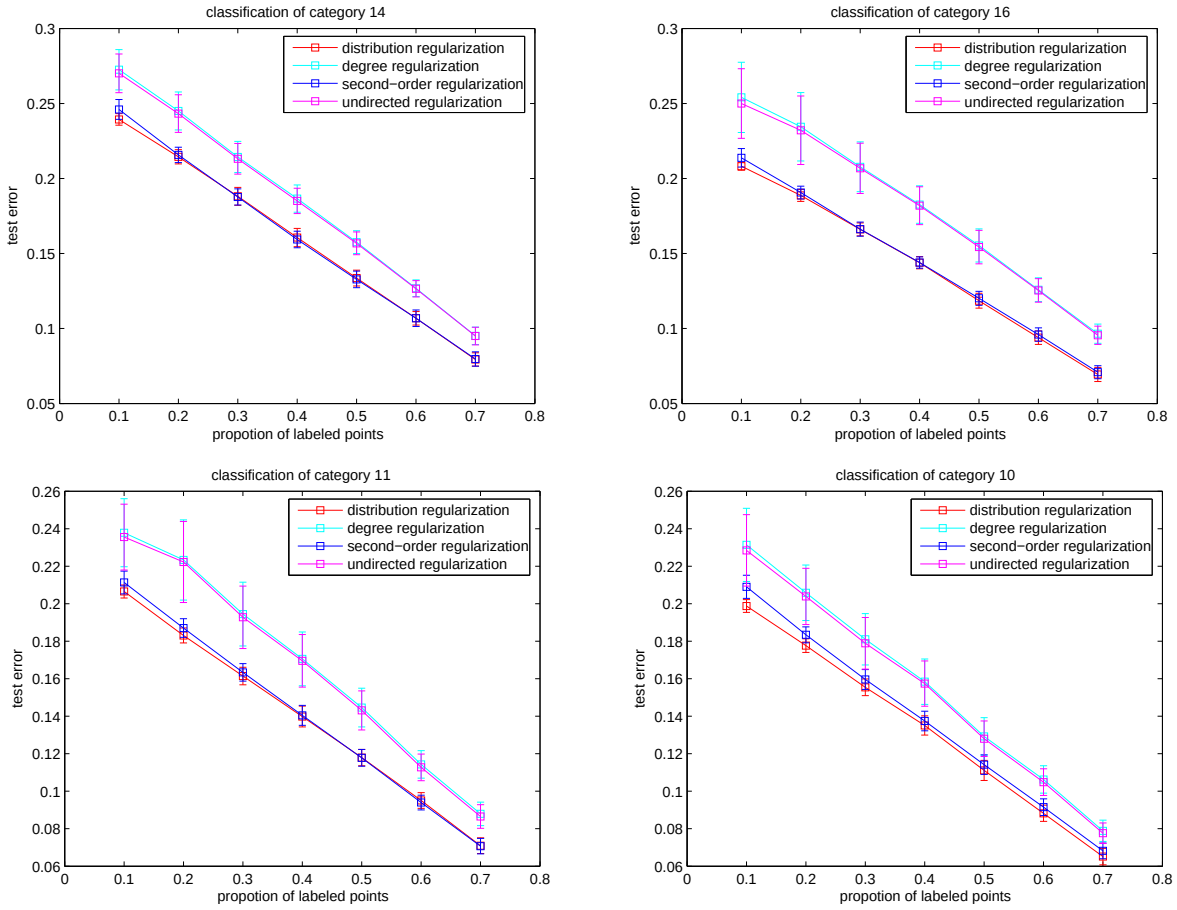


Figure 3.8: 4 methods comparison for different proportions of labelled proteins by test errors.

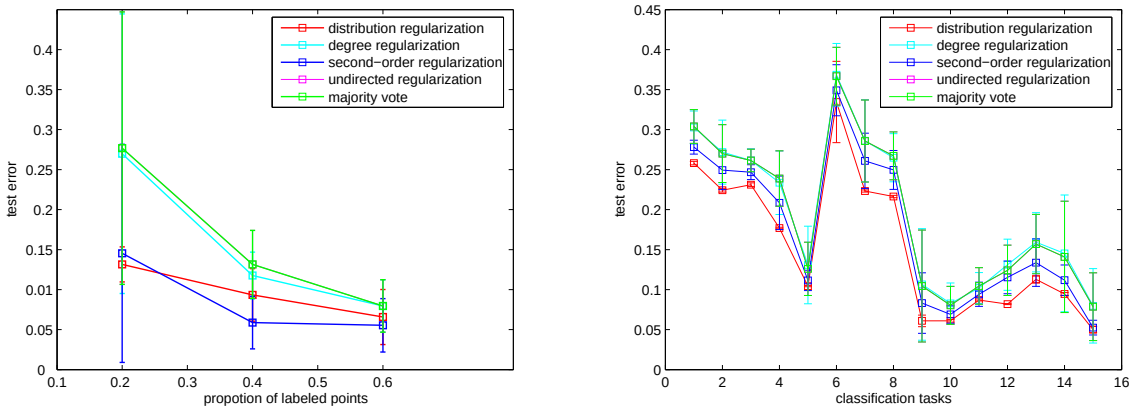


Figure 3.9: Left: A comparison to majority vote for 1 binary task. This category has the biggest number of positive labels of known proteins. Sampled protein proportion is from 0.2 to 0.6.; Right: A comparison to majority vote for 15 binary tasks. These categories have the most significant numbers of proteins with positive labels. Sampled protein proportion is 0.3.

et al., 2000; Vazquez et al., 2003), which is essentially a local nearest neighbor method based on an undirected interaction network. Much of the information contained in the global protein-protein interaction network is not fully explored.

Although more sophisticated methods have been proposed, all previous work took the protein-protein interaction network to be an undirected graph. Despite this, the network constructed via two-hybrid screens actually provides directionality information: given a pair of interacting proteins, one protein consists of a DNA-binding domain (DBD). The another protein consists of a transcriptional activation domain (AD) which is fused to a defined protein ORF. If these two proteins interact, a transcriptional activator is reconstituted that can activate transcription of a reporter gene and therefore the interaction generates an easily visible yeast colony that can be detected via the test. Therefore, in the interaction, one protein is considered to be the "bait" and the other the "predator" (or a "lock" and a "key" by another analogy). Thus the protein-protein interaction network can be more properly considered to be a directed graph, where an edge from protein A to protein B implies that A is a bait for predator B. This directionality information from

two-hybrid screens tests is ignored in all previous prediction methods and I would like to demonstrate the advantage of the new directed method comparing to the traditional undirected method.

I use the protein interaction information in (Ito and et al., 2001) extracted from EBI database (`ftp://ftp.ebi.ac.uk/pub/databases/IntAct/current/xml/`), and use the gold standard functional categories from the MIPS Comprehensive Yeast Genome Database (CYGD, `mips.gsf.de/genre/proj/yeast`) to provide the target labels. I consider all proteins that have functional categories in MIPS and obtain a resulting network of 3856 protein interactions involving 2926 proteins. There are 799 unclassified proteins among which only 161 proteins have at least one partner of known function and only 69 have two or more partners of known function. The categories are not mutually exclusive so they are appropriate to be used as independent binary classification tasks.

I examine the performance of the different algorithms when increasing the sampling proportion of proteins from 0.1 to 0.7 for each classification task. I test on 4 function prediction tasks for proteins selected from category IDs 14, 16, 11 and 10 in MIPS, which have 552, 476, 479 and 465 proteins respectively. The results are presented in Figure 3.8. I also compare to the simply majority vote method in Figures 3.9. The parameter setting is the same as in the Web classification experiment.

The experiments again prove that the directionality contains important information and it has significant impact on analysis of biology networks. By exploiting this information in learning methods, the performance is significantly improved. Therefore, the new directed method can be used as a general tool for assignment for protein function based on the interaction network constructed via two-hybrid screens tests.

3.5 Summary

I propose unsupervised and semi-supervised learning algorithms for learning from partially labeled data on a directed graph. The unsupervised learning algorithm generalizes the spectral clustering approach for undirected graphs. It is the first time that spectral methods have been successfully extended to directed graphs. The algorithms can be used to deal with structured data like the Web and biomedical networks. The empirical results in

Web classification and protein function prediction demonstrate the advantages of the new methods.

Additionally, to automatically identify Web communities from hyperlink topology via the new directed spectral clustering, I address a key component in directed spectral clustering, the random walk model, that is used to infer relationships between Web pages. I propose variations of random walk models raised from different Web topologies and investigate their effects for finding Web communities. The experiments show that the hyperlink structure of the Web provides very useful information, and that random walks are able to capture different relations based on various hyperlink topologies. The technique provides advanced tools for developing search engines in vertical searches.

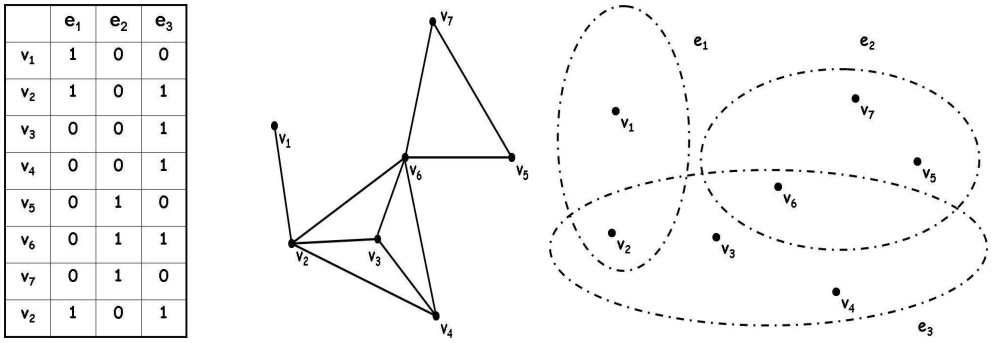
Please note that the new directed spectral clustering method I proposed here attacked clustering of a directed graph directly from the original asymmetric affinity matrix A. I have illustrated in experiments that traditional transformations used in previous literature, such as $A + A^T$ and $A^T A$, these transformations symmetrize A in a brute-forced way such that in many cases clusters in the original asymmetric A becomes partially or completely invisible after such symmetrization (Pentney and Meila, 2001). After our publication of the new directed spectral clustering method, Meila and Pentney (2007) observed that this new directed clustering method could be also interpreted as minimizing weighted cuts in directed graphs by spectral methods that amounts to symmetric spectral clustering on a "symmetrized" matrix Θ. However, the symmetrization is not trivial and this chapter provided an algorithmic approach to formulate the directed spectral clustering in a principled manner. The nice random walk interpretation derived from the principle can be further extended to methods on other types of graphs as we will see in the next two chapters.

Chapter 4

Beyond Pairs: Learning with Hypergraphs

In this chapter, I consider the problem of learning from more complex relationships between data items than just pairwise relationships encoded in a graph. In many real-world problems, we generally assume pairwise relationships among the objects of interest. An object set with pairwise relationships can be naturally illustrated as a graph. The graph can be undirected or directed, depending on whether the pairwise relationships are symmetric or not. However, representing a set of complex relational objects as undirected or directed graphs is not sufficient for many problems. To illustrate this point, consider the problem of grouping a collection of articles into different topics based only on author information. One could construct an undirected graph where two vertices are connected by an edge if the corresponding articles share at least one common author (Figure 4.1-middle). In this case an undirected graph based approach could be applied, e.g., using the spectral graph techniques introduced in Section 2.2. Although this method might sound natural, it obviously misses information about whether the same person wrote three or more articles. The lost information is potentially useful however, because articles by the same author are likely to belong to the same topic.

A natural way to remedy this type of information loss is to represent the data as a hypergraph instead. A hypergraph is a generalized form of graph where edges can connect more than two vertices. That is, each edge is a subset of vertices. Throughout



	e_1	e_2	e_3
v_1	1	0	0
v_2	1	0	1
v_3	0	0	1
v_4	0	0	1
v_5	0	1	0
v_6	0	1	1
v_7	0	1	0
v_2	1	0	1

Figure 4.1: Hypergraph vs. simple graph. Left: an author set $E = \{e_1, e_2, e_3\}$ and an article set $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$. The entry (v_i, e_j) is set to 1 if e_j is an author of article v_i, and 0 otherwise. Middle: an undirected graph in which two articles are joined together by an edge if there is at least one author in common. This graph cannot tell us whether the same person is the author of three or more articles or not. Right: a hypergraph which completely illustrates the complex relationships among authors and articles.

this chapter, I refer to undirected or directed graphs as simple graphs. Moreover, unless specialized otherwise simple graphs are assumed to be undirected. It is obvious that a simple graph is a special kind of hypergraph where each edge contains only two vertices. In the article clustering problem introduced above, one could construct a hypergraph with vertices representing articles and hyperedges corresponding to each author (Figure 4.1-right). In this case, each hyperedge would contain all the articles an author writes. In addition, one could simultaneously represent other relationships among the articles, such as the journal or conference proceedings where they were published. This information could naturally be represented by just adding further hyperedges. One can also use weights on the hyperedges to represent the relative importance of the different attributes. For instance, for an author working on a broad range of areas, we might assign a relatively small weight to his hyperedge.

The main contribution in this chapter is to develop new unsupervised and semi-supervised learning algorithms for hypergraphs. The unsupervised learning method introduced here generalizes previous spectral clustering techniques originally designed for simple graphs. I apply these hypergraph based approaches to real-world problems in embedding and clas-

sification. The results show the advantages of using hypergraphs over usual simple graphs in many cases.

I would like to note that there exists a significant literature on unsupervised learning on hypergraphs, which arises in a variety of practical problems, such as partitioning circuit netlists (Lengauer, 1990; Hagen and Kahng, 1992a), clustering categorial data (Gibson et al., 2000), and image segmentation (Agarwal et al., 2005). Unlike the present work however, these previous approaches generally transform the hypergraphs into simple graphs, using various sorts of heuristics and then applying standard graph based spectral clustering tehcniques. Gibson et al. (2000) proposed an iterative approach which was indeed specialized for hypergraphs, but did not consider a spectral method.

4.1 Preliminaries

Let V denote a finite set of vertices v, and let E be a family of subsets e of V such that $\cup_{e \in E} = V$. Then $G = (V, E)$ is a *hypergraph* with *vertex* set V and *hyperedge* set E. A hyperedge containing just two vertices is just a simple graph edge. There is a *hyperpath* between vertices v_1 and v_k if there is an alternating sequence of distinct vertices and hyperedges $v_1, e_1, v_2, e_2, \ldots, e_{k-1}, v_k$ such that $\{v_i, v_{i+1}\} \subseteq e_i$ for $1 \leq i \leq k-1$. A hypergraph is *connected* if there is a path for every pair of vertices. In the following, hypergraphs are always assumed to be connected. Given a set S, let $|S|$ denote the cardinality of S. Then the size of the set of vertices is denoted $|V|$, and the size of the set of hyperedges is $|E|$. A *weighted hypergraph* is a hypergraph that has a positive value $w(e)$ associated with each hyperedge e, referred to as the *weight* of hyperedge e. I denote a weighted hypergraph by $G = (V, E, w)$.

A hyperedge e is said to be *incident* with a vertex v when $v \in e$. For a vertex $v \in V$, the *degree* of v is defined by

$$d(v) = \sum_{\{e \in E | v \in e\}} w(e)$$

For a hyperedge $e \in E$, the degree is defined to be

$$\delta(e) = |e|$$

A hypergraph G can be represented by a $|V| \times |E|$ matrix H, called the *incidence matrix* of G such that H has entries $h(v, e) = 1$ if $v \in e$, and 0 otherwise, Then

$$d(v) = \sum_{e \in E} w(e) h(v, e) \tag{4.1}$$

and

$$\delta(e) = \sum_{v \in V} h(v, e) \tag{4.2}$$

Let D_v and D_e denote the diagonal matrices containing the vertex and hyperedge degrees respectively. Let W denote the diagonal matrix containing the weights. Then the *adjacency matrix* A of G is defined as

$$A = HWH^T - D_v \tag{4.3}$$

4.2 Unsupervised Learning on Hypergraphs

For a vertex subset $S \subset V$, let S^c denote the compliment of S. The problem of unsupervised learning on a hypergraph $G = (V, E)$ is to obtain a cut over G that partitions V into two parts S and S^c. A hyperedge e is cut if it is incident with the vertices in S and S^c simultaneously.

Given a vertex subset $S \in V$, define the *hyperedge boundary* ∂S of S to be a hypergedge set consisting of the hyeredges that are cut, i.e.,

$$\partial S := \{e \in E | e \cap S \neq \emptyset, e \cap S^c \neq \emptyset\}$$

For a vertex subset $S \subset V$, define the *volume* $\operatorname{vol} S$ of S to be the sum of the degrees of the vertices in S, that is

$$\operatorname{vol} S := \sum_{v \in S} d(v)$$

where $d(v)$ is defined in (4.1). Moreover, define the volume of ∂S by

$$\operatorname{vol} \partial S := \sum_{e \in \partial S} w(e) \frac{|e \cap S| \, |e \cap S^c|}{\delta(e)} \tag{4.4}$$

Clearly, $\operatorname{vol} \partial S = \operatorname{vol} \partial S^c$. The definition given by (4.4) can be understood as follows. Let us imagine each hyperedge e as a clique, i.e., a fully connected subgraph. To avoid

confusion, I call the edges in such an imaginary subgraph the subedges. Moreover, I assign the same weight $w(e)/\delta(e)$ to all subedges. Then, when a hyperedge e is cut, $|e \cap S|\,|e \cap S^c|$ subedges are cut. Hence a single term in (4.4) is the sum of the weights over the subedges that are cut.

4.2.1 Normalized Cut on Hypergraphs

A natural objective for clustering is to partition the hypergraph into disjoint components while cutting as few hyperedges as possible, while otherwise maintaining subgraphs that are as dense as possible. Therefore, I consider an analogue to the Normalized Cut criterion first introduced in Section 2.2

$$\begin{aligned} \operatorname*{argmin}_{\emptyset \neq S \subset V} \operatorname{Ncut}(S, S^c) &= \frac{\operatorname{vol} \partial S}{\operatorname{vol} S} + \frac{\operatorname{vol} \partial S^c}{\operatorname{vol} S^c} \\ &= \operatorname{vol} \partial S \left(\frac{1}{\operatorname{vol} S} + \frac{1}{\operatorname{vol} S^c} \right) \end{aligned} \tag{4.5}$$

For a simple graph, $|e \cap S| = |e \cap S^c| = 1$, and $\delta(e) = 2$, thus the right-hand side of equation reduces to the Normalized Cut on undirected graphs in Section 2.2, up to a constant factor.

In the following proposition, I relax (4.5) into a real-valued optimization problem to obtain an approximate solution.

Proposition 4.2.1. *Let α denote the ratio of* $\operatorname{vol} S / \operatorname{vol} V$. *Then*

$$\operatorname{Ncut}(S, S^c) = \frac{\sum\limits_{e \in E} \frac{w(e)}{\delta(e)} \sum\limits_{\{u,v\} \subseteq e} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2}{2\langle f, f \rangle}$$

where $f = \sqrt{d} \circ r$ such that

$$r(u) = \begin{cases} 2(1-\alpha) & u \in S \\ -2\alpha & u \in S^c \end{cases}$$

Moreover,

$$\sum_{v \in V} \sqrt{d(v)} f(v) = 0 \tag{4.6}$$

Proof. Let g be an indicator function with $g(v) = 1$ if $v \in S$ and -1 if $v \in S^c$. For a cut edge $e \in \partial S$ and the vertices $\{u, v\} \subseteq e, u \in S, v \in S^c$, one has $(g(u) - g(v))^2 = 4$; otherwise $(g(u) - g(v))^2 = 0$. Then the Normalized Cut criterion may be written

$$\mathrm{Ncut}(S, S^c) = \frac{\sum\limits_{e \in E} \frac{1}{\delta(e)} \sum\limits_{\{u,v\} \subseteq e} w(e) \left(g(u) - g(v)\right)^2}{8\alpha(1-\alpha) \sum\limits_{v \in V} g^2(v) d(v)}$$

Moreover, it is not hard to show that

$$\sum_{v \in V} d(v) r(v) = 0$$

which shows the constraint of (4.6) holds, and also we have that

$$\sum_{v \in V} r^2(v) d(v) = 4\alpha(1-\alpha) \sum_{v \in V} g^2(v) d(v)$$

Thus

$$\begin{aligned} \mathrm{Ncut}(S, S^c) &= \frac{\sum\limits_{e \in E} \frac{w(e)}{\delta(e)} \sum\limits_{\{u,v\} \subseteq e} \left(r(u) - r(v)\right)^2}{2 \sum\limits_{v \in V} r^2(v) d(v)} \\ &= \frac{\sum\limits_{e \in E} \frac{w(e)}{\delta(e)} \sum\limits_{\{u,v\} \subseteq e} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}}\right)^2}{2 \sum\limits_{v \in V} f^2(v)}, \end{aligned}$$

□

Efficient computation Let the elements of f take any continuous values, and define

$$\Omega(f) = \frac{1}{2} \sum_{e \in E} \frac{1}{\delta(e)} \sum_{\{u,v\} \subseteq e} w(e) \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}}\right)^2 \tag{4.7}$$

where Ω is just the numerator of $\mathrm{Ncut}(S)$ on a hypergraph.

To solve the problem of minimizing Normalized Cut objective efficiently, similarly to directed spectral clustering in Section 3.2.1, one can first define a matrix Δ as

$$\Delta = I - D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} \tag{4.8}$$

where I denotes the identity. The following lemma allows one to rewrite $\Omega(f)$ in terms of inner product that facilitates solving the optimization problem.

Lemma 4.2.2.

$$\Omega(f) = \langle f, \Delta f \rangle \tag{4.9}$$

Proof. To prove (4.9), consider the following reformulation of (4.7), recalling that h is the incidence matrix of the hypergraph as defined in Section 4.1 above.

$$\begin{aligned}
\Omega(f) &= \sum_{e\in E}\sum_{u,v\in V} \frac{w(e)h(u,e)h(v,e)}{\delta(e)} \left(\frac{f^2(u)}{d(u)} - \frac{f(u)f(v)}{\sqrt{d(u)d(v)}} \right) \\
&= \sum_{e\in E}\sum_{u\in V} \frac{w(e)h(u,e)f^2(u)}{d(u)} \sum_{v\in V} \frac{h(v,e)}{\delta(e)} - \sum_{e\in E}\sum_{u,v\in V} \frac{w(e)h(u,e)h(v,e)}{\delta(e)} \frac{f(u)f(v)}{\sqrt{d(u)d(v)}} \\
&= \sum_{u\in V} f^2(u) \sum_{e\in E} \frac{w(e)h(u,e)}{d(u)} - \sum_{e\in E}\sum_{u,v\in V} \frac{f(u)w(e)h(u,e)h(v,e)f(v)}{\sqrt{d(u)d(v)}\delta(e)} \\
&= \sum_{u\in V} f^2(u) - \sum_{e\in E}\sum_{u,v\in V} \frac{f(u)w(e)h(u,e)h(v,e)f(v)}{\sqrt{d(u)d(v)}\delta(e)}
\end{aligned}$$

The final expression can be shown to be equal to $f^T \Delta f$. This lemma also establishes that the matrix Δ is positive semi-definite. □

It is easy to verify that the smallest eigenvalue of Δ is 0 with eigenvector $\sqrt{d}$.

