DEPARTMENT OF STATISTICS
University of Wisconsin
1300 University Ave.
Madison, WI 53706

TECHNICAL REPORT NO. 1145

15 August 2008

Graph-Based Data Analysis:

Tree-Structured Covariance Estimation, Prediction by Regularized Kernel Estimation and Aggregate Database Query Processing for Probabilistic Inference

Héctor Corrada Bravo[1]

Department of Computer Sciences, University of Wisconsin, Madison WI

[1]Research supported in part by NIH Grant EY09946, NSF Grant DMS-0604572 and ONR Grant N0014-06-0095 and a Ford Foundation Predoctoral fellowship from the National Academies

GRAPH-BASED DATA ANALYSIS:
TREE-STRUCTURED COVARIANCE ESTIMATION, PREDICTION BY REGULARIZED KERNEL ESTIMATION AND AGGREGATE DATABASE QUERY PROCESSING FOR PROBABILISTIC INFERENCE

by

Héctor Corrada Bravo

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

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2008

A Jero, Mila, Talía y Lucas

ACKNOWLEDGMENTS

I would like to thank my advisors Grace Wahba and Raghu Ramakrishnan and the other members of my thesis committee: Jerry Zhu, Sündüz Keleş and Stephen Wright.

Parts I and II was joint work with Grace Wahba while Part III was joint work with Raghu Ramakrishnan. Sündüz Keleş collaborated in the work of Chapters 2 and 4, while Stephen Wright and Kevin Eng collaborated in the work of Chapter 2 as well. Kristine Lee, Barbara Klein and Ronald Klein of the Ophthalmology Department at the University of Wisconsin-Madison and Sudha K. Iyengar at Case Western Reserve University collaborated in the work presented in Chapter 3.

The great current and past colleagues of the Thursday spline group provided excellent feedback and ideas: Weiliang Shi, Hyonho Chun, Kevin Eng, Pei-Fen Kuan, Xiwen Ma, Alina Andrei, John Carew, Bin Dai and Xin Li, with a special acknowledgement to Fan Lu for starting the RKE work. My colleagues in the Database group also provided great camaraderie and assistance: Kristen Lefevre, Lei Chen, Bee-Chung Chen, Doug Burdick, Jiansheng Huang, Eric Robinson, Eric Chu and Fei Chen. I would like to extend special thanks to Ted Wild for particularly insightful conversation and help.

My most heartfelt gratitude must go to my wife Talía, parents Milagros and Jerónimo, and son Lucas. Without them, this process would not have been possible, and it would not have been as fun as it was.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

Page

Appendix

Page

Appendix

Page

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure Page

GRAPH-BASED DATA ANALYSIS:
TREE-STRUCTURED COVARIANCE ESTIMATION, PREDICTION BY REGULARIZED KERNEL ESTIMATION AND AGGREGATE DATABASE QUERY PROCESSING FOR PROBABILISTIC INFERENCE

Héctor Corrada Bravo

Under the supervision of Professors Grace Wahba and Raghu Ramakrishnan
At the University of Wisconsin-Madison

This dissertation presents a collection of computational techniques for the analysis of data where relationships between objects can be expressed through a graph. Data of this type can be found in many and diverse settings, including genomic and epidemiological applications, web search, social networking and decision making. Although taking relationships into account makes analysis of this type of data more challenging, the graph structure of these relationships can be used to make this analysis viable. In this dissertation, we implement a number of techniques for analyzing this type of data using well-known and tested computational tools. Furthermore, we explore these techniques over a wide array of biological and decision making applications.

In Part I, we present a method for estimating tree-structured covariance matrices directly from observed continuous data. Tree-structured covariance matrices encode probabilistic relationships between objects that can be described by rooted trees. In this case, we directly estimate graph structure from observed data under a specific probabilistic model.

Part II presents a methodology for graph-based prediction where a predictive model is estimated over data where relationships between objects are encoded by a known graph. We make extensive use of Regularized Kernel Estimation (Lu et al., 2005), a framework for estimating a positive semidefinite kernel from noisy, incomplete and inconsistent distance data. In this case, the graph structure of the data is used to define a distance from which a kernel matrix is estimated.

Finally, in Part III, we present techniques for efficiently evaluating aggregate queries of a particular type over views defining a large number of database records. The main assumption is that this view is the result of a stylized join over a number of much smaller tables, and is described by a graph. We make use of this graph structure to reduce the cost of single query evaluation and to cache intermediate results in a query workload setting. This framework was designed in part to address scalable probabilistic inference in relational databases.

Grace Wahba and Raghu Ramakrishnan

ABSTRACT

This dissertation presents a collection of computational techniques for the analysis of data where relationships between objects can be expressed through a graph. Data of this type can be found in many and diverse settings, including genomic and epidemiological applications, web search, social networking and decision making. Although taking relationships into account makes analysis of this type of data more challenging, the graph structure of these relationships can be used to make this analysis viable. In this dissertation, we implement a number of techniques for analyzing this type of data using well-known and tested computational tools. Furthermore, we explore these techniques over a wide array of biological and decision making applications.

In Part I, we present a method for estimating tree-structured covariance matrices directly from observed continuous data. Tree-structured covariance matrices encode probabilistic relationships between objects that can be described by rooted trees. In this case, we directly estimate graph structure from observed data under a specific probabilistic model.

Part II presents a methodology for graph-based prediction where a predictive model is estimated over data where relationships between objects are encoded by a known graph. We make extensive use of Regularized Kernel Estimation (Lu et al., 2005), a framework for estimating a positive semidefinite kernel from noisy, incomplete and inconsistent distance data. In this case, the graph structure of the data is used to define a distance from which a kernel matrix is estimated.

Finally, in Part III, we present techniques for efficiently evaluating aggregate queries of a particular type over views defining a large number of database records. The main assumption is that this view is the result of a stylized join over a number of much smaller tables, and is described by a graph. We make use of this graph structure to reduce the cost of single query evaluation and to

cache intermediate results in a query workload setting. This framework was designed in part to address scalable probabilistic inference in relational databases.

Chapter 1

Introduction

This dissertation presents a collection of computational techniques for the analysis of data where relationships between objects can be expressed through a graph[1]. Data of this type can be found in many and diverse settings, including genomic and epidemiological applications, web search, social networking and decision making. Although taking relationships into account makes analysis of this type of data more challenging, the graph structure of these relationships can be used to make this analysis viable. In this dissertation, we implement a number of techniques for analyzing this type of data using well-known and tested computational tools. Furthermore, we explore these techniques over a wide array of biological and decision making applications.

Data analysis comprises a large continuum, including querying, prediction and estimation. Presented in this dissertation are methods for the analysis of graph based data in each of these broad areas. In Part I, we present a method for estimating tree-structured covariance matrices directly from observed continuous data. Tree-structured covariance matrices encode probabilistic relationships between objects that can be described by rooted trees. In this case, we directly estimate graph structure from observed data under a specific probabilistic model. We use our methods in a case study analyzing gene expression from yeast gene families. We are able to verify existing results on the presence of phylogenetic influence in expression under a number of experimental conditions, as well as presenting evidence that estimating tree-structured covariance matrices directly from

[1]Throughout this dissertation, we use the standard definition of a graph as a tuple $G = (V, E)$, where V is a set of *nodes*, usually representing data objects, and E a set of edges, representing relationships between data objects. Edges are usually associated with a real number, further quantifying the relationship between objects.

the observed gene expression can guide investigators in their modelling choices for phylogenetic comparative analysis (Chapter 2).

Part II presents a methodology for graph-based prediction where a predictive model is estimated over data where relationships between objects are encoded by a known graph. In one case, we make use of graph structure encoding familial relationships to extend previously-used semi-parametric models of eye disease risk (Chapter 3). In the other, we address a protein prediction task using only graph structure, that is, there are no other features describing the data beyond the relationships encoded by a given graph (Chapter 4). In both cases, we make use of Regularized Kernel Estimation (Lu et al., 2005), a framework for estimating a positive semidefinite kernel from noisy, incomplete and inconsistent distance data. The graph structure of the data is used to define a distance from which a kernel matrix is estimated.

Finally, in Part III we present techniques for efficiently evaluating stylized aggregate queries over views defining a large set of database records. Our main assumption is that this view is the result of a stylized join over a number of much smaller tables, and that this operation is described by a graph (Chapter 5). We make use of this graph structure to reduce the cost of single query evaluation (Chapter 6) and to cache intermediate results in a query workload setting (Chapter 7). This framework was designed in part to address scalable probabilistic inference in relational databases.

The remainder of this introductory chapter provides further detail on each of the computational techniques described above and concludes with some general remarks regarding the work presented in this dissertation.

1.1 Estimating Tree-Structured Covariance Matrices

We present a novel method for estimating tree-structured covariance matrices directly from observed continuous data. A representation of these classes of matrices as linear combinations of rank-one matrices indicating object partitions is used to formulate estimation as instances of well-studied numerical optimization problems.

In particular, we present estimation based on projection where the covariance estimate is the nearest tree-structured covariance matrix to an observed sample covariance matrix. The problem is

posed as a linear or quadratic mixed-integer program (MIP) where a setting of its integer variables specifies a set of tree topologies for the structured covariance matrix. We solve these problems to optimality using efficient and robust existing MIP solvers. We also show that the least squares distance method of Fitch and Margoliash (1967) can be formulated as a quadratic MIP and thus solved exactly using existing, robust branch-and-bound MIP solvers.

1.1.1 Application to phylogenetic analysis of gene expression data

Our motivation for this method is the discovery of phylogenetic structure directly from gene expression data. Recent studies have adapted traditional phylogenetic comparative analysis methods to expression data (Fay and Wittkopp, 2007; Gu, 2004; Oakley et al., 2005; Rifkin et al., 2003; Whitehead and Crawford, 2006). Typically, these methods estimate a phylogenetic tree from genomic sequence data and then perform analysis of expression data using a covariance matrix constructed from the sequence-derived tree to correct for the lack of independence in phylogenetically related taxa. Given recent results on the sensitivity of sequence-derived trees to the genomic region chosen to build them, we propose a stable method for deriving tree-structured covariance matrices directly from gene expression as an exploratory step that can guide investigators in their modelling choices for these types of comparative analysis.

We present a case-study in phylogenetic analysis of expression in yeast gene families. Our method is able to corroborate the presence of phylogenetic structure in the response of expression in certain gene families under particular experimental conditions. On the other hand, when used in conjunction with transcription factor occupancy data, our methods show that alternative modelling choices should be considered when creating sequence-derived trees for this comparative analysis.

1.1.2 Contributions

The contributions of this work are the following:

1. defines a representation for tree-structured covariance matrices that make formulating estimation problems as numerical optimization problems possible;

2. defines a class of estimation problems based on projection to the set of tree-structured covariance matrices of an observed sample covariance matrix;

3. shows that projection-based estimation for problems with known tree topology are instances of linear or quadratic optimization programs depending on the projection norm used;

4. shows that projection-based estimation for problems with unknown tree-topology can be cast as linear or quadratic mixed integer programs depending on the projection norm used;

5. shows how this method can be successfully used to guide investigators carrying out phylogenetic comparative analysis by presenting a case study using an existing yeast gene-family analysis data set.

1.2 Graph-Based Prediction

We look at the Regularized Kernel Estimation (RKE) framework of Lu et al. (2005) as a methodology for building predictive models of graph-based data. RKE is a robust method for estimating dissimilarity measures between objects from noisy, incomplete, inconsistent and repetitious dissimilarity data. It is particularly useful in a setting where object classification is desired but objects do not easily admit description by fixed length feature vectors. Instead, there is access to a source of noisy, and possibly incomplete dissimilarity information between objects given by a graph.

RKE estimates a symmetric positive semidefinite kernel matrix K that induces a real squared distance admitting of an inner product. K is the solution to an optimization problem with semidefinite constraints that trades-off fit of the observed dissimilarity data and a penalty on the complexity of K of the form $\lambda_{rke}\mathrm{trace}(K)$, for positive regularization parameter λ_{rke}.

The RKE framework also provides the *newbie* method for embedding new objects into a low dimensional space induced by an RKE kernel K estimated from a training set of objects. The embedding is given as the solution of an optimization problem with semidefinite and second-order cone constraints. This method requires setting the dimensionality of the embedding space as a parameter.

1.2.1 Extending Smoothing Spline ANOVA Models with Pedigree Data

We present a novel method for incorporating pedigree data into smoothing spline ANOVA (SS-ANOVA) models. By expressing pedigree data as a positive semidefinite kernel matrix, the SS-ANOVA model is able to estimate a function over the sum of reproducing kernel Hilbert spaces: one or more representing information from environmental and/or genetic covariates for each subject and another representing pedigree relationships.

We propose a number of methods for creating positive semidefinite kernels from pedigree information, including the use of Regularized Kernel Estimation (RKE).

We present results on pigmentary abnormalities (PA) in the Beaver Dam Eye Study. Pigmentary abnormalities are a precursor to age-related macular degeneration (AMD), a leading cause of vision loss in the western world for people 60 years or older. A number of recent results have shown strong linkage between two genes (complement factor H, CFH and the ARMS2 gene) and AMD. Furthermore, known environmental risk factors have been identified for both AMD and PA. Further studies have shown that there is a familial component to both AMD and PA.

All of these results make combining these sources of information into a predictive model compelling. We have access to all three of this type of data, genetic marker data for the two genes, environmental risk factors, and familial pedigrees. Our goal is to extend existing SS-ANOVA models for PA with this data.

Our methodology both corroborates known facts about the epidemiology of this disease and reveals surprising results regarding the predictive ability of models that only include components for genetic markers and familial effects. In particular, it shows that a SS-ANOVA model containing terms for only genetic marker and familial components has the same predictive ability of an SS-ANOVA model containing terms for genetic markers and environmental covariates.

1.2.2 Protein Classification by Regularized Kernel Estimation

A setting where RKE can be especially useful is the classification of protein sequence data where measures of dissimilarity are easily obtained, but feature vector representations are difficult to obtain or justify. Some sources of dissimilarity in this case, such as BLAST (Altschul et al.,

1990), require setting a number of parameters that makes the resulting dissimilarities possibly inexact, inconsistent and noisy. The RKE method is robust to the type of noisy and incomplete data that arises in this setting.

We show how RKE can be used to successfully classify proteins in two different tasks using two very different sources of dissimilarity information. In the first, alignment of protein sequence data is used to generate dissimilarities (Section 4.3.1), while in the second, transcription factor occupancy data from the promoter region of genes is used (Section 4.3.2).

1.2.3 Tuning Procedures

This dissertation also presents results on methods for choosing values of the regularization parameter λ_{rke} of the RKE problem. We show the CV2 method for selecting regularization parameter values in clustering and visualization applications. We also describe a method for combining RKE with Support Vector Machines for object classification based on dissimilarity data. Based on an empirical study we make two main observations: 1) for clustering applications, the performance of estimated kernels is similar for large ranges of regularization parameters, suggesting that coarse tuning methods might be sufficient in these cases, and 2) the opposite holds for some classification applications, where good performance is highly dependent on the RKE regularization parameter. This suggests the need for methods that jointly tune regularization parameters in both the RKE and classification optimization problems (Appendix A, and Chapter 8).

To address this tuning problem in the classification setting for RKE, we analyze and compare a number of tuning methods for Support Vector Machines (SVMs). We hope that these methods can be extended to address the RKE tuning problem efficiently. These methods are based on bounding or approximating the Leave-One-Out estimate of misclassification rate. However, the cost of using these methods varies considerably. We show under which conditions are these methods equivalent, and thus provide a way of determining if the additional cost of using a particular method is admissible (Appendix B).

1.2.4 Contributions

The contributions of this work are the following:

1. extends Smoothing-Spline ANOVA models to include terms encoding relationships of graph-based data;
2. shows how this extension can be used in an eye disease risk modelling task where pedigree data encodes familial relationships between subjects;
3. shows how the Regularized Kernel Estimation framework can be used to classify proteins in two different tasks using diverse dissimilarity measures;
4. shows the apparent insensitivity of RKE for clustering tasks to the value of its regularization parameter;
5. also shows the apparent sensitivity of RKE to values of its regularization parameter when used in classification tasks;
6. characterizes and compares a number of adaptive tuning methods for Support Vector Machines.

1.3 MPF Aggregate Database Queries and Probabilistic Inference

Recent proposals for managing uncertain information require the evaluation of probability measures defined over a large number of discrete random variables. This document presents MPF (Marginalize a Product Function) queries, a broad class of relational aggregate queries capable of expressing this probabilistic inference task. By optimizing query evaluation in the MPF setting we provide direct support for scalable probabilistic inference in database systems. Further, looking beyond probabilistic inference, we define MPF queries in a general form that is useful for Decision Support, and demonstrate this aspect through several illustrative queries.

The MPF setting is based on the observation that functions over discrete domains are naturally represented as relations where an attribute (the value, or measure, of the function) is determined by

the remaining attributes (the inputs, or dimensions, to the function) via a Functional Dependency (FD). We define these *Functional Relations*, and present an extended Relational Algebra to operate on them. A view V can then be created in terms of a stylized join of a set of 'local' functional relations such that V defines a joint function over the union of the domains of the 'local' functions. MPF queries are a type of aggregate query that computes view V's joint function value in arbitrary subsets of its domain:

```
select Vars, Agg(V[f]) from V group by Vars.
```

We optimize the evaluation of MPF queries by extending existing database optimization techniques for aggregate queries to the MPF setting. In particular, we show how a modification to the algorithm of Chaudhuri and Shim (1994, 1996) for optimizing aggregate queries yields significant gains over evaluation of single MPF queries in current systems. We also extend existing probabilistic inference techniques such as Variable Elimination, Junction Trees and Belief Propagation to develop novel optimization techniques for single MPF queries, or expected workloads of MPF queries. To the best of our knowledge, we present the first approaches to probabilistic inference that provide scalability and cost-based query evaluation. We present an empirical evaluation of these optimization techniques in a modified PostgreSQL system (Chapter 6).

1.3.1 Optimization of MPF Queries

Like usual aggregate queries over views, there are two options for evaluating an MPF query: 1) the relation defined by view V is materialized, and queries are evaluated directly on the materialized view; or, 2) each query is rewritten using V's definition and then evaluated, so that constructing the relation defined by V is an intermediate step. The first approach requires that the materialized view is updated as base relations change. In the latter, the problem of view maintenance is avoided, but this approach is prohibitive if computing V's relation is too expensive. The rewriting option is likely to be appropriate for answering individual queries, and variations of the former might be appropriate if we have knowledge of the anticipated query workload. In this dissertation, we apply the query rewrite approach to the problem of evaluating single MPF queries

(Chapter 6), and a variant of the view materialization approach to the problem of evaluating expected MPF query workloads (Chapter 7).

Chaudhuri and Shim (1994, 1996) define an algorithm for optimizing aggregate query evaluation based on pushing aggregate operations inside join trees. We present and evaluate an extension of their algorithm and show that it yields significant gains over evaluation of MPF queries in existing systems (see Section 6.5). We also present and evaluate the Variable Elimination (VE) technique (Zhang and Poole, 1996) from the literature on optimizing probabilistic inference and show similar gains over existing systems. Additionally, we present extensions to VE based on ideas in the Chaudhuri and Shim algorithm that yield better plans than traditional VE. Finally, we extend these techniques in the context of view materialization to evaluate expected MPF query workloads (Chapter 7).

1.3.2 Contributions

The contributions of this work are the following:

1. introduces MPF queries, which significantly generalize the relational framework introduced by Wong (2001) for probabilistic models. With this generalized class of queries, probabilistic inference can be expressed as a query evaluation problem in a relational setting. MPF queries are also motivated by decision support applications;

2. extends the optimization algorithm of Chaudhuri and Shim for aggregate queries to the MPF setting, taking advantage of the semantics of functional relations and the extended algebra over these relations. This extension produces better quality plans for MPF queries than those given by the procedure in Chaudhuri and Shim (1994, 1996);

3. builds on the connection to probabilistic inference and extend existing inference techniques to develop novel optimization techniques for MPF queries. Even for the restricted class of MPF queries that correspond to probabilistic inference, to the best of our knowledge this is the first approach that addresses scalability and cost-based plan selection;

4. further extends these techniques to efficiently evaluate expected workloads of MPF queries;

5. implements our optimization techniques in a modified PostgreSQL system, and presents an empirical evaluation that demonstrates significant performance improvement.

Finally, we remark that the techniques introduced so far apply to the problem of scaling *exact* probabilistic inference. This is required in settings where results are composed with other functions that are not monotonic with respect to likelihood, including systems that compute expected risk or utility. In these settings approximate probability values are not sufficient. However, for other systems where only relative likelihood suffices, e.g., ranking in information extraction, approximate inference procedures (Wainwright and Jordan, 2003; Weiss, 2000; Yedidia et al., 2002) are sufficient and may be more efficient. We address some preliminary ideas in this direction in Chapter 9.

1.4 General Remarks

There are two general themes that, for the most part, characterize the work presented in this dissertation. First, existing computational tools are used in novel ways to address the problems defined. In estimating tree-structured covariance matrices we make use of robust existing solvers for linear and quadratic, continuous and mixed integer programming. Once an amenable representation for this class of matrices was defined, existing solvers were easily used to carry out estimation. In the Regularized Kernel Estimation framework, existing semidefinite solvers are used. Finally, in evaluating MPF queries, we make extensive use of existing query optimization techniques while adapting them to our specific setting.

The other general theme is that problems are defined over real-world applications and tested on real data sets. These include yeast gene expression data, data from a large epidemiological study of eye disease and protein dissimilarity measures. In the case of MPF queries, we present a real-world-viable decision making and probabilistic inference applications.

A by-product of this dissertation is a set of programs that have general impact beyond the techniques implemented in this dissertation. For example, an interface to the CPLEX optimization engine (Ilog, SA, 2003) is now publicly available for the `R` statistical computing framework (R

Development Core Team, 2007) as a result of the work on tree-structured covariance matrices (Corrada Bravo, 2008). An interface to R was also created for the CSDP semidefinite solver (Borchers, 1999), which will be made available in the near future along with an R package implementing the RKE framework used for this work. The implementation of MPF query evaluation required extending the optimization engine of the PostgreSQL database management system to evaluate general aggregate queries more efficiently, beyond the MPF setting. These extensions will be made available to the PostgreSQL system in the near future.

The dissertation concludes with two chapters on extensions of the work presented in the first seven chapters. Chapter 8 sketches an extension to the RKE framework where a trade-off between a regression objective and distance fit is optimized directly. It also shows a general methodology for deriving leave-one-out approximations adaptive tuning criteria for estimates obtained by solving linear semidefinite programs. Chapter 9 discusses further future work.

Part I

Estimating Tree-Structured Covariance Matrices

Chapter 2

Estimating Tree-Structured Covariance Matrices via Mixed-Integer Programming with an Application to Phylogenetic Analysis of Gene Expression

2.1 Introduction

Recent studies have adapted existing techniques in population genetics to perform evolutionary analysis of gene expression (Fay and Wittkopp, 2007; Gu, 2004; Oakley et al., 2005; Rifkin et al., 2003; Whitehead and Crawford, 2006). In particular, corrections for evolutionary dependence between taxa, e.g. species or strains, are used in regression (generalized least squares) or other likelihood models. These phylogenetic corrections are a well accepted methodology in phenotypic modeling (Felsenstein et al., 2004), since, without them, statistical analysis is subject to increased false positive rates and decreased power for hypothesis tests. These corrections take the form of a covariance matrix corresponding to a random diffusion process along a phylogenetic tree.

These studies assume that the single phylogenetic tree structure underlying the data is known, normally derived from DNA or amino acid sequence data. While this assumption might be valid for the analysis of *coarse* traits–beak size in birds, for example–as previously used in comparative phylogenetic studies, it might prove too restrictive when carrying out similar analysis at the genomic level taking into account recent findings of high variability in tree topology and branch length estimates contingent on the genomic region used to estimate the phylogeny (Frazer et al., 2004; Habib et al., 2007; Yalcin et al., 2004). If we are interested in a particular group of genes, given that they are spread throughout the genome, it makes more sense to develop a covariance estimate appropriate to those genes. We present a principled way of estimating tree-structured

covariance matrices directly from sample covariances of observed gene expression data. As an exploratory step, this can help investigators circumvent issues that arise from estimating a global phylogeny from sequence in an independent previous step.

In this chapter, we formulate the problem of estimating a tree-structured covariance matrix as mixed-integer programs (MIP) (Bertsimas and Weismantel, 2005; Wolsey and Nemhauser, 1999). In particular, we look at projection problems that estimate the nearest matrix in the structured class to the observed sample covariance. These problems lead to linear or quadratic mixed integer programs for which algorithms for global solutions are well-known and reliable production code exists. The formulation of these problems hinges on a representation of tree-structured covariance matrices as a linear expansion of outer products of indicator vectors specifying nested partitions of objects.

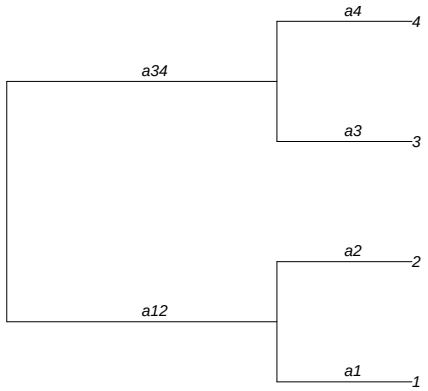
The chapter is organized as follows: in Section 2.2 we formulate the representation of tree structured covariance matrices and give some results regarding the space of such matrices; Section 2.4 shows how to define the constraints that ensure matrices are tree-structured as constraints in mixed-integer programs (MIPs); projection problems are specifically addressed in Section 2.4.3; we present our results on a case-study on phylogenetic analysis of expression in yeast gene families in Section 3.5; a discussion, including related work, follows in Section 3.7. Appendix 2.9 presents simulation results on estimating the tree topology from observed data that show that show how our MIP-based method compares favorably to the the well-known Neighbor-Joining(Saitou, 1987) method using distances computed from the observed covariances.

2.2 Tree-Structured Covariance Matrices

Our object of study are covariance matrices of diffusion processes defined over trees (Cavalli-Sforza and Edwards, 1967; Felsenstein et al., 2004). Usually, a Brownian motion assumption is made on the diffusion process where steps are independent and normally distributed with mean zero. However, covariance matrices of diffusion process with independent steps, mean zero and finite variance will also have the structure we are studying here. We do not make any normality assumptions on the diffusion process and, accordingly, fit covariance matrices by minimizing a

projection objective instead of maximizing a likelihood function. Thus, for a tree $\mathcal{T}$ defined for p objects, our assumption is that the observed data are realizations of a random variable $Y \in \mathbb{R}^p$ with $\mathrm{Cov}(Y) = B$, where B is a tree-structured covariance matrix defined by $\mathcal{T}$.

Figure 2.1 shows a tree with 4 leaves, corresponding to a diffusion process for 4 objects. A rooted tree defines a set of nested partitions of objects such that each node in the tree (both interior and leaves) corresponds to a subset of these objects. In our example, the lower branch exiting the root corresponds to subset $\{1, 2\}$. The root of the tree corresponds to the set of all objects and each leaf corresponds to singleton sets. The subset corresponding to an interior node is the union of the non-overlapping subsets of that node's children. Edges are labeled with real numbers indicating tree branch lengths.



(a)

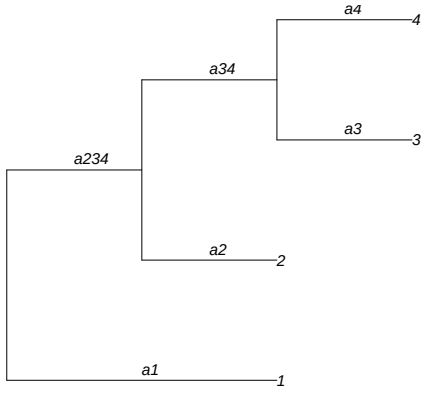
$$B = \begin{pmatrix} a_{12} + a_1 & a_{12} & 0 & 0 \\ a_{12} & a_{12} + a_2 & 0 & 0 \\ 0 & 0 & a_{34} + a_3 & a_{34} \\ 0 & 0 & a_{34} & a_4 \end{pmatrix}$$

(b)

Figure 2.1 A schematic example of a phylogenetic tree and corresponding covariance matrix. The root is the leftmost node, while leaves are the rightmost nodes. Branch lengths are arbitrary nonnegative real numbers.

Denoting $B = \mathrm{Cov}(Y)$, entry B_{ij} is the sum of branch lengths for the path starting at the root and ending at the last common ancestor of leaves i and j. In our example, $B_{12} = a_{12}$ is the length of the branch from the root to the node above leaves 1 and 2. For leaf i, B_{ii} is the sum of the branch lengths of the path from root to leaf. The covariance matrix B for our example tree is

given in Figure 2.1(b). If we swap the positions of labels 3 and 4 in our example tree such that label 3 is the topmost label and construct a covariance matrix accordingly we recover the same matrix B as before. In fact, any tree that specifies this particular set of nested partitions generates the same covariance matrix. All trees that define the same set of nested partitions are said to be of the same topology, and we say that covariance matrices that are generated from trees with the same topology belong to the same class. However, a tree that specifies a different set of nested partitions generates a different class of covariance matrices. For example, Figure 2.2 shows a tree that defines a different set of nested partitions and the matrix it generates.



(a)

$$A = \begin{pmatrix} a_1 & 0 & 0 & 0 \\ 0 & a_{234} + a_2 & a_{234} & a_{234} \\ 0 & a_{234} & a_{234} + a_{34} + a_3 & a_{34} \\ 0 & a_{234} & a_{34} & a_{234} + a_{34} + a_4 \end{pmatrix}$$

(b)

Figure 2.2 An example phylogenetic tree with different topology and corresponding covariance matrix.

2.2.1 Representing Tree-Structured Covariance Matrices

Let $d = \begin{bmatrix} a_{12} & a_1 & a_2 & a_{34} & a_3 & a_4 \end{bmatrix}^T$ be a column vector containing the branch lengths of the tree in Figure 2.1. We can write $B = \sum_{k=1}^{6} d_k M^k$ where M^k is a matrix such that $M^k_{i,j} = 1$ if objects i and j co-occur in the subset corresponding to the node where branch k ends. For the branch with length a_{12}

$$M^1 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.1}$$

Furthermore, we can use indicator vectors to specify the M^k matrices in the linear expansion for B as rank-one matrices v_k. For example, letting $v_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T$, we get

$$M^1 = v_1 v_1^T = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}. \tag{2.2}$$

Thus, using vectors v_k we can write $B = \sum_{k=1}^{6} d_k v_k v_k^T$ and defining matrices $V = \begin{bmatrix} v_1 & v_2 & \dots & v_6 \end{bmatrix}$ and $D = \mathrm{diag}(d)$, we can equivalently write

$$B = VDV^T. \tag{2.3}$$

For Figure 2.1, the expansion is given by

$$V = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{and } D = \mathrm{diag}(\begin{bmatrix} 0 & a_{12} & a_{34} & a_1 & a_2 & a_3 & a_4 \end{bmatrix}^T). \tag{2.4}$$

Since the basis matrix V in Equation (2.3) is determined by the nested partitions defined by the corresponding tree topology, all matrices of the same class are generated by linear expansions of a corresponding matrix V with branch lengths specified in the diagonal matrix D. On the other hand, a distinct basis matrix V corresponds to each distinct tree topology. Matrices spanned by the set of matrices V that correspond to valid partitions correspond to tree-structured covariance matrices. We now characterize this set of valid V matrices by defining a partition property, and give a representation theorem for tree-structured covariance matrices based on this property.

Definition 2.1 (Partition Property) A basis matrix V of size p-by-$2(p-1)$ with entries in $\{0, 1\}$ and unique columns has the partition property for trees of size p if it satisfies the following conditions:

- V contains the unit vector $e = (1, 1, \ldots, 1)^T \in \mathbb{R}^p$ as a column
- For every column w in V with more than one non-zero entry, it contains columns u and v such that $u + v = w$.

A matrix V with the partition property can be constructed by starting with the column $e \in \mathbb{R}^p$ and splitting it into two nonzero columns u and v with $u + v = e$. These form the next two columns of V. The remaining columns of V are generated by splitting previously unsplit existing columns recursively into the sum of two nonzero columns, until we finally obtain columns with a single nonzero. It is easy to see that the total number of splits is $p - 1$, with two columns generated at each split. It follows that V does not contain the the zero column, and contains all p vectors that contain $p-1$ zero terms and a single entry of 1. For example, the V matrix in Equation (2.4) would be constructed by starting with column 1, splitting into columns 2 and 3, and then splitting each recursively to obtain the remaining four columns.

Theorem 2.2 (Tree Covariance Representation) A matrix B is a tree-structured covariance matrix if and only if $B = VDV^T$ where D is a diagonal matrix with positive entries, and basis matrix V satisfies the partition property.

Proof. The proof is trivial. Assume B is a tree-structured covariance matrix, then construct matrix V using the method above starting from the root, splitting each vector according to the nested partitions at each node. By construction, V will satisfy the partition property and by placing branch lengths in diagonal matrix D we will have $B = VDV^T$. On the other hand, let $B = VDV^T$ with D diagonal and V satisfying the partition property. Then construct a tree by the reverse construction: starting at the root and vector $e \in \mathbb{R}^p$, create a nested partition from the vectors u and v such that $u + v = e$ which must exist since V has the partition property. Define branch lengths from D correspondingly, and continue this construction recursively. B will then be the covariance matrix defined by the resulting tree and therefore be tree-structured.

2.2.2 Characteristics of the Set of Tree-Structured Covariance Matrices

We now state some facts about the set of tree-structured covariance matrices which we make use of in our estimation procedures.

Proposition 2.3 The set of tree-structured covariance matrices $B = VDV^T$ generated by a single basis matrix V is convex.

Proof. Let d_1 and d_2 be the branch length vectors of tree-structured covariance matrices $B_1 = V\mathrm{diag}(d_1)V^T$ and $B_2 = V\mathrm{diag}(d_2)V^T$. Let $\theta \in [0,1]$, then $B = \theta B_1 + (1-\theta)B_2 = V\mathrm{diag}(\theta d_1 + (1-\theta)d_2)V^T$. So, B is a tree of the same structure with branch lengths given by $\theta d_1 + (1-\theta)d_2$.

We will use this fact to express estimation problems for trees of fixed topology as convex optimization problems. However, estimation of general tree-structured covariance matrices is not so simple, as the set of all tree-structured covariance matrices is *not convex* in general. We can see that this is true in the case $p = 3$ by considering the following example. Defining

$$V_1 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \qquad V_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix},$$

we see that V_1 and V_2 both have the partition property. Therefore by Theorem 2.2, the matrices $B_1 = V_1\mathrm{diag}(d_1)V_1^T$ and $B_2 = V_2\mathrm{diag}(d_2)V_2^T$ are both tree-structured covariance matrices when d_1 and d_2 contain all positive entries. If B is a convex combination of B_1 and B_2, we will have $B_{12} \neq 0$ and $B_{23} \neq 0$ but $B_{13} = 0$. It is not possible to identify a matrix V with the partition property such that $B = VDV^T$, since any such V may contain only a single column apart from the three "unit" columns $(1,0,0)^T$, $(0,1,0)^T$, and $(0,0,1)^T$, and none of the possible candidates for this additional column (namely, $(1,1,0)^T$, $(1,0,1)^T$, and $(0,1,1)^T$) can produce the required nonzero pattern for B. This example can be extended trivially to successively higher dimensions p by expanding V_1 and V_2 appropriately.

2.3 Fixed Topology Projection Problems

In this section, we address the problem of estimating a tree-structured covariance matrix from a known tree topology by minimizing the distance to an observed sample covariance matrix. That is, given a sample covariance matrix S and a basis matrix V, we find the nearest tree-structured covariance matrix in norm $\|\cdot\|$. We will look at problems using Frobenius norm, $\|B\|_F = \sqrt{\sum_{ij} B_{ij}^2}$, and sum-absolute-value (sav) norm, $\|B\|_{\text{sav}} = \sum_{ij} |B_{ij}|$.

As stated above, the set of covariance matrices corresponding to trees of a particular topology is convex. Since projection problems have convex objective functions, they are convex optimization problems for any norm $\|\cdot\|$. While our emphasis in this work is optimization over the non-convex set of all tree-structured covariance matrices, it is illustrative to show the convex optimization problem formulations for projection in Frobenius and sum-absolute-value norm with fixed-topologies.

For Frobenius norm, given a covariance matrix S, the nearest tree covariance B in the class determined by basis matrix V is given by the branch length vector that solves the problem

$$\min_{d \in \mathbb{R}^{2(p-1)}} \quad \|S - V\text{diag}(d)V^T\|_F^2 \tag{2.5}$$

$$\text{s.t.} \qquad d \geq 0. \tag{2.6}$$

We can simplify this to the following equivalent quadratic problem:

$$\min_{d \in \mathbb{R}^{2(p-1)}} \quad d^T Q d - 2c^T d \tag{2.7}$$

$$\text{s.t.} \qquad d \geq 0, \tag{2.8}$$

where $Q = (V^T V) \circ (V^T V)$ and $c = \text{diag}(V^T S V)$ with $\circ$ denoting element-wise (Hadamard) matrix multiplication. For sav norm, the branch lengths d corresponding to the nearest tree-structured matrix in the proper class are given by the solution to the following problem:

$$\min_{d \in \mathbb{R}^{2(p-1)}} \quad \|S - V\text{diag}(d)V^T\|_{\text{sav}} \tag{2.9}$$

$$\text{s.t.} \qquad d \geq 0. \tag{2.10}$$

Letting $s \in \mathbb{R}^{p(p+1)/2}$ be the vectorization of symmetric matrix S, we can we can rewrite this as the following linear problem:

$$\min_{\substack{d\in\mathbb{R}^{2(p-1)} \\ p,q\in\mathbb{R}^{p(p+1)/2}}} \quad e^T(p+q) \tag{2.11}$$

$$\text{s.t.} \quad \begin{bmatrix} H & I & -I \end{bmatrix} \begin{bmatrix} d \\ p \\ q \end{bmatrix} = s \tag{2.12}$$

$$d \geq 0, p \geq 0, q \geq 0 \tag{2.13}$$

where the row of H corresponding to S_{ij} is $V_{\cdot i} \circ V_{\cdot j}$ and e is the unit vector of the appropriate length.

2.4 Solving Estimation by Projection for Unknown Tree Topologies using Mixed-Integer Programming

The non-convexity of the set of tree-structured covariance matrices requires estimation procedures that handle the combinatorial nature of optimization over this set. We choose to model these problems as mixed-integer programs (MIPs). In particular, we make use of the fact that algorithms for mixed-integer linear and quadratic programs are well-understood and robust production code exists for their solution.

2.4.1 Mixed-Integer Programming

Mixed-integer programs (MIPs) place integrality constraints on some of the problem variables. The general statement of a MIP is:

$$\min_{x\in\mathbb{R}^n} \ f_0(x) \tag{2.14a}$$

$$\text{s.t. } g_i(x) \leq 0, \ \ i = 1, 2, \ldots, m \tag{2.14b}$$

$$x_j \in \mathbb{Z}, \ \ j = 1, 2, \ldots, t, \tag{2.14c}$$

for some $t \leq n$. The functions g_i are constraint functions and f_0 is the objective function, and $\mathbb{Z}$ is the set of integers. When f_0 and $g_i, \ i = 1, \ldots, m$, are linear we have a mixed-integer linear

program (MILP), and when f_0 is quadratic and $g_i,\ i = 1, \ldots, m$, are linear, we have mixed-integer quadratic program (MIQP). We will see that projection problems for tree-structured covariance matrices are MILPs for sav norm and MIQPs for Frobenius norm.

Although the problem (2.14) is intractable in general, many practical instances can be solved, and algorithms for finding solutions have been the subject of intense research for 50 years (see for example (Wolsey and Nemhauser, 1999)). Current state-of-the-art software combines two methodologies: branch-and-bound and branch-and-cut. Branch-and-bound is based on construction of a tree[1] of relaxations of the problem (2.14), where each node of the tree contains a subproblem in which some of the integer variables x_j are allowed to take non-integer values (but may be confined to some range). A node is a child of another node in the tree if there is exactly one component x_j that is fixed at an integer value in the current node but that is a continuous variable in the parent node. In the root node of the tree, *all* integer variables are relaxed and allowed to take non-integer values, while at the leaf nodes, all integer variables x_j, $j = 1, 2, \ldots, t$ are fixed at certain values. Each node of the tree is therefore a continuous linear program (with real variables), so it can be "evaluated" using the simplex method, usually by modifying the solution of its parent node. The optimal objective at a node gives a lower bound on the optimal objectives of any of its descendants, since the descendants have fewer degrees of freedom (that is, a more restricted feasible set). Hence, if this lower bound is worse than the best integer solution found to date, this node and all its descendants can be "pruned" from the tree; it is not necessary to evaluate them as they cannot contain the solution of (2.14). The branch-and-bound algorithm traverses this tree judiciously, avoiding evaluation of large parts of the tree that are determined *not* to contain the optimal solution.

Cutting planes are used to enhance the speed of this process. These are additional constraints that exclude from the feasible set those values of x that are determined not to be optimal. Cuts can be valid for the whole tree, or just at a certain node and its descendants.

The branching strategy which determines the order in which the search tree is traversed, and the cutting planes used to derive upper bounds, can have a significant effect on the efficiency of the

[1]The tree referred to in this paragraph is a tree of related relaxations of the MIP, not a phylogenetic tree.

MIP solver for particular problems. In Appendix 2.7, we provide details regarding the parameters chosen in our MIP solver for the projection problems we address here.

2.4.2 Mixed-Integer Constraints for Tree Topology

Every tree-structured covariance matrix satisfies the following properties derived from the linear expansion in Equation (2.3):

- $B_{ij} \geq 0 \quad \forall i, j$, since all entries in V and d are nonnegative.
- $B_{ii} \geq B_{ij} \quad \forall i, j$, since V has the partition property, every component of d that is added to an off-diagonal entry is added to the corresponding diagonal entries along with the component of d corresponding to the column in V with a single non-zero entry for the corresponding leaves.
- $B_{ij} \geq \min(B_{ik}, B_{jk}) \quad \forall i \neq j \neq k$, since V has the partition property, for every three off-diagonal entry there is one entry that has at least one fewer component of d added in than the other two components.

Since every tree-structured covariance matrix can be expressed as $B = VDV^T$ according to Theorem 2.2, it is also positive semidefinite (this follows from $VDV^T = \sum_i d_i v_i v_i^T$ being the sum of positive semidefinite matrices). Also, the three properties above follow from the expansion $B = VDV^T$, therefore any matrix that satisfies these properties is also positive semidefinite, and as such, we need not add semidefiniteness constraints in the optimization problems below. Therefore, we can solve estimation problems for unknown tree topologies by constraining covariance matrices to satisfy the above properties. However, the third constraint is not convex, and we use integrality constraints to model it.

