5-2012

Bayesian nonparametric clusterings in relational and high-dimensional settings with applications in bioinformatics.

Dazhuo Li
University of Louisville

Recommended Citation

Li, Dazhuo, "Bayesian nonparametric clusterings in relational and high-dimensional settings with applications in bioinformatics." (2012). *Electronic Theses and Dissertations.* Paper 827.
https://doi.org/10.18297/etd/827

BAYESIAN NONPARAMETRIC CLUSTERINGS IN RELATIONAL AND HIGH-DIMENSIONAL SETTINGS WITH APPLICATIONS IN BIOINFORMATICS

By

Dazhuo Li
B.S., Zhejiang University, China, 2002
M.SE., Zhejiang University, China, 2006

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky, USA

May 2012

BAYESIAN NONPARAMETRIC CLUSTERINGS IN RELATIONAL AND HIGH-DIMENSIONAL SETTINGS WITH APPLICATIONS IN BIOINFORMATICS

By

Dazhuo Li
B.S., Zhejiang University, China, 2002
M.SE., Zhejiang University, China, 2006

A Dissertation Approved on

March 23, 2012

by the following Dissertation Committee:

Eric Rouchka, D.Sc. Advisor

Ahmed Desoky, Ph.D.

Ming Ouyang, Ph.D.

Roman Yampolskiy, Ph.D.

Guy Brock, Ph.D.

Patrick Shafto, Ph.D.

ABSTRACT

BAYESIAN NONPARAMETRIC CLUSTERINGS IN RELATIONAL AND HIGH-DIMENSIONAL SETTINGS WITH APPLICATIONS IN BIOINFORMATICS

By

Dazhuo Li

March 23, 2012

Recent advances in high throughput methodologies offer researchers the ability to understand complex systems via high dimensional and multi-relational data. One example is the realm of molecular biology where disparate data (such as gene sequence, gene expression, and interaction information) are available for various snapshots of biological systems. This type of high dimensional and multi-relational data allows for unprecedented detailed analysis, but also presents challenges in accounting for all the variability. High dimensional data often has a multitude of underlying relationships, each represented by a separate clustering structure, where the number of structures is typically unknown *a priori*.

To address the challenges faced by traditional clustering methods on high dimensional and multi-relational data, we developed three feature selection and cross-clustering methods: 1) infinite relational model with feature selection (FIRM) which incorporates the rich information of multi-relational data; 2) Bayesian Hierarchical Cross-Clustering (BHCC), a deterministic approximation to Cross Dirichlet Process mixture (CDPM) and to cross-clustering; and 3) randomized approximation (RBHCC), based on a truncated hierarchy. An extension of BHCC, Bayesian Congruence Measuring (BCM), is proposed to measure incongruence between genes and to identify sets of congruent loci with identical evolutionary histories. We adapt our BHCC algorithm to the inference of BCM, where the intended structure of each view (congruent loci) represents consistent evolutionary processes.

We consider an application of FIRM on categorizing mRNA and microRNA. The model uses latent structures to encode the expression pattern and the gene ontology annotations. We also apply FIRM to recover the categories of ligands and proteins, and to predict unknown drug-target interactions, where latent categorization structure encodes drug-target interaction, chemical compound similarity, and amino acid sequence similarity. BHCC and RBHCC are shown to have improved predictive performance (both in terms of cluster membership and missing value prediction) compared to traditional clustering methods. Our results suggest that these novel approaches to integrating multi-relational information have a promising future in the biological sciences where incorporating data related to varying features is often regarded as a daunting task.

DEDICATION

To my grandma, my parents, and my wife.

ACKNOWLEDGEMENTS

I would first like to express my deepest gratitude to my Ph.D advisor, Professor Eric Rouchka. I joined his research group while I was looking for doctoral research topics in the field of machine learning. He introduced me to the complex world of life science and the exciting field of bioinformatics. During my nearly three years of study at the bioinformatics lab, he has guided me in so many different aspects, ranging from dissertation research, molecular biology, paper writing and presentation, to job interview and communication skills. He also provided a great atmosphere with invaluable research activities, including lab meeting, journal club, retreat, and summit, such that discussions and collaborations are always fun and insightful. Working with him has always been a tremendously delightful experience because of his insight, humility, patience, and generosity. Without his guidance and support, this dissertation would not have been possible, and I would not have had the wonderful experience which continues to guide me through life.