Then the combinatorial optimization problem (4.2.1) may be relaxed into

$$\begin{aligned}
&\operatorname*{argmin}_{f\in\mathbb{R}^{|V|}} \Omega(f) \\
&\text{subject to} \quad \|f\| = 1,\ \langle f, \sqrt{d} \rangle = 0
\end{aligned} \tag{4.10}$$

The solution of (4.10) is the normalized eigenvector Φ of the matrix Δ with the second smallest eigenvalue. Then the vertices of the hypergraph are partitioned into two parts

$S = \{v \in V | \Phi(v) \geq 0\}$ and $S^c = \{v \in V | \Phi(v) < 0\}$. I refer to this method as *hyperspectral Clustering*.

Notably, this method can be further generalized to *directed* hypergraphs, where each hyperedge e is an ordered pair (X, Y) such that $X \subseteq V$ is the *tail* of e and $Y \subseteq V \setminus X$ is its *head*.

As an aside, one can observe that for a simple graph the edge degree matrix D_e reduces to $2I$, and therefore

$$\begin{aligned} \Delta &= I - \frac{1}{2} D_v^{-1/2} H W H^T D_v^{-1/2} \\ &= I - \frac{1}{2} D_v^{-1/2} \left(D_v + A\right) D_v^{-1/2} \\ &= \frac{1}{2} \left(I - D_v^{-1/2} A D_v^{-1/2}\right) \end{aligned}$$

where A is the adjacency matrix for the graph. This corresponds to the definition of the undirected graph Laplacian given in Section 2.1, up to a constant factor. Therefore, Δ can be regarded as an analogue of the Laplacian for simple graphs, thus I suggestively call it *the hypergraph Laplacian*.

Correspondingly, for the special case of a graph, each edge is incident with only two vertices, and thus $\delta(e) = 2$. In this case (4.7) reduces to

$$\Omega(f) = \frac{1}{4} \sum_{e=(u,v) \in E} w(u, v) \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2$$

which is identical with the cut cost objective for normalized spectral clustering on undirected graphs in Section 2.2.3 and equivalently the regularizer for semi-supervised methods on undirected graphs in Section 2.4, up to a constant factor.

4.2.2 Random Walk Interpretation

Recall that there is a natural random walk interpretation for undirected spectral clustering (Section 2.2.4), and the random walk model is played a key role in directed spectral clustering as well (Section 3.2.2). The hypergraph normalized cut also has a nice random walk interpretation.

Consider a natural random walk on a hypergraph defined by the following transition rule: given a current vertex $u \in V$, first choose a hyperedge e among the hyperedges incident with u with probability proportional to the weight of e; then choose a vertex $v \in e$ uniformly at random and walk to v. Obviously, this procedure generalizes the natural random walk defined on simple graphs.

Let P denote the transition probability matrix of the random walk. Then each entry of P is

$$p(u,v) = \sum_{e \in E} w(e) \frac{h(u,e)}{d(u)} \frac{h(v,e)}{\delta(e)} \tag{4.11}$$

In matrix notation, $P = D_v^{-1} H W D_e^{-1} H^T$.

Proposition 4.2.3. *The random walk P has a stationary distribution π given by*

$$\pi(v) = \frac{d(v)}{\operatorname{vol} V}, \tag{4.12}$$

Proof. Equation (4.12) follows from the fact that

$$\begin{aligned}
\sum_{u \in V} \pi(u) p(u,v) &= \sum_{u \in V} \frac{d(u)}{\operatorname{vol} V} \sum_{e \in E} \frac{w(e) h(u,e) h(v,e)}{d(u) \delta(e)} \\
&= \frac{1}{\operatorname{vol} V} \sum_{u \in V} \sum_{e \in E} \frac{w(e) h(u,e) h(v,e)}{\delta(e)} \\
&= \frac{1}{\operatorname{vol} V} \sum_{e \in E} w(e) \sum_{u \in V} h(u,e) \frac{h(v,e)}{\delta(e)} \\
&= \frac{1}{\operatorname{vol} V} \sum_{e \in E} w(e) h(v,e) = \frac{d(v)}{\operatorname{vol} V} = \pi(v)
\end{aligned}$$

□

Interestingly, one can understand the Normalized Cut criterion proposed above in terms of this random walk. First note that the Normalized Cut criterion (4.2.1) can be transformed into

$$\operatorname{Ncut}(S, S^c) = \frac{\operatorname{vol} \partial S}{\operatorname{vol} V} \left(\frac{1}{\operatorname{vol} S / \operatorname{vol} V} + \frac{1}{\operatorname{vol} S^c / \operatorname{vol} V} \right)$$

From the stationary distribution (4.12), we have

$$\frac{\operatorname{vol} S}{\operatorname{vol} V} = \sum_{v \in S} \frac{d(v)}{\operatorname{vol} V} = \sum_{v \in V} \pi(v) \tag{4.13}$$

Hence this ratio is the probability with which the random walk occupies some vertex in S. Moreover,

$$\begin{aligned}
\frac{\operatorname{vol} \partial S}{\operatorname{vol} V} &= \sum_{e \in \partial S} \frac{w(e)}{\operatorname{vol} V} \frac{|e \cap S|\,|e \cap S^c|}{\delta(e)} \\
&= \sum_{e \in \partial S} \sum_{u \in e \cap S} \sum_{v \in e \cap S^c} \frac{w(e)}{\operatorname{vol} V} \frac{h(u,e) h(v,e)}{\delta(e)} \qquad (4.14) \\
&= \sum_{e \in \partial S} \sum_{u \in e \cap S} \sum_{v \in e \cap S^c} w(e) \frac{d(u)}{\operatorname{vol} V} \frac{h(u,e)}{d(u)} \frac{h(v,e)}{\delta(e)} \\
&= \sum_{u \in S} \sum_{v \in S^c} \frac{d(u)}{\operatorname{vol} V} \sum_{e \in S} w(e) \frac{h(u,e)}{d(u)} \frac{h(v,e)}{\delta(e)} \qquad (4.15) \\
&= \sum_{u \in S} \sum_{v \in S^c} \pi(u) p(u,v) \qquad (4.16)
\end{aligned}$$

Therefore, the ratio $\frac{\operatorname{vol} \partial S}{\operatorname{vol} V}$ is the probability that one witnesses a jump of the random walk from S to S^c under the stationary distribution. Thus the Normalized Cut criterion can be reinterpreted as seeking a cut where the probability of the random walk crossing between the different clusters is as small as possible, while the probability of staying within the same cluster is as large as possible. The intuitive consistency of this random walk view validates that the generalization of the normalized cut objective from simple graphs to hypergraphs is reasonable.

4.2.3 k-way Spectral Hypergraph Partitioning

It is straightforward to extend the hyperspectral clustering approach to computing a k-way partition. Define a k-way partition to be $V = V_1 \cup V_2 \cup \cdots \cup V_k$, where $V_i \cap V_j = \emptyset$ for all $1 \leq i, j \leq k$. Let P_k denote a k-way partition. Then a natural generalization of the previous criterion is

$$\operatorname{Ncut}(V_1, \ldots, V_k) = \sum_{1 \leq i \leq k} \frac{\operatorname{vol} \partial V_i}{\operatorname{vol} V_i} \tag{4.17}$$

Similarly, the combinatorial optimization problem can be relaxed into a real-valued one, where the solution is any orthogonal basis of the linear space spanned by the eigenvectors of Δ associated with the k smallest eigenvalues.

Theorem 4.2.4. *Assume a hypergraph G = (V,E,w) with $|V| = n$. Denote the eigenvalues of the Laplacian Δ of G by $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Define $c_k(G) = \min c(V_1, \ldots, V_k)$, where the minimization is over all k-way partitions. Then*

$$\sum_{i=1}^{k} \lambda_i \leq c_k(G)$$

Proof. Let r_i be a n-dimensional vector defined by $r_i(v) = 1$ if $v \in V_i$, and 0 otherwise. Then

$$c(V_1, \ldots, V_k) = \sum_{i=1}^{k} \frac{r_i^T (D_v - HWD_e^{-1}H^T) r_i}{r_i^T D_v r_i}$$

Define $s_i = D_v^{-1/2} r_i$, and $f_i = s_i / \|s_i\|$, where $\|\cdot\|$ denotes the usual Euclidean norm. Thus

$$c(V_1, \ldots, V_k) = \sum_{i=1}^{k} f_i^T \Delta f_i = \operatorname{tr} F^T \Delta F,$$

where $F = [f_1, \ldots, f_k]$. Clearly, $F^T F = I$. If allowing the elements of r_i to take arbitary continuous values rather than Boolean ones only, we have

$$c_k(G) = \min c(V_1, \ldots, V_k) \geq \min_{F^T F = I} \operatorname{tr} F^T \Delta F = \sum_{i=1}^{k} \lambda_i$$

The last equation follows from standard results in linear algebra. This complete the proof. □

The above result also shows that the real-valued optimization problem derived from the relaxation actually provides a lower bound of the original combinatorial optimization problem. Similar to the case for undirected graphs, it is unclear how to utilize multiple eigenvectors simultaneously to obtain a k-way partition, and how to determine the number of classes k. The same heuristics for undirected graphs can be applied here.

4.2.4 Evaluation of Hyperspectral Clustering

I considered clustering on the zoo data set from the UCI repository, which contains 100 animals with 17 Boolean-valued attributes classified into 7 different categories. The attributes include *hair, feathers, eggs, milk, legs, tail*, etc. Each attribute value is thought of as a hyperedge and the weights are simply set to 1. I embed the data set into Euclidean space using the eigenvectors of the hypergraph Laplacian (Figure 4.2). Clearly, the animals are well separated by the first three eigenvectors. Moreover, it is worth noticing that *seal* and *dolphin* are mapped to the positions between class 1 consisting of the animals having milk and living on land, and class 4 consisting of the animals living in the sea. A similar observation also holds for *seasnake*.

4.3 Supervised Learning on Hypergraphs

I consider learning a function for supervised learning on a hypergraph by optimizing the combination

$$\min_{f \in \mathbb{R}^{|V|}} \sum_{v \in V} \text{loss}(f(v), y_v) + \mu \Omega(f)$$

where $\Omega(f) : V \to \mathbb{R}^+$ is the cut cost defined from the Normalized Cut over labeled data only. Similar to undirected graphs and directed graphs, we can interpret the cut cost objective on hypergraphs to be a regularizer $\Omega : V \to \mathbb{R}^+$; that is, s a smoothness functional

$$\Omega(f) = \frac{1}{2} \sum_{e \in E} \frac{1}{\delta(e)} \sum_{\{u,v\} \subseteq e} w(e) \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2$$

We can also justify that the cut cost for hypergraphs plays the same role as a standard regularizer by defining a gradient operator on hypergraphs as for simple graphs.

4.4 Semi-supervised Learning on Hypergraphs

Again, I focus on a transducive learning problem in semi-supevised learning on hypergraphs. Recall that the transduction involves solving the following: Given a hypergraph

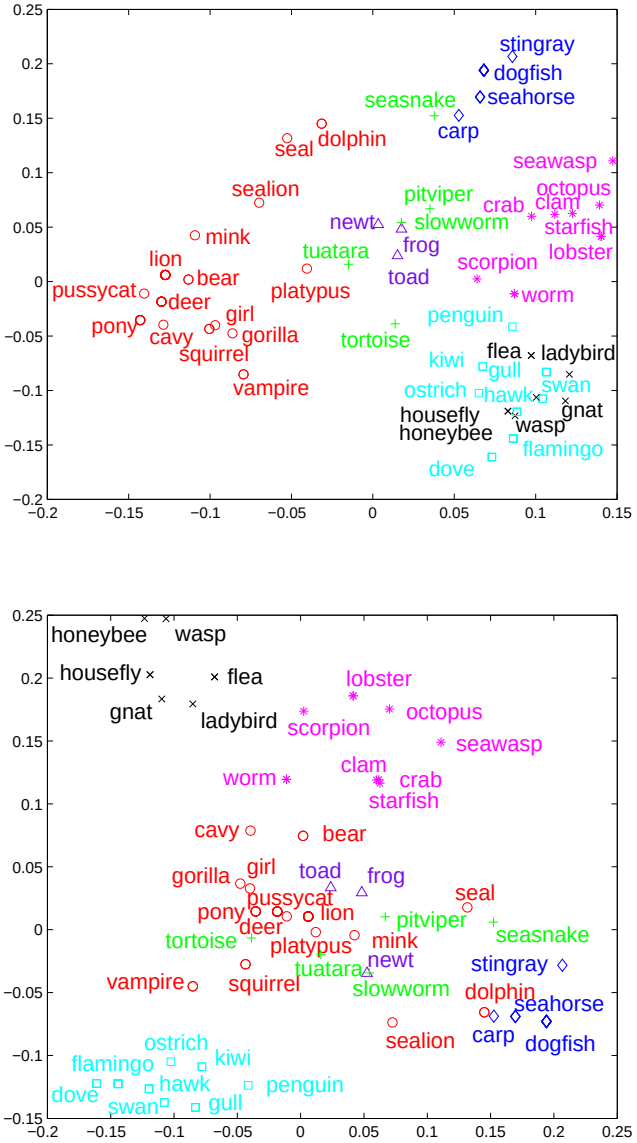


Figure 4.2: Embedding the zoo data set into Euclidean space. Top panel: the two eigenvectors of the hypergraph Laplacian corresponding to the second and third smallest eigenvalues. Bottom panel: the two eigenvectors of the hypergraph Laplacian corresponding to the third and fourth smallest eigenvalues. For animals having the same attributes, we randomly choose one as their representative to put in the figures. It is worth noticing that the animals like dolphin are between class 1 (denoted by $\circ$) containing the animals mostly milking and living on land, and class 4 (denoted by $\diamond$) containing the animals living in sea.

$G = (V, E)$ where the vertices in a nonempty subset $S \subset V$ are labeled as positive or negative, classify the remaining set of unlabeled vertices. Intuitively we want to assign the same labels to vertices that have many incident hyperedges in common.

It is straightforward to derive a transductive approach from the clustering scheme for hypergraphs. The optimization objective to solve this problem is

$$\arg \min_{f \in \mathbb{R}^{|V|}} \Omega(f) + \mu \|f - y\|^2 \tag{4.18}$$

where $\mu > 0$ is the tradeoff parameter.

As before, this is a general framework for learning with both labeled and unlabeled data on a hypergraph. This objective sums the changes of a classification function f over the hyperedges of the hypergraph. Of course, in addition to obtain a good cut on the hypergraph, one would also like f to match the initial label assignment as much as possible. To represent the initial assignment, let y denote the function in $\mathbb{R}^{|V|}$ defined by $y(v) = 1$ or -1 if vertex v has been labeled as positive or negative respectively, and 0 if it is unlabeled.

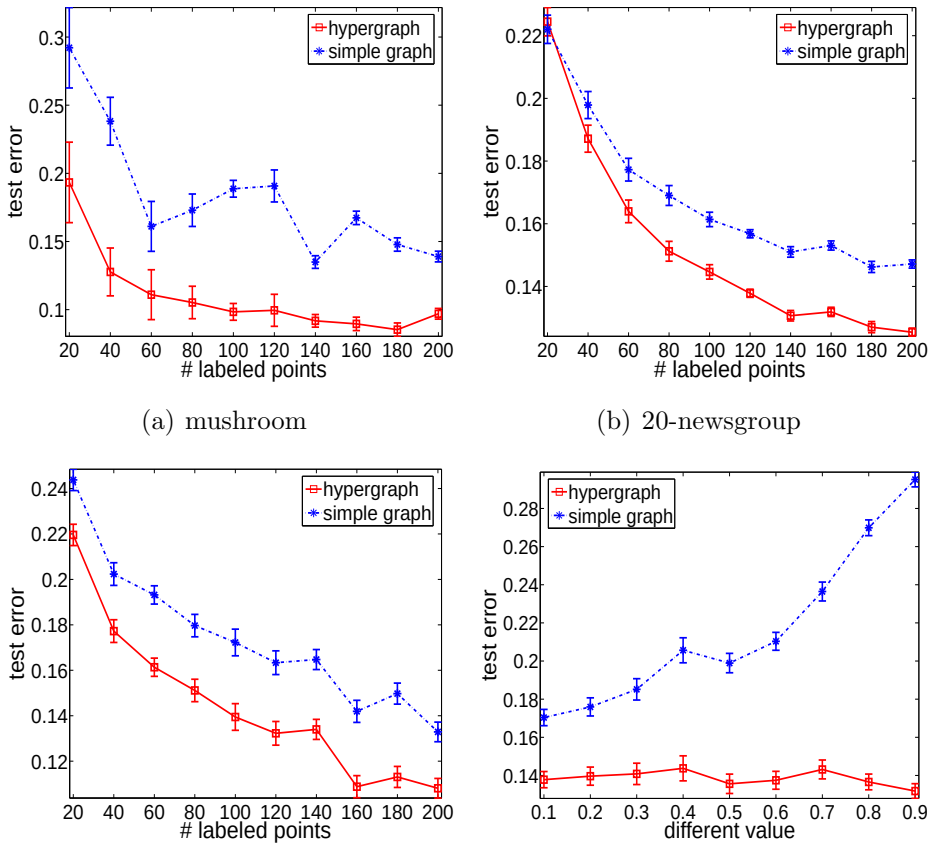
To solve (4.18), note that by (4.9) differentiating (4.18) with respect to f yields $\Delta f + \mu(f - y) = 0$, which is a linear equation. Hence we can obtain a closed form solution

$$f^* = (1 - \alpha)(I - \alpha\Theta)^{-1} y, \tag{4.19}$$

where $\alpha = 1/(1 + \mu)$ and $\Theta = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}$.

4.4.1 Empirical Evaluation

I applied the semi-supervised approach on hypergraphs to four real-world problems, and compared it to the method in (Zhou et al., 2004). I considered three data sets from the UCI repository, letter, mushroom and zoo, and the 20-newsgroup data set. In these data sets, the instances are described by vectors of attribute values. For the hypergraph approach, each attribute value is thought of as a hyperedge and the weights are simply set to 1. Choosing suitable weights is definitely an important problem that requires additional exploration however. I constructed a simple graph for each dataset—an edge is included if two instances share an attribute value, with an adjacency matrix defined in (4.3). The simple graph based approach is used as a baseline.



(a) mushroom (b) 20-newsgroup

(c) letter (d) α (letter)

Figure 4.3: Classification on the data sets with complex relationships. Fig. (a)-(c) depict the test errors of the hypergraph based approach and the baseline on three different data sets. The number of the labeled instances for each data set is increased from 20 to 200. Fig. (d) illustrates the influence of the regularization parameter α in the letter recognition task with 100 labeled instances.

The first data set I considered is the mushroom data set, which contains 8124 instances described by 22 categorical attributes (I remove the 11th attribute because it had missing values) classified into two classes, *edible* or *poisonous*. The two classes have have 4208 and 3916 instances respectively. The second data set I considered is the the 20-newsgroup data, which contains binary occurrence values for 100 words across 16242 news articles classified into 4 different classes corresponding the highest level of the original 20 newsgroups. These classes contain 4605, 3519, 2657 and 5461 articles respectively. The third task I considered is the letter data set, which contains images of five capital letters (A to E) represented by 16 integer attributes extracted from raster scan image of the letter. I use a subset of the data consisting of 789, 766, 736, 805 and 768 examples from each of the five classes respectively.

The experimental results of the last three tasks are shown in Figure 4.3(a)-4.3(c). The regularization parameter α for both the hypergraph and undirected graph approaches is fixed at 0.1. Each test error is averaged over 20 trials. In each trial, I randomly resample training points until there is an example from each class. The results clearly show that the hypergraph based method is consistently better than the baseline approach. The influence of the α for the letter recognition task is shown in Figure 4.3(d). It is interesting that the parameter α influences the performance of the baseline much more than the hypergraph based approach.

4.5 Summary

I generalized spectral clustering techniques to hypergraphs, and developed algorithms for unsupervised and semi-supervised learning on hypergraphs. These algorithms reduce to previous methods on simple graphs, which validates the new extension. This new extension also has a useful random walk interpretation similar to simple graphs.

It is interesting to consider applying the present methodology to a broader range of practical problems. One possible application is biological network analysis. Biological networks have mainly been modeled as simple graphs to date. It might be more sensible to model them as hypergraphs instead, to take into account more complex interactions. Another possible application is social network analysis. As recently pointed out by Bonacich

et al. (2004), many social transactions are supra-dyadic; they either involve more than two actors or they involve numerous aspects of the interaction setting. So standard network techniques are generally not adequate for analyzing these networks. Consequently, Bonacich et al. (2004) resorted to the concept of a hypergraph, and showed how the concept of network centrality can be adapted to hypergraphs.

Again we note that the hyperspectral clustering method also amounts to symmetric clustering on a "symmetrized" matrix Δ. However this symmetrization is non-trival. After the publication of our work, Agarwal et al. (2006) observed that the eigenvalue problem for Δ, the hypergraph Laplacian is equivalent to the problem derived from the normalized Laplacian for a transformed bipartite graph. Our new spectral clustering algorithm on hypergraphs provides a clear and principled algorithmic approach to formulate the eigenvalue problem directly that generalizes previous symmetrization on both undirected and directed graphs. The random walk interpretation again show the principled way of treating the graph as a Markov chain. We will see further extension on more complex graphs in next chapter.

