Large Scale Machine Learning in Biology

Anil Raj

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2011

ABSTRACT

Large Scale Machine Learning in Biology

Anil Raj

Rapid technological advances during the last two decades have led to a data-driven revolution in biology opening up a plethora of opportunities to infer informative patterns that could lead to deeper biological understanding. Large volumes of data provided by such technologies, however, are not analyzable using hypothesis-driven significance tests and other cornerstones of orthodox statistics. We present powerful tools in machine learning and statistical inference for extracting biologically informative patterns and clinically predictive models using this data.

Motivated by an existing graph partitioning framework, we first derive relationships between optimizing the regularized min-cut cost function used in spectral clustering and the relevance information as defined in the Information Bottleneck method. For fast-mixing graphs, we show that the regularized min-cut cost functions introduced by Shi and Malik over a decade ago can be well approximated as the rate of loss of predictive information about the location of random walkers on the graph. For graphs drawn from a generative model designed to describe community structure, the optimal information-theoretic partition and the optimal min-cut partition are shown to be the same with high probability.

Next, we formulate the problem of identifying emerging viral pathogens and characterizing their transmission in terms of learning linear models that can predict the host of a virus using its sequence information. Motivated by an existing framework for representing biological sequence information, we learn sparse, tree-structured models, built from decision rules based on subsequences, to predict viral hosts from protein sequence data using multi-class Adaboost, a powerful discriminative machine learning algorithm. Furthermore, the predictive motifs robustly selected by the learning algorithm are found to

show strong host-specificity and occur in highly conserved regions of the viral proteome.

We then extend this learning algorithm to the problem of predicting disease risk in humans using single nucleotide polymorphisms (SNP) — single-base pair variations — in their entire genome. While genome-wide association studies usually aim to infer individual SNPs that are strongly associated with disease, we use popular supervised learning algorithms to infer sufficiently complex tree-structured models, built from single-SNP decision rules, that are both highly predictive (for clinical goals) and facilitate biological interpretation (for basic science goals). In addition to high prediction accuracies, the models identify 'hotspots' in the genome that contain putative causal variants for the disease and also suggest combinatorial interactions that are relevant for the disease.

Finally, motivated by the insufficiency of quantifying biological interpretability in terms of model sparsity, we propose a hierarchical Bayesian model that infers hidden structured relationships between features while simultaneously regularizing the classification model using the inferred group structure. The appropriate hidden structure maximizes the log-probability of the observed data, thus regularizing a classifier while increasing its predictive accuracy. We conclude by describing different extensions of this model that can be applied to various biological problems, specifically those described in this thesis, and enumerate promising directions for future research.

Contents

List of Figures

List of Tables

Acknowledgements

It is a great pleasure to acknowledge the many individuals who have encouraged, supported, and contributed to the research presented here and, more generally, my overall education.

First, I would like to thank my advisor, Chris Wiggins, who has played an instrumental role in guiding my education and interests, and in opening up opportunities for me. Chris' endless curiosity, immense patience, and ability to stay focused and "think in equations" as a scientist and mathematician are truly inspirational; I have learned more from him than I can possibly list here. His character and sense of humor have made this experience thoroughly enjoyable.

I would like to thank my family for making any of this possible: my parents Sasikala and Rajendran, for being the most influential teachers in my life and for ensuring that I enjoyed more opportunities than they were afforded; my brother Vimal and cousin Rajeev for their unconditional support and confidence; and my uncle Radhakrishnan for encouraging my scientific curiosity and inspiring me to follow my passion for research.

I would like to thank my co-authors for their collaboration and numerous insights in our study characterizing virus host, including Mike Dewar, Gustavo Palacios and Raul Rabadan. I would also like to thank my collaborators Matan Hofree, Trey Ideker and Yoav Freund for providing the data and their excellent feedback in our study inferring genetic models for predicting disease. I am grateful to my dissertation committee, Adam Sobel, Chris Marianetti and David Madigan for their valuable time and attention in participating in my defense.

I also have the great pleasure of thanking my colleagues and friends: members of the Wiggins lab, including Mike Dewar, Jake Hofman, Andrew Mugler, Adrian Haimovich and Jonathan Bronson for the enlightening discussions, for teaching me the skills and sharing their wisdom necessary for graduate school and life after; and Yana Pleshivoy, Milena Zaharieva, Rebecca Plummer, Tatiana Pataria and Elena Norakidze for being excellent friends and a source of support and inspiration.

Anil Raj

New York, July 2011

To my family.

Chapter 1

Introduction

Biology in the twentieth-century has primarily focused on discovering molecular changes and interactions amongst them underlying various observed biological phenomena. A prime example of this can be found in genetics where investigation of heritable variation in the early twentieth-century led to the discovery of discrete, heritable, functional components called genes, the discovery of DNA as the underlying molecule that encodes genetic information and the articulation of the central dogma of molecular biology — DNA encodes for the structure and function of proteins whose synthesis and activity are regulated with the aid of intermediate molecules called RNA.

The traditional hypothesis-driven approach to this discovery process involves designing model systems that give rise to observed phenomena and making theoretical predictions using them. The key goal is to choose models complex enough for their predictions to closely match the observables, yet simple enough to generalize to future observations and have a biological interpretation that is meaningful in the context of related phenomena and within the constraints of evolution. This process of discovery, however, has mostly been successful in assigning functional relevance to individual molecules or small collections of molecules. Analyzing more complex systems involving molecular interactions and signaling pathways has been painstakingly slow, requiring numerous experiments to test the profusion of possible models governing such systems. Difficulties in positing meaningful molecular models and the lack of technological sophistication to compute and observe quantities of interest made understanding complex phenomena like viral infection,

pathogen evolution and tumorigenesis incredibly time-consuming.

Rapid technological advances during the last decade of the twentieth century have led to a data-driven revolution in molecular biology opening up a plethora of opportunities to infer functional, informative patterns that could lead to deeper biological understanding and facilitate medical innovation. Examples of such innovations include shotgun sequencing, DNA microarrays and chromatin immunoprecipitation. At this point, bench biologists and computational biologists agree that such technologies which completely transformed biology in the last decade, provide data which are not analyzable using statistics of the prior era. Case control studies with p-values and other cornerstones of orthodox statistics simply are not the appropriate high-dimensional statistical approaches to help biologists reveal, e.g., the sequence elements which control transcriptional regulation or the wirings of transcriptional regulatory networks. These advances have helped turn the traditional approach over its head, inspiring data-driven modeling — the use of massive quantities of data and powerful tools in machine learning and statistical inference to extract biologically informative patterns, generate relevant hypotheses and infer properties of complex systems.

Over the last two decades, high throughput experiments have produced large quantities of structured, yet unlabeled data across a variety of complex biological systems. Examples include the expression of thousands of genes in different human tissues and relational networks quantifying regulatory and physical interactions between genes and the proteins they encode in different single and multicellular organisms. The qualitative goal of unsupervised machine learning is to infer meaningful patterns and hidden structure in such unlabeled data; however, it is often unclear how one can quantify this goal in terms of an appropriate cost function to be optimized, making model evaluation a difficult task. For instance, the problem of inferring protein clusters in a protein-protein interaction network has been addressed using a variety of tools including spectral graph partitioning, Bayesian inference and information theoretic methods, each approach optimizing seemingly different cost functions. In Chapter 2, we will review two major approaches used in extracting clusters of nodes based on the topology of a network — spectral graph partitioning and Information Bottleneck — and show how the cost functions being optimized in each method

are approximately equal for fast-mixing networks.

Supervised machine learning provides a well-posed, principled framework for inferring models that are predictive of some observable of interest, given labeled examples. For instance, given the genome sequence of normal and tumor cells, supervised learning infers a discriminative model that can predict whether a newly observed cell is normal or cancerous based on its genetic sequence. The central goal of supervised learning is to quantify the 'goodness' of a model in terms of a cost function to be optimized, where the cost function includes a trade-off between model accuracy and model complexity. Having specified a biologically relevant cost function, powerful tools in convex optimization are often used to optimize these cost functions. The optimal trade-off between accuracy and complexity is chosen based on the ability of the learned model to generalize well to unobserved data.

The overall goal of supervised learning is to infer a model whose predictions on training data correlate well with their known labels or observables of interest. The accuracy of such a model is typically quantified in terms of a loss function, the most natural loss function being the difference between the predicted value and true value of the observable. In the case of binary labeled data, a natural loss function is the number of mistakes made by the model on the training data. Loss functions surrogate to this classification loss, however, are typically used since they are more amenable to mathematical analysis. For example, when trying to fit a polynomial to some observed data, one useful loss function to minimize is the squared difference between the predicted and true values of the observed quantity.

In addition to accurate predictions, applications in biology demand models that are simple and facilitate biological interpretation. Simplicity from an information theoretic perspective is often quantified by the number of variables in the model or the number of bits required to encode the model. In statistical inference, simplicity is typically quantified by the average magnitude of the coefficients of variables in the model. In the previous example, these notions of simplicity translate to the order of the polynomial and the sum of squares of the polynomial coefficients, respectively. However, since complex systems in biology are usually characterized by strong correlations and redundant subsystems, it is not entirely clear if simplicity renders a model biologically interpretable. A more meaning-

ful notion of model simplicity is quantified by mathematical functions that encode hidden structure (e.g., combinatorial interactions or functional groups) among the variables in the model.

In Chapter 3, we use a powerful machine learning algorithm to learn sufficiently complex tree-structured models that predict the host of a virus, built from simple decision rules based on the amino acid sequence of viral proteins. Identifying the host of an emerging virus and understanding what molecular changes in the virus facilitated human infection is an important first step towards restricting viral transmission during epidemics and developing appropriate vaccines. These key questions have typically been addressed using phylogenetics and other techniques based on sequence similarity. Lacking from these techniques, however, is the ability to identify host-specific motifs that can allow us to understand the essential functional changes that enabled the virus to infect a new host. Our results in chapter 3 demonstrate that the models inferred from protein sequence data of well-characterized viruses have host prediction accuracy comparable to phylogenetics, while robustly identifying protein subsequences that are strongly conserved among viruses that share a host type. These conserved protein subsequences can then offer us some insight into the necessary mutations that enabled the virus to infect a new host and into the biology of viral infection.

In Chapter 4, we extend this learning algorithm to infer a similar model that predicts disease phenotype of an individual based on variations in their entire genome. Genome-wide association studies (GWAS) aim to infer genetic variants from hundreds of thousands of whole genome sequence variants that are strongly associated with a phenotype of interest. Developing the appropriate high dimensional statistical framework to address this problem — one that is both predictive (for clinical goals) and interpretable (for basic science goals) — presents a deep machine learning challenge. Though molecular biologists have been open to machine learning approaches to answer fundamental biological questions, genetics remains more firmly entrenched in low-dimensional or one-dimensional statistical tools, which do little to help us escape the multiple-hypothesis nightmare inherent in such problem settings; this persists despite the fact that clinicians widely recognize the insufficiency of existing statistical approaches. Our results in chapter 4 demonstrate

that additive models based on simple decision rules inferred directly on measurements of sequence variation achieve accuracies significantly higher than predictive models learned using statistical tools popular in GWAS. In addition, the learned models identify 'hotspots' in the genome that contain putative causal variants for the disease and also suggest intra-locus (dominance) and inter-locus (epistatic) interactions that are relevant for the disease phenotype.

In Chapter 5, motivated by the insufficiency of quantifying biological interpretability in terms of model sparsity (enumerated in Chapters 3 and 4), we revisit the problem of regularizing loss functions using penalty terms that encode structured relationships between the model variables. We describe various approaches in the machine learning literature that aim to do this and argue for a more unified learning framework that infers hidden structured relationships between model variables whilst appropriately penalizing the loss function. We pose this problem within the framework of Bayesian inference and demonstrate one example of a classification model that automatically infers hidden group structure among features. We conclude this chapter by describing different extensions of this model that can be applied to various biological problems, specifically those described in Chapter 3 and 4, and enumerate promising directions for future research.

Chapter 2

Information theoretic derivation of min-cut based clustering

2.1 Context

Over the last two decades, rapid advancement in DNA microarray technologies has led to an explosion of massive volumes of noisy expression data, quantifying rate of production of mRNA, for several tens of thousands of genes across different cell types and cellular environments in a variety of organisms. These high-throughput technologies have facilitated the parallelization of experiments such as gene-knockouts and cellular stress response allowing biologists to construct maps of which genes regulate (and co-express with) which other genes. Simultaneously, very high-throughput binding assays facilitated the querying of several thousands of putative protein-DNA bindings (e.g., ChIP-chip experiments) and the construction of whole organism protein-protein interaction networks (e.g., the yeast two hybrid experiments). The sheer size of the networks built from these gene regulatory and protein interaction data demanded fast, efficient algorithmic approaches to model and reveal biologically informative patterns in these graphs.

Following the qualitative definition of a *module* [Hartwell *et al.*, 1999] as "a discrete entity whose function is separable from those of other modules", there has been a plethora of research aimed at modeling biological networks as a collection of functionally autonomous modules. Most of this research has focused on quantifying this notion of *biological modu-*

larity in terms of network *topological modularity*, leading to a variety of representative cost functions, along with an ever increasing number of algorithms for optimizing these various cost functions. These approaches hinge on the key assumption that topological modules inferred only from gene regulatory or protein interaction data would serve as useful proxies for biological functional modules, facilitating biological interpretation.

On the general problem of partitioning a graph into modules, one particularly strong thread of literature can be found in the social sciences. Based on the Stochastic Block Model (SBM) [Holland and Leinhardt, 1976], one of the earliest models of community structure in social graphs, there have been several papers [Newman and Girvan, 2004] [Newman, 2006] focused on computing clusters of nodes in a graph where pairs of nodes within a cluster have a higher probability of having an edge between them than pairs in two clusters. In this line of work, the 'goodness' of a partition of a graph was quantified by different cost functions comparing the observed within-cluster connectivity against the expected connectivity that would be observed under some appropriate null distribution of graphs. More recently, there have been several attempts at revisiting the SBM as a probabilistic model for generating graphs and using it to infer the latent group assignments of nodes, given the adjacency matrix of a network as data. Specifically, given an adjacency matrix $\mathbf{A}$ (defined below) of a network, the inferred distribution over hidden group assignments was computed by maximizing a lower bound on the evidence of the data $p(\mathbf{A}|K)$, where the SBM model parameters have been integrated out and K quantifies the complexity of the SBM (i.e., number of clusters). Model selection – the right choice for K – was performed during inference [Hofman and Wiggins, 2008] or predetermined using Bayesian information criterion (BIC) [Airoldi *et al.*, 2008] or minimum description length (MDL) [Rosvall and Bergstrom, 2007].

Min-cut based spectral graph partitioning has been used successfully to find clusters in networks, with applications predominantly in image segmentation as well as clustering biological and sociological networks. The central idea is to develop fast and efficient algorithms that optimally cut the edges between graph nodes, resulting in a separation of graph nodes into a pre-specified number of clusters. As shown by Czech mathematician Miroslav Fiedler, the cut of a partition of a graph can be related to a cost function that de-

pends on the *graph Laplacian* [Fiedler, 1973], a second order finite-difference discretization of the continuous Laplacian operator on a graph lattice.

Specifically, given a undirected, unweighted graph $\mathscr{G}$ represented by an adjacency matrix $\mathbf{A} := \{A_{xy} = 1 \iff \text{node } x \text{ is adjacent to } y\}$, we define its positive semi-definite Laplacian as $\mathbf{\Delta} = \mathrm{diag}(\mathbf{d}) - \mathbf{A}$, where $\mathbf{d}$ is a vector of vertex degrees and $\mathrm{diag}(\cdot)$ is a diagonal matrix with its argument on the diagonal. For any general vector $\mathbf{f}$ over the graph nodes, we have

$$\begin{aligned}
\mathbf{f}^{\mathrm{T}}\mathbf{\Delta}\mathbf{f} &= \mathbf{f}^{\mathrm{T}}\mathrm{diag}(\mathbf{d})\mathbf{f} - \mathbf{f}^{\mathrm{T}}\mathbf{A}\mathbf{f} \\
&= \sum_x d_x f_x^2 - \sum_{x,y} f_x f_y A_{xy} \\
&= \sum_x \left(\sum_y A_{xy} \right) f_x^2 - \sum_{x,y} f_x f_y A_{xy} \\
&= \frac{1}{2} \left(\sum_{x,y} f_x^2 A_{xy} - 2 \sum_{x,y} f_x f_y A_{xy} + \sum_{x,y} f_y^2 A_{xy} \right) \\
&= \frac{1}{2} \sum_{x,y} A_{xy} \left(f_x - f_y \right)^2 . \qquad (2.1)
\end{aligned}$$

Note that we use node variables x and y to index any vector or matrix associated with a graph, to make explicit the association between their rows (or columns) and the nodes of the graph. Also, in the rest of this thesis, summation over an index (or variable) runs over the entire relevant set, unless otherwise mentioned.

If $\mathbf{f}$ represents the cluster assignment of nodes for a 2-clustering, $\mathbf{f} = \mathbf{h}$, with $h_x \in \{-1, 1\}$, we have

$$\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\mathbf{h} = \frac{1}{2} \sum_{h_x h_y = -1} 4A_{xy} = 4 \times c. \qquad (2.2)$$

A direct minimization of this cost function over all vectors $\mathbf{h}$, however, is a combinatorially hard problem. This was resolved by relaxing the constraints on the optimization variable, allowing minimization over real-valued vectors $\mathbf{f} : f_x \in \mathbb{R}$. Under this relaxation, the problem can now be posed as computing the eigenvector of the graph Laplacian corresponding to its second smallest eigenvalue – Fiedler vector[1]. *Spectral graph partitioning*

[1]The smallest eigenvalue of the graph Laplacian is 0 with the corresponding eigenvector being the vector of all 1s.

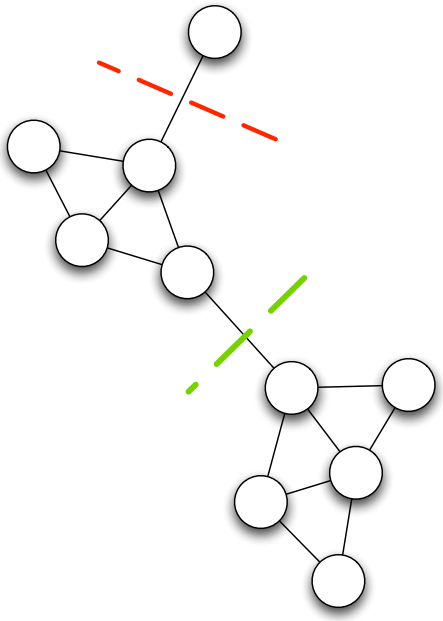

Figure 2.1: Minimizing the cut can lead to undesirable solutions as shown by the red dashed line. A more balanced solution to the min-cut problem, shown by the green dashed line, can be obtained by minimizing the regularized cut.

minimizes the cut by computing the eigen spectrum of the graph Laplacian; a 2-partition of the graph is then constructed by assigning nodes corresponding to elements of the Fiedler vector with the same sign into the same cluster.

Simply minimizing the cut, however, can result in mathematically valid, yet undesirable, partitions of the graph as shown in figure 2.1. To avoid such unbalanced solutions, Shi and Malik [Shi and Malik, 2000] proposed a set of regularizations of the cut: the *average* cut and the *normalized* cut (see equations 2.5 and 2.6). They successfully showed that these regularized cut-based cost functions were useful heuristics to be optimized to segment images into spatially colocated groups of pixels with similar intensities. Following this success, there has been tremendous research in the image segmentation community both showing the success of these cost functions, and in constructing better regularized cut-based cost functions and more efficient algorithms for optimizing these cost functions for various applications.

Despite the wide-spread empirical success of spectral clustering in the graph partition-

ing and image segmentation communities over the last decade, it is still unclear if these heuristics can be derived from a more general principle facilitating generalization to new problem settings. Several insightful works have focused on providing an interpretation and a justification for min-cut based clustering, within the framework of graph diffusion. Meila and Shi [Meila and Shi, 2001] showed rigorous connections between normalized min-cut based clustering and the lumpability of the Markov chains underlying the corresponding discrete-diffusion operator. More recently, Lafon and Lee [Lafon and Lee, 2006] and Nadler et al. [Nadler *et al.*, 2005] showed the close relationship between the problem of spectral clustering and that of learning locality-preserving embeddings of data, using diffusion maps.

The Information Bottleneck (IB) method [Slonim, 2002] is a clustering technique, based on rate-distortion theory [Shannon, 2001], that has been successfully applied in a wide variety of contexts including clustering word documents and gene expression profiles. The network information bottleneck (NIB) [Ziv *et al.*, 2005] algorithm is a variation of the IB method for discovering modules in a network, given the diffusive probability distribution over the network, and has been used successfully for discovering modules in synthetic and natural networks. Additionally, the NIB algorithm also computes a normalized, dimensionless measure of network modularity that quantifies the degree to which a network can be compressed without significant loss of information about some relevant variable of interest.

Specifically, given the probability distribution of the position of a random walker on the graph conditioned on its starting node, the NIB algorithm iteratively combines nodes (or groups of nodes) that are similar to each other, where similarity is measured by the Jensen-Shannon Divergence (JSD) between the conditional probability distributions associated with the nodes. For large graphs, a naive implementation of this algorithm, however, introduces numerical errors in the computation of the JSD when the probability distributions are very similar, leading to errors in the choice of nodes being grouped together.

Here, we derive a non-negative series expansion for the JSD between two probability distributions. This approximation avoids incurring numerical errors in the JSD when probability distributions are nearly equal, facilitating the application of the NIB algorithm

to very large biological networks. We also illustrate how minimizing the two cut-based heuristics introduced by Shi and Malik can be well-approximated by the rate of loss of *relevance information*, defined in the IB method applied to clustering graphs. To establish these relations, we must first define the graphs to be partitioned; we assume hard-clustering and the cluster cardinality to be K. We show, numerically, that maximizing mutual information and minimizing regularized cut amount to the same partition with high probability, for more modular 32-node graphs, where modularity is defined by the probability of inter-cluster edge connections in the SBM for graphs. We also show that the optimization goal of maximizing relevance information is equivalent to minimizing the regularized cut for 16-node graphs.[2]

2.2 Min-cut problem: formalized

For an undirected, unweighted graph $\mathscr{G} = (\mathrm{V}, \mathrm{E})$ with N nodes and M edges, represented by its adjacency matrix $\mathbf{A}$, we define for two not necessarily disjoint sets of nodes $\mathrm{V}_+, \mathrm{V}_- \subseteq \mathrm{V}$, the association [von Luxburg, 2007]

$$W(\mathrm{V}_+, \mathrm{V}_-) = \sum_{x \in \mathrm{V}_+, y \in \mathrm{V}_-} A_{xy}. \tag{2.3}$$

We define a bisection of V into $\mathrm{V}_\pm$ if $\mathrm{V}_+ \cup \mathrm{V}_- = \mathrm{V}$ and $\mathrm{V}_+ \cap \mathrm{V}_- = \emptyset$. For a bisection of V into V_+ and V_-, the 'cut' is defined as $c = W(\mathrm{V}_+, \mathrm{V}_-)$. We also quantify the size of a set $\mathrm{V}_+ \subseteq \mathrm{V}$ in terms of the number of nodes in the set V_+ or the number of edges with at least one node in the set V_+:

$$\begin{aligned} \omega(\mathrm{V}_+) &= \sum_{x \in \mathrm{V}_+} 1 \\ \Omega(\mathrm{V}_+) &= \sum_{x \in \mathrm{V}_+} d_x, \end{aligned} \tag{2.4}$$

where d_x is the degree of node x.

Shi and Malik [Shi and Malik, 2000] defined a pair of regularized cuts, for a bisection of V into V_+ and V_-; the *average cut* was defined as

$$\mathcal{A} = \frac{W(\mathrm{V}_+, \mathrm{V}_-)}{\omega(\mathrm{V}_+)} + \frac{W(\mathrm{V}_+, \mathrm{V}_-)}{\omega(\mathrm{V}_-)} \tag{2.5}$$

[2] We chose 16-node graphs so the network and its partitions could be parsed visually with ease.

and the *normalized cut* was defined as

$$\mathcal{N} = \frac{W(\mathrm{V}_+, \mathrm{V}_-)}{\Omega(\mathrm{V}_+)} + \frac{W(\mathrm{V}_+, \mathrm{V}_-)}{\Omega(\mathrm{V}_-)}. \tag{2.6}$$

For a K-partition of V into $\mathrm{V}_1, \mathrm{V}_2, \ldots, \mathrm{V}_K$, this definition can be generalized as

$$\mathcal{A} = \sum_k \frac{W(\mathrm{V}_k, \overline{\mathrm{V}}_k)}{\omega(\mathrm{V}_k)} \tag{2.7}$$

$$\mathcal{N} = \sum_k \frac{W(\mathrm{V}_k, \overline{\mathrm{V}}_k)}{\Omega(\mathrm{V}_k)} \tag{2.8}$$

where $\overline{\mathrm{V}}_k = \mathrm{V} \setminus \mathrm{V}_k$.

For a bisection of V, we also define the partition indicator vector $\mathbf{h}$

$$h_x = \begin{cases} +1 & \forall x \in \mathrm{V}_+ \\ -1 & \forall x \in \mathrm{V}_-. \end{cases} \tag{2.9}$$

Specifying two 'prior' probability distributions over the set of nodes V : (i) $p(x) \propto 1$ and (ii) $p(x) \propto d_x$, we then define the *average* of $\mathbf{h}$ to be

$$\begin{aligned} \overline{\mathbf{h}} &= \frac{\sum_x h_x}{N} \\ \langle \mathbf{h} \rangle &= \frac{\sum_x d_x h_x}{2M}. \end{aligned} \tag{2.10}$$

Using the definitions of the average and normalized cuts, we have

$$\begin{aligned} \mathcal{A} &= c \times \left(\frac{1}{\sum_{x:h_x=+1} 1} + \frac{1}{\sum_{x:h_x=-1} 1} \right) \\ &= c \times \left(\frac{1}{\sum_x \left(\frac{1+h_x}{2}\right)} + \frac{1}{\sum_x \left(\frac{1-h_x}{2}\right)} \right) \\ &= 2c \times \left(\frac{\sum_x (1 - h_x + 1 + h_x)}{\sum_x (1 + h_x) \sum_x (1 - h_x)} \right) \\ &= 2c \times \left(\frac{2}{N(1+\overline{\mathbf{h}})(1-\overline{\mathbf{h}})} \right) = \frac{4}{N} \frac{c}{1 - \overline{\mathbf{h}}^2}. \end{aligned} \tag{2.11}$$

$$\begin{aligned}
\mathcal{N} &= c \times \left(\frac{1}{\sum_{x:h_x=+1} d_x} + \frac{1}{\sum_{x:h_x=-1} d_x} \right) \\
&= c \times \left(\frac{1}{\sum_x d_x \left(\frac{1+h_x}{2}\right)} + \frac{1}{\sum_x d_x \left(\frac{1-h_x}{2}\right)} \right) \\
&= 2c \times \left(\frac{\sum_x d_x (1 - h_x + 1 + h_x)}{\sum_x \left(d_x(1+h_x)\right) \sum_x \left(d_x(1-h_x)\right)} \right) \\
&= 2c \times \left(\frac{1}{M(1+\langle \mathbf{h} \rangle)(1-\langle \mathbf{h} \rangle)} \right) = \frac{2}{M} \frac{c}{1-\langle \mathbf{h} \rangle^2}.
\end{aligned} \tag{2.12}$$

More generally, for a K-partition, we define the partition indicator matrix $\mathbf{Q}$ as

$$Q_{zx} \equiv p(z|x) = \quad 1 \quad \forall x \in \mathrm{V}_z \tag{2.13}$$

where $z \in \{1, 2, ..., K\}$ and define $\mathbf{P}$ as the diagonal matrix of the 'prior' probability distribution over nodes. The regularized cut can then be generalized as

$$\mathcal{C} = \sum_k \frac{[\mathbf{Q}\mathbf{\Delta}\mathbf{Q}^{\mathrm{T}}]_{kk}}{[\mathbf{Q}\mathbf{P}\mathbf{Q}^{\mathrm{T}}]_{kk}} \tag{2.14}$$

where for $p(x) \propto 1$, $\mathcal{C} = \mathcal{A}$, and for $p(x) \propto d_x$, $\mathcal{C} = \mathcal{N}$.

Inferring the optimal $\mathbf{h}$ (or $\mathbf{Q}$), however, has been shown to be an NP-hard combinatorial optimization problem [Wagner and Wagner, 1993].

2.3 Information bottleneck

Rate-distortion theory, which provides the foundations for lossy data compression, formulates clustering in terms of a compression problem; it determines the code with minimum average length such that information can be transmitted without exceeding some specified distortion. Here, the model-complexity, or *rate*, is measured by the mutual information between the data and their representative codewords (average number of bits used to store a data point). Simpler models correspond to smaller rates but typically suffer from relatively high *distortion*. The distortion measure, which can be identified with loss functions, usually depends on the problem; in the simplest of cases, it is the variance of the difference between an example and its cluster representative.

The Information Bottleneck (IB) method [Tishby and Slonim, 2000] [Tishby *et al.*, 2000] proposes the use of mutual information as a natural distortion measure. In this method, the data are compressed into clusters while maximizing the amount of information that the 'cluster representation' preserves about some specified *relevance* variable. For example, in clustering word documents, one could use the 'topic' of a document as the relevance variable; in the case of protein sequences, the protein fold could be the relevance variable.

For a graph $\mathscr{G}$, let x be a random variable over graph nodes, y be the relevance variable and z be the random variable over clusters. Graph partitioning using the IB method [Ziv *et al.*, 2005] learns a probabilistic cluster assignment function $p(z|x)$ which gives the probability that a given node x belongs to cluster z. The optimal $p(z|x)$ minimizes the mutual information between x and z, while minimizing the loss of predictive information between z and y. This complexity–fidelity trade-off can be expressed in terms of a functional to be minimized

$$\mathscr{F}[p(z|x)] = -\mathbf{I}[y;z] + T\mathbf{I}[x;z] \tag{2.15}$$

where the temperature T parameterizes the relative importance of precision over complexity. As $T \to 0$, we reach the 'hard clustering' limit where each node is assigned with unit probability to one cluster (i.e $p(z|x) \in \{0,1\}$). In the case where the number of clusters equals the number of nodes, we get back the trivial solution where the clusters z are just a copy of the nodes x.

Graph clustering, as formulated in terms of the IB method, requires a joint distribution $p(y,x)$ to be defined on the graph. Given only the adjacency matrix of the graph, a natural choice of distribution is one given by continuous-time graph diffusion as it naturally captures topological information about the network [Ziv *et al.*, 2005]. The relevance variable y then ranges over the nodes of the graph and is defined as the node at which a random walker ends at time t if the random walker starts at node x at time 0. For continuous-time diffusion, the conditional distribution $p(y|x)$ is given as

$$G^t_{yx} \equiv p(y|x) = \left[e^{-t\mathbf{\Delta}\mathbf{P}^{-1}}\right]_{yx} \tag{2.16}$$

where $\mathbf{\Delta}$ is the positive semi-definite graph Laplacian and $\mathbf{P}$ is a diagonal matrix of the prior distribution over the graph nodes, as described earlier. Note that the diagonal matrix

$\mathbf{P}$ can be any prior distribution over the graph nodes. The characteristic diffusion time scale τ of the system is given by the inverse of the smallest non-zero eigenvalue (Fiedler value) of the diffusion operator exponent $\mathbf{\Delta P}^{-1}$ and characterizes the slowest decaying mode in the system.

To calculate the joint distribution $p(y, x)$ from the conditional $\mathbf{G}^t$, we must specify an initial or prior distribution[3]; we use the two different priors $p(x)$, used in equation 2.10 to calculate $\overline{\mathbf{h}}$ and $\langle \mathbf{h} \rangle$: (i) $p(x) \propto 1$ and (ii) $p(x) \propto d_x$. For the remainder of this chapter, time dependence needs to be considered only when the conditional distribution $p(y|x)$ is replaced by the diffusion Green's function $\mathbf{G}$; thus, time dependence will be explicitly denoted only once $\mathbf{G}$ is invoked.

Given $p(y|x)$, the agglomerative IB algorithm optimizes equation 2.15 by iteratively combining nodes whose conditional distributions are similar to each other, where similar distributions have low Jensen-Shannon Divergence between them. The conditional distribution of this compressed representation is the weighted average of the distributions of the nodes being combined. For large graphs, a naive implementation of this algorithm, however, introduces numerical errors in the computation of the JSD when the probability distributions are very similar; precisely in the range where errors in the choice of nodes being grouped together can lead to drastically different compressions of the network. In the next section, we derive a non-negative series expansion for the JSD between two probability distributions that helps resolve such numerical errors.

2.4 Jensen-Shannon divergence : revisited

The Jensen-Shannon divergence (JSD) has been widely used as a dissimilarity measure between weighted probability distributions. The direct numerical evaluation of the exact expression for the JSD (involving difference of logarithms), however, leads to numerical errors when the distributions are close to each other (small JSD). When the elementwise

[3]Strictly speaking, any diagonal matrix $\mathbf{P}$ that we specify determines the steady-state distribution. Since we are modeling the distribution of random walkers at statistical equilibrium, we always use this distribution as our initial or prior distribution.

relative difference between the distributions is $\mathrm{O}(10^{-1})$, this naive formula produces erroneous values (sometimes negative) when used for numerical calculations. To resolve such issues, we derive a provably non-negative series expansion for the JSD which can be used in the small JSD limit, where the naive formula fails.

Consider two discrete probability distributions $\mathbf{p}_1$ and $\mathbf{p}_2$ over a sample space S of cardinality N with relative normalized weights π_1 and π_2 between them. The JSD between the distributions is then defined as [Lin, 1991]

$$\Lambda_{naive}[\mathbf{p}_1, \mathbf{p}_2; \pi_1, \pi_2] = \mathbf{H}[\pi_1\mathbf{p}_1 + \pi_2\mathbf{p}_2] - (\pi_1\mathbf{H}[\mathbf{p}_1] + \pi_2\mathbf{H}[\mathbf{p}_2]) \tag{2.17}$$

where the entropy (measured in nats) of a probability distribution is defined as

$$\mathbf{H}[\mathbf{p}] = -\sum_n h(p_n) = -\sum_n p_n \log(p_n). \tag{2.18}$$

Defining

$$\begin{array}{rcccc} \overline{p}_n & = & \frac{1}{2}(p_{1n} + p_{2n}) & ; & 0 \leq \overline{p}_n \leq 1,\ \sum_n \overline{p}_n = 1 \\ \eta_n & = & \frac{1}{2}(p_{1n} - p_{2n}) & ; & \sum_n \eta_n = 0 \\ \varepsilon_n & = & \eta_n/\overline{p}_n & ; & -1 \leq \varepsilon_n \leq 1 \\ \alpha & = & \pi_1 - \pi_2 & ; & -1 \leq \alpha \leq 1 \end{array} \tag{2.19}$$

we have

$$\begin{aligned} h(\pi_1 p_{1n} + \pi_2 p_{2n}) &= -(\pi_1(\overline{p}_n + \eta_n) + \pi_2(\overline{p}_n - \eta_n)) \log(\pi_1(\overline{p}_n + \eta_n) + \pi_2(\overline{p}_n - \eta_n)) \\ &= -\overline{p}_n(1 + \alpha\varepsilon_n)\left(\log(\overline{p}_n) + \log(1 + \alpha\varepsilon_n)\right) \end{aligned} \tag{2.20}$$

and

$$\begin{aligned} \pi_1 h(p_{1n}) + \pi_2 h(p_{2n}) &= -\pi_1(\overline{p}_n + \eta_n)\log(\overline{p}_n + \eta_n) - \pi_2(\overline{p}_n - \eta_n)\log(\overline{p}_n - \eta_n) \\ &= -\frac{1}{2}\overline{p}_n(1+\alpha)(1+\varepsilon_n)\log(\overline{p}_n(1+\varepsilon_n)) \\ &\quad -\frac{1}{2}\overline{p}_n(1-\alpha)(1-\varepsilon_n)\log(\overline{p}_n(1-\varepsilon_n)) \\ &= -\overline{p}_n(1+\alpha\varepsilon_n)\log(\overline{p}_n) - \frac{1}{2}\overline{p}_n(1+\alpha\varepsilon_n)\log(1-\varepsilon_n^2) \\ &\quad -\frac{1}{2}\overline{p}_n(\alpha+\varepsilon_n)\log\left(\frac{1+\varepsilon_n}{1-\varepsilon_n}\right). \end{aligned} \tag{2.21}$$

Thus,

$$\begin{aligned} & h(\pi_1 p_{1n} + \pi_2 p_{2n}) - (\pi_1 h(p_{1n}) + \pi_2 h(p_{2n})) \\ = \quad & \frac{1}{2}\overline{p}_n \left\{ (1+\alpha\varepsilon_n) \log\left(\frac{1-\varepsilon_n^2}{(1+\alpha\varepsilon_n)^2} \right) + (\alpha+\varepsilon_n) \log\left(\frac{1+\varepsilon_n}{1-\varepsilon_n} \right) \right\}. \end{aligned} \tag{2.22}$$

The Taylor series expansion of the logarithm function is given as

$$\log(1+x) = \sum_{i=1}^{\infty} c_i x^i; \quad c_i = \frac{(-1)^{i+1}}{i}. \tag{2.23}$$

The logarithms in the expression for the JSD can then be written as

$$\begin{aligned} \log(1+\varepsilon_n) &= \sum_{i=1}^{\infty} c_i \varepsilon_n^i \\ \log(1-\varepsilon_n) &= \sum_{i=1}^{\infty} (-1)^i c_i \varepsilon_n^i \\ \log(1+\alpha\varepsilon_n) &= \sum_{i=1}^{\infty} c_i \alpha^i \varepsilon_n^i. \end{aligned} \tag{2.24}$$

We then have $\Lambda = \frac{1}{2}\sum_n \overline{p}_n \delta_n$, with

$$\begin{aligned} \delta_n &= (1+\alpha\varepsilon_n)\left\{\log(1+\varepsilon_n) + \log(1-\varepsilon_n) - 2\log(1+\alpha\varepsilon_n)\right\} \\ &\quad + (\alpha+\varepsilon_n)\left\{\log(1+\varepsilon_n) - \log(1-\varepsilon_n)\right\} \\ &= (1+\alpha\varepsilon_n)\left\{ \sum_{i=1}^{\infty} c_i \varepsilon_n^i + \sum_{i=1}^{\infty} (-1)^i c_i \varepsilon_n^i - 2\sum_{i=1}^{\infty} c_i \alpha^i \varepsilon_n^i \right\} \\ &\quad + (\alpha+\varepsilon_n)\left\{ \sum_{i=1}^{\infty} c_i \varepsilon_n^i - \sum_{i=1}^{\infty} (-1)^i c_i \varepsilon_n^i \right\} \\ &= \sum_{i=1}^{\infty} c_i \left\{ \varepsilon_n^i + \alpha\varepsilon_n^{i+1} + (-1)^i \varepsilon_n^i + (-1)^i \alpha \varepsilon_n^{i+1} - 2\alpha^i \varepsilon_n^i - 2\alpha^{i+1}\varepsilon_n^{i+1} \right. \\ &\quad \left. + \ \alpha\varepsilon_n^i + \varepsilon_n^{i+1} + (-1)^{i+1}\alpha\varepsilon_n^i + (-1)^{i+1}\varepsilon_n^{i+1} \right\} \\ &= \sum_{i=1}^{\infty} c_i \left\{ \left((-1)^i - 2\alpha^i + \alpha + (-1)^{i+1}\alpha + 1 \right) \varepsilon_n^i \right. \\ &\quad \left. + \ \left((-1)^i \alpha - 2\alpha^{i+1} + 1 + (-1)^{i+1} + \alpha \right) \varepsilon_n^{i+1} \right\}. \end{aligned} \tag{2.25}$$

When $i = 1$, $\text{coeff}(\varepsilon_n) = c_1(-1 - 2\alpha + \alpha + \alpha + 1) = 0$. The first non-vanishing term in

the expansion is then of order 2. Shifting indices of the first term in equation 2.25 gives

$$\begin{aligned}
\delta_n &= \sum_{i=1}^{\infty} \left\{ c_{i+1} \left((-1)^{i+1} - 2\alpha^{i+1} + \alpha + (-1)^{i+2}\alpha + 1 \right) \right. \\
&\quad \left. + \; c_i \left((-1)^i \alpha - 2\alpha^{i+1} + 1 + (-1)^{i+1} + \alpha \right) \right\} \varepsilon_n^{i+1} \\
&= \sum_{i=1}^{\infty} (c_i + c_{i+1}) \left((-1)^i \alpha - 2\alpha^{i+1} + \alpha + 1 + (-1)^{i+1} \right) \right) \varepsilon_n^{i+1} \\
&= \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i(i+1)} \left((-1)^i \alpha - 2\alpha^{i+1} + \alpha + 1 + (-1)^{i+1} \right) \right) \varepsilon_n^{i+1} \\
&= \sum_{i=1}^{\infty} B_i \varepsilon_n^{i+1} \qquad (2.26)
\end{aligned}$$

where

$$\begin{aligned}
B_i &= \frac{1 - \alpha + (-1)^{i+1}(1 + \alpha - 2\alpha^{i+1})}{i(i+1)} \\
&= \begin{cases} 2(1 - \alpha^{i+1}) / \left(i(i+1) \right) & i \text{ odd}, \\ -2(\alpha - \alpha^{i+1}) / \left(i(i+1) \right) & i \text{ even}. \end{cases} \qquad (2.27)
\end{aligned}$$

This series expansion can be further simplified as

$$\begin{aligned}
\delta_n &= \sum_{i=1}^{\infty} \left(B_{2i-1} + B_{2i}\varepsilon_n \right) \varepsilon_n^{2i} \\
&= \sum_{i=1}^{\infty} B_{2i-1} \left(1 + \frac{B_{2i}}{B_{2i-1}} \varepsilon_n \right) \varepsilon_n^{2i}, \qquad (2.28) \\
\frac{B_{2i}}{B_{2i-1}} \varepsilon_n &= -\left(\frac{2i-1}{2i+1} \right) \alpha \varepsilon_n. \qquad (2.29)
\end{aligned}$$

Since $-1 \leq \alpha\varepsilon_n \leq 1$, we have $-1 \leq \frac{B_{2i}}{B_{2i-1}}\varepsilon_n \leq 1$. Thus, for every i, $(B_{2i-1} + B_{2i}\varepsilon_n)\varepsilon_n^{2i} \geqslant 0$, making δ_n — and the series expansion for Λ_{naive} — non-negative up to all orders.