I would especially like to thank Professor Patrick Shafto. I was very fortunate to take his excellent course on computational cognitive science during my time of searching for a principled and promising research topic. His course triggered my interest in the Bayesian framework, and particularly influenced the research presented in this dissertation. After finishing the course, I was lucky to have the opportunity to continue my research with him on probabilistic modeling (especially the approximation algorithms to the CDPM model, and the work on jointly modeling features and relations, as presented in this dissertation), which has been an extremely delightful and rewarding experience. This dissertation would not have been possible without his insightful ideas and passionate mentoring.

I am grateful to Professor Ming Ouyang. His support and encouragement helped me go through the difficulties and challenges that inevitably occur during graduate school. He also contributed interesting discussions to the work on phylogenomic analysis, as presented in this dissertation. I also greatly appreciate the research guidance from Professor Antonio Badia on database and query language design, which broadened and deepened my sense of research. I would also like to thank the other members of my committee: Professors Ahmed Desoky, Roman Yampolskiy, and Guy Brock.

Their thoughtful comments raised many interesting questions and strengthened my understanding on some technical details. I am also grateful to have studied my Ph.D in a great research atmosphere in the Department of Computer Engineering and Computer Science at the University of Louisville. I would like to thank the department Chair, Professor Adel Elmaghraby, for providing this excellent atmosphere and giving me the opportunity to be a part of it.

I had the benefit of interacting with the exciting bioinformatics research group. I would like to thank Dr. Robert Flight for supplying clever ideas and tools for my research, and for stimulating lively discussions in the lab meetings and the bioinformatics journal club. I am also grateful to Fahim Mohammad, Abdallah Eteleeb, and Ernur Saka for providing supporting tools, and for the innumerable interesting conversations. Russell Warner, Baxter Eaves, and other members of the computational cognitive science lab have always provided interesting discussions on cognition and learning.

This dissertation is also supported in part by the Department of Energy (DE-EM0000197), and the National Aeronautics and Space Administration (NNX10AJ36G, R. Keynton, PI). Its contents are solely the responsibility of the authors and do not represent the official views of DOE or NASA.

I am also grateful to all my friends, especially Mike Hagan, for their friendship, along with the support and encouragement during difficulties.

I owe an enormous amount to my family, who laid the foundation for this work. To my parents, for raising me and my sister with endless love, patience, and encouragement. To my sister, for always being there with mom and dad, talking with them over the phone, sharing their worries and joys, and bearing with my inexcusable absence. To my son Justin, who always amazes me. And to my wife, Yinlu, for her trust, patience, and sacrifice, and for being part of me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The research presented in this dissertation focuses on Bayesian nonparametric techniques to clustering in relational and high-dimensional settings, as well as its application in aiming to solve challenging biological problems. The complexity of nature and the inevitable lack of detailed observations make human understanding of biological systems extremely challenging. Steady progress towards the goal are achieved by successful integration of inferential methods (e.g. statistical modeling, graph theory, Bayesian network, causality models) and biological data (e.g. randomized observations and carefully designed experiments). One of the fundamental building blocks in human understanding of nature is categorization. Biologically sensible categories of genes, proteins and chemical ligands, although abstracted and simplified representation of the dynamic causal machinery underlying complex biological systems, may capture many of the system's essential characteristics. Despite the development of high-throughput experimental methods of molecular biology, which generate detailed high dimensional and relational data, our understanding is still hindered substantially by the lack of appropriate methodologies for extracting complex patterns effectively from many resources available.