Chapter 5

Beyond Homogeneity: Learning with Complex Networks

In this chapter I consider the problem of learning from even more complex relationships between data items, specifically considering heterogeneous networks that involve multiple objects types and relations. A common property of graphs that I have considered so far is that the vertices all represent data objects of same data type; e.g. Web pages or articles. However, in many other applications, e.g. citation network analysis, the data might involve multiple types of objects and relationships. For instance, a citation network could explicitly consider two types of objects, 'papers' and 'authors', that exhibit both paper-author interactions and paper-paper citation relationships. A typical learning problem we encounter in this scenario requires one to make inferences about one subset of objects (e.g. 'papers'), while using the remaining objects and relations to provide relevant information.

The main contribution of this chapter is to propose a simple, unified mechanism for incorporating information from multiple object types and relations when learning on a targeted subset. In this scheme, all sources of relevant information are efficiently propagated onto a target subgraph via marginalized random walks. I demonstrate that marginalized random walks can be used as a general technique for combining multiple sources of information in relational data. With this scheme, I formulate new inference algorithms for complex relational data, and quantify the performance of new approaches on real world data—achieving good results in many challenging problems.

5.1 Problem Overview

Before I present my work, I first briefly give an overview of the problem of learning with multiple relationships. Currently, *bipartite graphs* are the most commonly used representation in many text classification and clustering problems involving two types of data objects. For example, in document analysis one has documents and terms, where inference is based on the co-occurrence statistics of terms appearing in documents. Many algorithms have been developed for clustering in bipartite graphs, i.e., by Zha et al. (2001); Dhillon (2001); Dhillon et al. (2003); Tishby et al. (1999) and El-Yaniv and Souroujon (2001). The underlying intuition behind these approaches is that the similarities among one type of object can be used to cluster the other type of object.

One obvious limitation of current co-clustering methods is that they can only deal with two types of data objects, e.g. terms and documents, whereas most data sets may contain more than two types of objects. For example, in a paper clustering task on a citation network, beyond the bipartite interaction between papers and authors, it is also useful to consider other sources of relevant information, such as the conferences where the papers are published. Such additional paper-conference information could help enhance learning performance. In this case, one could construct a *tripartite graph* $G = (\langle A, B, C\rangle, E)$, where the vertex sets correspond to authors, papers, and conferences respectively, and E is the set of edges, as shown in Figure 5.1–left. One could consider addressing the problem of higher-order-partite graphs in a trivial manner by applying co-clustering on each pair of object types; that is, apply a co-clustering method on A, B, and then on B, C individually. However it is hard to ensure the solutions are consistent at the intersection on B. Bekkerman et al. (2005); Gao et al. (2005) proposed methods for solving clustering with interactive relationships among multiple object types using ideas from information theory and spectral graph clustering, but they need to employ sophisticated and computationally expensive methods like semidefinite programming to keep the partitions consistent.

Beyond tripartite clustering, more complex scenarios arise when one considers relationships among the same type of data objects. The work on clustering with bipartite and *k-partite* graphs has, for the most part, not taken the relationships *between* objects of the same type into account. Obviously, such information is simply ignored if we present the data as a k-partite graph.

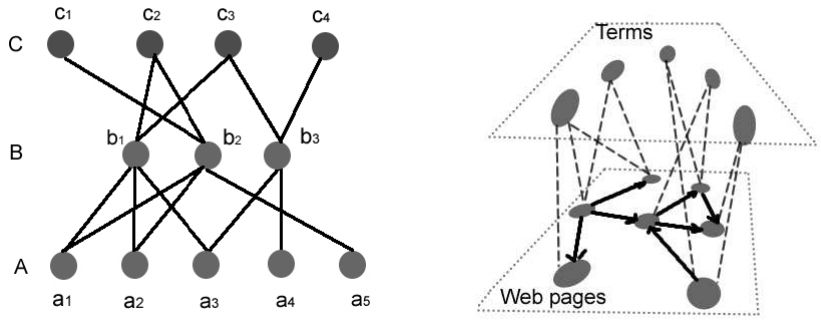


Figure 5.1: Left: A tripartite graph. Right: A graph of Web pages and terms.

Moving beyond documents and terms, if one considers clustering Web pages, it is clear that the bipartite graph information between Web pages and terms ignores significant relevant information encoded in the hyperlink structure (Page et al., 1998; Kleinberg, 1999; Zhou et al., 2005a). When clustering Web pages, it seems clear that both hyperlink structure and term co-occurrence are relevant sources of useful information that one would like to take account of in a unified way. Ideally, one would just model the relationships between Web pages and terms as vertices in a graph like the one shown in Figure 5.1–right. To the best of my knowledge, clustering in data sets with multiple object types, and multiple relationships between objects of various types has not been well studied in the graph partitioning literature.

Contributions I propose a simple and unified mechanism for learning in complex scenarios, like the ones shown above, in a graph based approach. I model all data objects as vertices in a graph; e.g., a k-partite graph or a mixed graph as shown in Figure 5.1–right. The graph based representation allows a simple mechanism for propagating useful information globally throughout a large database of objects: based on the graph, a natural random walk model can be defined that communicates information in a Markov chain. To summarize information from multiple object types and relations when making inferences about one object type, I marginalize the transition probability of the random walk onto the target subset, based on the transition probability of the *induced subgraph* and the transi-

tion probability between the subset and its *complement*. In this way, I obtain a valid, new random walk model on the induced subgraph that summarizes all external and internal sources of relevant information. Two objects in the target subgraph that share a lot of common external information will be highly linked in the induced random walk, even if they share no direct links in the induced subgraph. Once a valid random walk model has been defined, one can derive algorithms for various learning problems, by performing random walks over a Markov Chain. The idea of marginalization is a simple and elegant way of dealing with many types of complex scenarios uniformly. Interestingly, when dealing with graphs that happen to be bipartite, the unsupervised method implied by marginalization is equivalent to the spectral co-clustering method proposed by Zha et al. (2001) and Dhillon (2001). That is, I recover prominent bipartite graph based inference methods as a special case.

Furthermore, the marginalization idea can be extended to solve more general and interesting types of inference problems on graphs than having been commonly studied in graph partitoning. Consider the problem of clustering the set of blog pages on the Web. In a conventional approach, one could use the induced subgraph on blog pages (namely the subgraph of all the blog pages and their hyperlink structure) to classify the blog pages with respect to their common topics. However, the difficulty with this approach is that there is not much information in the hyperlinks between blog pages, as the owners of the blogs typically do not add links to other blogs if they do not know each other. Therefore, the information obtained directly from the subgraph is not enough to identify blogs of common interest. It therefore makes sense to explore the hyperlinks that *connect blog pages to other general web pages*. For example, people who are interested in computer programming might add a link from their blogs to the page "the art of computer programming" created by Donald Knuth. Although the blogs themselves may have only a few direct links, the blogs can still be clustered into identifiable communities by detecting the pages of common interest linked from the blogs. The scheme I propose can fully exploit all sources of relevant information in a graph of heterogeneous objects to achieve better performance on the target subset.

5.2 Preliminaries

A *bipartite graph* $G = (\langle A, B\rangle, E)$ is a graph that consists of two *disjoint* sets of vertices, A and B, and a set of unordered pairs as edges, E, between A and B. (Typically, the two sets represent different objects, e.g. documents and terms.) Each edge (a, b) is associated with a similarity weight $w(a, b)$. One can generalize bipartite graphs to higher order *k-partite graphs*, whose vertices are divided into k disjoint sets.

Given a graph $G = (V, E)$ (directed or undirected), and a subset $S \subset V$ of the vertices, the *induced subgraph* with respect to S is the subset V of vertices of G together with any edges whose endpoints are both in V.

Recall that given an undirected graph, a natural random walk can be defined by the transition probability $p : V \times V \to \mathbb{R}^+$ such that $p(a, b) = w(a, b)/d(a)$ for all $(a, b) \in E$, where $d(a) = \sum_b w(a, b)$. If the edges have directions, then p is defined by $p(a, b) = w(a, b)/d^+(a)$ for all $(a, b) \in E$ and 0 otherwise, where $d^+(u) = \sum_{u\to v} w(u, v)$. The random walk on a strongly connected and aperiodic graph has unique stationary distribution π that satisfies the balance equation $\pi p = \pi$.

5.3 Marginalized Random Walks on a Subgraph

We can model many versions of graph based inference problems as learning on an induced subgraph. Typical learning tasks in this setting are unsupervised and semi-supervised learning on a target subset, where one would like to utilize not only the original structure of the subgraph, but also the global structure and the interactions between the subgraph and its complement. To propagate the information needed to perform these tasks, the graph based approach depends upon a random walk model to communicate the relevant information globally throughout the graph. In the case where the inference problem is to be localized on a focused subset of the graph, we need a new random walk model that communicates the sources of relevant information to the subset. With an appropriate *marginalized* random walk model, we can then derive principled techniques for various learning problems on a heterogeneous network.

Given a graph $G = (V, E)$ (either directed or undirected), and a subset of vertices

$A \subset V$, we are interested in performing a learning task in A, e.g., learning a classification of A's vertices. Let A^c denote the complement of A. For example, in the blog example where A is the set of blog pages we want to classify based on topics, A^c is the set of non-blog Web pages that have connections to the blog pages. In the example of a tripartite graph for a citation network including papers, authors and conferences, A is the set of papers and A^c includes all the authors of the papers and the conferences.

Typically, the transition probability P of a natural random walk model on the graph can be written as in Section 5.2. Here one can equivalently rewrite the transition probability in a blockwise form with respect to A and A^c

$$P = \begin{pmatrix} P_{AA} & P_{AA^c} \\ P_{A^cA} & P_{A^cA^c} \end{pmatrix}$$

where P_{AA^c} denotes the transition probability between vertices in A and A^c, etc.

One could attempt to perform classification in A based only on P_{AA}, by applying the framework in Chapter 3. However this ignores the information that connects A and A^c, which could be significant. A extreme case is that when we have no interactive relationships in either A or A^c but only P_{AA^c} and P_{A^cA}; that is, a bipartite graph (when edges between A and A^c are undirected). We will see later in Section 5.3.2 that co-clustering methods on bipartite graphs actually utilize P_{AA^c} and P_{A^cA} in an undirected case. Now my goal is to define a new random walk in A incorporating all relevant information.

Given a vertex u in A, I first assume it has outlinks to a vertex v in A and a vertex v_c in A^c. The random walk has the following two options starting from u: it can follow the outlink to v (and so stay within A), or to v_c (and so leave A). The two walking options result in two transition probabilities: P_{in} and P_{out}.

P_{in} is the probability If the random surfer stays in A, which equals the probability P_{AA}.

If the random surfer jumps out of A to A^c, its walk will follow the transition probability P_{AA^c}. Once it enters A^c, there is a non-zero chance it will take any number of steps in A^c before possibly returning to A. Therefore, we have the following definition for P_{out}.

Definition 5.3.1. *Let P_{out} be the transition probability between u and v in A, such that the surfer re-entered A after transiting from A to A^c and back to A. Define P_{out} as*

$$P_{out} = P_{AA^c}\left(I + \sum_{i=1}^{n\to\infty} P_{A^cA^c}^i\right) P_{A^cA} = P_{AA^c}(I - P_{A^cA^c})^{-1} P_{A^cA}$$

Note that P_{out} considers any kind of transition that may occur in A^c, so that the random walk marginalizes the relational information in A^c.

Combining the two transition models P_{in} and P_{out} yields a new random walk P^*_{AA} on the *subgraph* A.

Definition 5.3.2. *Define the new marginalized random walk* P^*_{AA} *on the* subgraph A *as*

$$P^*_{AA} = P_{in} + P_{out}$$

P^*_{AA} is the new transition probability on A by marginalizing the random walk on subset A, taking all sources of information into account. The similarity among vertices in A is measured by a combination of the transition probability within A, P_{in}, and the probability of escaping from A to A^c and then returning to A, P_{out}.

To ensure P_{out} and P^*_{AA} are well defined, we assume P is ergodic. We then have the following propositions.

Proposition 5.3.3. $I - P_{A^cA^c}$ *is invertible.*

Proof. Assume $I - P_{A^cA^c}$ is singular. Then $(I - P_{A^cA^c})x = 0$ has a non-trivial solution $x = P_{A^cA^c}x$. Taking norms, we have $\|x\| = \|P_{A^cA^c}x\| \leq \|P_{A^cA^c}\| \, \|x\| < \|x\|$. The last inequality follows because the row sum of $P_{A^cA^c}$ is less than 1. Contradiction. □

Proposition 5.3.4. P^*_{AA} *is a valid transition probability; i.e. the sum of each row equals* 1.

Proof. Consider the ways a random surfer can start from a vertex u in A and return to another vertex v in A. In the first step, u has two choices, either follow links in A or jump out of A to A^c. If it stays in A, the transition probability is P_{in}. If it jumps out of A, then the surfer has an infinite number of paths lenghts that stay in A^c, before (possibly) returning to A. Here, P_{out} is the probability of transiting from u to v via A^c and P_{in} is the transition probability from u to v without entering A^c. Thus the sum of these two disjoint transition probabilities is a valid transition probability. □

5.3.1 Learning on a Subgraph

It is natural to utilize the framework (3.20) on directed graphs to produce graph based algorithms for unsupervised, semi-supervised and ranking on complex networks G, targeting on subset A:

$$f^* = \arg\min_{f \in \mathbb{R}^{|A|}} \Omega(f) \ + \ \mu \, ||f - y||^2$$

where $\Omega(f)$ is defined as in (3.15) while using the new marginalized random walk P^*_{AA}. Here $y = \langle y_i \rangle$ is the partially labeled vector; where each labeled data is either 1 or -1, and $y_i = 0$ for each unlabeled data point. For ranking, I label the *root* data as 1 and the rest as 0. Also, μ is a tuning parameter; where for clustering tasks I set $\mu = 0$ since we do not have any label information.

5.3.2 Special Case: Learning with a Bipartite Graph

Now I show that the original spectral co-clustering by Zha et al. (2001) and Dhillon (2001) on a bipartite graph can be equivalently interpreted as defining new random walk models on each subset of the bipartite graph in my scheme. This equivalence validates the generalization of the new proposed method.

Given a bipartite graph $G = (\langle A, B \rangle, E)$, where A and B are disjoint subsets of vertices, the transition probability P over G has the following blockwise form

$$P \ = \ \begin{pmatrix} 0 & P_{AB} \\ P_{BA} & 0 \end{pmatrix}$$

Thus, I can define new marginalized random walk in A and B as

$$P^A = P_{AB} P_{BA}, \tag{5.1}$$

$$P^B = P_{BA} P_{AB} \tag{5.2}$$

Intuitively, such random walks can be also understood as a two step random walk that I discussed before in Chapter 3 First consider the random walk among vertices in A (B will be isomorphic). If the random surfer is currently at vertex $a_i \in A$, it first takes a backward step along edge (a_i, b) to some vertex $b \in B$. Then if b also has an edge connected to a_j, the surfer will visit a_j along the edge (b, a_j).

The two-step transition probability $p^A(a_i, a_j)$ is determined by the surfer taking one backward step and one forward step. Therefore,

$$p^A(a_i, a_j) = \sum_b p(a_i, b)p(b, a_j) = \sum_b \frac{w(a_i, b)w(b, a_j)}{d(a_i)d(b)} \tag{5.3}$$

which is exactly the same as the P^A obtained in (5.1).

Proposition 5.3.5. *The stationary distribution π^A of the random walk defined by P^A is*

$$\pi^A(a) \;=\; \frac{d(a)}{\operatorname{vol} G_A} \tag{5.4}$$

where $\operatorname{vol} G_A = \sum_{a \in A} d(a)$.

The proof is the same as the one for two-step random walks in Proposition 3.2.5

Similarly, one can define the two step transition process among nodes in B, yielding the transition probability

$$p^B(b_i, b_j) = \sum_a p(b_i, a)p(a, b_j) = \sum_a \frac{w(b_i, a)w(a, b_j)}{d(b_i)d(a)} \tag{5.5}$$

which corresponds to (5.2). Moreover, the stationary distribution π^B is

$$\pi^B(b) = \frac{d(b)}{\operatorname{vol} G_B} \tag{5.6}$$

To obtain unsupervised and semi-supervised results on both subsets simultaneously, I define a smoothness function f over A from Equation (3.15) that is measured by

$$\Omega_A(f) = \frac{1}{2} \sum_{a_i, a_j} P^A(a_i, a_j)\pi(a_i) \left(\frac{f(a_i)}{\sqrt{\pi(a_i)}} - \frac{f(a_j)}{\sqrt{\pi(a_j)}} \right)^2$$

Similarly, the smoothness function g over B is defined as

$$\Omega_B(g) = \frac{1}{2} \sum_{b_i, b_j} P^B(b_i, b_j)\pi(b_i) \left(\frac{g(b_i)}{\sqrt{\pi(b_i)}} - \frac{g(b_j)}{\sqrt{\pi(b_j)}} \right)^2$$

Proposition 5.3.6. *Let W be the weight matrix between A and B. Define $D_A = We$, $D_B = W^T e$. Then $\Omega_A(f)$ and $\Omega_B(f)$ satisfy*

$$\Omega_A(f) = \frac{1}{\operatorname{vol} G_A} f^T \Delta_A f$$

and

$$\Omega_B(g) = \frac{1}{\operatorname{vol} G_B} g^T \Delta_B g$$

where

$$\begin{aligned} \Omega_A &= I - D_A^{-1/2} W^T D_B^{-1} W D_A^{-1/2} = I - MM^T \\ \Omega_B &= I - D_B^{-1/2} W D_A^{-1} W^T D_B^{-1/2} = I - M^T M \end{aligned}$$

and $M = D_A^{-1/2} W^T D_B^{-1/2}$.

The proof is obtained simply by applying (5.3), (5.4), (5.5) and (5.6) to the left sides of $\Omega_A(f)$ and $\Omega_B(f)$, which quickly yields the equality.

It is not hard see that the solutions for f and g by minimizing $\Omega_A(f)$ and $\Omega_B(f)$ are the eigenvectors of MM^T and $M^T M$ with second largest eigenvalues.

It is known the solution of spectral co-clustering on A and B is the second largest left and right singular vectors of M (Zha et al., 2001; Dhillon, 2001). It is easy to see that from the singular value decomposition, that the non-zero left singular eigenvalues of M are the square roots of the non-zero eigenvalues of MM^T with the same eigenvector space. The eigenvector space of M's right eigenvectors is the same as the one of $M^T M$. Therefore, the two solutions are exactly the same, but with different motivations.

The advantage of having marginalized random walk models on each subset is that we can treat each set individually while using their mutual relationships. As expected, the solution is exactly the same as when we considered the combinatorial cut problem in bipartite graphs. In spectral co-clustering method, the goal is to define a cut criterion for the weight matrix that minimizes the cut over the unmatched edges and maximizes the matched vertices in the subgraphs. Such cuts naturally partition the bipartite graph into two parts in each set. The solution is not clear though if we want different number of partitions on each subset. While using the new scheme, we can obtain k-cluster results using the first k eigenvectors of Δ_A and Δ_B. Moreover, as discussed in Section 5.3, this

method can be easily generalized into more complex graphs, which would have been difficult from graph cut perspective.

5.4 Evaluation

I demonstrate several problem settings that involve data represented in complex graph structures. I evaluate the information marginalization approach by applying it to two datasets: WebKB and CiteSeer, to solve these problems.

5.4.1 Web Classification

The first dataset is from WebKB (`www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data`), which includes pages from four universities: Cornell, Texas, Washington and Wisconsin. After removing isolated pages, the Web pages have been manually classified into seven categories: *student, faculty, staff, department, course, project* and *other*. I take advantage of the link structure and page-word relationships for the following two learning tasks.

(1a) Given the link structure of all the pages and the words used in them, discriminate student (course) pages from non-student (non-course) pages. Here, A corresponds to the web pages, and A^c to the words. See Figure 5.1.

(1b) Given only the link structure, discriminate student pages (labeled as 1) from course pages (labeled as -1). For this task, A corresponds the pages of students and courses, and A^c to the web pages from other classes.

I compare the performance of two algorithms for Web page classification in transductive setting. The first transductive algorithm uses the marginalized random walk P^*, and the second one uses hyperlink structure P_{AA} only. I use canonical 0-1 weights over the directed hyperlinks. I set the tuning parameter $\mu = 2.5$ for both algorithms. I increase the size of the labeled data sample at each iteration. The comparison is based on 0/1 classification error, averaged by 20 iterations.

Figures 2 and 3 show the comparison results for problem (1a), and Figure 5.4, for problem (1b). It is clear that the methods using information marginalization outperforms the one with only the local hyperlink information from subset. Specifically, this implies

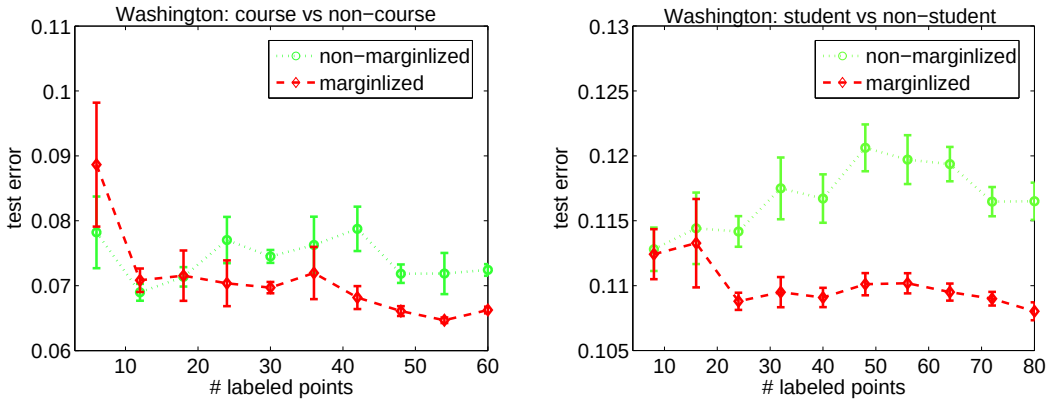


Figure 5.2: Classification error on discriminating course pages from non-course pages (left) and student pages from non-student pages (right) from Washington.