2.4.1 Numerical Results

The accuracy of the truncated series expansion can be compared with the naive formula by measuring the JSD between randomly generated probability distributions. Pairs of probability distributions with $-4 \leq \log_{10} \|\boldsymbol{\varepsilon}\| < 0$, where $\|\boldsymbol{\varepsilon}\| = \sqrt{(\sum_n \varepsilon_n^2)/N}$, were randomly generated and the JSD between each pair was calculated by both a direct evaluation of the

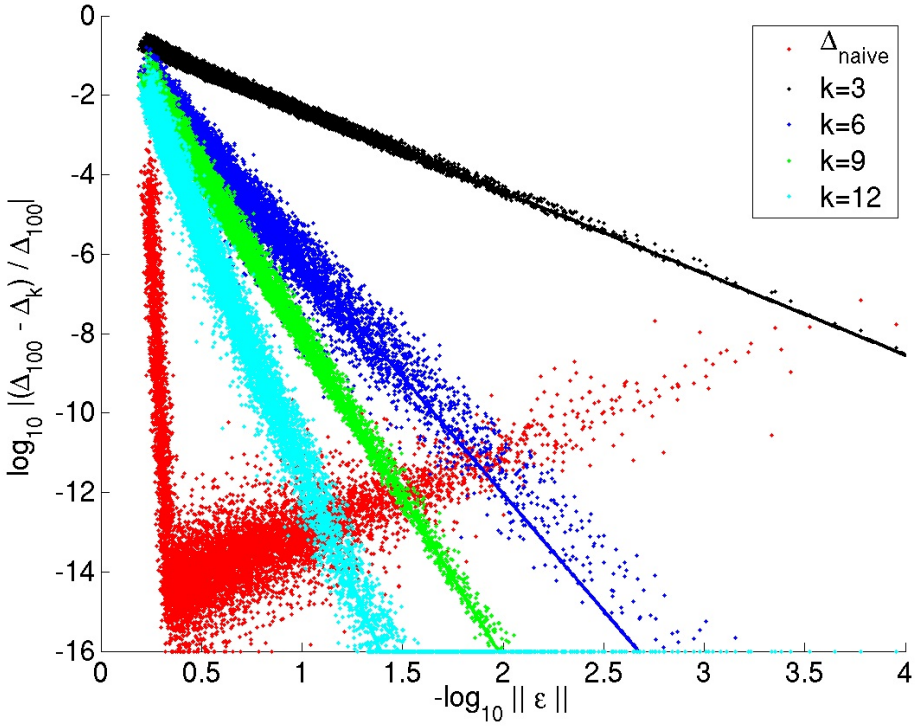

Figure 2.2: Plot comparing the naive and approximate formulae, truncated at different orders for calculating JSD as a function of the normalized l_2-distance ($\|\boldsymbol{\varepsilon}\|$) between pairs of randomly generated probability distributions. In this figure, $\Delta \equiv \Lambda$. Best fit slopes are: -2.05 ($k = 3$), -5.89 ($k = 6$), -8.14 ($k = 9$), -11.91 ($k = 12$) and -105.43 (comparing naive with $k = 100$).

exact expression (Λ_{naive}) and the approximate expansion ($\Lambda_k; k \in \{3, 6, 9, 12\}$), where

$$\Lambda_k = \frac{1}{2} \sum_n \overline{p}_n \delta_{nk} \quad ; \quad \delta_{nk} = \sum_{i=1}^{k} B_i \varepsilon_n^{i+1}. \tag{2.30}$$

The results shown in Figure 2.2 suggest the series expansion to be a more numerically useful formula when the probability distributions differ by $\|\boldsymbol{\varepsilon}\| \sim O(10^{-0.5})$. Figure 2.3 further shows that when $\|\boldsymbol{\varepsilon}\| \sim O(10^{-6})$, a direct evaluation of the exact formula for JSD gives negative values (when implemented in MATLAB).

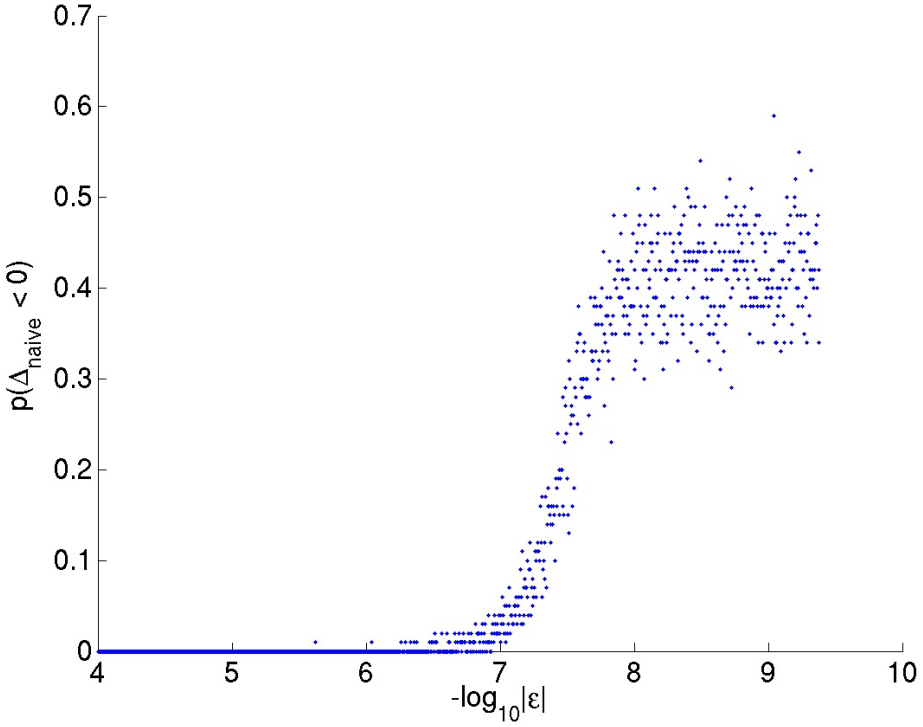

Figure 2.3: Probability of obtaining (erroneous) negative values, when directly evaluating JSD using its exact expression, is plotted as a function of $\|\boldsymbol{\varepsilon}\|$. In this figure, $\Delta \equiv \Lambda$. When implemented in MATLAB, we observe that the naive formula gives negative JSD when $\|\boldsymbol{\varepsilon}\|$ is merely of $O(10^{-6})$.

2.5 Rate of information loss in graph diffusion

We analyze here the rate of loss of predictive information between the relevance variable y and the cluster variable z, during diffusion on a graph $\mathscr{G}$, after the graph nodes have been hard-partitioned into K clusters.

2.5.1 Well-mixed limit of graph diffusion

For a given partition $\mathbf{Q}$ of the graph, defined in equation 2.13, we approximate the mutual information $\mathbf{I}\,[y;z]$ when diffusion on the graph reaches its well-mixed limit. We introduce

the linear *dependence* $\eta(y, z)$ such that

$$p(y, z) = p(y)p(z)(1 + \eta). \tag{2.31}$$

This implies $\mathbf{E}_y[\eta] = \mathbf{E}_z[\eta] = 0$ and $\mathbf{E}_y\left[\mathbf{E}_z\left[\eta^2\right]\right] = \mathbf{E}[\eta]$ where $\mathbf{E}[\cdot]$ denotes expectation over the joint distribution and $\mathbf{E}_y[\cdot]$ and $\mathbf{E}_z[\cdot]$ denote expectation over the corresponding marginals. Note that the quantity $\eta(\cdot)$ defined here has no relation to the η defined in the previous section.

In the well-mixed limit, we have $|\eta| \ll 1$. The predictive information (expressed in nats) can then be approximated as:

$$\begin{aligned}
\mathbf{I}[y; z] &= \mathbf{E}\left[\ln \frac{p(z, y)}{p(z)p(y)}\right] \\
&= \mathbf{E}_z\left[\mathbf{E}_y\left[(1 + \eta)\ln(1 + \eta)\right]\right] \\
&\approx \mathbf{E}_z\left[\mathbf{E}_y\left[(1 + \eta)(\eta - \frac{1}{2}\eta^2)\right]\right] \\
&\approx \mathbf{E}_z\left[\mathbf{E}_y\left[\eta + \frac{1}{2}\eta^2\right]\right] \\
&= \frac{1}{2}\mathbf{E}_z\left[\mathbf{E}_y\left[\eta^2\right]\right] \qquad (2.32) \\
&= \frac{1}{2}\sum_{y,z} p(y)p(z)\left(\frac{p(z, y)}{p(z)p(y)} - 1\right)^2 \\
&= \frac{1}{2}\left(\sum_{y,z} \frac{p(y, z)^2}{p(y)p(z)} - 1\right) \equiv \iota. \qquad (2.33)
\end{aligned}$$

Here, we define ι as a first-order approximation to $\mathbf{I}[y; z]$ in the well-mixed limit of graph diffusion. This quadratic approximation for $\mathbf{I}[y; z]$ is known as the χ^2-approximation.

Note that the joint and marginal distributions can also be related by the exponential dependence $\theta(y, z)$ defined by

$$p(y, z) = p(y)p(z)e^{\theta}. \tag{2.34}$$

Under this definition, the domain of the dependence is unbounded (i.e. $\theta \in \mathbb{R}$) and the mutual information is easily expressed as $\mathbf{I}[y; z] = \mathbf{E}[\theta]$. We also have

$$\sum_{i=1}^{\infty} \frac{\mathbf{E}_y\left[\theta^i\right]}{i!} = \sum_{i=1}^{\infty} \frac{\mathbf{E}_z\left[\theta^i\right]}{i!} = 0. \tag{2.35}$$

However, in the well-mixed limit $|\theta| \ll 1$, to first non-trivial order, $\theta \approx \eta$ and the expression for $\mathbf{I}[y;z]$ in terms of θ has the same form as equation 2.32.

We also have

$$\begin{aligned}
\eta(y,z) &= \frac{p(z|y)}{p(z)} - 1 \\
&\leq \frac{1}{p(z)} - 1 \\
&\leq \max_z \left(\frac{1}{p(z)}\right) - 1 \\
\eta(y,z) &\leq \max_y \left(\frac{1}{p(y)}\right) - 1.
\end{aligned}$$

$$\Rightarrow \eta(y,z) \leq \min\left(\max_z\left(\frac{1}{p(z)}\right), \max_y\left(\frac{1}{p(y)}\right)\right) - 1. \tag{2.36}$$

Thus, $\eta(y,z)$ is bounded from below by -1 (by definition) and from above as shown in equation 2.36. However, $\theta(y,z)$ is unbounded and negatively divergent for short times. Since η is much better behaved than θ for short times, and for the sake of simplicity, we choose to use the linear dependence instead of the exponential dependence.

2.5.2 Well-mixed K-partitioned graph

As in the IB method, the Markov condition $z - x - y$ allows us to make several simplifications for the conditional distributions and associated information theoretic measures. For a K-partition $\mathbf{Q}$ of the graph, we have

$$\begin{aligned}
p(y,z) &= \sum_x p(x,y,z) \\
&= \sum_x p(z|y,x)p(y|x)p(x) \\
&= \sum_x p(z|x)p(y|x)p(x) \equiv \mathbf{Q}\mathbf{P}\mathbf{G}^{t^{\mathrm{T}}}. \qquad (2.37) \\
p(y,z)^2 &= \left(\sum_x p(z|x)p(y|x)p(x)\right)^2 \\
&= \sum_{x,x'} p(z|x)p(y|x)p(x)p(z|x')p(y|x')p(x') \\
&= \sum_{x,x'} Q_{zx}G^t_{yx}P_xQ_{zx'}G^t_{yx'}P_{x'}. \qquad (2.38)
\end{aligned}$$

$$\begin{aligned} p(z) &= \sum_x p(z|x)p(x) \\ &= \sum_x Q_{zx}P_x. \end{aligned} \tag{2.39}$$

Graph diffusion being a Markov process, we have $\sum_y G^t_{x'y}G^t_{yx} = G^{2t}_{x'x}$. Using this and Bayes rule $G^t_{yx}P_x = G^t_{xy}P_y$, we have

$$\begin{aligned} \iota &= \frac{1}{2}\left(\sum_{y,z}\frac{\sum_{x,x'}Q_{zx}G^t_{yx}P_xQ_{zx'}G^t_{yx'}P_{x'}}{(\sum_{x''}Q_{zx''}P_{x''})P_y} - 1\right) \\ &= \frac{1}{2}\left(\sum_{y,z}\frac{\sum_{x,x'}Q_{zx}Q_{zx'}P_yG^t_{x'y}G^t_{yx}P_x}{(\sum_{x''}Q_{zx''}P_{x''})P_y} - 1\right) \\ &= \frac{1}{2}\left(\sum_{z}\frac{\sum_{x,x'}Q_{zx}Q_{zx'}(\sum_y G^t_{x'y}G^t_{yx})P_x}{(\sum_{x''}Q_{zx''}P_{x''})} - 1\right) \\ &= \frac{1}{2}\left(\sum_{z}\frac{\sum_{x,x'}Q_{zx}Q_{zx'}G^{2t}_{x'x}P_x}{(\sum_{x''}Q_{zx''}P_{x''})} - 1\right). \end{aligned} \tag{2.40}$$

In the hard clustering case, $\sum_x Q_{zx}P_x = p(z) = [\mathbf{QPQ}^{\mathrm{T}}]_{zz}$ and we have

$$\iota = \frac{1}{2}\left(\sum_z \frac{[\mathbf{Q}(\mathbf{G}^{2t}\mathbf{P})\mathbf{Q}^{\mathrm{T}}]_{zz}}{[\mathbf{QPQ}^{\mathrm{T}}]_{zz}} - 1\right). \tag{2.41}$$

2.5.3 Well-mixed 2-partitioned graph

We can re-write ι as

$$\begin{aligned} \iota &= \frac{1}{2}\mathbf{E}_z\left[\mathbf{E}_y\left[\eta^2\right]\right] \\ &= \frac{1}{2}\mathbf{E}_y\left[\mathbf{E}_z\left[\frac{(p(z|y)-p(z))^2}{p(z)^2}\right]\right]. \end{aligned} \tag{2.42}$$

For a bisection $\mathbf{h}$ of the graph, $z \in \{+1, -1\}$ and we have

$$p(z|x) = \frac{1}{2}(1 \pm h_x) \equiv \frac{1}{2}(1 + zh_x). \tag{2.43}$$

$$\begin{aligned} p(z|y) &= \frac{1}{p(y)}\sum_x p(z,y,x) \\ &= \frac{1}{p(y)}\sum_x p(z|x)p(y|x)p(x) \\ &= \frac{1}{2}\sum_x (1 + zh_x)p(x|y) \\ &= \frac{1}{2}(1 + z\langle \mathbf{h}|y\rangle). \end{aligned} \tag{2.44}$$

$$\begin{aligned} p(z) &= \sum_x p(z,x) = \sum_x p(z|x)p(x) \\ &= \frac{1}{2}\sum_x (1+zh_x)p(x) \\ &= \frac{1}{2}(1+z\langle \mathbf{h}\rangle). \end{aligned} \tag{2.45}$$

$$\begin{aligned} p(z|y)-p(z) &= \frac{1}{2}(1+z\langle \mathbf{h}|y\rangle) - \frac{1}{2}(1+z\langle \mathbf{h}\rangle) \\ &= \frac{1}{2}z(\langle \mathbf{h}|y\rangle - \langle \mathbf{h}\rangle). \end{aligned} \tag{2.46}$$

We then have

$$\begin{aligned} \mathbf{E}_z\left[\frac{(p(z|y)-p(z))^2}{p(z)^2}\right] &= \sum_{z=-1,1} \frac{\frac{1}{4}(\langle \mathbf{h}|y\rangle - \langle \mathbf{h}\rangle)^2}{\frac{1}{2}(1+z\langle \mathbf{h}\rangle)} \\ &= \frac{(\langle \mathbf{h}|y\rangle - \langle \mathbf{h}\rangle)^2}{2}\sum_{z=-1,1}\frac{1}{1+z\langle \mathbf{h}\rangle} \\ &= \frac{(\langle \mathbf{h}|y\rangle - \langle \mathbf{h}\rangle)^2}{1-\langle \mathbf{h}\rangle^2}. \end{aligned} \tag{2.47}$$

The mutual information $\mathbf{I}[y;z]$ can then be approximated as

$$\begin{aligned} \iota &= \frac{1}{2}\frac{\mathbf{E}_y\left[(\langle \mathbf{h}|y\rangle - \langle \mathbf{h}\rangle)^2\right]}{1-\langle \mathbf{h}\rangle^2} \\ &= \frac{1}{2}\frac{\sigma_y^2\left(\langle \mathbf{h}|y\rangle\right)}{1-\langle \mathbf{h}\rangle^2}. \end{aligned} \tag{2.48}$$

Using Bayes rule $p(x|y)p(y) = p(y|x)p(x)$, we have

$$\langle \mathbf{h}|y\rangle = \sum_x h_x p(x|y) = \sum_x \frac{h_x p(y|x)p(x)}{p(y)}. \tag{2.49}$$

$$\begin{aligned} \mathbf{E}_y\left[\langle \mathbf{h}|y\rangle^2\right] &= \sum_y p(y)\sum_{x,x'} h_x h_{x'}\frac{p(y|x)p(x)p(x'|y)}{p(y)} \\ &= \sum_y\sum_{x,x'} h_x h_{x'} p(x'|y)p(y|x)p(x). \end{aligned} \tag{2.50}$$

Again, graph diffusion being a Markov process,

$$\begin{aligned} \mathbf{E}_y\left[\langle \mathbf{h}|y\rangle^2\right] &= \sum_{x,x'} h_x h_{x'} p^{2t}(x'|x)p(x) \\ &= \mathbf{E}_{2t}[h_x h_{x'}]. \end{aligned} \tag{2.51}$$

Time dependence is explicitly denoted here to highlight the fact that diffusion on the graph is till time $2t$. Substituting $\langle \mathbf{h}|y\rangle$ in equation 2.48, we get

$$\begin{aligned}
\sigma^2\left(\langle \mathbf{h}|y\rangle\right) &= \mathbf{E}_y\left[\langle \mathbf{h}|y\rangle^2\right] - \langle \mathbf{h}\rangle^2 \\
&= \mathbf{E}_{2t}[h_x h_{x'}] - \langle \mathbf{h}\rangle^2. \qquad (2.52)\\
\iota &= \frac{1}{2}\frac{\mathbf{E}_{2t}[h_x h_{x'}] - \langle \mathbf{h}\rangle^2}{1 - \langle \mathbf{h}\rangle^2}. \qquad (2.53)
\end{aligned}$$

2.5.4 Fast-mixing graphs

When diffusion on a graph reaches its well-mixed limit in short times, we have $\mathbf{G}^{2t} \approx \mathbb{1} - 2t\mathbf{\Delta}\mathbf{P}^{-1}$, where $\mathbb{1}$ is the identity matrix. Thus, for a K-partition of a graph

$$\begin{aligned}
\mathbf{Q}(\mathbf{G}^{2t}\mathbf{P})\mathbf{Q}^{\mathrm{T}} &\approx \mathbf{Q}(\mathbf{P} - 2t\mathbf{\Delta})\mathbf{Q}^{\mathrm{T}} \\
&= \mathbf{Q}\mathbf{P}\mathbf{Q}^{\mathrm{T}} - 2t\mathbf{Q}\mathbf{\Delta}\mathbf{Q}^{\mathrm{T}}. \qquad (2.54)
\end{aligned}$$

For bisections, the short-time approximation of $\mathbf{E}_{2t}[h_x h_{x'}]$ can be written as

$$\begin{aligned}
\mathbf{E}_{2t}[h_x h_{x'}] &= \sum_{x,x'} h_{x'} p^{2t}(x', x) h_x \\
&= \mathbf{h}^{\mathrm{T}}\mathbf{G}^{2t}\mathbf{P}\mathbf{h} \\
&\approx \mathbf{h}^{\mathrm{T}}(\mathbb{1} - 2t\mathbf{\Delta}\mathbf{P}^{-1})\mathbf{P}\mathbf{h} \\
&= \mathbf{h}^{\mathrm{T}}\mathbf{P}\mathbf{h} - 2t\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\mathbf{h} \\
&= 1 - 2t\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\mathbf{h}. \qquad (2.55)
\end{aligned}$$

Note that this approximation to $\mathbf{E}_{2t}[h_x h_{x'}]$ makes no assumption about the choice of prior distribution $\mathbf{P}$ on the nodes of the graph. Furthermore, if the discrete-time diffusion operator is used instead, $\mathbf{E}_{2t}[h_x h_{x'}]$ does not approximate to $\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\mathbf{h}$ in such a simple manner.

For discrete-time diffusion, the conditional distribution $p(y|x)$ is given as

$$\widetilde{G}^s_{yx} = p(y|x) = \left[\left(\mathbf{A}\mathrm{diag}(\mathbf{d})^{-1}\right)^s\right]_{yx} \qquad (2.56)$$

where $\mathrm{diag}(\mathbf{d})$ is the diagonal matrix of node degrees, $\mathbf{A}$ is the adjacency matrix and s is the number of time steps. For any s, substituting $\mathbf{A} = \mathrm{diag}(\mathbf{d}) - \mathbf{\Delta}$ and expanding the

binomial gives

$$\begin{aligned}\widetilde{\mathbf{G}}^{2s} &= \left(\not\Vdash - \mathbf{\Delta}\mathrm{diag}(\mathbf{d})^{-1}\right)^{2s} \\ &= \not\Vdash - 2s\mathbf{\Delta}\mathrm{diag}(\mathbf{d})^{-1} + \sum_{j=2}^{2s}(-1)^j\binom{2s}{j}\left(\mathbf{\Delta}\mathrm{diag}(\mathbf{d})^{-1}\right)^j \end{aligned} \tag{2.57}$$

Thus, for $p(x) \propto d_x$, the expression for $\mathbf{E}_{2s}\left[h_x h_{x'}\right]$ becomes

$$\begin{aligned}\mathbf{E}_{2s}\left[h_x h_{x'}\right] &= 1 - \frac{2s}{m}\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\mathbf{h} \\ &+ \sum_{j=2}^{2s}\frac{(-1)^j}{m}\binom{2s}{j}\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\left(\mathrm{diag}(\mathbf{d})^{-1}\mathbf{\Delta}\right)^j\mathbf{h}\end{aligned} \tag{2.58}$$

From the above equation, we see that even when $s = 1$, unlike in the continuous-time diffusion case, $\mathbf{E}_{2s}\left[h_x h_{x'}\right]$ does not approximate as simply to the cut and ι does not approximate to the normalized or average cut.

For fast-mixing graphs, the long-time and short-time approximations for $\mathbf{I}\left[y;z\right]$ and $\mathbf{E}_{2t}[h_x h_{x'}]$, respectively, hold simultaneously.

$$\begin{aligned}\mathbf{I}\left[y;z\right](t) \approx \quad \iota(t) \quad &\approx \left(\tfrac{1}{2} - t\tfrac{\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\mathbf{h}}{1-\langle\mathbf{h}\rangle^2}\right) \\ \Rightarrow d\mathbf{I}\left[y;z\right]/dt \approx \quad d\iota/dt \quad &\propto \begin{cases} \mathcal{A} & ; \quad p(x) \propto 1 \\ \mathcal{N} & ; \quad p(x) \propto d_x. \end{cases}\end{aligned} \tag{2.59}$$

We have shown analytically that, for fast mixing graphs, the heuristics introduced by Shi and Malik are proportional to the rate of loss of relevance information. The error incurred in the approximations $\mathbf{I}\left[y;z\right] \approx \iota$ and $\mathbf{E}_{2t}[h_x h_{x'}] \approx 1 - 2t\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\mathbf{h}$ can be defined as

$$\mathcal{E}_0(t) = \left|\frac{\mathbf{E}_{2t}[h_x h_{x'}] - (1 - 2t\mathbf{h}^{\mathrm{T}}\mathbf{\Delta}\mathbf{h})}{\mathbf{E}_{2t}[h_x h_{x'}]}\right| \tag{2.60}$$

$$\mathcal{E}_1(t) = \left|\frac{\mathbf{I}\left[y;z\right](t) - \iota(t)}{\mathbf{I}\left[y;z\right](t)}\right|. \tag{2.61}$$

2.6 Numerical experiments

The validity of the two approximations can be seen in a typical plot of $\mathcal{E}_0(t)$ and $\mathcal{E}_1(t)$ as a function of normalized diffusion time $\tilde{t} = t/\tau$, for the two different choices of prior

distributions over the nodes. $\mathcal{E}_1$, as seen in figure 2.4, is often found to be non-monotonic and sometimes exhibits oscillations. This suggests defining $\mathcal{E}_\infty$, a modified monotonic '$\mathcal{E}_1$':

$$\mathcal{E}_\infty(t) \equiv \max_{t' \geq t} \mathcal{E}_1(t'). \tag{2.62}$$

$\mathcal{E}_\infty(t)$ is the maximum of $\mathcal{E}_1$ over all time greater than or equal to t. We do not need to define a monotonic form for $\mathcal{E}_0$ since this error is always found to be monotonically increasing in time.

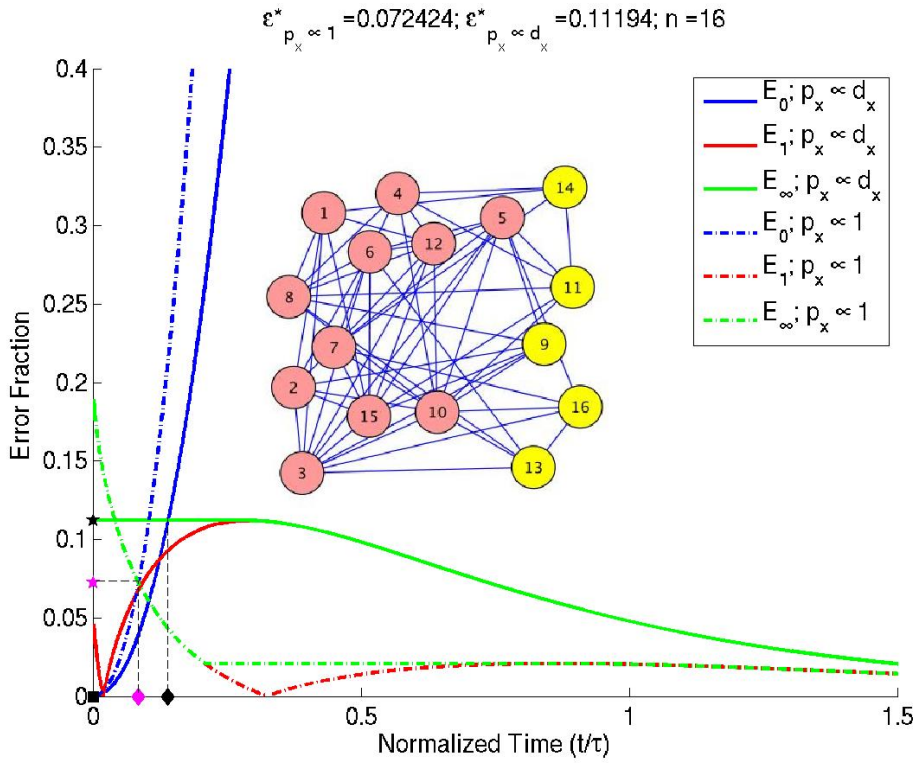

Figure 2.4: $\mathcal{E}_1$ and $\mathcal{E}_0$ vs normalized diffusion time for two choices of priors over the graph nodes. $\mathcal{E}_1$ (red) typically tends to have a non-monotonic behavior which motivates defining a monotonic $\mathcal{E}_\infty$ (green). Black – $p_x \propto d_x$, Magenta – $p_x \propto 1$. ★ – $\mathcal{E}^*$, ■ – $\tilde{t}^*_-$, ◆ – $\tilde{t}^*_+$.

By fast-mixing graphs, we mean graphs which become well-mixed in short times, i.e. graphs for which both the long-time and short-time approximations hold simultaneously

within a certain range of time $\tilde{t}^*_- \leq \tilde{t} \leq \tilde{t}^*_+$, as illustrated in figure 2.4, where we define

$$\mathcal{E}(t) = \max(\mathcal{E}_\infty(t), \mathcal{E}_0(t)) \tag{2.63}$$

$$\mathcal{E}^* = \min_t \mathcal{E}(t) \tag{2.64}$$

$$\tilde{t}^*_- = \min(\arg\min_{\tilde{t}} \mathcal{E}(\tilde{t})) \tag{2.65}$$

$$\tilde{t}^*_+ = \max(\arg\min_{\tilde{t}} \mathcal{E}(\tilde{t})). \tag{2.66}$$

$\mathcal{E}(t)$ is the larger of the modified long– and short–time errors, $\mathcal{E}_\infty$ and $\mathcal{E}_0$, at time t. $\mathcal{E}^*$ is the minimum of $\mathcal{E}(t)$ over all time. For some graphs, the plot of $\mathcal{E}(t)$ at its minimum might exhibit a plateau instead of a single point, as in figure 2.4 (for prior proportional to degree). $\tilde{t}^*_-$ and $\tilde{t}^*_+$ denote the left– and right– limits of this plateau. Note that the use of $\mathcal{E}_\infty$ instead of $\mathcal{E}_1$ overestimates the value of $\mathcal{E}^*$; the $\mathcal{E}^*$ calculated is an upper bound.

Graphs were drawn randomly from a Stochastic Block Model (SBM) distribution, with block cardinality 2, to analyze the distribution of $\mathcal{E}^*$, $\tilde{t}^*_-$ and $\tilde{t}^*_+$. As is commonly done in community detection [Danon *et al.*, 2005], for a graph of N nodes, the average degree per node is fixed at $N/4$ for graphs drawn from the SBM distribution: two nodes are connected with probability p_+ if they belong to the same block, but with probability $p_- < p_+$, if they belong to different blocks. The two probabilities are, thus, constrained by the relation

$$p_+ \left(\frac{N}{2} - 1\right) + p_- \left(\frac{N}{2}\right) = \frac{N}{4} \tag{2.67}$$

leaving only one free parameter p_- that tunes the 'modularity' of graphs in the distribution. Starting with a graph drawn from a distribution specified by a p_- value and specifying an initial cluster assignment as given by the SBM distribution, we make local moves — adding or deleting an edge in the graph and / or reassigning a node's cluster label — and search exhaustively over this move-set for local minima of $\mathcal{E}^*$. Figure 2.5 compares the values of $\mathcal{E}^*$ and $\{\tilde{t}^*_-, \tilde{t}^*_+\}$ for graphs obtained in this systematic search, starting with a graph drawn from a distribution with $p_- = 0.02$ and $N = \{16, 32, 64\}$. We note that the scatter plots for graphs of different sizes collapse on one another when $\mathcal{E}^*$ is plotted against normalized time, confirming the Fiedler value $1/\tau$ to be an appropriate characteristic diffusion time-scale [Ziv *et al.*, 2005]. A plot of $\mathcal{E}^*$ against actual diffusion time shows that the scatter plots of graphs of different sizes no longer collapse (see figure 2.6).

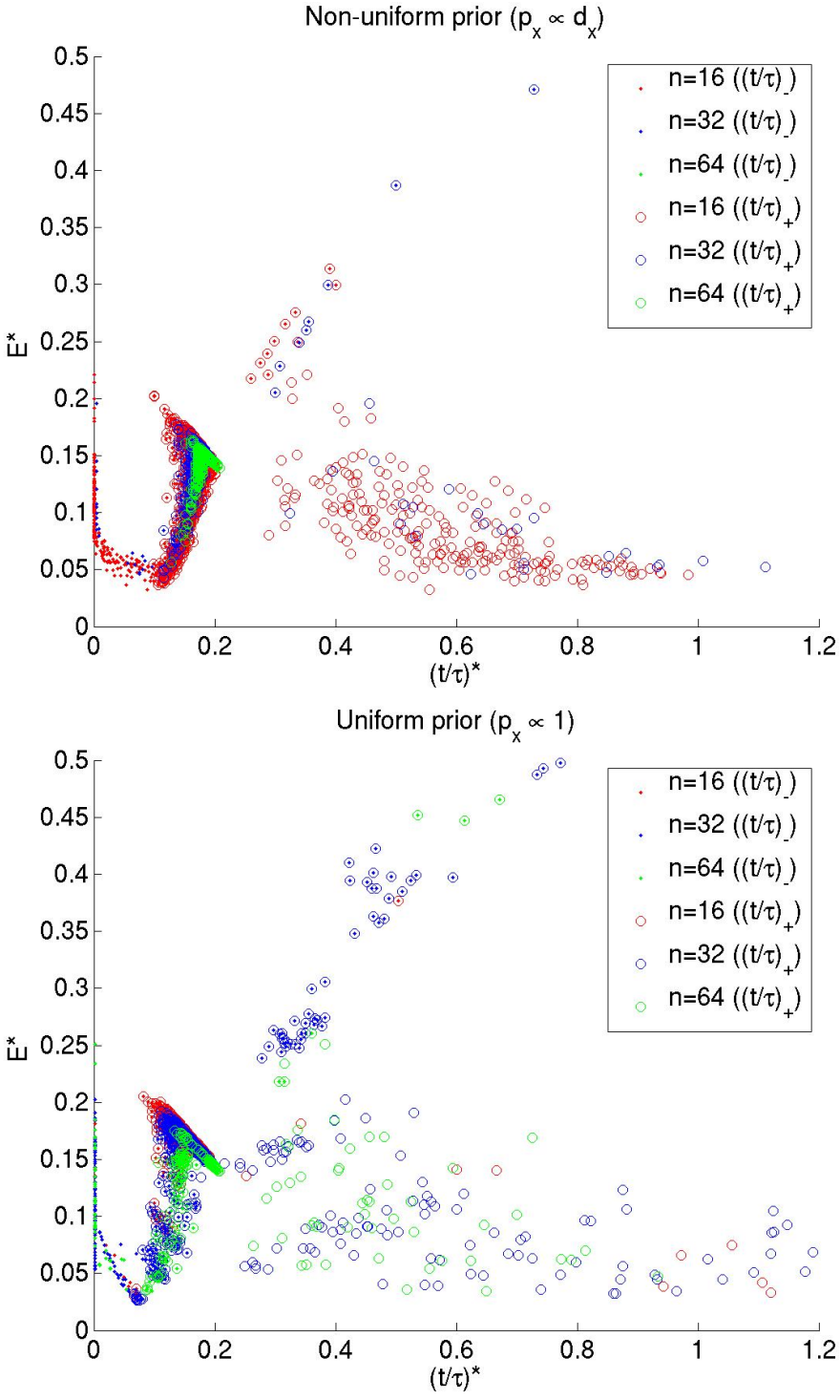

Figure 2.5: $\mathcal{E}^*$ vs $\tilde{t}^*$ for graphs of different sizes and different prior distributions over the graph nodes. In the above plot, $\tilde{t}^*_-$ and $\tilde{t}^*_+$ are represented by $\cdot$ and $\circ$, respectively.

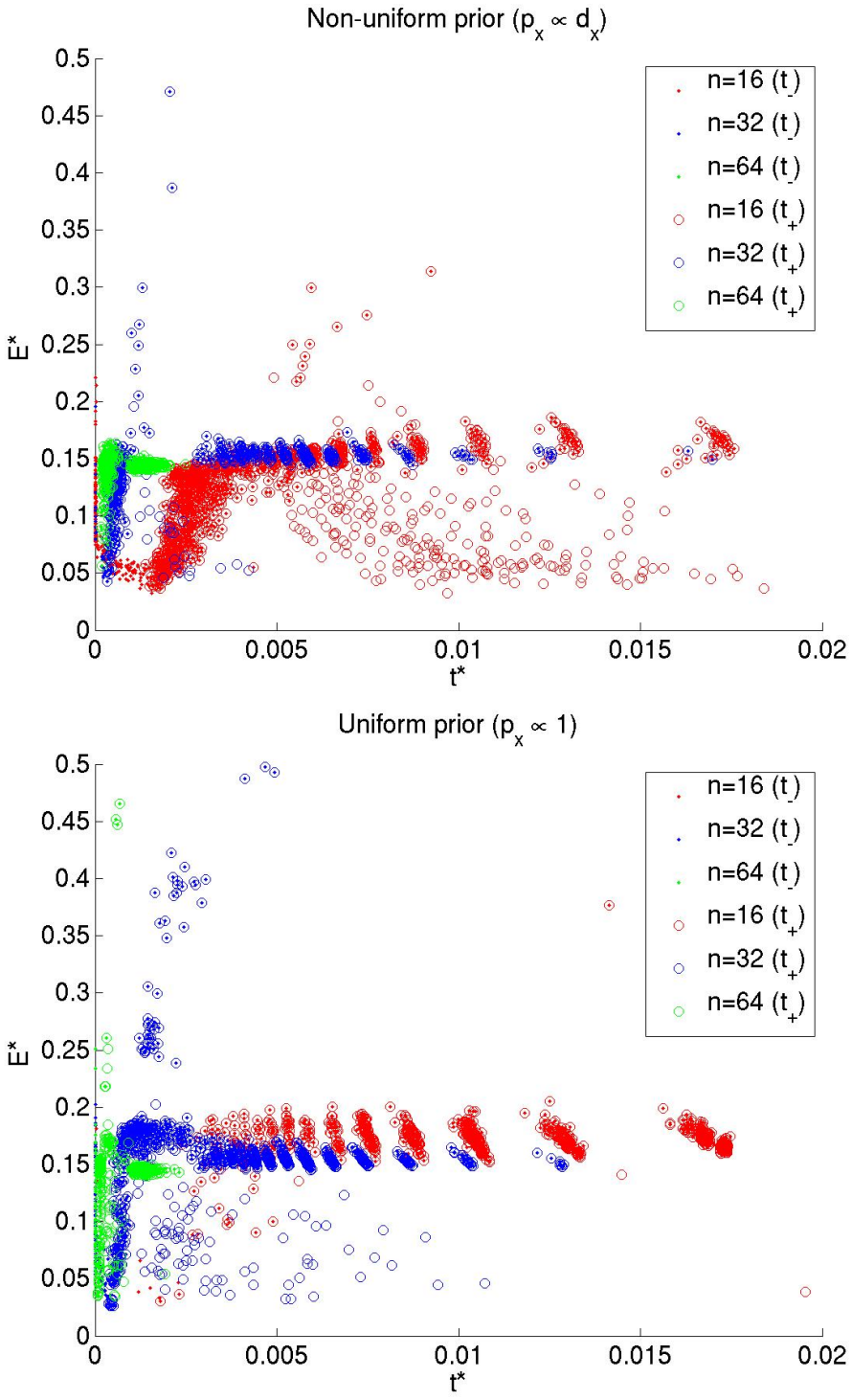

Figure 2.6: E^* vs t^* for graphs of different sizes and different prior distributions over the graph nodes. In the above plot, t^*_- and t^*_+ are represented by $\cdot$ and $\circ$, respectively.