Statistical modeling aims to derive from observations a summarized interpretation which can be used to predict future events. Parametric statistical modeling assumes that the observations have been generated from a parametrized probability distribution where the parameters are unknown and estimated from the likelihood function. Bayesian modeling puts parameters on the same conceptual level as observations such that the parameters are modeled through probability distributions as well. Both classical and Bayesian parametric modeling require a fixed and finite number of parameters. To avoid over-fitting or under-fitting caused by the inappropriate complexity assumption of a model, model selection and comparison techniques are applied along with the process of parameter estimation. Bayesian nonparametrics incorporates infinite dimensional parameters into models and automatically adjusts model complexity to the data. There has been a great deal of previous work applying Bayesian and nonparametric Bayesian methods to a broad range of scientific fields, including density estimation and clustering [Escobar and West, 1995, Rasmussen, 2000, Beal et al., 2002, Teh et al., 2003, Heller and Ghahramani, 2005a], document modeling [Blei et al., 2003, Rosen-Zvi et al., 2004, Li and McCallum, 2006, Blei et al., 2004], visual detection and object recognition [Sudderth et al., 2003, Sigal et al., 2004, Fei-Fei and Perona, 2005, Li et al., 2009], and bioinformatics [Medvedovic and Sivaganesan, 2002, Savage et al., 2009].

To address the challenges faced by traditional clustering methods on high dimensional and multi-relational data, we developed three feature selection and cross-clustering methods. We represent the extended clustering problems in these novel settings using a Bayesian nonparametric approach and probabilistic graphical model. Similar to finite mixtures or Dirichlet process mixtures, we assume each data point is generated by a mixture of distributions rather than a standard distribution (e.g. Gaussian distribution). As a result, analytical inference is not possible for the presented models of this dissertation. We provide general procedures for Bayesian inference, learning, and prediction. Alternatively, by integrating greedy and randomized algorithmic techniques, we provide more scalable inference algorithms while still retaining the desirable properties from a Bayesian paradigm. First, we propose joint feature selection and infinite relational model (FIRM) which allows for more robust collective inference and often results in significant performance gains. Traditional IRMs define the same form of mixture density functions over all features, and feature selection, an essential part of clustering, is ignored or must be done prior to the application of the methods. In contrast with these methods, FIRM incorporates latent variables of features to explicitly represent feature saliency in a relational context and results in structures with more intuitive interpretation and better prediction accuracy. By recasting the feature selection problem as parameter estimation, FIRM is able to efficiently avoid the combinatorial search through the space of all feature subsets. Second, we propose Bayesian Hierarchical Cross-Clustering (BHCC), a greedy and deterministic approximation to CDPM which relaxes the single-DPM assumption, allowing the possibility of one data set having multiple different views of clusterings. This provides the capability to separate structured features from noisy features and the ability to identify cases where different dimensions of the data are best described by different DPMs. Third, we provide a more efficient computational algorithm, the randomized BHCC (RBHCC), based on a randomization and truncated hierarchy, that scales linearly in the number of dimensions and data points; last, an extension of BHCC, Bayesian Congruence Measuring (BCM), is proposed to measure incongruence between genes and to identify sets of congruent loci with identical evolutionary histories.

To validate the methods performance, we first perform experiments and compare the methods' prediction accuracy. Our results show that FIRM, by incorporating feature selection into relational learning, are more robust to noise, and adjust better to model complexity. This leads to a more intuitive interpretation and greater predictive accuracy. Meanwhile, results on synthetic and real-world data sets demonstrate that the cross-clustering based algorithms perform as well or better than the clustering based algorithms, our deterministic approaches perform as well as the MCMC-based CDPM, and the randomized approximation provides a remarkable speed-up relative to the full deterministic approximation with minimal cost in predictive error. As particular examples, we consider applications of these methods in bioinformatics. First, we apply FIRM on the discovery of microRNA and mRNA modules. MicroRNA (miRNA) plays an important role in biological processes by translational repression or degradation of mRNAs. For the later case, the expression levels of genes may be substantially affected by miRNAs. It is thus interesting to discover co-expressed mRNAs and miRNAs that are potentially involved in the same regulatory network. FIRM takes an miRNA-mRNA correlation matrix and gene ontology (GO) annotation data as the input, and aims to find mRNA and miRNA clusterings that yield clean blocks representing co-expression and potentially co-regulatory relationships between the genes and miRNAs. The results suggest interesting gene modules and GO terms. Second, we study the identification of interactions between ligands (chemical compounds, drugs) and proteins (receptors, targets), which is an important step of drug discovery. Various data types have been collected for drug-target interaction prediction, including chemical compound descriptors, protein sequences, ligand-target bindings and pharmaceutical effects. We apply a special case of FIRM (where all input data are relational) to jointly detect biologically sensible ligand groups and protein groups, and to predict drug-target interactions. Third, we demonstrate BCM on estimating the phylogeny relationships amongst ray-finned fish (*Actinopterygii*) with 10 alignments of protein-coding genes. BCM accounts for evolutionary heterogeneities and identifies congruent gene subsets using Bayesian hypothesis testing, and approximates the posterior probability of genes being congruent in a fast deterministic manner. The result shows that the model recovers interesting congruence structure among genes. A brief description of our approaches in this dissertation are outlined as follows.