We can rewrite the third constraint for each distinct triplet $i > j > k$ as a disjunction of three constraints:

$$B_{ij} \geq B_{ik} = B_{jk} \tag{2.15a}$$

$$B_{ik} \geq B_{ij} = B_{jk} \tag{2.15b}$$

$$B_{jk} \geq B_{ij} = B_{ik}. \tag{2.15c}$$

A standard way of modeling disjunctions is to use $\{0, 1\}$ variables in the optimization problem (Bertsimas and Weismantel, 2005). In our case we can use two integer variables ρ_{ijk1}, ρ_{ijk2}, under the constraint that $\rho_{ijk1} + \rho_{ijk2} \leq 1$, that is, they can both be 0, or, strictly one of the two is allowed to take the value 1. With these binary variables we can write the constraints above in a way such that the constraint corresponding to the nonzero-valued binary variable must be satisfied. For example, constraint (2.15a) is transformed to:

$$B_{ij} \geq B_{ik} - (1 - \rho_{ijk1})M$$

$$B_{ik} \geq B_{jk} - (1 - \rho_{ijk1})M$$

$$B_{jk} \geq B_{ik} - (1 - \rho_{ijk1})M,$$

where M is a very large positive constant. Constraints (2.15b) and (2.15c) are transformed similarly, yielding the full set of mixed-integer constraints in Table 2.1. When $\rho_{ijk1} = 1$, these constraints imply that constraint 2.15a is satisfied. However, since $\rho_{ijk1} = 1$ we must have $\rho_{ijk2} = 0$ which implies that constraints 2.15b and 2.15c need not be satisfied for a solution to be feasible. When $\rho_{ijk1} = \rho_{ijk2} = 0$, then constraint 2.15c must be satisfied.

2.4.3 Projection Problems

Let S be a sample covariance matrix, the nearest tree structured covariance matrix in norm $\|\cdot\|$ to S is given by the solution of the mixed-integer problem:

$$\min_{B \in \mathcal{S}^p} \quad \|S - B\| \tag{2.17}$$

$$\text{s.t.} \quad \text{constraints 2.16a-2.16m hold for } B. \tag{2.18}$$

Table 2.1 Mixed integer constraints defining tree-structured covariance matrices

$$B_{ij} \geq 0 \ \ \forall i, j \tag{2.16a}$$

$$B_{ii} \geq B_{ij} \ \ \forall i \neq j \tag{2.16b}$$

$$B_{ij} \geq B_{ik} - (1 - \rho_{ijk1})M \tag{2.16c}$$

$$B_{ik} \geq B_{jk} - (1 - \rho_{ijk1})M \tag{2.16d}$$

$$B_{jk} \geq B_{ik} - (1 - \rho_{ijk1})M \tag{2.16e}$$

$$B_{ik} \geq B_{ij} - (1 - \rho_{ijk2})M \tag{2.16f}$$

$$B_{ij} \geq B_{jk} - (1 - \rho_{ijk2})M \tag{2.16g}$$

$$B_{jk} \geq B_{ij} - (1 - \rho_{ijk2})M \tag{2.16h}$$

$$B_{jk} \geq B_{ij} - (\rho_{ijk11} + \rho_{ijk2})M \tag{2.16i}$$

$$B_{ij} \geq B_{ik} - (\rho_{ijk11} + \rho_{ijk2})M \tag{2.16j}$$

$$B_{ik} \geq B_{ij} - (\rho_{ijk11} + \rho_{ijk2})M \tag{2.16k}$$

$$\rho_{ijk1} + \rho_{ijk2} \leq 1 \tag{2.16l}$$

$$\rho_{ijk1}, \rho_{ijk2} \in \{0, 1\} \ \ \forall \ i > j > k \tag{2.16m}$$

For Frobenius norm $\|\cdot\|_F$, the problem reduces to a mixed-integer quadratic program. Let s_2 be the vectorization of symmetric matrix S such that $\|S\|_F = \|s_2\|_2$, then the nearest tree-structured covariance matrix in Frobenius norm to matrix S is given by the corresponding matrix representation of solution $\hat{b}$ of the following mixed integer quadratic program:

$$\min_{b\in\mathbb{R}^{p(p+1)/2},\rho\in\mathbb{R}^{\bar{p}}} \quad \tfrac{1}{2}b^Tb - s_2^Tb \tag{2.19}$$

$$\text{s.t.} \quad \text{constraints 2.16a-2.16m hold for } B, \tag{2.20}$$

where $\bar{p} = \frac{p!}{(p-3)!}$.

We can similarly find the nearest tree structured covariance matrix in sum-absolute-value (sav) norm. Let s_1 be the vectorization of symmetric matrix S such that $\|S\|_{sav} = \|s_1\|_1$, then the nearest tree covariance in sum-absolute-value norm is given by the corresponding matrix representation of solution $\hat{b}$ of the following mixed integer linear program:

$$\min_{b\in\mathbb{R}^{p(p+1)/2},\rho\in\mathbb{R}^{\bar{p}}} \quad \|s_1 - b\|_1$$

$$\text{s.t.} \quad \text{constraints 2.16a-2.16m hold for } B$$

2.5 A Case Study in Gene Family Analysis of Yeast Gene Expression

We applied our methods to the analysis of gene expression in *Saccharomyces cerevisiae* gene families as presented in Oakley et al. (2005) [2]. Following the methodology of Gu et al. (2002), the yeast genome is partitioned into gene families using an amino acid sequence similarity heuristic. The largest 10 of the resulting families are used in this analysis with family sizes ranging from $p = 7$ to $p = 18$ genes. Names and sizes for the gene families used in the analysis are given in Table 2.3 of Appendix 2.8. We refer to Oakley et al. (2005) for further details.

The gene expression data is from 19 cDNA microarray time course experiments. Each time point in the series is the $\log_2$ ratio of expression at the given time point to expression at the base line under varying experimental conditions. To make our results comparable to the analysis in Oakley

[2] All data for this analysis was retrieved from "http://www.lifesci.ucsb.edu/eemb/labs/oakley/pubs/MBE2005data/"

et al. (2005), we do not model correlation between measurements at different time points. However, refer to Oakley et al. (2005) and Gu (2004) for a discussion regarding this violation of the independence assumption among measurements.

The analysis in Oakley et al. (2005) proceeded as follows:

1. Phylogenetic trees were derived for each family from DNA sequence using Maximum Likelihood methods. In particular, an alignment of amino acid sequences from the entire gene coding region was used to derive a DNA sequence alignment which was then used to estimate a phylogenetic tree. As stated by the authors (Oakley et al., 2005), this is one of many possible choices, including for example, flanking upstream non-coding regions that could have a significant role in expression regulation.

2. Based on the resulting trees, gene expression data was analyzed using Maximum Likelihood methods under a Brownian diffusion process under two families of models: a phylogenetic class, where the covariance of the diffusion process has a tree structure, and a non-phylogenetic class where the covariance of the diffusion process is diagonal. The AIC score of the resulting ML estimate is used to classify each gene family-experiment pair as evolving under a phylogenetic or non-phylogenetic model.

For each gene family and experiment we have a matrix Y_{gi} of size n_i-by-p where n_i is the number of time points in the ith experiment and p is the gene family size. We partition the experiments of each gene family into two disjoints sets $P = \{1, \ldots, l\}$ and $NP = \{l+1, \ldots, 19\}$ where l is the number of experiments classified as phylogenetic in Oakley et al. (2005). This partition yields two matrices of measurements for each gene family $Y_{gP} = \begin{bmatrix} Y_{g1}^T & \cdots & Y_{gl}^T \end{bmatrix}^T$ and similarly for Y_{gNP}, obtained by concatenating the measurement matrices of experiments in the corresponding set. The idea of concatenating gene expression measurement matrices directly to estimate covariance was sparked by the success of Stuart et al. (2003) where gene expression measurements were concatenated directly to measure correlation between genes. Since we will treat the rows of these two matrices as samples from distributions with $\mathbb{E}Y = 0$, we center each row independently to have mean 0.

One of the constraints in Section 2.4.2 that characterize tree-structured covariance matrices is the nonnegativity of their entries. Therefore, to initialize our projection solvers, we first estimate Maximum-Likelihood covariance matrices B^+_{gP} and B^+_{gNP} constrained to have nonnegative entries from sample matrices Y_{gP} and Y_{gNP}. Treating the rows of n-by-p matrix Y as independent samples from a multivariate normal distribution $N(0, B^+)$ the goal is to find matrix B^+ that maximizes likelihood, where B^+ is constrained to have nonnegative entries. Following the constrained maximum-likelihood formulation in Vandenberghe et al. (1998), we define the following convex determinant maximization problem

$$\max_{R \in \mathcal{S}^p} \quad n \log\det R - \mathrm{tr}(RS) \tag{2.21a}$$

$$\text{s.t.} \quad R_{ij} \leq 0, \ \forall i \neq j \tag{2.21b}$$

$$R \succ 0, \tag{2.21c}$$

where $\mathcal{S}^p$ is the space of p-by-p symmetric matrices, n is the number of samples in matrix Y, and $S = YY^T$ its sample covariance matrix. The expression $R \succ 0$ denotes that R is positive definite and we take variable R to be the inverse of the estimate $B^+ = R^{-1}$. By the nonpositivity element-wise constraints, along with the positive definite constraint, feasible solutions to Problem (2.21) will be members of the class of M-matrices (Horn and Johnson, 1991) which have the property that their inverse are matrices with nonnegative entries (Theorem 2.5.3 in Horn and Johnson (1991)). Therefore, the constraints in Problem (2.21) imply that estimate $\hat{B}^+$ will be the maximum likelihood estimate with nonnegative entries.

From estimates $\hat{B}^+_{gP}$ and $\hat{B}^+_{gNP}$ we estimate tree-structured covariance matrices $\hat{B}_{gP}$ and $\hat{B}_{gNP}$ using our MIP projection methods. To describe the strength of the hierarchical structure of these estimated covariances we define the *structural strength* metric as follows:

$$SS(B) = \frac{1}{p} \sum_{i=1}^{p} \frac{\max_{i \neq j} B_{ij}}{B_{ii}}. \tag{2.22}$$

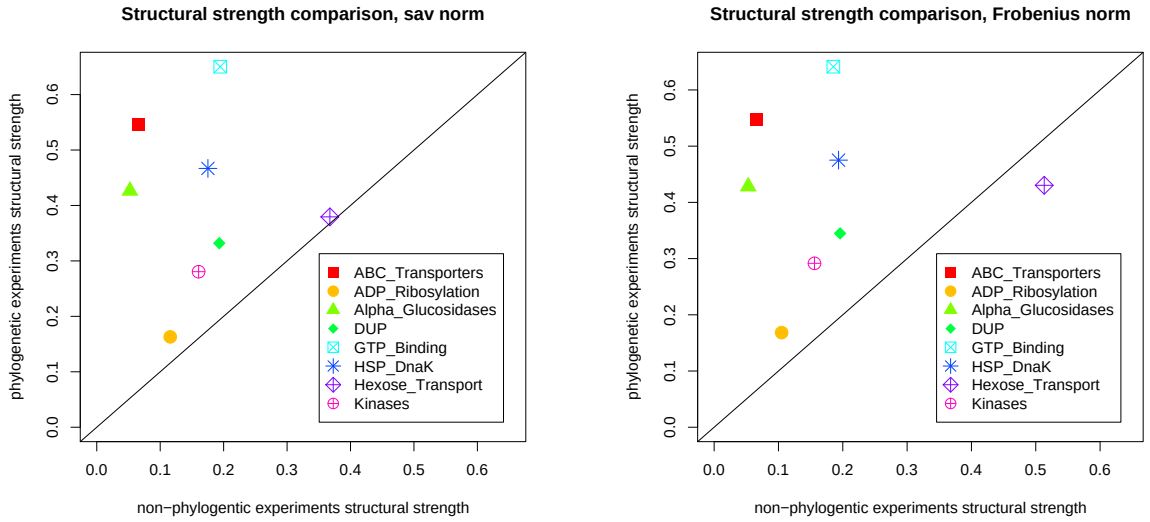
The term $\max_{i \neq j} B_{ij}$ is the largest covariance between object i and a different object j. This is the length of the path from the root to the immediate ancestor of leaf i in the corresponding tree.

Therefore, the ratio in $SS(B)$ compares the length of the path from the root to leaf i to the length of the subpath from the root to i's immediate ancestor. A value of $SS(B)$ near zero means that on average objects have zero covariance, values near one means that the tree is strongly hierarchical where objects spend very little time taking independent steps in the diffusion process.

Under the classification of experiments as undergoing phylogenetic versus non-phylogenetic evolution we expect that the structural strength metric should be quite different for estimated tree-structured covariance matrices $\hat{B}_{gP}$ and $\hat{B}_{gNP}$. That is, we expect that $SS(\hat{B}_{gP}) \geq SS(\hat{B}_{gNP})$ for most gene families g. We show our results in Figure 2.3 which validate this hypothesis. We plot $SS(\hat{B}_{gP})$ versus $SS(\hat{B}_{gNP})$ for each gene family g. The diagonal is the area where $SS(\hat{B}_{gP}) = SS(\hat{B}_{gNP})$. We see that in fact $SS(\hat{B}_{gP}) > SS(\hat{B}_{gNP})$ for all gene families g except the Hexose Transport Family.

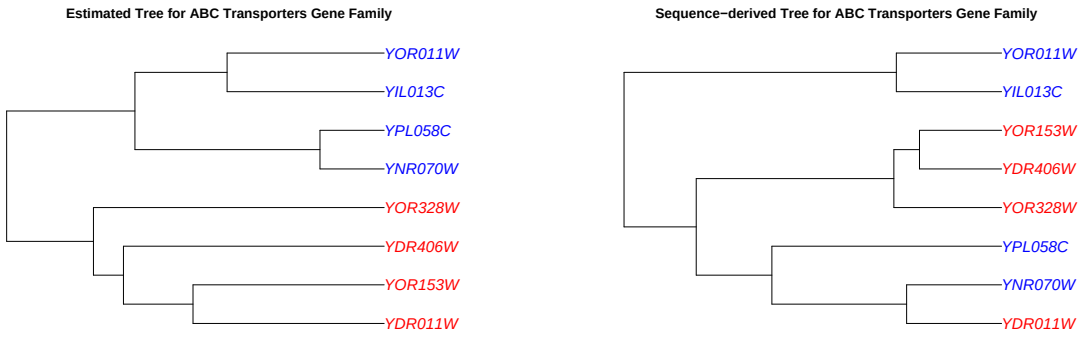
We next look at the resulting tree for the ABC (ATP-binding cassette) Transporters gene family (see Jungwirth and Kuchler (2006) for a short literature review). In particular, the eight genes included in this group are members of the subfamily conferring pleiotropic drug resistance (PDR) and are all located in the plasma membrane. A number of transcription factors have been found for the PDR subfamily, including the PDR3 factor considered one of the master regulators of the PDR network (Delaveau et al., 1994). Figure 2.4 shows the tree estimated by the MIP projection method for this family along with the sequence-derived tree reported by Oakley et al. (2005). We can notice topological differences between the two trees, in particular, the subtree in Figure 2.4(a) containing genes YOR328W, YDR406W, YOR153W and YDR011W.

In order to elucidate this topological difference, we turn to the characteristics of the promoter (regulatory) regions of the genes and asked whether transcription factor (TF) binding site contents of the upstream regions could account for this difference. We compiled a list of known yeast transcription factor binding site consensus sequences using Gasch et al. (2004) and the Promoter Database of *Saccharomyces cerevisiae* (SCPD) (`http://rulai.cshl.edu/SCPD/`). Then, we generated a transcription factor binding site occurrence vector for each gene by simply counting the number of occurrences of each consensus sequence in the 1000 base pairs upstream of the coding region. Putting these profiles together we obtained a 8-by-128 matrix where rows represent



(a) (b)

Figure 2.3 Comparison of structural strengths for tree-structured covariance estimates $\hat{B}_{gP}$ and $\hat{B}_{gNP}$ for projection under sav (a) and Frobenius (b) norms. Each point represents a gene family. The x-axis is $SS(\hat{B}_{gNP})$. We can see that for all, except the Hexose Transport gene family, $SS(\hat{B}_{gP}) > SS(\hat{B}_{gNP})$. Only eight families are shown since the Putative Helicases and Permeases families did not have any experiments classified as phylogenetic.



(a) (b)

Figure 2.4 (a) shows the tree estimated by the MIP projection method using Frobenius norm for the ABC Transporters gene family. (b) shows the sequence-derived tree reported by Oakley et al. (2005) for the ABC Transporters gene family. The red tips correspond to genes YOR328W, YDR406W, YOR153W and YDR011W which form a subtree in (a) but not in (b).

the 8 genes in the ABC Transporters gene family and columns represent 128 transcription factors. Inspection of this matrix once the rows are permuted to follow the hierarchy in the tree estimated by the MIP projection method (Figure 2.4(a)) immediately revealed that the presence or absence of the PDR3 transcription factor binding site in the flanking upstream region may account for the topological difference apparent in the two estimated trees. Table 2.2 shows the number of times the motif for the PDR3 factor was detected in the upstream region of each gene.

Table 2.2 Number of occurrences of the PDR3 transcription factor motif in the 1000 bp upstream region for each gene in the ABC Transporters family. Colors match those of Figure 2.4.

	gene	Occurrences of PDR3
1	YOR011W	0
2	YIL013C	0
3	YPL058C	0
4	YNR070W	0
5	YDR406W	3
6	YOR328W	4
7	YDR011W	6
8	YOR153W	9

It is known (Delaveau et al., 1994) that the four genes in Table 2.2 with multiple PDR3 binding sites are, as opposed to the other four genes, targets of this transcription factor which controls the multi-drug resistance phenomenon. The structure of the subtree in Figure 2.4(a) corresponding to the PDR3 target genes essentially follows the frequency of PDR3 occurrences. On the other hand, the structure of subtree for the non-PDR3 target genes follows that of the sequence-derived tree of Figure 2.4(b). Namely, pairs (YOR011W,YIL013C) and (YPL058C,YNR070W) are near each other in both the sequence-derived and the MIP-derived trees. Therefore, after taking into account the initial split characterized by the presence of the PDR3 transcription factor, the MIP estimated tree (Figure 2.4(a)) is similar to the sequence-derived tree (Figure 2.4(b)).

We reiterate the observation of Oakley et al. (2005) that the choice of sequence region to create the reference phylogenetic trees in use in their analysis plays a crucial role and results could vary accordingly. From our methods we have found evidence that using upstream sequence flanking the coding region might yield a tree that is better suited to explore the influence of evolution in gene expression for this particular gene family. We believe that finding a good estimate for tree-structured covariance matrices directly from expression measurements can help investigators guide their choices for downstream comparative analysis like that of Oakley et al. (2005).

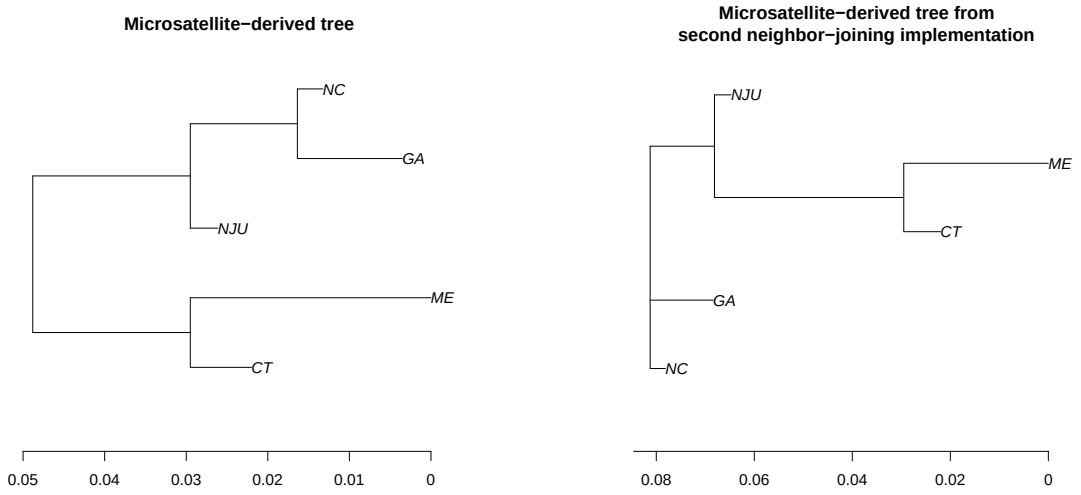
Appendices 2.7 and 2.8 detail implementation choices and running times of our implementation of the mixed-integer estimation procedure.

2.6 Discussion

The issues we hope to address by estimating tree-structured covariance matrices directly from observed sample covariances from gene expression data can be illustrated using the work of Whitehead and Crawford (2006) who characterize evolution patterns of the expression of 329 genes in five strains of the *Fundulus heteroclitus* fish. One of their analyses uses generalized least squares regression of gene expression on habitat temperature using a tree-structured covariance matrix for correction. This structured covariance matrix is derived from a phylogeny constructed from five microsatellite markers (short repeating strings) which are random characters expected to not be influenced by selection and to evolve at the same base rate as the whole genome. The tree is constructed with the greedy neighbor-joining algorithm (Saitou, 1987) from Cavalli-Sforza and Edward's (CSE) chord distances between the five microsatellite markers. We reproduce this microsatellite-derived tree in Figure 2.5(a). The neighbor-joining algorithm is a greedy algorithm susceptible to generating different solutions depending on how the algorithm is implemented. For example, the implementation of this algorithm in the ape R package [3] yields a different tree (Figure 2.5(b)) given the CSE distances. For the purpose of generalized least squares, and therefore

[3] Version 1.10-2. We thank Dr. Andrew Whitehead for providing the distance data through personal communication.

the evolutionary statements asserted as a result, this difference in topology can be significant. Considering this instability of the resulting neighbor-joining tree and the importance it plays in the authors' analyses, we posit that deriving tree-structured covariance matrices directly from the expression data can guide investigators in comparing sequence-derived phylogenetic trees for use in subsequent comparative analysis.



(a) (b)

Figure 2.5 Microsatellite-derived trees built by two implementations of the neighbor-joining algorithm from Cavalli-Sforza and Edward's chord distances. Figure 2.5(a) is the tree reported in Whitehead and Crawford (2006), and Figure 2.5(b) was obtained by the `ape` `R` package.

To address these shortcomings and motivated by what we think is a problem of genomic resolution as described in the Introduction, we have described a method for estimating tree-structured covariance matrices directly from observed sample covariance matrices by projection methods. We showed that projection problems for known topologies are linear or quadratic programs depending on the approximation norm used. For unknown topology problems, we proposed and evaluated a mixed-integer formulation which can be solved to optimality by existing branch-and-bound solvers.

The work of McCullagh (2006) on tree structured covariance matrices is the closest to our work. He proposes the *minimax projection* to estimate the tree-structure of a given sample covariance matrix. Given this structure, likelihood is maximized as in Anderson (1973). The *minimax projection* is independent of the estimation problem being solved as opposed to our MIP method which minimizes the estimation objective while finding tree structure simultaneously. Furthermore, the MIP solver guarantees optimality upon completion, at the cost of longer execution in difficult cases where the optimal trees in many tree topologies have similar objective values.

Rifkin et al. (2003) use expression directly to estimate phylogenetic structure, but use a distance-based method using the number of pairwise differentially expressed genes as the source of distances. They observe that for the resulting distance matrix the neighbor joining tree-building algorithm (Saitou, 1987) produces a tree estimate that matches the sequence derived tree for a subgroup of Drosophila.

Using the MIP formulation to model tree-structured matrix constraints, we can also address the need to solve existing tree estimation problems exactly. In particular, the least squares method of Fitch and Margoliash (1967) estimates a tree that minimizes the least-squares deviation of the distance between objects in the tree and a given distance matrix D. However, from a covariance matrix B we can compute squared distances between objects using the linear expression $D_{ij}^2 = B_{ii} + B_{jj} - 2B_{ij}$, which implies that the least squares distance-deviance objective is a quadratic function of the entries of covariance matrix B. Therefore, using the MIP formulation of Section 2.4 and the quadratic least squares distance-deviance objective we can express the least-squares method of Fitch and Margoliash (1967) as a MIQP. Therefore, generic branch-and-bound solvers of quadratic MIPs fill the gap observed in Felsenstein et al. (2004) which states that no branch-and-bound method to solve the least-squares problem exactly has been proposed.

Along the same line, MIPs have been used to solve phylogeny estimation problems for haplotype data Brown and Harrower (2006); Huang et al. (2005); Sridhar et al. (2008); Wang and Xu (2003). The observed data from the tree leaves in this case is haplotype variation represented as sequences of ones and zeros. Although our MIP formulation is related, the data in our case is

assumed to be observations from a diffusion process along a tree, suitable for continuous traits like gene expression.

We can place the problem of estimating tree-structured covariance matrices in the broader context of structured covariance matrix estimation (Anderson, 1973; Li et al., 1999; Schulz, 1997). The work of Anderson (1973) is especially relevant since an iterative procedure is used to fit matrices, or matrix inverses, which can be expressed as linear combinations of known symmetric matrices. For known topologies, this method solves likelihood maximization problems where a normality assumption is made on the diffusion process underlying the data. However, for unknown topologies, maximum likelihood problems require that we extend our computational methods to, for example, determinant maximization problems. Solving these and similar types of nonlinear MIPs is an active area of research in the optimization community (Lee, 2007). In recent years, the problem of structured covariance matrix estimation has been mainly addressed in its application to sparse Gaussian Graphical Models (Banerjee and Natsoulis, 2006; Chaudhuri et al., 2007; Drton and Richardson, 2003, 2004; Yuan and Lin, 2007). In this instance, sparsity in the inverse covariance matrix induces a set of conditional independence properties that can be encoded as a sparse graph (not necessarily a tree).

Although we presented a descriptive metric of structural strength in our estimates in Section 3.5, future work will concentrate on leveraging these methods in principled hypothesis testing frameworks that better assess the presence of hierarchical structure in observed data. We expect that the resulting methods are likely to impact how evolutionary analysis of gene expression traits is conducted.

2.7 Implementation Details

In this work we used CPLEX 9.0 (Ilog, SA, 2003) to solve the mixed-integer programs described above. This solver allows the user to specify a number of options to control the behavior of the branch-and-cut algorithm. Some of the options that we found to be very useful to solve these projection problems are the following:

1. `MIP_EMPHASIS`: The default behavior in CPLEX is to balance the traversal of the search tree to both tighten the lower bound of the optimum and find integer-feasible solutions. Since the set of tree-structured covariance matrices is non-empty, we know there exists an integer-feasible solution. Therefore, we specify that the emphasis should be solely in tightening the lower bound.

2. `VARSEL` and `NODESEL`: These parameters determine the order in which the search tree is traversed. `VARSEL` determines which variables are branched on while `NODESEL` determines the order in which nodes in the search tree are explored. We set `VARSEL` to *strong branching* so that a small number of branches are explored quickly before deciding which one to take. We set `NODESEL` to *best estimate* where an estimate of the optimum value for integer-feasible solutions under this node is used to determine order.

3. `DISJCUTS` and `FLOWCOVERS`: These parameters controls how often *disjunctive* and *flowcover* cutting planes are generated. We set both to *generate aggressively*.

4. `PROBE` *Probing* is a preprocessing step where the logical implications of setting binary variables to 1 or 0 are explored. We set this parameter to the maximum level of probing.

The determinant maximization Problem (2.21) using the SDPT3 Tütüncü et al. (2003) semidefinite programming solver. Except for this problem, all experiments and analyses were carried out in R (R Development Core Team, 2007), and many utilities of the `ape` package (Paradis et al., 2004) were used. CPLEX was used through an interface to R written by the authors available at `http://cran.r-project.org/web/packages/Rcplex/`. An R package including the MI projection solvers will be made available by the authors. Since CPLEX is proprietary software, our published code will also allow the of Rsymphony interface (`http://cran.r-project.org/web/packages/Rsymphony/index.html`) to the SYMPHONY MILP solver (`http://www.coin-or.org/SYMPHONY/`).

2.8 Running Times in Gene Family Analysis

family	p	norm	class	n	time	gap
ABC_Transporters	8	sav	phy	13	0.49	
ABC_Transporters	8	sav	nonphy	148	0.66	
ABC_Transporters	8	sav	all	161	0.26	
ABC_Transporters	8	fro	phy	13	2.01	
ABC_Transporters	8	fro	nonphy	148	0.70	
ABC_Transporters	8	fro	all	161	0.72	
ADP_Ribosylation	7	sav	phy	44	0.17	
ADP_Ribosylation	7	sav	nonphy	100	0.02	
ADP_Ribosylation	7	sav	all	144	0.07	
ADP_Ribosylation	7	fro	phy	44	0.05	
ADP_Ribosylation	7	fro	nonphy	100	0.09	
ADP_Ribosylation	7	fro	all	144	0.33	
Alpha_Glucosidases	6	sav	phy	20	0.02	
Alpha_Glucosidases	6	sav	nonphy	148	0.02	
Alpha_Glucosidases	6	sav	all	168	0.00	
Alpha_Glucosidases	6	fro	phy	20	0.11	
Alpha_Glucosidases	6	fro	nonphy	148	0.01	
Alpha_Glucosidases	6	fro	all	168	0.01	
DUP	10	sav	phy	15	112.21	
DUP	10	sav	nonphy	106	27.81	
DUP	10	sav	all	121	19.91	
DUP	10	fro	phy	15	34.86	
DUP	10	fro	nonphy	106	294.61	
DUP	10	fro	all	121	600.02	0.29%
GTP_Binding	11	sav	phy	9	22.92	
GTP_Binding	11	sav	nonphy	152	55.05	

GTP_Binding	11	sav	all	161	63.36	
GTP_Binding	11	fro	phy	9	20.93	
GTP_Binding	11	fro	nonphy	152	600.02	0.55%
GTP_Binding	11	fro	all	161	106.19	
HSP_DnaK	10	sav	phy	61	31.71	
HSP_DnaK	10	sav	nonphy	75	81.72	
HSP_DnaK	10	sav	all	136	26.49	
HSP_DnaK	10	fro	phy	61	21.60	
HSP_DnaK	10	fro	nonphy	75	412.33	
HSP_DnaK	10	fro	all	136	34.45	
Hexose_Transport	18	sav	phy	96	600.05	75.89%
Hexose_Transport	18	sav	nonphy	12	600.02	68.78%
Hexose_Transport	18	sav	all	108	600.02	76.78%
Hexose_Transport	18	fro	phy	96	600.04	2.64%
Hexose_Transport	18	fro	nonphy	12	600.08	7.39%
Hexose_Transport	18	fro	all	108	600.11	4.93%
Kinases	7	sav	phy	31	0.65	
Kinases	7	sav	nonphy	100	0.08	
Kinases	7	sav	all	131	0.09	
Kinases	7	fro	phy	31	1.04	
Kinases	7	fro	nonphy	100	0.81	
Kinases	7	fro	all	131	0.81	
Permeases	17	sav	nonphy	97	600.04	76.92%
Permeases	17	sav	all	97	600.06	76.92%
Permeases	17	fro	nonphy	97	600.01	4.49%
Permeases	17	fro	all	97	600.03	4.49%
Putative_Helicases	11	sav	nonphy	96	481.55	
Putative_Helicases	11	sav	all	96	481.50	

Putative_Helicases	11	fro	nonphy	96	600.01	0.42%
Putative_Helicases	11	fro	all	96	600.02	0.42%

Table 2.3: Run times for gene family analysis tree fitting. Each row corresponds to the MIP approximation problem for the given family and approximation norm. p is the size of the gene family, n is the number of replicates in the data matrix, and *class* indicates which class of experiments are included in the data matrix. Time reported is CPU user time in seconds. For those MIPs reaching the 10 minute time limit, we report the relative optimality gap of the returned solution.

2.9 Simulation Study: Comparing MIP Projection Methods and Neighbor-Joining

An alternative method to estimate a tree-structured covariance matrix from an observed sample covariance is to use a distance-matrix method such as the Neighbor-Joining (NJ) algorithm (Saitou, 1987) as follows: given sample covariance B, create a distance matrix D such that $D_{ij} = B_{ii} + B_{jj} - 2B_{ij}$, and use the NJ algorithm to estimate a tree and its corresponding tree-structured covariance matrix. In this simulation, we compare how close to the correct tree structure is the estimated tree-structured covariance matrix when using this NJ-based method against using our MIP-based projection methods. We measure how close the structure of estimated tree-structured matrix B_i^j is to the true structure of matrix B_i by using the tree topological distance defined by Penny and Hendy (1985) which essentially counts the number of mismatched nested partitions defined by the trees.

The simulation setting was the following: 1) we first generated 10 $\{\mathcal{T}_1, \ldots, \mathcal{T}_{10}\}$ trees with 10 leaves each at random using the `rtree` function of the `R ape` library (Paradis et al., 2004), which gives 10 associated tree-structured covariance matrices $\{B_1, \ldots, B_{10}\}$ of size 10-by-10; 2) from each tree-structured covariance matrix B_i we draw 10 sample covariances randomly $\{B_i^1, \ldots, B_i^{10}\}$

using a Wishart distribution with mean B_i and the desired degrees of freedom df, this corresponds to the sample covariance matrix of a sample with df observations from a multivariate normal random variable distributed as $N(0, B_i)$, note that the resulting sample covariances are not necessarily tree-structured; from each sample covariance matrix B_i^j we estimate a tree-structured covariance matrix $\hat{B}_i^j$ and record its topological distance to the true matrix B_i. In Figure 2.6 we report the mean topological distance of the resulting 100 estimates as a function of the degrees of freedom df, or number of observations. The values of the x-axis are defined to satisfy $df = 10 \times 2^x$, so for $x = 0$ there are 10 observations in each sample and so on.

We can see that the method based on NJ is unable to recover the correct structure even for large numbers of observations. On the other hand the MIP-based method is able to converge to the correct structure for both loss functions when the sample size is 16 times the number of taxa. Although the topological distances even for smaller sample sizes are not too large, this simulation also illustrates that, as expected, having a large number of replicates is better for this method. This observation is partly the reason for concatenating different experiments in the yeast gene-family analysis of Section 3.5.

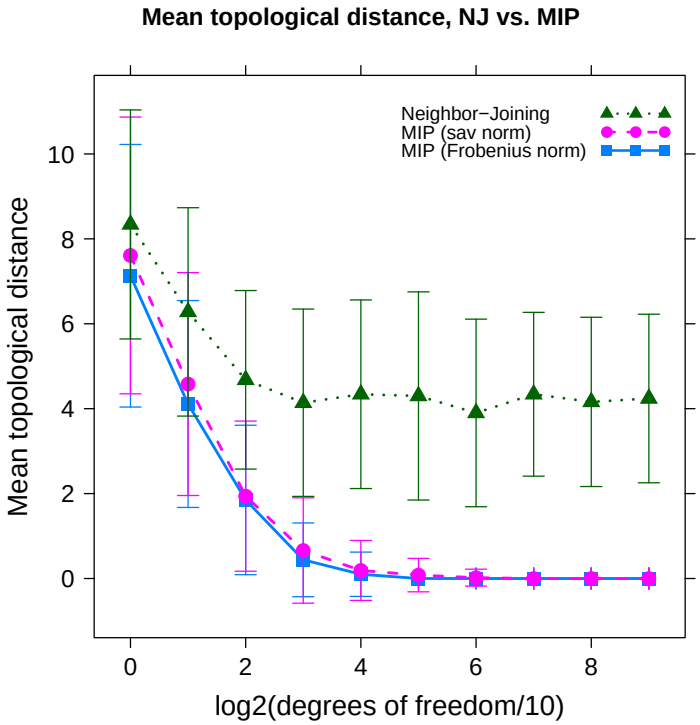


Figure 2.6 Mean topological distance between estimated and true tree-structured covariance matrices.

Part II

Graph-Based Prediction

Chapter 3

Extending Smoothing Spline ANOVA Models with Pedigree Data and its Application to Eye-Disease Prediction

3.1 Introduction

Smoothing Spline ANOVA (SS-ANOVA) models (Gu, 2002; Lin et al., 2000; Wahba et al., 1995; Xiang and Wahba, 1996) have a successful history in modeling eye disease risk. In particular, the SS-ANOVA model of pigmentary abnormalities (PA) in Lin et al. (2000) was able to show an interesting nonlinear protective effect of high total serum cholesterol for a cohort of subjects in the Beaver Dam Eye Study (BDES). We replicate those findings in Figure 3.1.[1]

More recently, genome-wide association studies have been able to link variation in a number of genomic regions to the risk of developing age-related macular degeneration (AMD), a leading cause of blindness and visual disability (Klein et al., 2004). Since pigmentary abnormalities are a precursor to the development of AMD, we want to make use of this genetic data to extend the SS-ANOVA model for pigmentary abnormality risk. For example, by extending the SS-ANOVA model of Lin et al. (2000) with a marker in the ARMS2 gene region, we were able to see that the protective effect of cholesterol disappears in subjects which have the risky variant of this allele (Figure 3.2).

Beyond genetic and environmental effects, we want to extend the SS-ANOVA for pigmentary abnormalities with familial effects. Pedigrees (see Section 3.2) have been ascertained for a large number of subjects of the BDES. We will make use of these pedigrees to include a term to the SS-ANOVA model for familial effects. The main thrust of this chapter is how to incorporate pedigree

[1] We give details regarding this model in Section 3.5.

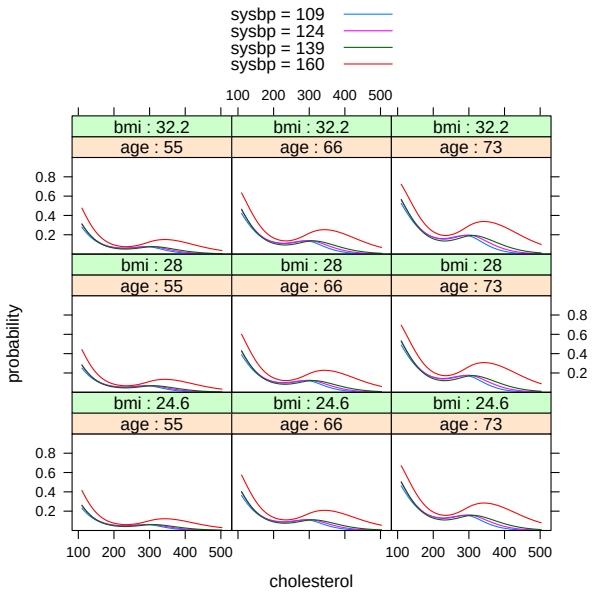


Figure 3.1 Probability from smoothing spline logistic regression model. The x-axis of each plot is cholesterol, each line is for a value of systolic blood pressure, each plot fixes body mass index and age to the shown values. $hist = 0,\ horm = 0,\ smoke = 0$ (see Table 3.1 for an explanation of model terms).

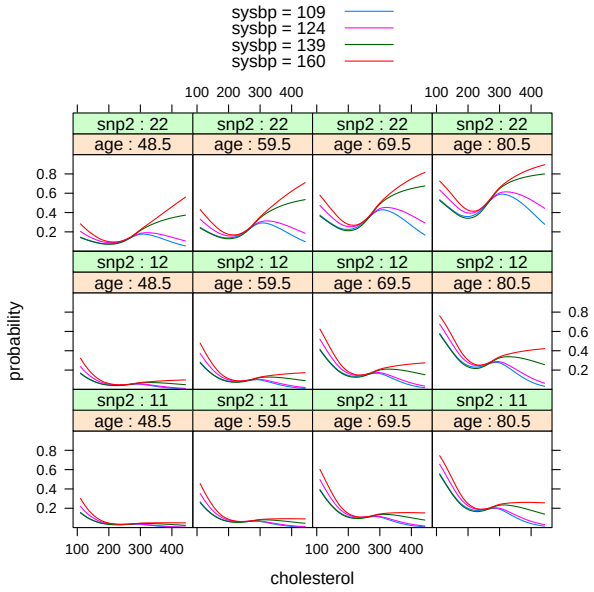


Figure 3.2 Probability for smoothing spline logistic regression model including marker from ARMS2 gene. The x-axis of each plot is cholesterol, each line is for a value of systolic blood pressure. *bmi* is fixed at the data median, with *horm*=0, *hist*=0 and *smoke*=0. Each age level is the midpoint in each range of the four age groups (see Table 3.1 for an explanation of model terms).

data into SS-ANOVA models. In fact, we present a general method that is able to incorporate arbitrary relationships that are encoded by a graph into SS-ANOVA models, from which a measure of the relative importance of graph relationships in a predictive model can be retrieved.

The goal of this chapter is to estimate models of log-odds of pigmentary abnormality risk (see Section 3.3) of the form

$$f(t_i) = \mu + g_1(t_i) + g_2(t_i) + h(z(t_i)),$$

where g_1 is a term that includes only genetic marker data, g_2 is a term containing only environmental covariate data and h is a smooth function over a space encoding relationships given by a graph, where each subject may be thought of being represented by a "pseudo-attribute" $z(t_i)$ (see Section 3.4). In the remainder of the chapter we will refer to these model terms as S (for SNP), C (for covariates) and P for pedigrees; so a model containing all three components will be referred to as S+C+P. In particular, we use models where the g_1 component is an additive linear model, and g_2 is built from cubic splines[2].

An SS-ANOVA model is defined over the tensor sum of multiple reproducing kernel Hilbert spaces (RKHS). It is estimated as the solution of a penalized likelihood problem with an additive penalty including a term for each RKHS in the ANOVA decomposition (Section 3.3), each weighted by a coefficient. These coefficients are treated as tunable hyper-parameters, which, when tuned using the GACV criterion, for example, can be interpreted as relative weights for the importance of each model component (S,C or P depending on the model). Our main tool in extending SS-ANOVA models with pedigree data is the Regularized Kernel Estimation framework of Lu et al. (2005). More complex models involving interactions between these three sources of information are possible but beyond the scope of this work.

The chapter is organized as follows: Section 3.2 defines pedigrees which encode the familial relationships we want to include in the SS-ANOVA model, which is itself discussed in Section 3.3. The methodology used to extend the SS-ANOVA model with pedigree data is given in Section 3.4.

[2]See Section 3.5 for further model details

Results on the extensions of the pigmentary abnormalities model for the BDES are given in Section 3.5, while simulation results are given in Section 3.6. We conclude with a discussion of future work in Section 3.7.

3.2 Pedigrees

A pedigree is an acyclic graph representing a set of genealogical relationships, where each node corresponds to a member of the family. The graph has an arc from each parent to an offspring, so that each node, except nodes for founders which have no incoming arcs, have two arcs, one for its father and one for its mother, in addition to arcs to its offspring. Figure 3.3 shows an example of a pedigree.

To capture genetic relationships between pedigree members, we use the well-known kinship coefficient φ of Malécot (1948) to define a pedigree dissimilarity measure. The kinship coefficient between individuals i and j in the pedigree is defined as the probability that a randomly selected pair of alleles, one from each individual, is *identical by descent*, that is, they are derived from a common ancestor. For a parent-offspring pair, $\varphi_{ij} = 1/4$ since there is a 50% chance that the allele inherited from the parent is chosen at random for the offspring, and a 50% chance that the same allele is chosen at random for the parent.