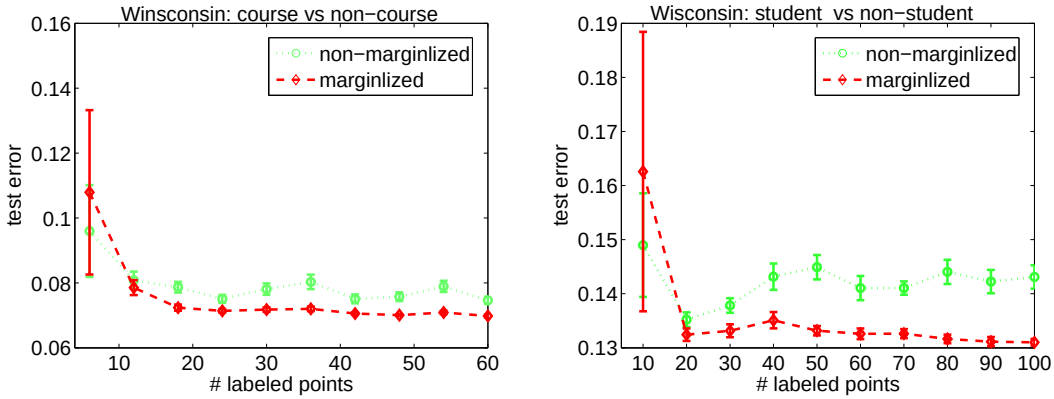


Figure 5.3: Classification error on discriminating course pages from non-course pages (left) and student pages from non-student pages (right) from Wisconsin.

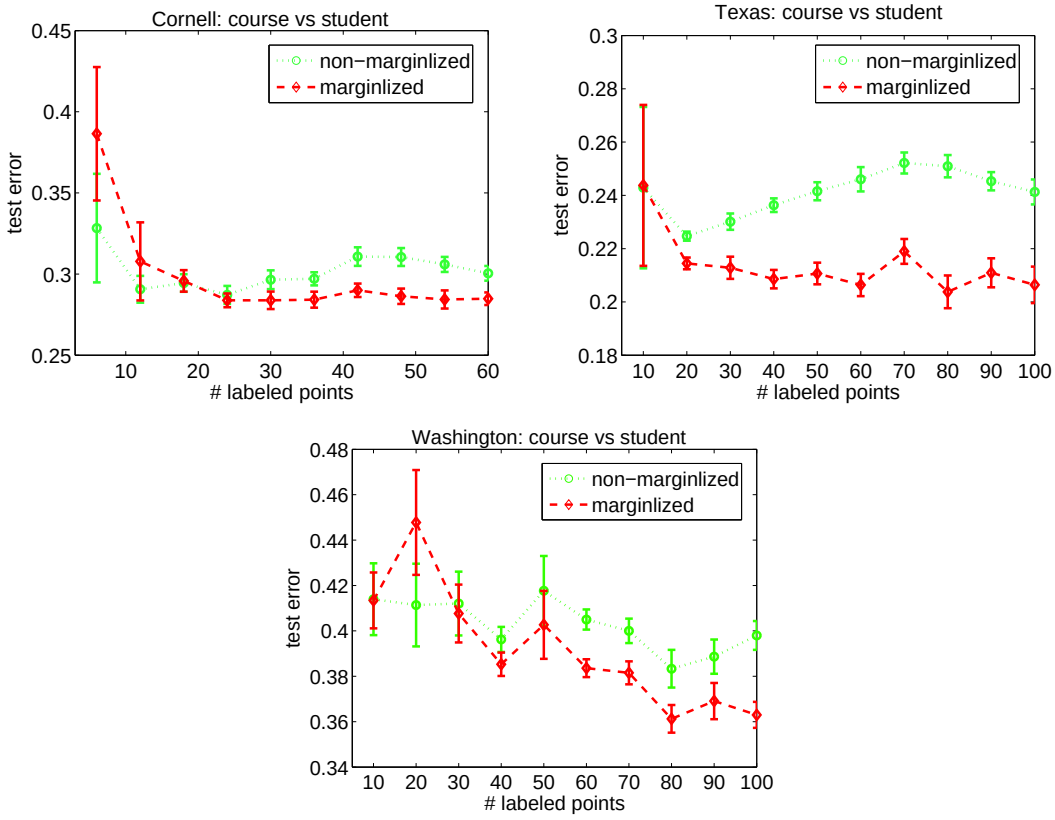


Figure 5.4: Classification error on discriminating course pages from student pages.

that the marginalized random walk is able to convey more global information onto the subset, efficiently improving the performance in classifications.

5.4.2 Ranking in Citation Networks

The second dataset is based on CiteSeer (`citeseer.ist.psu.edu`)—a well-known scientific digital library that catalogues primarily computer and information science literature. I construct citation networks based on paper-paper and paper-author relationships from CiteSeer. I extract a set of papers P with authors U. Here, I focus on two kinds of ranking problems.

(2a) Given some papers (i.e., *seed* papers) in P labeled as relevant to a specific topic

T, rank the rest of the papers based on their relevance to T. Here, A is P, A^c is U.

(2b) Given some authors (i.e., *seed* authors) in A identified as relevant since they share similar research interests, rank the remaining authors based on how much they share the research interests with these seed authors. A is U, A^c is P.

To build citation networks, I scout ahead following the paper citation and corresponding authors information from the OAI records (`citeseer.ist.psu.edu/oai.html`). I start a crawl from a set of pre-selected authors (i.e., *root authors*), then collect all their papers and the co-authors of these papers. The co-authors are added to a growing set of authors that is used in the next iteration. I repeat this iteration $n = 3$ times to collect a number of related authors and papers. In my experiment, I choose the root authors from two different areas:

Root authors	# Authors	# Papers
"Berhard Scholkopf" + "John Kleinberg"	7156	4979
"Vladimir Vapnik" + "Jianbo Shi"	3048	2097

Therefore, the citation network contains authors with different research subjects, which is more realistic.

For problem (2a), Table 5.1 shows the top 20 results of paper ranking with respect to the labeled paper "Kernel Principal Component Analysis"; and Table 5.2 shows the top 10 papers ranked with respect to "Authoritative Sources in a Hyperlinked Environment". We can see that the information maginalization method works better than only using citation links information as the highly ranked papers are closer to the labeled paper in information marginalization scheme. If one only considers citation links, some papers from slightly different domain may be included in the top ranking list because they may have citations with similar papers. With the help of author-paper relationships, the relationship between the labeled paper and other papers become more clear thus lead more accurate ranking results.

For problem (2b),Table 5.3 lists the ranking results of authors with respect to Vladimir Vapnik in the second citation network. The information from the citation links moves some authors—Chris Burges, Bernhard Scholkopf, Olivier Chapelle and Alex Smola—to higher ranking positions than only using author-paper relationships. The reason is that these authors also have many citation links among their papers that strengthen the similarities

Table 5.1: Papers Ranked closest to "Kernel Principal Component Analysis"

	marginalized random walk		**use only citation links**
	title		title
1.	Regression Estimation with Support Vector Learning Machines	1.	Model Selection for Support Vector Machines
2.	Model Selection for Support Vector Machines	2.	SV Estimation of a Distribution's Support
3.	Support Vector Method for Novelty Detection	3.	Support Vector Method for Novelty Detection
4.	A Generalized Representer Theorem	4.	Optimal Hyperplane Classifier with Adaptive Norm
5.	Optimal Hyperplane Classifier with Adaptive Norm	5.	Inclusional Theories in Declarative Programming
6.	Incorporating Invariances in Support Vector Learning Machines	6.	Studies on the Formal Semantics of Pictures
7.	Latent Semantic Kernels	7.	A Noise-Tolerant Hybrid Model of a Global and a Local Learning Module
8.	Sparse Kernel Feature Analysis	8.	Latent Semantic Kernels
9.	Extracting Support Data for a Given Task	9.	Incorporating Invariances in Support Vector Learning Machines
10.	Support-Vector Networks	10.	A Generalized Representer Theorem
11.	Kernel Methods: A Survey of Current Techniques	11.	Equivalent Conditions for the Solvability of Nonstandard LQ-Problems with Applications to Partial Differential Equations with Continuous Input-Output Solution Map
12.	A Training Algorithm for Optimal Margin Classifiers	12.	Hyperbolic Conservation Laws with a Moving Source
13.	Improving the Accuracy and Speed of Support Vector Machines	13.	Extracting Support Data for a Given Task
14.	The Connection between Regularization Operators and Support Vector Kernels	14.	Support-Vector Networks
15.	Generalization Performance of Regularization Networks and Support Vector Machines	15.	On Molecular Approximation Algorithms for NP Optimization Problems
16.	Statistical Learning and Kernel Methods	16.	Kernel Methods:A Survey of Current Techniques
17.	The Kernel Trick for Distances	17.	CPU Management for UNIX-based MPEG Video Applications
18.	On a Kernel-based Method for Pattern "Recognition," "Regression," "Approximation"	18.	Efficient Lossless Compression of Trees and Graphs
19.	Advances in Kernel Methods - Support Vector Learning	19.	A Precise Semantics For Vague Diagrams
20.	Estimating the Support of a High-Dimensional Distribution	20.	Redescription, Information And Access

Table 5.2: Papers Ranked closest to "Authoritative Sources in a Hyperlinked Environment"

	marginalized random walk		**use only citation links**
	title		title
1.	Fast Monte-Carlo Algorithms for finding low-rank approximations	1.	volutionary Strategies For Solving Frustrated Problems
2.	Evolutionary Strategies For Solving Frustrated Problems	2.	Fast Monte-Carlo Algorithms for finding low-rank approximations
3.	The Anatomy of a Large-Scale Hypertextual Web Search Engine	3.	Reconstruction From The Multi-Component Am-Fm Image
4.	Latent Semantic Indexing: A Probabilistic Analysis	4.	The Anatomy of a Large-Scale Hypertextual Web Search Engine
5.	Challenges in Web Search Engines	5.	Latent Semantic Indexing: A Probabilistic Analysis
6.	How to Personalize the Web	6.	Learning Decision Strategies with Genetic Algorithms
7.	Efficient and Effective Metasearch for Text Databases Incorporating Linkages among Documents	7.	A Model for Sequence Databases
8.	The PageRank Citation Ranking: Bringing Order to the Web	8.	Semantically Driven Automatic Hyperlinking
9.	New Results for Online Page Replication	9.	Applications of a Web Query Language
10.	Searching the Web: General and Scientific Information Access	10.	Efficient and Effective Metasearch for Text Databases Incorporating Linkages among Documents

Table 5.3: Author ranking result in network 2.

marginalized	**only author-paper relationships**	**marginalized**	**only author-paper relationships**
name	name	name	name
1.Chris Burges	1.Sayan Mukherjee	11.Mark Stitson	11.Vladimir Vovk
2.Bernhard E.Boser	2.Chris Burges	12.Alex Gammerman	12.Alex Gammerman
3.Isabelle M. Guyon	3.Bernhard E. Boser	13.Vladimir Vovk	13.Mark Stitson
4.Sayan Mukherjee	4.Isabelle M.Guyon	14.Chris Watkins	14.Klaus-Robert Muller
5.Donghui Wu	5.Donghui Wu	15.Partha Niyogi	15.Federico Girosi
6.Bernhard Scholkopf	6.Steven E.Golowich	16.Olivier Chapelle	16.Koh.Sung
7.Heinrich H.Bulthoff	7.Volker Blanz	17.Alex Smola	17.Partha Niyogi
8.Thomas Vetter	8.Bernhard Scholkopf	18.Adnan Aziz	18.Jason Weston
9.Volker Blanz	9.Thomas Vetter	19.Jason Weston	19.Olivier Chapelle
10.Steven Golowich	10.Chris Watkins	20.Koh.Sung	20.Alex Smola

with respect to the labeled author.

5.5 Summary

I propose a unified mechanism for incorporating information from multiple object types and relations in a heterogeneous network when making inferences on a targeted subgraph. The new mechanism allows me to only focus on the subgraph as the information has been marginalized onto the subgraph. The marginalization is achieved via a valid composition of random walk models that propagate information both externally and internally. Moreover, interestingly, this new method generalizes the special case in bipartite spectral clustering. If we take one vertex set in the bipartite graph to be a hyperedge, then it is not hard to see that this method also generalizes previous hyperspectral clustering in Chapter 4.

I quantify the performance of my new schemes on two real world relational data and achieve good results in challenging inference problems.

Chapter 6

Learning under Distribution Shifting with Unlabeled Data

So far, the previous chapters have discussed how to solve unsupervised and semi-supervised learning problems with data embedded in different types of graph structures. In this chapter, I switch to a distribution based approach and also shift from discrete to continuous domains, to address these problems. In the distribution based approach, the learned functions are defined on a continuous domain $\mathcal{X}$, encoding the instances of the learner's world, with a distribution $\Pr(x)$.[1] Additionally, an instance of $\mathcal{X}$ determines a probability distribution on an output space $\mathcal{Y}$. The labeled examples $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ are independently and identically (iid) drawn from a fixed target distribution $\Pr(x, y)$ over $\mathcal{X} \times \mathcal{Y}$. The learned function $f : \mathcal{X} \to \mathcal{Y}$ is constructed to predict Y from X.

One of the major limitations of current semi-supervised learning methods in the distribution based approach is that there is no straightforward way for these methods to make predictions on test points that are not drawn from $\Pr(x)$. That is, most current methods experience difficulty when the distributions between training to test data are different. This distribution shifting may come from a censoring mechanism that has control over the assignment of labels to data points. Such test points from different distribution could be harmful as they provide misleading information about $\Pr(x)$. Therefore for semi-supervised learning, the test points have to be drawn from $\Pr(x)$, or some distribution closely related

[1] Throughout this chapter I will use Pr to define a probability distribution on $\mathcal{X}$.

to $\Pr(x)$ (Chapelle et al., 2006). This is an underlying assumption for most semi-supervised learning methods, as well as for supervised learning.

Therefore, a natural interesting question is raised: Is it possible to make predictions on the test points that are not drawn from $\Pr(x)$, or some distribution that has been skewed or shifted from $\Pr(x)$? The contribution of this chapter is that it presents a method that is able to resolve this challenging situation by using unlabeled data. Most algorithms for this setting in literature try to first recover sampling distributions and then make appropriate corrections based on the distribution estimate. The method I present is a novel nonparametric method that directly produces resampling weights to correct the distribution bias effects by matching distributions between training and testing sets in a feature space, bypassing the problem of explicit density estimation. The technique can be easily applied to many different supervised learning algorithms, automatically adapting their behavior to cope with distribution shifting between training and test data. Experimental results demonstrate that the new method works well in practice.

6.1 Learning under Distribution Shifting

This section covers the background in learning with distribution shifting. I first give the motivation by illustrating a toy example and discussing some real world phenomenon. Then I briefly review some related methods for this problem in literature. Finally I highlight the differences and advantages of my new approach from existing ones.

The default assumption in many classical learning scenarios is that training and test data are drawn iid from the *same* distribution. When the distributions on training and test set do not match, we are facing *sample selection bias* which I referred to as *distribution shifting*. In this case, given a domain of patterns $\mathcal{X}$ and labels $\mathcal{Y}$, the training samples $Z = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathcal{X} \times \mathcal{Y}$ we obtained are from a probability distribution $\Pr(x, y)$, and test samples $Z' = \{(x'_1, y'_1), \ldots, (x'_{m'}, y'_{m'})\} \subseteq \mathcal{X} \times \mathcal{Y}$ are drawn from another distribution $\Pr'(x, y)$. This is a difficult situation for standard statistical learning approaches. It is not hard to imagine that it will result a biased function with potentially poor performance in learning if we do not correct the shifting.

6.1.1 Motivation

I first illustrate the phenomenon of learning under distribution shifting via a toy example.

Toy regression example The toy example is to demonstrate the effect of learning functions with and without bias correction on data of different training and test distributions. The data is generated according to the polynomial regression example from Shimodaira (2000), for which $q_0 \sim \mathcal{N}(0.5, 0.5^2)$ and $q_1 \sim \mathcal{N}(0, 0.3^2)$ are two normal distributions. The observations are generated according to $y = -x + x^3$, and are observed in Gaussian noise with standard deviation 0.3 (see Figure 6.5.1; the blue curve is the noise-free signal).

I sample 100 training (blue circles) and testing (red circles) points from q_0 and q_1 respectively. I attempt to model the observations with a degree 1 polynomial. The red line is directly derived only from the training data via ordinary least squared (OLS), and predicts the test data very poorly as the test data has different distribution of the training data. The black dashed line is the optimal function which is fitted using OLS on the red test points. The other three dashed lines are obtained by fitting on the labeled test points with bias correction in OLS (I will present each of the methods later). We can see that all these methods give much better performance to the red one by taking the bias correction into account.

Real World Problems Although there are some work addressing the learning problem with different training and test distributions, distribution shifting is typically ignored in standard learning algorithms. Nonetheless, in reality the problem occurs rather frequently: While the available data have been collected in a biased manner, the test is usually performed over a more general target population. Below, I give three examples; but similar situations may occur in many other domains.

1. Suppose we wish to generate a model to diagnose breast cancer. Suppose, moreover, that most women who participate in the breast screening test are middle-aged and likely to have attended the screening in the preceding three years. Consequently our sample includes mostly older women and those who have low risk of breast cancer because they have been tested before. The examples do not reflect the general

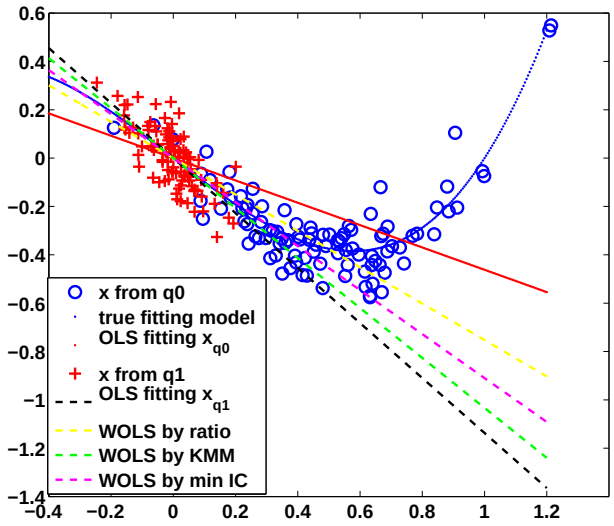


Figure 6.1: Four dashed lines Polynomial models of degree 1 fit with OLS and WOLS with bias correction; Labels are *Ratio* for ratio of test to training density; KMM for our approach; *min IC* for the approach of Shimodaira (2000); and *OLS* for the model trained on the labeled test points.

population with respect to age (which amounts to a bias in $\Pr(x)$) and they only contain very few diseased cases (i.e. a bias in $\Pr(y|x)$).

2. Gene expression profiles are commonly used in tumor diagnosis. A typical problem is that the samples are obtained using certain protocols, microarray platforms and analysis techniques. And they typically have relatively small sample sizes. The test cases are recorded under different conditions, resulting in a different distribution of gene expression values.

3. Consider performing data analysis using a Brain Computer Interface where the distribution over incoming signals is known to change as experiments go on, because the subjects get tired, the sensor setup changes, and so on. In this case it is necessary to adapt the estimator to the patterns with the new distribution in order to improve performance.

The generalization of the learned model in supervised learning can be significantly defected if we do not have enough or the wrong information of $\Pr(x)$ or $\Pr(y|x)$ in the above problems.

Similar to supervised learning, semi-supervised learning methods will also face problems with distribution shifting. The reason is that generally, semi-supervised learning takes the test distribution into account implicitly, where the unlabeled data provides useful information of $\Pr(x)$, e.g., Chapelle et al. (2003) designs kernels to find decision boundary based on the density from unlabeled data. When the unlabeled data provide misleading or less useful information, traditional semi-supervised methods will fail. For example, the model using generative methods in semi-supervised learning may be misspecified by the unlabeled data since maximum likelihood tries to model $\Pr(x)$ rather than $\Pr(y|x)$ (Cozman et al., 2002). The biased estimate from misleading unlabeled sample with inconsistent distribution may lead to dramatic error to the method by Szummer and Jaakkola (2002), as it adds information regularization on $\Pr(x)$ where $\Pr(x)$ is obtained from an empirical estimate obtained from the unlabeled sample.

While the unlabeled data might be harmful in learning, it will do good for us if we pay attention to the phnomenon and appropriately correct the bias in our learning procedures.

6.1.2 Problem Overview

In the following, I list some methods that are mostly related to my work in solving sample selection bias problem. Some of them achieved better performance by using unlabeled data. I will also investigate the problem of how to utilize unlabeled data in this distribution shifting scenario later. One may refer to (Chawla and Karakoulas, 2005) for more literature review.

Heckman's method (Heckman, 1979) Heckman studied the sample selection bias for modeling labor supply in the field of Econometrics, which is a Nobel-prize winning work in 2000. He developed a procedure for correcting sample selection bias by estimating the probability that an observation is selected into the training set to correct the linear regression model. Heckman's sample selection model consists of a linear regression model and a binary probit selection model. The method contains two steps. The first step is to

use the selection model to explicitly model the censoring mechanism and correct for bias. The second step is to generate a regression model only for data that satisfies the selection equation. Specifically, the two models associated with a random sample of observations are

$$Y_1 = \beta_1^T X_1 + u_1$$

$$Y_2 = \beta_2^T X_2 + u_2$$

where u_1, u_2 are from normal distribution and they form a joint distribution with correlation ρ as $u_1, u_2 \sim N[0, 0, \sigma^2, \rho]$. Y_2 indicate whether the data is labeled or not. Data is labeled if $y_2 = 1$, and not labeled if $y_2 = 0$. The information from Y_2 will be applied to Y_1 where y_1 has the observed value if $y_2 > 0$ and y_1 is missing if $y_2 \leq 0$. The estimate of β_1 will be unbiased if u_1 and u_2 are uncorrelated and therefore the data on Y_1 are missing randomly. In the biased case, the regression will be effected based on ρ and σ. Then the conditional regression function for selected samples can be written as

$$E(Y_1|X_1, Y_2 \geq 0) = \beta_1^T X_1 + E(u_1|u_2 \geq -\beta_2^T X_2)$$

where $E(u_1|u_2 \geq -\beta_2^T X_2)$ is estimated by assuming u_1, u_2 has a bivariate normal distribution.

By recovering the censoring mechanism, the bias is filtered out in the regression model. Heckman's procedure requires a regression based model that is commonly used in Econometrics. However, we may interested in other types of learning models that handle the problem.