Chapter 2: Bayesian Methods

We begin by reviewing a broad range of topics on both Bayesian methods upon which the models in this dissertation are based. We first describe the fundamental concept of Bayesian learning in the world of uncertainty. We then provide an introduction to the exponential families of probability distributions and conjugate priors which are used extensively in later chapters. Turning to clustering and probability density estimation, we discuss several algorithms based on different formulations or inference techniques, including K-means, maximum likelihood estimation, and maximum *a posterior* estimation. The chapter concludes with an introduction to the Dirichlet process, which is widely used in Bayesian nonparametric statistics. We cover the probabilistic theory underlying these robust methods, before discussing the learning algorithms.

Chapter 3: Molecular Biology

Chapter 3 begins with a brief introduction to molecular biology, followed by a description of microarray experiment technology. Turning to molecular evolution, we describe nucleotide substitution, which is the basic process in the evolution of DNA sequences and is fundamental for estimating the rate of evolution and for reconstructing the evolutionary history of organisms. We then survey several different model-based methods for molecular phylogenetics, including maximum likelihood estimation and MCMC sampling algorithms.

Chapter 4: Infinite Relational Model with Feature Selection

In chapter 4, we develop a novel Bayesian nonparametric model for simultaneous feature selection and clustering. We begin by reviewing unsupervised feature selection, which aims to discover a subset of features representing the structure of greatest interest. We then propose FIRM (a joint Feature selection and Infinite Relational Model) which aims to address the drawbacks of traditional algorithms due to the absence of class labels and prior knowledge of the underlying structure. Via latent graphical model and the Chinese Restaurant Process, FIRM fuses feature data and the rich information contained in different relational data, which are increasingly available for many problem domains. Although an analytic solution is not possible in FIRM for the same reason as in other mixture models (e.g DPM) where data are represented by a mixture of distributions rather than a standard distribution, through MCMC and Gibbs sampling, we provide an efficient inference method for learning the posterior distribution of the latent structures. We conclude by validating FIRM's performance in clustering and predicting hand-written digits.

Chapter 5: Application of FIRM on Biological Problems

The fifth chapter considers applications of FIRM to challenging problems in Bioinformatics. We begin with the problem of discovering biological sensible groups of mRNA and microRNA. The model encodes latent categorization of mRNA and microRNA, and the latent saliency of gene ontology terms. The latent structures further encode the gene expressions of microRNA and mRNA, and the gene ontology annotation mappings. Applying blocked Gibbs sampling, we iteratively and repetitively draw different parts of the latent structures from their posterior distributions. We then apply FIRM to detect biologically sensible ligand (drug) groups and target (protein) groups and to predict drug-target interactions. The model encodes latent categorization structure of drugs and proteins, which in turn encodes drug-target interaction, chemical compound similarity, and amino acid sequence similarity.

Chapter 6: Bayesian Hierarchical Cross-Clustering

In chapter 6, we develop an approximate inference algorithm for the Cross Dirichlet Process Mixture (CDPM) model which accounts for a more complex structure than the Dirichlet Process Mixture (DPM) model. Standard clustering models, for example the DPM, assume a single clustering structure to account for all the variability in the data. However, as the number of dimensions increases, the assumption becomes less realistic and effective in explaining the heterogeneity in the data. We begin by reviewing an alternative set of clustering algorithms, which relax the traditional assumption and allow for multiple views, each describing the data using a subset of the dimensions. Standard joint feature selection and clustering also boils down to the case where the number of views is fixed and known, and is two. Motivated further by the fact that typically the number of views is not known *a priori*, we describe the CDPM model that allows potentially many views, and infers the correct number for a given data set. We then propose Bayesian Hierarchical Cross-Clustering (BHCC), a greedy and deterministic approximation to CDPM. Our bottom-up, deterministic approach results in a hierarchical clustering of dimensions, and at each node, a hierarchical clustering of data points. We also provide a more efficient computational algorithm, the randomized BHCC (RBHCC), based on a randomization and truncated hierarchy, that scales linearly in the number of dimensions and data points. We conclude by validating BHCC and RBHCC's predictive performance on synthetic and real-world data sets.