Definition 3.1 (Pedigree Dissimilarity) The pedigree dissimilarity between individuals i and j is defined as $d_{ij} = -\log_2(2\varphi_{ij})$, where φ is Malecot's kinship coefficient.

This dissimilarity is also the *degree of relationship* between pedigree members i and j (Thomas, 2004). Another dissimilarity based on the kinship coefficient can be defined as $1 - 2\varphi$. However, since we use Radial Basis Function kernels, defined by an exponential decay with respect to the pedigree dissimilarities, including the exponential decay in φ resulted in overly-diffused kernels (Section 3.4).

In studies such as the BDES, not all family members are subjects of the study, therefore, the graphs we will use to represent pedigrees in our models only include nodes for subjects rather than the entire pedigree. For example, Figure 3.4 shows the relationship graph for five BDES

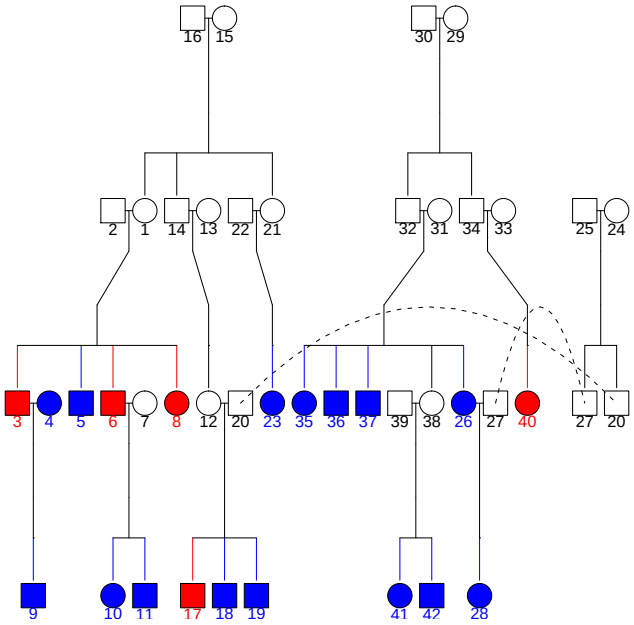


Figure 3.3 Example pedigree from the Beaver Dam Eye Study. Red nodes are subjects with reported pigmentary abnormalities, blue nodes are subjects reported as not having pigmentary abnormalities. Circles are females, rectangles are males. The cohort used in our experiments includes only blue and red circles, that is, females that have been tested for pigmentary abnormalities.

subjects from the pedigree in Figure 3.3. Edge labels are the pedigree dissimilarities derived from the kinship coefficient, and dotted lines indicate unrelated pairs.

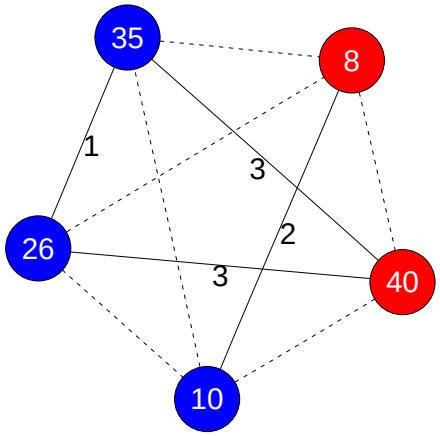


Figure 3.4 Relationship graph for five subjects in the pedigree of Figure 3.3. Colors again indicate presence of pigmentary abnormalities. Edge labels are the distances defined by the kinship coefficient. Dotted edges indicate unrelated pairs.

The main thrust of our methodology is how to incorporate into predictive models these relationship graphs derived from pedigrees and weighted by a pedigree dissimilarity that captures genetic relationship. In particular, we want to use nonparametric predictive models that incorporate other data, both genetic and environmental. In the next two Sections we will introduce the SS-ANOVA model for Bernoulli data and propose two methods extend them using relationship graphs.

3.3 Smoothing-Spline ANOVA Models

Assume we are given a data set of environmental and/or genetic covariates for each of n subjects, represented as numeric feature vectors x_i, along with responses $y_i \in \{0, 1\}$, $i \in \mathcal{N} = 1, \ldots, n$. We use the SS-ANOVA model to estimate the log-odds ratio function $f(x) = \log \frac{p(x)}{1-p(x)}$, where $p(x) = \Pr(y = 1|x)$ (Gu, 2002; Lin et al., 2000; Wahba et al., 1995; Xiang and Wahba,

1996). In particular, we will assume that f is in an RKHS of the form $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$, where $\mathcal{H}_0$ is a finite dimensional space spanned by a set of functions $\{\phi_1, \ldots, \phi_m\}$, and $\mathcal{H}_1$ is an RKHS induced by a given kernel function $k(\cdot, \cdot)$ with the property that $\langle k(x, \cdot), g\rangle_{\mathcal{H}_1} = g(x)$ for $g \in \mathcal{H}_1$, and thus, $\langle k(x_i, \cdot), k(x_j, \cdot)\rangle_{\mathcal{H}_1} = k(x_i, x_j)$. Therefore, f has a semiparametric form given by

$$f(x) = \sum_{j=1}^{m} \phi_j(x) + g(x),$$

where the functions ϕ_j have a parametric form and $g \in \mathcal{H}_1$. In the SS-ANOVA model, the RKHS $\mathcal{H}_1$ is decomposed in a particular form we discuss below.

The SS-ANOVA estimate of f given data (x_i, y_i), $i = 1, \ldots, n$, is given by the solution of the following penalized likelihood problem:

$$\min_{f \in \mathcal{H}} I_\lambda(f) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f_i) + J_\lambda(f), \tag{3.1}$$

where $l(y_i, f_i) = -y_i f(x_i) + \log(1 + e^{f(x_i)})$ is the negative log likelihood of $(y_i = 1 | f(x_i))$ and $J_\lambda(f)$ is of the form $\lambda \|P_1 f\|^2_{\mathcal{H}_1}$, with $P_1 f$ being the projection of f into RKHS $\mathcal{H}_1$. The penalty term $J_\lambda(f)$ penalizes the complexity of the function f using the norm of the RKHS $\mathcal{H}_1$ in order to avoid over-fitting f to the training data and is parametrized by the regularization parameter λ.

By the representer theorem of Kimeldorf and Wahba (1971), the minimizer of Problem (3.1) has a finite representation of the form

$$f(\cdot) = \sum_{j=1}^{m} d_j \phi_j(\cdot) + \sum_{i=1}^{n} c_i k(x_i, \cdot).$$

Thus, for a given value of the regularization parameter λ the minimizer f_λ can be estimated by solving the following convex nonlinear optimization problem

$$\min_{c \in \mathbb{R}^n, d \in \mathbb{R}^m} \sum_{i=1}^{n} -y_i f_i + \log(1 + e^{f_i}) + n\lambda c^T K c, \tag{3.2}$$

where $f = Td + Kc$, $T_{ij} = \phi_j(x_i)$ and $K_{ij} = k(x_i, x_j)$. The fact that the optimization problem is specified completely by the model matrix T and kernel matrix K is essential to the methods we will use below to incorporate pedigree data to this model.

A method for choosing the value of the regularization parameter λ that gives the estimate f_λ with best performance for unseen data in general is required. In this work, we will use the GACV method, which is an approximation to the leave-one-out approximation of the conditional Kullback-Leibler distance between the estimate f_λ and the unknown "true" log-odds ratio f (Xiang and Wahba, 1996). We note that the kernel function may be parametrized by a set of hyper-parameters that may be chosen using the GACV criterion as well. For example, the Gaussian RBF kernel

$$k(x_i, x_j) = \exp\{-\gamma \|x_i - x_j\|^2\}, \tag{3.3}$$

has γ as a hyper-parameter.

In the SS-ANOVA model, the RKHS $\mathcal{H}_1$ is assumed to be the direct sum of multiple RKHSs, so that the function $g \in \mathcal{H}_1$ is defined as

$$g(x) = \sum_\alpha g_\alpha(x_\alpha) + \sum_{\alpha<\beta} g_{\alpha\beta}(x_\alpha, x_\beta) + \cdots$$

where $\{g_\alpha\}$ and $\{g_{\alpha\beta}\}$ satisfy side conditions that generalize the standard ANOVA side conditions. Functions g_α encode "main effects", $g_{\alpha\beta}$ encode "second order interactions" and so on. An RKHS $\mathcal{H}_\alpha$ is associated with each component in this sum, along with its corresponding kernel function k_α. We can write the penalty term in (3.1) as

$$J_{\lambda,\theta}(f) = \lambda \left[\sum_\alpha \theta_\alpha^{-1} \|P_\alpha f\|^2_{\mathcal{H}_\alpha} + \sum_{\alpha\beta} \theta_{\alpha\beta}^{-1} \|P_{\alpha\beta} f\|^2_{\mathcal{H}_{\alpha\beta}} + \cdots \right], \tag{3.4}$$

where the coefficients θ are tunable hyper-parameters that allow weighting the effect of each component's penalty in the total penalty term. For the penalty of Equation (3.4), the kernel function $k(\cdot,\cdot)$ associated with $\mathcal{H}_1$ can then be itself decomposed as $k(\cdot,\cdot) = \sum_\alpha \theta_\alpha k_\alpha(\cdot,\cdot) + \sum_{\alpha\beta} \theta_{\alpha\beta} k_{\alpha\beta}(\cdot,\cdot) + \cdots$. The hyper-parameters to be chosen, by GACV for example, now include λ and the coefficients θ of the ANOVA decomposition. These coefficients θ can be interpreted as relative importance weights for each model component. Thus, in models that have genetic, environmental and familial components, the ANOVA decomposition can be used to measure the relative importance of each data component.

For genetic and environmental components, standard kernel functions can be used to define the corresponding RKHS. However, pedigree data is not represented as feature vectors for which standard kernel functions can be used. On the other hand, in order to specify the penalized likelihood problem, only the kernel matrix is required. Therefore, we will build kernel matrices that encode familial relationships, and use those in the estimation problem. In the next Section, we will show two methods for defining pedigree kernels.

3.4 Representing Pedigree Data as Kernels

The requirement for a valid kernel matrix to be used in the penalized likelihood estimation problem of Equation (3.2) is that the matrix be positive semidefinite: for any vector $\alpha \in \mathbb{R}^n$. This is denoted as $K \succeq 0$. We saw in the previous Section, that there is a close relationship between the inner product of the RKHS $\mathcal{H}_1$ and its associated kernel function k. In fact, the kernel matrix K is the matrix of inner products of the evaluation representers in $\mathcal{H}_1$ of the given data points.

A property of positive semidefinite matrices, is that they may be interpreted as the matrix of inner products of objects in a space equipped with an inner product. Therefore, since $K \succeq 0$ contains the inner products of objects in some space, we can define a distance metric over these objects as $d_{ij}^2 = K_{ii} + K_{jj} - 2K_{ij}$. We make use of this connection between distances and inner products in the Regularized Kernel Estimation framework to define a kernel based on the pedigree dissimilarity of Definition 3.1.

3.4.1 Regularized Kernel Estimation

The Regularized Kernel Estimation (RKE) framework was introduced by Lu et al. (2005) as a robust method for estimating dissimilarity measures between objects from noisy, incomplete, inconsistent and repetitious dissimilarity data. The RKE framework is useful in settings where object classification or clustering is desired but objects do not easily admit description by fixed length feature vectors. Instead, there is access to a source of noisy and incomplete dissimilarity information between objects.

RKE estimates a symmetric positive semidefinite kernel matrix K which induces a real squared distance admitting of an inner product. K is the solution to an optimization problem with semidefinite constraints that trades-off fit to the observed dissimilarity data and a penalty of the form $\lambda_{rke}\mathrm{trace}(K)$ on the complexity of K, where λ_{rke} is a non-negative regularization parameter.

The solution to the RKE problem is a symmetric positive semidefinite matrix K, which has a spectral decomposition $K = \Gamma\Lambda\Gamma^T$, with Λ a diagonal matrix with Λ_{ii} equal to the ith leading eigenvalue of K and Γ an orthogonal matrix with eigenvectors as columns in the corresponding order. An embedding $X \in \mathbb{R}^{N\times r}$ in r-dimensional Euclidean space can be derived from this decomposition by setting $X = \Gamma(:, 1:r)\Lambda(1:r)^{1/2}$, where only the r leading eigenvalues and eigenvectors are used. A method for choosing r is required, which we discuss in Section 3.5.

RKE problem Given a training set of N objects, assume dissimilarity information is given for a subset Ω of the $\binom{N}{2}$ possible pairs of objects. Denote the dissimilarity between objects i and j as $d_{ij} \in \Omega$. We make the requirement that Ω satisfies a connectivity constraint: the undirected graph consisting of objects as nodes and edges between them, such that an edge between nodes i and j is included if $d_{ij} \in \Omega$, is connected. Additionally, optional weights w_{ij} may be associated with each $d_{ij} \in \Omega$.

RKE estimates an N-by-N symmetric positive semidefinite kernel matrix K of size N, such that, the fitted distance between objects induced by K, $\hat{d}_{ij} = K(i,i)+K(j,j)-2K(i,j)$, is as close as possible to the observed distance $d_{ij} \in \Omega$. Formally, RKE solves the following optimization problem with semidefinite constraints:

$$\min_{K\succeq 0} \sum_{d_{ij}\in\Omega} w_{ij}|d_{ij} - \hat{d}_{ij}| + \lambda_{rke}\mathrm{trace}(K). \tag{3.5}$$

The parameter $\lambda_{rke} \geq 0$ is a regularization parameter that trades-off fit of the dissimilarity data, as given by absolute deviation, and a penalty, $\mathrm{trace}(K)$, on the complexity of K. The trace may be seen as a proxy for the rank of K, therefore, RKE is regularized by penalizing high dimensionality of the space spanned by K. Note that the trace was used as a penalty function by Lanckriet et al. (2004a).

As in the SS-ANOVA model, a method for choosing the regularization parameter λ_{rke} is required. However, since our final goal is to build a predictive model that performs well in general, choosing this parameter in terms of prediction performance makes sense. That is, we treat λ_{rke} as a hyper-parameter to the kernel matrix of the SS-ANOVA problem.

Figure 3.5 shows a three-dimensional embedding derived by RKE of the relationship graph in Figure 3.4. Notice that the x-axis is order of magnitudes larger than the other two axes and that the unrelated edges in the relationship graph occur along this dimension. That is, the first dimension of this RKE embedding separates the two clusters of relatives in the relationship graph. The remaining dimensions encode the relationship distance.

Not all relationship graphs can be embedded in three-dimensional space, and thus analyzed by inspection as in Figure 3.5. For example, Figure 3.8 shows the embedding of a larger relationship graph that requires more than three-dimensions to embed the pedigree members uniquely. For example, subjects coded 27 and 17 are superposed in this three dimensional embedding, with the fourth dimension separating them.

We may consider the embedding resulting from RKE as providing a set of "pseudo"-attributes $z(i)$ for each subject in this pedigree space. Thus, a smooth predictive function may be estimated in this space. In principle, we should impose a rotational invariance when defining this smooth function since only distance information was used to create the embedding. For this purpose we use radial basis function kernels, like the Gaussian kernel of Equation 3.3 and the Matérn kernels of Section 3.4.3, to define this smooth pedigree predictive function.

The fact that RKE operates on inconsistent dissimilarity data, rather than distances, is significant in this context. The pedigree dissimilarity of Definition 3.1 is not a distance since it does not satisfy the triangle inequality for general pedigrees. In Figures 3.6 and 3.7 we show an example where this is the case, where the dissimilarities between subjects labeled 17, 7 and 5 do not satisfy the triangle inequality. An embedding given by RKE for this graph is shown in Figure 3.8.

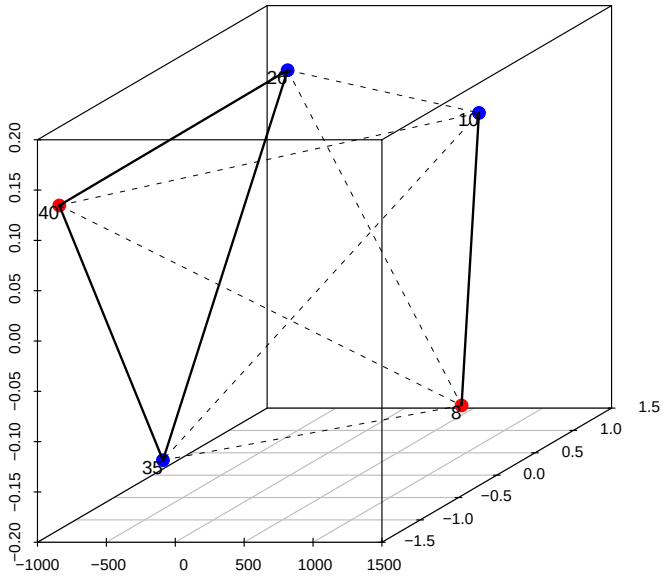


Figure 3.5 Embedding of pedigree by RKE. The x-axis of this plot is order of magnitudes larger than the other two axes. The unrelated edges in the relationship graph occur along this dimension, while the other two dimensions encode the relationship distance.

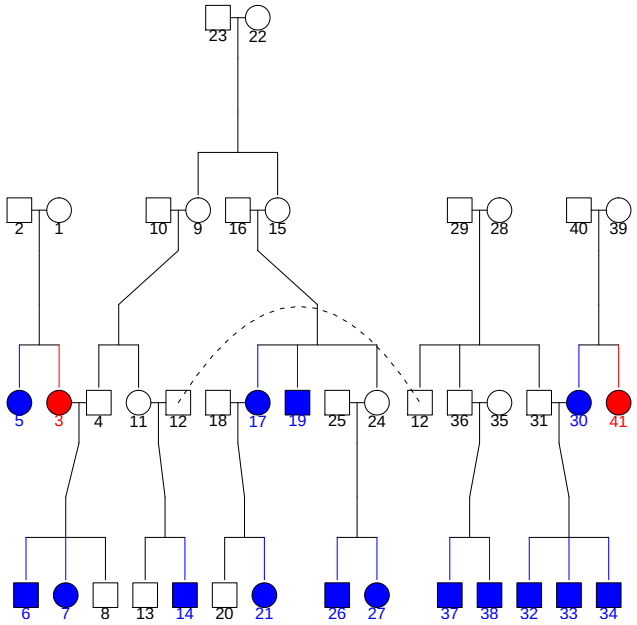


Figure 3.6 A different example pedigree. We use this pedigree to show in Figure 3.7 that the pedigree dissimilarity of Definition 3.1 is not a distance.

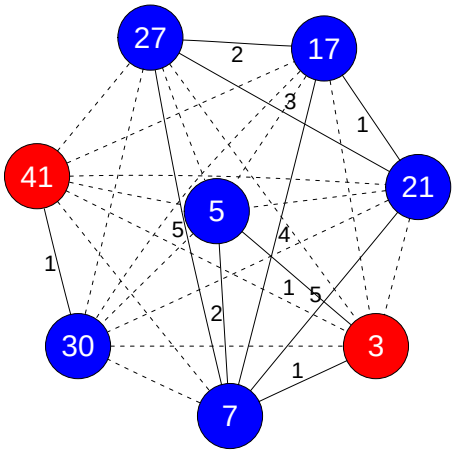


Figure 3.7 A different relationship graph. The dissimilarities between nodes labeled 17, 7 and 5 show that the pedigree dissimilarity of Definition 3.1 is not a distance.

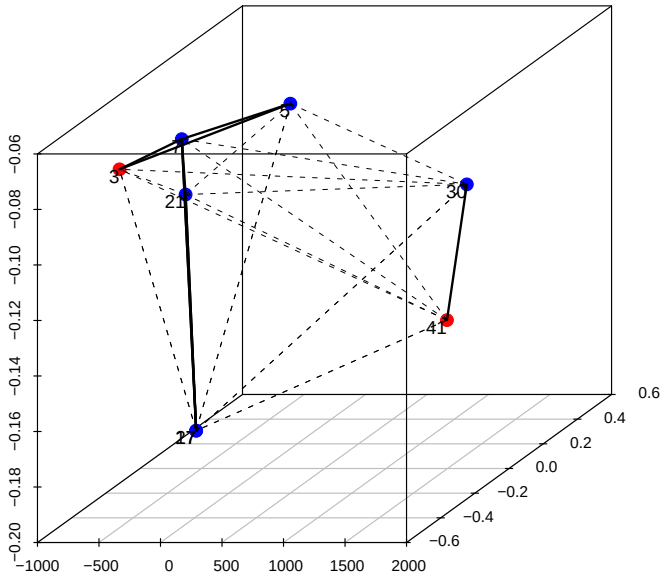


Figure 3.8 RKE Embedding for second example graph. Subjects 27 and 17 are superimposed in this three dimensional plot, but are separated by the fourth dimension.

3.4.2 Graph Kernels

Since we are encoding pedigree data as a weighted graph, we can use existing methods for defining kernels over graphs. For example, using a setting similar to Smola and Kondor (2003), we can define a pedigree Gaussian kernel as

$$K_{ij} = \exp\{-\gamma d_{ij}^2\}, \tag{3.6}$$

where d_{ij} is the pedigree dissimilarity of Definition 3.1, and γ is a kernel hyper-parameter to be chosen. However, since this pedigree dissimilarity is not a distance, the kernel resulting from applying Equation (3.6) is not positive semidefinite. In our implementation, we compute the projection under Frobenius norm of the result of Equation 3.6 to the cone of positive semidefinite matrices. This is easily computed by setting the negative eigenvalues of the matrix to zero.

3.4.3 Matérn Kernel Family

We have so far only discussed the use of the Gaussian kernel (Equation (3.3)) as basis functions for our nonparametric models. This kernel is a good candidate for this task since it depends only on the distance between objects and is rotationally invariant. However, its exponential decay poses a problem in this setting since the relationship graphs derived from pedigrees are very sparse, and the dissimilarity measure of Definition 3.1 makes the kernel very diffuse, in that most non-zero entries are relatively small.

The Matérn family of radial basis functions (Matern, 1986; Stein, 1999) also have the same two appealing features of the Gaussian kernel–dependence only on distance and rotational invariance–while providing a parametrized way of controlling exponential decay. The ν-th order Matérn function is given by

$$k_\nu(i,j) = \exp\{-\alpha d_{ij}\}\pi_\nu(\alpha, d_{ij}), \tag{3.7}$$

where α is a tunable scale hyper-parameter and π_ν is a polynomial of a certain form. In the results of Sections 3.5 and 3.6, we use the third order Matérn function:

$$k_3(i,j) = \frac{1}{\alpha^7}\exp\{-\alpha\tau\}[15 + 15\alpha\tau + 6\alpha^2\tau^2 + \alpha^3\tau^3], \tag{3.8}$$

where $\tau = d_{ij}$. The general recursion relation for the $m+1$-th Matérn function is

$$k_{m+1}(i,j) = \frac{1}{\alpha^{2m+1}}\exp\{-\alpha\tau\}\sum_{i=0}^{m+1} a_{m+1,i}\alpha^i\tau^i, \tag{3.9}$$

where $a_{m+1,0} = (2m+1)a_{m,0}$, $a_{m+1,i} = (2m+1)a_{m,i}+a_{m,i-1}$, for $i = 1,\ldots,m$ and $a_{m+1,m+1} = 1$. The Matérn family is defined for general positive orders but closed form expressions are available only for integral orders.

3.5 Case Study: Beaver Dam Eye Study

The Beaver Dam Eye Study (BDES) is an ongoing population-based study of age-related ocular disorders. Subjects were a group of 4926 people aged 43-86 years at the start of the study who lived in Beaver Dam, WI and were examined at baseline, between 1988 and 1990. A description of the population and details of the study at baseline may be found in Klein et al. (1991). Although we will only use data from this baseline study for our experiments, five, ten, and fifteen year follow-up data has been obtained (Klein et al., 1997, 2002, 2007). Familial relationships of participants were ascertained and pedigrees were constructed (Lee et al., 2004). Genetic marker data for specific SNPs was subsequently generated for those participants included in the pedigree data.

Our goal is to use this new genetic and pedigree data to extend previous work studying the association between pigmentary abnormalities and a number of environmental covariates in the context of SS-ANOVA models (Lin et al., 2000). The presence of pigmentary abnormalities is an early stage of age-related macular degeneration (AMD), which, in it's late stages, is a leading cause of blindness and visual disability (Klein et al., 2004). We use genetic marker data for the Y402H region of the complement factor H (CFH) gene and for SNP rs10490924 in the LOC387715 (ARMS2) gene. Variations in these locations have been shown to significantly alter the risk of AMD (Baird et al., 2006; Edwards et al., 2005; Fisher et al., 2005; Fritsche et al., 2008; Hageman

et al., 2005; Haines et al., 2005; Kanda et al., 2007; Klein et al., 2005; Li et al., 2006; Magnusson et al., 2006; Thompson et al., 2007a,b).

Extending the methodology of Lin et al. (2000), we estimate a SS-ANOVA models of the form

$$f(t) = \mu + d_{\text{SNP1,1}} \cdot I(X_1 = \texttt{12}) + d_{\text{SNP1,2}} \cdot I(X_1 = \texttt{22}) + d_{\text{SNP2,1}} \cdot I(X_2 = \texttt{12}) + d_{\text{SNP2,2}} \cdot I(X_2 = \texttt{22}) + f_1(\text{sysbp}) + f_2(\text{chol}) + f_{12}(\text{sysbp}, \text{chol}) + d_{\text{age}} \cdot \text{age} + d_{\text{bmi}} \cdot \text{bmi} + d_{\text{horm}} \cdot I_1(\text{horm}) + d_{\text{hist}} \cdot I_2(\text{hist}) + d_{\text{smoke}} \cdot I_3(\text{smoke}) + h(z(t)). \quad (3.10)$$

The terms in the first line of Equation (3.10) encode the effect of the two genetic markers (SNPs). A variable for each SNP is coded according to which of three variants (`11,12,22`) the subject carries for that SNP. For identifiability, the `11` level is modeled by the intercept μ for both SNPs, while an indicator variable is added for the other two levels. This results in each level (other than the `11` level) having its own model coefficient.

The next few terms encode the effect of the environmental covariates listed in Table 3.1. Functions f_1, f_2 and f_{12} constructed from cubic splines (see Gu, 2002, for the tensor product construction of f_{12}), and the remaining linear terms have I_j as indicator functions. Both systolic blood pressure and cholesterol were scaled to lie in the interval $[0, 1]$. A model of PA of this form for these environmental covariates was shown to report a protective effect of hormone replacement therapy and a suggestion of a nonlinear protective effect of cholesterol (Lin et al., 2000, and Figure 3.1). The term $h(z(t))$ encodes familial effects and is defined by the kernels presented in Section 3.4.

Models tested include combinations of the following components: 1) P (for pedigree) which defines a function only on an RKHS encoding the pedigree data (term $h(z(t))$ in Equation (3.10)), 2) S (for SNP) which includes data for the two genetic markers (terms 2 through 5 in Equation (3.10)), and 3) C (for covariates) which includes the remaining terms in Equation (3.10) encoding environmental covariates. For example, P-only refers to a model containing only a pedigree component; S+C, to a model containing components for genetic markers and environmental covariates; and P+S+C to a model containing components for all three data sources.

code	units	description
`horm`	yes/no	current usage of hormone replacement therapy
`hist`	yes/no	history of heavy drinking
`bmi`	kg/m^2	body Mass Index
`age`	years	age at baseline
`sysbp`	$mmmHg$	systolic blood pressure
`chol`	mg/dL	serum cholesterol
`smoke`	yes/no	history of smoking

Table 3.1 Environmental covariates for BDES pigmentary abnormalities SS-ANOVA model

We also compare the two methods presented for incorporating pedigree data. We refer to the method using a kernel defined over an embedding resulting from RKE (Section 8.1) as RKE/GAUSSIAN or RKE/MATERN according to the kernel function used over the embedding, and to the kernel defined over the graph dissimilarities directly (Section 3.4.2), as GAUSSIAN or MATERN accordingly. Therefore, the abbreviation P+S+C (MATERN) refers to a model containing all three data sources, where pedigree data is incorporated using the graph kernel method with Matern third order kernel.

The penalized likelihood Problem (3.2) is solved by the quasi-Newton method implemented in the `gss R` package (Gu, 2007). The RKE semidefinite Problem (8.2) is solved using the CSDP library (Borchers, 1999) with input dissimilarities given by Definition 3.1. A number of additional edges between unrelated individuals encoding the "infinite" dissimilarity are added randomly to the graph. The dissimilarity encoded by these edges is arbitrarily chosen to be the sum of all dissimilarities in the entire cohort. The number of additional edges is chosen such that each subject has an edge to at least twenty-five other subjects in the cohort (including all relatives). The kernel matrix obtained from RKE is then truncated to those leading eigenvalues that account for 95% of the matrix trace to create a "pseudo"-attribute embedding. An RBF kernel is then defined over this embedding. Pedigree dissimilarities were derived from kinship coefficients calculated using the `kinship R` package (Atkinson and Therneau, 2007).

The cohort used are females subjects of the BDES for which we have full genetic marker, covariate and pedigree data, and are from pedigrees containing two or more observations within the cohort ($n = 684$). This results in 175 pedigrees in the data set, with sizes ranging from 2 to 103 subjects. More than a third of the subjects are in pedigrees with 8 or more observations.

We will use area under the ROC curve (Fawcett, 2004, referred to as AUC), to compare predictive performance of model/method combinations, and will be estimated using ten-fold cross-validation. The cross-validation folds were created such that for every test subject in the fold, at least one other member of their pedigree is included in the training set. In each fold, pedigree kernels were built on all members of the pedigree in the cohort, however, hyper-parameters were chosen for each fold independently, using GACV on the labeled data. That is, in this scenario there

is no off-sample testing points in the sense that we have full pedigree information for all testing points.

Table 3.2 shows the resulting mean and standard deviations of the cross-validation AUC of each model/method combination. Figure 3.9 summarizes the same result by plotting the AUC of the best method for each model type. We can make the following observations based on Figure 3.9[3]:

1. the model with the highest overall mean AUC is the S+C+P model (RKE/MATERN), but models S+C (NO/PED) and S+P (MATERN) are not statistically different (p-values: 0.753 and 0.73 respectively);

2. for pedigree-less models, the S+C model containing both markers and covariates has better AUC than either the S-only or C-only models (p-values: 0.00250 and 0.065 respectively);

3. adding pedigree data to the C-only model did not increase AUC significantly (p-value 0.854);

4. adding pedigree data to the S-only model increased AUC significantly (p-value 0.0121);

5. the P-only (MATERN) and S-only models have AUC that is not statistically different (p-value 0.464)

The second result states that for pedigree-less models, combining genetic markers and environmental covariates yields a better model than either data source by itself. This is consistent with the fact that pigmentary abnormality risk is associated to both the genetic markers and environmental covariates included in the model.

Part of the first result states that model S+P performs as well as the best scoring methods is striking. For example, it states that substituting the environmental covariates in the S+C model with the pedigree data (S+P) yields the same predictive ability. This is surprising considering that pedigree data strictly encodes genetic relationships. Further investigation of this result is an avenue for future research.

For this cohort, adding pedigree data to models containing the environmental covariates did not increase predictive ability (results 1 and 3).

[3]Reported p-values are for pairwise t-tests. Pedigree results refer to the best scoring method for each model type.

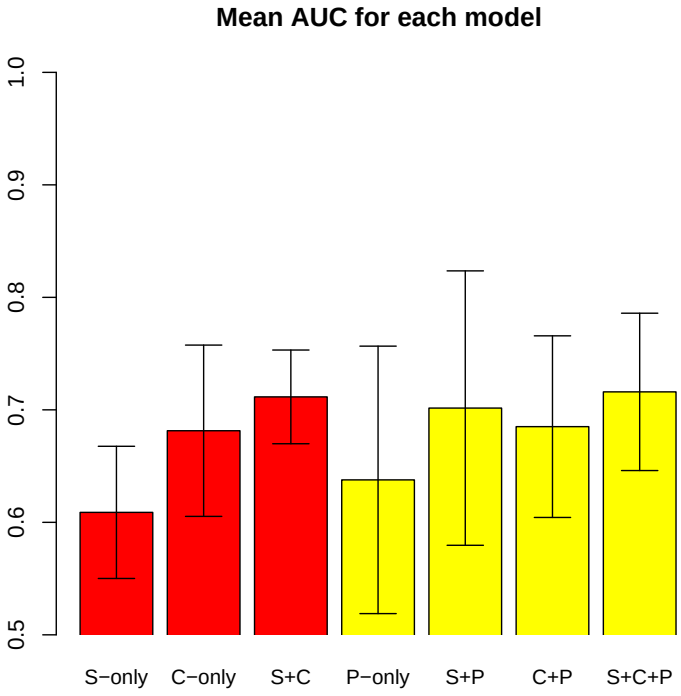


Figure 3.9 AUC comparison of models. S-only is a model with only genetic markers, C-only is a model with only environmental covariates and S+C is a model containing both data sources. P-only is a model with only pedigree data, P+S is a model with both pedigree data and genetic marker data, P+C is a model with both pedigree data and environmental covariates, P+S+C is a model with all three data sources. Error bars are one standard deviation from the mean. Yellow bars indicate models containing pedigree data. For models containing pedigrees, the best AUC score for each model is plotted. All AUC scores are given in Table 3.2.

The last two results are also interesting in that pedigree-only models, that is, models that include only familial effects, have the same predictive ability than the genetic marker-only model, while adding pedigree data to the genetic marker model increases predictive ability.

3.6 Simulation Study

In the previous Section we saw that no predictive ability is gained from adding pedigree data to the pedigree-less pigmentary abnormality SS-ANOVA model with both genetic markers and environmental covariates (P+S+C vs. S+C). We carried out a simulation study to test that our methods are not biased against including the pedigree term to the SS-ANOVA model.

We simulated an extremely simplified disease model where risk is determined by two genetic markers and a single covariate. Letting X_{1i} and X_{2i} be indicator function for the risk alleles of the two markers respectively, the log-odds ratio of the true model is given by

$$f_i = \mu + 3 * X_{1i} + 20 * X_{2i} + 24 * X_i(1 - X_i),$$

where X_i is a simulated environmental covariate drawn uniformly at random from [0,1] and independently from the markers. The constant μ is set so that the numbers of subjects with and without the disease are expected to be balanced.

We used the same cohort and pedigree structure from Section 3.5. The two genetic markers were simulated using the `ibdreg` (Sinnwell and Schaid, 2007) `R` package as follows: for each pedigree with observations in the cohort, the alleles for the founders (pedigree members without parents in the pedigree) are drawn randomly so that the risk allele is drawn with a probability of 30%; once the founder alleles are generated, inheritance by descent is simulated in the pedigree under an autosomal inheritance mode (Sinnwell and Schaid, 2007; Thomas, 2004); this generates the alleles for every member of the pedigree. The two markers were generated independently.

The purpose of this simulation is to show that if only one of the two markers are included in a model including SNPs and the covariate, adding the pedigree term to the model serves as a proxy

for the left-out SNP. We test two models: P+S+C, of the form

$$f_i = \mu + d_1 X_{1i} + g(X_i) + h_i,$$

where g is a nonparametric term for the covariate X constructed with a cubic spline and h_i is a pedigree term; and S+C, of the form

$$f_i = \mu + d_1 X_{1i} + g(X_i).$$

Under these simulation conditions, we expect that the predictive ability of the P+S+C model to be higher than that of the S+C model.

Table 3.3 shows the result for this simulation. Area under the ROC curve for the S+P+C (MATERN) method is significantly better than the S+C model (p-value 0.0314).

We note that this result hinges on the large relative weight given to the second genetic marker in the true model. For lower weights, the AUC of S+C+P is similar to that of S+C. Notice also that in this simple simulation setting the Gaussian kernel performed better than the Matérn kernel.

3.7 Discussion

Throughout our experiments and simulations we have used genetic marker data in a very simple manner by including single markers for each gene in an additive model. A more realistic model should include multiple markers per gene and would include interaction terms between these markers. While we have data on two additional markers for each of the two genes included in our case study (CFH and ARMS2) for a total of six markers (three per gene), we chose to use the additive model on only two markers since, for this cohort, this model showed the same predictive ability as models including all six markers with interaction terms (analysis not shown). Furthermore, due to some missing entries in the genetic marker data, including multiple markers reduced the sample size.

Along the same lines, we currently use a very simple inheritance model to define pedigree dissimilarity. Including, for example, dissimilarities between unrelated subjects should prove advantageous. A simple example would be including a spousal relationship when defining dissimilarity

since this would be capturing some shared environmental factors. Extensions to this methodology that include more complex marker models and multiple or more complex dissimilarity measures are fertile grounds for future work.

Methods for including graph-based data in predictive models have been proposed recently. They range from semi-supervised methods that regularize a predictive model by applying smoothness penalties over the graph (Goldberg et al., 2007; Sindhwani et al., 2005; Zhu, 2005), to discriminative graphical models (Chu et al., 2007; Getoor, 2005; Lafferty et al., 2004; Taskar et al., 2004), and methods closer to ours which define kernels from graph relationships (Smola and Kondor, 2003; Zhu et al., 2006).

There are issues in the disease risk modelling setting with general pedigrees, where relationship graphs encode relationships between a subset of a study cohort, that are usually not explicitly addressed in the general graph-based setting. Most important is the assumption that, while graph structure has some influence in the disease risk model, it is not necessarily an overwhelming influence. Thus, a model that produces relative weights between components of the model, one being graph relationships, is required. That is the motivation for using the SS-ANOVA framework in this work. While graph regularization methods have a parameter that controls the influence of the graph structure in the predictive model, it is not directly comparable to the influence of other model components, e.g. genetic data or environmental covariates. On the other hand, graphical model techniques define a probabilistic model over the graph to define the predictive model. This gives the graph relationships too much influence over the predictive model.

The relationship graphs in this setting lead to kernels that are highly diffuse in the sense that, due to the nature of the pedigree dissimilarity, there is rapid decay as the Gaussian basis function extends away from each subject. The use of the third order Matérn kernel function significantly improved the predictive ability of our methods in Section 3.5 over the Gaussian kernel, since the Matérn kernel can soften the diffusion effect. Tuning the order of the Matérn kernel could further improve our models. Note, however, that in the simple simulation setting of Section 3.6, the faster decay of the Gaussian kernel performed better than the slower decay of the Matérn kernel. Further

understanding of the type of situations in which the Matérn kernel would perform better than the Gaussian is another direction for further research.

		S-only	C-only	S+C
NO/PED		0.6089 (0.05876)	0.6814 (0.07614)	0.7115 (0.04165)
	P-only	S+P	C+P	S+C+P
GAUSSIAN	0.6226 (0.11346)	0.6909 (0.12284)	0.6533 (0.07967)	0.6991 (0.05725)
MATERN	**0.6377** (0.11889)	**0.7016** (0.12197)	0.6503* (0.10707)	0.6188* (0.12930)
RKE/GAUSSIAN	0.5684 (0.09858)	0.6360 (0.06716)	0.6262 (0.07475)	0.6469 (0.07076)
RKE/MATERN	0.6149 (0.09881)	0.6563 (0.08333)	**0.6851** (0.08073)	**0.7160** (0.06993)

Table 3.2 Ten-fold cross-validation mean for area under ROC curve. Columns correspond to models indexed by components: P (pedigrees), S (genetic markers), C (environmental covariates). Rows correspond to method tested (NO/PED is regular SS-ANOVA models without pedigree data). Numbers in parentheses are standard deviations. Numerical instabilities in the quasi-Newton solver caused many tuning runs for entries marked with (*) to fail. As a result model selection was not properly done for these entries.

	mean AUC	std. dev.
NO-PED	0.65	0.08
GAUSSIAN	0.74	0.09
MATERN	0.72	0.07
RKE/GAUSSIAN	0.69	0.09
RKE/MATERN	0.67	0.10

Table 3.3 Mean AUC for simulation setting.

Chapter 4

Protein Classification by Regularized Kernel Estimation

The Regularized Kernel Estimation (RKE) framework was introduced by Lu et al. (2005) as a robust method for estimating dissimilarity measures between objects from noisy, incomplete, inconsistent and repetitious dissimilarity data. The RKE framework is useful in settings where object classification or clustering is desired but objects do not easily admit description by fixed length feature vectors. Instead, there is access to a source of noisy and incomplete dissimilarity information between objects.

RKE estimates a symmetric positive semidefinite kernel matrix K which induces a real squared distance admitting of an inner product. K is the solution to an optimization problem with semidefinite constraints that trades-off fit to the observed dissimilarity data and a penalty of the form $\lambda_{rke}\mathrm{trace}(K)$ on the complexity of K, where λ_{rke} is a non-negative regularization parameter.

Given an RKE kernel K estimated from a training set of objects, the RKE framework provides the *newbie* method for embedding new objects into a low dimensional space spanned by K. The embedding is given as the solution of an optimization problem with semidefinite and second-order cone constraints which requires that the dimensionality of the embedding space is given as a parameter.

An example of a setting where RKE is suitable is the classification of protein sequence data where measures of dissimilarity are easily obtained, whereas feature vector representations are difficult to obtain or justify. Some sources of dissimilarity in this case, such as BLAST (Altschul et al., 1990), require setting a number of parameters that makes the resulting dissimilarities possibly inexact, inconsistent and noisy. The RKE method is robust to the type of noisy and incomplete data that arises in this setting.

In this chapter, we will show how this framework can be successfully applied to a protein classification task, where data consists of dissimilarity data between a number of proteins: 1) a sequence dissimilarity measure derived from BLAST (Altschul et al., 1990), 2) a dissimilarity derived from transcription factor occupancy data in promoter regions of genes. In the first case, each protein is labeled as belonging to one of two sub-families determined by low-level molecular structural features. In the second case, proteins are classified by their cellular localization. Using a kernel matrix estimated by RKE, we can successfully learn a Support Vector Machine that classifies these proteins into their respective classes based on pseudo-data vectors obtained from the estimated kernel matrix.

Appendix A contains results on methods for choosing values of the regularization parameter λ_{rke} in the RKE problem. We show the CV2 method which selects regularization parameter values in clustering and visualization applications. Based on a empirical study using a modified version of the protein sequence data, we make the observation that similar clustering performance is achievable for a range of values of the RKE regularization parameter, indicating that precise tuning in these applications might not be required. However, based on the same empirical study we make the observation that classification performance, in contrast to clustering, may be highly dependent on the RKE regularization parameter. This indicates that methods that jointly tune regularization parameters in both the RKE and classification optimization problems are required. Furthermore, we present a simulation study that furthers demonstrate this phenomenon, where clustering is relatively invariant to a large range of tuning parameter values, whereas classification must be tuned carefully to obtain optimal prediction performance.