Having shown analytically that, for fast mixing graphs, the regularized mincut is approximately the rate of loss of relevance information, it would be instructive to compare the actual partitions that optimize these goals. Graphs of size $N = 32$ were drawn from the SBM distribution with $p_- = \{0.1, 0.12, 0.14, 0.16\}$. Starting with an equal-sized partition specified by the model itself, we performed iterative coordinate descent to search (independently) for the partition that minimized the regularized cut ($\mathbf{h}_{\text{cut}}$) and one that minimized the relevance information ($\mathbf{h}_{\text{inf}}(t)$); i.e. we reassigned each node's cluster label and searched for the reassignment that gave the new lowest value for the cost function being optimized. Plots comparing the partitions $\mathbf{h}_{\text{inf}}(t)$ and $\mathbf{h}_{\text{cut}}$, learned by optimizing the two goals (estimated using 500 graphs drawn from each distribution), are shown in figure 2.7.

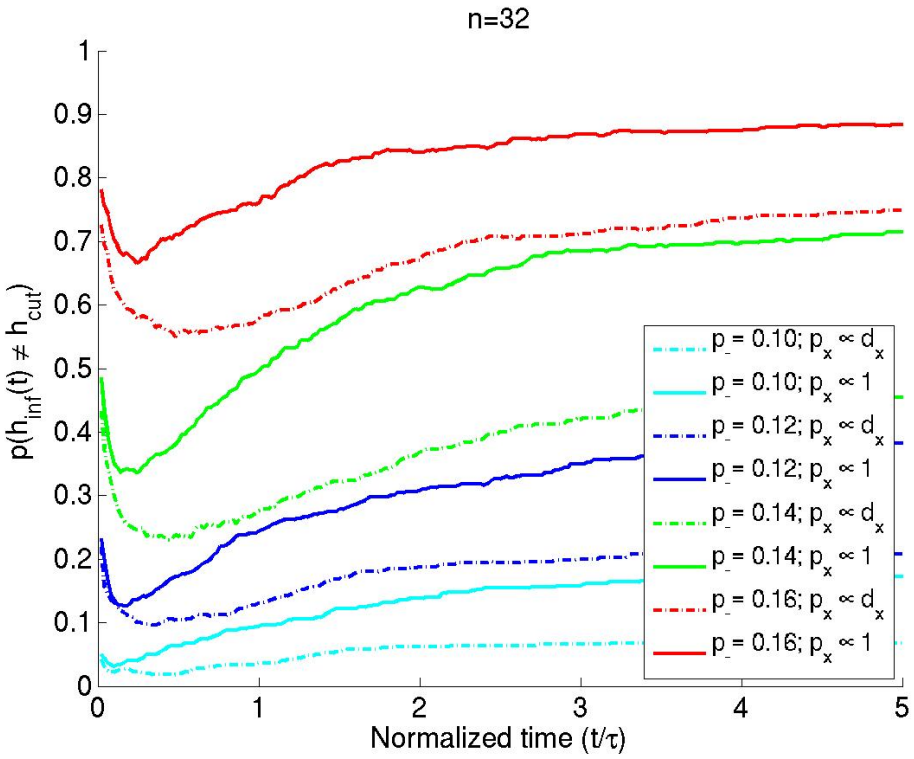

Figure 2.7: $p(\mathbf{h}_{\text{inf}}(t) \neq \mathbf{h}_{\text{cut}})$ vs normalized diffusion time, estimated using 500 graphs drawn from a distribution parameterized by a given p_- value, is plotted for different graph distributions.

2.7 Concluding remarks

In this chapter, we have shown that the normalized cut and average cut, introduced by Shi and Malik as useful heuristics to be minimized when partitioning graphs, are well approximated by the rate of loss of predictive information for fast-mixing graphs. Deriving these cut-based cost functions from rate-distortion theory gives them a more principled setting, makes them interpretable, and facilitates generalization to appropriate cut-based cost functions in new problem settings. We have also shown that the inverse Fiedler value is an appropriate normalization for diffusion time, justifying its use in the network information bottleneck algorithm to capture long-time behaviors on a network.

Absent from this derivation is a discussion of how not to overpartition a graph, i.e. a criterion for selecting K, when employing spectral graph partitioning or the network information bottleneck algorithm. It is hoped that by showing how these heuristics can be derived from a more general problem setting, lessons learned by investigating stability, cross-validation or other approaches may benefit those using min-cut based approaches as well. Furthermore, a derivation of some rigorous bounds on the magnitude of the approximation errors, under some conditions, and analysis of algorithms used in rate-distortion theory and min-cut minimization are highly promising avenues for research.

Chapter 3

Identifying virus hosts from sequence data

3.1 Context

Emerging pathogens, exemplified by the West Nile outbreak in New York (1999), SARS outbreak in Hong Kong (2003), H1N1 influenza outbreak in Mexico and the US (2009), and the more recent cholera outbreak in Haiti (2010) and E. coli outbreak in Germany (2011) are a critical threat to human society. Rapid and effective public health measures during viral epidemics typically involve identifying and classifying an outbreak from unusual clinical diagnoses, characterizing and restricting viral transmission, and development of appropriate vaccines and treatments. An integral part of this response is the accurate identification and characterization of the virus and understanding what molecular changes in the virus facilitated human infection; a notoriously difficult task in the initial stages of the outbreak when, often, very little reliable, biological information about the virus is known. Complete identification of an organism involves determining the sequence of its genome — a unique blueprint that encodes all the information necessary for the organism to function, within the context of its environment, and reveals details of its evolutionary history. Spurred by rapid advances in high-throughput sequencing technologies, genome sequencing has become one of the most promising and reliable tools to identify and characterize a novel organism. For example, LUJO was identified as a novel, very distinct virus after the

sequence of its genome was compared to other arenaviruses [Briese *et al.*, 2009].

This chapter will primarily focus on the goal of predicting the host of a virus from the viral genome. The most common approach to deduce a likely host of a virus from the viral genome is sequence / phylogenetic similarity (i.e., the most likely host of a particular virus is the one that is infected by related viral species). Host inference from phylogenetic trees is consistent with our picture of evolution. Molecular phylogenetic trees constructed using multiple alignment or maximum likelihood methods have been used extensively to determine the original host and evolution of a variety of pathogens. Examples include the swine-origin H1N1 influenza virus [Smith *et al.*, 2009], influenza A virus [Nelson and Holmes, 2007], human immunodeficiency virus [Rambaut *et al.*, 2004], and *Vibrio cholerae* [Chin *et al.*, 2011].

Inference of phylogenies from sparse data, however, is both statistically difficult and methodologically contentious. Techniques based on sequence similarity can also give ambiguous and misleading results when dealing with species very distant to known, annotated species. Additionally, armed with a phylogenetic tree, one still requires a principled and accurate assessment of how placement in the tree should be interpreted as association to a host. Moreover, lacking from these techniques is the ability to identify host-specific motifs that can allow us to understand the essential functional changes that enabled the virus to infect a new host. Alternative approaches used in the virus community are typically based on the fact that viruses undergo mutational and evolutionary pressures from the host. For instance, viruses could adapt their codon bias for a more efficient interaction with the host translational machinery or they could be under pressure of deaminating enzymes (e.g. APOBEC3G or HIV infection). All these factors imprint characteristic signatures in the viral genome. Several techniques have been developed to extract these patterns (e.g., nucleotide and dinucleotide compositional biases, and frequency analysis techniques [Touchon and Rocha, 2008]). Although most of these techniques could reveal an underlying biological mechanism, they lack sufficient accuracy to provide reliable assessments.

Another promising area of research is metagenomics, in which DNA and RNA samples from different environments are sequenced using shotgun approaches. Metagenomics provides an unbiased understanding of the different species that inhabit a particular niche.

Examples include the human microbiome and virome, and the Ocean metagenomics collection [Williamson *et al.*, 2008]. It has been estimated that there are more than 600 bacterial species living in the mouth but that only 20% have been characterized. Pathogen characterization and metagenomic analysis point to an extremely rich diversity of unknown species, where partial genomic sequence is often the only information available. Our main goal here is to develop approaches that can help infer categorical characteristics of an organism from subsequences of its genomic sequence (e.g., host, oncogenicity, and drug-resistance).

Using contemporary machine learning techniques, we present an approach to learn complex, yet sparse, tree-structured models built from simple decision rules that predict the hosts of unseen viruses, based on the amino acid sequences of proteins of viruses whose hosts are well characterized. Using sequence and host information of known viruses, we learn a multi-class classifier composed of simple sequence-motif based questions (e.g., does the viral sequence contain the motif 'DALMWLPD'?) that achieves high prediction accuracies on held-out data. Prediction accuracy of the classifier is measured by the area under the ROC curve, and is compared to a straightforward nearest-neighbor classifier. Importantly (and quite surprisingly), a post–processing study of the highly predictive sequence-motifs selected by the algorithm identifies strongly conserved regions of the viral genome, facilitating biological interpretation.

Our approach is to develop a model that is able to predict the host of a virus given its sequence; those features of the sequence that prove most useful are then assumed to have a special biological significance. Hence, an ideal model is one that is parsimonious and easy to interpret, whilst incorporating combinations of biologically relevant features. In addition, the interpretability of the results is improved if we have a simple learning algorithm which can be straightforwardly verified.

3.2 Mismatch feature space

Formally, for a given virus family, we learn a function $g : \mathcal{S} \to \mathcal{H}$, where $\mathcal{S}$ is the space of viral sequences and $\mathcal{H}$ is the space of viral hosts. The space of viral sequences $\mathcal{S}$ is generated by an alphabet $\mathcal{A}$ where, $|\mathcal{A}| = 4$ (genome sequence) or $|\mathcal{A}| = 20$ (primary protein

sequence). Defining a function on a sequence requires representation of the sequence in some feature space. Below, we specify a representation $\phi : \mathcal{S} \to \mathcal{X}$, where a sequence $s \in \mathcal{S}$ is mapped to a vector of counts of subsequences $x \in \mathcal{X} \subset \mathbb{N}_0^D$. Given this representation, we have the well-posed problem of finding a function $f : \mathcal{X} \to \mathcal{H}$ built from a space of simple binary-valued functions.

The collected data consist of N primary protein sequences, denoted $s_1 \dots s_N$, of viruses whose host class, denoted $h_1 \dots h_N$ is known. For example, these could be 'plant', 'vertebrate' and 'invertebrate'. The label for each virus is represented numerically as $\mathbf{y} \in \mathcal{Y} = \{0, 1\}^L$ where $y_l = 1$ if the index of the host class of the virus is l, and where L denotes the number of host classes. Note that this representation allows for a virus to have multiple host classes. In the remainder of this thesis, we use notation that treats the index n over examples (an *extensive* index) as different from an index into a specific vector or matrix (*intensive* indices), i.e. when referring to a vector associated with the n^{th} example, we use lowercase boldface variables. For example, $\mathbf{y}_n$ is a label vector associated with the n^{th} example while y_{nl} is the l^{th} element of the label vector for the n^{th} example.

A possible feature space representation of a viral sequence is the vector of counts of exact matches of all possible k-length subsequences (k-mers). However, due to the high mutation rate of viral genomes [Duffy *et al.*, 2008] [Pybus and Rambaut, 2009], a predictive function learned using this simple representation of counts of exact matches would fail to generalize well to new viruses. Instead, we count not just the presence of an individual k-mer but also the presence of subsequences within m mismatches from that k-mer [Leslie *et al.*, 2004]. The m-neighborhood of a k-mer κ, denoted $\mathcal{N}_\kappa^m$, is the set of all k-mers with a Hamming distance [Hamming, 1950] at most m from it, as shown in Table 3.1. Let $\delta_{\mathcal{N}_\kappa^m}$ denote the indicator function of the m-neighborhood of κ such that

$$\delta_{\mathcal{N}_\kappa^m}(\beta) = \begin{cases} 1 & \text{if } \beta \in \mathcal{N}_\kappa^m \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

We can then define, for any possible k-mer β, the mapping ϕ from the sequence s onto the number of the elements of β's m-neighborhood in s as

$$\phi_{k,m}(s, \beta) = \sum_{\substack{\kappa \in s \\ |\kappa| = k}} \delta_{\mathcal{N}_\kappa^m}(\beta). \tag{3.2}$$

Finally, the d^{th} element of the feature vector for a given sequence is then defined element-wise as

$$x_d = \phi_{k,m}(s, \beta_d) \tag{3.3}$$

for every possible k-mer $\beta_d \in \mathcal{A}^k$, where $d = 1 \dots D$ and $D = |\mathcal{A}^k|$.

$m = 0$		$m = 1$		$m = 2$	
kmer	count	kmer	count	kmer	count
⋮	⋮	⋮	⋮	⋮	⋮
DQGPS	0	DQGPS	0	DQGPS	1
CQGPS	0	CQGPS	1	CQGPS	1
CQHPS	1	CQHPS	1	CQHPS	1
CQIPS	0	CQIPS	1	CQIPS	1
DQIPS	0	DQIPS	0	DQIPS	1
⋮	⋮	⋮	⋮	⋮	⋮
APGPQ	0	APGPQ	0	APGPQ	1
AQGPQ	0	AQGPQ	1	AQGPQ	1
AQGPR	1	AQGPR	1	AQGPR	1
AQGPS	0	AQGPS	1	AQGPS	1
ASGPS	0	ASGPS	0	ASGPS	1
⋮	⋮	⋮	⋮	⋮	⋮
ARGMP	0	ARGMP	0	ARGMP	1
ARGSP	0	ARGSP	1	ARGSP	1
YRGSP	1	YRGSP	1	YRGSP	1
WRGSP	0	WRGSP	1	WRGSP	1
WRGNP	0	WRGNP	0	WRGNP	1
⋮	⋮	⋮	⋮	⋮	⋮

Table 3.1: The mismatch feature space representation of a segment of a protein sequence ...AQGPRIYDDTCQHPSWWMNFEYRGSP...

Note that when $m = 0$, $\phi_{k,0}$ exactly captures the simple count representation described earlier. This biologically realistic relaxation allows us to learn discriminative functions that better capture rapidly mutating, yet functionally conserved, regions in the viral genome,

facilitating generalization to new viruses.

3.3 Alternating decision trees

Given this representation of the data, we aim to learn a discriminative function that maps features $\mathbf{x}$ onto host class labels $\mathbf{y}$, given some training data $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$. We want the discriminative function to output a measure of "confidence" [Schapire and Singer, 1999] in addition to a predicted host class label. To this end, we learn on a class of functions $\mathbf{f} : \mathcal{X} \to \mathbb{R}^L$, where the indices of positive elements of $\mathbf{f}(\mathbf{x})$ can be interpreted as the predicted labels to be assigned to $\mathbf{x}$ and the magnitudes of these elements to be the confidence in the predictions. We will use square brackets to denote the selection of a specific element in the vector output of the discriminative function, i.e., $[\mathbf{f}(x)]_l$ is the l^{th} element of the output of the function $\mathbf{f}$.

A simple class of such real-valued discriminative functions can be constructed from the linear combination of simple binary-valued functions $\psi : \mathcal{X} \to \{0, 1\}$. The functions ψ can, in general, be a combination of single-feature decision rules or their negations:

$$\mathbf{f}(\mathbf{x}) = \sum_{p=1}^{P} \mathbf{a}_p \psi_p(\mathbf{x}) \tag{3.4}$$

$$\psi_p(\mathbf{x}) = \prod_{d \in \mathrm{S}_p} \mathbb{I}(x_d \geq \theta_d) \tag{3.5}$$

where $\mathbf{a}_p \in \mathbb{R}^L$, P is the number of binary-valued functions, $\mathbb{I}(\cdot)$ is 1 if its argument is true, and zero otherwise, $\theta \in \{0, 1, \ldots, \Theta\}$, where $\Theta = \max_{d,n} x_{nd}$, and S_p is a subset of feature indices. This formulation allows functions to be constructed using combinations of simple rules. For example, we could define a function ψ as the following

$$\psi(\mathbf{x}) = \mathbb{I}(x_5 \geq 2) \times \neg\mathbb{I}(x_{11} \geq 1) \times \mathbb{I}(x_1 \geq 4) \tag{3.6}$$

where $\neg\mathbb{I}(\cdot) = 1 - \mathbb{I}(\cdot)$.

Alternatively, we can view each function ψ_p to be parameterized by a vector of thresholds $\boldsymbol{\theta}_p \in \{0, 1, \ldots, \Theta\}^D$, where $\theta_{pd} = 0$ indicates ψ_p is not a function of the d^{th} feature x_d. In addition, we can decompose the weights [Busa-Fekete and Kegl, 2009] $\mathbf{a}_p = \alpha_p \mathbf{v}_p$ into

a vote vector $\mathbf{v} \in \{+1, -1\}^L$ and a scalar weight $\alpha \in \mathbb{R}_+$. The discriminative model, then, can be written as

$$\mathbf{f}(\mathbf{x}) = \sum_{p=1}^{P} \alpha_p \mathbf{v}_p \psi_{\boldsymbol{\theta}_p}(\mathbf{x}), \tag{3.7}$$

$$\psi_{\boldsymbol{\theta}_p}(\mathbf{x}) = \prod_{d} \mathbb{I}(x_d \geq \theta_{pd}). \tag{3.8}$$

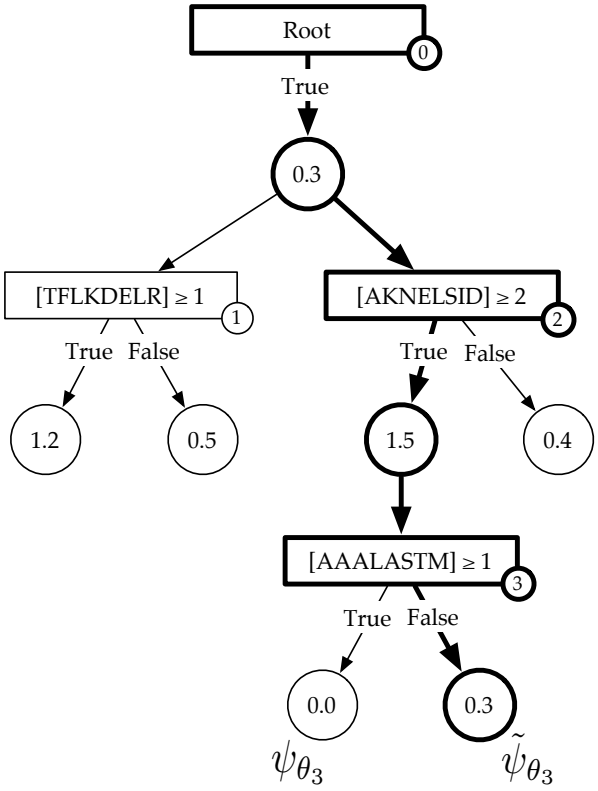

Figure 3.1: An example of an ADT where rectangles are decision nodes, circles are output nodes and, in each decision node, $[\beta] = \phi_{k,m}(s, \beta)$ is the feature associated with the k-mer β in sequence s. The output nodes connected to each decision node are associated with a pair of binary-valued functions $(\psi, \widetilde{\psi})$. The binary-valued function corresponding to the highlighted path is given as $\widetilde{\psi}_{\boldsymbol{\theta}_3}(\mathbf{x}) = \mathbb{I}([\text{AKNELSID}] \geq 2) \times \neg\mathbb{I}([\text{AAALASTM}] \geq 1)$ and the associated $\widetilde{\alpha}_3 = 0.3$. Not shown in the figure is the vote vector $\mathbf{v}$ associated with each output node.

Every function in this class of models can be concisely represented as an alternating decision tree (ADT) [Freund and Mason, 1999]. Similar to ordinary decision trees, ADTs have

two kinds of nodes: decision nodes and output nodes. Every decision node is associated with a single-feature decision rule, the attributes of the node being the relevant feature and corresponding threshold. Each decision node is connected to two output nodes corresponding to the associated decision rule and its negation. Thus, binary-valued functions in the model come in pairs $(\psi, \widetilde{\psi})$; each pair is associated with the the pair of output nodes for a given decision node in the tree (see Figure 3.1). Note that ψ and $\widetilde{\psi}$ share the same threshold vector $\boldsymbol{\theta}$ and only differ in whether they contain the associated decision rule or its negation. The attributes of the output node pair are the vote vectors $(\mathbf{v}, \widetilde{\mathbf{v}})$ and the scalar weights $(\alpha, \widetilde{\alpha})$ associated with the corresponding functions $(\psi, \widetilde{\psi})$.

Each function ψ has a one-to-one correspondence with a path from the root node to its associated output node in the tree; the single-feature decision rules in ψ being the same as those rules associated with decision nodes in the path, with negations applied appropriately. Combinatorial features can, thus, be incorporated into the model by allowing for trees of depth greater than 1. Including a new function ψ in the model is, then, equivalent to either adding a new path of decision and output nodes at the root node in the tree or growing an existing path at one of its output nodes. This tree-structured representation of the model will play an important role in specifying how Adaboost, the learning algorithm, greedily searches over an exponentially large space of binary-valued functions. It is important to note that, unlike ordinary decision trees, each example runs down an ADT through every path originating from the root node.

3.4 Multi-class Adaboost

Having specified a representation for the data and the model, we now describe Adaboost, a large-margin supervised learning algorithm which we use to learn an ADT given a data set. Ideally, a supervised learning algorithm learns a discriminative function $\mathbf{f}^*(\mathbf{x})$ that minimizes the number of mistakes on the training data, known as the Hamming loss [Hamming, 1950]:

$$\mathbf{f}^*(\mathbf{x}) = \arg\min_{\mathbf{f}} \mathcal{L}_h(\mathbf{f}) = \arg\min_{\mathbf{f}} \sum_{\substack{1 \leq n \leq N \\ 1 \leq l \leq L}} \mathbb{I}\left(H([\mathbf{f}(\mathbf{x}_n)]_l) \neq y_{nl}\right) \tag{3.9}$$

where $H(.)$ denotes the Heaviside function. The Hamming loss, however, is discontinuous and non-convex, making optimization intractable for large-scale problems.

Adaboost is the unconstrained minimization of the exponential loss, a smooth, convex upper-bound to the Hamming loss, using a coordinate descent algorithm.

$$\widetilde{\mathbf{f}}^{*}(\mathbf{x}) = \arg\min_{\mathbf{f}} \mathcal{L}_e(\mathbf{f}) = \arg\min_{\mathbf{f}} \sum_{n,l} \exp\left(-y_{nl}[\mathbf{f}(\mathbf{x}_n)]_l\right). \tag{3.10}$$

Adaboost learns a discriminative function $\mathbf{f}(\mathbf{x})$ by iteratively selecting the ψ that maximally decreases the exponential loss. Since each ψ is parameterized by a D-dimensional vector of thresholds $\boldsymbol{\theta}$, the space of functions ψ is of size $O((\Theta+1)^D)$, where Θ is the largest k-mer count observed in the data, making an exhaustive search at each iteration intractable for high-dimensional problems.

To avoid this problem, at each iteration, we only allow the ADT to grow by adding one decision node to one of the existing output nodes. To formalize this, let us define $\mathrm{Z}(\boldsymbol{\theta}) = \{d : \theta_d \neq 0\}$ to be the set of active features corresponding to a function ψ. At the t^{th} iteration of boosting, the search space of possible threshold vectors is then given as $\{\boldsymbol{\theta} : \exists \tau < t, \mathrm{Z}(\boldsymbol{\theta}) \supset \mathrm{Z}(\boldsymbol{\theta}_\tau), |\mathrm{Z}(\boldsymbol{\theta})| - |\mathrm{Z}(\boldsymbol{\theta}_\tau)| = 1\}$. In this case, the search space of thresholds at the t^{th} iteration is of size $O(t\Theta D)$ and grows linearly in a greedy fashion at each iteration (see Figure 3.1). Note, however, that this greedy growth of the search space, enforced to make the algorithm tractable, is not relevant when the class of models are constrained to belong to ADTs of depth 1.

In order to pick the best function ψ, we need to compute the decrease in exponential loss admitted by each function in the search space, given the model at the current iteration. Formally, given the model at the t^{th} iteration, denoted $\mathbf{f}^t(\mathbf{x})$, the exponential loss upon inclusion of a new decision node, and hence the creation of two new paths $(\psi_{\boldsymbol{\theta}}, \widetilde{\psi}_{\boldsymbol{\theta}})$, into the model can be written as

$$\begin{aligned}
\mathcal{L}_e(\mathbf{f}^{t+1}) &= \sum_{n,l} \exp\left(-y_{nl}[\mathbf{f}^t(\mathbf{x}_n) + \alpha\mathbf{v}\psi_{\boldsymbol{\theta}}(\mathbf{x}_n) + \widetilde{\alpha}\widetilde{\mathbf{v}}\widetilde{\psi}_{\boldsymbol{\theta}}(\mathbf{x}_n)]_l\right) &(3.11)\\
&= \sum_{n,l} w_{nl}^t \exp\left(-y_{nl}[\alpha\mathbf{v}\psi_{\boldsymbol{\theta}}(\mathbf{x}_n) + \widetilde{\alpha}\widetilde{\mathbf{v}}\widetilde{\psi}_{\boldsymbol{\theta}}(x_n)]_l\right) &(3.12)
\end{aligned}$$

where $w_{nl}^t = \exp\left(-y_{nl}[\mathbf{f}^t(\mathbf{x}_n)]_l\right)$. Here w_{nl}^t is interpreted as the weight on each sample, for each label, at boosting round t. If, at boosting round $t-1$, the model disagrees with

the true label l for sample n, then w_{nl}^t is large. If the model agrees with the label then the weight is small. This ensures that the boosting algorithm chooses a decision rule at round t, preferentially discriminating those examples with a large weight, as this will lead to the largest reduction in $\mathcal{L}_e$.

For every possible new decision node that can be introduced into the tree, Adaboost finds the (α,$\mathbf{v}$) pair that minimizes the exponential loss on the training data. These optima can be derived as

$$v_l^* = \begin{cases} 1 & \text{if } \omega_{+,l}^t \geq \omega_{-,l}^t \\ -1 & \text{otherwise} \end{cases} \tag{3.13}$$

$$\alpha^* = \frac{1}{2} \ln \frac{W_+^t}{W_-^t} \tag{3.14}$$

where for each new path ψ_n associated with each new decision node

$$\omega_{\pm,l}^t = \sum_{n:\psi_n y_{nl}=\pm 1} w_{nl}^t \tag{3.15}$$

$$W_\pm^t = \sum_{n,l:v_l\psi_n y_{nl}=\pm 1} w_{nl}^t. \tag{3.16}$$

Corresponding equations for the ($\widetilde{\alpha}$,$\widetilde{\mathbf{v}}$) pair can be written in terms of $\widetilde{W}_{\pm,l}^t$ and $\widetilde{W}_\pm^t$ obtained by replacing ψ_n with $\widetilde{\psi}_n$ in the equations above. The minimum loss function for the threshold $\boldsymbol{\theta}$ is then given as

$$\mathcal{L}_e(\mathbf{f}^{t+1}) = 2\sqrt{W_+^t W_-^t} + 2\sqrt{\widetilde{W}_+^t \widetilde{W}_-^t} + W_o^t \tag{3.17}$$

where $W_o^t = \sum_{n,l:\psi_n=\widetilde{\psi}_n=0} w_{nl}^t$. Based on these model update equations, each iteration of the Adaboost algorithm involves building the set of possible binary-valued functions to search over, selecting the one for which the loss function given by equation 3.17 is minimum and computing the associated $(\alpha, \mathbf{v})$ pair using equation 3.13 and equation 3.14.

3.5 Application to data

We use this framework to learn a predictive model to identify hosts of viruses belonging to a specific family; we show results for *Picornaviridae* and *Rhabdoviridae*. *Picornaviridae* is a family of viruses that contain a single stranded, positive sense RNA. The viral

genome usually contains about 1-2 Open Reading Frames (ORF), each coding for protein sequences about 2000-3000 amino acids long. *Rhabdoviridae* is a family of negative sense single stranded RNA viruses whose genomes typically code for five different proteins: large protein (L), nucleoprotein (N), phosphoprotein (P), glycoprotein (G), and matrix protein (M). The data consist of 148 viruses in the *Picornaviridae* family and 50 viruses in the *Rhabdoviridae* family. For some choice of k and m, we represent each virus as a vector of counts of all possible k-mers, up to m-mismatches, generated from the amino-acid alphabet. Each virus is also assigned a label depending on its host: vertebrate / invertebrate / plant in the case of *Picornaviridae*, and animal / plant in the case of *Rhabdoviridae*. The viruses used in the learning algorithm, along with their host label and subfamily annotation, are listed in Appendix A. Using multiclass Adaboost, we learn an ADT classifier on training data drawn from the set of labeled viruses and test the model on the held-out viruses.

3.5.1 BLAST achieves high classification accuracies

Given whole protein sequences, a straightforward classifier is given by a nearest neighbor approach based on the Basic Local Alignment Search Tool (BLAST) [Altschul *et al.*, 1990]. We can use the BLAST score (or p-value) as a measure of the distances between the unknown virus and a set of viruses with known hosts. The nearest neighbor approach to classification then assigns the host of the closest virus to the unknown virus. Intuitively, as this approach uses the whole protein to perform the classification, we expect the accuracy to be very high. This is indeed the case – BLAST, along with a 1-nearest neighbor classifier, successfully classifies all viruses in the *Rhabdoviridae* family, and all but 3 viruses in the *Picornaviridae* family. What is missing from this approach, however, is the ability to ascertain and interpret host relevant motifs.

3.5.2 Adaboost learns ADTs with accuracies comparable to BLAST

The accuracy of the ADT model, at each round of boosting, is evaluated using a multi-class extension of the area under the curve (AUC). Here the 'curve' is the receiver operating characteristic (ROC) which traces a measure of the classification accuracy of the ADT for

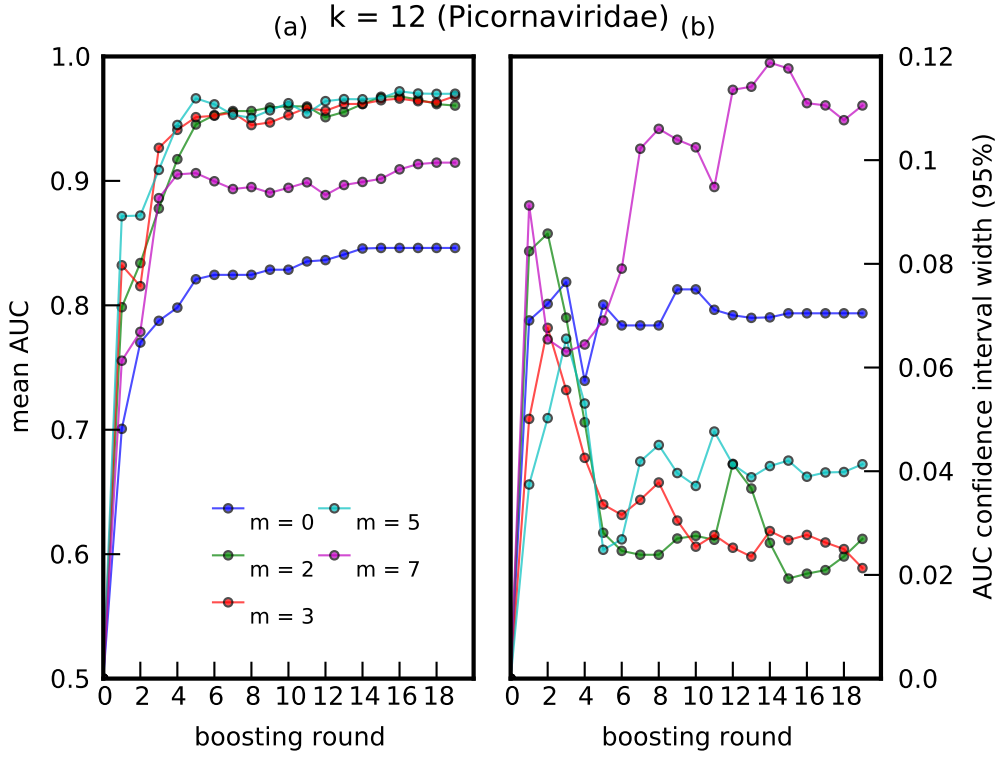

Figure 3.2: A plot of (a) mean AUC vs boosting round, and (b) 95% confidence interval vs boosting round. The mean and standard deviation were estimated over 10-folds of held-out data, for *Picornaviridae*, where $k = 12$.

each value of a real-valued discrimination threshold. As this threshold is varied, a virus is considered a true (or false) positive if the prediction of the ADT model for the true class of that protein is greater (or less) than the threshold value. The ROC curve is then traced out in true positive rate – false positive rate space by changing the threshold value and the AUC score is defined as the area under this ROC curve.

The ADT is trained using 10-fold cross validation, calculating the AUC at each round of boosting for each fold using the held-out data. The mean AUC and standard deviation over all folds is plotted against boosting round in figures 3.2 and 3.3. Note that the 'smoothing effect' introduced by using the mismatch feature space allows for improved prediction accuracy for larger values of m. For *Picornaviridae*, the best accuracy is achieved at $m = 5$, for a choice of $k = 12$; this degree of 'smoothing' is optimal for the algorithm to

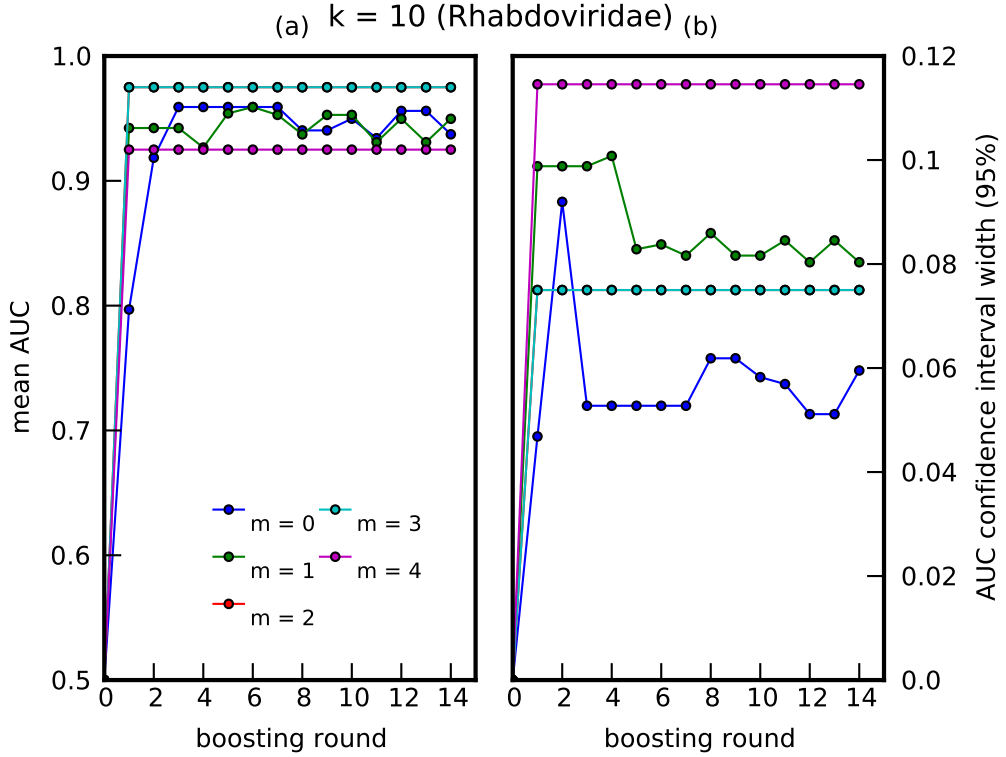

Figure 3.3: A plot of (a) mean AUC vs boosting round, and (b) 95% confidence interval vs boosting round. The mean and standard deviation were estimated over 10-folds of held-out data, for *Rhabdoviridae*, where $k = 10$. The relatively higher uncertainty for this virus family was likely due to very small sample sizes. Note that the cyan curve lies on top of the red curve.

capture predictive amino-acid subsequences present, up to a certain mismatch, in rapidly mutating viral protein sequences. For *Rhabdoviridae*, near perfect accuracy is achieved with merely one decision rule, i.e., plant and animal *Rhabdoviridae* can be distinguished based on the presence or absence of one highly conserved region in the L protein.

3.5.3 Predictive subsequences are conserved within hosts

Having learned a highly predictive model, we would like to locate where the selected k-mers occur in the viral proteomes. We visualize the k-mer subsequences selected in a specific ADT by indicating elements of the mismatch neighborhood of each selected subsequence on the virus protein sequences. In figure 3.4, the virus proteomes are grouped

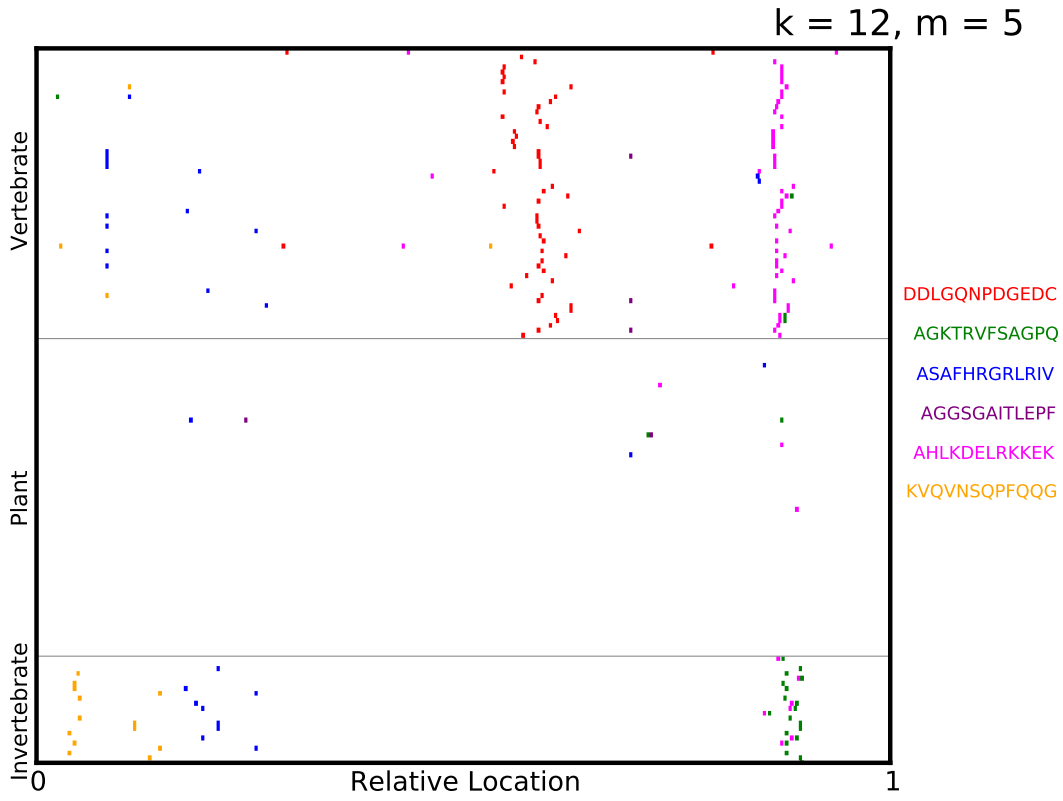

Figure 3.4: A visualization of the mismatch neighborhood of the first 6 k-mers selected in an ADT for *Picornaviridae*, where $k = 12, m = 5$. The virus proteomes are grouped vertically by their label with their lengths scaled to $[0, 1]$. Regions containing elements of the mismatch neighborhood of each selected k-mer are then indicated on the virus proteome. Note that the proteomes are not aligned along the selected k-mers but merely stacked vertically with their lengths normalized.

vertically by their label with their lengths scaled to $[0, 1]$. Quite surprisingly, the predictive k-mers occur in regions that are strongly conserved among viruses sharing a specific host. Note that the representation we used for viral sequences retained no information regarding the location of each k-mer on the virus protein. Furthermore, these selected k-mers are significant as they are robustly selected by Adaboost for different choices of train / test split of the data, as shown in figure 3.5.