Chapter 7: Bayesian Congruence Measuring in Phylogenomics

In this chapter, we generalize our BHCC algorithm to the more complex field of molecular evolutionary biology. We begin by describing the gene incongruence problem whereby separate molecular phylogenies inferred from individual loci disagree with each other. This incongruence among phylogenies can be the result of systematic error, but can also be the result of different evolutionary histories. We review different methodologies for measuring gene incongruence . We then present Bayesian Congruence Measuring (BCM) to estimate the degree of incongruence and to identify sets of congruent loci within which the evolutionary histories are identical. The inference for BCM is adapted from our BHCC algorithm. Instead of clustering structures, The intended structure of each view (congruent loci) in BCM represents evolutionary processes rather than clustering structures as originally developed for CDPM. We demonstrate the method on a gene sequence data of 10 nuclear genes from 20 ray-finned fish (*Actinopterygii*) species.

Chapter 8: Summary and Future Work

We conclude by surveying the contributions of this dissertation, and outline directions for future research.

CHAPTER 2

BAYESIAN METHODS

Probabilistic modeling methods play an essential role in the design and analysis of complex systems. We review several probabilistic learning techniques upon which our contributions are based. The fundamental concept of Bayesian learning is described in Sec. 2.1. Sec. 2.2 describes exponential families of probability distributions, highlighting sufficiency and conjugacy, two properties essential to Bayesian learning. Mixture based density estimation and clustering are discussed in Sec. 2.3, followed by Sec. 2.4, an introduction to Dirichlet process and its mixtures, a Bayesian nonparametric method more robust and flexible than finite mixtures.

2.1 Bayesian Inference

From a Bayesian perspective, **probabilities** represent **degrees of belief** about events in the world, and data are used to update those degrees of belief [Pearl, 2009, Jaynes, 2003, MacKay, 2003]. The events, or **generative models** which give rise to the data, can be deterministic and expressed in the form of deterministic functions, or stochastic and expressed in the form of probabilistic functions. Intuitively, the updated degree of belief about an event should reflect both our prior belief about the event and the plausibility of data being generated by the event. Formally, according to Bayes' rule, the beliefs over models given the data is expressed by the **posterior distribution** (or posterior belief) of models:

$$p(m|D) = \frac{p(m)p(D|m)}{p(D)} \tag{2.1}$$

$$\propto p(m) \prod_{n=1}^{N} p(x_n|m) \tag{2.2}$$

where m denotes the model, D denotes the data, $p(m)$ is the **prior distribution** (or prior belief) over the model, and $p(D|m)$ is the **likelihood** of the model given the data.

In many situations, statistical models are used to predict future observations given current observations. Since it is uncertain which events have generated the observed data, it is important to account for all the possibilities. The **predictive distribution** of a new observation x^* is:

$$p(x^*|D) = \sum_{m=1}^{M} p(x^*|m)p(m|D) \tag{2.3}$$

Note that the posterior distribution $p(m|D)$ represents the updated degree of beliefs over the models.