4.1 Regularized Kernel Estimation

The RKE framework provides a unified solution to two problems: 1) *The RKE Problem* estimating full relative position information for a set of objects, preferably in a low dimensional space with the purpose of visualization or further processing such as clustering or classification, and 2)

The Newbie Problem embedding new objects in this estimated low dimensional space for the purpose of determining its relative position to training objects or for classification given a classification function over this embedding space.

RKE problem Given a training set of N objects assume dissimilarity information is given for a subset Ω of size r of the $\binom{N}{2}$ possible pairs of objects. Denote the dissimilarity between objects i and j as $d_{ij} \in \Omega$. We make the requirement that Ω satisfies a connectivity constraint: the undirected graph consisting of objects as nodes and edges between them, such that an edge between nodes i and j is included if $d_{ij} \in \Omega$, is connected. Additionally, optional weights w_{ij} may be associated with each $d_{ij} \in \Omega$.

RKE estimates an N-by-N symmetric positive semidefinite kernel matrix K of size N, such that, the fitted distance between objects induced by K, $\hat{d}_{ij} = K(i,i)+K(j,j)-2K(i,j)$, is as close as possible to the observed distance $d_{ij} \in \Omega$. Formally, RKE solves the following optimization problem with semidefinite constraints:

$$\min_{K \succeq 0} \sum_{d_{ij} \in \Omega} w_{ij} |d_{ij} - \hat{d}_{ij}| + \lambda_{rke} \mathrm{trace}(K). \tag{4.1}$$

The parameter $\lambda_{rke} \geq 0$ is a regularization parameter that trades-off fit of the dissimilarity data, as given by absolute deviation, and a penalty, $\mathrm{trace}(K)$, on the complexity of K. The trace may be seen as a proxy for the rank of K, therefore RKE is regularized by penalizing high dimensionality of the space spanned by K. Note that the trace was used as a penalty function by Lanckriet et al. (2004a).

The Newbie Algorithm Given an RKE kernel K_N estimated as above, assume that Γ_x contains dissimilarity information between new object x and a subset of the N training set objects, thus, $d_{xj} \in \Gamma_x$ where $j \in \{1, \ldots, N\}$. Optionally, weights w_{xj} may be associated with each $d_{xj} \in \Gamma_x$. The kernel matrix K_N is,sub-optimally, extended to embed x in the space spanned by K_N. Formally we find K_x of the form:

$$K_x = \begin{pmatrix} K_N & b \\ b' & c \end{pmatrix}$$

that solves the optimization problem:

$$\min_{c\in\mathbb{R}\ b\in\mathbb{R}^N} \quad c + \sum_{d_{xj}\in\Gamma_x} w_{xj}|d_{xj} - \hat{d}_{xj}| \tag{4.2}$$

$$\text{s.t} \qquad b \in \text{range}(K) \tag{4.3}$$

$$c - b'K^{\dagger}b \geq 0, \tag{4.4}$$

where b' is the transpose of column vector b and $K^{\dagger}$ is the pseudo-inverse of K. The constraints on c and b are necessary and sufficient for K_x to be positive semidefinite. Eq. 4.2 can be formulated as a problem with semidefinite and second-order cone constraints. The Newbie Algorithm takes as a parameter the dimensionality of the embedding space.

4.2 Using RKE for Classification

In the setting where classification of objects is desired based on noisy dissimilarity data, we take the approach of using solutions to the RKE problem as kernel matrices to fit a Support Vector Machine (SVM) (Scholkopf and Smola, 2002; Vapnik, 1998). Let $y = (y_1, \ldots, y_N)'$ be a labeling of the N objects used to estimate an RKE kernel K. We find a function f of the form $f_\lambda(x) = \sum_{i=1}^{N} c_i K(x, i) + d$ where $K(x, i)$ is the corresponding entry for an RKE kernel K for objects x and i. For an SVM, f is the solution of the following optimization problem:

$$\min_{c\in\mathbb{R}^N, d\in\mathbb{R}} \sum_{i=1}^{N} (1 - y_i f_i)_+ + \lambda_{svm} c' K c, \tag{4.5}$$

where $(\tau)_+ = \max 0, \tau$ is the hinge-loss function and $f_i = \sum_{j=1}^{N} c_j K(i, j) + d$ where i, j are pairs of objects in the training set.

The regularization parameter λ_{svm} trades off fidelity to the data given by hinge loss and the squared norm of the resulting classification function in the space induced by K. The generalization performance of an SVM is sensitive to both the choice of kernel and regularization parameter λ_{svm}, thus in a joint RKE-SVM system a method for choosing both regularization parameters λ_{rke} and λ_{svm} is required.

An initial approach is to base tuning for RKE-SVM systems on tuning criteria for SVMs, for example, the GACV (Wahba et al., 1999) criterion which approximates the leave-one-out (LOO)

error of an estimated SVM. The GACV can be shown to be equal to the Chapelle-Vapnik Support Vector Span rule (Chapelle and Vapnik, 1999; Vapnik and Chapelle, 2000) LOO estimate under certain conditions. Another candidate method is the $\xi\alpha$ method (Joachims, 2000) or its GACV-like approximation (Wahba et al., 2001). Appendix B gives a result which characterizes and compares these adaptive tuning methods.

4.3 Protein Classification

In this Section we extend the protein clustering task introduced by Lu et al. (2005) by applying the Regularized Kernel Estimation (RKE) framework to the task of protein classification. In addition, we present results in a second protein classification task where classes are determined by cellular localization and dissimilarity is given by transcription factor occupancy in the gene promoter region.

4.3.1 Classification by Structural Feature

The data set for low-level structural feature classification consists of the amino-acid sequence of 630 members of the globin protein family. This protein family is partitioned into sub-families, α and β-chains, according to known low-level structural features of the protein. For our experiments, we randomly chose 100 members each of the α and β-chain sub-families, as annotated in the SwissProt database (Gasteiger et al., 2003).

For each pair of protein sequences, we obtain a normalized global alignment score using the Bioconductor PairSeqSim package (Gentleman et al., 2006). We sample a set of dissimilarities from the $\binom{200}{2} = 19,900$ available similarities as follows: for each object we sample the dissimilarity with 20% of the remaining proteins chosen uniformly at random. This results in 3,994 dissimilarity measures. Given a value for λ_{rke}, we estimate a 200-by-200 kernel by solving the RKE problem 8.2 using the DSDP5 semidefinite solver (Benson et al., 2000).

Figure 4.1 shows the result of embedding the 200 objects into the space induced by the kernel estimated with $\log_{10}(\lambda_{rke}) = 0.5$. Members of the α-chain sub-family are displayed as red crosses, while members of the β-chain family are displayed as blue circles. This two-dimensional

embedding was obtained by projecting the kernel matrix to its two leading eigenvectors. In fact, in Figure 4.2 we can see that the two leading eigenvectors of K dominate its eigenspectrum.

By inspecting Figure 4.1, we can see that a linear classifier can achieve perfect classification of these proteins. To prove this, we fit an Support Vector Machine spanned by the estimated kernel (with $\log_{10}(\lambda_{rke}) = 0.5$, for example). To reduce the complexity of the SVM spanning space, we make the kernel rank-deficient, and in effect embedding the proteins in a low-dimensional Euclidean space. We determine this dimensionality of embedding by using the kernel's eigenspectrum: we set all eigenvalues of K smaller than $1e^{-8}$ times the largest eigenvalue to zero and embed the data in the space spanned by the remaining eigenvectors. For $\log_{10}(\lambda_{rke}) = 0.5$, we find that the SVM is capable of classifying the data perfectly. The regularization parameter λ_{svm} was chosen using the GACV approximation of misclassification rate (Wahba et al., 2001).

Figure 4.3 shows the error rate of the estimated SVM as a function of the RKE regularization parameter λ_{rke} derived using ten-fold cross-validation. Figure 4.4 shows the dimensionality of embedding used for each SVM as a function of the regularization parameter. We can see that regularization parameter values $\lambda_{rke} < 10^2$ the RKE-SVM achieves perfect prediction. Furthermore, for values close to $\lambda_{rke} = 1$, this prediction performance can be achieved using an embedding dimensionality much smaller than $n = 200$.

4.3.2 Classification by Cellular Localization

Next, we apply Regularized Kernel Estimation and SVMs to a cellular localization protein classification task. (Lanckriet et al., 2004b). Genome-wide location profiles of 106 yeast transcription factors have recently been generated by Lee et al. (2002). These experiments provided for each gene[1] a measure of regulatory region occupancy (log ratio of the Ip-enriched versus control signalled averaged over three replicate experiments) for each of the 106 transcription factors. As a measure of dissimilarity we used the cosine angle measure, commonly employed in the cluster analysis of the gene expression data, between pairs of genome-wide location profiles.

[1] Specifically, each identified open reading frame (ORF)

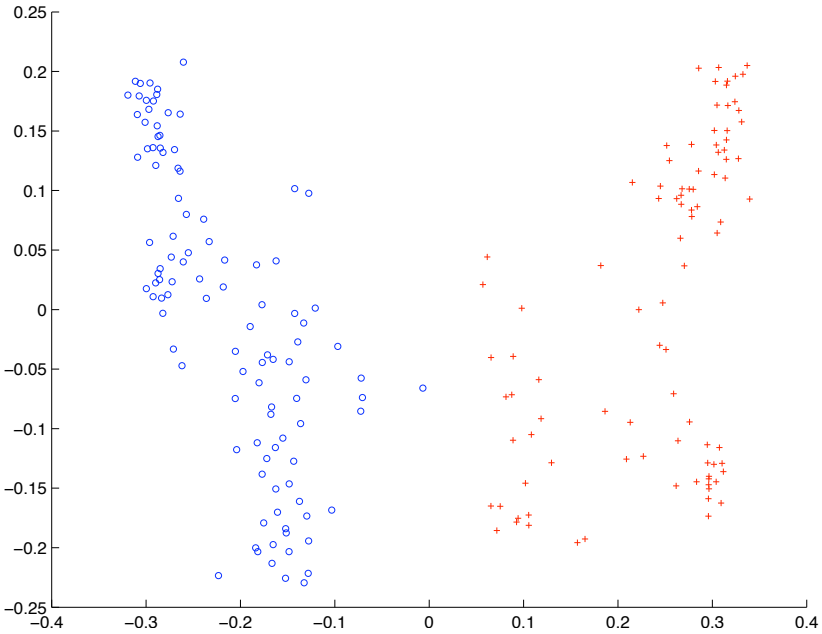


Figure 4.1 Embedded protein sequence data for $\log_{10}(\lambda_{rke}) = 0.5$.

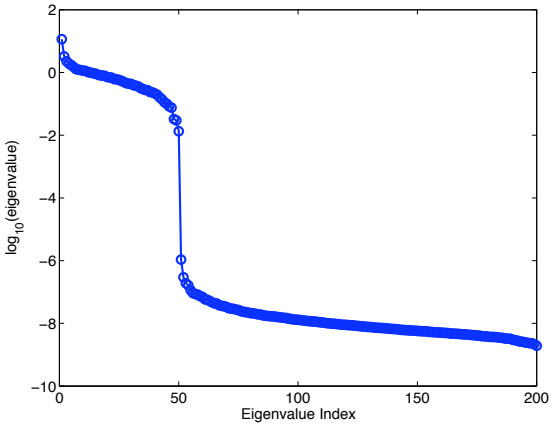


Figure 4.2 Eigenspectrum of estimated kernel for $\log_{10}(\lambda_{rke}) = 0.5$.

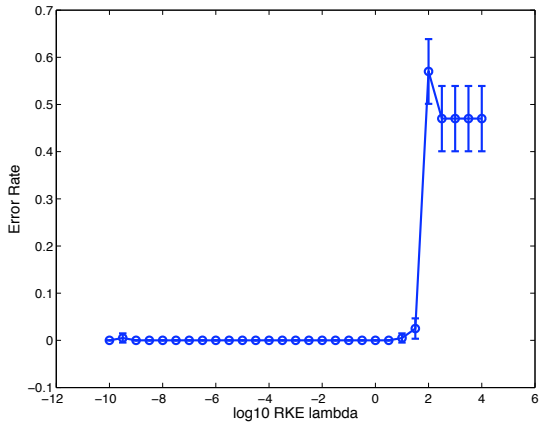


Figure 4.3 SVM misclassification rate using kernel estimated with given $\log_{10}(\lambda_{rke})$.

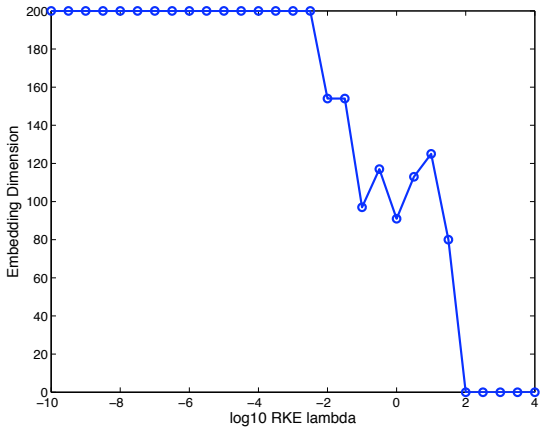


Figure 4.4 Embedding dimensionality for given $\log_{10}(\lambda_{rke})$

The task is to classify each protein as ribosomal or not, that is, is it located in the cell's ribosomes or elsewhere. This classification is known for 1040 of the 6112 proteins in the data set used in Lanckriet et al. (2004b), of which 132 (13%) are classified as positive. We created a balanced sample of size 264 such that half of the proteins in the sample are positive and half are negative. Thus, this includes all the ribosomal proteins and a random sample of non-ribosomal proteins.

To use RKE for this task we sampled the occupancy dissimilarities as follows: for each protein we randomly connect 40% percent of the remaining proteins in the relationship graph. Thus, only about 40% of the distance information is used to create the RKE kernel.

We use a transductive learning setting where the RKE kernel is created using both training and testing data. However, for each of the cross-validation folds, the SVM is estimated only using the kernel submatrix for the training data, and prediction performance estimated on the held-out test set. The SVM parameter was chosen using GACV (Wahba et al., 2001). As in the previous task, we choose the embedding dimensionality by keeping eigenvalues that are greater than 10^{-8} times the biggest eigenvalue. Given a value for λ_{rke}, we estimate a 264-by-264 kernel by solving the RKE problem 8.2 using the DSDP5 semidefinite solver (Benson et al., 2000).

Figure 4.5 shows the test set error in this task as function of the λ_{rke} regularization parameter. We see that although a relatively wide range of parameters show similar result, there is a region where underperformance occurs. As opposed to the previous task, this points to the need of careful tuning when using RKE for prediction.

4.4 Discussion

We have shown how the RKE framework can be used to successfully classify proteins in two distinct protein classification tasks. Furthermore, we have shown the generality of the RKE framework where two very different dissimilarity measures are used in each task: one based on sequence information, the other on experimental transcription factor occupancy.

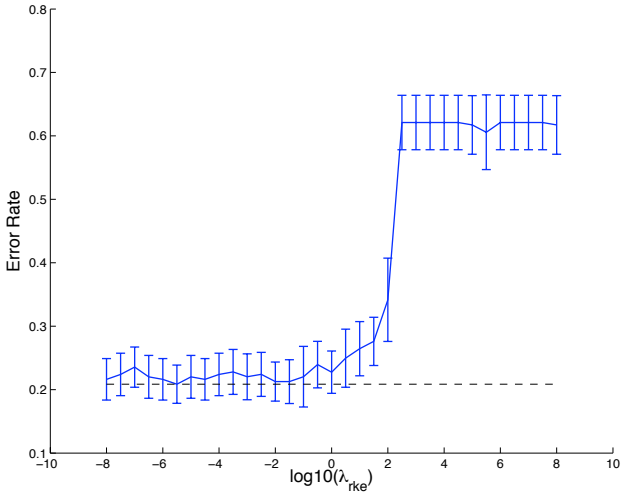


Figure 4.5 Test set error for the cellular localization task as a function of the RKE regularization parameter λ_{rke}

Part III

MPF Queries: Decision Support and Probabilistic Inference

Chapter 5

MPF Queries: Decision Support and Probabilistic Inference

5.1 Introduction

Recent proposals for managing uncertain information require the evaluation of probability measures defined over a large number of discrete random variables. The next three chapters present MPF queries, a broad class of aggregate queries capable of expressing this probabilistic inference task. By optimizing query evaluation in the MPF (Marginalize a Product Function) setting we provide direct support for scalable probabilistic inference in database systems. Further, looking beyond probabilistic inference, we define MPF queries in a general form that is useful for Decision Support, and demonstrate this aspect through several illustrative queries.

The MPF setting is based on the observation that functions over discrete domains are naturally represented as relations where an attribute (the value, or measure, of the function) is determined by the remaining attributes (the inputs, or dimensions, to the function) via a Functional Dependency (FD). We define these *Functional Relations*, and present an extended Relational Algebra to operate on them. A view V can then be created in terms of a stylized join of a set of 'local' functional relations such that V defines a joint function over the union of the domains of the 'local' functions. MPF queries are a type of aggregate query that computes view V's joint function value in arbitrary subsets of its domain:

```
select Vars, Agg(V[f]) from V group by Vars.
```

In the rest of this chapter, we outline the probabilistic inference problem and explain the connection to MPF query evaluation, and illustrate the value of MPF queries for decision support.

5.1.1 Probabilistic Inference as Query Evaluation

Consider a joint probability distribution P over discrete random variables A, B, C and D (see Section 5.3 for an example). The probabilistic inference problem is to compute values of the joint distribution, say $P(A = a, B = b, C, D)$, or values from conditional distributions, $P(A|B = b, C = c, D = d)$ for example, or values from marginal distributions, for example $P(A, B)$. All of these computations are derived from the joint distribution $P(A, B, C, D)$. For example, computing the marginal distribution $P(A, B)$ requires summing out variables C and D from the joint.

Since our variables are discrete we can use a relation to store the joint distribution with a tuple for each combination of values of A, B, C and D. The summing out operation required to compute marginal $P(A, B)$ can then be done using an aggregate query on this relation. However, the size of the joint relation is exponential in the number of variables, making the probabilistic inference problem potentially expensive.

If the distribution was "factored" (see Section 5.3 for specifics) the exponential size requirement could be alleviated by using multiple smaller relations. Existing work addresses how to derive suitable factorizations (Heckerman, 1999), but that is not the focus of this paper; we concentrate on the inference task.

Given factorized storage of the probability distribution, probabilistic inference still requires, in principle, computing the complete joint before computing marginal distributions, where reconstruction is done by multiplying distributions together. In relational terms, inference requires reconstructing the full joint relation using joins and then computing an aggregate query. This chapter addresses how to circumvent this requirement by casting probabilistic inference in the MPF setting, that is, as aggregate query evaluation over views. We will see conditions under which queries can be answered without complete reconstruction of the joint relation, thus making probabilistic inference more efficient. By optimizing query evaluation in a relational setting capable of expressing probabilistic inference, we provide direct scalable support to large-scale probabilistic systems. For a more complete discussion of Bayesian Networks and inference using MPF queries, see Section 5.3.2.

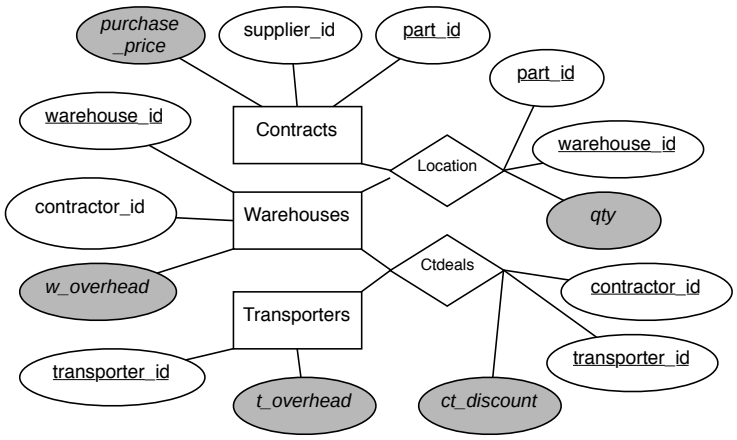


Figure 5.1 A supply chain decision support schema. Entity relations are rectangles, Relationship relations are diamonds. Attributes are ovals, with measure attributes shaded.

5.1.2 MPF Queries and Decision Support

So far, we have emphasized the relationship between the MPF setting and probabilistic inference. However, MPF queries can be used in a broader class of applications. Consider the enterprise schema shown in Figure 5.1:

1) *Contracts:* stores terms for a part's purchase from a supplier;

2) *Warehouses:* each warehouse is operated by a contractor and has an associated multiplicative factor determining the storage overhead for parts;

3) *Transporters:* transporters entail an overhead for transporting a part;

4) *Location:* the quantity of each part sent to a warehouse;

5) *Ctdeals:* contractors may have special contracts with transporters which reduce the cost of shipping to their warehouses when using that transporter.

Since contracts with suppliers, storage and shipping overheads, and deals between contractors and transporters are not exclusively controlled by the company, it draws these pieces of information from diverse sources and combines them to make decisions about supply chains.

Total investment on each supply chain is given by the product of these base relations for a particular combination of dimension values. This can be computed by the following view:

```
create view invest(pid,sid,wid,cid,tid,inv) as
  select pid, sid, wid, cid, tid,
      (p_price * w_overhead * t_overhead * qty * ct_discount) as inv
  from contracts c, warehouses w, transporters t, location l,ctdeals ct
  where c.pid = l.pid and l.wid = w.wid ...
```

Now consider querying this view, not for a complete supply chain, but rather, only for each part. For example, we may answer the question *What is the minimum supply chain investment on each part?* by posing the MPF query:

```
select pid, min(inv) from invest group by pid
```

Several additional types of queries over this schema are natural: *What is the cost of taking warehouse w1 offline? What is the cost of taking warehouse w1 offline if, hypothetically, part p1 had a 10% lower price?* See Section 5.2.2.

5.2 MPF Setting Definition

We now formalize the MPF query setting. First, we define functional relations:

Definition 5.1 Let s be a relation with schema $\{A_1, \ldots, A_m, f\}$ where $f \in \mathbb{R}$. Relation s is a **functional relation** (FR) if the Functional Dependency $A_1 A_2 \cdots A_m \rightarrow f$ holds. The attribute f is referred to as the **measure** attribute of s.

We make several observations about FRs. First, any dependency of the form $A_i \rightarrow f$ can be extended to the maximal FD in Definition 5.1 and is thus sufficient to define an FR. Second, we do not assume relations contain the entire cross product of the domains of $A_1, \ldots, A_m$, although this is required in principle for probability measures. We refer to such relations as *complete*. Finally, any relation can be considered an FR where f is implicit and assumed to take the value 1.

Functional relations can be combined using a stylized join to create functions with larger domains. This join is defined with respect to a product operation on measure attributes:

Definition 5.2 Let s_1 and s_2 be functional relations, the **product join** of s_1 and s_2 is defined as:

$$s_1 \overset{*}{\bowtie} s_2 = \pi_{\mathrm{Var}(s_1)\cup\mathrm{Var}(s_2),s_1[f]*s_2[f]}(s_1 \bowtie s_2),$$

where $\mathrm{Var}(s)$ is the set of non-measure attributes of s.

This definition is clearer when expressed in SQL:

```
select A1,...,Am,(s1.f * s2.f) as f
from s1,s2
where s1.A1 = s2.A1,..., s1.Ak = s2.Ak
```

where $\{A1,\ldots,Am\} = \mathrm{Var}(s_1) \cup \mathrm{Var}(s_2)$, and
$\{A1,\ldots,Ak\} = \mathrm{Var}(s_1) \cap \mathrm{Var}(s_2)$.

Implicit in the Relational Algebra expression for product join are the assumptions that tables define a unique measure, and that measure attributes are never included in the set of join conditions. Note that the domain of the resulting joined function is the union of the domains of the operands, and that the product join of two FRs is itself an FR.

We propose the following SQL extension for defining views based on the product join:

```
create mpfview r as
  (select vars, measure = (* s1.f,s2.f,...,sn.f)
  from s1, s2, ..., sn
  where joinquals)
```

where the last argument in the select clause lists the measure attributes of base relations and the multiplicative operation used in the product join. This simplifies syntax and makes explicit that a single product operation is used in the product join.

For example, our decision support schema can be defined as:

```
create mpfview invest(pid,sid,wid,cid,tid,inv) as
  select pid, sid, wid, cid, tid,
      measure=(* p_price, w_overhead, t_overhead, qty, ct_discount) as inv
  from contracts c, warehouses w, transporters t, location l,ctdeals ct
  where c.pid = l.pid and l.wid = w.wid ...
```

5.2.1 MPF Queries

We are now in position to define MPF queries.

Definition 5.3 MPF Queries. Given view definition r over base functional relations s_i, $i = 1, 2, \ldots, n$ such that $r = s_1 \overset{*}{\bowtie} s_2 \overset{*}{\bowtie} \cdots \overset{*}{\bowtie} s_n$, compute

$$\pi_{X,\mathrm{AGG}(r[f])}\mathrm{GroupBy}_X(r)$$

where $X \subseteq \bigcup_{i=1}^{n} \mathrm{Var}(s_i)$, and AGG is an aggregate function. We refer to X as the **query variables**.

Note that the result of an MPF query is an FR, thus MPF queries may be used as subqueries defining further MPF problems.

To clarify the definition, we have not specified the MPF setting at its full generality. FRs may contain more than a single measure attribute as long as the required functional dependency holds for each measure attribute. For simplicity of presentation, all examples of FRs we use will contain a single measure attribute. Also, the requirement that the measure attribute f is real-valued ($f \in \mathbb{R}$) is not strictly necessary. However, f must take values from a set where a multiplicative and an additive operation are defined in order to specify the product operation in product join and the aggregate operation in the MPF query. For the real numbers we may, obviously, take $\times$ as the multiplicative operation and $+$, min or max as the additive operation. Another example is the set $\{0, 1\}$ with logical $\wedge$ and $\vee$ as the multiplicative and additive operations.

For the purposes of query evaluation, significant optimization is possible if operations are chosen so that the multiplicative operation distributes with respect to the additive operation. This corresponds to the condition that the set from which f takes values is a commutative semi-ring (Aji and McEliece, 2000; Kschischang et al., 2001). Both the real numbers and $\{0, 1\}$ with their corresponding operations given in the previous paragraph possess this property.

5.2.2 MPF Query Forms

We can identify a number of useful MPF query variants that arise frequently. Using the schema in Figure 5.1, we present templates and examples for variants in a decision support context. In the following, we assume that r is as in Definition 5.3.

Basic: This is the query form used in the definition of MPF queries above:

```
select X,AGG(r.f) from r group by X
```

Example: What is the minimum investment on each part?

```
select pid, min(inv) from invest group by pid
```

Restricted answer set: Here we are only interested in a subset of a function's measure as given by specific values of the query variables. We add a `where X=c` clause to the Basic query above. *Example*: How much would it cost for warehouse w1 to go off-line?

```
select wid, sum(inv) from invest where wid=w1
  group by wid
```

Constrained domain: Here we compute the function's measure for the query variables conditioned on given values for other variables. We add a `where Y=c` clause to the Basic query with $Y \notin X$. *Example*: How much money would each contractor lose if transporter t1 went off-line?

```
select cid, sum(inv) from invest where tid=t1
  group by cid
```

The optimization schemes we present in Chapter 6 are for the three query types above. Of course, there are other useful types of MPF queries. Future work might consider optimizing the following types:

Constrained range: Here function values in the result are restricted. This is useful when only values that satisfy a given threshold are required. This is accomplished by adding a `having f<c` clause to the basic query.

The next two query types are of a hypothetical nature where alternate measure or domain values are considered.

Alternate measure: here the measure value of a given base relation is hypothetically updated. For example, how much money would contractor c1 lose if warehouse w1 went off-line if, hypothetically, part p1 was a different price?

Alternate domain: alternatively, variable values in base relations may be hypothetically updated. For example, how much money would contractor c1 lose if warehouse w1 went off-line under a hypothetical transfer of contractor c1's deal with transporter t1 to transporter t2?

5.3 MPF Queries and Probabilistic Inference

Modeling and managing data with uncertainty has drawn considerable interest recently. A number of models have been proposed by the Statistics and Machine Learning (Buntine, 1994; Friedman et al., 1999; Heckerman et al., 2004; Singla and Domingos, 2005), and Database (Burdick et al., 2005; Dalvi and Suciu, 2005, 2004; Fuhr and Rölleke, 1997) communities to define probability distributions over relational domains. For example, the DAPER formulation (Heckerman et al., 2004) extends Entity-Relationship models to define *classes* of conditional independence constraints and local distribution parameters.

5.3.1 Probabilistic Databases

Dalvi and Suciu (Dalvi and Suciu, 2004; Ré et al., 2006b), and Ré et al. (2006a,b) define a representation for probabilistic databases (Fuhr and Rölleke, 1997), and present an approximate procedure to compute the probability of query answers. They represent probabilistic relations as what we have called functional relations, where each tuple is associated with a probability value. Queries are posed over these functional relations, with the probability of each answer tuple given by the probability of a boolean formula. Ré et al. (2006a) define a middleware solution to approximate the probability of the corresponding boolean formula.

A significant optimization in their framework pushes evaluation of suitable subqueries to the relational database engine. These subqueries are identical to MPF queries, that is, aggregate queries over the product join of functional relations. Thus, their optimization is constrained by the engine's ability to process MPF queries. Our optimization algorithms in Chapter 6 allow for significantly

more efficient processing of these subqueries than existing systems, thus improving the efficiency of their middleware approximation method.

They specify two aggregates used in these subqueries: SUM, and PROD, where $\mathrm{PROD}(\alpha, \beta) = 1-(1-\alpha)(1-\beta)$. Optimization of the SUM case is handled directly by the algorithms we present, but the distributivity assumptions we require for optimization (see Chapter 6) are violated by the PROD aggregate, since $\mathrm{PROD}(\alpha\beta, \alpha\gamma) \neq \alpha\mathrm{PROD}(\beta, \gamma)$. However, we may bound the non-distributive PROD aggregate as follows:

$$\alpha\mathrm{PROD}(\beta, \gamma) \leq \mathrm{PROD}(\alpha\beta, \alpha\gamma) \leq 2\alpha \max(\beta, \gamma).$$

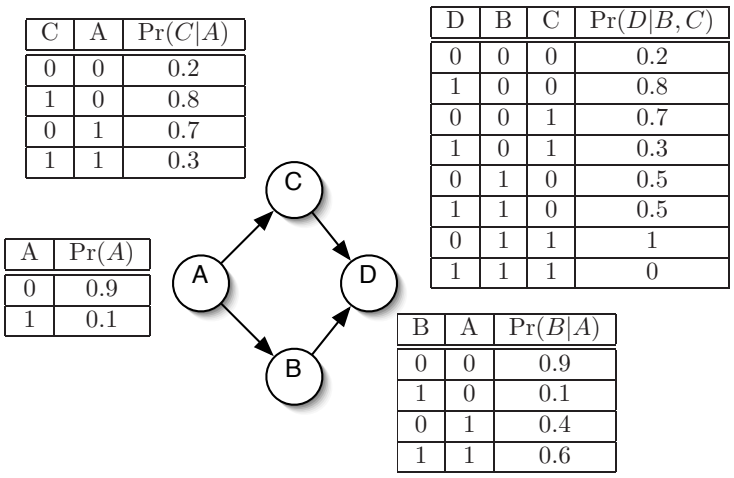
We can compute each of the two bounds in the MPF setting, so optimization is possible. In cases where this loss of precision is allowable, ranking applications for example, the gains of using the MPF setting is significant due to its optimized evaluation.

5.3.2 Bayesian Networks

In general, we can use the MPF setting to represent discrete multivariate probability distributions that satisfy certain constraints. In this section, we show how MPF queries can be used to query Bayesian Network (BN) models of uncertain data. BNs (Cowell et al., 1999; Jensen, 2001; Pearl, 1988) are widely-used probabilistic models that satisfy some conditional independence properties that allow the distribution to be factored into local distributions over subsets of random variables.

To understand the intuition behind BNs, consider a probabilistic model over the cross product of large discrete domains. A functional relation can represent this distribution but its size makes its use infeasible. However, if the function was factored, we could use the MPF setting to express the distribution using smaller local functional relations. For probability distributions, factorization is possible if some conditional independence properties hold; a BN represents such properties graphically.

Consider binary random variables A, B, C, D. A functional relation of size 2^4 can be used to represent a joint probability distribution. If, however, a set of conditional independencies exists



C	A	$\Pr(C\|A)$
0	0	0.2
1	0	0.8
0	1	0.7
1	1	0.3

A	$\Pr(A)$
0	0.9
1	0.1

D	B	C	$\Pr(D\|B,C)$
0	0	0	0.2
1	0	0	0.8
0	0	1	0.7
1	0	1	0.3
0	1	0	0.5
1	1	0	0.5
0	1	1	1
1	1	1	0

B	A	$\Pr(B\|A)$
0	0	0.9
1	0	0.1
0	1	0.4
1	1	0.6

Figure 5.2 A simple Bayesian Network

such that

$$\Pr(A, B, C, D) = \Pr(A)\Pr(B|A)\Pr(C|A)\Pr(D|B, C)$$

then the BN in Figure 5.2 may be used instead. For this admittedly small example, the gains of factorization are not significant, but for a large number of large domains, factorization can yield a significant size reduction. The joint distribution is specified by the MPF view:

```
create mpfview joint as (
  select A,B,C,D, measure = (* tA.p, tB.p, tC.p, tD.p) as p
  from tA, tB, tC, tD
  where tA.A=tB.A and tA.A=tC.A ...  )
```

The set of conditional independence properties that induce a factorization may be given by domain knowledge, or estimated from data (Heckerman, 1999). Given the factorization, the local function values themselves are estimated from data (Heckerman, 1999). In either case, counts from data are required to derive these estimates. For data in multiple tables, where a join dependency holds, the MPF setting can be used to compute the required counts.

After the estimation procedure computes the local functional relations we can use MPF queries to infer exact values of marginal distributions. An example inference task is given by the MPF query

```
select C,SUM(p) from joint where A=0 group by C
```

which computes the marginal probability distribution of variable C when $A = 0$ is observed, $\Pr(C|A = 0)$.

5.3.3 Discussion and Related Work

Wong (2001); Wong et al. (1995, 2003) address the probabilistic inference task in relational terms and propose an extended relational model and algebra that expresses exactly this problem. The MPF setting we present here is a generalization and reworking of their formulation. A major benefit of framing this task in a relational setting is that existing and new techniques for efficient query evaluation can then be used. This opportunity has not, to the best of our knowledge, been investigated; our study of MPF query optimization in Chapters 6 and 7 is a first step in this direction.

To conclude, we have introduced the MPF class of queries and showed its value in a variety of settings. Our work is an early step in synthesizing powerful ideas from database query evaluation and probabilistic inference. A number of models have recently been proposed for defining probability distributions over relational domains, e.g., Plate Models (Buntine, 1994), PRMs (Friedman et al., 1999), DAPER (Heckerman et al., 2004), and MLNs (Singla and Domingos, 2005). Applying MPF query optimization to directly support inference in such settings is a promising and valuable next step.

Theoretical properties of MPF queries, for example, the complexity of deciding containment, are intriguing. While general results for arbitrary aggregate queries exist, we think that the MPF setting specifies a constrained class of queries that might allow for interesting and useful results.

Chapter 6

Single MPF Query Optimization

Section 5.2 hinted at the optimization benefit possible when MPF views and queries are defined over domains with operations chosen such that the multiplicative operation distributes with respect to the additive operation. We develop this observation in this section. A generic algorithm has been proposed for efficiently solving MPF problems (Aji and McEliece, 2000; Kschischang et al., 2001) in non-relational settings. It makes use of this key distributive property to reduce the size of function operands, thus making evaluation more efficient. We may cast this in relational terms as follows: the Group By ('additive') operation distributes with the product join ('multiplicative') operation so that Group By operator nodes can be pushed down into the join tree thus reducing the size of join operands.

We study two algorithms and their variants that use the distributivity property to optimize MPF query evaluation by pushing down Group By nodes into join trees: *(CS)* Chaudhuri and Shim's algorithm for optimizing aggregate queries (Chaudhuri and Shim, 1994, 1996); *(CS+)* our simple extension of CS that yields significant gains over the original; *(VE)* the greedy heuristic Variable Elimination algorithm (Zhang and Poole, 1996) proposed for probabilistic inference; and *(VE+)* our extension to VE based on Chaudhuri and Shim's algorithm that finds significantly better plans than VE by being robust to heuristic choice. These algorithms optimize basic, restricted answer and constrained domain MPF query types. To the best of our knowledge, this is the first method to cast VE as a join tree transformation operation.

In this central section of the chapter, we will define and describe each of the optimization algorithms; present conditions under which evaluation plans can be restricted to the linear class, thus

Table 6.1 Example cardinalities and domain sizes

Table	# tuples
contracts	100K
warehouses	5K
transporters	500
location	1M
ctdeals	500K

Variable	# ids
part_ids	100K
supplier_ids	10K
warehouse_ids	5K
contractor_ids	1K
transporter_ids	500

avoiding the extra overhead of searching over nonlinear plans[1]; we will characterize and compare the plan spaces explored by each of the algorithms given and show that the plan space explored by CS+ contains the space explored by VE; we will analyze the optimization time complexity of the algorithms, and also give conditions based on schema characteristics where VE will have significantly lower optimization time complexity than CS+; we will extend VE so that its plan space is closer to the space of CS+ plans without adding much optimization overhead; and finally, we will propose a cost-based ordering heuristic for Variable Elimination.

6.1 MPF Query Evaluation Algorithms

In this section, we will define the CS and VE algorithms along with our extensions. We make use of the example schema in Figure 5.1 again, with Q1 as a running example:

```
Q1: select wid, SUM(inv) from invest group by wid;
```

and consider an instance with table cardinalities and variable domain sizes given in Table 6.1.

We need to define linear and nonlinear plans. In linear plans, every interior node in a join tree has at least one leaf node as a child. Conversely, in nonlinear plans both children of interior nodes may be interior nodes as well. Leaf nodes are base relations that appear in the query, whereas interior nodes are intermediate relations that result from performing join or Group By operations.

[1] We define linear and nonlinear plans in Section 6.1.

The CS Algorithm Chaudhuri and Shim (1994, 1996) define an optimization scheme for aggregate queries that pushes Group By nodes into join trees. The CS algorithm explores the space of linear plans using an extension of the dynamic programming optimization algorithm of Selinger et al. (1979). They also define a condition that ensures the semantic correctness of the plan transformation.

Algorithm 1 illustrates the CS procedure. As in Selinger's dynamic programming algorithm, $\mathrm{joinplan}()$ in line 2 finds the best linear plan that joins base relation r_j to the optimal plan for relation set S_j ($\mathrm{optPlan}(S_j)$). However, the usual algorithm is modified so that line 3 finds the best linear plan that joins r_j to the optimal plan for relation set S_j, this time modified to include a Group By node as its topmost node. Grouping in this added node is done on query variables and variables appearing in a join condition on any relation not yet joined into S_j. This ensures the semantic correctness of the plan transformation. The cheapest of these two candidate plans is selected in line 4. The authors showed that this greedy-conservative heuristic produces a plan that is no worse in terms of IO cost than the naïve plan with a single Group By node at the root of the join tree.

Algorithm 1 The CS optimization algorithm

1: **for all** r_j, S_j such that $Q' = S_j \cup \{r_j\}$ **do**
2: $\quad q_{1j} = \mathrm{joinplan}(\mathrm{optPlan}(S_j), r_j)$
3: $\quad q_{2j} = \mathrm{joinplan}(\mathrm{GroupBy}(\mathrm{optPlan}(S_j)), r_j)$
4: $\quad p_j = \mathrm{minCost}_i(q_{ij})$
5: **end for**
6: $\mathrm{optPlan}(Q') = \mathrm{minCost}_j(p_j)$

As defined, the CS procedure cannot evaluate MPF queries efficiently. It does not consider the distributivity of Group By and functional join nodes since it assumes that aggregates are computed on a single column and not on the result of a function of many columns. The resulting evaluation plan would be as in Figure 6.1, same as the best plan without any Group By optimization.

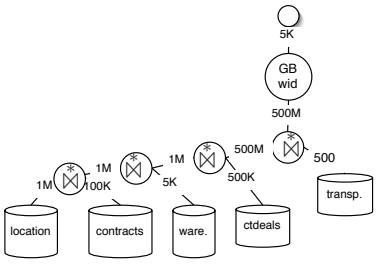


Figure 6.1 A CS plan for Q1

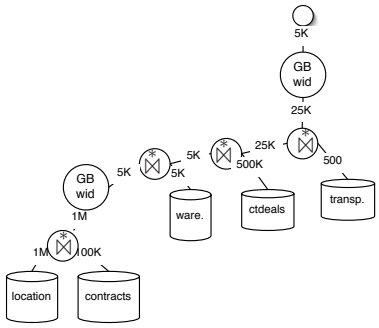


Figure 6.2 A CS+ plan for Q1

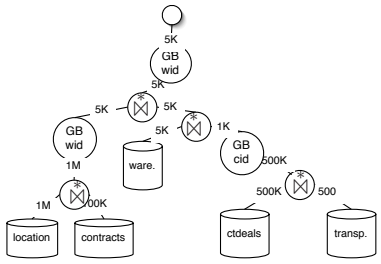


Figure 6.3 A VE plan for Q1

The CS+ Algorithm We make a simple extension to the CS algorithm, denoted CS+, that produces much better plans. In the CS+ algorithm, joins are annotated as product joins and the distributive property of the aggregate and product join is verified. As in the CS algorithm, Group By interior nodes must have as grouping variables both query variables and variables appearing in any join condition on any relation not yet joined into the current subplan. This again ensures the semantic correctness of the resulting plan. Figure 6.2 shows the CS+ plan for Q1. A Group By node is added after the join of *Location* and *Contracts* since the subplan joining *Warehouses* is cheaper.

The Nonlinear CS+ Algorithm We extend the CS+ procedure to consider nonlinear plans as follows: (a) for relation set S_j we consider joining every relation set of size $< j$; (b) we change $\mathrm{joinplan}()$ so that it returns the best nonlinear plan joining two relations; (c) instead of comparing two plans we now compare four: one without any Group By nodes (corresponding to line 2); another with a Group By on S_j (corresponding to line 3); another with a Group By on the operand (say, s') being joined to S_j; and finally, a plan with Group By nodes on both S_j and s'. The cheapest of these four plans is selected. From this point forward, will refer to this nonlinear extension as CS+.

The VE Algorithm Variable Elimination (Zhang and Poole, 1996) is based on a purely functional interpretation of MPF queries; our work is the first to apply VE to relational query optimization. The domain of the function defined by the MPF view is reduced one variable at a time until only the query variables remain. While this is an entirely different approach to query optimization, not based on transformations between equivalent Relational Algebra expressions, we can cast it in relational terms: to eliminate a variable, all the tables that include it are product-joined, and the result is aggregated and grouped by the variables that have not been eliminated so far. Algorithm 2 lists the VE algorithm. We denote the set of relations in S where variable v_j appears as $\mathrm{rels}(v_j, S)$. So $\mathrm{optPlan}(\mathrm{rels}(v_j, S))$ is the optimal plan found by the optimizer for joining the set of relations where variable v_j appears. We abuse notation slightly in line 9 where p denotes the relation resulting from executing plan p of line 6.