We can now BLAST the selected k-mers in figure 3.4 against the GenBank database [Benson *et al.*, 2010] of *Picornaviridae* to determine known functional relevance of the asso-

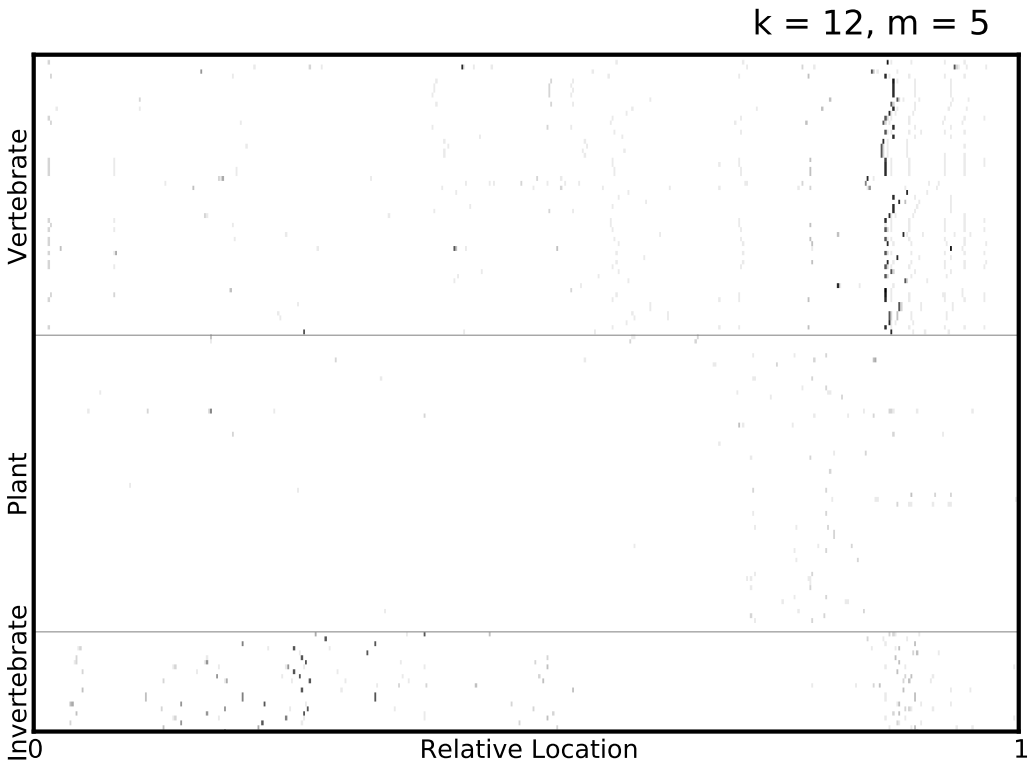

Figure 3.5: A visualization of the mismatch neighborhood of the first 6 k-mers, selected in all ADTs over 10-fold cross validation, for *Picornaviridae*, where $k = 12, m = 5$. Regions containing elements of the mismatch neighborhood of each selected k-mer are indicated on the virus proteome, with the grayscale intensity on the plot being inversely proportional to the number of cross-validation folds in which some k-mer in that region was selected by Adaboost. Thus, darker spots indicate that some k-mer in that part of the proteome was robustly selected by Adaboost over different train / test splits of the data. Furthermore, a vertical cluster of dark spots indicates that region, selected by Adaboost to be predictive, is also strongly conserved among viruses sharing a common host type.

ciated regions on the viral genomes. The k-mer 'DDLGQNPDGEDC' occurs in a region that contains genes coding for P-loop Nucleoside Triphosphate Hydrolases, important for energy-dependent assembly, operation and disassembly of protein complexes. While genes coding for these proteins occur in a variety of viruses, this specific motif aligned strongly with proteins from vertebrate viruses like human cosavirus, saffold virus and Theiler's murine encephalomyelitis virus. The k-mer 'AHLKDELRKKEK' occurs in a region coding for RNA-dependent RNA polymerase, a protein found in almost all RNA

viruses essential for direct replication of RNA from an RNA template. This motif strongly aligned with proteins from hepatitis A virus, Ljungan virus and rhinovirus isolated in humans and ducks, while the k-mer 'AGKTRVFSAGPQ' occurs in a functionally similar region for invertebrate viruses. Finally, the k-mers 'ASAFHRGRLRIV' and 'KVQVNSQPFQQG' occur in regions coding for viral capsid protein; variations in the amino acid sequence of these proteins are important both for determining viral host-specificity and contributing to antigenic diversity.

3.6 Concluding remarks / Future directions

We have presented a supervised learning algorithm that learns a model to classify viruses according to their host and identifies a set of highly discriminative oligopeptide motifs. As expected, the k-mers selected in the ADT for Picornaviridae (figures 3.4 and 3.5) occur either in the replicase motifs of the polymerase, one of the most conserved parts of the viral proteome, or in the amino acid sequence for the capsid, a protein important for host-specificity. Thus, given that partial genomic sequence is normally the only information available, we could achieve quicker bioinformatic characterization by focusing on the selection and amplification of these highly predictive regions of the genome, instead of full genomic characterization and contiguing. Moreover, in contrast with generic approaches currently under use, such a targeted amplification approach might also speed up the process of sample preparation and improve the sensitivity for viral discovery.

Overrepresentation of highly similar viruses within the data used for learning is an important source of overfitting that should be kept in mind when using this technique. Specifically, if the data largely consist of nearly similar viral sequences (e.g., different sequence reads from the same virus), the learned ADT model would overfit to insignificant variations within the data (even if 10-fold cross validation were employed), making generalization to new subfamilies of viruses extremely poor. To check for this, we hold out viruses corresponding to a particular subfamily (see Appendix A for subfamily annotation), run 10-fold cross validation on the remaining data and compute the expected fraction of misclassified viruses in the held-out subfamily, averaged over the learned ADT models.

For *Picornaviridae*, viruses belonging to the subfamilies *Parechovirus* (0.47), *Tremovirus* (0.8), *Sequivirus* (0.5), and *Cripavirus* (1.0) were poorly classified with misclassification rates indicated in parentheses. Note that the *Picornaviridae* data used consist mostly of *Cripaviruses*; thus, the high misclassification rate could be attributed to a significantly lower sample size available for learning when holding out the *Cripavirus* subfamily. For *Rhabdoviridae*, viruses belonging to *Novirhabdovirus* (0.75) and *Cytorhabdovirus* (0.77) were poorly classified. The poorly classified subfamilies, however, contain a very small number of viruses, showing that the method has very good generalization properties on average.

Other applications for this technique include identification of novel pathogens using genomic data and classification of metagenomic data using genomic information. For example, an alternative application of our approach would be the automatic discovery of multi-locus barcoding genes. Multi-locus barcoding is the use of a set of genes which are discriminative between species, in order to identify known specimens and to flag possible new species [Seberg and Petersen, 2009]. While we have focused on virus host in this chapter, ADTs could be applied straightforwardly to the barcoding problem, replacing the host label with a species label. Additional constraints on the loss function would have to be introduced to capture the desire for suitable flanking sites of each selected k-mer in order to develop the universal PCR primers important for a wide application of the discovered barcode [Kress and Erickson, 2008].

Boosting is inherently a greedy algorithm, enforcing a sparsity constraint on the model that may be inconsistent with the underlying biology of the problem. At each round of boosting, reweighting the examples after selecting the best k-mer essentially masks the predictive signal of all k-mers that are correlated with the selected k-mer. Thus, k-mers selected in consecutive boosting rounds are essentially 'orthogonal' to each other in their predictive ability. Furthermore, given the small sample sizes, the selected k-mer is often only marginally better than the next best k-mer at a given boosting round. A more biologically informative model structure would be one in which all predictive k-mers are included in the model and nearly equal weights are assigned to those k-mers that are similarly predictive of the host. For example, some k-mers that are in the 1-mismatch neighborhood of the first k-mer selected by boosting could, intuitively, be considered to be equally predic-

tive of host label and should be encouraged to have similar weights in the classification model.

One way to enforce such a constraint on model complexity is to penalize the boosting loss function using a graph-induced norm of the vector of weights $\boldsymbol{\alpha}$. Specifically, let us assume that we are given the adjacency matrix $\mathbf{A}$ of a graph specifying some relational structure between k-mers, e.g., a k-dimensional De Bruijn graph on the relevant alphabet. Minimization of the graph-regularized cost function can then be specified as:

$$\widetilde{\mathbf{f}}^*(\mathbf{x}) = \arg\min_{\mathbf{f}} \sum_{n,l} \exp\left(-y_{nl}[\mathbf{f}(\mathbf{x}_n)]_l\right) + \lambda \sum_{d,d'} A_{dd'}|\alpha_d - \alpha_{d'}|^q \tag{3.18}$$

where λ parameterizes the relative importance of predictability over interpretability and q quantifies how strongly we would like to set weights of related features to be equal. However, using such regularizations to the current problem require

1. a prespecified, problem-relevant graph of all the k-mers used in learning (i.e., $\mathbf{A}$) and
2. fast, efficient optimization algorithms to minimize the regularized loss-function

Note that, given this rather different constraint on model complexity, we can no longer additively grow the model one decision rule at a time – a key feature at the heart of boosting algorithms. A radically different model is necessary for such problems – one that allows for more biologically reasonable sparsity constraints, obviates the need for a pre-specified graph structure among k-mers and uses existing optimization paradigms – and will be addressed in Chapter 5.

Chapter 4

Predicting disease phenotype from genotype

4.1 Genome-wide association studies

Motivated by the strong association between family history and several diseases, human geneticists seek to identify inherited genetic differences between individuals with different disease phenotypes — differences that might underlie functional changes in proteins, gene regulation or biological pathways that play a role in the pathogenesis of the disease. Insights gained from such causal mechanisms could then aid both in predicting the risk of disease for a specific individual and in improving or inventing relevant treatments. Classical tools to map variations[1] in the genome to disease typically fall into two categories: family-based linkage studies and population-based association studies. Founded on the observation of structured patterns of linkage disequilibrium (LD)[2] in the human genome, the common goal of both approaches is to genotype a group of individuals at specific ge-

[1] At a specific location on a single chromosome, variants across a population form a discrete set whose elements are called *alleles*.

[2] Linkage disequilibrium is the non-random association between alleles at two loci in the genome within a population. Typically loci in close proximity on the same chromosome are in LD, due to low recombination between them; however, this association can also arise between loci on different chromosomes due to selection or non-random mating.

nomic markers[3] and find those markers that are associated with disease prevalence. Under the assumption that the disease-relevant gene is in LD with the associated marker, the location of the causal gene can then be easily resolved for further analysis.

Family-based linkage studies recruit individuals across different generations from several large families with multiple members affected by a disease and find correlations in patterns of inheritance between disease and genomic markers. These studies have proven to be quite successful for diseases caused by a single gene or variant (e.g., cystic fibrosis and X-linked muscular dystrophy). However, given the difficulty in finding such large families, these studies generally have low power making it difficult to find strong associations and fine-map the location of the causal variant, leading to very limited success in the study of common diseases like cardiovascular disease, diabetes and psychiatric illnesses.

Population-based association studies, on the other hand, aim to find systematic differences in genotype frequencies between cases and controls, by either focusing on markers in disease-relevant candidate genes or by analyzing a larger set of markers across the genome. Candidate gene-based studies typically focus on markers in the coding region of disease-relevant proteins that cause a change in its amino-acid sequence (non-synonymous mutation) or cause the production of truncated proteins (nonsense mutation). While these studies enjoy larger sample sizes compared to linkage studies, they depend strongly on prior knowledge of disease genes and are incapable of identifying variants in introns that cause disease through changes in regulatory mechanisms. In contrast, whole genome population-based association studies do not require prior knowledge of putative disease genes and are well-suited to identify causal variants for common diseases, through indirect association between disease and a genotyped marker conferred by LD. The success of these studies, however, strongly depends on an accurate, quantitative mapping of the variation in the human genome in different human populations.

One of the earliest representations of the human genome [Botstein *et al.*, 1980] was based on restriction fragment length polymorphisms (RFLP). Changes in the genome sequence (e.g., substitutions, additions or deletions) cause variations in the presence of re-

[3]Genotype at a specific locus refers to the combination of alleles of that locus. A k-allelic locus can have at most $\binom{k}{2}$ genotypes for an organism in which chromosomes occur in pairs.

striction enzyme recognition sites; these variants were quantified by variation in fragment lengths produced by restriction enzymes and measured laboriously using Southern blots. RFLP maps were followed by maps of the human genome based on microsatellites (short tandem repeats of 2, 3 or 4 base pair sequences at a given genomic location) measured by fast, polymerase chain reaction (PCR) based assays. Microsatellites, though low in number in any given population, have a high degree of polymorphism (i.e., several states or alleles for a given microsatellite) and played an important role in the success of family-based linkage studies. Population-based association studies, however, required maps based on variants that had a low mutation rate, could be easily genotyped on a large scale and allowed for very high coverage of the genome, motivating the use of single nucleotide polymorphisms (SNP)[4] to map the human genome. Characterization of LD in the human genome [Pritchard and Przeworski, 2001] and availability of databases of SNPs, along with the development of fast, inexpensive, accurate sequencing technologies and the subsequent success of the Human Genome Project, stimulated the International Hapmap Project [International Hapmap Consortium, 2003] — a multiphase project to create a genome-wide database of common SNPs that could guide genetic studies of clinical phenotypes.

Spurred by rapid, technological advances, decreasing costs in high-throughput sequencing and genotyping, and recent advances in quantifying patterns of inheritance of SNPs among evolutionarily different populations [International Hapmap Consortium, 2005], genome-wide association studies (GWAS) have become a promising tool to answer fundamental questions on the genetic basis of complex traits and diseases. GWAS are designed on the foundation of the 'common disease – common variant' hypothesis [Reich and Lander, 2001] which posits that the causal mechanisms of polygenic diseases that are common in a human population are influenced by common genetic variants that occur in the population with high frequency, making them susceptible to detection using moderately large population association studies. These large-scale studies aim to infer the genotype of hundreds of thousands of common genetic polymorphisms for several hundreds of cases

[4]Single nucleotide polymorphism is a single-base pair locus that varies within a population. In humans, SNPs are typically bi-allelic with the common and rare alleles also referred to as major and minor alleles respectively.

and controls for a phenotype of interest and elucidate those variants that have strong, significant association with that phenotype. Armed with phenomenal amounts of such genomic data on common polymorphisms, geneticists can then infer the genetic architecture of common traits — particularly the number of genetic loci that underlie variation in heritable traits, the distribution of their effect sizes[5], their complex mechanisms of action[6], possible epistatic[7] interactions and their dependence on environmental conditions. The phenotypes of interest typically include common, large spectrum diseases and disorders such as type-1 diabetes, type-2 diabetes, bipolar disorder, and autism, and quantitative traits such as height, BMI, and blood cholesterol level.

In contrast to candidate gene-based studies, GWAS are hypothesis-generating studies discovering polymorphisms that contribute to the expression of a disease or trait (henceforth, called risk variants). Given the strong LD structure in the human genome [Pritchard and Przeworski, 2001], and the limitations on the number of SNPs that current chips can hold, most studies choose a suitable set of common tag SNPs, with minor allele frequency (MAF) greater than 5%, that have high average correlation with all known SNPs and well approximate the variation in the human genome. Since it is unknown whether the causal variants are included in the genotyped set of SNPs, the inferred risk variants can only suggest genomic regions that contribute to the phenotype of interest. Once these associations are replicated in independent studies and data sets, the relevant genomic regions can further be fine-mapped to identify rarer, putative causal variants. Gene expression studies and identification of expression Quantitative Trait Loci (eQTL) in these regions are other promising tools that can reveal relevant causal pathways [Jallow *et al.*, 2009] [Nejentsev *et al.*, 2009].

---

[5]Effect size of a locus or polymorphism is its contribution to variation in a phenotype.

[6]Polymorphisms can confer risk in an additive or non-additive (dominant or recessive) manner. An additive mechanism implies that having two copies of the allele (homozygous) confers twice the risk as having one copy (heterozygous).

[7]Epistasis is the phenomenon in which the phenotypic effects of a variant is modified by the states of other variants. Epistatic effects typically occur when a phenotype results from the physical or functional interaction between multiple genes.

4.2 Statistics of case-control studies

Given the inferred genotypes of cases and controls for a given categorical phenotype, most GWAS to-date have employed traditional single-variate statistical tools to test the null hypothesis of no association between the inferred genotype and observed disease state. A typical study chooses between a 2 degree-of-freedom (df) Pearson test or the Fisher exact test to compute the association between the rows and columns of a 3×2 matrix containing the counts of the three genotypes (common homozygous, rare homozygous and heterozygous) among cases and controls. Furthermore, to increase the power of the study, disease risk from individual SNPs is assumed to be additive, allowing one to use the Pearson test on 2×2 matrices containing counts of alleles, instead of genotypes. However, it is unclear a priori what fraction of disease-relevant SNPs function in an additive manner and what fraction function in a dominant or recessive manner. For continuous phenotypes, studies typically employ multi-factorial analysis of variance (ANOVA) – a statistical test similar to the Pearson test – to resolve traits into contributing loci on molecular marker maps. Using a variety of such tools, several large scale GWAS over the last decade have discovered a number of common SNPs to be strongly implicated in age-related macular degeneration [Maller *et al.*, 2006], Type-1 diabetes [Barrett *et al.*, 2009], obesity [Speliotes *et al.*, 2010], and several other traits and diseases [Yang *et al.*, 2010] [Weiss *et al.*, 2009].

The use of single-variate statistics for such high-dimensional problems, however, demands tight statistical constraints to correct for the overall Type-1 error rate introduced by the inevitable multiple hypothesis nightmare [Hunter and Kraft, 2007]. The error rate is traditionally reduced by setting a threshold (typically 5%) for the probability of detecting a false positive association among **all** the statistical tests conducted. Given the large number of SNPs being genotyped in a typical study, this threshold translates to a very stringent p-value significance level of 1×10^{-8} per SNP, set to weed out spurious correlations between SNPs and phenotype. Such strict constraints often lead to reduced statistical power, requiring larger and larger sample sizes, careful meta-analysis [Speliotes *et al.*, 2010] or multi-tiered studies [Easton *et al.*, 2007] to detect putative associations that did not pass the necessary statistical constraints. Furthermore, these studies typically do not test for

association between multiple SNPs and disease due to the exponentially large number of tests to be carried out, thus inevitably neglecting information in the joint distribution of groups of SNPs. Despite several successes of GWAS, an oft-mentioned failure is the low fraction of sibling recurrent risk that is accounted for by the risk variants that have been detected to-date for several heritable traits [Goldstein, 2009]. Finally, very few studies have explored the inference of models from genotypic data that are predictive of genetic risk of disease. Most of these studies have focused on building a predictive model purely from associated variants that passed stringent statistical controls; a striking feature of their results is the extremely poor predictive power conferred by risk variants detected by current study sizes [Jakobsdottir *et al.*, 2009] [Janssens and van Duijn, 2008] [Speliotes *et al.*, 2010] [Purcell *et al.*, 2009].

The restricted success of GWAS to strongly heritable diseases, despite large study sizes, suggests that student t-tests, case control studies with p-values, and other cornerstones of orthodox statistics simply are not the appropriate high-dimensional statistical approaches to build disease predictive models and reveal disease relevant genetic variants for complex diseases. The variety of sequence loci constitute an overwhelmingly large number of features; yet given a typical GWAS experimental study, the number of individuals and the diversity of phenotypic variation are not sufficient to reveal which of these hundreds of thousands of covariates constitute the predictive risk loci. Despite the fact that clinicians widely recognize the insufficiency of existing statistical approaches [Hunter and Kraft, 2007], genetics remains firmly entrenched in low-dimensional or one-dimensional statistical tools, which do little to help us escape the above-mentioned multiple hypothesis nightmare.

4.3 Beyond single-variate statistics in GWAS

More recently, there has been some attempt to move away from the single-variate tools that have been extremely popular in GWAS. One popular approach to increase complexity involves analyzing groups of SNPs for association with disease. While this added complexity potentially leads to a worsening of the multiple-hypothesis problem, differ-

ent studies have reduced the exponentially growing space of combinatorial features by searching only over functionally associated groups of variants — groups of SNPs that are co-located within the same gene, co-located within groups of genes sharing the same ontology [Holmans *et al.*, 2009] or co-located within genes coding for proteins that interact in some common biological pathway [Emily *et al.*, 2009] [Baranzini *et al.*, 2009]. While these techniques allow for the detection of disease-relevant epistatic effects, results obtained from these studies will be strongly biased towards well-characterized parts of the human genome — uncharacterized intergenic regions that might play a role in disease via changes in gene regulation will be completely ignored.

Classification algorithms, like support vector machines (SVM) and random forests (RF), that have been popular in other applications in computational biology, have also been used to build black-box models for predicting disease. While models learned using SVMs seem to achieve high prediction accuracies [Wei *et al.*, 2009], the models were learned only on those SNPs whose p-value of association with disease crossed a threshold. Thus, it is unclear if the reported accuracies were inflated by preselecting 'predictive' SNPs based on p-value of association computed using the entire data set. Furthermore, models learned using nonlinear SVMs and RFs require the design of additional metrics to quantify the effect size of each SNP (or epistatic interactions between SNPs) in the model, making biological interpretation of these models difficult [Goldstein *et al.*, 2010] [Meng *et al.*, 2009].

Developing the appropriate statistical framework — one that is both predictive (for clinical goals) and interpretable (for basic science goals) — presents a deep machine learning challenge. Adaboost [Freund and Schapire, 1997] is an iterative large-margin classification 'meta-algorithm' that has successfully been used to learn predictive, interpretable models in other applications of computational biology [Kundaje *et al.*, 2006] [Middendorf *et al.*, 2005]. Alternating Decision Trees (ADT) [Freund and Mason, 1999] are tree-structured linear models built from simple decision rules that allow us to represent predictive combinatorial interactions between SNPs. Using Adaboost to learn ADTs from the vast amounts of genotypic data, we can learn models highly predictive of disease risk whilst allowing the algorithm to automatically infer model complexity (i.e., size of the model and presence of epistatic interactions). We compare the predictive accuracy of models learned using

this algorithm, across several diseases studied by the Wellcome Trust Case Control Consortium (WTCCC) [Burton, 2007], with that achieved by other statistical tools, and also identify predictive genomic regions selected by Adaboost.

4.4 Genotype-phenotype data

The data used in this study was obtained from one of the largest GWAS conducted by the WTCCC. The individuals included in the study were self-identified white Europeans living within Scotland, England, and Wales. Individuals were chosen from an evolutionarily homogeneous population to minimize false-positive associations and other confounding effects arising from overrepresentation of a certain subpopulation within cases, caused by large variations in disease prevalence between populations. The phenotypes studied by the consortium included common diseases of major global public health importance — Type-1 diabetes (T1D), Type-2 diabetes (T2D), Bipolar Disorder (BD), Hypertension (HT), Coronary Artery Disease (CAD), Rheumatoid Arthritis (RA), and Crohn's Disease (CD). There were two sources of controls used in the study – individuals from the 1958 British Birth Cohort (58BC) and individuals selected from blood donors to the UK Blood Services (UKBS). Since the data collection was carried out in different laboratories, having two independent control groups allowed the consortium to test for differential genotyping errors caused by differences in DNA collection and preparation. The consortium collected (and made publicly available) genomic, geographic and clinical data for $2,000$ cases per disease and $3,000$ common controls. Each of the $17,000$ samples were genotyped using GeneChip 500K Mapping Array Set, containing over $500,000$ SNPs, at Affymetrix Services Lab.

In general, genotyping arrays quantify the presence of an allele at a given SNP locus by measuring the intensity of hybridization of that locus with the complement of the flanking sequence of that SNP allele, attached to probes on the array. The hybridization intensities were then normalized using standard quantile normalization to reduce variability across arrays and then converted to log-scale to reduce the skewness of intensity distributions. To correct the log-normalized hybridization intensities for average background hybridization, each allele of each SNP had a perfect match probe and a mismatch probe, from which a

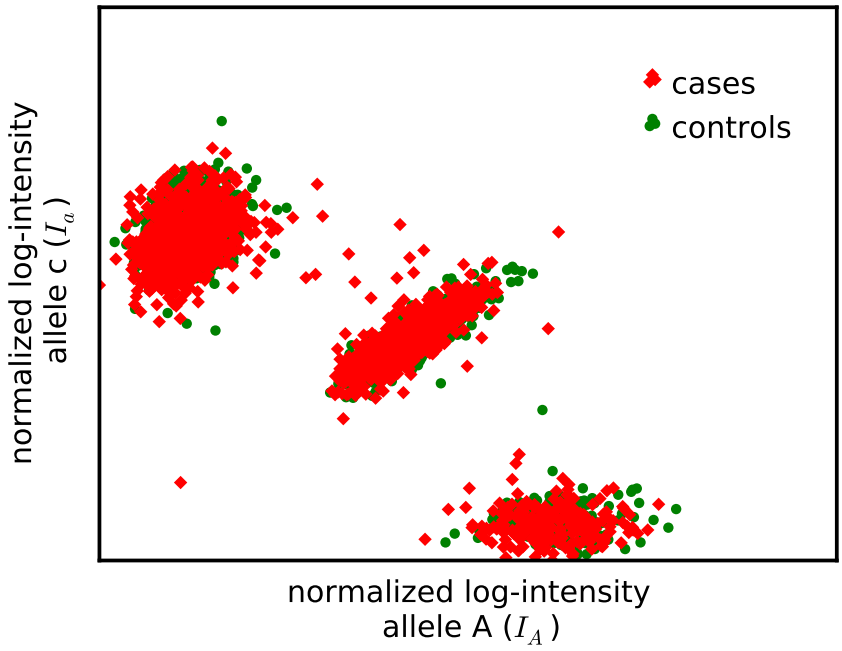

Figure 4.1: A scatter plot of the normalized log-intensities of the two alleles for a given SNP (major allele 'A' and minor allele 'c' in this case). Each point on the plot corresponds to a specific individual genotyped at that SNP. A typical scatter plot would have three modes, each corresponding to one of three genotypes – AA, Ac and cc. The genotype at that SNP for each individual can then be determined using a clustering algorithm.

set of transformed log-intensities were computed. Each SNP on each array was genotyped using multiple probes (approximately 6 to 10 probes per SNP) to reduce probe-induced differences in hybridization; a simple averaging of the background-corrected log-intensities was used to assign a pair of normalized log-intensities (one for each allele) for each SNP of each individual. The genotype for each SNP was then inferred using CHIAMO, a calling algorithm based on a hierarchical Bayesian model for clustering, developed by the consortium. The data, made available by the consortium, consist of normalized log-intensities and processed genotype calls for approximately $490,000$ SNPs for each of the $17,000$ samples.

A key step to learning interpretable models is choosing a biologically informative representation of the data. To choose between the available log-intensity data and the geno-

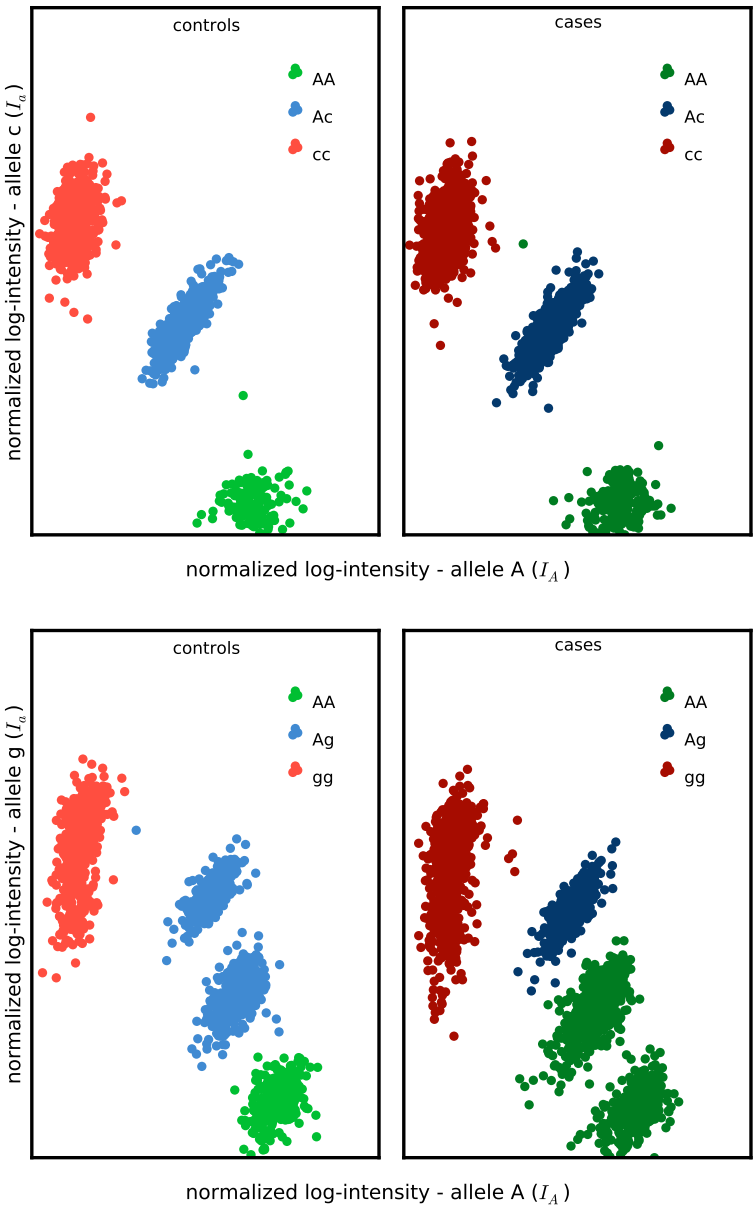

Figure 4.2: A scatter plot of the normalized log-intensities for two different SNPs. Different colors in a subfigure correspond to different genotypes, computed using CHIAMO. The top two subfigures show accurate calling by the algorithm, while the bottom two subfigures show pathologies in the genotype calls, presumably caused by clustering the cases and controls separately.

type calls, we generated cluster plots of intensities for each SNP, indicating the genotype calls computed using CHIAMO, to visually verify that the genotype calls were reasonably accurate. A detailed search through several hundred SNP cluster plots brought to light pathologies (see figure 4.2) where systematic differences between case / control genotype calls were induced, possibly due to the flexibility of the model in allowing for differences in cluster parameter values between cases and controls. From the inferred genotypes in the bottom subfigure of figure 4.2, we see that there are far more cases with genotype 'AA' while controls tend to be biased toward the genotype 'Aa'. Any learning algorithm would select this SNP as strongly predictive of disease, despite the fact that in each mode in the 4-mode intensity scatter plot, the number of cases and controls is roughly equal. To avoid confounders introduced by the calling algorithm, we chose to apply our learning algorithm directly on the log-intensity data.

Intuitively, a SNP can be said to be predictive of disease if the fraction of cases and controls in a specific mode of the intensity plot differ significantly. Lacking access to reliable genotype data (information that could distinguish the different modes in a scatter plot), we can model such differences by simply asking if there are more cases than controls on one side of an angular decision boundary, motivating data representation in terms of angles. Specifically, for each sample, given the normalized log-intensity for the major and minor alleles, I_A and I_a, of a given SNP, we represent that SNP by an angle computed as $x = \tan^{-1} \frac{I_a}{I_A}$. Thus, each sample is now represented by a vector of angles, instead of a vector of genotypes. Given that our multivariate model is learned directly on the angle data, in contrast to preprocessing steps for single-variate methods, we do not need to discard those SNPs deemed "bad" by the WTCCC (e.g., SNPs that did not pass stringent quality control metrics, SNPs with low minor-allele frequency, SNPs with very poor clustering or SNPs that departed from Hardy-Weinberg equilibrium); our algorithm learns on the full set of SNP measurements provided by the study.

Armed with this representation of the data, our overall goal is to infer associations between genotype and disease phenotype, for any given disease. To this end, we develop a model that can predict the disease state of an individual given their genotype measurements across several SNPs. An ideal model is one that is simple and easy to interpret,

whilst incorporating putative combinatorial interactions between SNPs. In addition, the interpretability of the results is improved if we have a simple, intuitive learning algorithm which can be easily verified and allows us to infer the correct model complexity.

4.5 Alternating decision trees

We aim to learn a discriminative function that maps a vector of SNP angles $\mathbf{x} \in \mathcal{X} = [0, \frac{\pi}{2}]^D$ onto disease labels $y \in \{1, -1\}$, given some training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where D is the number of SNPs genotyped and N is the sample size. In addition to a predicted label, we want the discriminative function to output a measure of "confidence" for that prediction [Schapire and Singer, 1999]. To this end, we learn on a class of functions whose range is the real line; the sign of the output can be interpreted as the predicted disease label and the magnitude can be interpreted as the confidence in the predictions.

A simple class of such real-valued discriminative functions can be constructed from the linear combination of simple binary-valued functions $\phi : \mathcal{X} \rightarrow \{0, 1\}$. Each function ϕ can, in general, be a combination of single-SNP decision rules or their negations:

$$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t \phi_t(\mathbf{x}) \tag{4.1}$$

$$\phi_t(\mathbf{x}) = \prod_{d \in S_t} \mathbb{I}(x_d \geq \theta_d) \tag{4.2}$$

where $\alpha_t \in \mathbb{R}$, T is the number of binary-valued functions, $\mathbb{I}(\cdot)$ is 1 if its argument is true and zero otherwise, $\theta_d \in [0, \frac{\pi}{2}]$, and S_t is a subset of SNP indices. This representation allows functions to be constructed using combinations (logical conjunctions) of single-SNP rules, facilitating the inference of putative epistatic interactions that are predictive of disease state. For example, we could define a function ϕ as the following

$$\phi(\mathbf{x}) = \mathbb{I}(x_5 \geq 0.5) \times \neg\mathbb{I}(x_{11} \geq 1.) \times \mathbb{I}(x_1 \geq 0.7) \tag{4.3}$$

where $\neg\mathbb{I}(\cdot) = 1 - \mathbb{I}(\cdot)$. The weights of all the decision rules for a particular SNP straightforwardly correspond to the effects sizes of the different genotypes of the SNP. For example, if the terms in the learned model corresponding to SNP x_4 with alleles 'A' and 'g' can be

written as

$$f(\mathbf{x}) = 1.2 \times \mathbb{I}(x_4 \geq 1.05) + 0.5 \times \mathbb{I}(x_4 \leq 0.98) - 1.1 \times \mathbb{I}(x_4 \leq 0.5), \tag{4.4}$$

a sample with the genotype 'gg' satisfies the first rule with an effect size of 1.2, a sample with the genotype 'Ag' satisfies the second rule with an effect size of 0.5 and a sample with the genotype 'AA' satisfies the second and third rules with an effect size of -0.6.

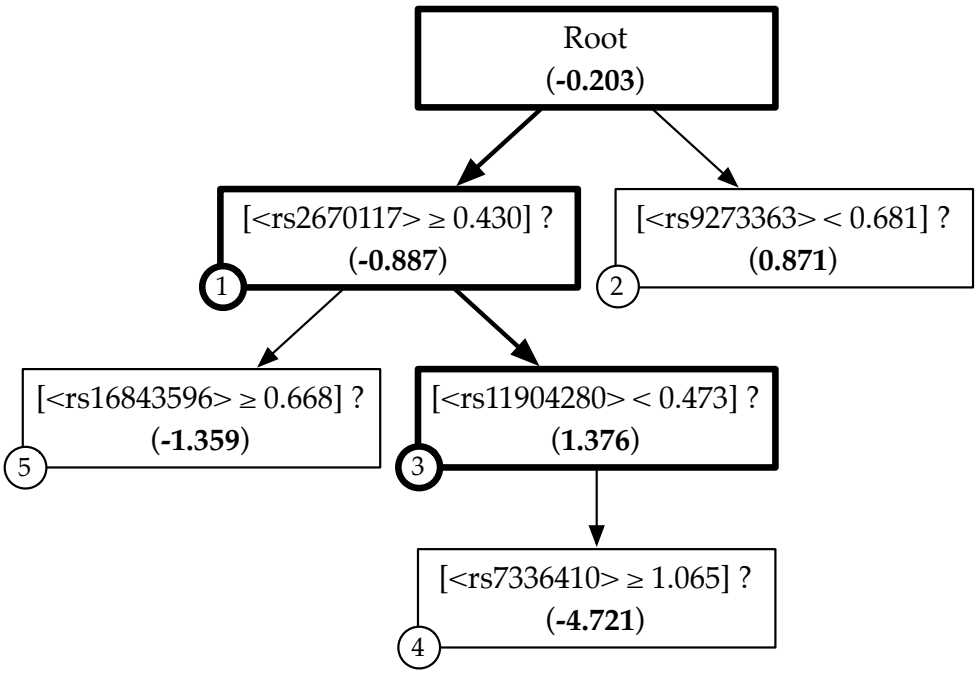

Figure 4.3: Shown here is an example of a one-sided ADT where decision nodes are represented by rectangles containing the associated SNP and angle threshold. Each path from the root to a decision node in an ADT is associated with a binary-valued function ϕ built from combinations of single-SNP decision rules in that path. The weight α associated with a binary-valued function is denoted in parentheses within the terminating decision node in its associated path. For example, the function ϕ corresponding to the highlighted path is given as $\phi_3 = \mathbb{I}(\angle\text{rs2670117} \geq 0.430) \times \neg\mathbb{I}(\angle\text{rs11904280} \geq 0.473)$ with $\alpha_3 = 1.376$. The numbers in the lower-left corner of each box denotes the boosting round in which the decision node is selected.

Every function in this class of models can be concisely represented as an Alternating Decision Tree (ADT) [Freund and Mason, 1999] built from one-sided decision rules. The decision rules are called 'one-sided' because they contribute to a prediction only if the rule

associated with them is satisfied; i.e., no information is conveyed when the rule is not satisfied. Unlike ordinary decision trees, ADTs with one-sided rules have only decision nodes. Every decision node is associated with a single-SNP decision rule, the attributes of the node being the relevant SNP and corresponding angle threshold. A path from the root to a decision node is associated with a function ϕ built from a conjunction of single-SNP decision rules in that path, with negations applied appropriately. Combinatorial rules can, thus, be incorporated into the model by allowing for trees of depth greater than 1. The attribute of a path is the weight α assigned to its associated binary-valued function.

Based on this representation of the model, adding a new function ϕ into the model is equivalent to either adding a new path of decision nodes at the root node in the tree or growing an existing path at one of the existing decision nodes. This tree-structured representation of the model will play an important role in specifying how Adaboost, the learning algorithm, greedily searches over an exponentially large space of binary-valued functions. It is crucial to note that, unlike ordinary decision trees, each example runs down an ADT through every path originating from the root node, with each path contributing to the prediction if all the decision rules in that path are satisfied.