Weighting the log-likelihood (Shimodaira, 2000) Shimodaira (2000) proposed a weighting technique to reweight the observed samples in maximizing the log-likelihood (MLE) function. The original maximum likelihood may be poorly estimated due to the distribution shifting and results in misspecification of the model. Consider the loss

$$\text{loss}(\theta) = -\int \text{Pr}^*(\text{x}) \int \text{Pr}(\text{y}|\text{x}) \log \text{Pr}(\text{y}|\text{x}, \theta) \text{dydx}$$

where $\text{Pr}^*(x)$ is the density of training ($\text{Pr}(x)$) or the test data ($\text{Pr}'(x)$). If the densities are not consistent, then the estimated loss from the training data will not match the one

that should come from the test data. Therefore, MLE does not provide a good inference in this case. To achieve better performance, the author proposed to apply weights $w(x) = \Pr'(x)/\Pr(x)$ motivated from importance sampling. Therefore, the log-likelihood function becomes

$$L_w(\theta|x,y) = -\sum_{i=1}^{n}(-w(x_i)\log\Pr(y_i|x_i,\theta))$$

The weights are estimated by minimizing an information criterion IC_w. The information criterion is an estimate of the expected loss unbiased up to $O(n^{-1})$ term. The search for optimal weights has high computational cost thus the author proposed to do line search with parameter λ as

$$w(x) = \left(\frac{\Pr'(x)}{\Pr(x)}\right)^{\lambda}, \quad \lambda \in [0,1]$$

Shimodaira (2000) numerically finds a $\hat{\lambda}$ that minimizes IC_w by searching over $[0,1]$, which is a kind of heuristic for finding the optimal $w(x)$. Specifically, for normal linear regression, the information criterion can be calculated in a simpler form that assumes the residual of the regression model is distributed normally and the distribution parameters are known. For more complicated models, this approach requires taking first and second derivatives of $\log\Pr(y|x\theta)$ and L_w with respect of θ, which is not easy to calculate in general.

Shimodaira (2000) provides an important intuition for bias correction by re-weighting the log-likelihood function. I will also investigate this re-wighting mechanism and compare with this method to my approach later.

Sample selection bias in machine learning (Zadrozny, 2004) Zadrozny (2004) formalized the sample selection bias problem in machine learning and studied analytically how learned classifiers are affected by it. The paper considers four cases selection bias. Define examples (x, y, s) drawn i.i.d. from a distribution with domain $\mathcal{X} \times \mathcal{Y} \times \mathcal{S}$, where $\mathcal{S}$ is a binary space indicating whether the data is selected as a training input or not: $s = 1$ means the data is selected and $s = 0$ means not selected. The four cases of denpendece of s on the sample are

1. s is independent of x and y, which indicates that the selected sample is not biased.

2. $\Pr(s|x,y) = \Pr(s|y)$, s is independent of y given x. This indicates that the bias depends on feature x.

3. $\Pr(s|x,y) = \Pr(s|x)$, s is independent of x given y. This indicates that the bias depends on label y.

4. No independence assumption between x, y and s.

Zadrozny (2004) focused on Case (2), as it occurs more frequently in real world datasets. The paper experimentally shows how the sample selection bias affects different types of classification methods, including Bayesian classifiers, logistic regression, SVM and decision trees. In addition, it presents a bias correction method that can be applied to any classifier learner. However, the model for the selection probabilities $\Pr(s=1|x)$ has to be known, which is not typically true in real world problems. The method also applies a re-wighting approach, that tries to minimizing the expected value of a loss function over the distribution of training examples; that is, it minimizes $E_{x,y\sim\Pr'}(loss(x,y,\theta))$. Let Pr be another distribution such that $Pr(x,y,s) = P(s=1)\frac{\Pr'(x,y,s)}{\Pr(s=1|x)}$, then we have

$$E_{x,y\sim\Pr'} loss(x,y,\theta) = E_{x,y\sim\Pr}(loss(x,y,\theta)|s=1)$$

The proof is based on a similar idea of importance sampling as being used in (Shimodaira, 2000) where the weight is $\frac{P(s=1)}{P(s=1|x)}$. However, the question of how to obtain a correct selection probability $\Pr(s=1|x)$ to infer the weights remains unsolved in this paper.

Inferring label sampling mechanisms in semi-supervised learning (Rosset et al., 2005) Rosset et al. (2005) improved the work of Zadrozny (2004) by proposing a method to infer the sampling model using unlabeled data. Therefore, it is a semi-supervised method. The work establishes a general equality

$$E(f(X,Y,S)) = E(g(X))$$

for any feature function $g(x)$ and $f(x,y,s) = \frac{g(x)}{P(s=1|x,y)}$ if $s=1$ and 0 otherwise. Therefore, the weighting function is $\frac{1}{P(s=1|x,y)}$. This procedure more general than (Zadrozny, 2004). The main goal in (Rosset et al., 2005) is to estimate $w(x,y) = P(s=1|x,y)$ with a "method

of moments" approach that estimates $w(x, y)$ by applying k different representative feature functions $g(x)$ with respect to different matching moments. To solve for the correction weights, it involves solving a least squares optimization problem based on both labeled and unlabeled data. This work proposed a parameterized method to infer the sampling mechanism from unlabeled data, but only solves the problem for a predefined parameterized sampling model. Intuitively, we hope to obtain better weights by matching all possible moments.

Others There are some other related work that I would like to mention briefly. The problem of the distribution shifting is referred to as a nonstandard situation in (Lin et al., 2002). It explained why SVMs are not suitable for nonstandard situations and introduced a simple procedure for adapting SVMs to this caseThe idea of this method is to apply correction weights to the expected hinge loss term in SVM, where the weights are estimated based on the prior probabilities of positive and negative classes in the test population. However, this prior knowledge is not typically known in practice.

Elkan (2001) addresses a similar case when the sampling mechanism is dependent on the class label. Sugiyama and Muller (2005) proposed to reweight the training examples that fall in areas of high density among test examples. Dudik et al. (2005) proposed three bias correction approaches in the problem of maximum entropy density estimation. However, all of them assume the sampling distribution is known.

More recent work can be found in (Ben-David et al., 2006a) and (Storkey and Sugiyama, 2006). In (Ben-David et al., 2006a), the problem is referred to as *domain adaptation*. The training and test sets are samples from the target domain and source domain respectively. This work formalized a bound on the target generalization error of a classifier trained from the source domain without assuming any relationship between labels and the structure of unlabeled data.

Comments As one can see, many existing methods, e.g., in (Zadrozny, 2004; Dudik et al., 2005; Shimodaira, 2000), use re-weighting approaches to solve the sample selection bias problem where the training sample is selected in a biased manner while the test domain targets at a more general population. A common property of these approaches is that the re-weighting idea is adopted from important sampling: the "important" training

observations are emphasized by penalizing the risk more significantly; the biased model estimator is reweighted that ensures the new estimator is unbiased. It is assumed that the support of $\Pr'$ is contained in the support of Pr and the selection probabilities are greater than zero for all observations (the importance weights are the inversion of the selection probabilities in above methods).

In a more general scenario, where the training and test distributions could be arbitrary far apart, the problem is typically unsolvable. It is not hard to imagine that if training and test data are generated from significantly different distributions, there is little hope one could find a function that performs well on both datasets. Therefore in the following we will make certain assumptions as described in Section 6.1.3 to proceed with our methodology.

6.1.3 Contributions

As one can see, there have been several algorithms proposed for solving the sample selection bias problem. Some of these approaches exploit unlabeled data to infer the sampling mechanism. However, one of the main drawbacks in previous work is the requirement that biased densities be explicitly estimated (Zadrozny, 2004; Dudik et al., 2005; Shimodaira, 2000). Some approaches require the class prior to be known in advance (Lin et al., 2002).

I also attempt to utilize the availability of unlabeled data to direct a sample selection de-biasing procedure based on a re-weighting approach. However, unlike previous work, I infer the resampling weight *directly*, by distribution matching between training and testing sets in feature space. The method does not require parametric distributional assumptions; rather, I account for the difference between $\Pr(x, y)$ and $\Pr'(x, y)$ by reweighting the training points such that the means of the training and test points in a reproducing kernel Hilbert space (RKHS) are close. I refer to this re-weighting process kernel mean matching (KMM).

The required optimization is a simple QP problem. The reweighted sample can be straightforwardly incorporated into a variety of regression and classification algorithms. I apply the method to a variety of regression and classification benchmarks, as well as to classification of microarrays from prostate and breast cancer patients. These experiments demonstrate that KMM greatly improves learning performance compared with training on unweighted data, and that the reweighting scheme can in some cases outperform reweight-

ing using the true sample bias distribution.

Key Assumption 1: In general, the estimation problem with two different distributions $\Pr(x, y)$ and $\Pr'(x, y)$ is unsolvable, as the two terms could be arbitrarily far apart. In particular, for arbitrary $\Pr(y|x)$ and $\Pr'(y|x)$, there is no way we could infer a good estimator based on the training sample when the two distributions are different. Hence we make the simplifying assumption that $\Pr(x, y)$ and $\Pr'(x, y)$ only differ via their marginals on $\mathcal{X}$; that is, $\Pr(x, y) = \Pr(y|x) \Pr(x)$ and $\Pr(y|x) \Pr'(x)$. In other words, the conditional probabilities of $y|x$ remain *unchanged*. (This particular case of sample selection bias has been termed *covariate shift* (Shimodaira, 2000).)

Key Assumption 2: We note that assumption 1 is not sufficient to guarantee that this problem is always solvable. Therefore, assumption 2 additionally assume the marginal distributions are close in the way that first, the ratio of the marginal distribution $\Pr' / \Pr \leq B$; second, the empirically estimated β_i satisfies the inequality (6.10). When the difference is bounded, depend on the assumptions, we can provide a guarantee of the performance, as we will see later in Section 6.4.

6.2 Sample Reweighting

I begin by stating the problem of regularized risk minimization.[2] In general a learning method targets at minimizing the expected risk

$$R[\Pr, \theta, l(x, y, \theta)] = \mathbf{E}_{(x,y)\sim \Pr}\left[l(x, y, \theta)\right] \tag{6.1}$$

of a loss function $l(x, y, \theta)$ depending on a parameter θ. For instance, the loss function could be the negative log-likelihood, $-\log \Pr(y|x, \theta)$, a misclassification loss, or some form of regression loss. However, since typically we only observe examples (x, y) drawn from $\Pr(x, y)$ rather than $\Pr'(x, y)$, one resorts to computing the empirical average

$$R_{\text{emp}}[Z, \theta, l(x, y, \theta)] = \frac{1}{m} \sum_{i=1}^{m} l(x_i, y_i, \theta). \tag{6.2}$$

[2]The thesis Appendix provide some background in classical learning theory.

To avoid overfitting, instead of minimizing R_{emp} directly one often minimizes a regularized variant:

$$R_{\text{reg}}[Z, \theta, l(x, y, \theta)] = R_{\text{emp}}[Z, \theta, l(x, y, \theta)] + \lambda \Omega[\theta]$$

where $\Omega[\theta]$ is a regularizer.

6.2.1 Sample Correction

The problem is more involved if $\Pr(x, y)$ and $\Pr'(x, y)$ are different. The training set is drawn from Pr, however what we would really like is to minimize $R[\Pr', \theta, l]$ as we wish to generalize to test examples drawn from $\Pr'$. An observation from the field of importance sampling is that

$$R[\Pr', \theta, l(x, y, \theta)] = \mathbf{E}_{(x,y)\sim\Pr'}\left[l(x, y, \theta)\right] = \mathbf{E}_{(x,y)\sim\Pr}\Big[\underbrace{\tfrac{\Pr'(x,y)}{\Pr(x,y)}}_{:=\beta(x,y)} l(x, y, \theta)\Big] \tag{6.3}$$

$$= R[\Pr, \theta, \beta(x, y) l(x, y, \theta)], \tag{6.4}$$

provided that the support of $\Pr'$ is contained in the support of Pr. Given $\beta(x, y)$, we can thus compute the risk with respect to $\Pr'$ using Pr, Similarly, we can *estimate* the risk with respect to $\Pr'$ by computing $R_{\text{emp}}[Z, \theta, \beta(x, y) l(x, y, \theta)]$.

The key problem is that the coefficients $\beta(x, y)$ are usually unknown. We need to estimate the coefficients from data. When Pr and $\Pr'$ differ in $\Pr(x)$ and $\Pr'(x)$ only as being stated in the assumption before, we have $\beta(x, y) = \Pr'(x)/\Pr(x)$, where β is a reweighting factor for the training examples. We thus reweight every observation (x, y) such that observations that are under-represented in Pr obtain a higher weight, whereas over-represented cases are downweighted.

Now we could estimate Pr and $\Pr'$ and subsequently compute β based on those estimates. This is closely related to the methods in (Zadrozny, 2004; Lin et al., 2002) as they have to either estimate the selection probabilities or have prior knowledge of the class distributions. Though being intuitive, this approach has two major problems: first, it only works whenever the density estimates for Pr and $\Pr'$ (or potentially, the selection probabilities or class distributions) are good. In particular, small errors in estimating Pr can lead to large coefficients β and consequently to a serious overweighting of the corresponding observations. Second, estimating both densities just for the purpose of computing

reweighting coefficients may be overkill: we may be able to directly estimate the coefficients $\beta_i := \beta(x_i, y_i)$ without having to estimate the two distributions. Furthermore, we can regularize β_i directly with more flexibility, taking prior knowledge into account similar to learning methods for other problems.

6.3 Distribution Matching

6.3.1 Kernel Mean Matching and its Relation to Importance Sampling

Let $\Phi : \mathcal{X} \to \mathcal{F}$ be a feature map into a feature space $\mathcal{F}$ and denote by $\mu : \mathcal{P} \to \mathcal{F}$ the expectation operator

$$\mu(\Pr) := \mathbf{E}_{x \sim \Pr(x)} \left[\Phi(x) \right]. \tag{6.5}$$

Clearly μ is a *linear* operator mapping the space of all probability distributions $\mathcal{P}$ into feature space.

In the following, we will consider universal reproducing kernel Hilbert spaces as defined by Steinwart (2002b).

Definition 6.3.1. *A continuous kernel k on a compact metric space $(\mathcal{X}, d)$ is called universal if the space of all functions induced by k is dense in $C(\mathcal{X})$ where $C(\mathcal{X})$ is the space of continuous bounded functions on $\mathcal{X}$, i.e., for every function $f \in C(\mathcal{X})$ and every $\epsilon > 0$ there exists a function g induced by k with*

$$\|f - g\|_\infty \leq \epsilon$$

Theorem 6.3.2 (Huang et al. (2006b))**.** *The operator μ is* bijective *if $\mathcal{F}$ is an RKHS with a universal kernel $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$.*

Proof. Let $\mathcal{F}$ be a universal RKHS, and let $\mathcal{G}$ be the unit ball in $\mathcal{F}$. We need to prove that $\Pr = \Pr'$ if and only if $\mu(\Pr) = \mu(\Pr')$, or equivalently $\|\mu(\Pr) - \mu(\Pr')\| = 0$. We may

write

$$\begin{aligned}\|\mu(\Pr) - \mu(\Pr')\| &= \sup_{f\in\mathcal{G}} \langle f, \mu(\Pr) - \mu(\Pr')\rangle \\ &= \sup_{f\in\mathcal{G}} \left(\mathbf{E}_{\Pr}[f] - \mathbf{E}_{\Pr'}[f]\right) \\ &=: \Delta\left[\mathcal{G}, \Pr, \Pr'\right].\end{aligned}$$

It is clear that $\Delta\left[\mathcal{G}, \Pr, \Pr'\right] = 0$ is zero if $\Pr = \Pr'$. To prove the converse, we begin with the following result from Dudley (2002, Lemma 9.3.2): If $\Pr, \Pr'$ are two probability mesures defined on a separable metric space $\mathcal{X}$, then $\Pr = \Pr'$ if and only if $\mathbf{E}_{\Pr}[f] = \mathbf{E}_{\Pr'}[f(x')]$ for all $f \in C(\mathcal{X})$, where $C(\mathcal{X})$ is the space of continuous bounded functions on $\mathcal{X}$. If we can show that $\Delta\left[C(\mathcal{X}), \Pr, \Pr'\right] = D$ for some $D > 0$ implies $\Delta\left[\mathcal{G}, \Pr, \Pr'\right] > 0$: this is equivalent to $\Delta\left[\mathcal{G}, \Pr, \Pr'\right] = 0$ implying $\Delta\left[C(\mathcal{X}), \Pr, \Pr'\right] = 0$ (where this last result implies $\Pr = \Pr'$). If $\Delta\left[C(\mathcal{X}), \Pr, \Pr'\right] = D$, then there exists some $\tilde{f} \in C(\mathcal{X})$ for which $\mathbf{E}_{Pr}\left[\tilde{f}\right] - \mathbf{E}_{\Pr'}\left[\tilde{f}\right] \geq D/2$. By definition of universality, $\mathcal{F}$ is dense in $C(\mathcal{X})$ with respect to the L_∞ norm: this means that for all $\epsilon \in (0, D/8)$, we can find some $f^* \in \mathcal{F}$ satisfying $\left\|f^* - \tilde{f}\right\|_\infty < \epsilon$. Thus, we obtain $\left|\mathbf{E}_{\Pr}\left[f^*\right] - \mathbf{E}_{\Pr}\left[\tilde{f}\right]\right| < \epsilon$ and consequently

$$\left|\mathbf{E}_{Pr}\left[f^*\right] - \mathbf{E}_{\Pr'}\left[f^*\right]\right| > \left|\mathbf{E}_{Pr}\left[\tilde{f}\right] - \mathbf{E}_{\Pr'}\left[\tilde{f}\right]\right| - 2\epsilon > \tfrac{D}{2} - 2\tfrac{D}{8} = \tfrac{D}{4} > 0.$$

Finally, using $\|f^*\| < \infty$, we have

$$\left|\mathbf{E}_{Pr}\left[f^*\right] - \mathbf{E}_{\Pr'}\left[f^*\right]\right| / \|f^*\| \geq D/(4\|f^*\|) > 0,$$

and hence $\Delta\left[\mathcal{G}, \Pr, \Pr'\right] > 0$. □

To summerize the main idea of the proof, we know that under the sated conditions, a sufficient condition for $\Pr = \Pr'$ is that for all continuous functions f, we have $\int f d\Pr = \int f d\Pr'$. Such functions f, can be arbitrarily well approximated using functions in a universal RKHS. The intuition comes from that to mimic a distribution one can match the moments of a distribution and by using universal RKHS, we are able to matching all the moments. We note that a limitation of the consequence is that the only RHKSs with Gauss and Laplace kernels that are bounded satisfy the conditions. However Gauss kernel is the most useful kernel in many applications. Therefore the theorem is still widely applicable.

The practical consequence of this result is that if we know $\mu(\Pr')$, we can infer a suitable β by solving the following minimization problem: this is the kernel mean matching (KMM) procedure:

$$\underset{\beta}{\text{minimize}} \left\| \mu(\Pr{}') - \mathbf{E}_{x\sim\Pr(x)}\left[\beta(x)\Phi(x)\right] \right\| \text{ subject to } \beta(x) \geq 0 \text{ and } \mathbf{E}_{x\sim\Pr(x)}\left[\beta(x)\right] = 1. \tag{6.6}$$

Lemma 6.3.3 (Huang et al. (2006b))**.** *The optimization problem (6.6) is convex. Moreover, assume that* $\Pr'$ *is absolutely continuous with respect to* $\Pr$ *(so* $\Pr(A) = 0$ *implies* $\Pr'(A) = 0$*). And assume that* k *is universal. Then the solution of (6.6),* β*, is* $Pr'(x) = \beta(x)Pr(x)$*.*

Proof. The convexity of the objective function comes from the facts that the norm is a convex function and the integral is a linear functional in β. The other constraints are convex, too. By the virtue of the constraints, any feasible solution of β corresponds to a distribution, as $\int \beta(x) d\Pr(x) = 1$. Moreover, it is not hard to see that $\hat{\beta}(x) := \Pr'(x)/\Pr(x)$ is feasible as it minimizes the objective function with value 0, and that such a $\beta(x)$ exists due to the absolute continuity of $\Pr'(x)$ with respect to $\Pr(x)$. Theorem 6.3.2 implies that there can be only one distribution $\beta(x)\Pr$ such that $\mu(\beta(x)\Pr) = \mu(\Pr')$. Hence $\beta(x)\Pr(x) = \Pr'(x)$. □

6.3.2 Convergence of the Reweighted Means in Feature Space

Lemma 6.3.3 shows that in principle, if we knew Pr and $\mu[\Pr']$, we could fully recover $\Pr'$ from it by solving a simple quadratic program. In practice, however, neither $\mu(\Pr')$ nor Pr is known. Instead, we only have samples X and X' of size m and m', drawn iid from Pr and $\Pr'$ respectively.

Naively we could just replace the expectations in (6.6) by empirical averages and hope that the resulting optimization problem will provide us with a good estimate of β. However, it is to be expected that empirical averages will differ from each other due to finite sample size effects. In this section, we explore two such effects. First, we demonstrate that in the finite sample case, for a fixed β, the empirical estimate of the expectation of β is normally distributed: this provides a natural limit on the precision with which we should enforce

the constraint $\int \beta(x) d\Pr(x) = 1$ when using empirical expectations (we will return to this point in the next section).

Lemma 6.3.4 (Huang et al. (2006b)). *If $\beta(x) \in [0, B]$ is some fixed function of $x \in \mathcal{X}$, then given $x_i \sim \Pr$ iid, the sample mean $\frac{1}{m}\sum_i \beta(x_i)$ has an asymptotically Gaussian distribution about its expectation $\int \beta(x) d\Pr(x)$ with standard deviation bounded by $\frac{B}{2\sqrt{m}}$.*

This lemma is a direct consequence of the central limit theorem (Casella and Berger, 2002).

The second result demonstrates the deviation between the empirical means of $\Pr'$ and $\beta(x)\Pr$ in feature space, given $\beta(x)$ is chosen perfectly in the population sense. In particular, this result shows that to achieve good convergence when the difference in the density probability of $\Pr'$ and Pr is large, we would need more sample from $\Pr'$ and Pr.

Lemma 6.3.5 (Huang et al. (2006b)). *In addition to the Lemma 6.3.4 conditions, assume that we draw $X' := \{x'_1, \ldots, x'_{m'}\}$ iid from $\mathcal{X}$ using $\Pr' = \beta(x)\Pr$, and $\|\Phi(x)\| \leq R$ for all $x \in \mathcal{X}$. Then with probability at least $1 - \delta$*

$$\left\| \frac{1}{m}\sum_{i=1}^{m} \beta(x_i)\Phi(x_i) - \frac{1}{m'}\sum_{i=1}^{m'} \Phi(x'_i) \right\| \leq \left(1 + \sqrt{-2\log\delta/2}\right) R\sqrt{B^2/m + 1/m'} \tag{6.7}$$

Proof. Let $\Xi(X, X') := \left\| \frac{1}{m}\sum_{i=1}^{m} \beta(x_i)\Phi(x_i) - \frac{1}{m'}\sum_{i=1}^{m'} \Phi(x'_i) \right\|$. The proof follows by first bounding the tail behavior using a concentration inequality and subsequently by bounding the expectation.