Algorithm 2 The Variable Elimination Algorithm

1: Set $S = \{s_1, s_2, \ldots, s_n\}$
2: Set $V = \mathrm{Var}(r) \setminus X$
3: set $p =$ null
4: **while** $V \neq \emptyset$ **do**
5: select $v_j \in V$ according to heuristic order
6: set $p = \mathrm{GroupBy}(\mathrm{optPlan}(\mathrm{rels}(v_j, S)))$
7: set $V = V \setminus \{v_j\}$
8: remove relations containing v_j from S
9: set $S = S \cup \{p\}$
10: **end while**

Figure 6.3 shows the VE plan for Q1 with elimination order *tid,pid,cid*. The efficiency of VE for query evaluation is determined by the variable elimination order (see Section 6.4). We again require that grouping in interior nodes contain query variables and variables required for any subsequent joins as grouping variables to ensure semantic correctness of the resulting plans. In VE this is satisfied by definition since query variables are not candidates for elimination and variables

are candidates for elimination as long as there is a relation in the current set that includes it in a join condition.

6.2 MPF Optimization Plan Spaces

6.2.1 Nonlinear MPF Query Evaluation

Including nonlinear plans in the space searched by an optimization algorithm for MPF queries is essential since there are join operand reductions available to these plans that are not available to linear plans. When query variables are of small domain, but appear in large tables, this is a significant advantage. The example plan in Figure 6.2 illustrates this point. Also note that the elimination order in Figure 6.3 induces a nonlinear join order. In fact, an advantage of VE is that it produces nonlinear plans with, usually, small optimization time overhead.

For an MPF query on variable X we can, conservatively, determine if a linear plan can efficiently evaluate it. We can check this using an expression that depends on the domain size of X, $\sigma_X = |X|$, and the size of the smallest base relation containing X, $\hat{\sigma}_X = \min_{s \in \text{rels}(X)} |s|$. Both of these statistics are readily available in the catalog of RDBMs systems. To see the intuition behind this test, consider the following example: X occurs in only two base relations s_1 and s_2, where $|s_1| > |s_2|$, thus $\hat{\sigma}_X = |s_2|$. A linear plan must, at best, join s_2 to an intermediate relation s' of size σ_X resulting from a join or Group By node where s_1 is already included. On the other hand, a nonlinear plan is able to reduce s_2 to size σ_X before joining to s'. Under a simple cost model where joining R and S costs $|R||S|$ and computing an aggregate on R costs $|R| \log |R|$, a linear plan is admissible if the following inequality holds:

$$\sigma_X^2 + \hat{\sigma}_X \log \hat{\sigma}_X \geq \sigma_X \hat{\sigma}_X. \tag{6.1}$$

6.2.2 Plan Spaces

We now turn to a characterization of the plan spaces explored by nonlinear CS+ and VE.

Definition 6.1 (Evaluation Plan Space $\mathscr{P}$) Denote as $\mathscr{P}$ the space of all nonlinear semantically correct evaluation plans where either Group By or join nodes are interior nodes, and are equivalent to a plan with only join interior nodes and a single Group By node at the root.

CS+ performs a complete (but bounded) search of nonlinear join orders using dynamic programming with a local greedy heuristic that adds interior Group By nodes.

Definition 6.2 (CS+ Plan Space $\mathscr{P}(CS+)$) Let $p \in \mathscr{P}$ have the following property: if a *single* interior Group By node is removed, the cost of the subplan rooted at its parent node is greater. We define $\mathscr{P}(CS+)$ to be the set of all plans in $\mathscr{P}$ that satisfy this property.

As we saw before, CS+ yields a plan that is no worse than the plan with a single Group By at the root.

Definition 6.3 (VE Plan Space $\mathscr{P}(VE)$) Let $p \in \mathscr{P}$ have the following properties for every non-query variable v: 1) a Group By node immediately follows the join node closest to the root where v appears as a join condition, and 2) all joins where v appears as a join condition are contiguous. We define $\mathscr{P}(VE)$ as the set of all plans in $\mathscr{P}$ that satisfy these properties.

VE does not guarantee optimality due to its greedy heuristic search, and it is known that finding the variable ordering that yields the minimum cost plan is NP-complete in the number of variables.

Theorem 6.4 characterizes these plan spaces. We say that $p \in \mathscr{P}(A)$ if optimization algorithm A either computes its cost, or can *guarantee* that there exists a plan $p' \in \mathscr{P}(A)$ that is cheaper than p. Although CS+ uses dynamic programming, its greedy heuristic for adding Group By nodes makes its search through $\mathscr{P}$ incomplete. Not surprisingly, the plan space searched by VE is also incomplete. However, we see that the plan space searched by CS+ includes the plan space searched by VE. That is, CS+ will consider the the minimum cost plan returned by VE for a given ordering.

Theorem 6.4 [Inclusion Relationships] Using the notation above, we have:

$$\mathscr{P} \supset \mathscr{P}(CS+) \supset \mathscr{P}(VE).$$

To prove this theorem, we need the following Lemma:

Lemma 6.5 Consider relations $S_n = \{r_1, \ldots, r_n\}$ where variable v only appears in r_k. Let

$$S'_n = \{r_1, \ldots, \mathrm{GroupBy}(r_k), \ldots, r_n\}.$$

For the CS+ algorithm, the following holds: for each output tuple ordering, $\mathrm{Cost}(\mathrm{optPlan}(S_n)) \leq \mathrm{Cost}(\mathrm{optPlan}(S'_n))$.

Proof. By induction on n. If $n = 2$, the Lemma follows since the plans are compared directly in line 4 of Algorithm 1. Now assume Lemma is true for $m \leq n-1$. If $r_k = r_n$ then the Lemma follows since, again, the plans are compared directly in line 4 of Algorithm 1. Otherwise, if $r_k \neq r_n$ then $r_k \in S_{n-1}$ we have by the inductive hypothesis $\mathrm{Cost}(\mathrm{optPlan}(S_{n-1})) \leq \mathrm{Cost}(\mathrm{optPlan}(S'_{n-1}))$ for each tuple ordering of S_{n-1} so the Lemma follows.

Proof. (Theorem 6.4)

($\mathscr{P}(CS+) \subseteq \mathscr{P}$) This follows by definition of CS+ and the semantic correctness of its plan transformation.

($\mathscr{P}(CS+) \neq \mathscr{P}$) By the greedy heuristic, any plan $p' \in \mathscr{P}$ extending the plan not chosen in line 4 of Algorithm 1 is not included in $\mathscr{P}(CS+)$. However, no guarantee is given that p' is more expensive than the plans extending the least expensive plan of line 4.

($\mathscr{P}(VE) \subseteq \mathscr{P}(CS+)$) Let p be the best VE plan for elimination order $v_1, \ldots, v_n$. We prove this statement by induction on n. If $n = 1$, the statement holds trivially. Now assume the statement is true for $m \leq n-1$ and consider variables v_m and v_n and $S_m = \mathrm{rels}(v_m, S)$. By the inductive hypothesis we have that the subplan in p that eliminates v_m is in $\mathscr{P}(CS+)$. But, since v_m only appears in the relation resulting from $\mathrm{optPlan}(S_m)$, by Lemma 6.5 we have that the subplan in p eliminating v_n is in $\mathscr{P}(CS+)$ as well. Thus $p \in \mathscr{P}(CS+)$.

($\mathscr{P}(VE) \neq \mathscr{P}(CS+)$) Consider a plan $p \in \mathscr{P}(VE)$ for a variable ordering where v_1 is preceded by v_2 but $\mathrm{rels}(v_1) \subseteq \mathrm{rels}(v_2)$. In this case, VE does not consider adding Group By nodes to eliminate v_1 in the subplan that eliminates v_2, but there exists a plan $p' \in \mathscr{P}(CS+)$ that attempts to add a Group By node to 'eliminate' v_1 once $\mathrm{rels}(v_1)$ are joined in p. Thus $p' \notin \mathscr{P}(VE)$.

6.2.3 Extending the Variable Elimination Plan Space

We saw in the previous Section that the plan space considered by VE is a subset of the plan space considered by CS+. In this section, we extend VE to narrow this gap by delaying the elimination of variables if that results in cheaper plans and by pushing Group By nodes into elimination sub-plans. We use Functional Dependency information to implement the delay strategy, and also use cost-based local decisions similar to those used by the CS+ algorithm to implement both the delay and pushing strategies.

As defined, VE considers all variables as candidates for elimination; however, the elimination of some variables might have no effect, that is, the result of Group By is the same as projection. In other words there is exactly one tuple for each group in the Group By clause. The following property captures this:

Proposition 6.6 Let r be an MPF view over base relations $s_1, \ldots, s_n$, and $Y \in \mathrm{Var}(r)$. If for each $i, 1 \leq i \leq n$ an FD $X_i \rightarrow s_i[f]$ holds where $X_i \subseteq \mathrm{Var}(s_i)$ and $Y \notin X_i$, then $\mathrm{GroupBy}_{\mathrm{Var}(r)\setminus Y}(r) = \pi_{\mathrm{Var}(r)\setminus Y}(r)$.

Proof. First, we note that for any functional relation s with $XY = \mathrm{Var}(s)$ where the FD $X \rightarrow s[f]$ holds, then $\mathrm{GroupBy}_{X'}(s) = \pi_{X'}(s)$ for all $X' \supseteq X$ since the FD implies that there is only one row per value of X'. By the condition that FD's $X_i \rightarrow s_i[f]$ hold, we have that $\cup_i X_i \rightarrow r[f]$ holds. That means we can partition $\mathrm{Var}(r)$ into $\cup_i X_i$ and Z with $Y \in Z$ and the Proposition follows.

A sufficient condition for Proposition 6.6 to apply is that primary keys are given for each base relation where Y is not part of any key. Furthermore, this Proposition holds for any set of relations, so in any iteration of the VE algorithm, if a variable satisfies the Proposition for the current set of relations, that variable can be removed from the set of elimination candidates. Applying this Proposition has the effect of avoiding the addition of unnecessary Group By nodes.

In the absence of FD information, we present an extension to Variable Elimination that uses cost-estimation to both delay variable elimination and push Group By nodes into elimination sub-plan join trees.

The VE+ Algorithm Algorithm 2 requires two changes: 1) in line 6 we set $p = \text{optPlan}(\text{rels}(v_j, S))$ to potentially delay elimination to later iterations of the algorithm, and 2) we assume that the function $\text{optPlan}()$ uses the local greedy conservative heuristic of CS+ to push Group By nodes into elimination subplan join trees. The first modification removes the Group By node in line 6 which eliminates the variable chosen at the current iteration. This is done so that the greedy heuristic of the second modification (from the CS algorithm) is used to decide on the addition of this Group By node if it yields a locally better plan.

These additions have the effect of extending $\mathscr{P}(VE)$ as follows:

Definition 6.7 (VE+ Plan Space $\mathscr{P}(VE+)$) Let $p \in \mathscr{P}$ satisfy the following conditions: 1) if a *single* interior Group By node is removed, the cost of the subplan rooted at its parent node is greater; and 2) for every non-query variable v all join nodes where v appears as a join condition are either contiguous or separated by only Group By nodes; that is, no join node where v does not appear as a join condition separates them. We define $\mathscr{P}(VE+)$ as the set of all plans that satisfy these properties.

Now we may update our inclusion relationship:

Theorem 6.8 (Extended VE Space) Using the notation above, we have:

$$\mathscr{P}(VE) \subset \mathscr{P}(VE+) \subset \mathscr{P}(CS+).$$

Proof. The proof is similar to that of Theorem 6.4.

($\mathscr{P}(VE) \subseteq \mathscr{P}(VE+)$) Given an elimination order, the same proof for $CS+$ and VE shows this case.

($\mathscr{P}(VE) \neq \mathscr{P}(VE+)$) Consider an elimination order where v_i follows v_j but $\text{rels}(v_i) \subset \text{rels}(v_j)$, $VE+$ considers adding Group By nodes to eliminate v_i while creating the plan for $\text{rels}(v_j)$, whereas VE does not. This is the same argument given above for VE and $CS+$.

($\mathscr{P}(VE+) \subseteq \mathscr{P}(CS+)$) The proof for this is the same as the proof of $\mathscr{P}(VE) \subseteq \mathscr{P}(CS+)$.

($\mathscr{P}(VE+) \neq \mathscr{P}(CS+)$) The issue here is that $VE+$ only considers plans where the joins for a given variable are contiguous, whereas $CS+$ does not follow that constraint. In the presence

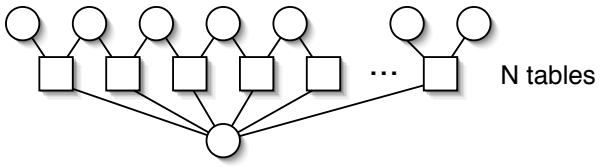


Figure 6.4 An example star MPF view.

of indices and alternative access methods, contiguous joins are not necessarily optimal, therefore $CS+$ is able to produce plans that are not reachable to $VE+$.

Although there is still a gap between $\mathscr{P}(VE+)$ and $\mathscr{P}(CS+)$ corresponding to plans where join nodes for a variable are not necessarily contiguous, our experimental results in Section 6.5 show that $CS+$ rarely produces plans that are not reachable by $VE+$.

6.3 Optimization Complexity

Another dimension of comparison between these procedures is time required to find optimum plans. Since search for optimal sub-plans in VE only occurs in line 6, for views where variables exhibit low connectivity, that is, variables appear only in a small subset of base relations, the cost of finding a VE plan is low.

As opposed to CS+, VE optimization time can be insensitive to variables that have high connectivity if average connectivity is low. Consider the star schema in Figure 6.4. This is the classic example where the optimization time of Selinger-type dynamic programming procedures degrades. In fact, the optimization time complexity for CS+ is $O(N2^N)$ for N relations. For VE with a proper ordering heuristic, only two relations have to be joined at a time for each variable, yielding optimization time complexity of $O(M)$ for M variables.

Theorem 6.9 summarizes these findings. We refer to an ordering heuristic for VE as proper if it orders variables by connectivity. Of course, while this guarantees good performance in terms of optimization time, it does not guarantee good performance in terms of query evaluation time since the resulting plan with a 'proper' heuristic might be sub-optimum.

Theorem 6.9 (Optimization Time Complexity) Let S be *average variable connectivity*, let M be the number of variables, and N the number of tables. The worst-case optimization time complexity of VE with a proper heuristic computable in linear time is $O(MS2^S)$. The worst-case optimization time complexity of CS+ is $O(N2^N)$.

Proof. The CS+ result is the standard complexity result for Salinger-type dynamic programming algorithms. For VE, a proper heuristic chooses a variable v_j in line 5 of Algorithm 2 where, on average, $|\text{rels}(v_j)| = S$. Finding a plan for these tables in line 6 takes $O(S2^S)$. At worst, this is done M times, once for each variable.

6.4 Elimination Heuristics

We now define statistics to decide heuristic variable elimination orderings.

Definition 6.10 Define the degree and width statistics for variable v as:

1. $degree(v) = |\text{GroupBy}(\text{optPlan}(\text{rels}(v, S)))|$;
2. $width(v) = |\text{optPlan}(\text{rels}(v, S))|$.

The degree heuristic orders variables increasingly according to estimates of the size of relation p in line 6 of Algorithm 2, while the width heuristic orders variables increasingly according to estimates of the size of p without its topmost Group By node.

In the VE literature (El Fattah and Dechter, 1996) these statistics are estimated by the domain sizes of variables. For example, the degree heuristic computes the size of the cross-product of the domains of variables in p. This is an effect of the fact that the cost metric minimized in VE, as defined in the MPF literature (Aji and McEliece, 2000; Kschischang et al., 2001), is the number of addition and multiplication operations used in evaluating the query. This is a valid cost metric in that setting since operands are assumed to be memory-resident, and more significantly, single algorithms are assumed to implement each of the multiplication and summation operations. These are not valid assumptions in the relational case where there are multiple algorithms to implement join (multiplication) and aggregation (summation), and the choice of algorithm is based on the

cost of accessing disk-resident operands. Thus, relational cardinality estimates are used in our implementation to compute these statistics.

The degree heuristic greedily minimizes the size of join operands higher in the join tree. However, there are cases where executing the plan that yields these small operands is costly, whereas plans that use a different order are less expensive. In this case, looking at estimates of the cost of eliminating a variable as an ordering heuristic is sensible:

Definition 6.11 Define the elimination cost statistic for variable v as

$$elimcost(v) = \text{Cost}(\text{optPlan}(\text{rels}(v, S)).$$

A straightforward way of implementing the elimination cost heuristic is to call the query optimizer on the set of relations that need to be joined to estimate the cost of the plan required to eliminate a variable. However, for this heuristic to be computed efficiently, both average variable connectivity *and* maximum variable connectivity must be much lower than the number of tables, otherwise Variable Elimination would exhibit the same optimization time complexity as CS+.

While *width* and *elimination cost* estimate the cost of eliminating variables, the *degree* heuristic seeks to minimize the cost of future variable eliminations. There is a trade-off between greedily minimizing the cost of the current elimination subplan vs. minimizing the cost of subsequent elimination sub-plans. To address this trade-off we combine the *degree* and either *width* or *elimination cost* heuristics by computing the mean of their normalized values. We study the effect of these heuristics and their combinations in Section 6.5.3.

To summarize the contributions of this central section: 1) we presented a necessary condition under which evaluation plans can be restricted to the linear class; 2) we characterized the plan spaces explored by each of the algorithms given; 3) we extended VE so that its plan space is closer to the space of CS+ plans without adding much optimization overhead; 4) we analyzed the optimization time complexity of both algorithms, and gave conditions based on schema characteristics where one would be better than the other; and 5) we proposed a cost-based ordering heuristic for Variable Elimination.

6.5 Experimental Results

We now present experimental results illustrating the discussion in Sections 6.2– 6.4. We modified the PostgreSQL 8.1 optimizer to implement each algorithm at the server (not middleware) level. The extensions in Section 5.2 were added to the PostgreSQL language. Experiments were performed on a 3 GHz Pentium IV Linux desktop with 2.4 GB of RAM and 38 GB of hard disk space. In most of these experiments, we do not compare the CS algorithm since its performance is substantially worse and distorts the scale of the plots, making it harder to see the relative performance of the other (much better) algorithms. However, the results in Section 6.5.4 make this comparison and illustrate the significant difference in performance.

We use two testbeds for our experiments. The first is the decision support schema of Figure 5.1 for which we create a number of instances at random. The *Contracts, Warehouses* and *Transporters* relations were populated according to a *Scale* parameter, whereas *Location* and *CTdeals* were populated according to *Density* parameters. The cardinalities and domain sizes in Table 6.1 correspond to *Scale* $= 100$, *Density(CTDeals)* $= 100\%$ and *Density(Location)* $= 20\%$. These are default settings unless specified otherwise. Non-key attributes in *Contracts* and *Warehouses*, compound keys in *Location* and *CTdeals* and all measure attributes are populated uniformly at random.

The second testbed consists of three variants of the Schema in Figure 6.4: a) a star view exactly like Figure 6.4, b) a linear view where the variable connecting all tables is removed, and c) a 'multistar' schema where instead of a single common variable there are multiple common variables each connecting to a distinct set of three tables in the linear part. The number of tables $N = 5$, all variables have domain size 10 and all functional relations are complete. Measure attributes are populated uniformly at random from the interval $[0, 1]$.

This section is organized as follows. First, in Section 6.5.1 we test the benefit of nonlinear evaluation of MPF queries and the linearity condition of Section 6.2.1. We will see that nonlinear evaluation performs better than linear evaluation except when linear plans are admissible as given by the linearity condition. In Section 6.5.2 shows how the extension of the Variable Elimination

algorithm given in Section 6.2.3 benefits evaluation. We will see that VE+ with the degree heuristic finds the optimal CS+ plan, while never finding a plan that is worse than VE. Section 6.5.3 illustrates the effect of elimination heuristics for Variable Elimination. We will see that schema characteristics are the main determinant of performance of each heuristic. However, we will also see that VE+ is robust to heuristic choice and is able to find near-optimal plans for all three heuristics we have defined. Finally, Section 6.5.4 tests the trade-off between optimization complexity and plan quality in each of the algorithms presented. We will see that all algorithms proposed produce better quality plans than existing systems while, in some cases, not adding significant optimization time. Furthermore, we will also see that schema characteristics are the main determinants of both quality and planning time for these algorithms.

6.5.1 Nonlinear Evaluation

Section 6.2.1 showed the benefit of nonlinear plans for MPF query evaluation. The experiment in Figure 6.5 illustrates how the plan linearity condition is applied. On our first testbed we run two queries:

```
Q1:select cid, SUM(inv) from invest group by cid;
Q2:select tid, SUM(inv) from invest group by tid;
```

We plot evaluation time as the *Density(CTdeals)* parameter is increased. For Q1, we see that as density increases nonlinear plans execute faster, whereas for Q2, a linear plan is optimal for all densities. Since the nonlinear version of CS+ also considers linear plans, the Q2 running times for both plans coincide. For Q1, we have that $\sigma_{cid} = 1000$ and $\hat{\sigma}_{cid} = 5000$, so the inequality in Eq. 6.1 does not hold, whereas for Q2, we have $\sigma_{tid} = \hat{\sigma}_{tid} = 500$ which makes the inequality hold showing the applicability of the linearity condition.

6.5.2 Extended Variable Elimination Space

Section 6.2.3 showed how to extend the VE plan space closer to that of nonlinear CS+. Figure 6.6 compares the resulting plan quality for CS+ and VE with the degree heuristic with and

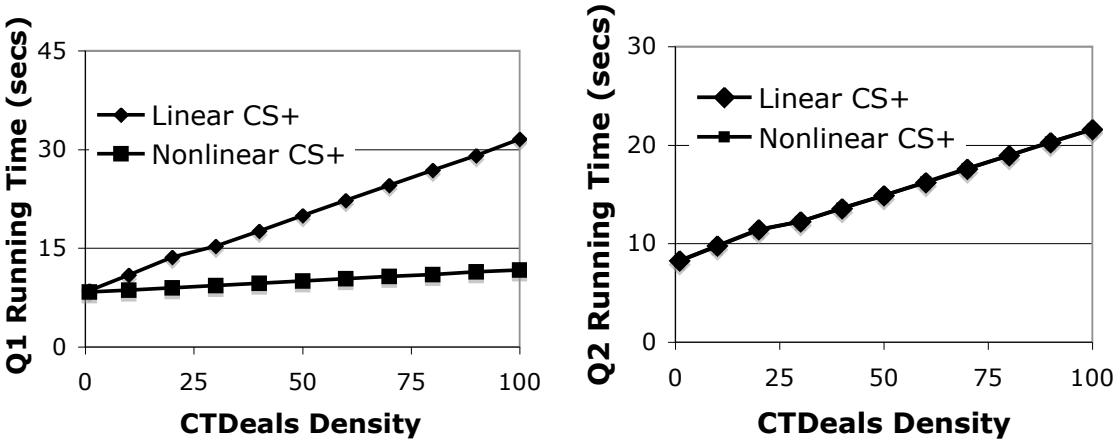


Figure 6.5 Plan Linearity Experiment

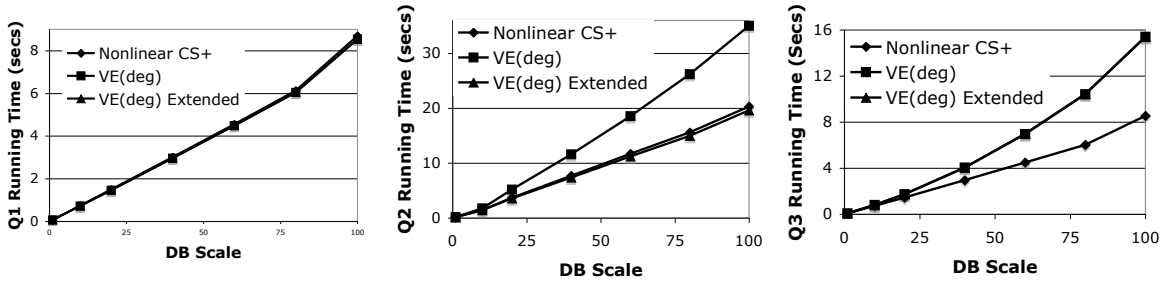


Figure 6.6 VE Extended Space Experiment

without the space extension. We ran the following three queries as the *Scale* parameter is increased:

```
Q1:select cid, SUM(inv) from invest group by cid;
Q2:select sid, SUM(inv) from invest group by sid;
Q3:select wid, SUM(inv) from invest group by wid;
```

For Q1, the degree heuristic produced the optimal CS+ nonlinear plan without the VE extension. For Q2, the degree heuristic produced a suboptimal plan, but with the space extension we obtain the optimal plan. Q3 is a different case where we have that the degree heuristic is not able to find the optimal plan even with the extended space. The VE+ extension to VE guarantees that we find a plan no worse than the plan obtained by VE without the extension; this is reflected in the results shown here.

6.5.3 Elimination Heuristics

We now show experimental results on the effect of ordering heuristic on plan quality for Variable Elimination. Using our first testbed, we run two queries and plot their running time as a function of the *Scale* parameter:

```
Q1:select cid, SUM(inv) from invest group by cid;
Q2:select pid, SUM(inv) from invest group by pid;
```

For Q1, the width heuristic yields a plan worse than both degree and elimination cost. Interestingly, width can be seen as an estimate of elimination cost, whereas degree seeks to minimize join

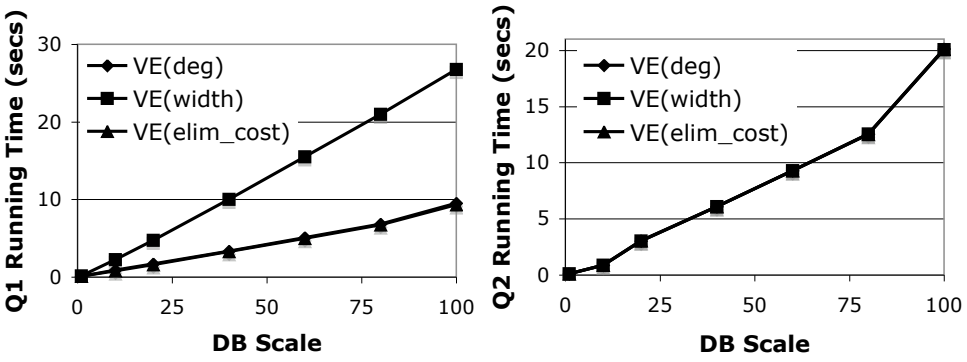


Figure 6.7 Ordering Heuristics Experiment

operands, or, equivalently, minimize the cost of future variable eliminations. For Q2, all heuristics derived the same plan.

Table 6.2 summarizes another experiment on order heuristics using our second testbed. A query on the first variable in the linear section was run on each schema. For each of the *degree, width* and *elimination cost* heuristics described in Section 6.4 we ran both the original VE algorithm and its extended space version described in Section 6.2.3. We implement the elimination cost heuristic using an overestimate: we fix a linear join ordering and allow choice of access paths and join operator algorithms. We also include results for combinations of the *degree and width* and *degree and elimination cost* heuristics[2]. We report the cost of the plan selected by the nonlinear CS+ algorithm, which is optimal in the plan space considered.

We see that for the star schema, the width heuristic performs best. This is not surprising since the degree heuristic will select the common variable first since after joining all of its corresponding tables, all but the query variable can be eliminated and the resulting relation is small (10 tuples). This requires joining all base tables, thus no Group By optimization is done. However, we see that by combining the degree and width heuristics we are able to produce a much better plan than degree but only slightly worse than width. The elimination cost heuristic performs better than the degree heuristic, but due to its overestimate, does not perform as well as the width heuristic. The difference in performance lessens as maximum variable connectivity drops.

[2]Combinations are implemented by normalizing each estimate and multiplying the normalized values

Table 6.2 Ordering Heuristics Experiment Result

Ordering	star	multistar	linear
Nonlinear CS+	**429.62**	**363.02**	**21.23**
VE(deg)	240225.15	843.84	34.57
VE(deg) ext.	**429.62**	**363.02**	**21.23**
VE(width)	705.03	593.43	34.57
VE(width) ext.	**429.62**	**363.02**	**21.23**
VE(elim_cost)	1045.44	936.34	73.78
VE(elim_cost) ext.	**429.62**	**363.02**	**21.23**
VE(deg & width)	950.44	843.84	34.57
VE(deg & width) ext.	**429.62**	**363.02**	**21.23**
VE(deg & elim_cost)	240225.15	843.84	34.57
VE(deg & elim_cost) ext.	**429.62**	**363.02**	**21.23**

Table 6.3 Random Heuristic Experiment Result

Schema	VE(rand)	VE(rand) ext.
star	30830.42 ± 1470.78	770.78 ± 5.60
multistar	11730.35 ± 298.86	4559.58 ± 149.03
linear	72.04 ± 0.29	51.78 ± 0.36

Interestingly, for all schemas, the extended VE algorithm with any heuristic produces optimal plans. This might indicate that the choice of elimination ordering becomes irrelevant when the extended version of VE is used. To study this phenomenon we implemented a heuristic that selects variables to eliminate at random. We ran the same query ten times using the random heuristic with and without the space extension. Table 6.3 reports the result. The cost displayed is the mean of the 10 runs and an estimated 95% confidence interval around the mean. We see that the minimum cost is not within the confidence interval in either case, which suggests that elimination ordering is still significant in the extended plan space version of VE.

6.5.4 Optimization Cost

The following experiment illustrates the trade-off between plan quality and optimization time of the algorithms. For each view in our second testbed (with $N = 7$), we query all variables in the linear part. In Figure 6.8 we plot the average estimated cost of evaluating the query against the average time required to derive the execution plan. Points closer to the origin are best.

We first note significant gains provided by the algorithms proposed here compared to the CS algorithm. Next we note that nonlinear plans provide gains of around one order of magnitude compared to linear plans. Variable Elimination with the degree heuristic performs better when maximum variable connectivity is low, but still achieves quality plans when considering the extended space. The width and elimination cost heuristics are not affected by maximum variable connectivity indicating that their performance is controlled by average connectivity. Finally we note the lower optimization time, in general, for VE compared to nonlinear CS+.

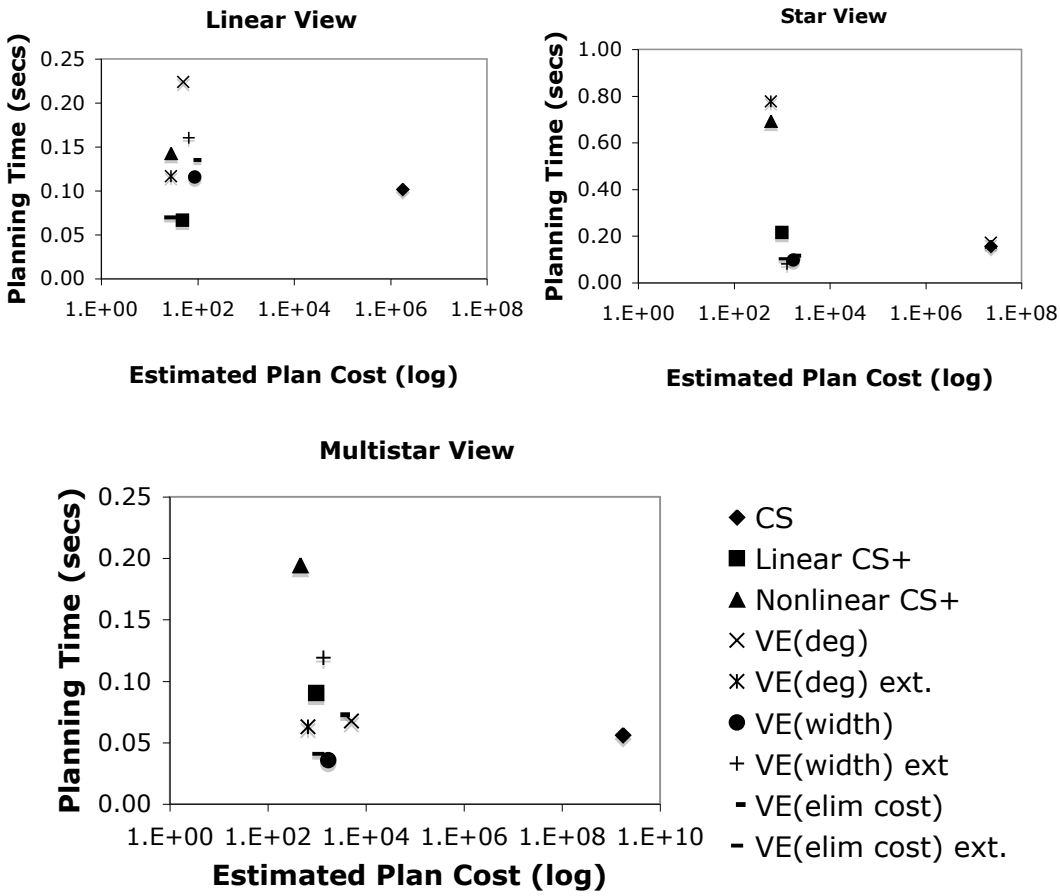


Figure 6.8 Optimization Time Tradeoff Experiment

6.6 Conclusion

In this chapter, we have defined and described the CS+ and VE single MPF query algorithms; we have presented conditions under which evaluation plans can be restricted to the linear class thus avoiding the extra overhead of searching over nonlinear plans; we have characterized and compared the plan spaces explored by each of the algorithms given and shown that the plan space explored by CS+ contains the space explored by VE; we have analyze the optimization time complexity of the algorithms, and also given conditions based on schema characteristics where VE will have significantly lower optimization time complexity than CS+; we have extended VE so that its plan space is closer to the space of CS+ plans without adding much optimization overhead; and finally, we have proposed a cost-based ordering heuristic for Variable Elimination. In the next chapter we present optimization techniques for anticipated workloads of MPF queries.

Chapter 7

Optimizing MPF Query Workloads: View Materialization Strategies for Probabilistic Inference

7.1 Introduction

In the previous Chapter, we presented methods for optimizing the evaluation of MPF queries. These methods extend existing database optimization techniques for aggregate queries to the MPF setting. In particular, we showed how a modification to the algorithm of Chaudhuri and Shim (1994, 1996) for optimizing aggregate queries yields significant gains over evaluation of MPF queries in current systems. We also extended existing probabilistic inference techniques such as Variable Elimination to develop novel optimization techniques for MPF queries. In this chapter, we extend our techniques to address the optimization of expected MPF query workloads.

In particular, we present the MPF-cache Algorithm (Algorithm 3) which extends our methods for optimizing single MPF queries using ideas from Junction Tree and Belief Propagation (Aji and McEliece, 2000). The MPF-cache Algorithm creates a cache of materialized views which can be used to evaluate workload queries directly, that is, without joining any other relations. Extensions to known methods occur along two related directions:

1. We define and incorporate a workload objective to our single query optimization techniques. This allows the search along plan space carried out by our algorithms to try to minimize a cost-based objective derived from an expected MPF query workload

2. We use the Junction Tree property in order to ensure that the caches produced are correct, that is, can be used to answer workload queries correctly. However, as opposed to the classical

formulation of the Junction Tree algorithm, we attempt to minimize a cost-based workload objective.

In this chapter we provide a proof that caches produced by the MPF-cache algorithm are in fact correct. We also outline how the workload objective is derived and used to guide search for plans that minimize the evaluation of anticipated query workloads.

7.2 MPF Query Workload Optimization

MPF queries are stylized aggregate queries that follow a strict syntax. This implies that workloads of MPF queries have a common structure that we want to exploit for efficient evaluation. In this section we describe an algorithm that creates a cache of materialized views which exploits these relationships to optimize the evaluation time of an expected query workload.

We define an expected MPF query workload as a set of basic, restricted-answer, or restricted-domain MPF queries (see Chapter 5), each associated with a probability of being issued by an user. Formally, given an MPF view definition $r = s_1 \overset{*}{\bowtie} \cdots \overset{*}{\bowtie} s_n$, we define a workload $W_r = (Q, P)$ as a set of MPF queries $Q = \{q_1, \ldots, q_n\}$, and an associated probability distribution $P = \{p_1, \ldots, p_n\}$ over Q where $p_i \geq 0$ and $\sum_{i=1}^{n} p_i = 1$.

To ensure correctness of query evaluation with respect to a cache of materialized views, we constrain the cache to satisfy the following invariant:

Definition 7.1 A set of functional relations S satisfies the workload correctness invariant if for at least one functional relation $s \in S$ that includes X_i as a variable, computing an MPF query q on X_i using s yields the same result as evaluating q over joint view r.

MPF Workload Problem We can now define the MPF Workload Problem: given an MPF query workload W_r as described above, build a cache S of materialized views satisfying the invariant in Definition 7.1, such that the following objective is minimized:

$$\mathscr{C}(W_r, S) = \mathbb{E}_P cost(Q(q, S)) + \lambda C(S) \tag{7.1}$$

where $C(S)$ is the cost of materializing cache S, λ is a trade-off parameter that we assume is set by the user, and $cost(Q(q, S))$ is the cost of evaluating query q using cache S. Expectation is taken over probability distribution P.

To build S so that it satisfies the invariant of Definition 7.1, we extend the Junction Tree and Belief Propagation algorithms (Aji and McEliece, 2000). We first modify the view r if it does not define an acyclic schema as in the Junction Tree algorithm. Then each of the resulting relations is updated in a manner similar to Belief Propagation (BP), a message passing algorithm that gathers in each local function information about the joint function. After the message passing algorithm is completed, each relation will now satisfy the correctness invariant in Definition 7.1. See Section 7.4 for a discussion of how the BP and JT algorithms are formulated in the relational setting. However, we cast this as an optimization problem where an objective based on the evaluation of a query workload is minimized.

7.2.1 The MPF-cache Algorithm

In this section we introduce the MPF-cache Algorithm (Algorithm 3) for MPF query workload optimization. MPF-cache first creates a plan p for the MPF query:

```
select AGG(inv) from r;
```

where AGG is a suitable aggregate. While executing plan p, MPF-cache materializes and includes in cache S some intermediate relations that precede Group By nodes. At this point, the resulting relation from plan p contains information about the complete joint functional view r, which has to be propagated to the relations in cache S.

The following semijoin operation extends the product join and is used in the algorithm for the propagation step.

Definition 7.2 Let $U = \mathrm{Var}(t) \cap \mathrm{Var}(s)$, define **update semijoin** as

$$t \ltimes s = t \overset{*}{\bowtie} (\mathrm{GroupBy}_{U,\mathrm{SUM}(s[f])}(s)) \overset{\div}{\bowtie} (\mathrm{GroupBy}_{U,\mathrm{SUM}(t[f])}(t)),$$

where $\stackrel{\div}{\bowtie}$ is defined exactly like product join, but uses the division operation instead of the product operation.

This operation is similar to the classical semi-join operation but uses aggregation instead of projection to reduce operands with respect to common variable subsets.

Algorithm 3 The MPF-cache Optimization Scheme

Output: Set of cached relations that satisfy the correctness invariant

1: Create a no-query-variable single query plan (Algorithm 1 or 2)
2: Select tables that precede a Group By node to cache, say $t_1, \ldots, t_k$
3: **for all** $t_j, j = k, \ldots, 1$ **do**
4: **for all** t_i, such that $j > i$ and $\mathrm{GroupBy}(t_i)$ was used to create t_j **do**
5: compute $t_i \ltimes t_j$
6: **end for**
7: **end for**

Example 7.3 As an example, consider the VE plan of Figure 6.3. Cache S will then contain three tables $t1(sid, pid, wid)$, $t2(cid, tid)$ and $t3(cid, wid)$ corresponding to relations that precede a Group By node. The propagation in steps 3–7 of Algorithm 3 requires the operations $t1 \ltimes t3$ and $t2 \ltimes t3$. As we will see, the materialized views resulting from this algorithm satisfy the correctness invariant, thus evaluating Q1 on $t2$ gives the correct answer.

Theorem 7.4 (Correctness of MPF-cache) The set S of materialized tables in MPF-cache (Algorithm 3) satisfies the correctness invariant of Definition 7.1

Proof of this theorem is given as Section 7.4.

As defined so far, it only considers workloads of basic and restricted-answer queries, but we discuss restricted-domain queries later.

7.2.2 Minimizing the Workload Objective

Having ensured the correctness of the MPF-cache algorithm, we turn to the problem of estimating and minimizing the query workload objective. Given a plan p, the MPF-cache algorithm takes the set of relations that precede a Group By node in p as the complete cache of materialized

views S. However, only a subset $T \subseteq S$ of views need to be materialized to evaluate a particular workload. In example 7.3, if the workload only queries variables sid and cid, it is sufficient to materialize $t1$ and $t2$ only.

Thus, we have two dimensions in which to minimize the workload objective of Equation 7.1: selection of the plan p from which the complete cache S is induced, and choosing a subset $T \subseteq S$ as the final cache. In this section we discuss how we estimate the workload objective, and how it is minimized in the context of the CS+ single-query optimization algorithm with respect to these two minimization dimensions.

The single-query CS+ algorithm performs a bounded search over the space of candidate plans to find the plan that minimizes a cost function based on the evaluation of a single query. We extend the CS+ algorithm by taking the workload objective as the cost function to minimize in the CS+ search algorithm.

Given a candidate plan p and its induced cache S, we find a subset $T \subseteq S$ that minimizes our estimate of the workload objective. We momentarily delay discussion of how to select subset T and concentrate on how to estimate the workload objective.

To calculate construction time $C(S)$, we must take into account the following: 1) the cost of executing plan p, 2) the cost of materializing T, and 3) the cost of the propagation operations in steps 3–7 of the MPF-cache algorithm. 1 and 3 can be readily estimate from statistics kept in the DBMS catalog. We will estimate the cost of materializing T as the cost of writing to disk each of its relations.

Once cache T is materialized, evaluating a query $q \in Q$ requires computing an aggregate on a proper relation $t \in T$. Therefore, we estimate query evaluation cost for query q with respect to table t as

$$cost(Q(q,t)) = \begin{cases} |t| & \text{if } t \text{ is sorted by } X_q \\ |t| \log |t| & \text{otherwise} \end{cases}, \tag{7.2}$$

where X_q are the query variables in query q. Expected query evaluation time is then

$$\mathbb{E}_P cost(Q(q,T)) = \sum_{i=1}^{n} p_i \min_{t \in T_i} cost(Q(q_i,t)),$$

where $T_i \subseteq T$ is the set of relations that may be used to evaluate query q_i.