4.6 Adaboost

Having specified a representation for the model, we now describe Adaboost, a large-margin supervised learning algorithm which we use to learn an ADT given a data set. Adaboost is the unconstrained minimization of the exponential loss, using a coordinate descent algorithm.

$$f^*(\mathbf{x}) = \arg\min_f \mathcal{L}(f) = \arg\min_f \sum_{1\leq n\leq N} \exp\left(-y_n f(\mathbf{x}_n)\right). \tag{4.5}$$

Adaboost learns a discriminative function $f(\mathbf{x})$ by iteratively selecting the binary-valued function ϕ that maximally decreases the exponential loss. Since ϕ could potentially be a combination of several single-SNP decision rules, each rule parameterized by a different angle threshold θ, the space of functions ϕ has a size complexity of $\mathrm{O}(N^D)$, where for a finite data set of size N the number of distinct threshold angles for each feature is at most

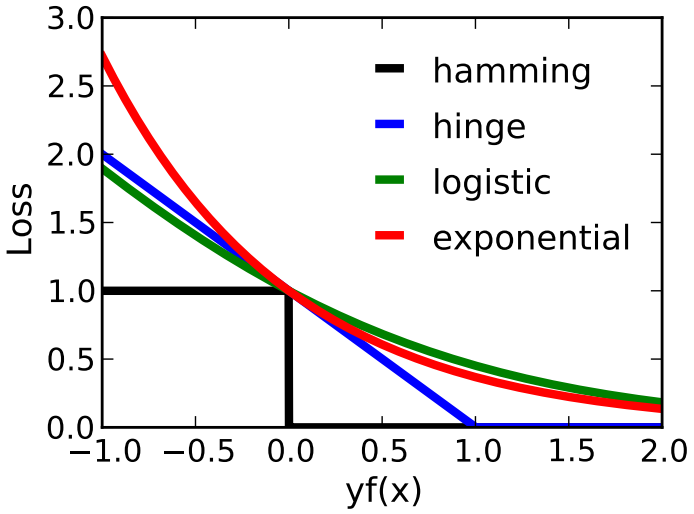

Figure 4.4: A comparison of the loss functions for three popular classification algorithms – the hinge loss for SVMs, exponential loss for Adaboost and logistic loss for logistic regression. These surrogate losses are continuous, upper-bounds to the Hamming loss, one which an ideal classification algorithm should aim to minimize.

N. Thus, an exhaustive search over the space of all functions ϕ is intractable for such high-dimensional problems.

To avoid this, at each iteration, we only allow the ADT to grow by adding one decision node to one of the existing decision nodes (see figure 4.5). In this case, the search space of functions ϕ at the t^{th} iteration has a space complexity of $\mathrm{O}(tND)$ and grows linearly in a greedy fashion at each iteration. Note, however, that this greedy growth of the search space, enforced to make the algorithm tractable, is not relevant when the class of models are constrained to belong to ADTs of depth 1; i.e., when no combinatorial interactions are allowed in the model.

In order to pick the best function ϕ, we need to compute the decrease in exponential loss admitted by each function in the search space, given the model at the current iteration. Formally, given the model at the t^{th} iteration, denoted $f^t(\mathbf{x})$, the exponential loss upon

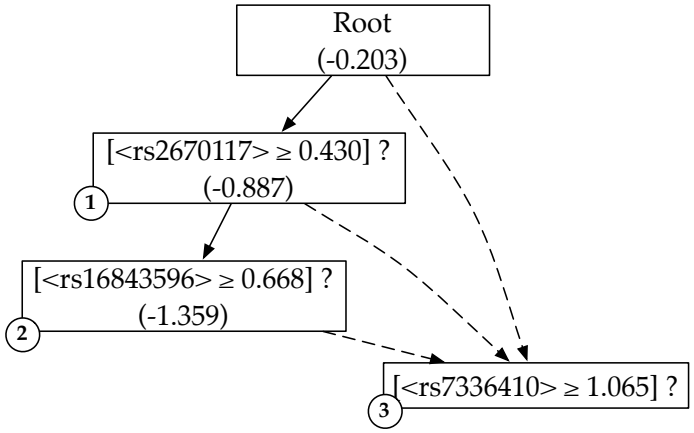

Figure 4.5: A visual representation of the constrained manner in which the ADT is allowed to grow, restricting the space of possible functions to search over.

inclusion of a new decision node into the model can be written as

$$\begin{aligned}\mathcal{L}(f^{t+1}) &= \sum_{n=1}^{N} \exp\left(-y_n(f^t(\mathbf{x}_n) + \alpha\phi(\mathbf{x}_n))\right) \\ &= \sum_{n} w_n^t \exp\left(-y_n\alpha\phi(\mathbf{x}_n)\right) \\ &= e^{-\alpha} \sum_{y_n\phi(\mathbf{x}_n)=1} w_n^t + e^{\alpha} \sum_{y_n\phi(\mathbf{x}_n)=-1} w_n^t + \sum_{\phi(\mathbf{x}_n)=0} w_n^t\end{aligned} \tag{4.6}$$

where $w_n^t = \exp\left(-y_n f^t(\mathbf{x}_n)\right)$. Here w_n^t can be interpreted as a weight on each sample, at boosting round t. If, at boosting round $t-1$, the model disagrees with the true disease label for the n^{th} example, then the example get upweighted, i.e., w_n^t becomes large. This ensures that the boosting algorithm chooses a decision rule at round t, preferentially discriminating those examples misclassified after round $t-1$.

For every possible new decision node that can be introduced to the tree, Adaboost computes the α that minimizes the exponential loss on the training data. Differentiating equation 4.6 with respect to α gives

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\alpha} &= \sum_{n} w_n^t(-y_n\phi(\mathbf{x}_n)) \exp\left(-y_n\alpha\phi(\mathbf{x}_n)\right) \\ &= \sum_{n}(-y_n\phi(\mathbf{x}_n)) \exp\left(-y_n f^t(\mathbf{x}_n) + \alpha\phi(\mathbf{x}_n)\right)\end{aligned} \tag{4.7}$$

$$\left.\frac{\partial\mathcal{L}}{\partial\alpha}\right|_{\alpha^*} = \sum_{n}(-y_n\phi(\mathbf{x}_n))w_n^{t+1} = 0 \tag{4.8}$$

The optimum value of α can be derived as

$$\alpha^* = \frac{1}{2} \ln \frac{W_+^t}{W_-^t} \tag{4.9}$$

where for the new path ϕ associated with the decision node added to the tree

$$W_\pm^t = \sum_{n: y_n \phi(\mathbf{x}_n) = \pm 1} w_n^t. \tag{4.10}$$

The minimum loss function attained for this function ϕ is then given as

$$\mathcal{L}(f^{t+1}) = 2\sqrt{W_+^t W_-^t} + W_o^t \tag{4.11}$$

where $W_o^t = \sum_{n:\phi(\mathbf{x}_n)=0} w_n^t$. Having derived these model update equations, the Adaboost algorithm can then be specified as additively growing a linear model by selecting the decision node ϕ that minimizes equation 4.11 at each round. Specifically, at the t^{th} round, given weights w^t over the examples, Adaboost selects the binary-valued function ϕ that maximally decreases the exponential loss, and then updates the weights over examples to w^{t+1} for the next round.

4.7 Entropy regularized LPboost

In the previous section, we derived the update equations for Adaboost, a coordinate descent algorithm for the unconstrained minimization of the exponential loss. Specifically, we showed in equation 4.8 that the weights over examples were updated to be orthogonal to the predictive signal in the function ϕ selected at the previous round. Intuitively, such reweighting of examples ensures that the function ϕ selected at the next round would add predictive power into the model that was not present in the previously selected function [Schapire and Singer, 1999]. In this section, we will derived relaxations to the Adaboost algorithm described earlier by first showing how the unconstrained minimization of the exponential loss is the dual problem for the constrained minimization of a relative entropy objective.

Assuming that the weights over examples at each round is normalized to be a distribution over examples, let us consider the optimization problem at round t [Kivinen and

Warmuth, 1999]

$$\begin{array}{ll} \text{minimize} & D_{KL}\left(\mathbf{w}\|\mathbf{w}^t\right) \\ \text{subject to} & \sum_n w_n y_n \phi_t(\mathbf{x}_n) = 0 \\ & \sum_n w_n = 1 \end{array} \tag{4.12}$$

where $D_{KL}\left(\cdot\|\cdot\right)$ is the relative entropy or KL divergence between two distributions. The solution to this problem gives us a distribution over examples that is closest (in KL divergence) to the current distribution and is orthogonal to the predictive signal of the selected binary-valued function ϕ_t. Introducing the constraints into the objective using Lagrange multipliers, we get

$$\begin{aligned} \mathcal{P}^* &= \min_{\mathbf{w}} \max_{\alpha,\beta} D_{KL}\left(\mathbf{w}\|\mathbf{w}^t\right) + \alpha\left(\sum_n w_n y_n \phi_t(\mathbf{x}_n)\right) + \beta\left(\sum_n w_n - 1\right) \\ &\geq \max_{\alpha,\beta} \min_{\mathbf{w}} D_{KL}\left(\mathbf{w}\|\mathbf{w}^t\right) + \alpha\left(\sum_n w_n y_n \phi_t(\mathbf{x}_n)\right) + \beta\left(\sum_n w_n - 1\right) \\ &= \mathcal{D}^* \end{aligned} \tag{4.13}$$

where the inequality is a consequence of weak duality. Minimizing the objective in the max-min problem of equation 4.13, we get the dual optimization problem of equation 4.12 as

$$\text{maximize} \quad -\log\left\{\sum_n w_n^t \exp(-\alpha y_n \phi_t(\mathbf{x}_n))\right\} \tag{4.14}$$

with the primal variables $\mathbf{w}$ and dual variables (α, β) related as

$$w_n = w_n^t \exp(-\alpha y_n \phi_t(\mathbf{x}_n)) \exp(-(1+\beta)) \tag{4.15}$$

This is exactly the problem being solved by Adaboost, with equation 4.15 matching the updates for the example weights derived earlier, up to a normalization constant. The weight for the function ϕ_t is exactly the Lagrange multiplier in the primal problem. Note that the above derivation can be applied even if the weights are unnormalized, as long as they belong to the positive orthant.

The orthogonality constraint in equation 4.12 ensures that the next selected function ϕ_{t+1} only corrects for the mistakes made by the current function ϕ_t. This constraint can

be straightforwardly extended to include all previously selected functions. The primal problem for this algorithm, called 'totally corrective' Adaboost, is given as

$$\begin{aligned} &\text{minimize} && D_{KL}\left(\mathbf{w}\|\mathbf{w}^t\right) \\ &\text{subject to} && \textstyle\sum_n w_n y_n \phi_q(\mathbf{x}_n) = 0 \quad \forall\, 1 \le q \le t \\ & && \textstyle\sum_n w_n = 1. \end{aligned} \tag{4.16}$$

The Lagrange multipliers for the t orthogonality constraints now become corrective updates for the weights of all previously selected functions ϕ. Following the steps described earlier, the dual problem can be written as

$$\text{maximize} \quad -\log\left\{\textstyle\sum_n w_n^t \exp\left(-\sum_q \alpha_q y_n \phi_q(\mathbf{x}_n)\right)\right\}. \tag{4.17}$$

The number of constraints in equation 4.16, however, increases with boosting round and it is quite possible that at higher rounds, there will be no feasible solution that satisfies all the constraints [Kivinen and Warmuth, 1999]. This motivates a relaxation of the totally corrective Adaboost, where the equality constraints are replaced by inequality bounds with the bounds being included as a penalty term into the objective.

$$\begin{aligned} &\text{minimize} && D_{KL}\left(\mathbf{w}\|\mathbf{w}^t\right) + \lambda\nu \\ &\text{subject to} && -\nu \le \textstyle\sum_n w_n y_n \phi_q(\mathbf{x}_n) \le \nu \quad \forall\, 1 \le q \le t \\ & && \textstyle\sum_n w_n = 1. \end{aligned} \tag{4.18}$$

The penalty ν encourages weights $\mathbf{w}$ to satisfy the 'orthogonality constraint' while the tuning parameter λ measures the relative importance between satisfying the 'orthogonality constraints' and finding a weight distribution closest to the current distribution. Indeed, λ can be interpreted as the 'step-size' with which Adaboost greedily traverses the space of ADTs. Introducing Lagrange multipliers α_q^+ and α_q^- for the left and right inequality constraints respectively, we can derive the dual problem to be

$$\begin{aligned} &\text{maximize} && -\log\left\{\textstyle\sum_n w_n^t \exp\left(-\sum_q \alpha_q y_n \phi_q(\mathbf{x}_n)\right)\right\} \\ &\text{subject to} && \alpha_q = \alpha_q^+ - \alpha_q^-,\ \alpha_q^+ \ge 0,\ \alpha_q^- \ge 0 \\ & && \|\boldsymbol{\alpha}\|_1 \le \textstyle\sum_q \alpha_q^+ + \alpha_q^- = \lambda \end{aligned} \tag{4.19}$$

Without loss of generality, the equality constraint $\sum_q \alpha_q^+ + \alpha_q^- = \lambda$ can be relaxed to an inequality constraint. Note that the inequality constraints in equation 4.18 correspond to a constraint on the l_∞-norm of the vector $\mathbf{u} := u_q = \sum_n w_n y_n \phi_q(\mathbf{x}_n)$; in the dual, this nicely becomes a constraint on the l_1-norm of the vector $\boldsymbol{\alpha}$ of weights on functions ϕ. This totally corrective l_1-regularized Adaboost is similar to entropy regularized LPboost with hard margins [Warmuth *et al.*, 2008]; to be consistent with literature this relaxed version of Adaboost will be called ERLPboost in the rest of this chapter.

4.8 Model evaluation

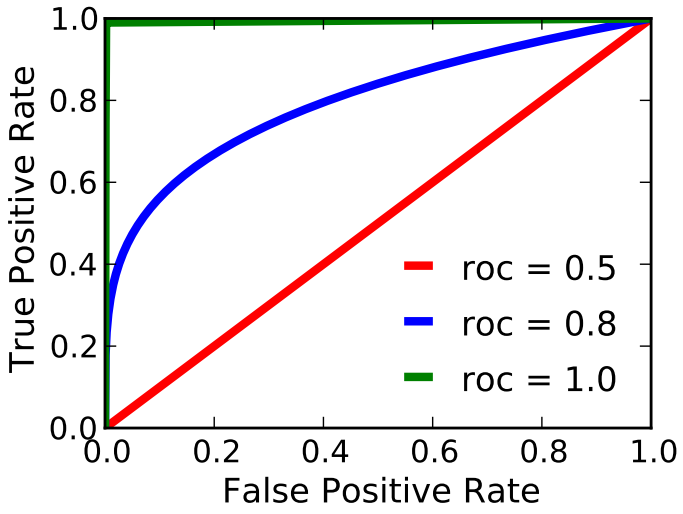

Figure 4.6: A receiver operating characteristic curve comparing the performance of three classifiers, quantified by the area under the respective curves.

The accuracy of the ADT model, at each round of boosting, is evaluated using the area under the curve (AUC). Here the 'curve' is the receiver operating characteristic (ROC) which traces the true positive rate (TPR) vs false positive rate (FPR) of the ADT for each value of a real-valued discrimination threshold; the AUC score is defined as the area under this ROC curve. For two-class classification problems, the AUC can also be computed from the Wilcoxon rank-sum test score. This non-parametric test measures the separability of two classes (cases and controls) based on some real-valued metric (the ADT prediction

score) assigned to each member of the two classes. Using 10-fold cross validation (CV), the AUC is computed on the held-out data, at each round of boosting, for each fold.

4.9 Results

In this study, we use Adaboost, a large-margin classification algorithm, and its l_1-norm regularized variant ERLPboost to infer models predictive of disease phenotype using SNP angle data for four major diseases studied by the WTCCC [Burton, 2007] — type-1 diabetes (T1D), type-2 diabetes (T2D), bipolar disorder (BD) and hypertension (HT). In this section, we demonstrate the success of these algorithms in inferring models (with and without combinatorial rules) that have high prediction accuracies and compare it to the predictive performance of traditional single-variate statistical tools that have been extremely popular in the GWAS community. We also show how boosting automatically and robustly infers sparse models, facilitating biological interpretation.

4.9.1 Boosting infers sparse, predictive ADTs with high AUC scores

In figure 4.7, we compare the AUC achieved by Adaboost and ERLPboost on held-out data using stumps (ADTs with depth 1) and trees (ADTs with depth ≥ 1) for type-1 diabetes, with the AUC score of predictive models reported for this disease in the literature. As expected, we achieve a very high AUC for T1D, a disease known to be strongly heritable [Kyvik *et al.*, 1995]. In contrast, the prediction accuracy indicated by (b) in figure 4.7 was achieved by using Support Vector Machines with radial basis function kernels on the genotype data of a subset of SNPs [Wei *et al.*, 2009]. Only those SNPs whose p-value of association with disease crossed a prespecified threshold and SNPs identified as disease-relevant in earlier studies were used in learning. Thus, it is unclear if the reported accuracies were inflated by preselecting SNPs based on p-values using the entire data set. The prediction accuracy indicated by (a) was achieved using LASSO [Kooperberg *et al.*, 2010] (l_1-regularized logistic regression) learned again on a preselected 'most significant' set of SNPs. The authors, however, critically evaluate the use of the entire data set in the preselection process – a rather common practice in the GWAS community.

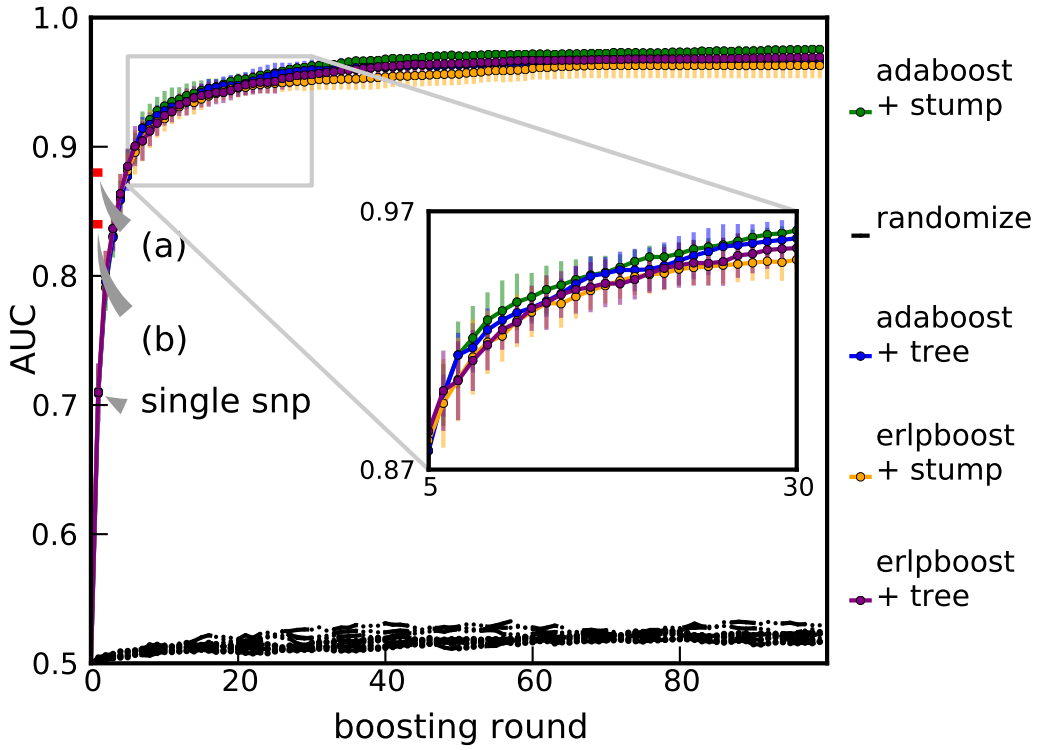

Figure 4.7: This plot illustrates the accuracy of Adaboost and ERLPboost for ADTs containing trees and stumps on Type-1 diabetes. Both algorithms and both models give very similar accuracies; the inset shows the accuracy plots zoomed, from round 5 to round 30. The accuracy of boosting is compared with the predictive accuracy of LASSO (a) and SVM (b) reported in the literature (see text for citations). The black lines plot the accuracy of each of the algorithms, with each of the models, on the data when the labels are randomized.

In addition to addressing the clinical goal of predicting disease risk accurately, Adaboost can also learn a sparse model (i.e., ADTs with a small number of decision nodes) if the underlying disease etiology is governed by a small subset of the feature space in which the data is represented. From figure 4.7, we see that the accuracy achieved by Adaboost reaches close to the maximum with a remarkably small subset of SNPs (≈ 30). In contrast, the best models learned using LASSO (marked (a)) assigned non-zero weights to over 100 predictors.

In figure 4.8, we observe a lower AUC achieved by boosting for Type-2 diabetes, consistent with the heritability estimates for this disease [Stumvoll *et al.*, 2005]. The prediction

accuracy indicated by (a) [Jakobsdottir *et al.*, 2009] [Lu and Elston, 2008] and (b) [Lango *et al.*, 2008] [Van Hoek *et al.*, 2008] were computed using odds ratios or likelihood ratios, and minor allele frequencies computed on variants with the highest odds ratios. Jakobsdottir et al., however, strongly argue against only using strongly associated SNPs to build disease predictive models – yet another common practice in the GWAS community.

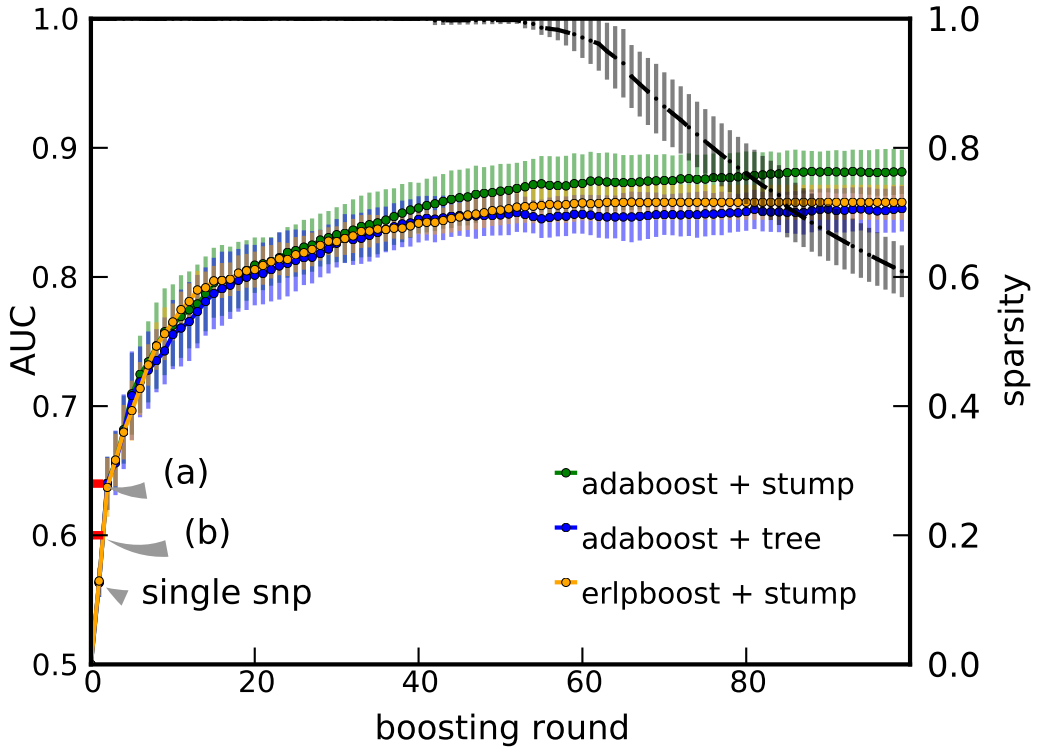

Figure 4.8: This plot illustrates the accuracy of Adaboost and ERLPboost for ADTs containing trees and stumps on Type-2 diabetes. The accuracy of boosting is compared with the performance of other predictive models reported in the literature for this disease (see text for citations). The tuning parameter for ERLPboost was chosen to give a sparser model without appreciable decrease in accuracy. The black line plots the average fraction of single-SNP decision rules with non-zero weights in the model at each boosting round (the total number of decision rules in the model is equal to the boosting round).

Furthermore, ERLPboost, a totally corrective algorithm that optimizes the weights of all the features in the model at each iteration, learns a sparser model than Adaboost while achieving comparable accuracy (see figure 4.8). This property of ERLPboost partially cor-

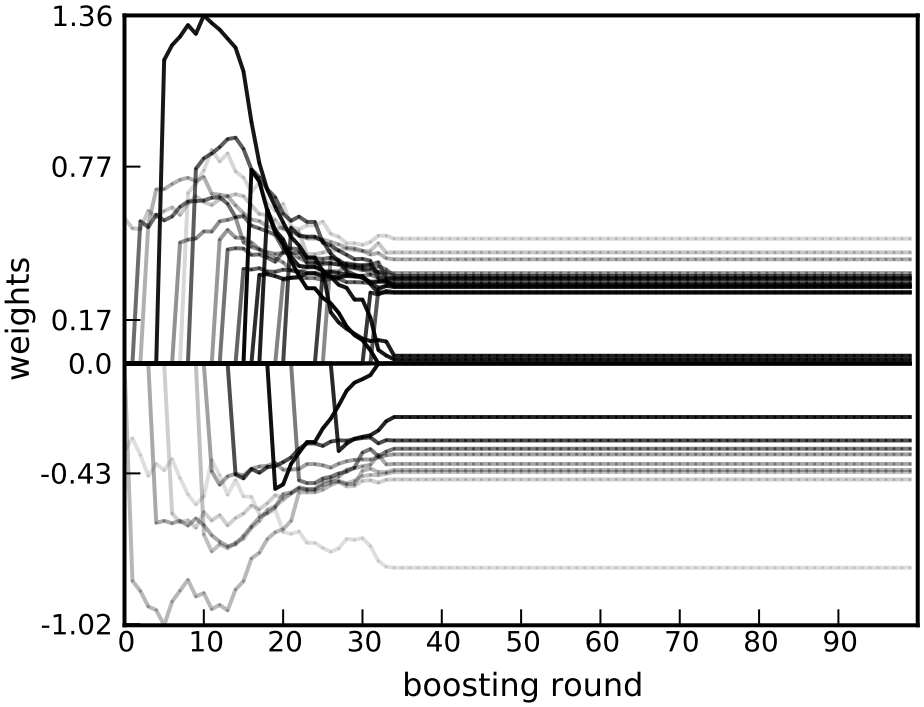

Figure 4.9: This plot illustrates the change in weight of a single-SNP decision rule as a function of boosting round, when ERLPboost is used for learning. The features shown here are the top 30 decision rules when ranked by the absolute value of their weights at the boosting round in which they were introduced into the model. The grayscale intensity of each line is proportional to the weight of the corresponding decision rule at the 100^{th} boosting round. Note that several features that were introduced into the model with very high weights were automatically deemed unimportant for prediction in later boosting rounds.

rects for the coordinate descent implementation of Adaboost by allowing features introduced into the model at earlier rounds to be automatically assigned zero weights in later rounds, if they are subsequently deemed unimportant. In figure 4.9, we plot the weights of features as a function of boosting round; the features shown are the top 30 when ranked by the absolute value of their weights at the round in which they were introduced into the model. We observe that an l_1-norm penalty enforces a corrective role by reweighting features and even removing unimportant ones from the final model without suffering a significant decrease in prediction accuracy. Both Adaboost and ERLPboost learn sparse

models by masking the predictive signal of features correlated to the feature last added into the model – the price to pay for any coordinate descent algorithm or l_1-regularized optimization problem. Thus, care should be taken when making biological inferences from this sparse set of predictive SNPs. It is possible that this small set of highly predictive SNPs will have a low overlap with risk variants that have currently been identified in the literature for any given disease.

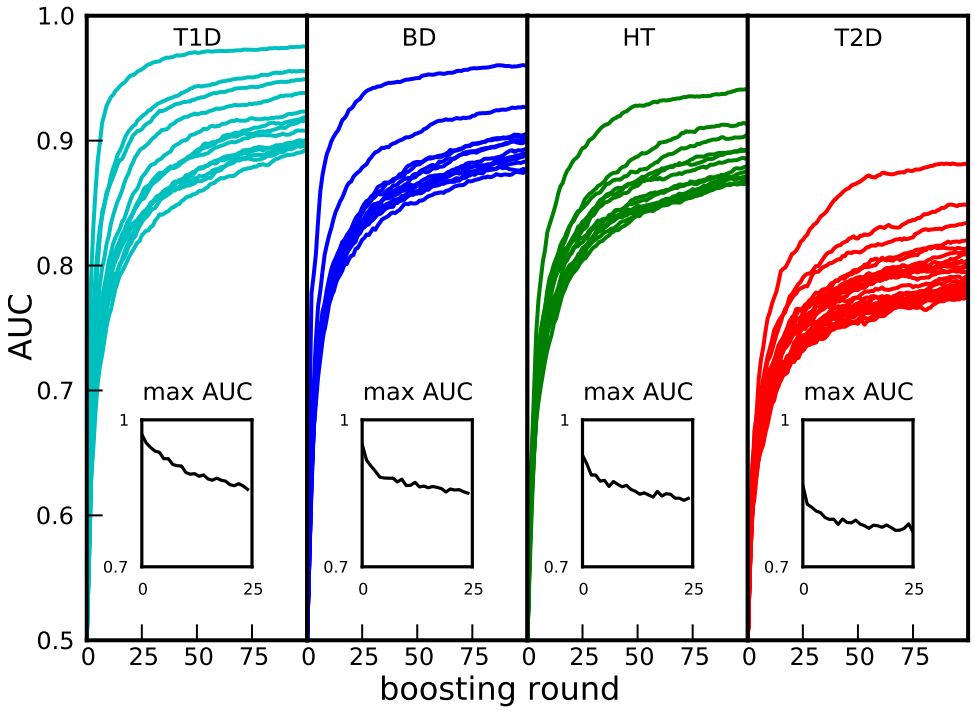

Figure 4.10: This plot illustrates the accuracy on held-out data versus boosting round as sets of SNPs selected by Adaboost are iteratively masked from subsequent learning. Each subfigure in the plot corresponds to one of four diseases – type 1 diabetes, bipolar disorder, hypertension and type 2 diabetes. Each curve traces the accuracy of Adaboost after SNPs selected by previous runs of Adaboost have been masked. For the sake of clarity, only AUC curves corresponding to every odd boost-remove iteration is shown. The inset figure in each panel plots the AUC at the 100^{th} round of boosting as a function of boost-remove iteration.

Given the strong LD structure in human genomes [Pritchard and Przeworski, 2001], it would be useful to see how much predictive signal is retained in the data if these highly

predictive SNPs selected by Adaboost are masked. By iteratively masking sets of predictive SNPs selected by Adaboost, we can identify those genomic regions containing putative causal variants that were in lower LD with the variants selected in the first round. In figure 4.10, we plot AUC versus boosting round as successive sets of SNPs selected by Adaboost are removed. While the most predictive signal is captured by the first 200 SNPs selected by boosting, quite surprisingly, we notice that a significant amount of signal remains in the data even after 25 boost-remove iterations. It is possible that systematic measurement biases between cases and controls in the WTCCC data set contributes to the relatively high predictive signal learned by Adaboost at much higher boost-remove iterations. Furthermore, we observe a high AUC for a strongly heritable disease like T1D and a relatively low AUC for T2D, known to have a higher environmental component, influenced by dietary and lifestyle choices, in its etiology. Since the model was learned on a large, geographically diverse population (in contrast to populations in twin-studies), the maximum predictive AUC achieved by such models could also be used as an alternative population-based measure for the genetic component or heritability of a disease [Visscher *et al.*, 2008].

4.9.2 Decision rules suggest intra-locus genetic interactions

A fundamental question that geneticists seek to answer is the mechanism by which the alleles of a SNP contribute to a phenotype. Specifically, is an allele of a SNP dominant or recessive for a given disease? Does being homozygous in an allele confer twice the risk as being heterozygous, for a given SNP? The decision rules used in learning ADTs naturally help answer these questions. We illustrate the ability of boosting with ADTs to automatically infer the putative mechanism (additive, dominant or recessive) of a SNP by plotting the histogram of angles for a given SNP and marking all inferred threshold angles for that SNP over different CV folds, as shown in figure 4.11. Histograms for cases and controls with different genotypes are plotted separately for the sake of clarity; the genotypes used were those provided by the WTCCC. In each subfigure, the sum of the heights of the bars of the histogram measures the number of cases or controls with a particular genotype.

A decision boundary marked red corresponds to a 'greater than or equal to' decision

while a decision boundary marked green corresponds to a 'less than' decision. For the SNP shown in figure 4.11, the red decision boundary which corresponds to having two copies of the rare allele, was consistently selected within the first 2 rounds of boosting with an average weight of 0.85 while the green decision boundary which corresponds to having two copies of the common allele was selected later with an average weight of -0.45 (usually within the first 20 rounds of boosting). Thus, a heterozygous genotype for this SNP does not satisfy either of the decision rule with an effect size of 0 while the homozygous major and minor genotypes satisfy only one of the two decision rules with average effect sizes of -0.45 and 0.85 respectively. This suggests a non-additive mechanism for this SNP; i.e., having two copies of the rare allele is necessary for increased disease risk while having two copies of the common allele possibly decreases disease risk. Note that the use of regular ADTs, similar to those used in chapter 3, would confer an effect size to the heterozygous genotype as well.

Similarly, for the SNP in figure 4.12, the green decision boundary has an average weight of 0.29 while the left and right red decision boundaries have average weights of -0.35 and -0.85 respectively. Thus, the homozygous minor genotype satisfies both 'red' decision rules with an average effect size of -1.2 while the homozygous major and heterozygous genotypes have effect sizes of 0.29 and -0.35, respectively. Interestingly, the common allele for this SNP is positively associated with risk for diabetes while having two copies of the rare allele strongly decreases disease risk. Furthermore, having two copies of the rare allele has thrice the effect size of having one copy, suggesting a strong non-additive mechanism for this SNP.

4.9.3 Predictive regions are robustly selected

Having addressed the clinical problem of learning a predictive model, we now identify where these sets of predictive SNPs occur in the human genome. Figure 4.13 is a straightforward visualization that elucidates the genomic locations of predictive SNPs selected by boosting. In the visualization, each horizontal panel corresponds to a chromosome, the length of each chromosome is proportional to the number of base pairs it contains and is divided into 1 Mb blocks while the width of each chromosome is split into the number

of CV folds (10). For each CV fold, we highlight a block with a bar if some SNP in the genomic region corresponding to that block was selected by boosting, the width of the bar being proportional to the largest relative magnitude of the weight of the SNP in the ADT.

In each panel, continuous vertical bands indicate that different SNPs in that genomic region were selected by boosting over different train / test splits of the data (broken bands indicate that the region was less consistently selected). Regions corresponding to these bands can then be interpreted as having been robustly selected by Adaboost to be highly predictive of disease phenotype and are more likely to harbor a causal risk variant for the disease. Consistently thick bands correspond to associated SNPs with large effect sizes and indicate regions that contain putative risk variants with equally large effect sizes. For example, the thickest band for T1D, located on chromosome 6 (see figure 4.13), contains variants co-located with the MHC complex that have been consistently identified as strongly associated with T1D by several GWAS over the last decade [Barrett *et al.*, 2009]. The inset in figure 4.13 shows this strongly associated 3 Mb region on chromosome 6 at a higher resolution.

The different colors in the visualization correspond to SNPs selected in different boost-remove iterations. From figure 4.13, we see that masking predictive SNPs allows boosting to infer other predictive SNPs in the same region, further adding evidence to the robustness of the observed predictive signal. Note, however, that when one set of highly predictive SNPs are masked, SNPs in completely different chromosomes also get selected as strongly predictive of disease, identifying these regions as containing putative causal variants. Lower effect sizes of causal variants in this region or the relatively lower LD between the causal variant and typed variants are possible reasons why these regions weren't selected in earlier boost-remove iterations.

4.10 Concluding remarks / Future directions

We have presented a supervised learning algorithm that infers a tree structured model built from simple single-SNP decision rules to predict the disease label of an individual. In addition to achieving very high prediction accuracies, the learned model enumerates a

sparse set of predictive SNPs and a set of combinatorial rules that are suggestive of putative epistatic interactions important for the disease. The decision rules in the model also quantify the additive or non-additive intra-locus allelic interactions of the predictive SNPs by straightforwardly assigning effect sizes to the three genotypes for each SNP.

The strength of a specific representation of SNP data and a learning algorithm is best tested by validating the model learned by the algorithm on data collected from a different population sharing the same set of SNPs. Each ADT learned by Adaboost is characterized by a tree of decision nodes, each node being associated with a SNP and an angle threshold. The use of raw signal intensity data to learn ADTs helped circumvent possible biases introduced by genotype calling algorithms. However, systematic differences in sample preparation protocols, reagents and cell lines between different laboratories and between different studies in the same laboratory would mean that the angle thresholds learned on one data set cannot easily be validated on data collected from a new study. Instead, we could use boosting to learn models on different data sets (for the same disease) independently and compute the overlap between predictive regions selected by the algorithm; validating the use of boosting with ADTs using this approach is a promising direction for research we are currently pursuing.

The coordinate descent minimization of the exponential loss is crucial for the application of boosting to such large problem settings; however, this algorithm is also its key limitation. The coordinate descent algorithm enforces an l_1-norm like penalty on the loss function [Rosset *et al.*, 2004]; given a set of predictive SNPs with strong LD, this regularization selects the most predictive SNP into the model while suppressing the predictive signal in the remaining SNPs. The boost-remove experiment described in this chapter works around this, identifying 'hot-spots' in a chromosome that are predictive of disease as shown in figure 4.13. However, it is unclear how one could combine the ADTs learned from different boost-remove iterations into one unified model where correlated SNPs are assigned similar weights.

The greedy algorithm also makes boosting susceptible to confounders from non-random differences in the distribution of SNP angles. For example, as seen in figure 4.14, systematic differences in the distribution of angles between cases and controls can lead to the

selection of decision rules that are statistically valid but do not seem to be biologically meaningful. Such decision rules are often selected in higher rounds of boosting (typically ≥ 10), presumably because each subsequent decision rule merely 'corrects' the mistakes of previously selected decision rules. Indeed, one way to correct for this would be to first infer the genotype of each sample and learn on the genotype data. Absent reliable genotype calls, an alternative approach would involve simultaneously inferring the genotype whilst automatically correcting the classification model for such confounding SNP measurements by using information from correlated / neighboring SNPs. In Chapter 5, we will describe such a unified model that can be learned using the framework of Bayesian inference; application of such models on the SNP data described in this chapter and inferring predictive LD blocks (instead of predictive SNPs) is an extremely promising avenue for future research.

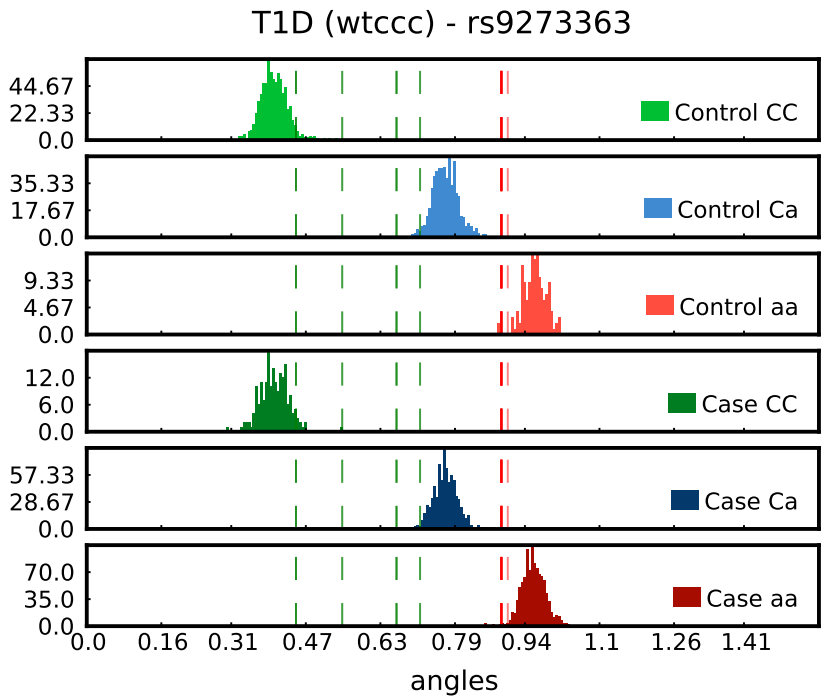

Figure 4.11: This plot compares the learned threshold angles for a SNP over 10-fold CV, with the histogram of angles for that SNP. The histograms for the three genotypes, for cases and controls, are plotted separately. The different colors of the histogram correspond to different genotypes, inferred using CHIAMO. The histogram is not normalized; thus, area under each histogram equals the number of individuals that have the corresponding genotype. A SNP can have multiple angle thresholds in the same ADT since each (SNP,angle) pair is treated as a separate decision rule. A decision boundary marked red corresponds to a 'greater than or equal to' decision while a decision boundary marked green corresponds to a 'less than' decision. A thick decision boundary (or a tight cluster of angle thresholds) indicate that the decision rule was robustly selected across different CV folds.