To apply McDiarmid's tail bound (McDiarmid, 1989) first we need to bound the change in $\Xi(X, X')$ if we replace any x_i by some $\bar{x}_i$ and likewise if we replace any x'_i by some arbitrary $\bar{x}'_i$ from $\mathcal{X}$. By the triangle inequality of function norm, a replacement of x_i by some arbitrary $x \in \mathcal{X}$ can change $\Xi(X, X')$ by at most $\frac{1}{m}\|\beta(x_i)\Phi(x_i) - \beta(x)\Phi(x)\| \leq \frac{2BR}{m}$. Likewise, a replacement of x'_i by x changes $\Xi(X, X')$ by at most $\frac{2R}{m'}$. Since $m(2BR/m)^2 + m'(2R/m')^2 = 4R^2(B^2/m + 1/m')$, then we have

$$\Pr\left\{|\Xi(X, X') - \mathbf{E}_{X,X'}\left[\Xi(X, X')\right]| > \epsilon\right\} \leq 2\exp\left(-\epsilon^2/2R^2(B^2/m + 1/m')\right)$$

Hence with probability $1-\delta$ the deviation of the random variable from its expectation is bounded by

$$|\Xi(X,X') - \mathbf{E}_{X,X'}\left[\Xi(X,X')\right]| \leq R\sqrt{-2\log\frac{\delta}{2}\left(\frac{B^2}{m}+\frac{1}{m'}\right)}$$

To bound the expected value of $\Xi(X,X')$ we use $\mathbf{E}_{X,X'}\left[\Xi(X,X')\right] \leq \sqrt{\mathbf{E}_{X,X'}\left[\Xi(X,X')^2\right]}$. Since all terms in $\Xi(X,X')$ have the same mean, $\mu(\Pr')$, we obtain

$$\begin{aligned}
&\mathbf{E}_{X,X'}\left\|\frac{1}{m}\sum_{i=1}^{m}\beta(x_i)\Phi(x_i) - \frac{1}{m'}\sum_{i=1}^{m'}\Phi(x_i')\right\|^2 \\
=&\frac{1}{m}\mathbf{E}_{x\sim\Pr}\left[\left\|\beta(x)\Phi(x)-\mu(\Pr{}')\right\|^2\right] + \frac{1}{m'}\mathbf{E}_{x\sim\Pr{}'}\left[\left\|\Phi(x)-\mu(\Pr{}')\right\|^2\right] \\
\leq&(B^2/m+1/m')\mathbf{E}_{x\sim\Pr'(x)}k(x,x) \leq R^2(B^2/m+1/m') \qquad (6.8)
\end{aligned}$$

Combining the bounds on the mean and the tail proves the claim. □

Note that this lemma shows that for a *given* $\beta(x)$, which is correct in the population sense, we can bound the deviation between the mean and the importance-sampled mean in feature space. It is *not* a guarantee that we will find coefficients β_i when solving the optimization problem, which are close to $\beta(x_i)$. But it gives us a useful upper bound on the outcome of the optimization problem.

Lemma 6.3.5 implies that we have $O(B\sqrt{1/m+1/m'B^2})$ convergence in m, m' and B. This means that, for very different distributions we need a large equivalent sample size to get reasonable convergence. The result also implies that it is unrealistic to assume that the empirical means (reweighted or not) should match exactly.

6.3.3 Empirical KMM Optimization

To find suitable values of $\beta \in \mathbb{R}^m$ we want to minimize the discrepancy between means subject to constraints $\beta_i \in [0,B]$ and $|\frac{1}{m}\sum_{i=1}^{m}\beta_i - 1| \leq \epsilon$. The upper bound is to limit the influence that a single point can effect, i.e., it has a really high weight, so that the method is more robust. The former limits the scope of discrepancy between Pr and Pr′ whereas the latter ensures that the corresponding measure $\beta(x)\Pr(x)$ is close to a probability

distribution. The objective function is given by the discrepancy term between the two empirical means. Using $K_{ij} := k(x_i, x_j)$ and $\kappa_i := \frac{m}{m'} \sum_{j=1}^{m'} k(x_i, x'_j)$ one may check that

$$\left\| \frac{1}{m} \sum_{i=1}^{m} \beta_i \Phi(x_i) - \frac{1}{m'} \sum_{i=1}^{m'} \Phi(x'_i) \right\|^2 = \frac{1}{m^2} \beta^\top K \beta - \frac{2}{m^2} \kappa^\top \beta + \text{const.}$$

Now we have all necessary ingredients to formulate a quadratic problem to find suitable β via

$$\underset{\beta}{\text{minimize}} \ \frac{1}{2} \beta^\top K \beta - \kappa^\top \beta \text{ subject to } \beta_i \in [0, B] \text{ and } \left| \sum_{i=1}^{m} \beta_i - m \right| \leq m\epsilon. \tag{6.9}$$

From Lemma 6.3.4, a good choice of ϵ should be $O(B/\sqrt{m})$. Note that (6.9) is a quadratic program which can be solved efficiently. It is expected that the empirical solution of β is well-concentrated with respect to the true weights derived from (6.6), although there is a trade off of the approximation quality by adding constraint on β to make sure that the data is not too unevenly weighted. If the true β is well-approximated by the empirical β_i, we could achieve the convergence guarantee of the difference in the density probabilities in terms of the empirical β as has been demonstrated in Section 6.3.2.

6.4 Risk Estimates

So far we consider the distribution matching for the purpose of finding a reweighting scheme between the empirical means on training X and test set X'. The section attempts to show that as long as the means on the test set are well enough approximated, we are able to obtain *almost unbiased* risk estimates *regardless* of the actual values of β_i vs. their importance sampling weights $\beta(x_i)$. The price is an increase in the variance of the estimate. $m^2 / \|\beta\|^2$ will act as an effective sample size. A key assumption is that the induced loss function class is well behaved.

For simplicity, we only consider the *transductive* case. That is, we will make uniform convergence statements with respect to $\mathbf{E}_{Y'|X'}$ and $\mathbf{E}_{Y|X}$ only. We are interested in the behavior of the loss induced function class $l(x, y, \theta)$ rather than $\langle \phi(x, y), \theta \rangle$. Thus the difference between ϕ used in Section 6.2, which relates to the parameterization of the model, and Φ used in the current section, relating to the loss.

Key Assumption 3: Denote by $\Theta : \mathcal{X} \to \mathcal{Y}$ a class of functions and let $l : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a expected loss function $l(x, \theta)(\ \theta \in \Theta : x \to y)$ over conditional distribution $p(y|x)$. Assume $l(x, \theta)$ is a smooth function and thus it can be well approximated in a universal Reproducing Kernel Hilbert Space $\mathcal{H}$ with bounded norm, therefore the smooth loss function $l(x, \theta)$ can be approximately expressed as inner product in the universal RKHS with kernel $k(x, x') \leq R^2$, i.e. $l(x, \theta) = \langle \Phi(x), \Theta \rangle$ such that each θ maps to some vector of parameters Θ in the feature space where $\|\Theta\|_{\mathcal{H}} \leq C$. That is we minimize the empirical risk over the well approximated smooth loss function. Examples of such smooth loss functions could be the modified quadratic loss l_q (Zhang and Oles, 2001) or the smoothed hinge loss l_h (Rennie and Srebro, 2005) that are defined as,

$$l_q(z) = \begin{cases} (1-z)^2 & z \leq 1 \\ 0 & z \geq 1 \end{cases}$$

$$l_h(z) = \begin{cases} 1/2 - z & z \leq 0 \\ \frac{1}{2}(1-z)^2 & 0 < z < 1 \\ 0 & z \geq 1 \end{cases}$$

where z is the product of the true label and the prediction.

The main conclusion in this section is the following corollary.

Corollary 6.4.1 (Huang et al. (2006b))**.** *Suppose that key assumptions 1, 2 and 3 are satisfied and let X, X' be iid samples drawn from* Pr *and* Pr$'$ *respectively, and let $Y|X$ be drawn iid from* Pr$(y|x)$*. Moreover, let $\mathcal{G}$ be a class of loss-induced functions $l(x, \theta)$ with $\|\theta\| \leq C$ and let $M := m^2 / \|\beta\|^2$. And assume that also $l(x, y, \theta)$ can be expressed as an element of an RKHS via $\langle \Phi(x, y), \Theta \rangle$ with $\|\Theta\| \leq C$ and $\|\Phi(x, y)\| \leq R$. In addition, assume for any β_i such that*

$$\left\| \frac{1}{m} \sum_{i=1}^{m} \beta_i \Phi(x_i) - \frac{1}{m'} \sum_{i=1}^{m'} \Phi(x'_i) \right\| \leq \epsilon \tag{6.10}$$

Then with probability at least $1 - \delta$

$$\sup_{l(\cdot,\cdot,\theta) \in \mathcal{G}} \left| \frac{1}{m} \sum_{i=1}^{m} \beta_i l(x_i, y_i, \theta) - \mathbf{E}_{Y'|X'} \left[\frac{1}{m'} \sum_{i=1}^{m} l(x'_i, y'_i, \theta) \right] \right| \leq \frac{(1 + \sqrt{(-\log \delta)/2}) 2CR}{\sqrt{M}} + C\epsilon$$

This means that if we minimize the reweighted empirical risk we will, with high probability, we minimize an upper bound on the expected risk on the test set. The minimization of the reweighted empirical risk is based on minimizing the empirical mean between training and test data in the feature space as in (6.10), assuming that the set of loss functions need to be smooth such that minimizing the expected risk everywhere can be accomplished by doing so on a selected subset of locations.

A direct practical consequence of this corollary is that one could have some prior knowledge of the expected performance using KMM. One would expect to have a reasonable low risk on the test data using KMM only if both feature map means are close. This also implies that if the risk of test data diverges from reasonable range, the training and test must have very different distributions. Therefore, it is useful to check the condition (6.10) before running KMM.

There are two steps to prove the corollary: first we show that for smooth functions expected loss $l(x,\theta) := \mathbf{E}_{y|x} l(x,y,\theta)$, the coefficients β_i can be used to obtain a risk estimate with low bias. Then, we show that the random variable $\sum_i \beta_i l(x_i,y_i,\theta)$ is concentrated around $\sum_i \beta_i l(x_i,\theta)$, if we condition $Y|X$. Combining the bounds from both lemmas below gives the result in Corollary 6.4.1.

Lemma 6.4.2 (Huang et al. (2006b))**.** *Under the assumption of Corollary 6.4.1 that there exist some β_i such that*

$$\left\| \frac{1}{m}\sum_{i=1}^{m} \beta_i \Phi(x_i) - \frac{1}{m'}\sum_{i=1}^{m'} \Phi(x_i') \right\| \leq \epsilon$$

Then the empirical risk estimates can be bounded as

$$\sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \left| \mathbf{E}_{Y|X}\left[\frac{1}{m}\sum_{i=1}^{m} \beta_i l(x_i,y_i,\theta) \right] - \mathbf{E}_{Y'|X'}\left[\frac{1}{m'}\sum_{i=1}^{m} l(x_i',y_i',\theta) \right] \right| \leq C\epsilon \tag{6.11}$$

Proof. First note that by key assumption 1 the conditional distributions $\Pr(y|x)$ are the same for $\Pr$ and $\Pr'$. By linearity, apply the epxectation $\mathbf{E}_{Y|X}$ to each sum individually. Then, by key assumption 3 the expected loss $l(x,\theta)$ can be written as $\langle \Phi(x), \theta \rangle$. Therefore, rewrite the left hand side of (6.11) as

$$\sup_{l(\cdot,\theta)\in\mathcal{G}} \left| \frac{1}{m}\sum_{i=1}^{m} \beta_i l(x_i,\theta) - \frac{1}{m'}\sum_{i=1}^{m'} l(x_i',\theta) \right| \leq \sup_{\|\Theta\|\leq C} \left| \left\langle \frac{1}{m}\sum_{i=1}^{m} \beta_i \Phi(x_i) - \frac{1}{m'}\sum_{i=1}^{m'} \Phi(x_i'), \Theta \right\rangle \right|$$

By the definition of norms this is bounded by $C\epsilon$, which proves the claim. □

The second step in relating a reweighted empirical average using (X, Y) and the expected risk with respect to $\Pr'$ requires us to bound deviations of the first term in (6.11).

Lemma 6.4.3 (Huang et al. (2006b))**.** *Suppose the assumptions in Corollary 6.4.1 satisfied. Then with probability at least* $1 - \delta$ *over all* $Y|X$

$$\sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \left| \frac{1}{m}\sum_{i=1}^{m} \beta_i l(x_i, y_i, \theta) - \frac{1}{m}\sum_{i=1}^{m} \beta_i l(x_i, \theta) \right| \leq (1 + \sqrt{(-\log\delta)/2}) 2CR/\sqrt{M} \tag{6.12}$$

Proof. The proof strategy is almost identical to the one of Lemma 6.3.5 Denote by

$$\Xi(Y) := \sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \beta_i \left[l(x_i, y_i, \theta) - l(x_i, \theta) \right] \tag{6.13}$$

the maximum deviation between empirical mean and expectation. The key is that the random variables $y_1, \ldots, y_m$ are conditionally independent given X. Using the Symmetrization idea, replacing one y_i by an arbitrary $y' \in \mathcal{Y}$ leads to a change in $\Xi(Y)$ which is bounded by $\frac{\beta_i}{m} C \left\| \Phi(x_i, y_i) - \Phi(x_i, y') \right\| \leq 2CR\beta_i/m$. Then apply McDiarmid's theorem yields

$$\Pr_{Y|X}\left\{ |\Xi(Y) - \mathbf{E}_{Y|X}\Xi(Y)| > \epsilon \right\} \leq \exp\left(-\epsilon^2 m^2 / \left(2C^2 R^2 \left\| \beta \right\|_2^2 \right) \right). \tag{6.14}$$

In other words, $M := m^2 / \left\| \beta \right\|_2^2$ acts as an effective sample size when determining large deviations. Next we use symmetrization to obtain a bound on the expectation of $\Xi(Y)$. Again, the proof routine is the same as supplied in the thesis appendix.

$$\mathbf{E}_{Y|X}[\Xi(Y)] \leq \frac{1}{m}\mathbf{E}_{Y|X}\mathbf{E}_{\bar{Y}|X}\left[\sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \left| \sum_{i=1}^{m} \beta_i l(x_i, y_i, \theta) - \beta_i l(x_i, \bar{y}_i, \theta) \right| \right] \tag{6.15}$$

$$\leq \frac{2}{m}\mathbf{E}_{Y|X}\mathbf{E}_{\sigma}\left[\sup_{l(\cdot,\cdot,\theta)\in\mathcal{G}} \left| \sum_{i=1}^{m} \sigma_i \beta_i l(x_i, y_i, \theta) \right| \right] \text{ where } \sigma_i \in \{\pm 1\}. \tag{6.16}$$

The first inequality follows from convexity. The second one follows from the fact that all $y_i, \bar{y}_i$ pairs are independently and identically distributed, hence we can swap these pairs.

Now we bound on the Rademacher average for constant β_i for the right hand side. We use the condition of the lemma, namely that $l(x, y, \theta) = \langle \Phi(x, y), \Theta \rangle$ for some Θ with

$\|\Theta\| \leq C$. This allows us to bound the supremum. Combine the fact of the convexity of x^2 yields the following bounds on the right hand side in (6.16)

$$\text{RHS} \leq \frac{2}{m}\mathbf{E}_{Y|X}\mathbf{E}_{\sigma}C\left\|\sum_{i=1}^{m}\sigma_i\beta_i\Phi(x_i,y_i)\right\| \quad \leq \frac{2}{m}C\sqrt{\mathbf{E}_{Y|X}\mathbf{E}_{\sigma}\left\|\sum_{i=1}^{m}\sigma_i\beta_i\Phi(x_i,y_i)\right\|^2} \tag{6.17}$$

$$= \frac{2}{m}C\sqrt{\sum_{i=1}^{m}\beta_i^2\mathbf{E}_{y_i|x_i}\left\|\Phi(x_i,y_i)\right\|^2} \quad \leq \frac{2}{m}CR\left\|\beta\right\|_2 = \frac{2CR}{\sqrt{M}}. \tag{6.18}$$

Combining the bound on the expectation and solving the tail bound for ϵ proves the lemma. □

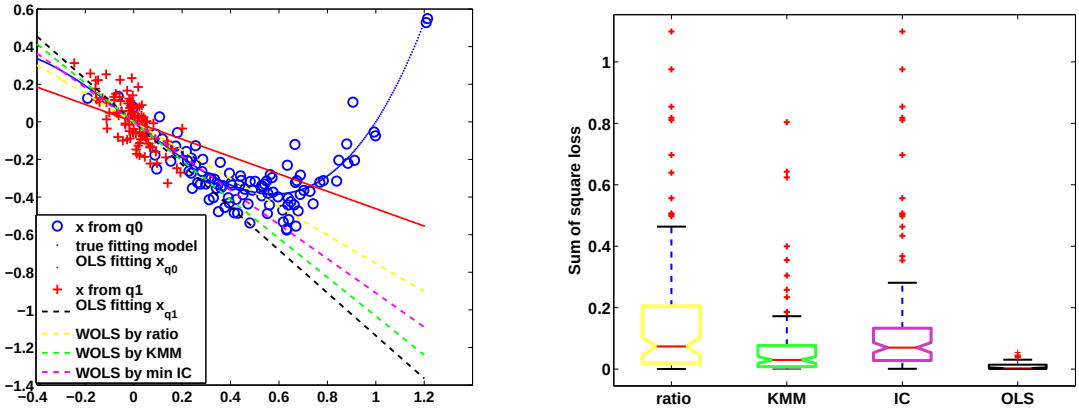
6.5 Evaluation

6.5.1 Toy Regression Example

My first experiment is on toy data. It is intended mainly to provide a comparison with the approach of Shimodaira (2000). Recall that this method uses an information criterion to optimise the weights, under certain restrictions on Pr and Pr′ (namely, Pr′ must be known, while Pr can be either known exactly, Gaussian with unknown parameters, or approxiated via kernel density estimation).

The statistics of of the toy data has been showned in Section 6.1.

I sample 100 training (blue circles) and testing (red circles) points from Pr as q_0 and Pr′ as q_1 respectively. I model the observations with a degree 1 polynomial. The black dashed line is a best-case scenario, which is shown for reference purposes: it represents the model fit using ordinary least squared (OLS) on the labeled test points. The red line is a second reference result, derived only from the training data via OLS, and predicts the test data very poorly. The other three dashed lines are fit with weighted ordinary least square (WOLS), using one of three weighting schemes: the ratio of the underlying training and test densities, KMM, and the information criterion of (Shimodaira, 2000) that I discussed in Section 6.1.2. A summary of the performance over 100 trials is shown in Figure 6.2(b). Clearly, the new method outperforms the two other reweighting methods.

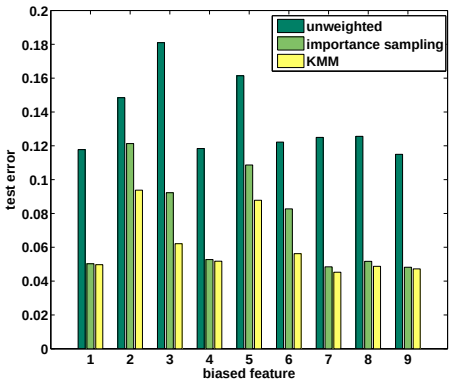


(a) (b)

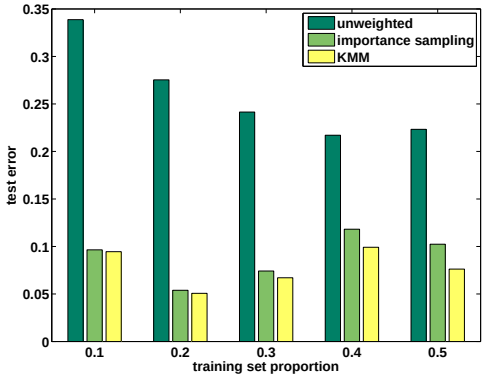
Figure 6.2: (a) Polynomial models of degree 1 fit with OLS and WOLS;(b) Average performance of three WOLS methods and OLS on the test data in (a). Labels are *Ratio* for ratio of test to training density; KMM for our approach; *min IC* for the approach of Shimodaira (2000); and *OLS* for the model trained on the labeled test points.

6.5.2 Real World Datasets

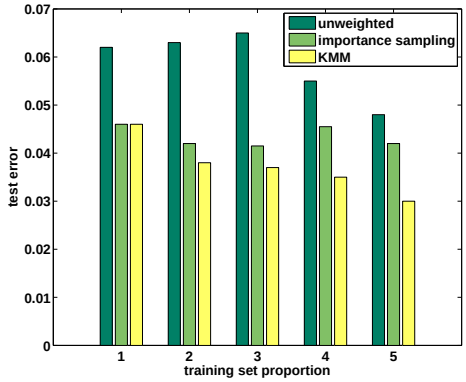
I next test the proposed approach on real world data sets, from which I select training examples using a deliberately biased procedure (as in (Zadrozny, 2004; Rosset et al., 2005)). To describe our biased selection scheme, I need to define an additional random variable s_i for each point in the pool of possible training samples, where $s_i = 1$ means the ith sample is included, and $s_i = 0$ indicates an excluded sample. Two situations are considered: the selection bias corresponds to our assumption regarding the relation between the training and test distributions, and $P(s_i = 1|x_i, y_i) = P(s_i|x_i)$; or s_i is dependent only on y_i, i.e. $P(s_i|x_i, y_i) = P(s_i|y_i)$, which potentially creates a greater challenge since it violates our key assumption 1. In the following, I compare the proposed method (labeled *KMM*) against two others: a baseline unweighted method (*unweighted*), in which no modification is made, and a weighting by the inverse of the true sampling distribution (*importance sampling*), as in (Zadrozny, 2004; Rosset et al., 2005). In the experiments, I use a Gaussian kernel $\exp(-\sigma\|x_i - x_j\|^2)$ in kernel classification and regression algorithms, and parameters



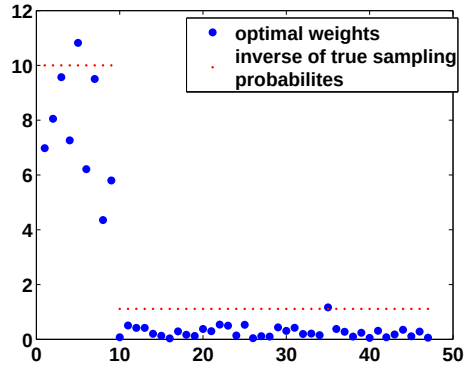
(a) Simple bias on features



(b) Joint bias on features



(c) Bias on labels



(d) β vs inverse sampling prob.