The problem of selecting cache subset $T \subseteq S$ that minimizes the workload objective is NP-hard [1]. We use a greedy procedure to create approximately optimal assignments: set $T = S$ and consider removing each relation $t_i \in T$ in turn, if the workload objective for, say, subset $T \setminus t_i$ is lower than that for T set $T = T \setminus t_i$ and repeat until the objective can not be improved. Of course, subsets considered must contain relations such that the workload Q can be evaluated.

Example 7.5 Consider a workload where Q1 is posed with probability p_1 and the following query:

```
Q2: select tid, SUM(inv) from invest group by tid;
```

is posed with probability $p_2 = 1 - p_1$. For the plan in Figure 6.3 S is as in Example 7.3. Assume that for this plan tables $t1$ and $t2$ are not sorted and table $t3$ is sorted on wid. The assignment procedure above sets $T = \{t2, t3\}$ since Q1 can be evaluated using $t3$ and thus $t1$ need not be materialized. Expected query evaluation time is then $p_1|t3| + p_2|t2| \log |t2|$.

7.2.3 Restricted Domain MPF Queries

We can add restricted domain queries to workloads and use the MPF-cache scheme for optimization. As the MPF-cache algorithm is defined, relations in the cache contains all the information concerning its variables from the joint view r without any restrictions placed on domain values. Thus, further joins are required to absorb information about the joint function under the constrained domain.

We propose the following protocol to carry this out: 1) apply the selection predicate to any cache table, say t containing the constrained variable, 2) perform semi-join reductions along paths defined by plan p to every other cache table.

Example 7.6 In our running example, if the following query were part of the workload:

```
Q3: select wid, min(inv) from investment where tid=1
    group by wid
```

then, after applying the selection on *t2*, the reduction $t3 \ltimes t2$ is required.

[1] This problem is equivalent to a nonlinear integer optimization program

Theorem 7.7 After carrying out the given protocol, the new MPF-cache tables satisfy the correctness invariant of Definition 7.1.

Proof. This protocol specifies a BP semijoin program over an acyclic schema, so result follows from Theorem 7.13.

As the protocol above is defined, all queries that have the same domain constraint predicate can be answered using a single cache relation. However, in our setting we assume that queries are posed at random with a given probability distribution. Thus, we modify the protocol slightly by performing reductions strictly along the path between a cache relation containing the constrained variable and a cache table containing the query variable, rather than the entire cache. It is easily seen that correctness is retained in this case.

In this case we modify our estimate of expected query evaluation when computing the workload objective. We add to the evaluation cost of Equation 7.2 the cost of performing these reductions. Since these semijoin reduction queries can be expressed as a program of product join queries according to Definition 7.2, their cost can be readily estimated by the query optimizer.

Example 7.8 Modify the workload of Example 7.5 so Q1 and Q2 are posed with probabilities p_1 and p_2 respectively and query Q3 is posed with probability $p_3 = 1 - p_1 - p_2$. Expected evaluation time is now

$$p_1|t3| + p_2|t2|\log|t2| + p_3(|t3| + cost(t3 \ltimes t2)),$$

where $cost(t3 \ltimes t2)$ estimates the cost of performing the reduction.

7.2.4 Variable Elimination and MPF-cache

As in the single-query case, when MPF-cache uses CS+ in step 1 it performs a (bounded) complete search over the space of candidate plans. We can use the relational VE algorithm to heuristically find this plan faster at the cost of sub-optimality. However, the trade-offs between CS+ and VE discussed in Chapter 6 still hold in this case.

The VE algorithm uses a number of heuristics to determine variable elimination order. These heuristics were based on approximations of the cost of executing a plan to evaluate a single query.

Similarly, we define heuristics for the workload case based on approximations of the workload objective. In particular, we make use of the Elimination Cost heuristic (Corrada Bravo and Ramakrishnan, 2006), which approximates the cost of the plan required to eliminate a variable. At each iteration of Variable Elimination (Algorithm 2,) we can evaluate the workload objective of the plan required to eliminate a variable by using an approximation based on a sub-optimal elimination plan. We approximate $\mathrm{optPlan}(\mathrm{rels}(v_j, S))$ as follows: choose a join order at random, and find the access paths that minimize the cost of that joining the relations in that order. Given this suboptimal plan, we can evaluate the workload objective for each variable as described in Section 7.2.2. In each iteration, we select for elimination the variable that minimizes the approximated workload objective.

7.3 Discussion

In this chapter we have introduced the MPF-cache algorithm for optimizing the evaluation of expected workloads of MPF queries. We have proven that it produces caches that satisfy a correctness invariant, which ensures that by answering a workload query with respect to a single cache relation yields the same result as evaluating the query on the original MPF view. We have also described how the MPF-cache algorithm is based on our methods for optimizing the evaluation of single MPF queries, where a workload objective is minimized.

7.4 Proof of MPF-Cache Correctness Theorem

We now prove the correctness of the MPF-cache algorithm by showing that it implements the GDL all-vertex algorithm. We first present the Belief Propagation algorithm to motivate the need for the acyclic schema the Junction Tree algorithm creates. Algorithm 4 is an adaptation of the Belief Propagation (BP) message passing algorithm to the relational setting.

BP selects an order of the relations in the schema according to some heuristic and reduces each functional relation in the order with respect to any table that precedes it with which it shares variables using the product semijoin operation $(\overset{*}{\ltimes})$ defined above. This step propagates values for

variable subsets from one function to another if they have common variables, in a sense, propagating information about those variables to the latter function. Once this first pass is completed, the reverse reductions are done, so that function values are propagated in the reverse direction for all pairs of overlapping functions. This reverse reduction uses the update semijoin operation above so that values propagated in the first pass are not propagated again in the second pass.

Algorithm 4 The Belief Propagation Algorithm

1: Choose a table order $s_1, s_2, \ldots, s_n$
2: **for all** Table s_i in order **do**
3: **for all** Table s_j, such that $i < j$ and s_i and s_j share variables **do**
4: compute $s_j \overset{*}{\ltimes} s_i$
5: **end for**
6: **end for**
7: **for all** Table s_j in *reverse* order **do**
8: **for all** Table s_i, such that $j > i$ and s_i and s_j share variables **do**
9: compute $s_i \ltimes s_j$
10: **end for**
11: **end for**

Belief Propagation defines a semijoin program reduction on the set of base relations which, as opposed to the classical semijoin setting where projection is used, grouping and aggregation is used to 'project' tables. This connection between Belief Propagation and semijoin programs was made by Wu and Wong (2004).

Theorem 7.9 [Pearl (1988)] The updated base relations resulting from BP satisfy the invariant of Definition 7.1.

Figure 7.1 shows the program resulting from BP with the order *Transporters (t), Ctdeals (ct), Warehouses (w), Location (l), Contracts (c)*. For illustration we expand the functional semijoins

1. $ct \overset{*}{\ltimes} t$	5. $l \ltimes c$
2. $w \overset{*}{\ltimes} ct$	6. $w \ltimes l$
3. $l \overset{*}{\ltimes} w$	7. $ct \ltimes w$
4. $c \overset{*}{\ltimes} l$	8. $t \ltimes ct$

Figure 7.1 A BP semijoin program

for the first and last steps of the program:

$$\begin{aligned} ct \overset{*}{\ltimes} t &= ct \overset{*}{\bowtie} (\mathrm{GroupBy}_{tid,\mathrm{SUM}(t.t\_overhead)}(t)) \\ t \ltimes ct &= t \overset{*}{\bowtie} (\mathrm{GroupBy}_{tid,\mathrm{SUM}(ct.ct\_discount)}(ct) \\ &\quad \overset{\div}{\bowtie} (\mathrm{GroupBy}_{tid,\mathrm{SUM}(t.t\_overhead)}(t)). \end{aligned}$$

The Belief Propagation algorithm is not correct for cyclic schemas. Consider an extension to our Decision Support schema that adds the table

Stdeals(supplier_id,transporter_id,st_discount)

which stores agreements between suppliers and transporters. Using the order *Transporters (t), Stdeals (st), Ctdeals (ct), Warehouses (w), Location (l), Contracts (c)* we get the program in Figure 7.2. In step 1, st is reduced with respect to t, and in step 3, c is reduced with respect to st, thus by step 3, c has been reduced with respect to t. However, in step 2, ct is reduced with respect to t, in steps 4,5 and 6 we have reductions from ct to c through w and l. Thus in step 6, c is reduced with respect to t again. Since each step involves the product of the measure attribute of the relations involved, the measure field of c has been incorrectly updated with the measure of t twice.

Acyclic schemas have the running intersection property:

Theorem 7.10 (Maier (1983)) Given schema $S = \{s_1, \ldots, s_n\}$ create undirected graph $G = (V, E)$ where $V = S$ and $(s_i, s_j) \in E$ if $\mathrm{Var}(s_i) \cap \mathrm{Var}(s_j) \neq \emptyset$, that is, the nodes of G are relations and an edge exists between two relations if they share variables. S is an acyclic schema if and only if there exists a tree T that spans G with the property that for vertices s_i, s_j, $\mathrm{Var}(s_i) \cap \mathrm{Var}(s_j)$ is contained in every relation in the path between s_i and s_j.

1. $st \overset{*}{\ltimes} t$	7. $l \ltimes c$
2. $ct \overset{*}{\ltimes} t$	8. $w \ltimes l$
3. $c \overset{*}{\ltimes} st$	9. $ct \ltimes w$
4. $w \overset{*}{\ltimes} ct$	10. $st \ltimes c$
5. $l \overset{*}{\ltimes} w$	11. $t \ltimes ct$
6. $c \overset{*}{\ltimes} l$	12. $t \ltimes st$

Figure 7.2 A BP semijoin program on a cyclic schema

The spanning tree with this property is also called a Junction Tree. Our original example schema has this property, while the schema with the addition of *Stdeals* does not.

Acyclic schemas have a further property:

Theorem 7.11 (Jensen (2001)) Given schema $S = \{s_1, \ldots, s_n\}$ create undirected graph $G = (V, E)$ where $V = \bigcup_i \mathrm{Var}(s_i)$ and $(v_i, v_j) \in E$ if there exists a relation s_k such that $v_i, v_j \in \mathrm{Var}(s_k)$, that is, the nodes of G are the variables appearing in the schema and there is an edge between two variables if they co-occur in a relation. S is an acyclic schema if and only if G is chordal.

A chordal graph is one where every cycle of length greater than 3 has a chord, that is, an edge between two non-consecutive nodes in the cycle. Figure 7.3 has the variable graph for our original acyclic schema. The addition of *Stdeals* would add an edge between *sid* and *tid* which creates a cycle of length 5 that has no chord. We refer the reader to Cowell et al. (1999) and Jensen (2001) for a more extended discussion of chordal graphs and junction trees in the context of probabilistic inference, and to Wu and Wong (2004) for further discussion on the links between Junction Trees, Belief Propagation and acyclic database schemas.

The Junction Tree algorithm creates an acyclic schema by transforming the variable graph of a cyclic schema into a chordal graph. The acyclic schema is then induced from this resulting chordal graph. Algorithm 5 lists the Junction Tree algorithm. Step 2 modifies the variable graph of the input schema to create a chordal graph using triangulization [2] which is listed as Algorithm 6. It

[2] A chordal graph is also said to be triangulated

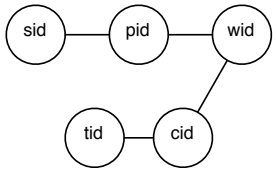


Figure 7.3 Variable graph for acyclic schema

adds edges to the graph by choosing a vertex, connecting any of its disconnected neighbors and then removing it from the graph. Figure 7.4 shows a chordal graph resulting from triangulization for our example cyclic schema using the vertex order *tid,sid* and added edges drawn dotted. Figure 7.5 shows the new schema and the Junction Tree resulting from that chordal graph. The final step of the algorithm populates the tables of the new schema by assigning relation s_i of the original schema to a relation s_j of the new schema such that $\mathrm{Var}(s_i) \subseteq \mathrm{Var}(s_j)$, and then computing the product join of tables assigned to each relation of the new schema.

Algorithm 5 The Junction Tree Algorithm

1: Construct variable graph G from schema S
2: Triangulate G to create new graph G'
3: Create new schema S' where each maximal clique in G' is a relation
4: Assign relations from schema S to relations in S' that contain all of its variables
5: Create the new relation by product joining all S tables assigned to each relation in S'

The size of the resulting schema, and thus the complexity of Belief Propagation on the resulting schema, is determined by the size of the cliques in the new graph. This in turn is determined by the order in which vertices are chosen during triangulization. The size of the largest clique in the resulting graph is called the induced width of the new graph.

Theorem 7.12 (Yannakakis (1981)) Finding the chordal graph with minimum induced width is NP-complete in the number of variables.

The equivalence between the Triangulization and the Variable Elimination algorithms is clear. Choosing a vertex and connecting any unconnected neighbors in triangulization is equivalent to

Algorithm 6 The Triangulization Procedure

Input: Graph $G = (V, E)$

Output: Chordal graph $G' = (V', E')$

1: Set $G' = (V', E')$ where $V' = V$ and $E' = E$
2: **while** $V \neq \emptyset$ **do**
3: select vertex $v \in V$ from a non-chordal cycle
4: for every pair (v, u_1) and $(v, u_2) \in E$, add (u_1, u_2) to E and E'
5: remove v from V
6: **end while**

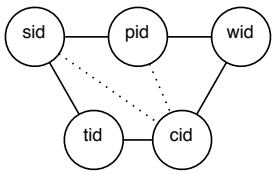


Figure 7.4 A chordal graph for the cyclic schema

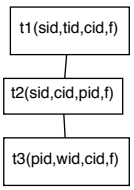


Figure 7.5 The resulting Junction Tree

selecting a variable v and joining the tables where it appears in Variable Elimination. The clique resulting from the added edges will be a relation in the new schema, caching the result of this join in Variable Elimination creates the relation in the new schema. Removing the vertex from the graph in triangulization yields a clique of its neighbors equivalent to the relation resulting from marginalizing, or, eliminating the chosen variable.

Theorem 7.13 Denote the set of cached tables in MPF-cache as $T = \{t_i : i = 1, \ldots, k\}$. Then the following hold:

1. T is the schema result of triangulating using the variable order given by the VE plan of line 1,
2. T is an acyclic schema, and
3. MPF-cache performs a BP semijoin program over T

Proof. (1) follows from the equivalence of triangulation and variable elimination and the fact that the relations that precede Group By nodes give the relations from triangulation. (2) follows from (1) since triangulation results in an acyclic schema. For (3) we first note that MPF-cache implements directly the backward pass of lines 7 through 10, and that by the definition of $\overset{*}{\ltimes}$ we have that MPF-cache also performs the forward pass when it executes the given VE plan. Proof. (Theorem 7.4). Follows directly from Theorems 7.13 and 7.9.

Part IV

Prospects and Perspectives

Chapter 8

Distance-Based Regression by Regularized Kernel Estimation

In this chapter, we propose an extension of RKE to a semi-supervised setting where real-valued responses are given for some of the objects with the goal of directly estimating a regression function from noisy, inconsistent and incomplete distance data. We show how to estimate both the kernel and regression functions jointly by minimizing a trade-off of fidelity to the distance data and a regression objective for the given responses as a linear semidefinite problem (SDP) where a set of regularization parameters determines the distance vs. response fidelity trade-off. Properly selecting values for the tuning parameters is of vital importance in this case. To that end, we present a tuning method based on an approximation to a cross-validation criterion for choosing values of the tuning parameters. We derive this approximation using perturbation arguments based on recent results on the sensitivity of linear SDPs to data perturbations.

For joint RKE and regression we have to make the distinction between semi-supervised and fully supervised settings. In the latter, distances between a set of objects are given along with labels for all the objects with the goal of learning a function that predicts the responses of unseen objects. Within the semi-supervised setting we distinguish the *transductive* setting where the set of objects for which the set of distances is given encompasses the entire set of objects of interest and thus there are no unseen objects. However, responses are given for only a subset of these objects and the goal is to predict the responses of these unlabeled objects. In the fully semi-supervised setting, there are unseen objects as in the inductive setting, however, responses are not given for the entire set of seen objects. The goal in this case is again to learn a function to predict the responses for unlabeled objects, both seen and unseen. In this chapter, we will address the *transductive* setting in particular in the adaptive tuning method, although the general estimation

methodology is applicable to the fully semi-supervised setting. For this chapter we will have some distances between all objects of interest, from which we can learn an embedding kernel function, and we have labels for a subset of these objects. The goal is to learn a regression function spanned by the embedding kernel to predict the responses of the unlabeled objects. We will address the remaining cases for both classification and regression in future work.

The chapter is structured as follows: we first reintroduce the RKE setting of Lu et al. (2005) and its extension to the transductive regression setting in Section 8.1; we continue by stating recent results on the sensitivity of linear SDPs along with a leave-one-out lemma for linear SDPs which we need for our tuning method in Section 8.2; we present the tuning method for the transductive regression setting in Section 8.3.

A note on notation: $\mathcal{S}^N$ is the space of N-by-N symmetric matrices; x_i is the ith entry of vector x and X_{ij} is the ijth entry of matrix X; x^T (X^T) is the vector (matrix) transpose; e is the unit vector of appropriate length for the context; e_i is the ith standard basis vector of appropriate length for the context such that $x_i = e_i^T x$; $\mathrm{tr}(AB) = \sum_{i,j=1}^N A_{ij}B_{ij}$ denotes the standard inner product in $\mathcal{S}^N$. Given matrices $A^1, \ldots, A^m \in \mathcal{S}^N$, we define the linear operators

$$\mathcal{A}(X) = \begin{bmatrix} \mathrm{tr}(A^1 X) \\ \vdots \\ \mathrm{tr}(A^m X) \end{bmatrix}, \tag{8.1}$$

and $\mathcal{A}^T(w) = \sum_{j=1}^m w_j A^j$, $w \in \mathbb{R}^m$. The non-negative orthant is denoted $\mathbb{R}^N_+$ and the cone of symmetric positive definite matrices $X \succeq 0$ as $X \in \mathcal{S}^N_+$.

8.1 Regularized Kernel Estimation for Regression

RKE estimates a symmetric positive semidefinite kernel matrix K which induces a real squared distance admitting of an inner product. K is the solution to an optimization problem with semidefinite constraints that trades-off fit to the observed dissimilarity data and a penalty of the form $\lambda_{rke}\mathrm{tr}(K)$ on the complexity of K, where λ_{rke} is a non-negative regularization parameter.

8.1.1 The RKE Problem

Given a set of N objects, assume dissimilarity information is given for a subset Ω of size m of the $\binom{N}{2}$ possible pairs of objects. Denote the dissimilarity between objects i and j as $d_{ij} \in \Omega$. We make the requirement that Ω satisfies a connectivity constraint: the undirected graph defined by Ω consisting of objects as nodes and including an edge between nodes i and j if $d_{ij} \in \Omega$ is connected.

Formally, RKE estimates a positive semidefinite kernel matrix $K \in \mathcal{S}^N$ such that the fitted squared distance between objects induced by K, $\hat{d}^2_{ij}(K) = K(i,i)+K(j,j)-2K(i,j) := \mathrm{tr}B^{ij}K$, are close to the square of the observed distances $d^2_{ij} \in \Omega$:

$$\min_{K \in \mathcal{S}^N} \quad \sum_{d_{ij} \in \Omega} |d^2_{ij} - \mathrm{tr}(B^{ij}K)| + \lambda_{rke}\mathrm{tr}(K) \tag{8.2a}$$

$$\text{s.t.} \quad K \succeq 0. \tag{8.2b}$$

Since the trace of K may be seen as a proxy for its rank, RKE is regularized by penalizing high dimensionality of the space spanned by K. The parameter $\lambda_{rke} \geq 0$ is a regularization parameter that trades-off fit of the dissimilarity data, as given by absolute deviation, and the trace penalty on the complexity of K. The tuning problem in the unsupervised case is finding a value of the regularization parameter λ_{rke} to minimize some generalization criterion with respect to the fitted distances.

8.1.2 Regularized Kernel Estimation for Regression

We base our joint Regression-RKE method on the setting of Lanckriet et al. (2004a), which gives a general result on optimizing performance measures derived from the dual of various SVM formulations over a convex subset of $\mathcal{S}^N_+$ can be cast as linear semidefinite programs. In our case, this set will be $\mathcal{S}^N$ itself but we will trade-off poor fit to the observed distances and minimizing the error of the regression function on the labeled objects.

In the classical nonparametric regression setting, we assume covariates $x_i \in \mathbb{R}^p$ along with outcomes $y_i \in \mathbb{R}$, $i = 1, \ldots, n$ are observed. A Reproducing Kernel Hilbert Space of the form

$\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$ where functions $\phi_1, \ldots, \phi_M$ span $\mathcal{H}_0$ and $\mathcal{H}_0$ is orthogonal to $\mathcal{H}_1$ is chosen to define a set of functions $f(\cdot) = f_0 + f_1$, $f_0 \in \mathcal{H}_0$, $f_1 \in \mathcal{H}_1$ and $\langle f_0, f_1 \rangle = 0$. The goal is to find the function $\mathcal{H}$ that minimizes the regularized empirical risk variational problem

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_i)^2 + \lambda_{reg} \|P_1 f\|^2_{\mathcal{H}_1} \tag{8.3}$$

where $\lambda_{reg} \geq 0$ is a regularization parameter that trades off fit to the observed outcomes and the norm of g in Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$.

covariates $x_i \in \mathbb{R}^p$ are given. For a given kernel function $k(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ and associated Reproducing Kernel Hilbert Space $\mathcal{H}$, and parametric functions By the Kimeldorf and Wahba Theorem (Kimeldorf and Wahba, 1971), the minimizer of (8.3) has a finite expansion in terms of the representers of training points x_i so that $\hat{f}(\cdot) = \sum_{i=1}^{n} c_i k(x_i, \cdot)$, for coefficient vector c to be estimated. Therefore, letting K be the Gram matrix resulting from evaluating $k(\cdot, \cdot)$ at every pair of training points, we get that vector $f = Kc + \gamma e$ satisfies $f_i = f(x_i)$. Equation(8.3) then becomes

$$\min_{c \in \mathbb{R}^n, \gamma \in \mathbb{R}} \frac{1}{2}(y - (Kc + \gamma e))^T (y - (Kc + \gamma e)) + \frac{\lambda_{reg}}{2} c^T K c. \tag{8.4}$$

Since our goal in joint regression-RKE is to estimate a kernel matrix K from both the observed distances and labels, we will show that the optimum value of Problem (8.4) is a convex function of K. For this purpose we will make use of Lagrange duality. First, we will rewrite Equation (8.4) as the following equivalent equality constrained optimization problem:

$$\min_{c, r \in \mathbb{R}^n, \gamma \in \mathbb{R}} \quad \frac{1}{2\lambda_{reg}} r^T r + \frac{1}{2} c^T K c \tag{8.5a}$$

$$\text{s.t.} \quad r = y - Kc - \gamma e. \tag{8.5b}$$

The Lagrangian for this problem is

$$L_{reg}(c, r, \gamma, \alpha) = \frac{1}{2\lambda_{reg}} r^T r + \frac{1}{2} c^T K c + \alpha^T (r - y + Kc + \gamma e). \tag{8.6}$$

Setting the gradient of L with respect to r, c and γ to zero yields that at a saddle point the following conditions must hold:

$$\frac{1}{\lambda_{reg}} r + \alpha = 0 \tag{8.7a}$$

$$c + \alpha = 0 \tag{8.7b}$$

$$e^T \alpha = 0 \tag{8.7c}$$

This yields the following Lagrange dual problem:

$$\max_{\alpha \in \mathbb{R}^n} \quad -\frac{1}{2}\alpha^T (K + \lambda_{reg} I)\alpha - y^T \alpha \tag{8.8a}$$

$$\text{s.t.} \quad e^T \alpha = 0. \tag{8.8b}$$

Seen as a function of K, Equation (8.8) is convex as it is the point-wise maximum of affine functions. This is an instantiation of the generalized performance measure of Lanckriet et al. (2004a).

Now consider the transductive setting discussed above where we assume that no covariates are given but rather some pairwise distances between N objects are observed along with responses $y_i \in \mathbb{R}$, $i = 1, \ldots, n$, $n < N$ for a subset of the objects. We want to estimate an N-by$N$ kernel matrix K from the observed distances and responses. We partition K as

$$K = \begin{bmatrix} K_{oo} & K_{ou} \\ K_{ou}^T & K_{uu} \end{bmatrix}, \tag{8.9}$$

where the n-by-n submatrix K_{oo} corresponds to the kernel matrix for the n objects with observed responses. We will minimize the optimal value of Problem (8.8) as a function of K over $\mathcal{S}_+^N$ and trade-off poor fit of the distance data with parameter λ_{dist} regularizing the solution with a penalty on the trace of K:

$$\min_{K \in \mathcal{S}^N} \quad g_{\lambda_{reg}}(K) + \lambda_{dist} \sum_{ij \in \Omega} |d_{ij}^2 - \text{tr}(B^{ij}K)| + \lambda_{rke}\text{tr}(K) \tag{8.10a}$$

$$\text{s.t.} \quad K \succeq 0, \tag{8.10b}$$

where $g_{\lambda reg}(K)$ is the optimal value of Problem (8.8) given K and parametrized by λ_{reg}. The regularization parameters λ_{dist} and λ_{rke} can be seen as Lagrange multipliers of the equality constrained optimization problem

$$\min_{K \in \mathcal{S}^N} \quad g_{\lambda_{reg}}(K) \tag{8.11a}$$

$$\text{s.t.} \quad |d_{ij}^2 - \text{tr}(B^{ij}K)| = 0, \ \forall d_{ij} \in \Omega \tag{8.11b}$$

$$\text{tr}(K) \leq \tau \tag{8.11c}$$

$$K \succeq 0 \tag{8.11d}$$

which minimizes regression regularized loss $g_{\lambda_{reg}}(K)$ over the set of symmetric positive semidefinite matrices that match the observed distances and have trace bounded by constant τ.

Using duality we can write $g_{\lambda_{reg}}(K)$ as

$$g_{\lambda_{reg}}(K) = \min_{\nu \in \mathbb{R}} \max_{\alpha \in \mathbb{R}^N} -\frac{1}{2}\alpha^T(K_{oo} + \lambda_{reg}I)\alpha - y^T\alpha + \nu e^T\alpha, \tag{8.12}$$

where the optimal value of the inner maximization problem has a closed form solution in terms of K which we get by setting the gradient with respect to α equal to 0:

$$\hat{\alpha} = (K_{oo} + \lambda_{reg}I)^{-1}(\nu e - y), \tag{8.13}$$

which yields $g_{\lambda_{reg}}(K) = \frac{1}{2}(\nu e - y)^T(K_{oo} + \lambda_{reg}I)^{-1}(\nu e - y)$, which includes new variable $\nu \in \mathbb{R}$.

Joint RKE-Regression Problem: The joint Regression-RKE optimization problem is:

$$\min_{K\in\mathcal{S}^N,\nu\in\mathbb{R}} \quad \frac{1}{2}(\nu e - y)^T(K_{oo} + \lambda_{reg}I)^{-1}(\nu e - y) \tag{8.14a}$$

$$+ \lambda_{dist}\sum_{ij\in\Omega}|d_{ij} - \mathrm{tr}(B^{ij}K)| + \lambda_{rke}\mathrm{tr}(K) \tag{8.14b}$$

$$\text{s.t.} \quad K \succeq 0. \tag{8.14c}$$

In Section 8.3 we show how to cast this problem as a linear SDP. The tuning problem in this case is finding values for $\lambda_{reg}, \lambda_{dist}$ and λ_{rke} that minimize some generalization criterion. We present a method for tuning in Section 8.3. Finally, note that given solutions $\hat{K}$ and $\hat{\nu}$, we can recover $\hat{c}$ and $\hat{\gamma}$ to define $\hat{f}$ as

$$\hat{c} = -(\hat{K} + \lambda_{reg}I)^{-1}(\hat{\nu}e - y) \tag{8.15a}$$

$$\hat{\gamma} = \hat{\nu} \tag{8.15b}$$

8.2 Tuning by Sensitivity Arguments for Linear SDPs

The goal of this section is present general results on the sensitivity of linear SDPs on which we base our tuning methods. Our tuning method defines a criterion that approximates leave-one-out error along the lines of GCV/GACV (Wahba, 1990). This approximation is based on approximating from the solution of a single linear SDP P, a performance criterion based on the solution of a number of linear SDPs where a single constraint is removed in each one. First, we will specify the standard form we will use for the primal and dual linear SDP and restate a recent sensitivity result for linear SDPs. Next, we present a *leave-one-out* lemma for linear SDPs. Finally, using this lemma and the sensitivity result we give a first-order approximation of the solution of the so-called *leave-one-out* problem.

8.2.1 SDPs in Standard Form

We will use the following standard form for the linear semidefinite problem:

$$\min_{X \in \mathcal{S}^N} \quad \operatorname{tr}(CX) \tag{8.16a}$$

$$\text{s.t.} \quad \mathcal{A}(X) = b \tag{8.16b}$$

$$X \succeq 0, \tag{8.16c}$$

where $C \in \mathcal{S}^N$, $b \in \mathbb{R}^m$ and $A^j \in \mathcal{S}^N$, $j = 1, \ldots, m$. The Lagrangian is

$$L(X, w, S) = \operatorname{tr}(CX) + w^T(b - A(X)) - \operatorname{tr}(SX), \tag{8.17}$$

where $w \in \mathbb{R}^m$ and $S \in \mathcal{S}^n_+$ are Lagrange multipliers. The resulting Lagrange dual is

$$\max_{w \in \mathbb{R}^m, S \in \mathcal{S}^N} \quad b^T w \tag{8.18a}$$

$$\text{s.t.} \quad \mathcal{A}^T(w) + S = C \tag{8.18b}$$

$$S \succeq 0, \tag{8.18c}$$

If there exists a matrix $X \succ 0$ that is feasible for Problem (8.16), we say that Problem (8.16) satisfies Slater's condition, and conversely for S and Problem (8.18). By weak duality, the optimum value of Problem (8.18) is a lower bound of the optimum value of Problem (8.16). Strong duality holds, that is, the optimum values of Problem (8.16) and Problem (8.18) coincide when either Problem (8.16) or Problem (8.18) satisfy Slater's condition. On the other hand, if both problems are feasible, then optimal solutions $\bar{X}$ and $(\bar{w}, \bar{S})$ exist and satisfy the complementarity condition $\bar{X}\bar{S} = 0$. Conversely, if X and (w, S) are feasible, and $XS = 0$ then X and (w, S) are optimal solutions.

Finally, we make the observation that if C is a diagonal matrix and operator $\mathcal{A}$ consists of only diagonal matrices, then X and S can be restricted to their diagonals, which make their semidefinite constraint equivalent to a non-negativity constraint on the diagonals. In this case, X and S can be represented by vectors x and s in $\mathbb{R}^N_+$ and $\mathcal{A}$ represented as matrix $A \in \mathbb{R}^{m \times N}$, in which case Problems (8.16) and (8.18) become linear problems. We can also allow free variables $x \in \mathbb{R}^n$

in standard form linear SDPs and implicitly assume that the problem will be transformed to an equivalent problem with non-negative variables $x^+ \in \mathbb{R}^n_+$ and $x^- \in \mathbb{R}^n$ which satisfy $x = x^+ - x^-$.

8.2.2 Perturbed Linear SDPs

In this section we provide an approximation of the solution of SDPs where the right-hand-side vector b is perturbed by vector u. The perturbed primal problem is

$$\begin{aligned} \min_{X \in \mathcal{S}^N} \quad & \operatorname{tr}(CX) && \text{(8.19a)} \\ \text{s.t.} \quad & \mathcal{A}(X) = b + u && \text{(8.19b)} \\ & X \succeq 0, && \text{(8.19c)} \end{aligned}$$

with Lagrangian (now including u as a variable) is

$$L(X, w, S, u) = \operatorname{tr}(CX) + w^T(b + u - A(X)) - \operatorname{tr}(SX), \tag{8.20}$$

The resulting Lagrange dual is

$$\begin{aligned} \max_{w \in \mathbb{R}^m, S \in \mathcal{S}^N} \quad & (b + u)^T w && \text{(8.21a)} \\ \text{s.t.} \quad & \mathcal{A}^T(w) + S = C && \text{(8.21b)} \\ & S \succeq 0, && \text{(8.21c)} \end{aligned}$$

Denote the solution of Problem (8.19) as $X(u)$ such that $\bar{X} = X(0)$, and $X^* = X(\Delta b)$ for some perturbation vector Δb. Denote also their associated dual solutions to Problem (8.21) as $(\bar{w}, \bar{S})$ and (w^*, S^*) respectively.

Our goal is to approximate X^* using the solutions $\bar{X}$ and $(\bar{w}, \bar{S})$ to the unperturbed primal and dual problems. To that end, we turn to recent sensitivity results that for semidefinite programs (Bonnans and Shapiro, 2000; Freund and Jarre, 2004; Sturm and Zhang, 2001; Yildirim and Todd, 2001). In particular we will make use of the perturbation results of Freund and Jarre (2004) on the differentiability of the optimal solution function of linear SDPs which we restate here:

Theorem 8.1 (Freund and Jarre (2004)) Let a linear operator $\mathcal{A} : \mathcal{S}^N \to \mathbb{R}^m$, a vector $b \in \mathbb{R}^m$ and a matrix $C \in \mathcal{S}^N$ be the data of a pair (8.16) and (8.18) of primal and dual linear semidefinite programs. Assume that programs (8.16) and (8.18) satisfy Slater's condition, and that $\bar{X} \in \mathcal{S}^N$, and $(\bar{w}, \bar{S}) \in \mathbb{R}^m \times \mathcal{S}^N$ are unique and strictly complementary solutions of (8.16) and (8.18), that is

$$\mathcal{A}(\bar{X}) = b,\ \mathcal{A}^T(\bar{w}) + \bar{S} = C,\ \bar{X}\bar{S} = 0,\ \bar{X} \succeq 0,\ \bar{S} \succeq 0, \bar{X} + \bar{S} \succ 0. \tag{8.22}$$

If the data is changed by sufficiently small perturbations $\Delta\mathcal{A}$, Δb, ΔC, then the optimal solutions of the perturbed semidefinite programs are differentiable functions of the perturbations. Furthermore, the derivatives

$$\dot{X} := D_{\mathcal{A},b,C}\bar{X}\left[\Delta A, \Delta b, \Delta C\right],\ \dot{w} := D_{\mathcal{A},b,C}\bar{w}\left[\Delta A, \Delta b, \Delta C\right],\ \dot{S} := D_{\mathcal{A},b,C}\bar{S}\left[\Delta A, \Delta b, \Delta C\right], \tag{8.23}$$

of the solution X, w and S at $\bar{X}$, $\bar{w}$ and $\bar{S}$ satisfy

$$\mathcal{A}(\dot{X}) = \Delta b - \Delta\mathcal{A}(\bar{X}), \tag{8.24a}$$

$$\mathcal{A}^T(\dot{w}) + \dot{S} = \Delta C - \Delta\mathcal{A}^T(\bar{w}), \tag{8.24b}$$

$$\bar{S}\dot{X} + \dot{S}\bar{X} = 0. \tag{8.24c}$$

Given these derivatives a first-order approximation of $X^* - \bar{X} \approx \dot{X}$ is obtained, where solving system 8.24 is required. However, the left-hand-side of this system of equations is the same as the predictor Newton step in many interior point implementations (Borchers, 1999; Toh et al., 1999). With that in mind, as described by Yildirim and Todd (2001), the Cholesky Factorization of the Schur Complement Matrix of the predictor step of the last iterate can be used to solve the system above (taking the next to last iterate as $\bar{X}$ and $(\bar{w}, \bar{S})$).

In particular, for perturbations of only the right-hand side vector b of the form $\Delta b = te_j$ we can solve the system as follows. First, using Equation (8.24b), set $\dot{S} = -\mathcal{A}^T(\dot{w})$ and substitute into Equation (8.24c). This yields $\dot{X} = \bar{S}^{-1}\mathcal{A}^T(\dot{w})\bar{X}$. Substituting into (8.24a) we get the linear system of equations

$$\mathcal{A}(\bar{S}^{-1}\mathcal{A}^T(\dot{w})\bar{X}) = te_j \tag{8.25}$$

which can be rewritten as

$$O\dot{w} = te_j \tag{8.26}$$

where O is the Schur matrix from, for example, the H..K..M.. predictor step (Helmberg et al., 1996): $O_{ij} = \mathrm{tr}\bar{S}^{-1}A^i\bar{X}A^j$. Since O is positive definite (Helmberg et al., 1996) we get $\dot{w} = tO^{-1}e_j$, and $\dot{X} = t\sum_{i=1}^m O_{ij}^{-1}\bar{S}^{-1}A^i\bar{X}$. Therefore, for perturbations of the form $\Delta b = te_j$ we will use the first-order approximation:

$$X^* - \bar{X} \approx t\sum_{i=1}^m O_{ij}^{-1}\bar{S}^{-1}A^i\bar{X}. \tag{8.27}$$

8.2.3 Leave-one-out Lemma

Let $\mathcal{A}$, b and C be data defining the primal SDP problem (8.16). Define the jth primal parametric SDP $P(u_j)$ as:

$$\begin{aligned}
\min_{X\in\mathcal{S}^n} \quad & \mathrm{tr}(CX) && \text{(8.28a)}\\
\text{s.t.} \quad & \mathcal{A}^{[-j]}(X) = b^{[-j]} && \text{(8.28b)}\\
& \mathrm{tr}(A^jX) = b_j + u_j && \text{(8.28c)}\\
& X \succeq 0 && \text{(8.28d)}
\end{aligned}$$

where $\mathcal{A}^{[-j]}$ is the linear operator $\mathcal{A}$ with matrix A^j removed, and $b^{[-j]}$ is vector b with component j removed. Also, define the jth primal leave-one-out SDP $\tilde{P}_j$ as:

$$\min_{X \in \mathcal{S}^n} \quad \mathrm{tr}(CX) \tag{8.29a}$$

$$\text{s.t.} \quad \mathcal{A}^{[-j]}(X) = b^{[-j]} \tag{8.29b}$$

$$X \succeq 0 \tag{8.29c}$$

Lemma 8.2 (SPD leave-one-out) Let $X^{[-j]}$ be an optimal solution of the jth leave-one-out SDP $\tilde{P}_j$ and let $b_j^* = \mathrm{tr}(A^j X^{[-j]})$. $X^{[-j]}$ is an optimal solution of $P(b_j^* - b_j)$.

Proof. Since $X^{[-j]}$ is feasible for $\tilde{P}_j$, we have by definition that $X^{[-j]}$ is feasible for $P(b_j^* - b_j)$. Let $\bar{X}$ be a feasible solution for $P(b_j^* - b_j)$, then we have that $\bar{X}$ is feasible for $\tilde{P}$. Since $X^{[-j]}$ is an optimal solution for $\tilde{P}_j$ we must have $\mathrm{tr}(CX^{[-j]}) \leq \mathrm{tr}(C\bar{X})$ for every feasible solution $\bar{X}$ of $P(b_j^* - b_j)$. Therefore, $X^{[-j]}$ is an optimal solution of $P(b_j^* - b_j)$.

Using this lemma, we have that the solution of the jth leave-one-out SDP is optimal for the perturbed primal Problem (8.19) by setting $u = (b^* j - b_j)e_j = \Delta b_j^* e_j$. Therefore, using the approximation of Equation (8.27) we have

$$X^{[-j]} - \bar{X} \approx \Delta b_j^* \sum_{i=1}^{m} O_{ij}^{-1} \bar{S}^{-1} A^i \bar{X}. \tag{8.30}$$

8.2.4 The Tuning Problem

Assume that problem data for the primal linear SDP (8.16) is parametrized by a vector λ. We want to find the vector λ that minimizes a cross-validation criterion based on the leave-one-out problem (8.29) for $n \leq m$ constraints:

$$V_0(\lambda) = \frac{1}{n} \sum_{i=1}^{n} g_i(b_i, f_i(X_\lambda^{[-i]})) \tag{8.31a}$$

$$= \frac{1}{n} \sum_{i=1}^{n} g_i(b_i, f_i(X_\lambda)) + \frac{1}{n} \sum_{i=1}^{m} \left[g_i(b_i, f_i(X_\lambda^{[-i]})) - g_i(b_i, f_i(X_\lambda)) \right], \tag{8.31b}$$

$$= OBS(\lambda) + D(\lambda), \tag{8.31c}$$

where X_λ is the solution to the linear SDP parametrized by λ and $X_\lambda^{[-i]}$ is the same for the ith leave-one-out problem. Here f_i is a prediction function and $g(b_i, f_i(X))$ is a "loss" that penalizes the prediction function $f_i(X)$ with respect to right-hand-side vector entry b_i. The notation $OBS(\lambda)$ and $D(\lambda)$ stresses that this approximation adds a divergence term to the observed loss for solution X_λ based on the sensitivity of the solution to perturbations in the right-hand-side vector arising from the leave-one-out criterion.

Assume for now that g and f_i for all i are differentiable functions. We use a first-order approximation of $g(b_i, \cdot)$ as a function of X and the leave-one-out approximation (8.30) to get

$$g(b_i, f_i(X_\lambda^{[-i]})) - g(b_i, f_i(X_\lambda)) \approx \frac{\partial g(f_i(X_\lambda))}{\partial f} \mathrm{tr}(D_X f_i(X_\lambda)(X_\lambda^{[-i]} - X_\lambda)) \tag{8.32a}$$

$$\approx \Delta b_i^* \frac{\partial g(f_i(X_\lambda))}{\partial f} \sum_{j=1}^{m} O(\lambda)_{ij}^{-1} \mathrm{tr}(F_\lambda^i S_\lambda^{-1} A^j X_\lambda), \tag{8.32b}$$

where $F_\lambda^i = D_X f_i(X_\lambda)$ and S_λ and $O(\lambda)$ are the corresponding dual solution and Schur matrix. We will use the approximation $\Delta b_i^* \approx f_i(X_\lambda) - b_i$, the motivation for which will become apparent once we look at particular applications of this general setting. Thus we have the following first-order approximation of $D(\lambda)$:

$$D(\lambda) \approx \frac{1}{n} \sum_{i=1}^{n} (b_i - f_i(X_\lambda)) \frac{\partial g(f_i(X_\lambda))}{\partial f} \sum_{j=1}^{m} O(\lambda)_{ij}^{-1} \mathrm{tr}(F_\lambda^i S_\lambda^{-1} A^j X_\lambda). \tag{8.33}$$

We will next do some further approximations for computational efficiency concerns. The first approximates $\sum_{j=1}^m O_{ij}^{-1} S_\lambda^{-1} A^j X_\lambda$ by $\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m O_{ij}^{-1} S_\lambda^{-1} A^j X_\lambda$ for all $i = 1, \ldots, n$; $\sum_{i=1}^n O_{ij}^{-1}$ by $\frac{1}{m} \sum_{j=1}^m \sum_{i=1}^n O_{ij}^{-1}$ for all $j = 1, \ldots, m$; and $\sum_{i=1}^n \sum_{j=1}^m O_{ij}^{-1}$ by $\frac{n}{m} e^T O^{-1} e$.