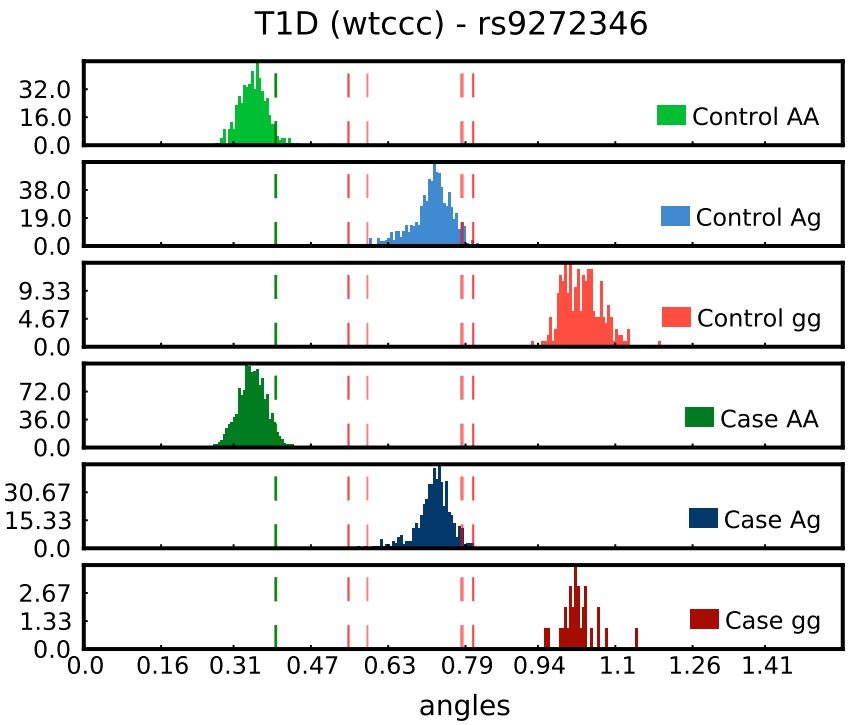

Figure 4.12: Histogram of angles with the decision boundaries indicated for a SNP with a negatively associated rare allele. A decision boundary marked red corresponds to a 'greater than or equal to' decision while a decision boundary marked green corresponds to a 'less than' decision.

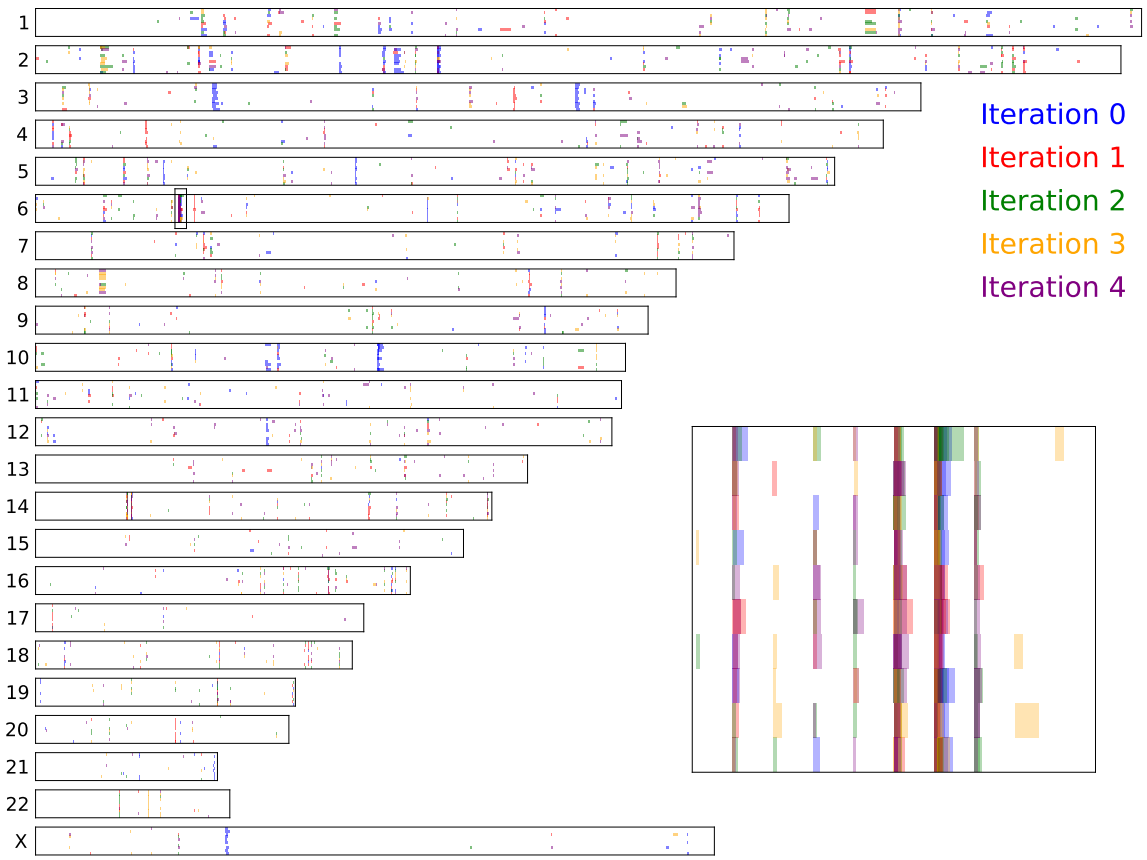

Figure 4.13: This figure illustrates the genomic locations of predictive SNPs selected by boosting. Each horizontal panel corresponds to a chromosome, with its length proportional to the number of base pairs in the chromosome. The x-axis of each panel is divided into 1 Mb blocks while the y-axis corresponds to CV fold. A block is highlighted with a bar if a SNP in the genomic region corresponding to that block was selected by boosting.The width of the bar is proportional to the relative magnitude of the weight of the SNP in the ADT. Vertical bands indicate that the genomic region was robustly selected by boosting to be predictive. Band thickness can be interpreted as the importance of that region in predicting disease. The inset figure shows a highly predictive region on chromosome 6 at $10\times$ resolution.

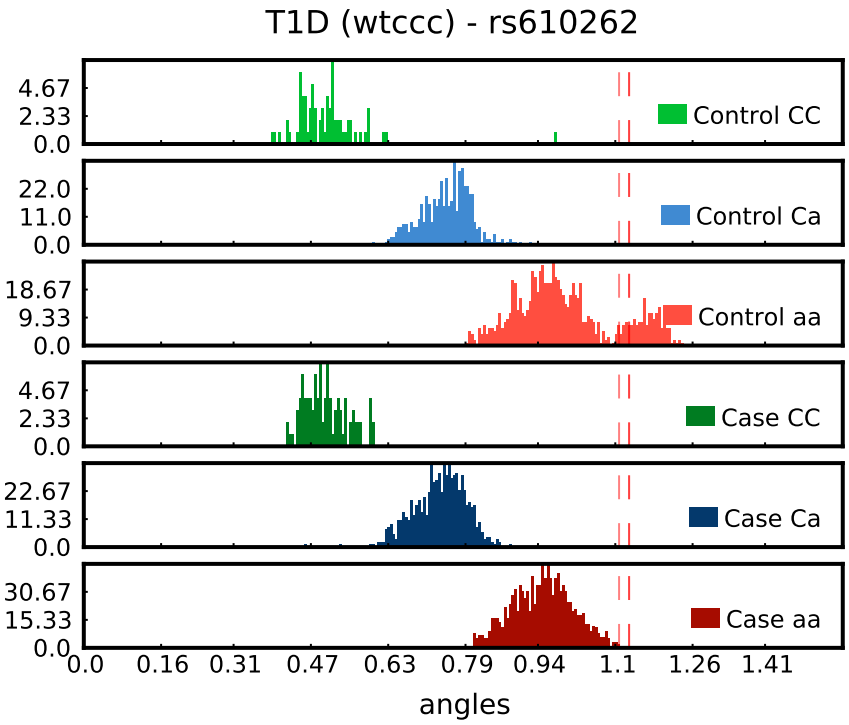

Figure 4.14: This plot compares the learned threshold angles for a SNP over 10-fold CV, with the histogram of angles for that SNP. A decision boundary marked red corresponds to a 'greater than or equal to' decision while a decision boundary marked green corresponds to a 'less than' decision. Note that while the number of cases and controls in each genotype do not differ appreciably, bimodality in the distribution observed for the controls but absent for the cases (a systematic difference) makes boosting susceptible to inferring decision boundaries that are not biologically meaningful.

Chapter 5

Inferring classification models with structured sparsity

5.1 Context

Supervised learning of sparse classification models from high–dimensional data is important to achieve both the engineering goal (prediction) and scientific goal (interpretability) in applications across biology, computer vision, medical imaging, and document and speech analysis. Black-box models learned using algorithms like support vector machines (SVM) and kernel density estimation often suffice for applications where high prediction accuracies are more important than the ability to interpret the classifier. Examples of such problems include predicting spam email and bot programs for web-based applications, and identifying faces in an image. However, several problems, particularly in medicine and biology, demand algorithms that learn interpretable models, where constraining the number of predictors in the model typically serves as a proxy for interpretability. Furthermore, in spite of the ubiquity of fast, cheap sensors, data collection in some domains (e.g., medical imaging and whole genome sequencing) is still relatively expensive, necessitating fast, efficient algorithms to learn sparse classifiers from a small number of samples, allowing subsequent measurements to be more problem-specific, and thus speeding up the decision process.

The traditional approach to learning structured or sparse classification models has fo-

cused on inferring a small number of features from a high-dimensional feature space that are strongly predictive of some output variable of interest. Adaboost, a popular large-margin iterative classification algorithm, constrains model complexity by using generalization error on held-out data to stop the algorithm early. It has been shown [Rosset *et al.*, 2004] that in the limit of infinitesimally small coordinate descent step sizes, early stopping of Adaboost effectively penalizes the exponential loss function with an l_1-norm of the weights[1] in the alternating decision tree. Another popular approach, used by algorithms like LASSO and its extensions, is to explicitly penalize the classification (or regression) loss function using an l_1-norm of the model weights. An l_1-norm penalty is the tightest convex upper bound to the cardinality of non-zero elements in the weight vector [Chen *et al.*, 1998] — this penalty term helps shrink the weights of unimportant features to zero. Entropy-regularized LPboost [Warmuth *et al.*, 2008] explicitly regularizes Adaboost by penalizing the exponential loss with the l_1-norm of the model weights.

Given a set of binary labeled examples $\{(\mathbf{y}_1, z_1), \ldots, (\mathbf{y}_N, z_N)\}$, each represented by a feature vector $\mathbf{y} \in \mathbb{R}^D$ and a label $z \in \{1, -1\}$, minimizing the hinge loss (used in SVMs) regularized by an l_1-norm on the model weights can be posed as a constrained convex optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\xi}, \mathbf{w}, b} \quad & \tfrac{1}{N} \textstyle\sum_n \xi_n + \mu \|\mathbf{w}\|_1 \\ \text{subject to} \quad & \xi_n \geq 0 \\ & \xi_n \geq 1 - z_n(\mathbf{w}^{\mathrm{T}} \mathbf{y}_n + b) \end{aligned} \tag{5.1}$$

where $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$ define the classification decision boundary, $\boldsymbol{\xi} \in \mathbb{R}^N$ quantifies the hinge loss over examples, and μ quantifies the relative importance between model complexity and model accuracy. The hinge loss penalizes a model if the 'agreement' between its prediction and the true label for each example, $z_n(\mathbf{w}^{\mathrm{T}} \mathbf{y}_n + b)$, is less than 1 (see figure 4.4 for a comparison with other loss functions). By introducing D new variables $\mathbf{v}$, we can

[1] The l_p-norm of a vector $\mathbf{w}$ is defined as $\|\mathbf{w}\|_p = (\sum_d |w_d|^p)^{\frac{1}{p}}$.

pose equation 5.1 as an equivalent linear program (LP).

$$\begin{aligned} \min_{\xi,\mathbf{w},\mathbf{v},b} \quad & \tfrac{1}{N}\textstyle\sum_n \xi_n + \mu \sum_d v_d \\ \text{subject to} \quad & -v_d \leq w_d \leq v_d \\ & \xi_n \geq 0 \\ & \xi_n \geq 1 - z_n(\mathbf{w}^{\mathrm{T}}\mathbf{y}_n + b) \end{aligned} \tag{5.2}$$

When either D or N is very large, a more efficient algorithm for solving such problems involve using first-order, projected subgradient methods, similar to that used by Pegasos [Shalev-Shwartz *et al.*, 2007]. Note, however, that theoretical guarantees for such algorithms require the hinge loss to be replaced by a strongly convex loss function like the huberized hinge-loss or the logistic loss. Despite the availability of extremely fast and efficient implementations [Efron *et al.*, 2004] [Beck and Teboulle, 2009], algorithms based on l_1-norm penalty are inherently greedy and penalize individual feature weights, and are not the appropriate tools when there are strong correlations between features, as is often observed in biological applications. For example, a classifier learned using boosting or LASSO that predicts disease state from gene expression data will be unable to infer genetically different yet functionally equivalent molecular changes that cause disease.

5.2 Structured-sparsity inducing norms

To address this limitation, several statistical models have been developed aimed at using prior knowledge of structured relationships among features. These models have primarily focused on more general normed penalizations of the classification loss function; examples include l_1/l_2-norm [Yuan and Lin, 2006], l_1/l_∞-norm [Quattoni *et al.*, 2009] and tree-structured norms [Jenatton *et al.*, 2010].

The data available for many applications exhibit a strong correlation structure among groups of features. If the set of features can be partitioned into correlated groups, or if some prior problem-relevant partition of the set of features is known, one regularization scheme that has proven effective [Meier *et al.*, 2008] is to penalize the loss function using an l_1/l_2 norm of the model parameters, defined by the known group structure. The relevant

optimization problem, for the hinge loss, can then be written as

$$\begin{aligned} \min_{\xi,\mathbf{w},b} \quad & \tfrac{1}{N}\textstyle\sum_n \xi_n + \mu \sum_{\mathrm{g}\in\mathrm{G}} \|\mathbf{w}_g\|_2 \\ \text{subject to} \quad & \xi_n \geq 0 \\ & \xi_n \geq 1 - z_n(\mathbf{w}^{\mathrm{T}}\mathbf{y}_n + b) \end{aligned} \tag{5.3}$$

where the elements of G are sets of feature indices belonging to each group and $\mathbf{w}_{\mathrm{g}}$ is a 'subvector' containing elements of $\mathbf{w}$ whose indices belong to g. By introducing K new variables $\{v_1, \ldots, v_K\}$, where $|\mathrm{G}| = K$ is the number of groups, we can pose equation 5.3 as a second-order cone program:

$$\begin{aligned} \min_{\xi,\mathbf{v},\mathbf{w},b} \quad & \tfrac{1}{N}\textstyle\sum_n \xi_n + \mu \sum_k v_k \\ \text{subject to} \quad & \|\mathbf{w}_{\mathrm{G}_k}\|_2 \leq v_k \\ & \xi_n \geq 0 \\ & \xi_n \geq 1 - z_n(\mathbf{w}^{\mathrm{T}}\mathbf{y}_n + b) \end{aligned} \tag{5.4}$$

Alternatively, if the relational structure between features can be specified in terms of a graph [Jacob *et al.*, 2009], we could penalize the loss function by the difference between weights of related features $\sum_{i,j} A_{ij}|w_i - w_j|^p$, where $\mathbf{A}$ is the adjacency matrix specifying a weighted graph between features and p specifies the norm being used. The relevant optimization problem can be specified as

$$\begin{aligned} \min_{\xi,\mathbf{w},b} \quad & \tfrac{1}{N}\textstyle\sum_n \xi_n + \mu \sum_{d,d'} A_{dd'}|w_d - w_{d'}|^p \\ \text{subject to} \quad & \xi_n \geq 0 \\ & \xi_n \geq 1 - z_n(\mathbf{w}^{\mathrm{T}}\mathbf{y}_n + b). \end{aligned} \tag{5.5}$$

In the case where $p = 2$, we can use equation 2.1 to rewrite the regularization term in equation 5.5 in terms of the associated graph Laplacian. This allows us to pose equation 5.5 as a quadratic program.

$$\begin{aligned} \min_{\xi,\mathbf{w},b} \quad & \tfrac{1}{N}\textstyle\sum_n \xi_n + \mu \mathbf{w}^{\mathrm{T}}\boldsymbol{\Delta}\mathbf{w} \\ \text{subject to} \quad & \xi_n \geq 0 \\ & \xi_n \geq 1 - z_n(\mathbf{w}^{\mathrm{T}}\mathbf{y}_n + b). \end{aligned} \tag{5.6}$$

In the case where $p = 1$ in equation 5.5, we need to introduce two sets of variables $\mathbf{w}^+$ and $\mathbf{w}^-$ such that $w_d = w_d^+ - w_d^-,\ w_d^+ \geq 0, w_d^- \geq 0\ \forall d$. We can then upper bound the absolute value term using triangle inequality as

$$\begin{aligned} |w_i - w_j| &= |(w_i^+ + w_j^-) - (w_i^- + w_j^+)| \\ &\leq w_i^+ + w_j^+ + w_i^- + w_j^- \end{aligned} \tag{5.7}$$

allowing us to pose equation 5.5 as an equivalent linear program

$$\begin{aligned} \min_{\xi, \mathbf{w}^+, \mathbf{w}^-, b} \quad & \tfrac{1}{N}\textstyle\sum_n \xi_n + 2\mu \sum_d (w_d^+ + w_d^-) \\ \text{subject to} \quad & w_d^+ \geq 0, w_d^- \geq 0 \\ & \xi_n \geq 0 \\ & \xi_n \geq 1 - z_n((\mathbf{w}^+ - \mathbf{w}^-)^{\mathrm{T}}\mathbf{y}_n + b). \end{aligned} \tag{5.8}$$

All of these approaches, however, assume the availability of prior knowledge about relationships between features and produce results that are strongly dependent on the graph or group structure being used. In addition to the choice of norm, application of such techniques to biological problems require the choice of an appropriate graph or group structure among features (e.g., gene regulatory networks). In the absence of such information, one regularization scheme that has proven successful in some applications is the elastic net [Zou and Hastie, 2005] – penalizing the loss function using both the l_1-norm and l_2-norm of the model parameters.

$$\begin{aligned} \min_{\xi, \mathbf{w}, b} \quad & \tfrac{1}{N}\textstyle\sum_n \xi_n + \nu(\mu\|\mathbf{w}\|_1 + (1-\mu)\|\mathbf{w}\|_2^2) \\ \text{subject to} \quad & \xi_n \geq 0 \\ & \xi_n \geq 1 - z_n(\mathbf{w}^{\mathrm{T}}\mathbf{y}_n + b) \end{aligned} \tag{5.9}$$

Similar to equation 5.2, by introducing D variables, we can rewrite equation 5.9 as a quadratic program. Penalizing the loss using both the l_1-norm and l_2-norm of the weights encourages correlated features to get similar weights while enforcing sparsity in the model [Wang *et al.*, 2006]. The tuning parameter ν quantifies the relative importance between fitting to the data and regularizing the model, while the tuning parameter $\mu \in [0, 1]$ parameterizes a convex combination of the l_1-norm and l_2-norm penalties. Inferring group

structure among features, however, requires clustering the weights of the features as a post-processing step.

5.3 Structured sparsity using Bayesian inference : intuition

The goal of this chapter is to address a key limitation of techniques based on structured-sparsity inducing norms — the need for reliable prior knowledge on relevant relationships between elements of a very high dimensional-feature space. For biological applications, partial knowledge of such prior relationships is often readily available in the form of gene ontologies, protein interaction networks and biological pathways. However, these large relational networks often represent strict binary relationships and are often biased toward genes or proteins that have been thoroughly studied in the literature, making novel discoveries using these networks relatively rare for certain phenotypes.

In order to predict observables such as disease state, one then needs to choose disease-relevant relationships between features (e.g., the protein interactions and biological pathways relevant to predicting diabetes) to avoid norm-based algorithms from incorrectly penalizing the classification loss function. Furthermore, protein networks and pathways often capture unrealistic binary relationships between genes, while it is well-known that a specific gene or protein can have different degrees of participation (and importance) in different pathways or complexes, each of which may have a different relevance to the phenotype being studied. For example, the *Wnt* signaling pathway, while well-known for its role in carcinogenesis, is also important for embryogenesis, morphogenic signaling and adult hippocampal neurogenesis in different organisms.

A more meaningful constraint is a weighted relational network that is problem-specific, where the weights capture both the importance of a protein-protein association and the uncertainty involved in measuring such associations. Absent such information, an alternative approach would be to infer a problem-relevant grouping of the features that increases prediction accuracy; this grouping could be inferred either from the data used for learning or from available network information.

Bayesian inference provides a principled and interpretable way to model such hid-

den group structure among features by inferring a latent variable for each feature, whilst classifying binary-labeled examples. Such a latent variable captures a relaxed relational structure as opposed to a strict, binary structure usually used in norm-based regularization schemes. Furthermore, the latent group structure should ideally capture statistical correlations and functional associations between features that aid in improved classification accuracy.

A Bayesian approach to model the observed data requires specifying a factorization of the joint distribution of all the variables involved in the generative process, usually represented as a probabilistic graphical model. A directed probabilistic graphical model [Bishop, 2007] (also called a Bayesian network) is a representation of the factorization of a joint probability distribution in the form of a directed acyclic graph, where the nodes of the graph correspond to variables in the joint distribution and a directed edge from variable Y to X indicates a conditional dependence of X on Y. Intuitively, besides the observed features and class labels, the variables involved would include the parameters of the classifier, the parameters of the distribution from which the features are drawn and the latent group assignment of the features.

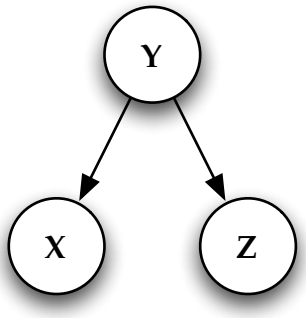

Figure 5.1: Shown here is an example of a directed graphical model. The joint distribution $p(X, Y, Z)$ factorizes according to this graphical model as $p(X, Y, Z) = p(X|Y)p(Z|Y)p(Y)$.

The generative model can then be informally described as:

1. roll a biased K-sided die to assign features to groups, where K is the number of groups

2. conditioned on group assignment, draw

   (a) parameters for feature mixture distribution

   (b) classifier weight for feature

3. draw elements of the feature vector from their appropriate mixture distributions

4. draw class label, given the classifier weights and feature vector

For example, if we are interested in discriminating between a cell from a tumor and a normal cell based on gene expression, the features would correspond to genes and feature groups could correspond to tumor relevant pathways. Examples include the p53 pathway (cell cycle regulation), RAS signaling pathway (cell growth and differentiation) and hypoxia pathway (regulating oxygen intake). The weights on genes quantify how important they are in discriminating between tumor and normal cells and are conditioned on the pathway they belong to. The up- or down- regulation of each gene is also conditional on the pathway they belong to (i.e., whether the relevant pathway has been 'up-regulated' or 'down-regulated'). Conditioned on the expression values of the genes and the weights assigned to them, we can then determine if a cell is normal or from a tumor.

5.4 Group structured classification model

Given observation pairs $\{(\mathbf{y}_1, z_1), \ldots, (\mathbf{y}_N, z_N)\}$ that include a feature vector $\mathbf{y} \in \mathbb{R}^D$ and a class label $z \in \{0, 1\}$, we model the observed sample features using a hierarchical Bayesian model, where parameters of feature distributions are treated as random variables drawn from group-specific distributions, while simultaneously modeling the observed sample labels. This is in contrast to discriminative learning where the goal is to simply learn the decision boundary that best separates observed feature vectors that have different labels and can accurately predict the label of unseen observations. Here, $p(z|\mathbf{y})$ is specified using the logistic model with $\mathbf{w}$ and b being the classifier weights (logistic regression coefficients) and the offset term respectively. For example, $\mathbf{y}$ can represent the expression of a set of genes for a cell while z represents a quantized binary cellular phenotype like oncogenicity, tissue type or cell fate. Each element of the observed feature vector $\mathbf{y}$ is modeled using

mixture distributions where the form of the distribution depends on application-specific assumptions and all the features are assumed to be drawn from distributions of the same functional form. For example, if the observed features are assumed to be drawn from some symmetric, fat-tailed distribution, they can be modeled using a mixture of Student distributions.

The key element of the group structured classification model (GSCM) is the assumption that the parameters of the feature-specific distributions and the weights of features in the classifier are each themselves drawn from mixture distributions where the mixture (group) membership is an attribute of a feature. Then, given some data, inference using this model returns a posterior distribution over groups for each feature and a distribution of weights for features in each group, with the presence of group structure among features inducing an appropriate grouping of feature weights. This model allows inference of group structure among features that is problem-relevant, constrains the complexity of the classifier and encompasses a larger set of relational patterns among features, including disjoint groups, overlapping groups, trees and networks.

For the GSCM, let D denote the number of features, N denote the number of samples, K denote the number of latent feature-groups and M denote the number of per-group mixture components. Let us also specify $\boldsymbol{\pi}$ to be a K-dimensional parameter of a multinomial distribution over groups, $\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_K\}$ to be the hyperparameters of K Dirichlet distributions on the M-dimensional simplex, $\{\theta\}$ to be the hyperparameters of the MK distributions conjugate to problem-specific distributions from which the features are drawn, $\{(\mu_1^w, \tau_1^w), \ldots, (\mu_K^w, \tau_K^w)\}$ to be the hyperparameters of the Gaussian distribution over feature weights in the classifier for each of the K groups and (μ^b, τ^b) to be the hyperparameters of the Gaussian distribution over the classifier offset term. Note that, in this chapter, the Gaussian distribution will be parameterized by its mean μ and precision τ.

Let $[\![\cdot]\!]$ be a unary operator that returns the non-zero indices of a vector, $\mathcal{N}(\cdot)$ denote the univariate Gaussian distribution (parameterized by a mean and precision), $\mathcal{B}(\cdot)$ the Bernoulli distribution $\mathrm{Mult}(\cdot)$ the multinomial distribution, $\mathrm{Dir}(\cdot)$ the Dirichlet distribution, $\mathcal{Z}(\cdot)$ some problem-specific distribution from which the features are drawn and $\hat{\mathcal{Z}}(\cdot)$ its conjugate, $\sigma(\cdot)$ the logistic function and $a_n = \sum_d y_{nd} w_d + b$. The generative process

under the GSCM can be specified as:

1. For $d \in \{1, \ldots, D\}$,
   (a) draw group assignments for features $x_d \sim \text{Mult}(\boldsymbol{\pi})$,
   (b) conditioned on group assignment, draw feature weights $w_d | x_d \sim \mathcal{N}(\mu^w_{[\![x_d]\!]}, \tau^w_{[\![x_d]\!]})$,
   (c) conditioned on group assignment, draw parameters of the feature distribution
      i. $\boldsymbol{\psi}_d | x_d \sim \text{Dir}(\boldsymbol{\alpha}_{[\![x_d]\!]})$,
      ii. for $m \in \{1, \ldots, M\}$, $\zeta_{dm} | x_d \sim \hat{\mathcal{Z}}(\theta_{m[\![x_d]\!]})$.
2. Draw classifier offset $b \sim \mathcal{N}(\mu^b, \tau^b)$.
3. For $n \in \{1, \ldots, N\}$,
   (a) draw states for features $s_{nd} | \boldsymbol{\psi}_d \sim \text{Mult}(\boldsymbol{\psi}_d) \; \forall \; d \in \{1, \ldots, D\}$,
   (b) draw features for a sample $y_{nd} | s_{nd}, \boldsymbol{\zeta}_d \sim \mathcal{Z}(\zeta_{d[\![s_{nd}]\!]}) \; \forall \; d \in \{1, \ldots, D\}$,
   (c) draw label for the sample $z_n | \mathbf{y}_n, \mathbf{w}, b \sim \mathcal{B}(\sigma(a_n))$.

A directed graphical model depicting the factorization of the joint distribution specified by this model is shown in Figure 5.2a.

The latent group variables $x_d \in \{0, 1\}^K$ and latent states $s_{nd} \in \{0, 1\}^M$ are vectors whose non-zero index denotes the group (or state) being assigned. In a slight abuse of notation, these variables will not be denoted using boldface. The choice of distribution $\mathcal{Z}(\cdot)$ depends on the type of features being used in the problem. For example, when classifying tumors using gene expression, the features could be modeled using a log-normal distribution, while classifying tumors using genotypes requires modeling the features using a multinomial distribution. In this chapter, for simplicity, we assume $\mathcal{Z}(\cdot)$ to be the Gaussian distribution $\mathcal{N}(\mu, \tau)$ and $\hat{\mathcal{Z}}(\cdot)$ to be the Gaussian-gamma distribution. Note, however, that the choice of distribution $\mathcal{Z}(\cdot)$ depends strongly on application-specific assumptions about the data.

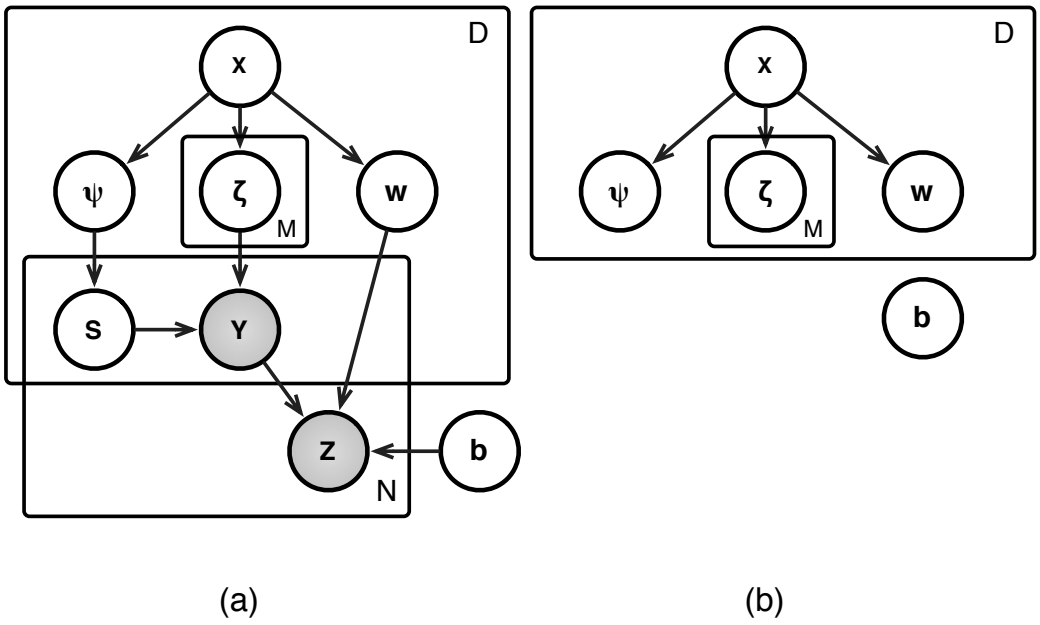

Figure 5.2: (a) Graphical model representation of the group structured classification model. The latent variable x captures group structure among features by explicitly modeling them, while constraining the classifier weights. (b) Graphical model representing the structured factorization of the variational distribution over model parameters.

5.5 Posterior inference of GSCM

Since the exact posterior distribution is intractable to compute, we make use of variational Bayesian methods to obtain efficient, structured approximations of the true posterior distribution. Variational Bayesian methods assume an appropriately factorized distribution of the latent variables, parameterized by free variables called variational parameters. The goal is to find the optimal variational parameters that minimize the Kullback-Leibler (KL) divergence between the true and approximate posterior. Minimizing this KL divergence is equivalent to maximizing a lower bound to the log probability of the data [Beal, 2003].

Using Jensen's inequality, we can lower bound the evidence as:

$$\begin{aligned}
\mathcal{E} &= \log p(z, \mathbf{y}) = \log \sum_{\mathbf{x},\mathbf{s}} \int p(z, \mathbf{y}, \mathbf{s}, \mathbf{w}, b, \boldsymbol{\psi}, \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{x})\, d\mathbf{w}\, db\, d\boldsymbol{\psi}\, d\boldsymbol{\mu}\, d\boldsymbol{\tau} \\
&\geq \mathbf{E}_q \left[\log p(z, \mathbf{y}, \mathbf{s}, \mathbf{w}, b, \boldsymbol{\psi}, \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{x})\right] + \mathbf{H}[q] \\
&= \sum_n \mathbf{E}_q \left[\log p(z_n | \mathbf{y}_n, \mathbf{w}, b)\right] + \sum_{n,d} \mathbf{E}_q \left[\log p(y_{nd} | s_{nd}, \mu_d, \tau_d)\right] \\
&+ \mathbf{E}_q \left[p(\mathbf{s}_n, \mathbf{w}, b, \boldsymbol{\psi}, \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{x})\right] + \mathbf{H}[q] \qquad (5.10)
\end{aligned}$$

where the expectation is taken with respect to the assumed variational distribution of the latent variables and $\mathbf{H}[q]$ denotes the entropy of that distribution. The factorization of the joint distribution as shown in equation 5.10 is specified by the graphical model in figure 5.2a.

Evaluating the first term of the lower bound in equation 5.10 involves computing the expectation of the log-sigmoid – an intractable problem even when the variational distribution is a simple Gaussian. We resolve the intractability of computing $\mathbf{E}_q \left[\log \sigma(a)\right]$ by locally approximating the sigmoid function $\sigma(\cdot)$ as a Gaussian using a free variational parameter [Bishop, 2007].

$$\log(\sigma(a)) \geq \log(\sigma(\widetilde{\eta})) + \frac{1}{2}(a - \widetilde{\eta}) - \lambda(\widetilde{\eta})(a^2 - \widetilde{\eta}^2) \qquad (5.11)$$

where $\widetilde{\eta}$ is a free variational parameter and $\lambda(\widetilde{\eta}) = \frac{1}{2\widetilde{\eta}}(\sigma(\widetilde{\eta}) - \frac{1}{2})$. In this chapter, all free variational parameters will be denoted using a tilde. Thus,

$$\begin{aligned}
p(z_n | \mathbf{y}_n, \mathbf{w}, b) &= \sigma(a_n)^{z_n} (1 - \sigma(a_n))^{(1-z_n)} \\
&= \sigma(a_n) \left(\frac{1 - \sigma(a_n)}{\sigma(a_n)}\right)^{(1-z_n)} \\
&= \sigma(a_n) \exp(a_n(z_n - 1)) \qquad (5.12) \\
\log p(z_n | \mathbf{y}_n, \mathbf{w}, b) &\geq a_n(z_n - 1) + \log(\sigma(\widetilde{\eta}_n)) \\
&+ \frac{1}{2}(a_n - \widetilde{\eta}_n) - \lambda(\widetilde{\eta}_n)(a_n^2 - \widetilde{\eta}_n^2) \qquad (5.13)
\end{aligned}$$

where $a_n = \sum_d w_d y_{nd} + b$. Note that $\widetilde{\eta}_n$ are extensive parameters (i.e., they scale with the number of examples) whose optimal values, computed by maximizing the expectation of the lower bound in equation 5.13 under the variational distribution, are given as $\widetilde{\eta}_n^2 = \mathbf{E}_q \left[a_n^2\right]$.

We assume a structured factorization of the variational posterior distribution (see Figure 5.2b):

$$q(\mathbf{s}_n, \boldsymbol{\psi}, \mathbf{w}, b, \boldsymbol{\mu}, \boldsymbol{\tau}, \mathbf{x}) \approx q(b) \prod_d \left\{ q(s_{nd}) q(\psi_d|x_d) q(w_d|x_d) \left(\prod_m q(\mu_{dm}, \tau_{dm}|x_d) \right) q(x_d) \right\}. \tag{5.14}$$

A structured factorization of the posterior distribution on model parameters is essential to reduce the number of free variational parameters and capture group structure among features, achieving an interpretable dimensionality reduction. The variational distributions, specified by free parameters of appropriate dimensionality, have the forms

$$q(s_{nd}) = \mathrm{Mult}(\widetilde{\psi}_{nd}),\ \widetilde{\psi}_{nd} \in \mathbb{S}^M \tag{5.15}$$

$$q(\boldsymbol{\psi}_d|x_d) = \mathrm{Dir}(\widetilde{\boldsymbol{\alpha}}_{[\![x_d]\!]}),\ \widetilde{\boldsymbol{\alpha}}_k \in \mathbb{R}_+^M \tag{5.16}$$

$$q(w_d|x_d) = \mathcal{N}(\widetilde{\mu}^w_{[\![x_d]\!]}, \widetilde{\tau}^w_{[\![x_d]\!]}),\ \widetilde{\mu}^w_k \in \mathbb{R},\ \widetilde{\tau}^w_k \in \mathbb{R}_+ \tag{5.17}$$

$$q(b) = \mathcal{N}(\widetilde{\mu}^b, \widetilde{\tau}^b),\ \widetilde{\mu}^b \in \mathbb{R},\ \widetilde{\tau}^b \in \mathbb{R}_+ \tag{5.18}$$

$$q(\mu_{dm}, \tau_{dm}|x_d) = \mathcal{N}(\widetilde{\nu}_{m[\![x_d]\!]}, \widetilde{\rho}_{m[\![x_d]\!]}\tau_{dm})\mathrm{Gam}(\widetilde{\beta}_{m[\![x_d]\!]}, \tilde{\gamma}_{m[\![x_d]\!]}),$$
$$\widetilde{\nu}_{mk} \in \mathbb{R},\ \widetilde{\rho}_{mk}, \widetilde{\beta}_{mk}, \tilde{\gamma}_{mk} \in \mathbb{R}_+ \tag{5.19}$$

$$q(x_d) = \mathrm{Mult}(\widetilde{\boldsymbol{\pi}}_d),\ \widetilde{\boldsymbol{\pi}}_d \in \mathbb{S}^K \tag{5.20}$$

where $\mathrm{Gam}(\cdot)$ denotes the gamma distribution and $\mathbb{S}^K$ is the K-dimensional unit simplex. Having specified the functional forms for the variational distributions, the lower bound for the evidence can then be written out in terms of the variational parameters and hyperparameters (see Appendix B).

Given a set of observations, equation 5.10 can be optimized with respect to the variational parameters using the variational Bayesian expectation-maximization (VBEM) algorithm – an iterative coordinate ascent algorithm that optimizes each parameter while holding the other parameters fixed. In the E-step, we maximize the bound by computing optimal values for the extensive parameters $\widetilde{\eta}$ and $\widetilde{\psi}$. In the M-step, we compute optimal values for the intensive parameters $\widetilde{\alpha}$, $\widetilde{\mu}^w$, $\widetilde{\tau}^w$, $\widetilde{\mu}^b$, $\widetilde{\tau}^b$, $\widetilde{\nu}$, $\widetilde{\rho}$, $\widetilde{\beta}$, $\tilde{\gamma}$ and $\widetilde{\pi}$. The E-step and M-step are repeated until the bound on the evidence converges. The update equations for the VBEM algorithm are given in Appendix B.