Figure 6.3: Classification performance analysis on breast cancer dataset from UCI

$\epsilon = (\sqrt{m} - 1)/\sqrt{m}$ and $B = 1000$ in the optimization (6.9).

Breast Cancer Dataset

This dataset is from the UCI Archive, and is a binary classification task. It includes 699 examples from 2 classes: benign (positive label) and malignant (negative label). The data are randomly split into training and test sets, where proportion of examples used for training varies from 10% to 50%. Test results are averaged over 30 trials, and were obtained using a support vector classifier with kernel size $\sigma = 0.1$.

First, I consider a biased sampling scheme based on the input features, of which there are nine, with integer values from 0 to 9. Since smaller feature values predominate in the unbiased data, I sample according to $P(s = 1|x \leq 5) = 0.2$ and $P(s = 1|x > 5) = 0.8$, repeating the experiment for each of the features in turn. Results are an average over 30 random training/test splits, with 1/4 of the data used for training and 3/4 for testing. Performances is shown in Figure 6.3(a): KMM consistently outperform the unweighted method, and match or exceed the performance obtained using the known distribution ratio. The reason that why KMM outperforms the one using true distribution ratio is that the test set is the empirical sample from test distribution, not the truly test distribution, while importance sampling uses the ratio of the true distribution which will be less accurate in some cases.

Next, I consider a sampling bias that operates jointly across multiple features. I select samples less often when they are further from the sample mean $\overline{x}$ over the training data, i.e. $P(s_i|x_i) \propto \exp(-\sigma\|x_i - \overline{x}\|^2)$ where $\sigma = 1/20$. Performance of KMM in 6.3(b) is again better than the unweighted case, and as good or better as reweighting using the sampling model.

Finally, I consider a simple biased sampling scheme which depends only on the label y: $P(s = 1|y = 1) = 0.1$ and $P(s = 1|y = -1) = 0.9$ (the data has on average twice as many positive as negative examples when uniformly sampled). Average performance for different training/testing split proportions are in Figure 6.3(c); remarkably, despite our assumption regarding the difference between the training and test distributions being violated, our method still improves the test performance, and outperforms the reweighting by density ratio for large training set sizes. Figure 6.3(d) shows the weigths β are proportional to the

inverse of true sampling probabilities: positive examples have higher weights and negative ones have lower weights.

Further Benchmark Datasets

I next compare the performance on further benchmark datasets[3] by selecting training data via various biased sampling schemes. Specifically, for the sampling distribution bias on labels, I use $P(s = 1|y) = \exp(a + by)/(1 + \exp(a + by))$ (datasets 1 to 5) and simple stepsize distribution $P(s = 1|y = 1) = a$, $P(s = 1|y = -1) = b$ (datasets 6 and 7). For the other datasets, I consistently generate biased sampling schemes over their features. I first do PCA, selecting the first principal component of training data and the corresponding projection values. Denote the minimum value of the projection as m and the mean as $\overline{m}$. Then I apply a normal distribution with mean $m+(\overline{m}-m)/a$ and variance $(\overline{m}-m)/b$ as the biased sampling scheme. Please refer to Table 6.5.2 for detailed parameter settings. I use penalized LMS for regression problems and SVM for classification problems. To evaluate generalization performance, I utilize the *normalized mean square error (NMSE)* given by $\frac{1}{n}\sum_{i=1}^{n}\frac{(y_i-\mu_i)}{\mathbf{var}y}$ for regression problems, and use standard average test error for classification problems. In 16 out of 23 experiments, KMM is the most accurate (see Table 1), despite having no prior information about the bias of the test sample (and, in some cases, despite the additional fact that the data reweighting does not conform to our key assumption 1). In addition, the KMM *always* improves test performance compared with the unweighted case.

Tumor Diagnosis using Microarrays

The next benchmark is a dataset of 102 microarrays from prostate cancer patients (Singh et al., 2002). Each of these microarrays measures the expression levels of 12,600 genes. The dataset comprises 50 samples from normal tissues (positive label) and 52 from tumor tissues (negative label). I simulate the realisitc scenario that two sets of microarrays A and B are given with dissimilar proportion of tumor samples, and we want to perform cancer diagnosis via classification, training on A and predicting on B. I select training examples

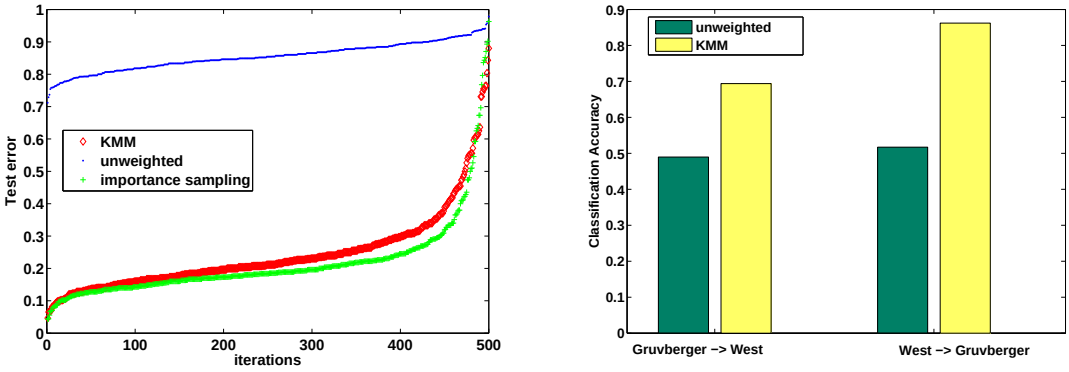
[3]Regression datasets cf. `http://www.liacc.up.pt/~ltorgo/Regression/DataSets.html`; classification sets are from UCI. Sets with numbers in brackets are examined by different sampling schemes.

Table 6.1: Test results for three methods on 18 datasets with different sampling schemes. Datasets marked with * are for regression problems. The results are the averages over 10 trials for regression problems and 30 trials for classification problems.

				NMSE / Test err.		
DataSet	n_{tr}	selected	n_{tst}	unweighted	import. sampling	KMM
1. Abalone*	2000	853	2177	1.00 ± 0.08	1.1 ± 0.2	$\mathbf{0.6 \pm 0.1}$
2. CA Housing*	16512	3470	4128	2.29 ± 0.01	1.72 ± 0.04	$\mathbf{1.24 \pm 0.09}$
3. Delta Ailerons(1)*	4000	1678	3129	0.51 ± 0.01	0.51 ± 0.01	$\mathbf{0.401 \pm 0.007}$
4. Ailerons*	7154	925	6596	1.50 ± 0.06	$\mathbf{0.7 \pm 0.1}$	1.2 ± 0.2
5. haberman(1)	150	52	156	0.50 ± 0.09	0.37 ± 0.03	$\mathbf{0.30 \pm 0.05}$
6. USPS(6vs8)(1)	500	260	1042	0.13 ± 0.18	0.1 ± 0.2	$\mathbf{0.1 \pm 0.1}$
7. USPS(3vs9)(1)	500	252	1145	0.016 ± 0.006	$\mathbf{0.012 \pm 0.005}$	0.013 ± 0.005
8. Bank8FM*	4500	654	3692	0.5 ± 0.1	$\mathbf{0.45 \pm 0.06}$	0.47 ± 0.05
9. Bank32nh*	4500	740	3692	23 ± 4.0	$\mathbf{19 \pm 2}$	$\mathbf{19 \pm 2}$
10. cpu-act*	4000	1462	4192	10 ± 1	4.0 ± 0.2	$\mathbf{1.9 \pm 0.2}$
11. cpu-small*	4000	1488	4192	9 ± 2	4.0 ± 0.2	$\mathbf{2.0 \pm 0.5}$
12. Delta Ailerons(2)*	4000	634	3129	2 ± 2	$\mathbf{1.5 \pm 1.5}$	1.7 ± 0.9
13. Boston house*	300	108	206	0.8 ± 0.2	$\mathbf{0.74 \pm 0.09}$	0.76 ± 0.07
14. kin8nm*	5000	428	3192	0.85 ± 0.2	$\mathbf{0.81 \pm 0.1}$	0.81 ± 0.2
15. puma8nh*	4499	823	3693	1.1 ± 0.1	$\mathbf{0.77 \pm 0.05}$	0.83 ± 0.03
16. haberman(2)	150	90	156	0.27 ± 0.01	0.39 ± 0.04	$\mathbf{0.25 \pm 0.2}$
17. USPS(6vs8) (2)	500	156	1042	0.23 ± 0.2	0.23 ± 0.2	$\mathbf{0.16 \pm 0.08}$
18. USPS(6vs8) (3)	500	104	1042	0.54 ± 0.0002	0.5 ± 0.2	$\mathbf{0.16 \pm 0.04}$
19. USPS(3vs9)(2)	500	252	1145	0.46 ± 0.09	0.5 ± 0.2	$\mathbf{0.2 \pm 0.1}$
20. Breast Cancer	280	96	419	0.05 ± 0.01	0.036 ± 0.005	$\mathbf{0.033 \pm 0.004}$
21. Indias diabets	200	97	568	0.32 ± 0.02	$\mathbf{0.30 \pm 0.02}$	$\mathbf{0.30 \pm 0.02}$
22. ionosphere	150	64	201	0.32 ± 0.06	0.31 ± 0.07	$\mathbf{0.28 \pm 0.06}$
23. German credit	400	214	600	0.283 ± 0.004	0.282 ± 0.004	$\mathbf{0.280 \pm 0.004}$

Table 6.2: Statistics of datasets used in the experiments

	1*	2*	3*	4*	5	6	7	8*	9*	10*	11*	12*
σ	1e-1	1e-1	1e3	1e-5	1e-2	1/128	1/128	1e-1	1e-2	1e-12	1e-12	1e3
a	1.5	10	1e3	1e4	0.2	0.1	0.1	20	4	4	4	1e3
b	-0.5	-5	-1	-5	0.8	0.9	0.9	8	8	8	8	-1
	13*	14*	15*	16	17	18	19	20	21	22	23	
σ	1e-4	1e-1	1e-1	1e-2	1/128	1/128	1/128	1e-1	1e-4	1e-1	1e-4	
a	2	4	4	0.2	4	4	4	2	2	2	4	
b	2	6	4	0.8	4	8	4	2	2	2	4	



(a) (b)

Figure 6.4: (a) test errors in 500 trials for cancer diagnosis Index sorted by error values. (b) Classification results when training and testing are from different sources of Microarray examples for breast cancer

via a biased selection scheme as $P(s = 1|y = 1) = 0.85$ and $P(s = 1|y = -1) = 0.15$, the remaining data points form the test set. This setting is similar to the women breast cancer example as I discussed before. I then perform SVM classification, once for the unweighted, the KMM, and the importance sampling approach. The experiment is repeated over 500 independent draws from the dataset according to our biased scheme; the 500 resulting test errors are plotted in Figure 6.4(a) in Figure 6.4(a) (sorted in order of test error size, to make clear how the test errors of the three approaches are consistently ordered for any given draw of the data, regardless of the actual test size). The KMM achieves much higher accuracy levels than the unweighted approach, and is very close to the importance sampling approach.

I study a very similar scenario on two breast cancer microarray datasets from (Gruvberger et al., 2001) and (West et al., 2001), measuring the expression levels of 2,166 common genes for normal and cancer patients (Warnat et al., 2005). I train an SVM on one of them and test on the other. KMM achieves significant improvement in classification accuracy over the unweighted, SVM as shown in Figure 6.4(b). Hence KMM promises to be a valuable tool for cross-platform microarray classification.

6.6 Summary

I have presented a new kernel method of dealing with sampling bias in various of learning problems via directly estimating the resampling weights by matching training and testing distributions in a feature space. In addition, I presented a general theory (Huang et al., 2006b) in bounding the matching error in terms of the support of the distribution and the sample sizes. The experiments demonstrated the advantage of correcting sampling bias using unlabeled data in various classification and regressions tasks. The new technique promises a tool for many other application problems, such as brain computer interface and data privacy preserving.

Chapter 7

Conclusions

Research presented in this thesis has focused on problems—unsupervised and semi-supervised learning—that involve learning with partially labeled data, approaching them mainly from two perspectives. In the first part of the thesis I approach these problems from a graph based perspective. In the second part, I considered a statistical setting where there is a shift between training and test distributions. Both parts of the thesis address non-standard learning scenarios for classical statistical learning.

In the graph based approach, as should be clear by now, unsupervised and semi-supervised learning methods collect heterogeneous and homogeneous information sources, and can be unified in a regularization framework. The unification of unsupervised, semi-supervised and supervised learning I proposed is based on the observation that the cut cost objectives used by unsupervised learning algorithms can also be taken as regularizers for labeling functions on graphs. Given this unification, I am able to develop unsupervised and semi-supervised learning algorithms on directed graphs, hypergraphs and complex networks, which encode more complex data relationships than simple relation on undirected graphs. Moreover, for each of these generalized graph structures, I show how information propagation can be globally characterized by distinct random walk models that underly the principle of graph based learning, and then we can use this characterization to develop new learning algorithms.

Second, the thesis also investigates a statistically oriented approach to solving a difficult learning scenario where the training and test examples come from different distributions.

By using an abundance of unlabeled data, the new method I have proposed produces resampling weights that correct bias by minimizing the empirical distribution discrepancy between training and test data in a feature space. This new approach has several advantages over previous methods; for example, it does not require explicit density estimation nor prior knowledge of the sampling schemes. The theoretical analysis examines the convergence properties of the empirical means and the empirical risk estimates in a RKHS.

Overall, the work presented in this thesis contributes methods that lead to state-of-art performance on tasks considered, and provides a number of useful algorithms for problems in learning with partially labeled data.

Future work The work motivates future research in the following directions.

- Machine learning problems

  Online learning on graphs The problems I considered in graph based learning so far are in an offline setting. It would be interesting to extend these algorithms to an online setting where the models are refined as the graph expands. This problem occurs in many real world scenarios where data is collected online and one has to make decision in real time. In this case, we would like to only label a subset of the examples so that hopefully the current labeled set would provide the good representation of the entire data stream. Online learning algorithms therefore are involved with issues in active learning.

  Learning with multiple graphs The graph based algorithms in this thesis mainly consider a single graph. In some cases, the relationships among same type of data objects can be observed from multiple sources. For example, in computer vision, the "set-of-patches" approach is a successful technique in image classification. This method represents an image by features generated from a set of small image patches. Each feature type provides a kind of image representation that corresponds to a similarity graph connecting the images. In this case we may obtain multiple graphs from the data. How to use a graph based approach to solve this problem has

not been intensively studied before and it would be interesting to explore further in this direction.

- Applications

Spam detection In the Web, it has been noticed that there have been many attempts to mischievously influence page ranking by constructing link farms, i.e. link spamming. Web spamming is a major problem to search engines and have negative impacts to the Web community. Learning algorithms on directed graphs might be able to provide efficient solutions for distinguishing detecting the "bad" pages from general Web pages via link analysis.

Data privacy preserving In many real world applications one of the key concerns is respecting privacy and protecting access to confidential information, while still allowing useful, large-scale patterns to be extracted from the data. Privacy and security considerations are typically enforced by restricting access to the original data, which forces data mining to be conducted without direct access. A general question to ask is whether one can develop accurate models without access to the original data? One scenario where these privacy and security issues arise is when an owner of a confidential database wishes to extract useful patterns without revealing any individual examples in the database. The situation can be also extended to preserve privacy in multiple databases when running a data mining algorithm on their union. The approach used to solve the distribution shifting problem can be applied in this case by preserving an underlying confidential database given a public "standard" database that is adopted as a proxy. We can extract the weights to refine the data mining algorithm on the proxy dataset so that the result will match the one achieved when analyzing the confidential database.

Some immediate applications are to insurance company customer profiles and hospital patient records. If successful, this technique could also be applied to many other domains, including medical record data mining, fraud detection in banking and financial databases, and Web log analysis.

- Theory

Discrete analysis on graphs As I have shown, regularization is a very useful concept when cosidering unsupervised and semi-supervised learning algorithms on graphs. We have already shown that some operators, the graph Laplacians, are a discrete analogue of standard differential operators in a continuous space. I conjecture that the regularization operators would be more powerful if we could define higher order differential operators and further construct discrete Taylor expansion on graphs. In addition, many other properties may be useful to discover on graphs. Chung (1997) has achieved some interesting results in exploiting concepts in the isoperimetric problem, Harnack inequalities, heat kernels and Sobolev inequalities. These achievements suggest a promising direction for future theoretical research on graphs.

Appendix

Functional Smoothness in Reproducing Kernel Hilbert Space

The simplest view of kernel methods is that they map the input observations into a very high dimensional feature space, and then solve the problem by considering linear models in that space. Such a space is often referred to as a Reproducing Kernel Hilbert Space(RKHS) and denoted as $\mathcal{H}$. Consistent with Regularization Theory, in kernel methods, the regularization is defined in RKHS. Examples include regularization networks and kernel SVMs (Vapnik, 1998).

We have the following theorem (Wahba, 1990; Schölkopf and Smola, 2002) in connection with regularization theory.

Theorem .0.1. *For an RKHS $\mathcal{H}$ associated with the reproducing kernel k, there is a unique corresponding regularization operator $D : \mathcal{H} \to \mathcal{L}_2$ such that for all $f \in \mathcal{H}$,*

$$\langle Dk(x, \cdot), Df(\cdot) \rangle_{\mathcal{L}_2} = f(x)$$

and in particular,

$$\langle Dk(x, \cdot), Dk(x', \cdot) \rangle_{\mathcal{L}_2} = k(x, x')$$

and vice versa.

According to the theorem we can have that

$$\|Df\|^2_{\mathcal{L}_2} = \|f\|^2_{\mathcal{H}} \tag{1}$$

which means that adding the smoothness penalty in the function norm of the $\mathcal{L}_2$ space gives an identical result to penalizing the function norm in Hilbert Space. Choosing different kernels, the norm in the corresponding $RKHS$ encodes different notions of smoothness. Moreover, the kernel associated with the differential operator D is the Green's function $D * D$ (Wahba, 1990; Schölkopf and Smola, 2002).

A specific example resulted from this theorem is RBF kernel. The square norm of the function in Hilbert space with RBF kernel can be explicitly represented in terms of differential operators as (Yuille and Grzywacz, 1988)

$$\|Df\|^2 = \int_{\mathcal{X}} \sum_n \frac{\sigma^{2n}}{n! 2^n} (O^n f(x))^2 dx \tag{2}$$

where $O^{2n} = \Delta^n$, and Δ is the Laplacian. The proof is obtained by rewriting the function as a Taylor expansion in $\mathcal{X}$ in terms of the differential operators. Eq.(2) explains why the Gaussian RBF kernel works so well in many kernel method applications. The reason is that f is smoothed by penalizing all of its derivatives. Therefore, the regularization is comprehensive and the resulting function is very smooth. One can imagine to define Taylor expansion on graphs to produce more powerful regularizers.

Random Walk Interpretation

I present random walk interpretations for methods in (Zhou et al., 2004) and (Zhu et al., 2003). The interpretation for Zhu et al. (2003) is not explicitly outlined in literature before. The interpretation shows that the graph based semi-supervised learning has a strong connection to random walks on Markov chains.

- Commute Time of a Random Walk

  The approach in (Zhou et al., 2004) is involved with commute time of random walk on a undirected graph. In a typical random walk, according to Aldous and Fill, *hitting time* is the expected number of steps to reach vertex v from u for the first time. Starting at vertex u, the expected number of steps to return to u is called *return time* $R(u, u)$. *commute time* $C(u, v)$ is the expected number of steps to go from vertex u through vertex v and back to u again.

Let G denote the inverse of the matrix $\Pi(I-P)$ where Π denotes the diagonal matrix with $\Pi(v,v)=\pi(v)$ for all $v \in V$. Then the commute time satisfies

$$C(u,v) = \begin{cases} G(u,u)+G(v,v)-G(u,v)-G(v,u) & \forall u \neq v \\ 1/\pi(u) & \forall u = v \end{cases} \tag{3}$$

Consider to normalize $H(u,v)$ by

$$\overline{H}(u,v) = \sqrt{\pi(u)\pi(v)}H(u,v)$$

Let $\overline{G}$ denote the inverse of the matrix $I-\alpha S$, and G denote the inverse of the matrix $D-\alpha W$. Then normalized commute time satisfies (Zhou and Schölkopf, 2004),

$$\overline{G}(u,v) = \frac{G(u,v)}{\sqrt{C(u,u)C(v,v)}}$$

Then the solution of (2.39) corresponds to picking larger value from $p_+(x_u)$ and $p_-(x_u)$, which is in turn comparing the normalized commute times to the labeled data of different classes.

$$p_+(x_u) = \sum_{v:y_v=1} \overline{G}(u,v), \text{and} \quad p_-(x_u) = \sum_{v:y_v=-1} \overline{G}(u,v)$$

- Absorbing Probability of a Random Walk

  The solution for semi-supervised learning in (Zhu et al., 2003) is the standard knowledge in absorbing Markov Chian in which we take all the labeled nodes as absorbing states (denote as A, all states belong to A can not walk to elsewhere after reaching the state) and all the unlabled nodes as transitive states (denote as T). In a Markov chain, the canonical form of the transition probability given there are absorbing and transitive states is:

$$P = \begin{bmatrix} P_{TT} & P_{TA} \\ P_{AA} & P_{AT} \end{bmatrix} = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}$$

  In this semi-supervised problem given both labeled and unlabeled data(l and u), $Q = P_{uu}$ and $R = P_{ul}$.

Lemma .0.2. *When times goes to infinity, the stationary distribution of* $\lim_{n\to\infty} P^n = P^*$ *is*

$$P^* = \begin{bmatrix} 0 & B \\ 0 & I \end{bmatrix}$$

where $B = (I - Q)^{-1}R$

Now the harmonic function method (Zhu et al., 2003) is equivalent to compare the absorbing probability when arriving stationary distribution. Assuming that $f_i \in [0, 1] \forall i$, then we have

$$f = P^* f$$

therefore we can have

$$f_u = (I - P_{uu})^{-1} P_{ul} f_l$$

which is exact the solution of (Zhu et al., 2003).