The SDP-GACV tuning criterion is then

$$V(\lambda) = \frac{1}{n} \sum_{i=1}^{n} g(b_i, f_i(X(\lambda))) + \hat{D}(\lambda), \tag{8.34}$$

where

$$\hat{D}(\lambda) = \frac{\sigma_\lambda}{n} \mathrm{tr} S_\lambda^{-1} \mathcal{A}^T(e) X_\lambda \left[\sum_{i=1}^{n} (b_i - f_i(X_\lambda)) \frac{\partial g(f_i(X_\lambda))}{\partial f} F_\lambda^i \right], \tag{8.35}$$

with $\sigma_\lambda = \frac{1}{m^2}\|R^{-1}e\|_2^2$ and R the (triangular) Cholesky factor of O.

8.3 Tuning RKE for Regression

To apply the tuning by sensitivity arguments presented in the previous Section we have to specify the join RKE-Regression Problem (8.10) as a linear SDP in standard form. We begin by writing it as the following equivalent problem:

$$\min_{\substack{K\in\mathcal{S}^N,\nu\in\mathbb{R}\\ t\in\mathbb{R},p,q\in\mathbb{R}^m}} \quad t + \lambda_{dist}e^T(p+q) + \lambda_{rke}\mathrm{tr}(K) \tag{8.36a}$$

$$\text{s.t.} \quad \begin{bmatrix} K_{oo} & \nu e \\ \nu e^T & t \end{bmatrix} \succeq \begin{bmatrix} -\lambda_{reg}I & y \\ y^T & 0 \end{bmatrix} \tag{8.36b}$$

$$\mathcal{B}(K) + p - q = d \tag{8.36c}$$

$$X \succeq 0, p \geq 0, q \geq 0. \tag{8.36d}$$

Non-negative variables p and q are used to represent the piece-wise linear absolute value term in the objective of Problem (8.10). By the Schur Complement Lemma and the fact that $K_{oo} + \lambda_{reg}I \succeq 0$, the linear matrix inequality (8.36b) implies $t \geq (\nu e - y)(K_{oo} + \lambda_{reg}I)^{-1}(\nu e - y) = g_{\lambda_{reg}}(K)$. To convert this linear matrix inequality to the equality constraint required for standard form we introduce a positive semidefinite slack variable $Z = \begin{bmatrix} Z_{11} & z \\ z^T & \zeta \end{bmatrix} \in \mathcal{S}^{n+1}$:

$$\min_{\substack{K\in\mathcal{S}^N,Z\in\mathcal{S}^{n+1}\\ \nu,t\in\mathbb{R},p,q\in\mathbb{R}^m}} \quad t + \lambda_{dist}e^T(p+q) + \lambda_{rke}\mathrm{tr}(K) \tag{8.37a}$$

$$\text{s.t.} \quad \begin{bmatrix} K_{oo} & \nu e \\ \nu e^T & t \end{bmatrix} - Z = \begin{bmatrix} -\lambda_{reg}I & y \\ y^T & 0 \end{bmatrix} \tag{8.37b}$$

$$\mathcal{B}(K) + p - q = d \tag{8.37c}$$

$$K \succeq 0, Z \succeq 0, p \geq 0, q \geq 0. \tag{8.37d}$$

Finally, we express the matrix equality element-wise by defining the linear operator: $\mathcal{V}(X)$ which extracts the lower triangular part of the leading n-by-n sub-matrix of matrix X into a vector of size $n(n+1)/2$:

$$\min_{\substack{K\in\mathcal{S}^N, Z\in\mathcal{S}^{n+1}\\ \nu,t\in\mathbb{R}, p,q\in\mathbb{R}^m}} \quad t + \lambda_{dist} e^T(p+q) + \lambda_{rke}\mathrm{tr}(K) \tag{8.38a}$$

$$\text{s.t.} \quad \mathcal{V}(K) - \mathcal{V}(Z) = -\lambda_{reg}\mathcal{V}(I) \tag{8.38b}$$

$$\nu e - z = y \tag{8.38c}$$

$$t - \zeta = 0 \tag{8.38d}$$

$$\mathcal{B}(K) + p - q = d \tag{8.38e}$$

$$X \succeq 0, Z \succeq 0, p \geq 0, q \geq 0. \tag{8.38f}$$

Note that problem data, labels and distances appear as right-hand-side vectors (8.38c) and (8.38e) in the standard form joint RKE-regression linear SDP.

Since in this chapter we are assuming a transductive setting, the goal is to learn a function that predicts the responses of the $N - n$ unlabeled objects. We proposes tuning the regularization parameters $\lambda = (\lambda_{reg}, \lambda_{dist}, \lambda_{rke})$ by minimizing an approximation to the ordinary leave-one-out cross-validation criterion where the joint Regression-RKE problem is solved withholding one response at a time and computing the error of the predicted label. That is, we minimize

$$V_0(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_{\lambda i}^{[-i]})^2, \tag{8.39}$$

where $f_{\lambda i}^{[-i]}$ is the response for object i predicted by the solution of the joint Regression-RKE problem where the response for object i is left out. In the notation of Section 8.2 we have $g(y_i, f_i(X_\lambda)) = (y_i - f_i(X_\lambda))^2$, where $X_\lambda = (K_\lambda, \nu_\lambda, t_\lambda, Z_\lambda)$ in the notation of Section 8.3. By the constraints in the joint problem, we can write $f_i(X_\lambda) = -e_i^T K_{\lambda oo} Z_{\lambda 11}^{-1} z_\lambda + \nu_\lambda$. To use the SDP-GACV approximation in this case matrix $H(\lambda)$ must be defined as in Section 8.2, although here $(f_i(X_\lambda) - b_i)\frac{\partial g(f_i(X_\lambda))}{\partial f} = 2(y_i - f_i(X_\lambda))^2$ so the following special case may be derived

$$V_{reg}(\lambda) = OBS(\lambda)\left[\frac{1}{n}\mathrm{tr}(I + \sigma_n Q(\lambda))\right]. \tag{8.40}$$

where matrix $Q(\lambda)$ is defined as

$$Q(\lambda)_{ij} = \begin{cases} 2\mathrm{tr}F_\lambda^i S_\lambda^{-1} A^j X_\lambda & \text{if } i \leq n \\ 0 & \text{o.w.} \end{cases}, \tag{8.41}$$

The motivation for the approximation $\Delta b_i^* = (f_i(X_\lambda) - b_i)$ can be explained now in terms of this criterion. We use $(f_i(X_\lambda) - b_i)$ as an approximation of $(f_i(X_\lambda^{[-i]} - b_i)$ which in this case makes the leave-one-out lemma equivalent to the leave-one-out lemmas proven in derivations of the GCV and GACV (Wahba, 1990), where it is shown that using the prediction from the leave-one-out solution as the corresponding response in the full data problem yields the leave-one-out solution. The assumption in this approximation is that $f_i(X_\lambda)$ and $f_i(X_\lambda^{[-i]})$ are "close".

8.4 Discussion

In this chapter we have delineated an extension to the RKE framework where a joint regression-distance fit objective is optimized. This technique begins to address the problems that arise when tuning the regularization parameter in RKE in an independent step discussed in Appendix A. As with the RKE framework, distances may be noisy, incomplete and/or inconsistent. Thus, this methodology will be the first to address prediction solely from this type of data.

Chapter 9

Further Prospects

In this final chapter we address some future directions in which the work presented can be extended.

9.1 Tree-Structured Covariance Matrix Estimation

One of the main goals of the work presented here on estimating tree-structured covariance matrices is that once the proper representation of this class of matrices is in place, estimation problems can be cast as instances of well-known numerical optimization problems, and thus, existing solvers can be employed. However, as a future direction, creating specialized solvers for this particular type of problems can allow for larger problem instances to be solved.

A promising avenue in the unknown topology case is to set aside modeling by mixed-integer constraints and use a methodology similar to the sparse reconstruction approach (Figueiredo et al., 2007). In this case, the basis matrix V can be, in principle, extended to include all possible columns that appear in valid basis matrices, that is, that satisfy the partition property (Section 2.2). With each column of the over-complete basis matrix V is associated an element of the vector of branch lengths d. Since the basis matrix is now over-complete, a penalty on d is used to enforce the partition property of the columns of V corresponding to non-zero entries in d. The composite absolute penalties defined in Zhao et al. (2006) is a first option. One last note, as in the sparse reconstruction setting, the design matrix in the optimization problem is assumed to be only available through look-up operations. Given the structure of the basis matrices in use in this case, this is easily implemented.

We presented results for estimates given by solutions of projection problems. It is of interest as well to make use of any distributional assumptions of the diffusion process over the tree, and get estimates through maximum likelihood. For example, under a normality assumption, we must extend our computational methods to determinant maximization problems. Solving these and similar types of nonlinear MIPs is an active area of research in the optimization community (Lee, 2007).

Finally, we can leverage these methods in principled hypothesis testing frameworks that better assess the presence of hierarchical structure in observed data.

9.2 Graph-Based Prediction in SS-ANOVA Models

Throughout the experiments and simulations presented in the Section of the dissertation on SS-ANOVA models that include pedigree data we have used genetic marker data in a very simple manner by including single markers for each gene in an additive model. A more realistic model should include multiple markers per gene and would include interaction terms between these markers. Along the same lines, we currently use a very simple inheritance model to define pedigree dissimilarity. Including, for example, dissimilarities between unrelated subjects might prove advantageous. A simple example would be including a spousal relationship when defining dissimilarity since this would be capturing some shared environmental factors. Extensions to this methodology that include more complex marker models and dissimilarity measures are fertile grounds for future work.

We found that results for the RKE/RBF methodology differed substantially depending on the tuning method used. For example, we found that the GACV criterion did not yield good results for the full marker, environmental covariates and pedigree model. Developing a version of the GACV criterion that is better suited to the type of kernel matrices arising in this setting is an important future direction.

Another promising avenue for future work is to test the applicability of this methodology in other settings. For example, in social networking settings, the structure of the network can be revealing and useful when predicting, say, purchasing patterns in a population. However, a number

of other features, traditionally used in data mining applications must be weighted against this network effect. The SS-ANOVA framework can be useful in elucidating that type of trade-off.

9.3 MPF Queries and Probabilistic Inference

The MPF query setting provides a framework where scalability can be addressed in the usual relational database sense. However, the hope for probabilistic inference is that it can be scaled to large web-scale models, especially models that include relationship information. Towards that end we propose to extend the optimization of MPF queries in two directions that address this need.

9.3.1 Approximate MPF Query Evaluation

Most recent activity in research for Probabilistic Inference in the Graphical Model community is centered on approximate methods. Of particular interest is the work on Variational methods (Wainwright and Jordan, 2003). Translating these methods to the MPF setting would provide extra insights into the characteristics of these approximate methods.

In particular, methods such as Generalized Belief Propagation (Yedidia et al., 2000) and Structured Mean Field (Saul and Jordan, 1996) can be seen as schema transformation techniques that allow faster query evaluation while, hopefully, controlling approximation error. As in the case of Variable Elimination, by recasting the objectives in this method in terms of cost-based database optimization we can provide for scalable versions of these methods.

9.3.2 Templetized Workloads

A common characteristic of many probabilistic relational models (Friedman et al., 1999; Heckerman et al., 2004; Singla and Domingos, 2005), is that inference is performed in models resulting by unrolling instances of classes, or templates, of probabilistic structures. These unrolled models share common structural features that can be exploited by set-oriented computations. Additionally, these methods can also gain from specifically tailored view materialization techniques.

9.3.3 Theoretical Properties

Theoretical properties of MPF queries, for example, the complexity of deciding containment, are intriguing. While general results for arbitrary aggregate queries exist, we think that the MPF setting specifies a constrained class of queries that might allow for interesting and useful results.

Bibliography

S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Trans. Info. Theory*, 46(2): 325–343, March 2000.

S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J. Mol. Biol*, 215(3):403–410, 1990.

T.W. Anderson. Asymptotically Efficient Estimation of Covariance Matrices with Linear Structure. *The Annals of Statistics*, 1(1):135–141, 1973.

B. Atkinson and T. Therneau. *kinship: mixed-effects Cox models, sparse matrices, and modeling data from large pedigrees*, 2007. R package version 1.1.0-18.

P.N. Baird, F.M.A. Islam, A.J. Richardson, M. Cain, N. Hunt, and R. Guymer. Analysis of the Y402H Variant of the Complement Factor H Gene in Age-Related Macular Degeneration. *Investigative Ophthalmology & Visual Science*, 47(10):4194–4198, 2006.

O. Banerjee and G. Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. *Proceedings of the 23rd international conference on Machine learning*, pages 89–96, 2006.

S.J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

D. Bertsimas and R. Weismantel. *Optimization over integers*. Dynamic Ideas, 2005.

J.F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer, 2000.

B. Borchers. CSDP, A C library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999.

D.G. Brown and I.M. Harrower. Integer programming approaches to haplotype inference by pure parsimony. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(2):141–154, 2006.

W.L. Buntine. Operations for learning with graphical models. *J. Artif. Intell. Res. (JAIR)*, 2: 159–225, 1994.

D. Burdick, P. Deshpande, T.S. Jayram, R. Ramakrishnan, and S. Vaithyanathan. OLAP over uncertain and imprecise data. In *VLDB*, pages 970–981, 2005.

L.L. Cavalli-Sforza and A.W.F. Edwards. Phylogenetic analysis: models and estimation procedures. *Evolution*, 21(3):550–570, 1967.

O. Chapelle and V. Vapnik. Model selection for support vector machines. *Advances in Neural Information Processing Systems*, 12, 1999.

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*, 46(1):131–159, 2002.

S. Chaudhuri and K. Shim. Including Group-By in Query Optimization. In *VLDB*, pages 354–366, 1994. URL `citeseer.ist.psu.edu/chaudhuri94including.html`.

S. Chaudhuri and K. Shim. Optimizing queries with aggregate views. In *Proc. 5th Int'nl. Conf. on Extending DB Technology*, pages 167–182. Springer-Verlag, 1996. ISBN 3-540-61057-X.

S. Chaudhuri, M. Drton, and T.S. Richardson. Estimation of a covariance matrix with zeros. *Biometrika*, 94(1):199, 2007.

W. Chu, V. Sindhwani, Z. Ghahramani, and S.S. Keerthi. Relational Learning with Gaussian Processes. *Advances in Neural Information Processing Systems: Proceedings of the 2006 Conference*, 2007.

H. Corrada Bravo. *Rcplex: R interface to CPLEX*, 2008. URL `http://www.r-project.org`. R package version 0.1-3.

H. Corrada Bravo and R. Ramakrishnan. Optimizing mpf queries: Decision support and probabilistic inference. Technical Report CS1567, Univ. of Wisconsin-Madison, 2006.

R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, New York, 1999.

N. Dalvi and D. Suciu. Answering queries from statistics and probabilistic views. In *VLDB*, pages 805–816, 2005.

N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, 2004.

T. Delaveau, A. Delahodde, E. Carvajal, J. Subik, and C. Jacq. PDR3, a new yeast regulatory gene, is homologous toPDR1 and controls the multidrug resistance phenomenon. *Molecular Genetics and Genomics*, 244(5):501–511, 1994.

L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Number 31 in Applications of Mathematics. Springer-Verlag, New York, 1996.

M. Drton and T.S. Richardson. A New Algorithm for Maximum Likelihood Estimation in Gaussian Graphical Models for Marginal Independence. *UAI (Uffe Kjærulff and Christopher Meek, eds.), San Francisco: Morgan Kaufmann*, pages 184–191, 2003.

M. Drton and T.S. Richardson. Iterative conditional fitting for Gaussian ancestral graph models. *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 130–137, 2004.

A.O. Edwards, R. Ritter, K.J. Abel, A. Manning, C. Panhuysen, and L.A. Farrer. Complement Factor H Polymorphism and Age-Related Macular Degeneration, 2005.

Y. El Fattah and R. Dechter. An evaluation of structural parameters for probabilistic reasoning: Results on benchmark circuits. In *UAI*, pages 244–251, 1996.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

T. Fawcett. ROC graphs: Notes and practical considerations for researchers. *Machine Learning*, 31, 2004.

J.C. Fay and P.J. Wittkopp. Evaluating the role of natural selection in the evolution of gene regulation. *Heredity*, 1:9, 2007.

J. Felsenstein et al. *Inferring phylogenies*. Sinauer Associates Sunderland, Mass., USA, 2004.

M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):586–597, 2007.

S.A. Fisher, G.R. Abecasis, B.M. Yashar, S. Zareparsi, A. Swaroop, S.K. Iyengar, B.E.K. Klein, R. Klein, K.E. Lee, J. Majewski, et al. Meta-analysis of genome scans of age-related macular degeneration. *Human Molecular Genetics*, 14(15):2257–2264, 2005.

W.M. Fitch and E. Margoliash. Construction of Phylogenetic Trees. *Science*, 155(3760):279–284, 1967.

K.A. Frazer, C.M. Wade, D.A. Hinds, N. Patil, D.R. Cox, and M.J. Daly. Segmental Phylogenetic Relationships of Inbred Mouse Strains Revealed by Fine-Scale Analysis of Sequence Variation Across 4.6 Mb of Mouse Genome. *Genome Research*, 14:1493–1500, 2004.

R.W. Freund and F. Jarre. A sensitivity result for semidefinite programs. *Operations Research Letters*, 32(2):126–132, 2004.

N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999. URL `citeseer.nj.nec.com/friedman99learning.html`.

L.G. Fritsche, T. Loenhardt, A. Janssen, S.A. Fisher, A. Rivera, C.N. Keilhauer, and B.H.F. Weber. Age-related macular degeneration is associated with an unstable ARMS2 (LOC387715) mRNA. *Nature Genetics*, 40(7):892–896, 2008.

N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Syst.*, 15(1):32–66, 1997.

A.P. Gasch, A.M. Moses, D.Y. Chiang, H.B. Fraser, M. Berardini, and M.B. Eisen. Conservation and evolution of cis-regulatory systems in ascomycete fungi. *PLoS Biol*, 2(12):e398, 2004.

E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R.D. Appel, and A. Bairoch. ExPASy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Research*, 31(13): 3784–3788, 2003.

R.C. Gentleman, V.J. Carey, D.M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, et al. Bioconductor: open software development for computational biology and bioinformatics. *feedback*, 2006.

L. Getoor. Link-based Classification. *Advanced Methods for Knowledge Discovery from Complex Data*, 2005.

A.B. Goldberg, X. Zhu, and S. Wright. Dissimilarity in Graph-Based Semi-Supervised Classification. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

C. Gu. *gss: General Smoothing Splines*, 2007. R package version 1.0-0.

C. Gu. *Smoothing Spline Anova Models*. Springer, 2002.

X. Gu. Statistical Framework for Phylogenomic Analysis of Gene Family Expression Profiles. *Genetics*, 167(1):531–542, 2004.

Z. Gu, A. Cavalcanti, F.C. Chen, P. Bouman, and W.H. Li. Extent of Gene Duplication in the Genomes of Drosophila, Nematode, and Yeast. *Molecular Biology and Evolution*, 19(3):256–262, 2002.

F. Habib, A.D. Johnson, R. Bundschuh, and D. Janies. Large scale genotype-phenotype correlation analysis based on phylogenetic trees. *Bioinformatics*, 23(7):785, 2007.

G.S. Hageman, D.H. Anderson, L.V. Johnson, L.S. Hancox, A.J. Taiber, L.I. Hardisty, J.L. Hageman, H.A. Stockman, J.D. Borchardt, K.M. Gehrs, et al. A common haplotype in the complement regulatory gene factor H (HF1/CFH) predisposes individuals to age-related macular degeneration. *Proceedings of the National Academy of Sciences*, 102(20):7227, 2005.

J.L. Haines, M.A. Hauser, S. Schmidt, W.K. Scott, L.M. Olson, P. Gallins, K.L. Spencer, S.Y. Kwan, M. Noureddine, J.R. Gilbert, et al. Complement Factor H Variant Increases the Risk of Age-Related Macular Degeneration, 2005.

D. Heckerman. A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, 1999.

D. Heckerman, C. Meek, and D. Koller. Probabilistic entity-relationship models, prms and plate models. In *SRL2004*. ICML, August 2004.

C. Helmberg, F. Rendl, R.J. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6(2):342–361, 1996.

R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

Y.T. Huang, K.M. Chao, and T. Chen. An approximation algorithm for haplotype inference by maximum parsimony. *Journal of Computational Biology*, 12(10):1261–1274, 2005.

Ilog, SA. Ilog Cplex 9.0 Users Manual, 2003.

F.V. Jensen. *Bayesian networks and decision graphs*. Springer-Verlag, 2001.

T. Joachims. Estimating the generalization performance of a SVM efficiently. *Proceedings of the International Conference on Machine Learning*, 2000.

H. Jungwirth and K. Kuchler. Yeast ABC transporters–A tale of sex, stress, drugs and aging. *FEBS Letters*, 580(4):1131–1138, 2006.

A. Kanda, W. Chen, M. Othman, K.E.H. Branham, M. Brooks, R. Khanna, S. He, R. Lyons, G.R. Abecasis, and A. Swaroop. A variant of mitochondrial protein LOC387715/ARMS2, not HTRA1, is strongly associated with age-related macular degeneration. *Proceedings of the National Academy of Sciences*, 104(41):16227, 2007.

G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. Math. Anal. Appl.*, 33:82–95, 1971.

R. Klein, BE Klein, KL Linton, and DL De Mets. The Beaver Dam Eye Study: visual acuity. *Ophthalmology*, 98(8):1310–5, 1991.

R. Klein, BE Klein, SC Jensen, and SM Meuer. The five-year incidence and progression of age-related maculopathy: the Beaver Dam Eye Study. *Ophthalmology*, 104(1):7–21, 1997.

R. Klein, B.E.K. Klein, S.C. Tomany, S.M. Meuer, and G.H. Huang. Ten-year incidence and progression of age-related maculopathy: The Beaver Dam eye study. *Ophthalmology*, 109(10): 1767–1779, 2002.

R. Klein, T. Peto, A. Bird, and M.R. Vannewkirk. The epidemiology of age-related macular degeneration. *American Journal of Ophthalmology*, 137(3):486–495, 2004.

R. Klein, B.E.K. Klein, M.D. Knudtson, S.M. Meuer, M. Swift, and R.E. Gangnon. Fifteen-Year Cumulative Incidence of Age-Related Macular Degeneration The Beaver Dam Eye Study. *Ophthalmology*, 114(2):253–262, 2007.

R.J. Klein, C. Zeiss, E.Y. Chew, J.Y. Tsai, R.S. Sackler, C. Haynes, A.K. Henning, J.P. SanGiovanni, S.M. Mane, S.T. Mayne, et al. Complement Factor H Polymorphism in Age-Related Macular Degeneration. *Science*, 308(5720):385–389, 2005.

F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Info. Theory*, 47(2):498–519, 2001.

J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. *ACM International Conference Proceeding Series*, 2004.

G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *The Journal of Machine Learning Research*, 5:27–72, 2004a.

G.R.G. Lanckriet et al. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16): 2626–2635, 2004b.

J. Lee. Mixed-integer nonlinear programming: Some modeling and solution issues. *IBM JOURNAL OF RESEARCH AND DEVELOPMENT*, 51(3/4):489, 2007.

K.E. Lee, B.E.K. Klein, R. Klein, and M.D. Knudtson. Familial Aggregation of Retinal Vessel Caliber in the Beaver Dam Eye Study. *Investigative Ophthalmology & Visual Science*, 45(11): 3929, 2004.

T.I. Lee, N.J. Rinaldi, F. Robert, D.T. Odom, Z. Bar-Joseph, G.K. Gerber, N.M. Hannett, C.T. Harbison, C.M. Thompson, I. Simon, et al. Transcriptional Regulatory Networks in Saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002.

H. Li, P. Stoica, and J. Li. Computationally efficient maximum likelihood estimation of structured covariance matrices. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 47(5):1314–1323, 1999.

M. Li, P. Atmaca-Sonmez, M. Othman, K.E.H. Branham, R. Khanna, M.S. Wade, Y. Li, L. Liang, S. Zareparsi, A. Swaroop, et al. CFH haplotypes without the Y402H coding variant show strong association with susceptibility to age-related macular degeneration. *Nature genetics*, 38(9):1049, 2006.

X. Lin, G. Wahba, D. Xiang, F. Gao, R. Klein, and B. Klein. Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. *Ann. Statist*, 28: 1570–1600, 2000.

F. Lu, S. Keles, S.J. Wright, and G. Wahba. Framework for kernel regularization with application to protein clustering. *Proceedings of the National Academy of Sciences*, 102(35):12332–12337, 2005.

K.P. Magnusson, S. Duan, H. Sigurdsson, H. Petursson, Z. Yang, Y. Zhao, P.S. Bernstein, J. Ge, F. Jonasson, E. Stefansson, et al. CFH Y402H confers similar risk of soft drusen and both forms of advanced AMD. *PLoS Med*, 3(1):e5, 2006.

D. Maier. *The Theory of Relational Databases*. Computer Science Press, 1983.

G. Malécot. *Les mathématiques de l'hérédité*. Masson, 1948.

B. Matern. Spatial variation, number 36 in lectures notes in statistics, 1986.

P. McCullagh. Structured covariance matrices in multivariate regression models. Technical report, Department of Statistics, University of Chicago, 2006.

T.H. Oakley, Z. Gu, E. Abouheif, N.H. Patel, and W.H. Li. Comparative Methods for the Analysis of Gene-Expression Evolution: An Example Using Yeast Functional Genomic Data. *Molecular Biology and Evolution*, 22(1):40–50, 2005.

M. Opper and O. Winther. Gaussian Processes for Classification: Mean-Field Algorithms. *Neural Computation*, 12:2655–2684, 2000.

E. Paradis, J. Claude, and K. Strimmer. Ape: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20:289–290, 2004.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988. ISBN 0-934613-73-7.

D. Penny and M.D. Hendy. The use of tree comparison metrics. *Syst. Zool*, 34(1):75–82, 1985.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

C. Ré, N. Dalvi, and D. Suciu. Efficient top-k query evaluation on probabilistic data. Technical Report 2006-06-05, University of Washington, 2006a.

C. Ré, N. Dalvi, and D. Suciu. Query evaluation on probabilistic databases. *IEEE Data Engineering Bulletin*, 29(1):25–31, 2006b.

S.A. Rifkin, J. Kim, and K.P. White. Evolution of gene expression in the Drosophila melanogaster subgroup. *Nature Genetics*, 33(2):138–144, 2003.

N. Saitou. The neighbor-joining method: a new method for reconstructing phylogenetic trees, 1987.

L.K. Saul and M.I. Jordan. Exploiting tractable substructures in intractable networks. *Advances in Neural Information Processing Systems*, 8:486–492, 1996.

B. Scholkopf and A.J. Smola. *Learning with Kernels*. MIT Press Cambridge, Mass, 2002.

T.J. Schulz. Penalized maximum-likelihood estimation of covariance matrices with linear structure. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 45(12):3027–3038, 1997.

P. Griffiths Selinger, M.M. Astrahan, D.D. Chamberlin, R.A. Lorie, and T.G. Price. Access path selection in a relational database management system. In *SIGMOD*, pages 23–34, 1979. ISBN 0-89791-001-X. doi: http://doi.acm.org/10.1145/582095.582099.

R. Sibson. Studies in the Robustness of Multidimensional Scaling: Perturbational Analysis of Classical Scaling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2): 217–229, 1979.

V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. *ACM International Conference Proceeding Series*, 119:824–831, 2005.

P. Singla and P. Domingos. Discriminative training of markov logic networks. In *AAAI*, pages 868–873, 2005.

J.P. Sinnwell and D.J. Schaid. *ibdreg: Regression Methods for IBD Linkage With Covariates*, 2007. URL `http://mayoresearch.mayo.edu/mayo/research/schaid_lab/software.cfm`. R package version 0.1.1.

A. Smola and R. Kondor. Kernels and regularization on graphs. *Conference on Learning Theory, COLT/KW*, 2003.

S. Sridhar, F. Lam, G. Blelloch, R. Ravi, and R. Schwartz. Mixed Integer Linear Programming for Maximum Parsimony Phylogeny Inference. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2008.

M.L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, 1999.

J.M. Stuart, E. Segal, D. Koller, and S.K. Kim. A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. *Science*, 302(5643):249–255, 2003.

J.F. Sturm. Using sedumi 1.0 x, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

J.F. Sturm and S. Zhang. On sensitivity of central solutions in semidefinite programming. *Mathematical Programming*, 90(2):205–227, 2001.

B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. *Advances in Neural Information Processing Systems*, 16:51, 2004.

D.C. Thomas. *Statistical Methods in Genetic Epidemiology*. Oxford Univ Press, 2004.

C.L. Thompson, G. Jun, B.E.K. Klein, R. Klein, J. Capriotti, K.E. Lee, and S.K. Iyengar. Genetics of Pigment Changes and Geographic Atrophy. *Investigative Ophthalmology & Visual Science*, 48(7):3005–3013, 2007a.

C.L. Thompson, B.E.K. Klein, R. Klein, Z. Xu, J. Capriotti, T. Joshi, D. Leontiev, K.E. Lee, R.C. Elston, and S.K. Iyengar. Complement factor H and hemicentin-1 in age-related macular degeneration and renal phenotypes. *Human Molecular Genetics*, 16(17):2135, 2007b.

K.C. Toh, M.J. Todd, and R.H. Tutuncu. SDPT3a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11(12):545–581, 1999.

R.H. Tütüncü, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2):189–217, 2003.

L. Vandenberghe, S. Boyd, and S.P. Wu. Determinant Maximization with Linear Matrix Inequality Constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.

V. Vapnik and O. Chapelle. Bounds on Error Expectation for Support Vector Machines. *Neural Computation*, 12:2013–2036, 2000.

V.N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

G. Wahba. Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV. *Advances in kernel methods: support vector learning table of contents*, pages 69–88, 1999.

G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein. Smoothing spline ANOVA for exponential families, with application to the Wisconsin Epidemiological Study of Diabetic Retinopathy. *Ann. Statist*, 23(1865):1895, 1995.

G. Wahba, Y. Lin, and H. Zhang. GACV for support vector machines, or, another way to look at margin-like quantities. *Advanced in Large Margin Classifiers*, 1999.

G. Wahba, Y. Lin, Y. Lee, and H. Zhang. On the relation between the GACV and Joachims' $\xi\alpha$ method for tuning support vector machines, with extensions to the nonstandard case. Technical report, Technical Report 1039, Statistics Department University of Wisconsin, Madison WI, 2001, 2001.

G. Wahba, Y. Lin, Y. Lee, and H. Zhang. Optimal properties and adaptive tuning of standard and nonstandard support vector machines. *Nonlinear Estimation and Classification, Springer*, pages 125–143, 2002.

M.J. Wainwright and M.I. Jordan. Graphical models, exponential families and variational inference. Technical Report 649, Department of Statistics, University of California, Berkeley, 2003.

L. Wang and Y. Xu. Haplotype inference by maximum parsimony, 2003.

Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

A. Whitehead and D.L. Crawford. Neutral and adaptive variation in gene expression. *Proceedings of the National Academy of Sciences*, 103(14):5425–5430, 2006.

L.A. Wolsey and G.L. Nemhauser. *Integer and Combinatorial Optimization*. Wiley-Interscience, 1999.

S.K.M. Wong. The relational structure of belief networks. *J. Intell. Inf. Syst.*, 16(2):117–148, 2001.

S.K.M. Wong, C.J. Butz, and Y. Xiang. A method for implementing a probabilistic model as a relational database. In *UAI*, pages 556–564, 1995.

S.K.M. Wong, D. Wu, and C.J. Butz. Probabilistic reasoning in bayesian networks: A relational database approach. In *Canadian Conference on AI*, pages 583–590, 2003.

D. Wu and S.K.M. Wong. Local propagation in bayesian networks versus semi-join program in databases. In *FLAIRS*, 2004.

D. Xiang and G. Wahba. A generalized approximate cross validation for smoothing splines with non-Gaussian data. *Statistica Sinica*, 6(3):675–692, 1996.

B. Yalcin, J. Fullerton, S. Miller, DA Keays, S. Brady, A. Bhomra, A. Jefferson, E. Volpi, RR Copley, J. Flint, et al. Unexpected complexity in the haplotypes of commonly used inbred strains of laboratory mice. *Proc Natl Acad Sci US A*, 101(26):9734–9739, 2004.

M. Yannakakis. Computing the minimum fill-in is np-complete. *SIAM J. Alg. Disc. Meth.*, 2(1): 77–79, March 1981 1981.

J.S. Yedidia, W.T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000. URL `citeseer.nj.nec.com/yedidia00generalized.html`.

J.S. Yedidia, W.T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report TR-2002-35, Mitsubishi Electric Research Laboratories, 2002. URL `www.merl.com/papers/docs/TR2002-35.pdf`.

E.A. Yildirim and M.J. Todd. Sensitivity analysis in linear programming and semidefinite programming using interior-point methods. *Mathematical Programming*, 90(2):229–261, 2001.

M. Yuan and Y. Lin. Model Selection and Estimation in the Gaussian Graphical Model. *Biometrika*, 94(1):19–35, 2007.

N.L. Zhang and D. Poole. Exploiting causal independence in bayesian network inference. *JAIR*, 5:301–328, 1996.

P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Preprint*, 2006.

X. Zhu. Semi-Supervised Learning Literature Survey. Technical Report TR 1530, Computer Science, University of Wisconsin-Madison, 2005.

X. Zhu, J. Kandola, J. Lafferty, and Z. Ghahramani. *Semi-Supervised Learning*, chapter Graph Kernels by Spectral Transforms. MIT Press, 2006.

DISCARD THIS PAGE

Appendix A: RKE: Tuning for Clustering and Classification

We have seen results in Chapter 4.3 from applying the RKE framework to protein classification tasks. There, we saw that in one of the classification tasks, prediction performance was relatively invariant for a large range of values of the RKE regularization parameter λ_{rke}. On the other hand, careful tuning of this parameter was required for good prediction in the second classification task. In this appendix, we will further explore the issues in tuning the RKE regularization parameter in both clustering and classification settings.

Section A.1 introduces the CV2 tuning method and shows how it may be used to select regularization parameter values for RKE in clustering and visualization applications. An empirical study in Section A.2 illustrates the observation that clustering, as opposed to classification, is less sensitive to a large range of values of the regularization parameter. A simulation study in Section A.3 further illustrates this observation.

A.1 The CV2 Tuning Method

In this Section, we present the CV2 pairwise tuning method for choosing the regularization parameter λ_{rke} in Eq. (8.2). CV2 is a set-aside tuning set method where pairwise dissimilarities are estimated for objects in a tuning set by embedding them in the space spanned by an RKE kernel estimated with regularization parameter λ_{rke}. After embedding the objects in the tuning set using the newbie algorithm, we compare their original dissimilarities with their squared distance in the embedding space.

Suppose we have dissimilarity data for a tuning set T of objects where T is disjoint from the set of N objects used for training RKE. Let $K_{\lambda_{rke}}$ be a kernel matrix estimated using RKE with regularization parameter λ_{rke} on the training set of N objects. Let d_{ij} be the dissimilarity of objects i and $j \in T$, and for object i, let Γ_i be a vector of dissimilarity measurements between object i and a subset of the N objects used for training RKE. Let $x_{\lambda_{rke}}(i)$ be the coordinate vector estimated

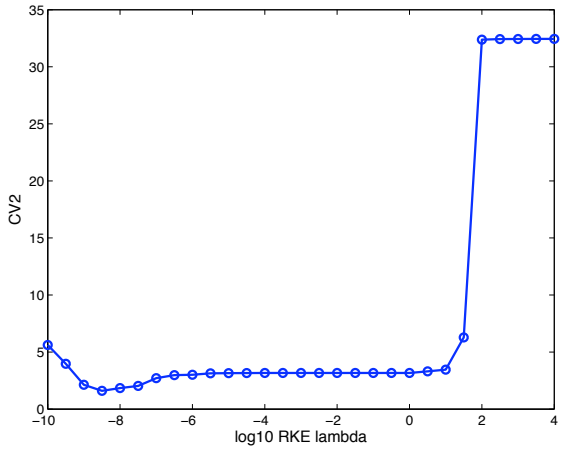


Figure A.1 CV2 curve as function of regularization parameter.

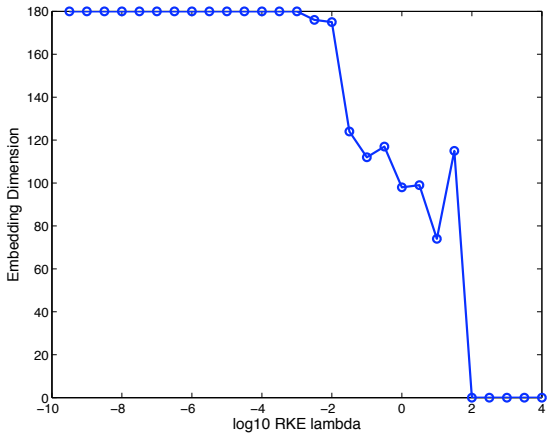


Figure A.2 Embedding dimensionality for newbie algorithm.

for object i by the newbie algorithm applied to Γ_i and $K_{\lambda_{rke}}$. We define $CV2(\lambda_{rke})$ as

$$CV2(\lambda_{rke}) = \sum_{i,j \in T} |\|x_{\lambda_{rke}}(i) - x_{\lambda_{rke}}(j)\|_2^2 - d_{ij}|. \tag{A.1}$$

Figure A.1 shows the CV2 curve for the data from structural classification task in Chapter 4.3 as a function of $\log_{10}(\lambda_{rke})$. The CV2 tuning set contains 10% of the objects in the original dataset, selected as follows: at first, an object was chosen at random to be in the training set, from then on, the next object is chosen at random from the set of unchosen neighbors of the current object, until 90% of the objects have been included in the training set. This maintains connectivity of the training set graph. The embedding dimensionality for the newbie algorithm was determined using the same relative zero procedure of Chapter 4.3; we show the resulting dimensionalities in Figure A.2. The newbie problem was solved using the SeDuMi Second-Order Cone Programming solver (Sturm, 1999).

Although we can see a clear minimum in Figure A.1, the CV2 curve in this case is rather flat, with a large range of values of the regularization parameter exhibiting similar performance. Figure A.3 shows the embedding of the data corresponding to $\log_{10}(\lambda_{rke}) = -8.5$, which minimizes CV2 in this case. The fact that this embedding is very similar to that of Figure 4.1 is consistent with

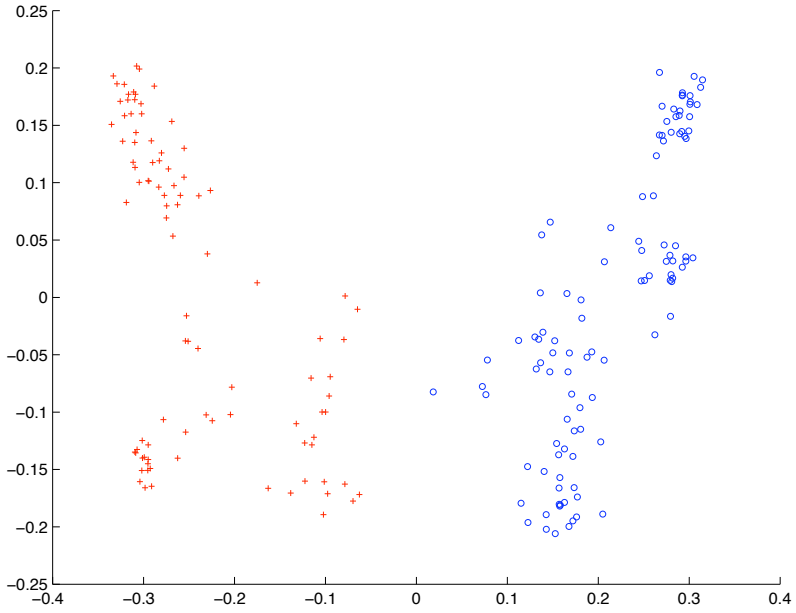


Figure A.3 Data embedding for $\log_{10}(\lambda_{rke}) = -8.5$.

our observation that performance for purposes of visualization and clustering is mostly invariant of the regularization parameter.

A.2 Tuning RKE for Classification

We have seen that both classification and clustering performance in our protein data set are invariant to the value of the regularization parameter in RKE. If this were a general phenomenon, RKE may be applied to large data sets since, lacking a need for careful tuning, the expensive to solve RKE problem would only have to be solved a small number of times. If, on the other hand, careful tuning is required, then efficient tuning strategies and the scalability of RKE would have to be addressed. In this section, we show an example based on the protein classification task where careful tuning is in fact required for classification, whereas clustering performance exhibits a similar behavior of invariance to the regularization parameter.

Our new data set is shown in Figure A.4. This was obtained by transforming the eigenspectrum given in Figure 4.2 by reducing the magnitude of the two leading eigenvalues. The remaining

eigenvalues and all eigenvectors were not transformed. We can see that the general characteristics of the two clusters are retained, while now, at least in low dimensions, good classification by a linear function becomes slightly harder. The eigenspectrum used to generate the data is shown in Figure A.5. With this transformation we have essentially reduced variance in the direction with the highest variance in the original data set. Thus, it is expected that cluster characteristics are maintained, while bringing the embedded data points together in the two dominant directions.

From this transformed embedding of protein sequences we compute Euclidean distances in 58-dimensional space. These distances were then given to RKE as input, with the same 3,994 pairs of objects selected in the original classification task. We now show how classification performance and clustering performance are affected under this data transformation.

Figure A.6 shows the CV2 curve for RKE given the distances for the transformed dataset. We see that this curve is almost identical to the CV2 curve of the original data shown in Figure A.1. Clustering performance is not significantly affected by the data transformation. In addition, we can see the same wide range of similar performance for CV2. This, again, indicates that for clustering purposes carefully tuning the regularization parameter might not be necessary.

If this phenomenon is again reflected in the classification performance, then we can safely say that in this case, careful tuning of the regularization parameter is not required. As we stated previously, this would make RKE much more efficient since the expensive step of solving the RKE problem would have to be executed a very small number of times.

Figure A.7 shows the error curve for the transformed dataset. Unfortunately, we see that performance is very sensitive to the value of the regularization parameter. While there is a value of the regularization parameter which yields an SVM with perfect classification accuracy, most other values of the regularization parameter do not perform well. This indicates that in this case, careful tuning of the regularization parameter is in fact needed. Thus, efficient tuning methods and the efficiency of solving the RKE problem must be addressed to make this an effective framework for classification.