5.6 Experimental results

In order to illustrate the GSCM, we performed numerical experiments on synthetic data generated from class-conditional distributions. The stochastic process for generating binary labeled data was designed to simulate the group structure among sample features that we would like to infer using the GSCM. Specifically, for a given value of ι and ϵ (described below), the data were generated using the following stochastic process:

1. Specify a K-dimensional multinomial parameter $\overline{\boldsymbol{\pi}}$, and K pairs $\{(\overline{\nu}_1, \overline{\beta}_1), \ldots, (\overline{\nu}_K, \overline{\beta}_K)\}$ parameterizing the Gaussian-gamma distribution from which feature parameters are drawn.

2. For $d \in \{1, \ldots, D\}$, assign features to groups $x_d \sim \mathrm{Mult}(\overline{\boldsymbol{\pi}})$.

3. For $n \in \{1, \ldots, N\}$, draw a binary label for each sample $z_n \sim \mathcal{B}(\bar{p})$, $z_n \in \{0, 1\}$.

4. Conditioned on group assignment, draw distribution parameters for each feature

   (a) $\hat{\mu}_d | x_d \sim \mathcal{N}(\overline{\nu}_{[x_d]}, \iota)$

   - $\overline{\mu}_{dm} = \hat{\mu}_d + (-1)^m \epsilon \overline{\tau}_{dm}$, for $m \in \{0, 1\}$

   (b) $\overline{\tau}_{dm} | x_d \sim \mathrm{Gam}(\overline{\beta}_{m[x_d]}, \iota)$, for $m \in \{0, 1\}$.

5. Conditioned on binary label, for each sample, draw features $y_{nd} | z_n \sim \mathcal{N}(\overline{\mu}_{dz_n}, \overline{\tau}_{dz_n})$, for $d \in \{1, \ldots, D\}$.

Here, $\mathcal{B}(\cdot)$ is the Bernoulli distribution and $\bar{p}$ measures the skewness between number of samples associated with each label. The strength of group structure among features, also called *identifiability*, can be tuned by ι and measured by $\mathbf{I}[x; \hat{\mu}]$, where $\mathbf{I}[\cdot; \cdot]$ denotes the mutual information (MI) between two random variables. The *separability* of the binary classes can be tuned by ϵ and measured by the average of $\mathbf{I}[y_d; z]$ over all features. By varying ι and ϵ, we generate data of varying degrees of identifiability and separability.

For different values of ι and ϵ, we generated 500 binary labeled data points in a 5000 dimensional feature space. The number of groups was set to be 4 and the parameters in the generative process were set as follows: $\overline{\boldsymbol{\pi}} = [0.5, 0.3, 0.15, 0.05]$, $\overline{\boldsymbol{\nu}} = [0, -3, -2, 1]$

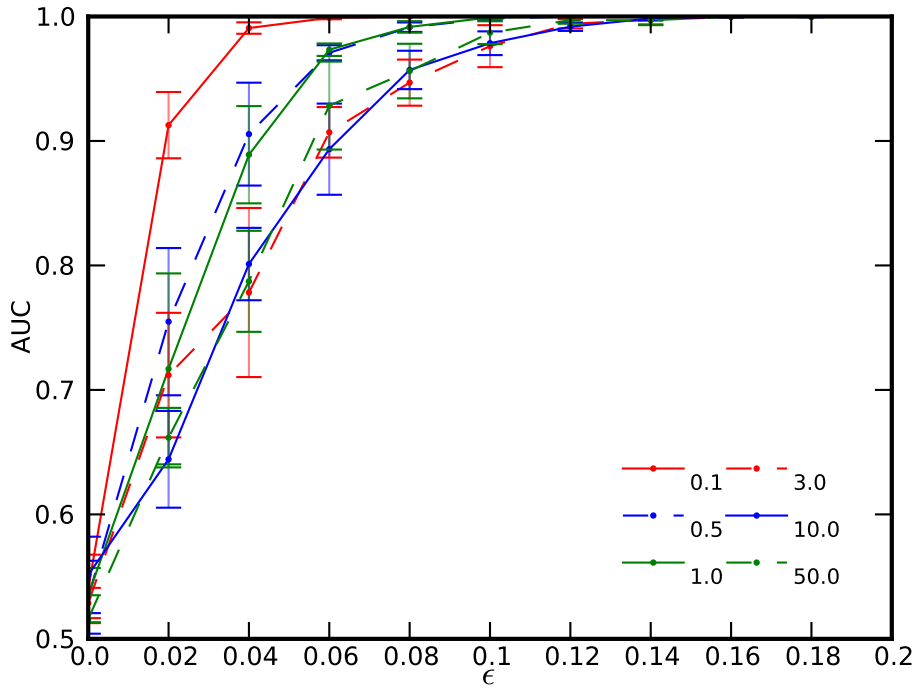

Figure 5.3: A plot of classifier AUC as a function of data separability ϵ, on held-out data averaged over 5-fold CV. Each curve corresponds to a different value of group identifiability of features ι.

and $\overline{\boldsymbol{\beta}} = [10000, 500, 100, 500]$. The parameter values were chosen to simulate a data set in which a large group for features are not discriminative of the sample labels. During inference, we specified the number of states $M = 3$ and the number of groups $K = 6$. The hyperparameters of the model during inference using the variational EM algorithm were specified to enforce flat prior distributions over parameters; i.e. μ^w, μ^b, ν, ρ and γ were set close to zero, $\alpha_{dm} = \frac{1}{M}\ \forall\ d, m$ and $\pi_{dk} = \frac{1}{K}\ \forall\ d, k$.

We quantify the accuracy of the classification model inferred using the variational EM algorithm by computing the area under the ROC curve (AUC) on held-out data. Specifically, given the inferred variational distributions $q(\mathbf{x})$, $q(\mathbf{w}|\mathbf{x})$ and $q(b)$, the expected decision boundary is given as $\mathbf{E}_{q(w_d)}[w_d] = \sum_k \widetilde{\pi}_{dk}\widetilde{\mu}^w_k$ and the expected classifier bias is $\widetilde{\mu}^b$. For a set of held-out data points $\{(\mathbf{y}_1, z_1), \ldots, (\mathbf{y}_l, z_l)\}$, we can compute the real-valued output

of the classifier as $f(\mathbf{y}_l) = \sum_{dk} \widetilde{\pi}_{dk} \widetilde{\mu}^w_k y_{ld} + \widetilde{\mu}^b$ and, subsequently, the AUC of the classifier given these real-valued outputs. Figure 5.3 compares the classifier AUC as a function of ϵ for different values of ι. As expected, we observe increasing classifier AUC with increase in class separability.

A measure of classification accuracy more appropriate within the current Bayesian framework would be $p(z_l = 1|\mathbf{y}_l)$ with $\mathbf{w}$ and b integrated out under their inferred variational distributions. While an exact computation is intractable, due to the presence of the log-sigmoid in $p(z_l = 1|\mathbf{y}_l)$, this expectation could be estimated by a simple sampling scheme. The classifier AUC measured in this manner, however, would not qualitatively change the results shown in Figure 5.3.

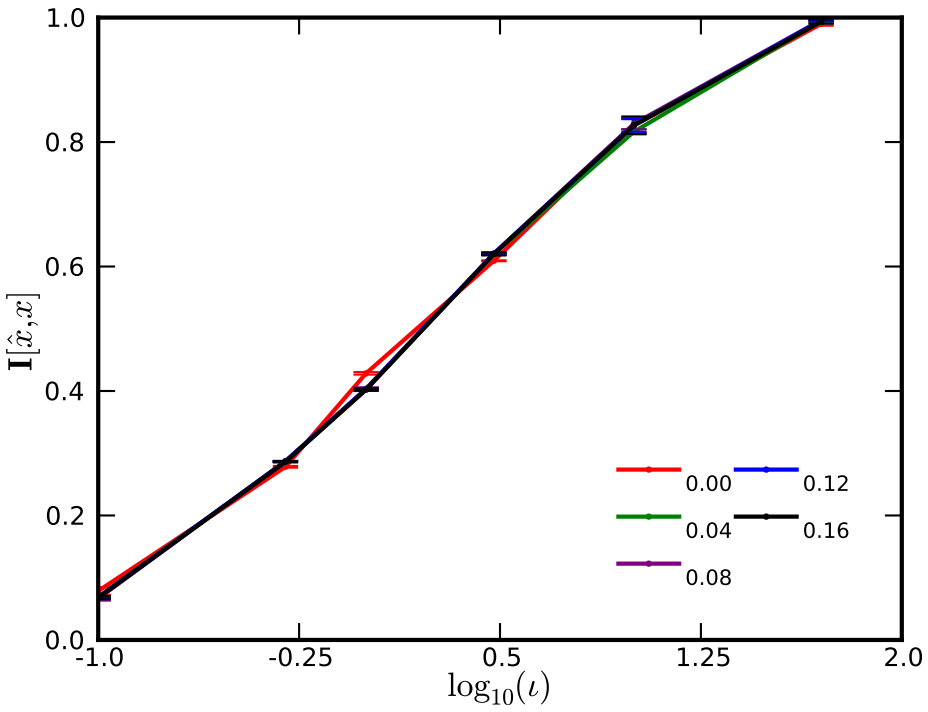

Figure 5.4: A plot of MI between the true group and inferred group assignments of the features, as a function of identifiability of group structure ι. Each curve corresponds to a different value of data separability ϵ.

Furthermore, we quantify the accuracy of the inferred posterior distribution over groups by computing the MI between the true group assignments x and the inferred group assignments $\hat{x}$. Specifically, the true group assignments can be represented by a probability distribution over groups, $\Pi_{dk} \equiv p(x_d = k) = 1$ if the d^{th} feature is assigned to group k and 0 otherwise. Then, given the inferred distribution over groups $\Lambda_{dk} \equiv q(\hat{x}_d = k)$, we can compute the joint distribution between true and inferred group assignments as $P(x_d = k, \hat{x}_d = k') \propto \sum_d \Pi_{dk}\Lambda_{dk'}$. The MI $\mathbf{I}[x; \hat{x}]$ computed using this joint distribution quantifies the decrease in uncertainty in identifying the true group of a feature given the inferred distribution over groups, and can be used as a measure for how accurately the algorithm infers the group structure among features. Figure 5.4 compares $\mathbf{I}[x; \hat{x}]$ as a function of ι for different values of ϵ. Again as expected, we observe increasing accuracy in inferring feature group structure with increase in group identifiability.

Finally, it would be useful to compare the performance of the GSCM with other popular algorithms based on sparsity-inducing norms on this synthetic data set. We chose algorithms that minimize the hinge-loss penalized using the l_1-norm on feature weights, l_1/l_2-norm specified by a disjoint partition of the features and the l_1, l_2 regularization as used in elastic net. Earlier in this chapter, we described how these norm-regularized hinge loss minimizations can be formulated as simple convex optimization problems. Specifically, an l_1-norm regularization can be formulated as a linear program, an elastic net regularization can be posed as a quadratic program and an l_1/l_2-norm regularization can be formulated as a second-order cone program. The l_1/l_2-norm is defined by the prespecified disjoint partition of the features given by their group assignments (computed during data generation).

From equations 5.2, 5.4 and 5.9, we see that these optimization problems have tuning parameters that quantify the importance of model complexity over model accuracy. We solved each optimization problem for parameter choices ranging over six orders of magnitude and selected the best parameter based on prediction accuracy (AUC) computed on held-out data, using 5-fold cross validation. Given the small problem sizes, the relevant convex optimization problems could be solved using standard solvers. Figure 5.5 compares the accuracy of the GSCM measured with the three algorithms, as a function of ϵ for

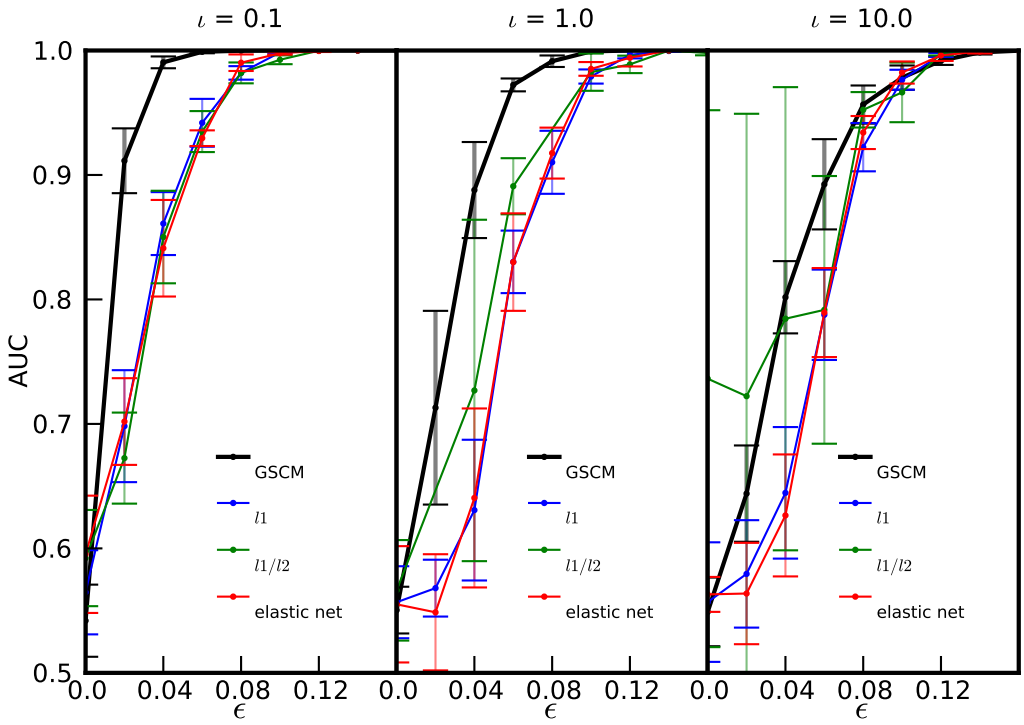

Figure 5.5: A plot comparing the AUC of the GSCM, inferred using variational Bayesian EM algorithm, against that of classifiers learned by minimizing the hinge-loss regularized by different norms on the classifier weights, on held-out data averaged over 5-fold CV. The AUC plotted for each norm-based classifier is the best accuracy achieved over a range of choices for the relevant tuning parameters.

different choices of ι. In addition to accurately inferring the latent group structure among the features, we see that the GSCM achieves a prediction accuracy that is comparable to or even better than the best accuracy of different norm-regularized algorithms.

5.7 Concluding remarks / Future directions

In this chapter, we have argued for a Bayesian approach to learning structured classification models that involves inferring latent group structure among the elements of the feature space whilst classifying binary-labeled data. We presented the GSCM that models a soft relational structure among features, a more realistic sparsity structure observed

in various applications, particularly in computational biology and genomics, as opposed to stricter constraints enforced by l_1 and l_1/l_2-regularized classification algorithms. It is, however, important to note that the GSCM makes explicit assumptions on the properties of the group structure it is designed to model — features belonging to the same group will have similar values.

In many applications, it would be more meaningful for features belonging to the same group to be strongly correlated with each other – a structure that the GSCM does not model explicitly. Motivated by the Stochastic Block Model [Holland and Leinhardt, 1976] [Hofman and Wiggins, 2008], the generative process for the Stochastic Group Model (SGM), one possible modification to the GSCM that models correlations, can be specified as follows:

1. Specify
   - a K-dimensional multinomial parameter $\boldsymbol{\pi}$,
   - two pairs of parameters $\{(a_+, b_+), (a_-, b_-)\}$ for beta distributions from which correlation coefficients of within-group feature pairs and between-group feature pairs are drawn,
   - D multinomial parameters $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_D\}$ for distributions over states for each of D features,
   - a set of DM pairs $\{(\mu_1, \tau_1), \ldots, (\mu_{DM}, \tau_{DM})\}$ parameterizing the state-conditional Gaussian distribution for the D features.
2. For each feature $d \in \{1, \ldots, D\}$,
   (a) assign the feature to a group $x_d \sim \text{Mult}(\boldsymbol{\pi})$
   (b) conditioned on group assignment, draw a weight for the feature $w_d | x_d \sim \mathcal{N}(\mu^w_{[\![x_d]\!]}, \tau^w_{[\![x_d]\!]})$
3. draw classifier bias $b \sim \mathcal{N}(\mu^b, \tau^b)$
4. For each pair of features $(d, d') \in \{1, \ldots, D\} \times \{1, \ldots, D\}, d > d'$
   (a) conditioned on group assignments, draw the correlation coefficient between the features

$$\begin{aligned} \Omega_{dd'}|x_d = x_{d'} &\sim \text{Beta}(a_+, b_+) \\ \Omega_{dd'}|x_d \neq x_{d'} &\sim \text{Beta}(a_-, b_-) \end{aligned} \tag{5.21}$$

5. For each example $n \in \{1, \ldots, N\}$,

   (a) draw states for features $s_{nd}|\boldsymbol{\psi}_d \sim \text{Mult}(\boldsymbol{\psi}_d)\ \forall\ d \in \{1, \ldots, D\}$

   - compute the mean vector $\hat{\boldsymbol{\mu}} \doteq \hat{\mu}_d = \mu_{d[\![s_{nd}]\!]}\ \forall\ d \in \{1, \ldots, D\}$

   (b) draw feature vector $\mathbf{y}_n|\mathbf{s}_n, \boldsymbol{\mu}, \boldsymbol{\tau}, \boldsymbol{\Omega} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \text{diag}(\boldsymbol{\tau})\boldsymbol{\Omega}^{-1}\text{diag}(\boldsymbol{\tau}))$

   (c) draw label for the sample $z_n|\mathbf{y}_n, \mathbf{w}, b \sim \mathcal{B}(\sigma(a_n))$

where $\text{Beta}(\cdot)$ denotes the beta distribution on the domain $[-1, 1]$, $\text{diag}(\cdot)$ is the diagonal matrix formed from its vector-valued argument, $\boldsymbol{\Omega}$ is the symmetric correlation matrix representing pairwise correlations between features and $\text{diag}(\boldsymbol{\tau})\boldsymbol{\Omega}^{-1}\text{diag}(\boldsymbol{\tau})$ is the precision matrix for the multivariate Gaussian distribution. Note that unlike the GSCM, the group assignment variable x in the SGM induces a block diagonal structure in the correlation matrix $\boldsymbol{\Omega}$ modeling correlated groups of features.

While the generative process for the SGM was specified for real-valued observation vectors $\mathbf{y}$, it can be easily extended to observation vectors representing counts of subsequences (for the problem discussed in Chapter 3) and observations representing SNP signal intensities (for the problem discussed in Chapter 4) by appropriately specifying a joint distribution between pairs of features. Inference using this model on real gene expression data and SNP intensity data, and extension of this Bayesian framework to use relational graphs like protein-protein interaction networks are extremely promising avenues for future research that we hope to pursue.

Chapter 6

Future work

We have presented large-scale supervised and unsupervised machine learning techniques applied to a variety of problems in biology. In the first part of this thesis, we have shown how the cost-functions being optimized in spectral graph partitioning can be well approximated by the the rate of loss of predictive information on random walkers for fast-mixing graphs. Using this approximation, we derived an equivalence between the regularized cost functions widely used in the image segmentation community and the relevant information as used in the Information Bottleneck method.

Following this, we described two seemingly different applications in biology — predicting viral host from sequence information and predicting disease phenotype from whole genome sequence variations — within the framework of sparse supervised machine learning. To predict the host of a virus, we used multiclass Adaboost, a powerful large-margin classification algorithm, to learn alternating decision trees built from simple decision rules based on sequence motifs. The sequence motifs incorporated into the model were found to be strongly conserved among viruses sharing a common host, suggesting functional relevance for these subsequences. For the problem of predicting disease phenotype, we used Adaboost and its l_1-regularized variant to learn one-sided alternating decision trees directly from measurements of whole genome single nucleotide polymorphisms. The single-SNP decision rules in the model identify predictive 'host-spots' in the genome that contain putative causal variants for the disease. The decision boundaries inferred for each SNP in the model capture non-additive effects, suggesting dominance and epistatic interactions

that are relevant for the disease.

Natural extensions to our work on predicting hosts of viruses to disease relevant problems include applying this technique to different strains of the Influenza virus, where the host label could be avian, swine or human. This technique can also be used to discover sequence elements that are predictive of oncogenicity of cells in human. As discussed in Chapter 3, one way to enforce structured model complexity is to penalize the boosting loss function using a graph-induced or group-induced norm of the vector of model weights. Developing a structured learning framework which infers overlapping groups of predictive k-mers, represented as position-specific scoring matrices (PSSM) [Hertz and Stormo, 1999], and assigns similar weights to k-mers in the same group is an exciting avenue for research.

Our work on bringing powerful machine learning tools to genome-wide association studies suggests several promising directions for future research. Despite the ability of decision rules to naturally encode biologically meaningful angle decision boundaries, the greedy, coordinate-descent property of Adaboost makes it susceptible to learning decision boundaries that do not seem biologically informative. Indeed, one way to resolve this would be to develop robust genotype inference algorithms, posed as inference of a latent state in a hierarchical Bayesian model, and applying Adaboost with alternating decision trees on the inferred distribution over genotypes. Additionally, given the strikingly similar performance of rather different models learned by Adaboost and ERLPboost, using trees and stumps, one straightforward approach to achieving biologically relevant sparse models involves guiding the tree-growing process in Adaboost using external relational information specified by protein interaction networks and gene ontologies. Specifically, we can restrict the space of decision rules that can be added to an ADT to ensure that SNPs in a path from the root to a leaf are all associated with genes that form a connected component in a gene-gene or protein-protein network. Models learned from such a constrained tree-growing algorithm can be interpreted as a linear combination of binary-valued functions of subcomponents of some protein-interaction network, assigning functional importance to such functions.

In the final chapter, we demonstrated a Bayesian approach to learning structured clas-

sification models that involves inferring latent group structure among the elements of the feature space whilst classifying binary-labeled data. Using synthetic data, we compared approximate Bayesian inference of this model with contemporary approaches to regularized learning using structured-sparsity inducing norms and illustrated the ability of the model to both predict accurately and encode group structure among the model variables. Motivated by the stochastic block model, we described an extension of the GSCM to model latent structure in features that capture correlations while simultaneously learning a classifier. Approximate Bayesian inference of these models applied to data from genome-wide association studies is yet another extremely promising avenue of research that we hope to pursue.

Bibliography

[Airoldi *et al.*, 2008] E.M. Airoldi, D.M. Blei, S.E. Fienberg, and E.P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[Altschul *et al.*, 1990] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[Baranzini *et al.*, 2009] Sergio E Baranzini, Nicholas W Galwey, Joanne Wang, Pouya Khankhanian, Raija Lindberg, Daniel Pelletier, Wen Wu, Bernard M J Uitdehaag, Ludwig Kappos, Chris H Polman, Paul M Matthews, Stephen L Hauser, Rachel a Gibson, Jorge R Oksenberg, and Michael R Barnes. Pathway and network-based analysis of genome-wide association studies in multiple sclerosis. *Human molecular genetics*, 18(11):2078–90, June 2009.

[Barrett *et al.*, 2009] J.C. Barrett, D.G. Clayton, Patrick Concannon, Beena Akolkar, J.D. Cooper, H.A. Erlich, Cécile Julier, Grant Morahan, Jø rn Nerup, C. Nierras, and Others. Genome-wide association study and meta-analysis find that over 40 loci affect risk of type 1 diabetes. *Nature genetics*, 41(6):703–707, 2009.

[Beal, 2003] Matthew J Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University of London, 2003.

[Beck and Teboulle, 2009] Amir Beck and Marc Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183, 2009.

[Benson *et al.*, 2010] Dennis a Benson, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and Eric W Sayers. GenBank. *Nucleic acids research*, 39(November 2010):32–37, November 2010.

[Bishop, 2007] C M Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*, volume 4. Springer, 2007.

[Botstein *et al.*, 1980] D Botstein, R L White, M Skolnick, and R W Davis. Construction of a genetic linkage map in man using restriction fragment length polymorphisms. *The American Journal of Human Genetics*, 32(3):314–331, 1980.

[Briese *et al.*, 2009] Thomas Briese, Janusz T Paweska, Laura K McMullan, Stephen K Hutchison, Craig Street, Gustavo Palacios, Marina L Khristova, Jacqueline Weyer, Robert Swanepoel, Michael Egholm, Stuart T Nichol, and W Ian Lipkin. Genetic detection and characterization of Lujo virus, a new hemorrhagic fever-associated arenavirus from southern Africa. *PLoS pathogens*, 5(5):e1000455, May 2009.

[Burton, 2007] P R Burton. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–78, June 2007.

[Busa-Fekete and Kegl, 2009] R Busa-Fekete and B Kegl. Accelerating AdaBoost using UCB. In *JMLR: Workshop and Conference Proceedings, KDD cup 2009*, volume 7, pages 111–122, 2009.

[Chen *et al.*, 1998] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

[Chin *et al.*, 2011] Chen-Shan Chin, Jon Sorenson, Jason B Harris, William P Robins, Richelle C Charles, Roger R Jean-Charles, James Bullard, Dale R Webster, Andrew Kasarskis, Paul Peluso, Ellen E Paxinos, Yoshiharu Yamaichi, Stephen B Calderwood, John J Mekalanos, Eric E Schadt, and Matthew K Waldor. The origin of the Haitian cholera outbreak strain. *The New England journal of medicine*, 364(1):33–42, January 2011.

[Danon *et al.*, 2005] L Danon, A Diaz-Guilera, J Duch, and A Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, January 2005.

[Duffy *et al.*, 2008] Siobain Duffy, Laura A Shackelton, and Edward C Holmes. Rates of evolutionary change in viruses: patterns and determinants. *Nature Reviews Genetics*, 9(4):267–276, 2008.

[Easton *et al.*, 2007] Douglas F Easton, Karen A Pooley, Alison M Dunning, Paul D P Pharoah, Deborah Thompson, Dennis G Ballinger, Jeffery P Struewing, Jonathan Morrison, Helen Field, Robert Luben, Nicholas Wareham, Shahana Ahmed, Catherine S Healey, Richard Bowman, Kerstin B Meyer, Christopher A Haiman, Laurence K Kolonel, Brian E Henderson, Loic Le Marchand, Paul Brennan, Suleeporn Sangrajrang, Valerie Gaborieau, Fabrice Odefrey, Chen-Yang Shen, Pei-Ei Wu, Hui-Chun Wang, Diana Eccles, D Gareth Evans, Julian Peto, Olivia Fletcher, Nichola Johnson, Sheila Seal, Michael R Stratton, Nazneen Rahman, Georgia Chenevix-Trench, Stig E Bojesen, Børge G Nordestgaard, Christen K Axelsson, Montserrat Garcia-Closas, Louise Brinton, Stephen Chanock, Jolanta Lissowska, Beata Peplonska, Heli Nevanlinna, Rainer Fagerholm, Hannaleena Eerola, Daehee Kang, Keun-Young Yoo, Dong-Young Noh, Sei-Hyun Ahn, David J Hunter, Susan E Hankinson, David G Cox, Per Hall, Sara Wedren, Jianjun Liu, Yen-Ling Low, Natalia Bogdanova, Peter Schürmann, Thilo Dörk, Rob A E M Tollenaar, Catharina E Jacobi, Peter Devilee, Jan G M Klijn, Alice J Sigurdson, Michele M Doody, Bruce H Alexander, Jinghui Zhang, Angela Cox, Ian W Brock, Gordon MacPherson, Malcolm W R Reed, Fergus J Couch, Ellen L Goode, Janet E Olson, Hanne Meijers-Heijboer, Ans Van Den Ouweland, André Uitterlinden, Fernando Rivadeneira, Roger L Milne, Gloria Ribas, Anna Gonzalez-Neira, Javier Benitez, John L Hopper, Margaret McCredie, Melissa Southey, Graham G Giles, Chris Schroen, Christina Justenhoven, Hiltrud Brauch, Ute Hamann, Yon-Dschun Ko, Amanda B Spurdle, Jonathan Beesley, Xiaoqing Chen, Arto Mannermaa, Veli-Matti Kosma, Vesa Kataja, Jaana Hartikainen, Nicholas E Day, David R Cox, and Bruce A J Ponder. Genome-wide association study identifies novel breast cancer susceptibility loci. *Nature*, 447(7148):1087–93, 2007.

[Efron *et al.*, 2004] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least Angle Regression. *Annals of Statistics*, 32(2):407–499, 2004.

[Emily *et al.*, 2009] Mathieu Emily, Thomas Mailund, Jotun Hein, Leif Schauser, and Mikkel Heide Schierup. Using biological networks to search for interacting loci in genome-wide association studies. *European journal of human genetics : EJHG*, 17(10):1231–40, October 2009.

[Fiedler, 1973] M Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 1973.

[Freund and Mason, 1999] Y Freund and L Mason. The Alternating Decision Tree Algorithm. In *Proceedings of the 16th International Conference on Machine Learning*, pages 124–133, 1999.

[Freund and Schapire, 1997] Y Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[Goldstein *et al.*, 2010] Benjamin a Goldstein, Alan E Hubbard, Adele Cutler, and Lisa F Barcellos. An application of Random Forests to a genome-wide association dataset: methodological considerations & new findings. *BMC genetics*, 11:49, January 2010.

[Goldstein, 2009] David B Goldstein. Common genetic variation and human traits. *The New England journal of medicine*, 360(17):1696–8, April 2009.

[Hamming, 1950] R W Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29:147—-160, 1950.

[Hartwell *et al.*, 1999] L H Hartwell, J J Hopfield, S Leibler, and a W Murray. From molecular to modular cell biology. *Nature*, 402(6761 Suppl):C47–52, December 1999.

[Hertz and Stormo, 1999] G Z Hertz and G D Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics (Oxford, England)*, 15(7-8):563–77, 1999.

[Hofman and Wiggins, 2008] Jake Hofman and Chris Wiggins. Bayesian Approach to Network Modularity. *Physical Review Letters*, 100(25):1–4, June 2008.

[Holland and Leinhardt, 1976] P W Holland and S Leinhardt. Local structure in social networks. *Sociological Methodology*, January 1976.

[Holmans *et al.*, 2009] Peter Holmans, Elaine K Green, Jaspreet Singh Pahwa, Manuel a R Ferreira, Shaun M Purcell, Pamela Sklar, Michael J Owen, Michael C O'Donovan, and Nick Craddock. Gene ontology analysis of GWA study data sets provides insights into the biology of bipolar disorder. *American journal of human genetics*, 85(1):13–24, July 2009.

[Hunter and Kraft, 2007] David J Hunter and Peter Kraft. Drinking from the fire hose–statistical issues in genomewide association studies. *The New England Journal of Medicine*, 357(5):436–439, 2007.

[International Hapmap Consortium, 2003] The International Hapmap Consortium. The International HapMap Project. *Nature*, 426(6968):789–796, 2003.

[International Hapmap Consortium, 2005] The International Hapmap Consortium. A haplotype map of the human genome. *Nature*, 437(7063):1299–320, October 2005.

[Jacob *et al.*, 2009] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. *Proceedings of the 26th Annual International Conference on Machine Learning ICML 09*, pages 1–8, 2009.

[Jakobsdottir *et al.*, 2009] Johanna Jakobsdottir, Michael B Gorin, Yvette P Conley, Robert E Ferrell, and Daniel E Weeks. Interpretation of Genetic Association Studies: Markers with Replicated Highly Significant Odds Ratios May Be Poor Classifiers. *PLoS Genetics*, 5(2):8, 2009.

[Jallow *et al.*, 2009] Muminatou Jallow, Yik Ying Teo, Kerrin S Small, Kirk A Rockett, Panos Deloukas, Taane G Clark, Katja Kivinen, Kalifa A Bojang, David J Conway, Margaret Pinder, Giorgio Sirugo, Fatou Sisay-Joof, Stanley Usen, Sarah Auburn, Suzannah J Bumpstead, Susana Campino, Alison Coffey, Andrew Dunham, Andrew E Fry, Angela Green, Rhian Gwilliam, Sarah E Hunt, Michael Inouye, Anna E Jeffreys, Alieu Mendy,

Aarno Palotie, Simon Potter, Jiannis Ragoussis, Jane Rogers, Kate Rowlands, Elilan Somaskantharajah, Pamela Whittaker, Claire Widden, Peter Donnelly, Bryan Howie, Jonathan Marchini, Andrew Morris, Miguel SanJoaquin, Eric Akum Achidi, Tsiri Agbenyega, Angela Allen, Olukemi Amodu, Patrick Corran, Abdoulaye Djimde, Amagana Dolo, Ogobara K Doumbo, Chris Drakeley, Sarah Dunstan, Jennifer Evans, Jeremy Farrar, Deepika Fernando, Tran Tinh Hien, Rolf D Horstmann, Muntaser Ibrahim, Nadira Karunaweera, Gilbert Kokwaro, Kwadwo A Koram, Martha Lemnge, Julie Makani, Kevin Marsh, Pascal Michon, David Modiano, Malcolm E Molyneux, Ivo Mueller, Michael Parker, Norbert Peshu, Christopher V Plowe, Odile Puijalon, John Reeder, Hugh Reyburn, Eleanor M Riley, Anavaj Sakuntabhai, Pratap Singhasivanon, Sodiomon Sirima, Adama Tall, Terrie E Taylor, Mahamadou Thera, Marita Troye-Blomberg, Thomas N Williams, Michael Wilson, and Dominic P Kwiatkowski. Genome-wide and fine-resolution association analysis of malaria in West Africa. *Nature Genetics*, 41(6):657–665, 2009.

[Janssens and van Duijn, 2008] A Cecile J W Janssens and Cornelia M van Duijn. Genome-based prediction of common diseases: advances and prospects. *Human molecular genetics*, 17(R2):R166–73, October 2008.

[Jenatton *et al.*, 2010] Rodolphe Jenatton, J. Mairal, G. Obozinski, and Francis Bach. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th international conference on Machine learning*, 2010.

[Kivinen and Warmuth, 1999] Jyrki Kivinen and Manfred K. Warmuth. Boosting as entropy projection. *Proceedings of the twelfth annual conference on Computational learning theory - COLT '99*, 26(23):134–144, 1999.

[Kooperberg *et al.*, 2010] Charles Kooperberg, Michael Leblanc, and Valerie Obenchain. Risk Prediction using Genome-Wide Association Studies. *Genetic epidemiology*, 34(7):643–652, 2010.

[Kress and Erickson, 2008] W John Kress and David L Erickson. DNA barcodes: genes, genomics, and bioinformatics. *Proceedings of the National Academy of Sciences of the United*

States of America, 105(8):2761–2762, 2008.

[Kundaje *et al.*, 2006] Anshul Kundaje, Manuel Middendorf, Mihir Shah, Chris H Wiggins, Yoav Freund, and Christina Leslie. A classification-based framework for predicting and analyzing gene regulatory response. *BMC Bioinformatics*, 7(Suppl 1):S5, 2006.

[Kyvik *et al.*, 1995] K O Kyvik, A Green, and H Beck-Nielsen. Concordance rates of insulin dependent diabetes mellitus: a population based study of young Danish twins. *BMJ British Medical Journal*, 311(7010):913–917, 1995.

[Lafon and Lee, 2006] Stephane Lafon and Ann B Lee. Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction, Graph Partitioning, and Data Set Parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, 2006.

[Lango *et al.*, 2008] Hana Lango, Colin N A Palmer, Andrew D Morris, Eleftheria Zeggini, Andrew T Hattersley, Mark I McCarthy, Timothy M Frayling, and Michael N Weedon. Assessing the Combined Impact of 18 Common Genetic Variants of Modest Effect Sizes on Type 2 Diabetes Risk. *Diabetes*, 57(11):3129–3135, 2008.

[Leslie *et al.*, 2004] C S Leslie, E Eskin, A Cohen, J Weston, and W S Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004.

[Lin, 1991] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[Lu and Elston, 2008] Qing Lu and R.C. Elston. Using the optimal receiver operating characteristic curve to design a predictive genetic test, exemplified with type 2 diabetes. *The American Journal of Human Genetics*, 82(3):641–651, 2008.

[Maller *et al.*, 2006] Julian Maller, Sarah George, Shaun Purcell, Jes Fagerness, David Altshuler, Mark J Daly, and Johanna M Seddon. Common variation in three genes, including a noncoding variant in CFH, strongly influences risk of age-related macular degeneration. *Nature Genetics*, 38(9):1055–1059, 2006.

[Meier *et al.*, 2008] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 70(1):53–71, 2008.

[Meila and Shi, 2001] M Meila and J Shi. A random walks view of spectral segmentation. In *AI and STATISTICS*, 2001.

[Meng *et al.*, 2009] Yan a Meng, Yi Yu, L Adrienne Cupples, Lindsay a Farrer, and Kathryn L Lunetta. Performance of random forest when SNPs are in linkage disequilibrium. *BMC bioinformatics*, 10:78, January 2009.

[Middendorf *et al.*, 2005] Manuel Middendorf, Anshul Kundaje, Mihir Shah, and Yoav Freund. Motif Discovery Through Predictive Modeling of Gene Regulation. In *Proceedings of the Ninth Annual International Conference on Research in Computational Molecular Biology*, pages 538–552, 2005.

[Nadler *et al.*, 2005] Boaz Nadler, Stéphane Lafon, Ronald R Coifman, and Ioannis G Kevrekidis. Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck Operators. In *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, 2005.

[Nejentsev *et al.*, 2009] Sergey Nejentsev, Neil Walker, David Riches, Michael Egholm, and John A Todd. Rare variants of IFIH1, a gene implicated in antiviral responses, protect against type 1 diabetes. *Science*, 324(5925):387–389, 2009.

[Nelson and Holmes, 2007] Martha I Nelson and Edward C Holmes. The evolution of epidemic influenza. *Nature reviews. Genetics*, 8(3):196–205, March 2007.

[Newman and Girvan, 2004] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):1–15, February 2004.

[Newman, 2006] M E J Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–82, June 2006.

[Pritchard and Przeworski, 2001] J K Pritchard and M Przeworski. Linkage disequilibrium in humans: models and data. *American journal of human genetics*, 69(1):1–14, July 2001.

[Purcell *et al.*, 2009] Shaun M Purcell, Naomi R Wray, Jennifer L Stone, Peter M Visscher, Michael C O'Donovan, Patrick F Sullivan, and Pamela Sklar. Common polygenic variation contributes to risk of schizophrenia and bipolar disorder. *Nature*, 460(7256):748–752, 2009.

[Pybus and Rambaut, 2009] Oliver G Pybus and Andrew Rambaut. Evolutionary analysis of the dynamics of viral infectious disease. *Nature reviews. Genetics*, 10(8):540–50, August 2009.

[Quattoni *et al.*, 2009] Ariadna Quattoni, Xavier Carreras, Michael Collins, and Trevor Darrell. An efficient projection for l1, regularization. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, 2009.

[Rambaut *et al.*, 2004] Andrew Rambaut, David Posada, Keith a Crandall, and Edward C Holmes. The causes and consequences of HIV evolution. *Nature reviews. Genetics*, 5(1):52–61, January 2004.

[Reich and Lander, 2001] D E Reich and E S Lander. On the allelic spectrum of human disease. *Trends in genetics : TIG*, 17(9):502–10, September 2001.

[Rosset *et al.*, 2004] Saharon Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *The Journal of Machine Learning Research*, 5:941–973, 2004.

[Rosvall and Bergstrom, 2007] Martin Rosvall and C.T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327, 2007.

[Schapire and Singer, 1999] R E Schapire and Y Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[Seberg and Petersen, 2009] Ole Seberg and Gitte Petersen. How many loci does it take to DNA barcode a crocus? *PloS one*, 4(2):e4598, January 2009.

[Shalev-Shwartz *et al.*, 2007] S. Shalev-Shwartz, Y. Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, pages 807–814, 2007.

[Shannon, 2001] C Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1), January 2001.

[Shi and Malik, 2000] J Shi and J Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, January 2000.

[Slonim, 2002] N Slonim. *The Information Bottleneck: Theory and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2002.