Learning Theory and some Inequalities

I provide some basic background in learning thoery and some well-known theorems used for deriving error bounds for Chapter 6. The material is based mainly on (Bousquet et al., 2004) and (Mendelson, 2003).

Consider an input space $\mathcal{X}$ and output space $\mathcal{Y}$. We consider binary classification case where $\mathcal{Y} = \{-1, 1\}$. We are given some empirical observations $D = (X, Y) \in \mathcal{X} \times \mathcal{Y}$ which are assumed to be i.i.d generated from an unknown distribution $\Pr(x, y)$ on $\mathcal{X} \times \mathcal{Y}$. This sampling assumption is important in the statistical learning theory. The independence assumption means that each new observation gives new information. The identical distribution means that the observations characterize the underlying probability distribution. The learning goal is to obtain a function $f : \mathcal{X} \to \mathcal{Y}$ that predicts Y from X.

To chose such function f, we hope it results in a low probability of error $P(f(X) \neq Y)$. The risk of f is defined as

$$R(f) = P(f(X)) = \mathbf{E}[\mathbf{1}_{f(X)\neq Y}].$$

The optimal function f^* is obtained by minimizing risk over all possible functions

$$R(f^*) = \inf_f R(f)$$

Since Pr is unknown, a commonly used way is to measure an empirical risk accordingly

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{f(X)\neq Y}$$

However, since the input space is infinite, we can always find an optimal f^* that predict perfectly on every training observations, thus $R_n(f) = 0$. The problem with this function is that it may perform badly on test data that causes overfitting. One way to fix this problem is to constrain f into a model $\mathcal{F}$ and add regularization that leads to a regularized empirical risk minimization problem

$$f_n^* = \operatorname*{argmin}_{f\in\mathcal{F}} R_n(f) + \lambda\Omega(f)$$

where $\Omega(f)$ decodes a prior knowledge in function smoothness.

We can judge whether the learned function f_n is good or not based on $R(f_n)$. However, we can not compute $R(f_n)$ from data as it depends on unknown distribution Pr and this quantity is a random variable since it depends on data. Therefore, in statistical learning, $R(f_n)$ is examined by relating it to an estimate such as the empirical risk $R_n(f_n)$. Typically, one would be interested at the upper and lower bounds for

$$\Pr[|R(g_n) - R_n(g_n)| > \epsilon]$$

which can be rewritten as

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \mathbf{E}[f(x)]\right| > \epsilon\right] \tag{4}$$

Now I reviewed some well-known theorems for deriving bounds for (4). The theorems are widely applied to proofs in Chapter 6.

Hoeffding's Inequality (Hoeffding, 1963)

Theorem .0.3. *Suppose $x_i, i = (1, \ldots, n)$ are n iid. random variables with $f(x_i) \in [a, b]$. Then for any $\varepsilon > 0$, we have*

$$P\left[\left|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \mathbf{E}[f(x)]\right|\right] > \varepsilon] \leq 2\exp\left(-\frac{2n\varepsilon^2}{(b-a)^2}\right)$$

McDiarmid's theorem (McDiarmid, 1989)

Theorem .0.4. *Denote by $f(x_1, \ldots, x_n)$ a function of n independent random variables. Moreover let*

$$|f(x_1, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, x', x_{i+1}, \ldots, x_n)| \leq c_i \tag{5}$$

for all $x_1, \ldots, x_n$ and x'. Denote by $C := \sum_i c_i^2$. In this case

$$\Pr\left\{|f(x_1, \ldots, x_n) - \mathbf{E}_{x_1,\ldots x_n}\left[f(x_1, \ldots, x_n)\right]| > \varepsilon\right\} \leq \exp(-2\varepsilon^2/C). \tag{6}$$

This is a generalization of the Hoeffding Inequality.

Symmetrization Symmetrization can be used together with McDiarmid's theorem to study the quantity

$$\left|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \mathbf{E}f\right|$$

which can be further rewritten as $|\sum_{i=1}^{n} Z_i(f)|$. The difficulty to analyze this quantity is that we don't know the distribution and therefore can not compute $\mathbf{E}f$. The idea of Symmetrization can be used to analyze this quantity. The main idea is that if $\frac{1}{n}\sum_{i=1}^{n} f(x_i)$ is close to $\mathbf{E}f$ for various data $x_1, \ldots, x_n$, then $\frac{1}{n}\sum_{i=1}^{n} f(x_i)$ is close to $\frac{1}{n}\sum_{i=1}^{n} f(x_i')$, given that the empirical average on $x_1', \ldots, x_n'$ that are independent copy of $x_1, \ldots, x_n$. Therefore, if the two empirical averages are far from each other, then empirical error is far from expected error.

Theorem .0.5. *Let $\mathcal{F}$ be a class of functions. Define an empirical process:*

$$Z(x_1, \ldots, x_n) = \sup_{f\in\mathcal{F}}\left|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \mathbf{E}f\right|$$

and a Rademacher Process:

$$R(x_1, \ldots, x_n, \epsilon_1, \ldots, \epsilon_n) = \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i f(x_i)$$

where $\epsilon_1, \ldots, \epsilon_n$ *are iid Rademacher random variables.*[1] *Then*

$$\mathbf{E}Z \leq 2\mathbf{E}R$$

Proof.

$$\begin{aligned}
\mathbf{E}Z &= \mathbf{E}_x \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \mathbf{E}f \right| \\
&= \mathbf{E}_x \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \mathbf{E}'_x \left(\frac{1}{n} \sum_{i=1}^{n} f(x'_i) \right) \right| \\
&= \mathbf{E}_x \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \mathbf{E}'_x \left(\frac{1}{n} \sum_{i=1}^{n} (f(x'_i)|X) \right) \right| \textit{ } (X = x_1, \ldots, x_n \textit{ is independent of } X') \\
&\leq \mathbf{E}_{x,x'} \left(\sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - f(x'_i)) \right| |X \right) \textit{ (convexity of } \sup \textit{ and } |\cdot|) \\
&= \mathbf{E}_x \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - f(x'_i)) \right|
\end{aligned}$$

Now note that the distribution of $f(x_i) - f(x'_i)$ is symmetric around 0, so it has same distribution as $\epsilon_i(f(x_i) - f(x'_i))$ for any fixed ϵ_i. So the above quantity does not change if multiply ϵ_i to any term in the summation. Since this is true for all fixed ϵ_i, we can also

[1] Rademacher random variable has values of -1 or 1 with probability 0.5. $\mathbf{E}R$ is called a *Rademacher Average*.

take expectation over ϵ_i. Therefore, continue with the last quantity,

$$\begin{aligned}
&= \mathbf{E}_x \sup_{f\in\mathcal{F}} \left| \frac{1}{n}\sum_{i=1}^{n} \epsilon_i \left(f(x_i) - f(x_i')\right) \right| \\
&\leq \mathbf{E}_x \sup_{f\in\mathcal{F}} \left| \frac{1}{n}\sum_{i=1}^{n} \epsilon_i f(x_i') \right| + \left| \frac{1}{n}\sum_{i=1}^{n} \epsilon_i f(x_i) \right| \textit{ (triangle inequality)} \\
&\leq \mathbf{E}_x \sup_{f\in\mathcal{F}} \left| \frac{1}{n}\sum_{i=1}^{n} \epsilon_i f(x_i') \right| + \mathbf{E}_x \sup_{f\in\mathcal{F}} \left| \frac{1}{n}\sum_{i=1}^{n} \epsilon_i f(x_i) \right| \textit{ } (\sup_i (a_i + b_i) \leq \sup_i a_i + \sup_i b_i) \\
&= 2\mathbf{E}R
\end{aligned}$$

□

Recall that our goal is to bound Z. The above theorem suggests that to control Z, we need follow two steps: 1) show Z is concentrated around its mean $\mathbf{E}Z$; 2) bound $\mathbf{E}R$ and use the above bound $\mathbf{E}Z \leq 2\mathbf{E}R$. Now we show how to achieve these two things.

1) Use McDiarmid's inequality to show the concentration of Z around $\mathbf{E}Z$.

Lemma .0.6. *Assume $f(x) \in [a, b]$ for all x and $f \in \mathcal{F}$. Then*

$$\Pr(|Z - \mathbf{E}Z| > \epsilon) \leq \exp\left(-n^2\epsilon^2 / \left(2(b-a)^2\right)\right)$$

Proof.

$$\begin{aligned}
& |Z(x_1, \ldots, x_i', \ldots, x_n) - Z(x_1, \ldots, x_i \ldots, x_n)| \\
= & \left| \sup_{f\in\mathcal{F}} |\mathbf{E}f - \frac{1}{n}\sum_{j=1}^{n} f(x_j) + \left(\frac{1}{n}f(x_i) - \frac{1}{n}f(x_i')\right)| - \sup_{f\in\mathcal{F}} |\mathbf{E}f - \frac{1}{n}\sum_{j=1}^{n} f(x_j)| \right| \\
\leq & \sup_{f\in\{} \frac{1}{n} |f(x_i) - f(x_i')| \leq \frac{b-a}{n} = c_i
\end{aligned}$$

Then apply McDiarmid's inequality, we have

$$\Pr(|Z - \mathbf{E}Z| > \varepsilon) \leq \exp\left(\frac{-\varepsilon^2}{2\sum_{i=1}^{n}(b-a)^2 n^2}\right) = \exp\left(-n\varepsilon^2/2(b-a)^2\right)$$

Now let

$$\delta = 2\exp\left(-n\varepsilon^2/2(b-a)^2\right)$$

then,

$$\varepsilon = (b-a)\sqrt{-2\log(\frac{\delta}{2})\frac{1}{n}}$$

which euivalently means with probability at least $1-\delta$

$$|Z - \mathbf{E}Z| < (b-a)\sqrt{-2\log(\frac{\delta}{2})\frac{1}{n}}$$

□

2) Use the Symmetrization,

$$Z \leq \mathbf{E}Z + (b-a)\sqrt{-2\log(\frac{\delta}{2})\frac{1}{n}} \leq 2\mathbf{E}R + (b-a)\sqrt{-2\log(\frac{\delta}{2})\frac{1}{n}}$$

So if we obtain the bound on $\mathbf{E}R$, we know how the random variable Z is concentrated around $\mathbf{E}Z$. It turns out that $\mathbf{E}R$ is always easier to bound than $\mathbf{E}Z$ given the convexity and linearity properties.

Bibliography

S. Agarwal, J. Lim, L. Zelini-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 838–845, 2005.

S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 17–24, 2006.

D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs*. In Preparation, http://stat-www.berkeley.edu/users/aldous/RWG/book.html.

R. Bekkerman, E. El-Yaniv, and A. McCallum. Multiway distributional clustering via pairwise interactions. In *Proceedings of the 21th International Conference on Machine Learning*, pages 41–48, 2005.

M. Belkin and P. Niyogi. Semi-supervised learning on manifolds. In *Machine Learning*, volume 56, pages 209–239, 2004.

M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based method methods. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.

S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19*, 2006a.

S. Ben-David, U. vonLuxburg, and D. Pal. A sober look on clustering stability. In *Proceedings of the 19th Annual Conference on Learning Theory*, pages 5–19, 2006b.

A. BenHur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clutered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

P. Bonacich, A. C. Holdren, and M. Johnston. Hyper-edges and multi-dimensional centrality. In *Social Networks*, volume 26, pages 189–203, 2004.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning Lecture Notes in Artificial Intelligence*, pages 169–207, 2004.

G. Casella and R. Berger. *Statistical Inference*. Duxbury, 2002.

S. Chakrabarti, D. Gibson, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Spectral filtering for resource discovery. In *ACM SIGIR workshop on Hypertext Information Retrieval on the Web*, 1998.

O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Proceedings of the Advances in Neural Information Processing Systems 15*, 2003.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. URL `http://www.kyb.tuebingen.mpg.de/ssl-book`.

N. V. Chawla and G. Karakoulas. Learning from labeled and unlabeled data: An empirical study across techniques and domains. *Journal of Artificial Intelligence Research*, (23): 331–366, 2005.

F. Chung. Laplacian and the cheeer inequality for directed graphs. In *Annals of Cominatorics*, volume 9, pages 1–19, 2005.

F. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. Amer. Math. Soc., Providence, 1997.

T.H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.

F. Cozman, I. Cohen, and M. Cirelo. Unlabeled data can degrade classification performance of generative classifiers. In *Proceedings Fifteenth International Florida Artificial Intelligence Society Conference*, pages 327–331, 2002.

M. Craven, D. DiPasquo, D. Freitag, A. K. McCallum, T. M. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, pages 509–516, 1998.

I. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioing. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001.

I. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003.

C. Ding, X. He, P. Husbands, H. Zha, and H. Simon. PageRank, HITS and a unified framework for link analysis. Technical report, LBNL, 2002.

M. Dudik, R.E. Schapire, and S.J. Phillips. Correcting sample selection bias in maximum entropy density estimation. In *Advances in Neural Information Processing Systems 18*, 2005.

R. M. Dudley. *Real analysis and probability*. Cambridge University Press, 2002.

R. El-Yaniv and O. Souroujon. Iterative double clustering for unsupervised and semi-supervised learning. In *Proceedings of ECML-01, 12th European Conference on Machine Learning*, pages 121–132, 2001.

C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

B. Everitt. *Cluster Analysis*. New York: Halsted Press, 1980.

G. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, 2000.

G. Flake, S. Lawrence, C. L. Giles, and F. Coetzee. Self-organization and identification of web communities. 35:66–71, 2002.

B. Gao, T. Liu, X. Zheng, Q. Cheng, and W. Ma. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 41–50, 2005.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *UK Conference on Hypertext*, pages 225–234, 1998.

D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. In *VLDB Journal: Very Large Data Bases*, volume 8, pages 222–236, 2000.

E. Gine and V. Koltchinskii. Empirical graph Laplacian approximation of laplace-beltrami operators: large sample results. In *Proceedings of the 4th International Conference on High Dimensional Probability*, 2005.

S. Gruvberger, M. Ringner, Y. Chen, S. Panavally, L.H. Saal, A. Borg, M. Ferno, C. Peterson, and P. S. Meltzer. Estrogen receptor status in breast cancer is associated with remarkably distinct gene expression patterns. In *Cancer Res*, volume 61, pages 5979–5984, Aug 2001.

M. Gu, H. Zha, C. Ding, X. He, and H. Simon. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering, 2001.

L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning and clustering. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, volume 11, pages 1074–1085, 1992a.

L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, volume 11, pages 1074–1085, 1992b.

J. Heckman. sample selection bias as a specification error. In *Econometrica*, 1979.

M. Hein. Uniform convergence of adaptive graph-based regularization. In *Proceedings of the 19th Annual Conference on Learning Theory*, pages 50–64, 2006.

M. Hein, J. Y. Audibert, and U. von Luxburg. From graphs to manifolds-weak and strong pointwise consistency of graph Laplacians. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 470–485, 2005.

M.R. Henzinger. Algorithmic challenges in web search engines. In *Internet Mathematics*, volume 1, pages 115–123, 2003.

M.R. Henzinger. Hyperlink analysis for the web. In *IEEE Internet Computing*, volume 5, pages 45–50, 2001.

H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Tagaki. Assessment of prediction accuracy of protein function from protein-protein interaction data. In *Yeast*, volume 18, pages 523–531, 2001.

W. Hoeffding. Probability inequalities for sums of bounded random varianbles. In *Journal of the American Statistical Association*, 1963.

J. Huang. A combinatorial view of the graph Laplacians. Technical Report 144, MPI, 2005.

J. Huang, A. J. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 18*, 2006a.

J. Huang, A. J. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. Technical report, CS-2006-44, University of Waterloo, 2006b.

J. Huang, T. Zhu, R. Greiner, D. Zhou, and D. Schuurmans. Information marginalization on subgraphs. In *10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 4213, pages 199–210, 2006c.

J. Huang, T. Zhu, and D. Schuurmans. Web communities identification from random walks. In *10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 4213, pages 187–198, 2006d.

H. Ino, M. Kudo, and A. Nakamura. Partitioning of web graphs by community topology. In *Proceedings of the 14th international conference on World Wide Web*, 2005.

T. Ito and et al. A comprehensive two-hybrid analysis to explore the yeast protein interactome. In *PNAS*, volume 98, pages 4596–4574, 2001.

M. Kessler. Bibliographic coupling between scientific papers. In *American Documentation*, 1963.

J. M. Kleinberg. authoritative sources in a hyperlinked environment. In *Journal of the ACM*, volume 46, pages 604–632, 1999.

T. Lange, V. Roth, M. Braun, and J. Buhmann. Stability-based validation of clustering solutions. In *Neural Computation*, volume 16, pages 1299–1323, 2004.

R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. In *WWW*, pages 387–401, 2000.

T. Lengauer. *Combinatorial algorithms for integrated circuit layout*. New York: Wiley, 1990.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. In *Machine Learning*, 2002.

L. Lovasz. Random walks on graphs: a survey. In *Combinatorics, Paul Erdos is Eighty*, volume 2, pages 353–397, Budapest, 1996. Janos Bolyai Math. Soc.

H. Lutkepohl. *Handbook of Matrices*. Chichester: Wiley, 1997.

D. W. Matula and F. Shahrokhi. Sparsest cuts and bottlenecks in graphs. In *Journal of Disc. Applied Math.*, volume 27, 1990.

C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics 1989*, pages 148–188. Cambridge University Press, 1989.

M. Meila and W. Pentney. Clustering by weighted cuts in directed graphs. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, 2007.

M. Meila and J. Shi. A random walks view of spectral segmentation. In *8th International Workshop on Artificial Intelligence and Statistics*, 2001.

S. Mendelson. A few notes on statistical learning theory. In *Advanced Lectures in Machine Learning*, volume LNCS 2600, pages 1–40. Springer, 2003.

B. Mohar. The Laplacian spectrum of graphs. In *Graph theory, combinatorics, and applications*, volume 2, 1991.

B. Mohar. Some applications of laplace eigenvalues of graphs. In *Graph Symmetry: Algebraic methods and applications*, volume NATO ASI Ser. C497, 1997.

V. A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer-Verlag, New York, NY, 1984.

A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the web. In *Technical Report*. Standford University, 1998.

W. Pentney and M. Meila. Spectral clustering of biological sequence data. In *Aritifical Intelligence and Statistics AISTATS*, 2001.

P. Perona and W. Freeman. A factorization approach to grouping. In *Proceedings of the 5th European Conference on Computer Vision*, volume 1, pages 655–670, 1998.

A. Pothen, H. Simon, and K.P. Liou. Partitioning sparse matrices with eigenvector of graphs. In *SIAM Journal of Mathematical Analysis and Applications*, volume 11, pages 430–452, 1990.

J. Rennie and N. Srebro. Loss functions for preference levels: regression with discrete ordered labels. In *IJCAI Multidisciplinary Workshop on Advances in Prefenrece Handling*, 2005.

S. Rosset, J. Zhu, H. Zou, and T. Hastie. A method for inferring label sampling mechanisms in semi-supervised learning. In *In Advances in Neural Information Processing Systems 17*, pages 1161–1168, 2005.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. In *Nature Biotechnology*, volume 18, pages 1257–1261, 2000.

J. Shi and J. Malik. Normalized cuts and image segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22, pages 888–905, 2000.

H. Shimodaira. Improving predictive inference under convariance shift by weighting the log-likelihood function. In *Journal of Statistical Planning and Inference*, volume 90, pages 247–244, 2000.

D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. DAmico, and J. Richie. Gene expression correlates of clinical prostate cancer behavior. In *Cancer Cell*, volume 1, 2002.

I. Steinwart. support vector machines are universally consistent. In *Journal of Complexity*, 2002b.

A. J. Storkey and M. Sugiyama. Mixture regression for covariance shift. In *Advances in Neural Information Processing Systems 19*, 2006.

M. Sugiyama and K.R. Muller. Input-dependent estimation of generalization error under covariance shift. In *Statistics and Decisions*, volume 23, pages 249–279, 2005.

M. Szummer and T. Jaakkola. Information regularization with partially labeled data. In *In Advances in Neural Information processing systems 15*, pages 1025–1032, 2002.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. In *J. Royal. Statist. Soc. B*, volume 63, pages 411–423, 2001.

A.N. Tikhonov and V.Y. Arsenin. *Solutions for Ill-Posed Problems*. Winston, Washinton, 1977.

N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings 37th Allerton Conference*, 1999.

P. Uetz and et al. A comprehensive analysis of protein-pertein interactions in saccharomyces cerevisiae. In *Nature*, volume 403, pages 623–627, 2000.

V.N. Vapnik. *Statistical Learning Theory*. Wiley, NY, 1998.

A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global protein function prediction from protein-protein interaction networks. In *Nature Biotechnology*, volume 21, pages 697–700, 2003.

G. Wahba. *Smoothing and Ill-Posed Problems: Solutions Methods for Integral Equations and Applications*. Plenum Press, New York, 1979.

G. Wahba. *Spline Models for Observational Data*. Number 59 in CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, 1990.

P. Warnat, R. Eils, and B. Brors. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. In *BMC Bioinformatics*, volume 6, page 265, Nov 2005.

M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. r. Olson, J. R. Marks, and J. R. Nevins. Predicting the clinical status of human breast cancer

by using gene expression profiles. In *Proc Natl Acad Sci U S A*, volume 98, pages 11462–11467, Sep 2001.

A. Yuille and N. Grzywacz. The motion coherence theory. In *Proceedings of the International conference on Computer Vision*, pages 344–353, Washington, D.C., December 1988. IEEE Computer Society Press.

B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceeding of the 20 th International Conference on Machine Learning*, 2004.

H. Zha, X. He, C. Ding, and M. Gu. Bipartite graph partitioning and data clustering. In *Proceedings of ACM CIKM 2001*, 2001.

T. Zhang and F. J. Oles. Text categorization based on regularized linear classification methods. In *Information Retrieval*, volume 4, pages 5–31, 2001.

D. Zhou and B. Schölkopf. Learning from labeled and unlabeled data using random walks. In *DAGM'04: 26th Pattern Recognition Symposium*, August 2004.

D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Proceedings of the 27th DAGM Symposium*, pages 361–368, 2005.

D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2004.

D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22th International Conference on Machine Learning*, 2005a.

D. Zhou, B. Schölkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In *Advances in Neural Information Processing Systems 17*, 2005b.

D. Zhou, J. Huang, and Schölkopf. Learning with hypergraphs: clustering, classification, and embedding. In *Advances in Neural Information Processing Systems 18*, 2006.

X. Zhu. Semi-supervised learning literature survey. Technical Report Computer Science TR 1530, University of Wisconsin-Madison, 2006.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.