The degradation in classification performance might also be explainable by the fact that the distances given to RKE were given by Euclidean distance in the protein embedding space. To verify

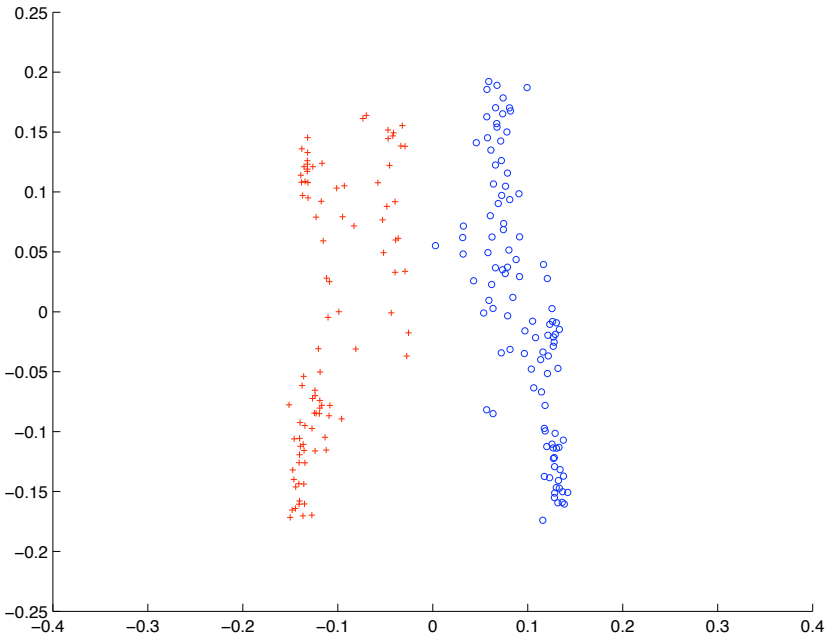


Figure A.4 Transformed protein dataset.

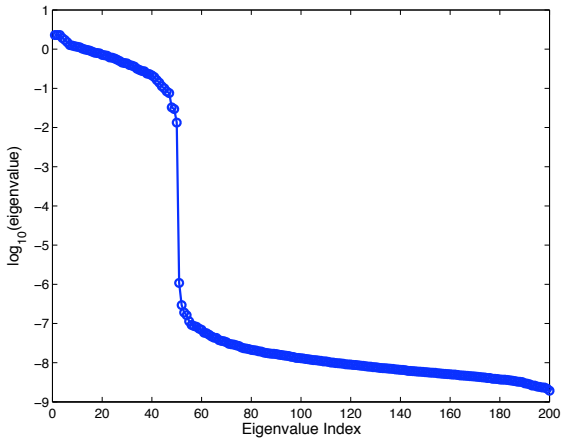


Figure A.5 Eigenspectrum of transformed data.

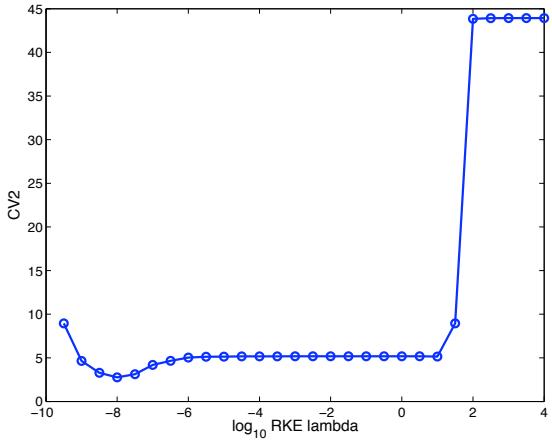


Figure A.6 CV2 curve for transformed dataset.

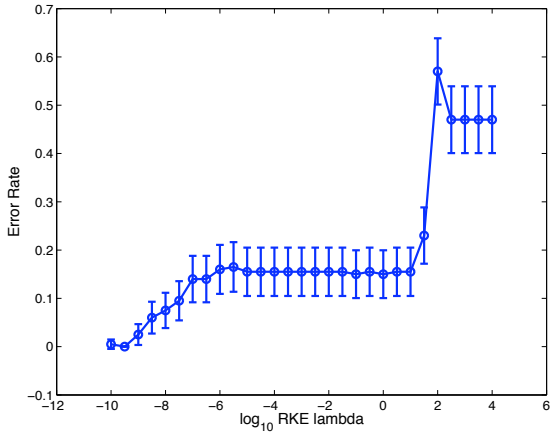


Figure A.7 Error curve for transformed dataset.

that this had a limited effect on classification performance we performed the same experiment as above with distance data obtained from the protein embedding space without transformation, that is, without changing the two leading eigenvalues. Figure A.8 shows the misclassification rate for this experiment. Although there is some degradation in performance it is not nearly as large as that in Figure A.7.

A.3 Simulation Study

To further study the properties of the tuning methods for clustering and classification presented above, we created the artificial *slashdot* dataset. Figure A.9 plots the three signal dimensions for one instance of this dataset. 100 samples were generated from two three dimensional normal distributions respectively. To each sample we append three additional spurious coordinates of independent normal noise ($\sigma = 0.3$). Euclidean distance was computed for each pair of points and the distances binned into n_b bins of equal size. Smaller values of n_b generate noisier dissimilarity data. For each point we include dissimilarity information for 20 other randomly selected points in the training set. 20% of the training set was selected at random as the tuning set for CV2. The distance between clusters is determined by parameter τ. The smaller τ is, the less separable the resulting dataset. For Figure A.9 we have $\tau \approx 4$.

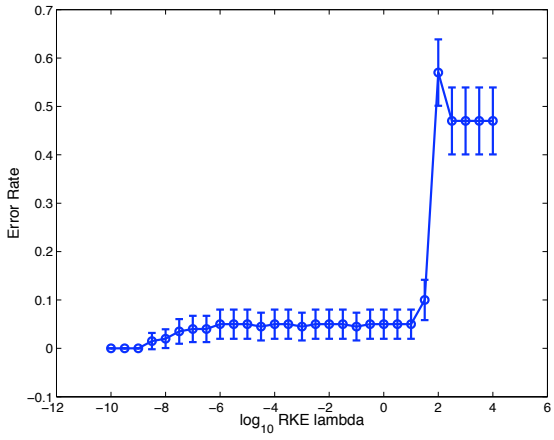


Figure A.8 Error curve for Euclidean distance data from untransformed protein embedding space.

In the following, we use 90% of the trace of a kernel $K_{\lambda_{rke}}$ to determine the dimensionality of the embedding space for the newbie algorithm, and to truncate $K_{\lambda_{rke}}$ before using it for SVMs.

To determine the suitability of $CV2$ as a tuning criteria for RKE we compare it to the normalized Procrustes measure (Sibson, 1979). Given a set of points in Euclidean $\mathbb{R}^d$ space, its Gram matrix K is such that $K_{ij} = x_i' x_j$, where x_i and x_j are column vectors. The normalized Procrustes measure determines the positional similarity after matching two centered Gram matrices under rotation, translation and reflection:

$$D(K, K_{\lambda_{rke}}) = \frac{\text{trace}(K) + \text{trace}(K_{\lambda_{rke}}) - 2\text{trace}\left[K^{1/2} K_{\lambda_{rke}} K^{1/2}\right]^{1/2}}{\sqrt{\text{trace}(K)\text{trace}(K_{\lambda_{rke}})}}. \tag{A.2}$$

In our simulations, K will refer to the Gram matrix of our simulated data points and $K_{\lambda_{rke}}$ is the RKE kernel estimated with regularization parameter λ_{rke}.

To measure the sensitivity of classification performance, we extend the GACV criterion (Wahba et al., 2001) to define the RGACV criterion for RKE regularization parameters λ_{rke}:

$$RGACV(\lambda_{rke}) = \min_{\lambda_{svm}} GACV(\lambda_{svm}, K_{\lambda_{rke}}), \tag{A.3}$$

where $GACV(\lambda_{svm}, K_{\lambda_{rke}})$ is the GACV value of an SVM estimated with parameter λ_{svm} and kernel matrix $K_{\lambda_{rke}}$, estimated by RKE with regularization parameter λ_{rke}.

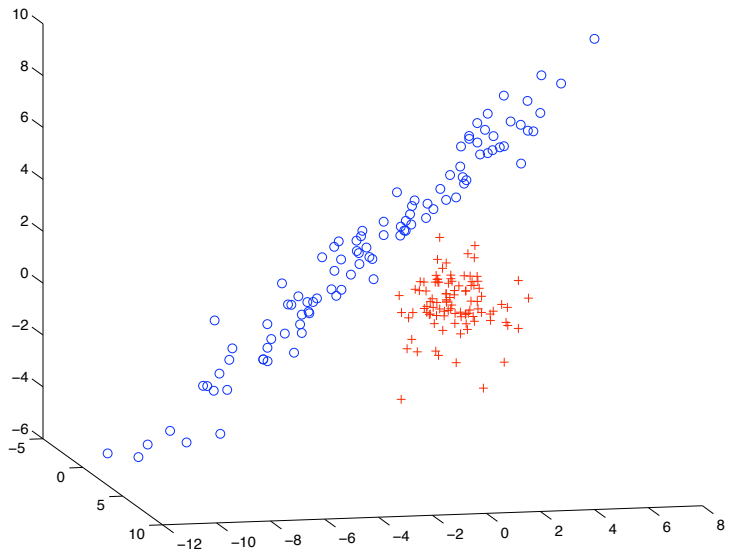


Figure A.9 Signal dimensions for *slashdot* simulation dataset

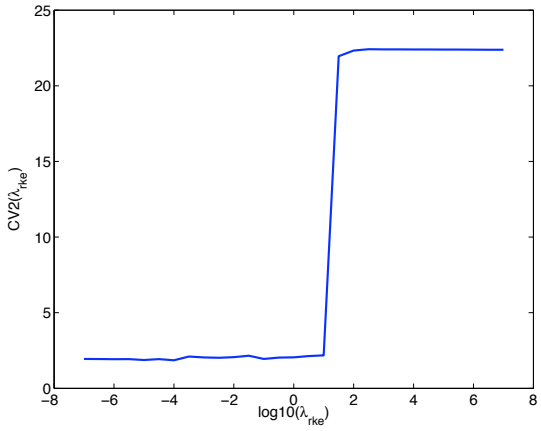


Figure A.10 CV2 curve

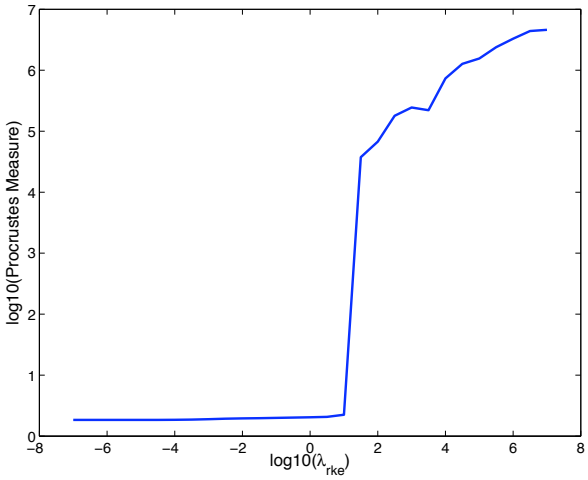


Figure A.11 Procrustes curve

Figure A.10 plots the CV2 score as a function of $\log_{10}(\lambda_{rke})$ for increasing values of the regularization parameter. Figure A.11 plots the Procrustes measure as a function of $\log_{10}(\lambda_{rke})$. We see that in this case CV2 displays the same effect of the regularization parameter on clustering performance as the Procrustes measure. However, both the CV2 and Procrustes curves in this example show that the effect of the regularization parameter is almost negligible for values of $\lambda_{rke} < 1.5$ where all RKE kernels are able to estimate distances relatively well. For values of $\lambda_{rke} \geq 1.5$ the eigenvalues of every RKE kernel are shrunk 0, thus their poor performance is expected.

Figure A.12 plots the RGACV score for this case. We see that in contrast with the CV2 and Procrustes plots in Figures A.10 and A.11, the plot in Figure A.12 exhibits a sharp minimum. According to CV2, for clustering and visualization applications, small variations in the RKE regularization parameter have little effect. This is contrary to what is shown in Figure A.12 for this classification application.

For both CV2 and Procrustes, although a number of λ_{rke} values display the same performance, one may expect that a suitable choice for kernel would be the most regularized, $\lambda_{rke} = 1.5$ in this case since that would imply lower dimensionality of the spanned space. However, in the RGACV plot of Figure A.12 we see that the kernel that does best for classification, $\lambda_{rke} \approx -0.5$, is not the most regularized. A possible explanation is that the flexibility of larger dimensionality allows for

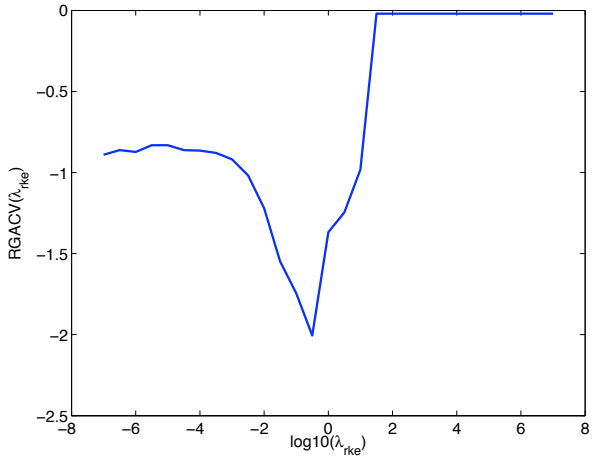


Figure A.12 RGACV curve

classification functions that are expected to have better generalization since it reduces the bounds on LOO error while still retaining low complexity.

A.4 Discussion

Chapter 4.3 has shown the utility of RKE for classification in settings where noisy dissimilarity data is provided. In particular, we have shown that for a sample of globins, an SVM fit using a kernel estimated with RKE is capable of classifying them into sub-families perfectly. Furthermore, we have shown that in this task, performance is invariant of the choice of RKE regularization parameter.

Using a transformation of this globin data we have shown that performance of SVMs and RKE can be sensitive to the choice of regularization parameter, requiring careful tuning of this parameter. In the next chapter we analyze a number of tuning methods for SVMs that might be extended to the joint tuning of RKE and SVM. Furthermore we have shown that in this instance, using performance with respect to distance recovery, via the CV2 criterion, is not suitable for tuning a classification task.

Thus, tuning methods that target classification performance of a joint RKE-SVM system are required. This has a number of implications. Due to the inefficiency of semidefinite programming in general, a tuning procedure must be able to find suitable values of the regularization parameter while solving as few RKE problems as possible. Otherwise, the task of solving the RKE problem must be made much more efficient, where a tuning procedure that is not capable of reducing the number of RKE problems to solve can be used. Another direction is solving the RKE and SVM problems jointly, but the naïve way of doing this leads to a non-convex optimization problem. We address some of these directions in Chapter 9.

Appendix B: Adaptive Tuning of Support Vector Machines

The Support Vector Machine (SVM) (Scholkopf and Smola, 2002; Vapnik, 1998) has proven to be a successful nonlinear classification method for a broad range of applications. There are two main reasons for its, at least theoretical, success: as other kernel methods, the representation of SVMs as finite expansions of kernel functions implicitly maps input data to possibly infinite spaces where linear decision functions can perform well; on the other hand, the SVM problem can be specified as the solution of a particular optimization problem whose solution has strong properties with respect to optimal decision functions.

However, properly choosing tuning parameters, both to parametrize kernel functions and the SVM optimization problem, is fundamental for the successful application of SVMs. In this chapter we review and compare a number of adaptive tuning methods for SVMs. In particular, we show that the GACV approximation to expected misclassification given in Wahba et al. (2001) is equivalent to the Support Vector Span Rule given by Chapelle and Vapnik (Chapelle et al., 2002) under certain assumptions.

B.1 The SVM Variational Problem

The Support Vector Machine (SVM) can be cast as the solution to a data-fit + penalty-term optimization problem (Scholkopf and Smola, 2002; Vapnik, 1998). Any positive definite function k induces a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$ with reproducing kernel k. That is, if $\mathcal{T}$ is some index set, and $k : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ is a positive semidefinite function, then $\mathcal{H}_k$ is a Hilbert Space of functions $f : \mathcal{T} \to \mathbb{R}$ endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$ with properties $\langle k_y(\cdot), k_x(\cdot) \rangle_{\mathcal{H}_k} = k(y, x)$, where $k_x(\cdot) = k(x, \cdot)$ and $h(x) = \langle h(\cdot), k_x(\cdot) \rangle_{\mathcal{H}_k}$ for all $h \in \mathcal{H}_k$. See Wahba (1990) for more on Reproducing Kernel Hilbert Spaces.

Given data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ with $x_i \in \mathcal{T}$, $y_i \in \{+1, -1\}$, and a kernel function $k : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$, the SVM problem is to find function $f(x) = d + h(x)$, $h(x) \in \mathcal{H}_K$ and $d \in \mathbb{R}$, to

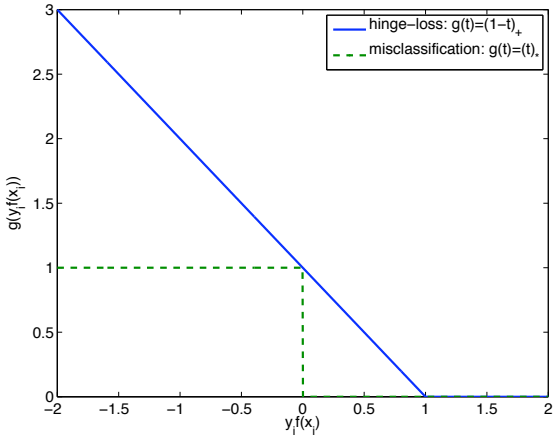


Figure B.1 Hinge-loss and Misclassification loss functions

solve the following optimization problem:

$$\min_{d\in\mathbb{R},\, h\in\mathcal{H}_k} \frac{1}{n}\sum_{i=1}^{n}(1-y_i f(x_i))_+ + \lambda\|h\|^2_{\mathcal{H}_k}, \tag{B.1}$$

where $\|h\|^2_{\mathcal{H}_k} = \langle h, h\rangle_{\mathcal{H}_k}$, and $\lambda >= 0$ is a regularization parameter. The loss function used here, referred to as "hinge-loss", is a piecewise linear function given by $(\tau)_+ = \max\{0, \tau\}$. It is a convex upper bound on misclassification $(y_i f(x_i))_*$, where

$$(\tau)_* = \begin{cases} 1 & \text{if } \tau \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

That is, $(y_i f(x_i))_* = 1$ if the signs of y_i and $f(x_i)$ disagree, indicating that the decision function f has misclassified point x_i. Figure B.1 shows both the hinge-loss and misclassification error functions.

By the Kimeldorf and Wahba Representer Theorem (Kimeldorf and Wahba, 1971), the minimizer $\hat{f}$ of (B.1) has finite representation

$$\hat{f}(\cdot) = d + \sum_{i=1}^{n} c_i K(\cdot, x_i); \tag{B.2}$$

implying $\|h\|^2_{\mathcal{H}_k} = c'Kc$, where K is an n-by-n matrix such that $K_{ij} = k(x_i, x_j)$, and c' is the transpose of vector c. Thus, we can write (B.1), after adding slack variables for hinge-loss as the

optimization problem:

$$\begin{aligned} \min_{d,c,z} \quad & e'z + n\lambda c'Kc \\ \text{s.t.} \quad & 1 - y_i(c'K_{\cdot i} + d) \leq z \ \forall i \\ & z_i \geq 0 \ \forall i. \end{aligned} \tag{B.3}$$

We'll describe various adaptive tuning methods in terms of the solution of the dual problem of (B.3). Let $H = \frac{1}{2n\lambda} YKY$ where $Y = \mathrm{diag}(y)$ is the diagonal matrix with vector y in the main diagonal, then the dual problem is:

$$\begin{aligned} \max_{\alpha} \quad & e'\alpha - \tfrac{1}{2}\alpha' H \alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq 1 \ \forall i \\ & y'\alpha = 0. \end{aligned} \tag{B.4}$$

If there exists a function $\hat{f}$ (given by the solution c and d of (B.3)) that separates the training data perfectly, that is, $\hat{f}(x_i) < 0$ if and only if $y_i = -1$, then the distance between the two closest points with different labels is a quantity of interest. Also known as the optimal separating margin, this distance γ is given by

$$\gamma^2 = (c'Kc)^{-1} = \frac{2n\lambda}{\sum_{i:y_i f(x_i)\leq 1} \alpha_i}. \tag{B.5}$$

Intuitively, if the margin is large and the underlying data distribution does not change, the resulting SVM can be expected to perform well on new data points, that is, will have good generalization performance.

B.1.1 The Tuning Problem

The generalization performance of the solution $\hat{f}$ to (B.1) depends on the given value of the trade-off parameter λ. For example, large values of λ makes $\hat{f}$ tend to constant functions. On the other hand, small values of λ allow $\|h\|^2_{\mathcal{H}_K}$ to be large, which according to Vapnik's (Vapnik, 1998) convergence bounds, as we will see below, implies slow convergence of the empirical risk of $\hat{f}$ to its expected risk. Furthermore, kernel function K might also be parametrized and different values of these parameters also affect the generalization performance of $\hat{f}$.

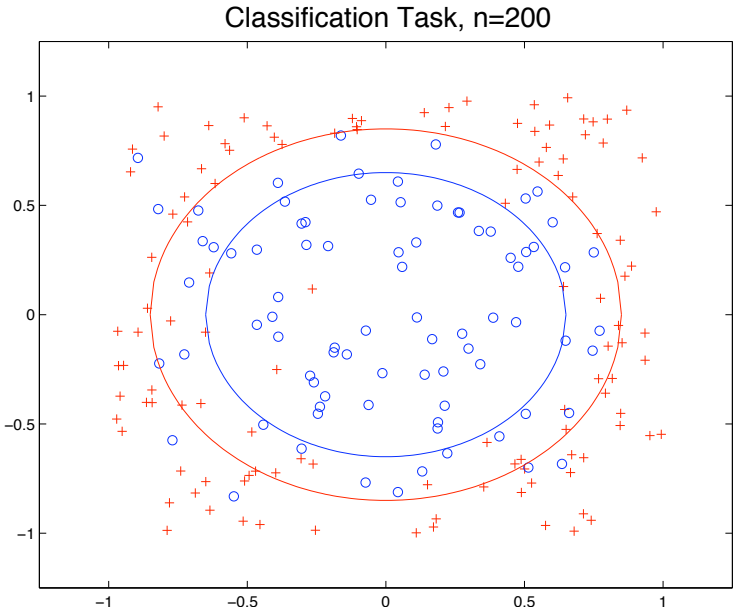


Figure B.2 A toy example classification task

Consider the, non-separable, classification task in Figure B.2, which we want to solve using an SVM with a Gaussian kernel

$$k(x, y) = \exp\left\{-\tau \|x - y\|_2^2\right\}.$$

In figure B.3, we plot three solutions to the SVM problem obtained with three different settings of regularization parameter λ, and kernel bandwidth parameter τ where we can see the effect of the regularization and bandwidth parameter. We expect the generalization performance to be better in the bottom case, where the function better resembles the optimal decision function: a concentric circle between the two circles in Figure B.2.

The tuning problem is then to find a set of parameters such the solution to the respective SVM problem has low *expected* misclassification. The difficulty lies in having access to only a finite number of data points, thus this expectation must be estimated from this finite dataset. We will compare a number of methods that attempt to estimate this expectation efficiently. From now on, we refer to both the regularization parameter and any set of kernel parameters jointly as λ.

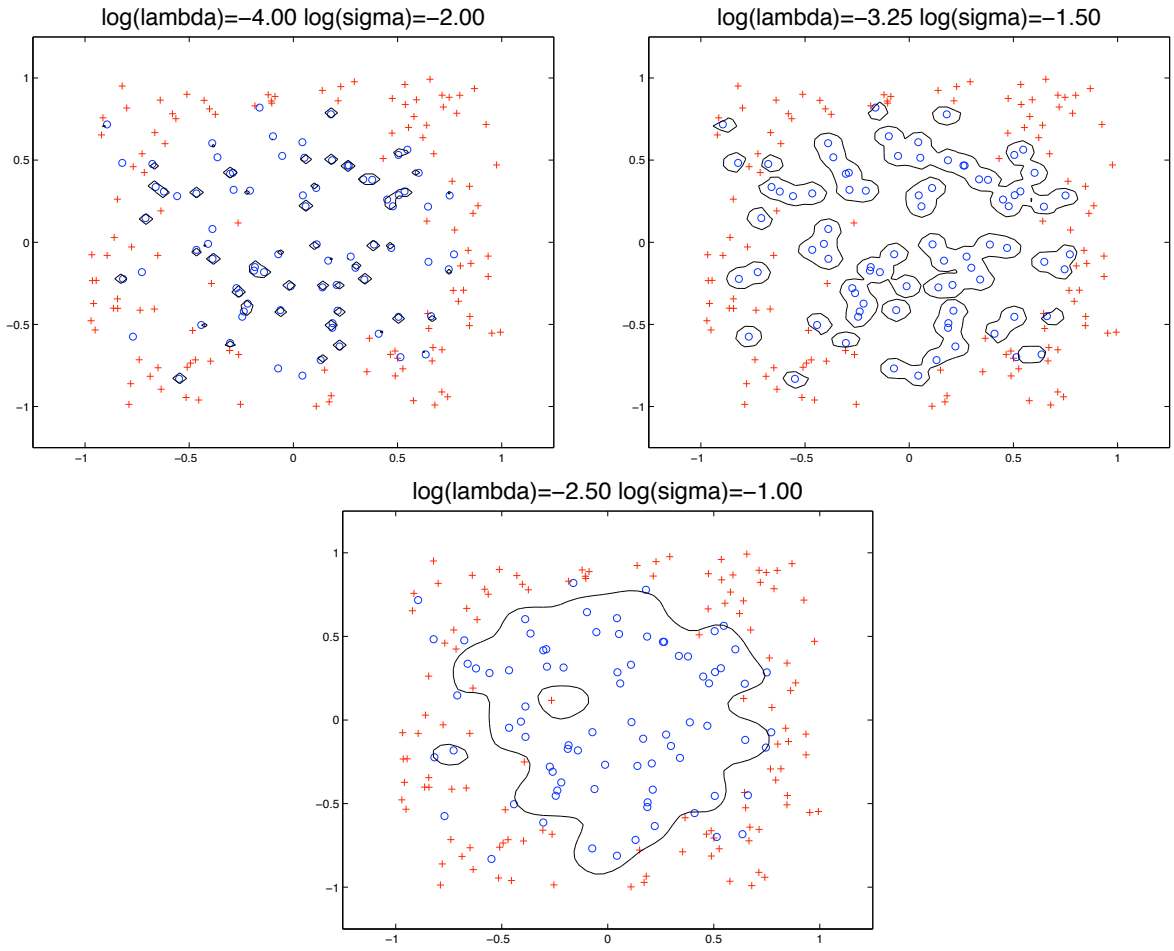


Figure B.3 Three classification functions obtained with three different settings of tuning parameters

B.1.2 The SRM Interpretation

There is another interpretation of the regularization parameter in terms of Vapnik's Structural Risk Minimization (SRM) Principle (Vapnik, 1998). This argument is given by both Evgeniou et al. (2000) and Vapnik (1998). First, we define the notions of *empirical* and *expected* risk, we have mentioned previously. Given a loss function $l(y, f(x))$, such as hinge-loss or misclassification, the *expected* risk of function f is

$$R(f) = \mathbb{E}_P \left[l(y, f(x)) \right],$$

where expectation is taken over an unknown data probability distribution $P(x, y)$. On the other hand, given a finite data set of size n, the *empirical* risk of f is

$$R_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i)).$$

In the learning setting, we have access to R_{emp} but not R which is the function we want to minimize. Vapnik proves that, in general, minimizing R_{emp} does not imply minimizing R.

Vapnik's result on the convergence of empirical risk to expected risk gives bounds of the type

$$R(f) \leq R_{emp}(f) + \Phi(v, n, \eta) \tag{B.6}$$

which hold with probability $1 - \eta$ for all f in a given function class $\mathcal{F}$. The quantity v is referred to as the VC dimension and is a measure of the complexity of function class $\mathcal{F}$. $\Psi(v, n, \eta)$ is referred to as the confidence interval.

The Structural Risk Minimization Principle defines a set of function classes $\mathcal{F}_j$, each with associated VC-dimension v_j for which the relationships $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_l$ and $v_1 \leq v_2 \leq \cdots \leq v_l$ hold. For SVMs this structure is given by an increasing sequence of constants $a_1, \ldots, a_l$ and function class $\mathcal{F}_j = \{f(x) = d + h(x) \ : \ \|h\|^2_{\mathcal{H}_K} \leq a_j^2\}$. SRM then finds the function $\hat{f}_j$ in each class $\mathcal{F}_j$ that minimizes empirical risk, and from those selects the function $\hat{f}_j$ that minimizes the right hand side of (B.6).

For SVMs with $\|h\|^2_{\mathcal{H}_K} \leq a^2$, the following holds for VC-dimension v

$$v \leq O(min(N, R^2 a^2)) \tag{B.7}$$

where R^2 is the radius of the smallest sphere containing the points $k(x_i, \cdot)$ for each data point x_i, and N is the dimensionality of $\mathcal{H}_K$.

The SRM principle could then be implemented by solving the following problem for each a_j:

$$\begin{aligned} \min_{f=d+h(x)} \quad & \tfrac{1}{n}\textstyle\sum_{i=1}^{n}(1-y_i f(x_i))_+ \\ \text{s.t.} \quad & \|h\|^2_{\mathcal{H}_K} \le a_j^2. \end{aligned} \tag{B.8}$$

The Lagrangian for this problem is:

$$\min_{f=d+h(x)} \quad \tfrac{1}{n}\textstyle\sum_{i=1}^{n}(1-y_i f(x_i))_+ + \lambda(\|h\|^2_{\mathcal{H}_K} - a_j^2). \tag{B.9}$$

Now suppose we know that the function class $\mathcal{F}_j$ contains the function which, if selected by minimizing empirical risk, then it minimizes the right-hand side of (B.6). If we have the Lagrange multiplier λ corresponding to the constant a_j which defines $\mathcal{F}_j$, then we can drop constant a_j from the Lagrangian and recover the original variational problem (B.1) from (B.9). Thus, choosing the proper value of λ implies finding $\mathcal{F}_j$, the function class for which minimizing empirical risk implies minimizing the right hand side of (B.6), therefore minimizing a bound on expected risk.

B.2 Adaptive Tuning Methods

We analyze and compare a few methods for tuning the parameters of an SVM, all of which estimate the expected risk by approximating or bounding the leave-one-out (LOO) risk. It has been shown that LOO error is an 'almost' unbiased estimate of expected error (Devroye et al., 1996).

We denote the solution of the SVM problem where the ith training point is removed as $f^{[-i]}$, which we will refer to as the leave-one-out SVM. Then, the leave-one-out risk is

$$\frac{1}{n}\sum_{i=1}^{n} l(y_i, f^{[-i]}(x_i)). \tag{B.10}$$

The methods we analyze approximate or bound this quantity:

1. $GACV$ The Generalized Approximate Cross-Validation method (Wahba, 1999; Wahba et al., 1999, 2002) approximates LOO risk for hinge-loss.

2. XA_J The $\xi\alpha$ method (Joachims, 2000) bound LOO risk for misclassification.

3. XA The version of $\xi\alpha$ given by Wahba et al. (2001) which approximates LOO risk for misclassification

4. *Support Vector Span Rule* This is a bound on LOO risk for misclassification, which can be tightened to an exact estimate under certain conditions (Chapelle and Vapnik, 1999; Chapelle et al., 2002; Vapnik and Chapelle, 2000).

The main result in this section, Proposition B.1 states that under certain conditions the GACV approximation to LOO risk for misclassification (XA), is equivalent to the Support Vector Span Rule.

B.2.1 GACV

The GACV (Wahba et al., 1999) was proposed to approximate the Generalized Comparative Kullback-Leibler (GCKL) distance of two distributions in penalized likelihood settings. Denote the solution to (B.1) for a given λ as f_λ and $f_{\lambda i} = f_\lambda(x_i)$. Then the GCKL of f_λ, $GCKL(f_\lambda) = GCKL(\lambda)$, is defined as:

$$GCKL(\lambda) = \mathrm{E}_{true}\frac{1}{n}\sum_{i=1}^{n}(1 - y_i f_{\lambda i})_+, \tag{B.11}$$

where expectation is taken with respect to an unknown conditional probability $P(y|x)$. That is, given a fixed function f_λ and a set of observations $x_1, ..., x_n$, GCKL is the expected risk (with hinge-loss) of f_λ on this set of observations. The LOO estimate of GCKL, which is also the LOO estimate of expected risk, can be written as

$$LOO(\lambda) = R_{emp}(f_\lambda) + D(\lambda),$$

where $D(\lambda) \approx \frac{1}{n}\sum_{i=1}^{n} g(y_i(f_{\lambda i} - f_{\lambda i}^{[-i]}))$. The quantity $y_p(f_{\lambda p} - f_{\lambda p}^{[-p]})$ measures how much the decision function of the leave-one-out SVM differs from the SVM trained on the entire dataset for the data point x_p that was left out of the dataset. The methods we analyze here essentially differ on how they either approximate or bound this quantity.

For the GACV, $y_p(f_{\lambda p} - f_{\lambda p}^{[-p]})$ is approximated, using a finite differences argument and a leave-one-out lemma similar to that used for GCV spline estimates (Wahba, 1990), by $\frac{\partial f_{\lambda p}}{\partial y_p}$, which, referring to the primal and dual SVM problem, may be interpreted as $\frac{\alpha_{\lambda p}}{2n\lambda} K_{pp}$, where $K_{pp} = k(x_p, x_p) = \|k(\cdot, x_p)\|_{\mathcal{H}_k}$ and $\alpha_{\lambda p}$ is the corresponding component of the dual variable α from the solution of the SVM dual problem (B.4).

The GACV is defined as:

$$GACV(\lambda) = \frac{1}{n}\left[\sum_{i=1}^{n} z_i + 2 \sum_{i:y_i f_{\lambda i} < -1} \frac{\alpha_{\lambda i}}{2n\lambda} K_{ii} + \sum_{i:y_i f_{\lambda i}} \frac{\alpha_{\lambda i}}{2n\lambda} K_{ii}\right], \tag{B.12}$$

where $z_i = (1 - y_i f_{\lambda i})_+$ can be obtained from variable z in the primal problem (B.3).

B.2.2 Joachim's $\xi\alpha$

As opposed to the GACV, the $\xi\alpha$ procedure Joachims (2000) bounds expected misclassification rate. Let $R^2 \geq K_{ii} - K_{ij}$ for all i, j, then the $\xi\alpha$ bound is

$$XA_J(\lambda) = \frac{1}{n}\left[\sum_{i=1}^{n} \xi_i + \sum_{i:y_i f_{\lambda i} \leq 1} I_{\left[\frac{\rho\alpha_{\lambda i}}{2n\lambda} R^2\right]}(y_i f_{\lambda i})\right], \tag{B.13}$$

where $\xi_i = (y_i f_{\lambda i})_*$ and

$$I_{[\theta]}(\tau) = \begin{cases} 1 & \text{if } \tau \in (0, \theta] \\ 0 & \text{otherwise} \end{cases}$$

Joachim proves that XA_J bounds the LOO to expected risk for misclassification if $\rho = 2$.

Since $R^2 \geq K_{ii} - K_{ij}$ for all i, j, we have

$$\begin{aligned} 2R^2 &\geq K_{ii} - K_{ij} + K_{jj} - K_{ij} \\ &= \langle k_{x_i}(\cdot) - k_{x_j}(\cdot), k_{x_i}(\cdot) - k_{x_j}(\cdot)\rangle_{\mathcal{H}_k} \\ &= \|K_{\cdot i} - K_{\cdot j}\|^2_{\mathcal{H}_k}. \end{aligned}$$

That is, R^2 is an upper bound on the radius of the smallest sphere in $\mathcal{H}_K$ containing the representers ($k_x(\cdot)$) for the data points. Thus the term $\frac{\alpha_{\lambda i}}{2n\lambda} R^2$ is the ith term in the radius-margin VC dimension bound given in (B.7). For XA_J, we have that $y_p(f_{\lambda p} - f_{\lambda p}^{[-p]}) \leq \frac{\alpha_{\lambda p}}{2n\lambda} R^2$.

B.2.3 A GACV version of $\xi\alpha$

The relationship between GACV and $\xi\alpha$ is given in Wahba et al. (2001) where an approximation of the LOO estimate of misclassification rate is found using a similar derivation to GACV. As we saw above, GACV is treated as a proxy for GCKL, in this case, however, for misclassification rate. Now, we take

$$GCKL(\lambda) = \mathrm{E}_{true}\frac{1}{n}\sum_{i=1}^{n}(y_i f_{\lambda i})_*. \tag{B.14}$$

The GACV approximation in this case is

$$XA(\lambda) = \frac{1}{n}\left[\sum_{i=1}^{n}\xi_i + \sum_{i:y_i f_{\lambda i}\leq 1} I_{\left[\frac{\alpha_{\lambda i}}{2n\lambda}K_{ii}\right]}(y_i f_{\lambda i})\right]. \tag{B.15}$$

In this case, $y_p(f_{\lambda p} - f_{\lambda p}^{[-p]}) \approx \frac{\alpha_{\lambda p}}{2n\lambda}K_{pp}$. Note that $\max_i K_{ii}$ can be used in place of R^2 in XA_J since $\max_i K_{ii} \geq K_{jj} - K_{jk}$ for all j, k.

B.2.4 Vapnik-Chapelle Support Vector Span Rule

Vapnik and Chapelle define the support vector span rule to estimate LOO misclassification risk. For convenience, assume that the first n^* data points are support vectors of the SVM solution to (B.1) given λ. That is, for the first n^* data points, $y_i f_i \leq 1$, or equivalently, $\alpha_i \neq 0$.

Vapnik and Chapelle define the support vector span with respect to support vector $k_{x_p}(\cdot)$ as

$$\Lambda_{\lambda p} = \left\{ \sum_{i=1,\, i\neq p}^{n^*} \beta_i k_{x_i}(\cdot) \;:\; \sum_{i=1,\, i\neq p} \beta_i = 1,\; 0 \leq \frac{1}{2n\lambda}(\alpha_{\lambda i} + \alpha_{\lambda p} y_p \beta_i) \leq 1 \right\}$$

and give the span rule in terms of $S_{\lambda p}^2 = \min_{k_x(\cdot)\in\Lambda_{\lambda p}} \|k_{x_p}(\cdot) - k_x(\cdot)\|_{\mathcal{H}_K}$, that is, the projection of $k_{x_p}(\cdot)$ to $\Lambda_{\lambda p}$.

The span rule states that assuming the set of support vectors is unchanged during the leave-one-out procedure then

$$y_p(f_{\lambda p} - f_{\lambda p}^{[-p]}) = \frac{\alpha_{\lambda p}}{2n\lambda}S_{\lambda p}^2$$

holds for every support vector x_p. Assuming the support vectors are unchanged during the leave-one-out procedure is equivalent to removing the box constraints in the definition of $\Lambda_{\lambda p}$. Also, assuming that $d = 0$ in the SVM solution is equivalent to removing the $\sum_{i=1,\, i\neq p}\beta_i = 1$ constraint.

Proposition B.1 XA is equivalent to the Support Vector Span Rule under the following conditions:

1. The set of support vectors is unchanged during the leave-one-out procedure
2. the intercept of f_λ is zero ($d = 0$x).
3. The set of support vectors are orthogonal to each other.

Proof. First, we restate the span rule problem as a constrained optimization problem:

$$\min_{\beta} W_P(\beta) = \|k_{x_p}(\cdot) - \sum_{i=1,\, i\neq p}^{n^*} \beta_i k_{x_i}(\cdot)\|_{\mathcal{H}_K} \tag{B.16}$$

$$\text{s.t.} \quad \sum_{i=1,\, i\neq p} \beta_i = 1 \tag{B.17}$$

$$0 \leq \frac{1}{2n\lambda}(\alpha_{\lambda i} + \alpha_{\lambda p} y_p \beta_i) \leq 1 \ \forall i. \tag{B.18}$$

The Lagrangian for this problem is:

$$L(\beta, r, s, t) = K_{pp} - 2K_p'\beta + \beta' K_{\overline{p}}\beta + r(e'\beta - 1) \tag{B.19}$$

$$+ \frac{1}{2n\lambda} s'(\alpha + y_p \alpha_{\lambda p} Y\beta - e) \tag{B.20}$$

$$- \frac{1}{2n\lambda} t'(\alpha + y_p \alpha_{\lambda p} Y\beta). \tag{B.21}$$

where the kernel matrix K is now restricted to its first n^* columns and rows, that is, the kernel evaluated only for pairs of support vectors. K_p is the restricted kernel matrix's pth column, and $K_{\overline{P}}$ is the sub-matrix of K resulting of removing its pth row and column. Also, α and Y are restricted to support vectors $k_{x_i}(\cdot)$ where $i \neq p$.

The dual is then:

$$\max_{r,s,t} W_D(r, s, t) = K_{pp} - \frac{1}{4} z' K_{\overline{p}}^{-1} z + \frac{1}{2n\lambda}(s - t)'\alpha - \frac{1}{2n\lambda} s'e - r \tag{B.22}$$

$$\text{s.t} \quad r, s, t \geq 0, \tag{B.23}$$

where $z = 2K_p - re - \frac{y_p \alpha_{\lambda p}}{2n\lambda} Y(s - t)$.

We observe that $W_D(0,0,0) = K_{pp} - K_p' K_{\bar{p}}^{-1} K_p$, and we can recover the GACV approximation, K_{pp}, as an upper bound to $W_D(0,0,0)$ by assuming that x_p is orthogonal to all other support vectors, i.e., $K_p = 0$.

Since setting $r = s = t = 0$ is equivalent to solving the unconstrained span rule problem, the proposition follows. □

Consider also the mean-field approximation to leave-one-out error of Opper and Winther (2000):

$$y_p(f_{\lambda p} - f_{\lambda p}^{[-p]}) = \frac{1}{2n\lambda} \frac{\alpha_{\lambda p}}{K_{pp}^{-1}}. \tag{B.24}$$

This approximation can also be derived from the dual of the span rule problem after rewriting $\Lambda_{\lambda p}$ as

$$\Lambda_{\lambda p} = \{\sum_{i=1}^{n^*} \beta_i k_{x_i}(\cdot) \ : \ \beta_p = -1 \ \sum_{i=1}^{n^*} \beta_i = 0 \ 0 \leq \frac{1}{n\lambda}(\alpha_{\lambda i} + \alpha_{\lambda p} y_p \beta_i) \leq 1\}. \tag{B.25}$$

Rewriting the primal and dual problem under these equivalent constraints gives $W_D(0,0,0) = \frac{1}{K_{pp}^{-1}}$.

B.3 Discussion

We have analyzed various methods to select the tuning parameters in the SVM problem. In particular, we have shown that the GACV approximation of the LOO estimate of misclassification rate is equivalent to that given by the Chapelle and Vapnik Support Vector Span rule.