[Smith *et al.*, 2009] Gavin J D Smith, Dhanasekaran Vijaykrishna, Justin Bahl, Samantha J Lycett, Michael Worobey, Oliver G Pybus, Siu Kit Ma, Chung Lam Cheung, Jayna Raghwani, Samir Bhatt, J S Malik Peiris, Yi Guan, and Andrew Rambaut. Origins and evolutionary genomics of the 2009 swine-origin H1N1 influenza A epidemic. *Nature*, 459(7250):1122–5, June 2009.

[Speliotes *et al.*, 2010] Elizabeth K Speliotes, Cristen J Willer, Sonja I Berndt, Keri L Monda, Gudmar Thorleifsson, Anne U Jackson, Hana Lango Allen, Cecilia M Lindgren, Jian'an Luan, Reedik Mägi, Joshua C Randall, Sailaja Vedantam, Thomas W Winkler, Lu Qi, Tsegaselassie Workalemahu, Iris M Heid, Valgerdur Steinthorsdottir, Heather M Stringham, Michael N Weedon, Eleanor Wheeler, Andrew R Wood, Teresa Ferreira, Robert J Weyant, Ayellet V Segrè, Karol Estrada, Liming Liang, James Nemesh, Ju-Hyun Park, Stefan Gustafsson, Tuomas O Kilpeläinen, Jian Yang, Nabila Bouatia-Naji, Tõnu Esko, Mary F Feitosa, Zoltán Kutalik, Massimo Mangino, Soumya Raychaudhuri, Andre Scherag, Albert Vernon Smith, Ryan Welch, Jing Hua Zhao, Katja K Aben, Devin M Absher, Najaf Amin, Anna L Dixon, Eva Fisher, Nicole L Glazer, Michael E Goddard, Nancy L Heard-Costa, Volker Hoesel, Jouke-Jan Hottenga, Å sa Johansson, Toby Johnson, Shamika Ketkar, Claudia Lamina, Shengxu Li, Miriam F Moffatt, Richard H Myers, Narisu Narisu, John R B Perry, Marjolein J Peters, Michael Preuss, Samuli Ripatti, Fernando Rivadeneira, Camilla Sandholt, Laura J Scott, Nicholas J Timpson, Jonathan P

Tyrer, Sophie van Wingerden, Richard M Watanabe, Charles C White, Fredrik Wiklund, Christina Barlassina, Daniel I Chasman, Matthew N Cooper, John-Olov Jansson, Robert W Lawrence, Niina Pellikka, Inga Prokopenko, Jianxin Shi, Elisabeth Thiering, Helene Alavere, Maria T S Alibrandi, Peter Almgren, Alice M Arnold, Thor Aspelund, Larry D Atwood, Beverley Balkau, Anthony J Balmforth, Amanda J Bennett, Yoav Ben-Shlomo, Richard N Bergman, Sven Bergmann, Heike Biebermann, Alexandra I F Blakemore, Tanja Boes, Lori L Bonnycastle, Stefan R Bornstein, Morris J Brown, Thomas a Buchanan, Fabio Busonero, Harry Campbell, Francesco P Cappuccio, Christine Cavalcanti-Proença, Yii-Der Ida Chen, Chih-Mei Chen, Peter S Chines, Robert Clarke, Lachlan Coin, John Connell, Ian N M Day, Martin Den Heijer, Jubao Duan, Shah Ebrahim, Paul Elliott, Roberto Elosua, Gudny Eiriksdottir, Michael R Erdos, Johan G Eriksson, Maurizio F Facheris, Stephan B Felix, Pamela Fischer-Posovszky, Aaron R Folsom, Nele Friedrich, Nelson B Freimer, Mao Fu, Stefan Gaget, Pablo V Gejman, Eco J C Geus, Christian Gieger, Anette P Gjesing, Anuj Goel, Philippe Goyette, Harald Grallert, Jürgen Gräß ler, Danielle M Greenawalt, Christopher J Groves, Vilmundur Gudnason, Candace Guiducci, Anna-Liisa Hartikainen, Neelam Hassanali, Alistair S Hall, Aki S Havulinna, Caroline Hayward, Andrew C Heath, Christian Hengstenberg, Andrew a Hicks, Anke Hinney, Albert Hofman, Georg Homuth, Jennie Hui, Wilmar Igl, Carlos Iribarren, Bo Isomaa, Kevin B Jacobs, Ivonne Jarick, Elizabeth Jewell, Ulrich John, Torben Jø rgensen, Pekka Jousilahti, Antti Jula, Marika Kaakinen, Eero Kajantie, Lee M Kaplan, Sekar Kathiresan, Johannes Kettunen, Leena Kinnunen, Joshua W Knowles, Ivana Kolcic, Inke R König, Seppo Koskinen, Peter Kovacs, Johanna Kuusisto, Peter Kraft, Kirsti Kvalø y, Jaana Laitinen, Olivier Lantieri, Chiara Lanzani, Lenore J Launer, Cecile Lecoeur, Terho Lehtimäki, Guillaume Lettre, Jianjun Liu, Marja-Liisa Lokki, Mattias Lorentzon, Robert N Luben, Barbara Ludwig, Paolo Manunta, Diana Marek, Michel Marre, Nicholas G Martin, Wendy L McArdle, Anne McCarthy, Barbara McKnight, Thomas Meitinger, Olle Melander, David Meyre, Kristian Midthjell, Grant W Montgomery, Mario a Morken, Andrew P Morris, Rosanda Mulic, Julius S Ngwa, Mari Nelis, Matt J Neville, Dale R Nyholt, Christopher J O'Donnell, Stephen O'Rahilly, Ken K Ong, Ben Oostra, Guillaume Paré, Alex N Parker, Markus Perola, Irene Pichler,

Kirsi H Pietiläinen, Carl G P Platou, Ozren Polasek, Anneli Pouta, Suzanne Rafelt, Olli Raitakari, Nigel W Rayner, Martin Ridderstrå le, Winfried Rief, Aimo Ruokonen, Neil R Robertson, Peter Rzehak, Veikko Salomaa, Alan R Sanders, Manjinder S Sandhu, Serena Sanna, Jouko Saramies, Markku J Savolainen, Susann Scherag, Sabine Schipf, Stefan Schreiber, Heribert Schunkert, Kaisa Silander, Juha Sinisalo, David S Siscovick, Jan H Smit, Nicole Soranzo, Ulla Sovio, Jonathan Stephens, Ida Surakka, Amy J Swift, Mari-Liis Tammesoo, Jean-Claude Tardif, Maris Teder-Laving, Tanya M Teslovich, John R Thompson, Brian Thomson, Anke Tönjes, Tiinamaija Tuomi, Joyce B J van Meurs, Gert-Jan van Ommen, Vincent Vatin, Jorma Viikari, Sophie Visvikis-Siest, Veronique Vitart, Carla I G Vogel, Benjamin F Voight, Lindsay L Waite, Henri Wallaschofski, G Bragi Walters, Elisabeth Widen, Susanna Wiegand, Sarah H Wild, Gonneke Willemsen, Daniel R Witte, Jacqueline C Witteman, Jianfeng Xu, Qunyuan Zhang, Lina Zgaga, Andreas Ziegler, Paavo Zitting, John P Beilby, I Sadaf Farooqi, Johannes Hebebrand, Heikki V Huikuri, Alan L James, Mika Kähönen, Douglas F Levinson, Fabio Macciardi, Markku S Nieminen, Claes Ohlsson, Lyle J Palmer, Paul M Ridker, Michael Stumvoll, Jacques S Beckmann, Heiner Boeing, Eric Boerwinkle, Dorret I Boomsma, Mark J Caulfield, Ste. Association analyses of 249,796 individuals reveal 18 new loci associated with body mass index. *Nature Genetics*, October 2010.

[Stumvoll *et al.*, 2005] Michael Stumvoll, Barry J Goldstein, and Timon W van Haeften. Type 2 diabetes: principles of pathogenesis and therapy. *Lancet*, 365(9467):1333–46, 2005.

[Tishby and Slonim, 2000] N Tishby and N Slonim. Data clustering by Markovian relaxation and the information bottleneck method. *Advances in Neural Information Processing Systems*, January 2000.

[Tishby *et al.*, 2000] N Tishby, F C Pereira, and W Bialek. The information bottleneck method. *arXiv preprint physics*, January 2000.

[Touchon and Rocha, 2008] Marie Touchon and Eduardo P C Rocha. From GC skews to wavelets: a gentle guide to the analysis of compositional asymmetries in genomic data. *Biochimie*, 90(4):648–59, 2008.

[Van Hoek *et al.*, 2008] Mandy Van Hoek, Abbas Dehghan, Jacqueline C M Witteman, Cornelia M Van Duijn, André G Uitterlinden, Ben A Oostra, Albert Hofman, Eric J G Sijbrands, and A Cecile J W Janssens. Predicting Type 2 Diabetes Based on Polymorphisms From Genome-Wide Association Studies. *Diabetes*, 57(11):3122–3128, 2008.

[Visscher *et al.*, 2008] Peter M Visscher, William G Hill, and Naomi R Wray. Heritability in the genomics era–concepts and misconceptions. *Nature Reviews Genetics*, 9(4):255–266, 2008.

[von Luxburg, 2007] U von Luxburg. A Tutorial on Spectral Clustering. *arXiv*, cs.DS, November 2007.

[Wagner and Wagner, 1993] Dorothea Wagner and Frank Wagner. Between Min Cut and Graph Bisection. In *MFCS '93: Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science*, pages 744–750, London, UK, 1993. Springer-Verlag.

[Wang *et al.*, 2006] Li Wang, Ji Zhu, and Hui Zou. The doubly regularized support vector machine. *Statistica Sinica*, 16(2):589, 2006.

[Warmuth *et al.*, 2008] M. Warmuth, K. Glocer, and S. Vishwanathan. Entropy regularized lpboost. In *Algorithmic Learning Theory*, pages 256–271. Springer, 2008.

[Wei *et al.*, 2009] Zhi Wei, Kai Wang, Hui-Qi Qu, Haitao Zhang, Jonathan Bradfield, Cecilia Kim, Edward Frackleton, Cuiping Hou, Joseph T Glessner, Rosetta Chiavacci, Charles Stanley, Dimitri Monos, Struan F a Grant, Constantin Polychronakos, and Hakon Hakonarson. From disease association to risk assessment: an optimistic view from genome-wide association studies on type 1 diabetes. *PLoS genetics*, 5(10):e1000678, October 2009.

[Weiss *et al.*, 2009] Lauren A Weiss, Dan E Arking, Mark J Daly, and Aravinda Chakravarti. A genome-wide linkage and association scan reveals novel loci for autism. *Nature*, 461(7265):802–8, 2009.

[Williamson *et al.*, 2008] Shannon J Williamson, Douglas B Rusch, Shibu Yooseph, Aaron L Halpern, Karla B Heidelberg, John I Glass, Cynthia Andrews-Pfannkoch, Douglas Fadrosh, Christopher S Miller, Granger Sutton, Marvin Frazier, and J Craig Venter. The Sorcerer II Global Ocean Sampling Expedition: metagenomic characterization of viruses within aquatic microbial samples. *PloS one*, 3(1):e1456, January 2008.

[Yang *et al.*, 2010] Jian Yang, Beben Benyamin, Brian P McEvoy, Scott Gordon, Anjali K Henders, Dale R Nyholt, Pamela a Madden, Andrew C Heath, Nicholas G Martin, Grant W Montgomery, Michael E Goddard, and Peter M Visscher. Common SNPs explain a large proportion of the heritability for human height. *Nature genetics*, 42(7):565–9, July 2010.

[Yuan and Lin, 2006] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, February 2006.

[Ziv *et al.*, 2005] Etay Ziv, Manuel Middendorf, and Chris Wiggins. Information-theoretic approach to network modularity. *Physical Review E*, 71(4Pt2):046117, April 2005.

[Zou and Hastie, 2005] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, April 2005.

Appendix A

List of viruses

Table A.1: List of viruses in *Rhabdoviridae* family used in learning.

Identifier	Name	Host	Subfamily
NC_009609	Orchid fleck virus RNA 2	Plant	unassigned
NC_006942	Taro vein chlorosis virus	Plant	Nucleorhabdovirus
NC_005975	Maize mosaic virus	Plant	Nucleorhabdovirus
EF614258	West Caucasian bat virus	Animal	Lyssavirus
DQ186554	Iranian maize mosaic nucleorhabdovirus	Plant	Nucleorhabdovirus
NC_007642	Lettuce necrotic yellows virus	Plant	Cytorhabdovirus
EF687738	Lettuce yellow mottle virus	Plant	Cytorhabdovirus
NC_002251	Northern cereal mosaic virus	Plant	Cytorhabdovirus
NC_001615	Sonchus yellow net virus	Plant	Cytorhabdovirus
NC_005974	Maize fine streak virus	Plant	Nucleorhabdovirus
NC_003746	Rice yellow stunt virus	Plant	Nucleorhabdovirus
DQ491000	Spring viremia of carp virus isolate A2	Animal	Vesiculovirus
EU373657	Cocal virus Indiana 2	Animal	Vesiculovirus
AJ318079	Spring Viremia of Carp	Animal	Vesiculovirus
NC_002803	Spring viremia of carp virus	Animal	Vesiculovirus
AF104985	Hirame rhabdovirus strain CA 9703	Animal	Novirhabdovirus

NC_005093	Hirame rhabdovirus	Animal	Novirhabdovirus
EU177782	Spring viremia of carp virus isolate BJ0505-2	Animal	Vesiculovirus
EU373658	Vesicular stomatitis Alagoas virus Indiana 3	Animal	Vesiculovirus
AJ810084	Isfahan virus N gene	Animal	Vesiculovirus
DQ097384	Spring viremia of carp virus isolate A1	Animal	Vesiculovirus
NC_001652	Infectious hematopoietic necrosis virus	Animal	Novirhabdovirus
X89213	Infectious haematopoietic necrosis virus (IHNV)	Animal	Novirhabdovirus
NC_000855	Viral hemorrhagic septicemia virus	Animal	Novirhabdovirus
Y18263	Viral hemorrhagic septicemia virus strain Fil3 RNA	Animal	Novirhabdovirus
AY840978	Tupaia rhabdovirus	Animal	Dimarhabdovirus
NC_007020	Tupaia rhabdovirus	Animal	Dimarhabdovirus
NC_008514	Siniperca chuatsi rhabdovirus	Animal	Dimarhabdovirus
DQ399789	Siniperca chuatsi rhabdovirus from China	Animal	Dimarhabdovirus
AF147498	Snakehead rhabdovirus	Animal	Novirhabdovirus
NC_000903	Snakehead rhabdovirus	Animal	Novirhabdovirus
AF081020	Australian bat lyssavirus	Animal	Lyssavirus
EF614261	Khujand lyssavirus	Animal	Lyssavirus
EF614259	Aravan virus	Animal	Lyssavirus
AF418014	Australian bat lyssavirus	Animal	Lyssavirus
EU293116	Rabies virus isolate 9704ARG	Animal	Lyssavirus
EU293115	Rabies virus isolate 9147FRA	Animal	Lyssavirus
EU293121	Rabies virus isolate 8743THA	Animal	Lyssavirus
EU293114	European bat lyssavirus 2 isolate 9018HOL	Animal	Lyssavirus
NC_009528	European bat lyssavirus 2	Animal	Lyssavirus
NC_001542	Rabies virus	Animal	Lyssavirus
NC_006429	Mokola virus	Animal	Lyssavirus
EU293117	Mokola virus isolate 86100CAM	Animal	Lyssavirus
EU293118	Mokola virus isolate 86101RCA	Animal	Lyssavirus
NC_009527	European bat lyssavirus 1	Animal	Lyssavirus
EF614260	Irkut virus	Animal	Lyssavirus

EU293110	Lagos bat virus isolate 8619NGA	Animal	Lyssavirus
EU293108	Lagos bat virus isolate 0406SEN	Animal	Lyssavirus
NC_002526	Bovine ephemeral fever virus	Animal	Ephemerovirus

Table A.2: List of viruses in *Picornaviridae* family used in learning.

Identifier	Name	Host	Subfamily
NC_014137	Honey bee slow paralysis virus	Invertebrate	Cripavirus
NC_001366	Theilovirus	Vertebrate	Cardiovirus
NC_003005	Taura syndrome virus	Invertebrate	Cripavirus
NC_003783	Triatoma virus	Invertebrate	Cripavirus
NC_003077	Equine rhinitis B virus 2	Vertebrate	Erbovirus
NC_001834	Drosophila C virus	Invertebrate	Cripavirus
NC_003784	Black queen cell virus	Invertebrate	Cripavirus
NC_002066	Sacbrood virus	Invertebrate	Iflavirus
NC_008182	Black raspberry necrosis virus RNA1	Plant	Cripavirus
NC_013115	Human enterovirus 107	Vertebrate	Enterovirus
NC_013114	Human enterovirus 98	Vertebrate	Enterovirus
NC_005092	Ectropis obliqua picorna-like virus	Invertebrate	unassigned
NC_013695	Simian picornavirus strain N203	Vertebrate	Enterovirus
NC_010384	Simian picornavirus strain N125	Vertebrate	Enterovirus
NC_008183	Black raspberry necrosis virus RNA2	Plant	Cripavirus
NC_011829	Porcine kobuvirus swine/S-1-HUN/2007/Hungary	Vertebrate	Kobuvirus
NC_013755	Kobuvirus pig/JY-2010a/CHN	Vertebrate	Kobuvirus
NC_012986	Human klassevirus 1	Vertebrate	unassigned
NC_012957	Salivirus NG-J1	Vertebrate	unassigned
NC_011190	Mikania micrantha mosaic virus RNA1	Plant	Cripavirus
NC_011189	Mikania micrantha mosaic virus RNA2	Plant	Cripavirus
NC_010354	Bovine rhinitis B virus	Vertebrate	Erbovirus

NC_008250	Duck hepatitis A virus	Vertebrate	Hepatovirus
NC_006553	Avian sapelovirus	Vertebrate	Sapelovirus
NC_013219	Turnip ringspot virus RNA 2	Plant	Cripavirus
NC_013218	Turnip ringspot virus RNA 1	Plant	Cripavirus
NC_011451	Foot-and-mouth disease virus - type SAT 1	Vertebrate	Aphthovirus
NC_011450	Foot-and-mouth disease virus - type A	Vertebrate	Aphthovirus
NC_005266	Raspberry ringspot virus RNA1	Plant	Cripavirus
NC_003992	Foot-and-mouth disease virus - type SAT 2	Vertebrate	Aphthovirus
NC_004915	Foot-and-mouth disease virus - type Asia 1	Vertebrate	Aphthovirus
NC_004807	Kashmir bee virus	Invertebrate	Cripavirus
NC_004451	Simian picornavirus 1	Vertebrate	unassigned
NC_004365	Aphid lethal paralysis virus	Invertebrate	Cripavirus
NC_004004	Foot-and-mouth disease virus - type O	Vertebrate	Aphthovirus
NC_003987	Porcine enterovirus 8	Vertebrate	Enterovirus
NC_003924	Cricket paralysis virus	Invertebrate	Cripavirus
NC_003113	Perina nuda virus	Invertebrate	Iflavirus
NC_002554	Foot-and-mouth disease virus - type C	Vertebrate	Aphthovirus
NC_001874	Rhopalosiphum padi virus	Invertebrate	Cripavirus
NC_001490	Human rhinovirus 14	Vertebrate	Rhinovirus
NC_001479	Encephalomyocarditis virus	Vertebrate	Cardiovirus
NC_012212	Chaetoceros socialis f. radians RNA virus segment 1	Plant	Cripavirus
NC_012802	Human cosavirus D1	Vertebrate	unassigned
NC_012801	Human cosavirus B1	Vertebrate	unassigned
NC_012800	Human cosavirus A1	Vertebrate	unassigned
NC_012798	Human cosavirus E1	Vertebrate	unassigned
NC_010411	Simian picornavirus 17	Vertebrate	unassigned
NC_003446	Strawberry mottle virus RNA 2	Plant	Cripavirus
NC_003445	Strawberry mottle virus RNA 1	Plant	Cripavirus
NC_010415	Simian enterovirus SV6	Vertebrate	Enterovirus
NC_010413	Simian enterovirus SV43	Vertebrate	Enterovirus

NC_010412	Simian enterovirus SV19	Vertebrate	Enterovirus
NC_003792	Cycas necrotic stunt virus RNA 2	Plant	Cripavirus
NC_003791	Cycas necrotic stunt virus RNA 1	Plant	Cripavirus
NC_010988	Tomato marchitez virus RNA 2	Plant	Cripavirus
NC_010987	Tomato marchitez virus RNA 1	Plant	Cripavirus
NC_009891	Seal picornavirus type 1	Vertebrate	unassigned
NC_009758	Marine RNA virus JP-B	Plant	unassigned
NC_009757	Marine RNA virus JP-A	Plant	unassigned
NC_009530	Brevicoryne brassicae picorna-like virus	Invertebrate	unassigned
NC_009448	Saffold virus	Vertebrate	Cardiovirus
NC_009032	Tomato torrado virus RNA2	Plant	Cripavirus
NC_009013	Tomato torrado virus RNA1	Plant	Cripavirus
NC_006964	Strawberry latent ringspot virus RNA1	Plant	Cripavirus
NC_005281	Heterosigma akashiwo RNA virus SOG263	Plant	Marnavirus
NC_005097	Tobacco ringspot virus RNA 1	Plant	Cripavirus
NC_005096	Tobacco ringspot virus RNA 2	Plant	Cripavirus
NC_004439	Tomato black ring virus RNA 1	Plant	Cripavirus
NC_004421	Bovine kobuvirus	Vertebrate	Kobuvirus
NC_003988	Simian enterovirus A	Vertebrate	Enterovirus
NC_003983	Equine rhinitis B virus 1	Vertebrate	Erbovirus
NC_003974	Patchouli mild mosaic virus RNA 2	Plant	Cripavirus
NC_003840	Tomato ringspot virus RNA 1	Plant	Cripavirus
NC_003839	Tomato ringspot virus RNA 2	Plant	Cripavirus
NC_003788	Apple latent spherical virus segment 2	Plant	Cripavirus
NC_003787	Apple latent spherical virus segment 1	Plant	Cripavirus
NC_003741	Red clover mottle virus RNA 1	Plant	Cripavirus
NC_003738	Red clover mottle virus RNA 2	Plant	Cripavirus
NC_003694	Beet ringspot virus RNA 2	Plant	Cripavirus
NC_003693	Beet ringspot virus RNA 1	Plant	Cripavirus
NC_003622	Grapevine chrome mosaic virus RNA 1	Plant	Cripavirus

NC_003621	Grapevine chrome mosaic virus RNA 2	Plant	Cripavirus
NC_003615	Grapevine fanleaf virus RNA 1	Plant	Cripavirus
NC_003550	Cowpea mosaic virus RNA 2	Plant	Cripavirus
NC_003549	Cowpea mosaic virus RNA 1	Plant	Cripavirus
NC_003509	Blackcurrant reversion virus RNA1	Plant	Cripavirus
NC_003502	Blackcurrant reversion virus RNA 2	Plant	Cripavirus
NC_003545	Cowpea severe mosaic virus RNA 1	Plant	Cripavirus
NC_003544	Cowpea severe mosaic virus RNA 2	Plant	Cripavirus
NC_003975	Patchouli mild mosaic virus RNA 1	Plant	Cripavirus
NC_010710	Radish mosaic virus RNA2	Plant	Cripavirus
NC_010709	Radish mosaic virus RNA1	Plant	Cripavirus
NC_009996	Human rhinovirus C	Vertebrate	Rhinovirus
NC_009887	Human enterovirus 100	Vertebrate	Enterovirus
NC_009750	Duck hepatitis virus AP	Vertebrate	Hepatovirus
NC_009025	Israel acute paralysis virus of bees	Invertebrate	Cripavirus
NC_006965	Strawberry latent ringspot virus RNA2	Plant	Cripavirus
NC_006272	Cherry rasp leaf virus RNA2	Plant	Cripavirus
NC_006271	Cherry rasp leaf virus	Plant	Cripavirus
NC_006057	Arabis mosaic virus RNA 1	Plant	Cripavirus
NC_006056	Arabis mosaic virus RNA 2	Plant	Cripavirus
NC_005290	Broad bean wilt virus 1 RNA 2	Plant	Cripavirus
NC_005289	Broad bean wilt virus 1 RNA 1	Plant	Cripavirus
NC_005267	Raspberry ringspot virus RNA 2	Plant	Cripavirus
NC_004830	Deformed wing virus	Invertebrate	Cripavirus
NC_004441	Porcine enterovirus B	Vertebrate	Enterovirus
NC_004440	Tomato black ring virus RNA 2	Plant	Cripavirus
NC_003990	Avian encephalomyelitis virus	Vertebrate	Tremovirus
NC_003985	Porcine teschovirus 1	Vertebrate	Teschovirus
NC_003982	Equine rhinitis A virus	Vertebrate	Aphthovirus
NC_003976	Ljungan virus	Vertebrate	Parechovirus

NC_003800	Squash mosaic virus RNA 2	Plant	Cripavirus
NC_003799	Squash mosaic virus RNA 1	Plant	Cripavirus
NC_003782	Himetobi P virus	Invertebrate	Cripavirus
NC_003781	Infectious flacherie virus	Invertebrate	Iflavirus
NC_003628	Parsnip yellow fleck virus	Plant	Sequivirus
NC_003626	Maize chlorotic dwarf virus	Plant	Waikivirus
NC_003623	Grapevine fanleaf virus RNA 2	Plant	Cripavirus
NC_003496	Bean pod mottle virus RNA 1	Plant	Cripavirus
NC_003495	Bean pod mottle virus RNA 2	Plant	Cripavirus
NC_003004	Broad bean wilt virus 2 RNA2	Plant	Cripavirus
NC_003003	Broad bean wilt virus 2 RNA1	Plant	Cripavirus
NC_002548	Acute bee paralysis virus	Invertebrate	Cripavirus
NC_001918	Aichi virus	Vertebrate	Kobuvirus
NC_001897	Human parechovirus	Vertebrate	Parechovirus
NC_001859	Bovine enterovirus	Vertebrate	Enterovirus
NC_001632	Rice tungro spherical virus	Plant	Waikavirus
NC_001617	Human rhinovirus 89	Vertebrate	Rhinovirus
NC_001612	Human enterovirus A	Vertebrate	Enterovirus
NC_002058	Poliovirus	Vertebrate	Enterovirus
NC_001489	Hepatitis A virus	Vertebrate	Parechovirus
NC_001472	Human enterovirus B	Vertebrate	Enterovirus
NC_001430	Human enterovirus D	Vertebrate	Enterovirus
NC_001428	Human enterovirus C	Vertebrate	Enterovirus
NC_005876	Kakugo virus	Invertebrate	Iflavirus
NC_003779	Plautia stali intestine virus	Invertebrate	Cripavirus
NC_010810	Human TMEV-like cardiovirus	Vertebrate	Cardiovirus
NC_011349	Seneca valley virus	Vertebrate	Senecavirus
NC_007522	Schizochytrium single-stranded RNA virus	Plant	unassigned
NC_006559	Solenopsis invicta virus 1	Invertebrate	Cripavirus
NC_003785	Satsuma dwarf virus RNA 1	Plant	Cripavirus

NC_003786	Satsuma dwarf virus RNA 2	Plant	Cripavirus
NC_008029	Homalodisca coagulata virus-1	Invertebrate	Cripavirus

Appendix B

Variational Bayesian EM algorithm updates

Given observations $\{(\mathbf{y}_1, z_1), \ldots, (\mathbf{y}_N, z_N)\}$, the evidence of the data under the group structured classification model can be given as

$$\begin{aligned}
\mathcal{E} &= \sum_n \Bigg\{ -\lambda(\widetilde{\eta}_n) \left(\sum_{d,k} \widetilde{\pi}_{dk} y_{nd}^2 \left(\widetilde{\mu}_k^{w^2} + \frac{1}{\widetilde{\tau}_k^w} \right) + 2\widetilde{\mu}^b \sum_{d,k} \widetilde{\pi}_{dk} y_{nd} \widetilde{\mu}_k^w + \widetilde{\mu}^{b^2} \right. && \text{(B.1)} \\
&+ \left. \frac{1}{\widetilde{\tau}^b} + \sum_{d \neq d', k, k'} y_{nd} y_{nd'} \widetilde{\pi}_{dk} \widetilde{\pi}_{d'k'} \widetilde{\mu}_k^w \widetilde{\mu}_{k'}^w \right) + (z_n - \frac{1}{2}) \left(\sum_{d,k} \widetilde{\pi}_{dk} \widetilde{\mu}_k^w y_{nd} + \widetilde{\mu}^b \right) && \text{(B.2)} \\
&+ \log(\sigma(\widetilde{\eta}_n)) - \frac{1}{2}\widetilde{\eta}_n + \lambda(\widetilde{\eta}_n)\widetilde{\eta}_n^2 \Bigg\} && \text{(B.3)} \\
&+ \frac{1}{2} \sum_{n,d,m,k} \widetilde{\psi}_{ndm} \widetilde{\pi}_{dk} \left\{ \Psi(\widetilde{\beta}_{mk}) - \log \tilde{\gamma}_{mk} - \frac{1}{\widetilde{\rho}_{mk}} - \frac{\widetilde{\beta}_{mk}}{\tilde{\gamma}_{mk}} (y_{nd} - \widetilde{\nu}_{mk})^2 \right\} && \text{(B.4)} \\
&+ \sum_{n,d,m,k} \widetilde{\psi}_{ndm} \widetilde{\pi}_{dk} \left(\Psi(\widetilde{\alpha}_{mk}) - \Psi(\widetilde{\alpha}_{ok}) \right) - \sum_{n,d,m} \widetilde{\psi}_{ndm} \log \widetilde{\psi}_{ndm} && \text{(B.5)} \\
&+ \frac{1}{2} \sum_{d,k} \widetilde{\pi}_{dk} \left\{ \log \frac{\tau_k^w}{\widetilde{\tau}_k^w} - \frac{\tau_k^w}{\widetilde{\tau}_k^w} - \tau_k^w (\widetilde{\mu}_k^w - \mu_k^w)^2 + 1 \right\} && \text{(B.6)} \\
&+ \frac{1}{2} \left\{ \log \frac{\tau^b}{\widetilde{\tau}^b} - \frac{\tau^b}{\widetilde{\tau}^b} - \tau^b (\widetilde{\mu}^b - \mu^b)^2 + 1 \right\} && \text{(B.7)} \\
&+ \sum_{d,k} \widetilde{\pi}_{dk} \left\{ \sum_m \left(\log \frac{\Gamma(\widetilde{\alpha}_{mk})}{\Gamma(\alpha_{mk})} - (\widetilde{\alpha}_{mk} - \alpha_{mk})(\Psi(\widetilde{\alpha}_{mk}) - \Psi(\widetilde{\alpha}_{ok})) \right) - \log \frac{\Gamma(\widetilde{\alpha}_{ok})}{\Gamma(\alpha_{ok})} \right\} && \text{(B.8)}
\end{aligned}$$

$$+ \quad \sum_{d,m,k} \widetilde{\pi}_{dk} \left\{ \frac{1}{2} \left(\log \frac{\rho_{mk}}{\widetilde{\rho}_{mk}} + 1 - \rho_{mk} \frac{\widetilde{\beta}_{mk}}{\widetilde{\gamma}_{mk}} (\nu_{mk} - \widetilde{\nu}_{mk})^2 - \frac{\rho_{mk}}{\widetilde{\rho}_{mk}} \right) \right. \tag{B.9}$$

$$\left. + \quad \beta_{mk} \log \frac{\gamma_{mk}}{\widetilde{\gamma}_{mk}} + \log \frac{\Gamma(\widetilde{\beta}_{mk})}{\Gamma(\beta_{mk})} + (\beta_{mk} - \widetilde{\beta}_{mk}) \Psi(\widetilde{\beta}_{mk}) + \widetilde{\beta}_{mk} \left(1 - \frac{\gamma_{mk}}{\widetilde{\gamma}_{mk}} \right) \right\} \tag{B.10}$$

$$+ \quad \sum_{d,k} \widetilde{\pi}_{dk} \log \frac{\pi_k}{\widetilde{\pi}_{dk}} \tag{B.11}$$

where $\widetilde{\alpha}_{ok} = \sum_m \widetilde{\alpha}_{mk}$.

The update equations for the E and M steps of the variational Bayesian inference algorithm that optimizes the evidence can be given as follows.

VBE step

$\tilde{\boldsymbol{\psi}}$:

$$\widetilde{\psi}_{ndm} \propto \sum_k \widetilde{\pi}_{dk} \left\{ \Psi(\widetilde{\alpha}_{mk}) - \Psi(\widetilde{\alpha}_{ok}) + \frac{1}{2} \left(\Psi(\widetilde{\beta}_{mk}) - \log \tilde{\gamma}_{mk} - \frac{1}{\widetilde{\rho}_{mk}} - \frac{\widetilde{\beta}_{mk}}{\widetilde{\gamma}_{mk}} (y_{nd} - \widetilde{\nu}_{mk})^2 \right) \right\} \tag{B.12}$$

VBM step

$\tilde{\boldsymbol{\mu}}^{\mathbf{w}}$:

$$\left(\tau_k^w + 2 \frac{\sum_{n,d} \lambda(\widetilde{\eta}_n) \widetilde{\pi}_{dk} y_{nd}^2}{\sum_d \widetilde{\pi}_{dk}} \right) \widetilde{\mu}_k^w \quad + \quad 2 \sum_{k'} \frac{\sum_{d' \neq d} (\sum_n \lambda(\widetilde{\eta}_n) y_{nd} y_{nd'}) \widetilde{\pi}_{dk} \widetilde{\pi}_{d'k'}}{\sum_d \widetilde{\pi}_{dk}} \widetilde{\mu}_{k'}^w$$

$$= \mu_k^w \tau_k^w \quad + \quad \frac{\sum_{n,d} (z_n - \frac{1}{2}) \widetilde{\pi}_{dk} y_{nd} - 2 \widetilde{\mu}^b \sum_{n,d} \lambda(\widetilde{\eta}_n) \widetilde{\pi}_{dk} y_{nd}}{\sum_d \widetilde{\pi}_{dk}} \tag{B.13}$$

$\tilde{\boldsymbol{\tau}}^{\mathbf{w}}$:

$$\widetilde{\tau}_k^w = \tau_k^w + 2 \frac{\sum_{n,d} \lambda(\widetilde{\eta}_n) \widetilde{\pi}_{dk} y_{nd}^2}{\sum_d \widetilde{\pi}_{dk}} \tag{B.14}$$

$\tilde{\boldsymbol{\mu}}^{\mathbf{b}}$:

$$\widetilde{\mu}^b = \frac{\mu^b \tau^b + \sum_n (z_n - \frac{1}{2}) - 2 \sum_{n,d,k} \lambda(\widetilde{\eta}_n) \widetilde{\pi}_{dk} y_{nd} \widetilde{\mu}_k^w}{\tau^b + 2 \sum_n \lambda(\widetilde{\eta}_n)} \tag{B.15}$$

$\tilde{\boldsymbol{\tau}}^{\mathbf{b}}$:

$$\widetilde{\tau}^b = \tau^b + 2 \sum_n \lambda(\widetilde{\eta}_n) \tag{B.16}$$

$\tilde{\boldsymbol{\nu}}$:

$$\widetilde{\nu}_{mk} = \frac{\rho_{mk}\nu_k \sum_d \widetilde{\pi}_{dk} + \sum_{n,d} \widetilde{\psi}_{ndm}\widetilde{\pi}_{dk} y_{nd}}{\rho_{mk} \sum_d \widetilde{\pi}_{dk} + \sum_{n,d} \widetilde{\psi}_{ndm}\widetilde{\pi}_{dk}} \tag{B.17}$$

$\tilde{\boldsymbol{\alpha}}$, $\tilde{\boldsymbol{\beta}}$, $\tilde{\boldsymbol{\rho}}$:

$$\begin{aligned} \widetilde{\alpha}_{mk} &= \alpha_{mk} + A_{mk} \\ \widetilde{\rho}_{mk} &= \rho_{mk} + A_{mk} \\ \widetilde{\beta}_{mk} &= \beta_{mk} + \frac{1}{2} A_{mk} \end{aligned} \tag{B.18}$$

where

$$A_{mk} = \frac{\sum_{n,d} \widetilde{\psi}_{ndm}\widetilde{\pi}_{dk}}{\sum_d \widetilde{\pi}_{dk}} \tag{B.19}$$

$\tilde{\boldsymbol{\gamma}}$:

$$\tilde{\gamma}_{mk} = \gamma_{mk} + \frac{1}{2}\rho_{mk}(\widetilde{\nu}_{mk} - \nu_{mk})^2 + \frac{1}{2}\frac{\sum_{n,d} \widetilde{\psi}_{ndm}\widetilde{\pi}_{dk}(y_{nd} - \widetilde{\nu}_{mk})^2}{\sum_d \widetilde{\pi}_{dk}} \tag{B.20}$$

$\tilde{\boldsymbol{\pi}}$:

$$\begin{aligned} \widetilde{\pi}_{dk} \propto\ & \pi_k \exp\left[\sum_n \left\{ -\lambda(\widetilde{\eta}_n)\left(y_{nd}^2 \widetilde{\mu}_k^{w^2} + \frac{y_{nd}^2}{\widetilde{\tau}_k^w} + 2\widetilde{\mu}^b y_{nd}\widetilde{\mu}_k^w \right.\right.\right. \\ &+ \left.\left.\left. 2y_{nd}\widetilde{\mu}_k^w \left(\sum_{d' \neq d, k'} y_{nd'}\widetilde{\pi}_{d'k'}\widetilde{\mu}_{k'}^w \right) \right) + (z_n - \frac{1}{2})\widetilde{\mu}_k^w y_{nd} \right. \right. \\ &+ \left. \frac{1}{2}\sum_m \widetilde{\psi}_{ndm}\left(\Psi(\widetilde{\beta}_{mk}) - \log\tilde{\gamma}_{mk} - \frac{1}{\widetilde{\rho}_{mk}} - \frac{\widetilde{\beta}_{mk}}{\tilde{\gamma}_{mk}}(y_{nd} - \widetilde{\nu}_{mk})^2 \right) \right\} \\ &+ \sum_{n,m} \widetilde{\psi}_{ndm}(\Psi(\widetilde{\alpha}_{mk}) - \Psi(\widetilde{\alpha}_{ok})) + \frac{1}{2}\left(\log\frac{\tau_k^w}{\widetilde{\tau}_k^w} - \frac{\tau_k^w}{\widetilde{\tau}_k^w} - \tau_k^w(\widetilde{\mu}_k^w - \mu_k^w)^2 + 1 \right) \\ &+ \sum_m \left(\log\frac{\Gamma(\widetilde{\alpha}_{mk})}{\Gamma(\alpha_{mk})} - (\widetilde{\alpha}_{mk} - \alpha_{mk})(\Psi(\widetilde{\alpha}_{mk}) - \Psi(\widetilde{\alpha}_{ok})) \right) - \log\frac{\Gamma(\widetilde{\alpha}_{ok})}{\Gamma(\alpha_{ok})} \\ &+ \left\{ \frac{1}{2}\left(\log\frac{\rho_{mk}}{\widetilde{\rho}_{mk}} - \frac{\rho_{mk}}{\widetilde{\rho}_{mk}} - \rho_{mk}\frac{\widetilde{\beta}_{mk}}{\tilde{\gamma}_{mk}}(\nu_{mk} - \widetilde{\nu}_{mk})^2 \right) + \beta_{mk}\log\frac{\gamma_{mk}}{\tilde{\gamma}_{mk}} \right. \\ &+ \left. \left. \log\frac{\Gamma(\widetilde{\beta}_{mk})}{\Gamma(\beta_{mk})} + (\beta_{mk} - \widetilde{\beta}_{mk})\Psi(\widetilde{\beta}_{mk}) + \widetilde{\beta}_{mk}\left(1 - \frac{\gamma_{mk}}{\tilde{\gamma}_{mk}}\right) \right\} \right] \end{aligned} \tag{B.21}$$