In statistical modeling, standard models are usually defined as a certain type of parametric probability distributions. The likelihoods are usually written as $p(D|\theta)$, with θ as the parameters of the model m. Similarly, the prior distributions are usually written as $p(\theta|\lambda)$, which is typically itself a member of a family of densities with **hyperparameters** λ. Recursively, the hyperparameters may also be placed in a prior distribution $p(\lambda)$. For the moment, we assume these hyperparameters are set to some fixed value. Then the posterior distribution of the parameters can be written

$$\begin{aligned} p(\theta|D,\lambda) &= \frac{p(\theta|\lambda)p(D|\theta,\lambda)}{p(D)} && (2.4) \\ &\propto p(\theta|\lambda)\prod_{n=1}^{N} p(x_n|\theta) && (2.5) \end{aligned}$$

The predictive distribution then takes the form:

$$p(x^*|D,\lambda) = \int p(x^*|\theta)p(\theta|D,\lambda)d\theta \tag{2.6}$$

Contrary to the Bayesian paradigm, one may choose to ignore the uncertainty over the events and search for a **point-estimate** of the parameters. One such choice is to approximate the parameters' posterior distribution (Eq. 2.4) by a single **maximum** *a posteriori* (MAP) estimate:

$$\begin{aligned} \hat{\theta}_{\text{MAP}} &= \arg\max_{\theta} p(\theta|D,\lambda) && (2.7) \\ &= \arg\max_{\theta} p(\theta|\lambda)\prod_{n=1}^{N} p(x_n|\theta) && (2.8) \\ &= \arg\max_{\theta}(\log p(\theta|\lambda) + \sum_{n=1}^{N} \log p(x_n|\theta)) && (2.9) \end{aligned}$$

Another popular choice is the **maximum likelihood** (ML) parameter estimate:

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} p(D|\theta, \lambda) \tag{2.10}$$

$$= \arg\max_{\theta} \sum_{n=1}^{N} \log p(x_n|\theta)) \tag{2.11}$$

2.2 Exponential Family

Exponential family represents a broad class of probability distributions having many important properties (e.g. sufficient statistics, conjugate priors) in common. This section studies several probability distributions in the exponential family.

2.2.1 Distribution on Binary Variables

Bernoulli Distribution

Consider a binary random variable $x \in \{0, 1\}$. The probability of $x = 1$ is parameterized by μ: $p(x = 1|\mu) = \mu$ where $0 \leq \mu \leq 1$). The probability distribution over x can also be written in the form

$$\mathrm{Bernoulli}(x|\mu) = \mu^{x}(1-\mu)^{1-x} \tag{2.12}$$

which is know as the **Bernoulli** distribution. The mean and variance of the distribution has the following simple form

$$\mathbb{E}[x] = \mu \tag{2.13}$$

$$\mathrm{var}[x] = \mu(1-\mu) \tag{2.14}$$

Given N i.i.d. observations as in Eq. 2.12, the likelihood function on μ is

$$p(D|\mu) = \prod_{n=1}^{N} p(x_n|\mu) = \prod_{n=1}^{N} \mu^{x_n}(1-\mu)^{1-x_n} \tag{2.15}$$

$$= \mu^{\sum_{n=1}^{N} x_n}(1-\mu)^{N-\sum_{n=1}^{N} x_n} \tag{2.16}$$

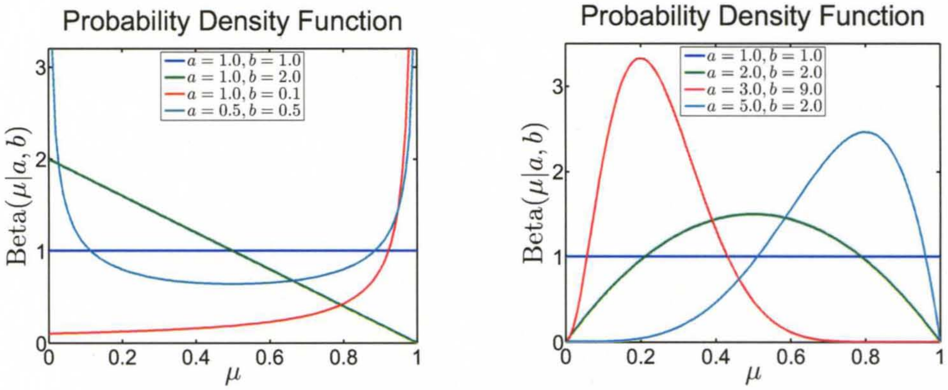


Figure 2.1: Examples of beta distributions. (Left) Beta densities with small hyperparameters. (Right) Beta densities with large hyperparameters.

The maximum likelihood estimate of the parameter μ given D is obtained by setting the derivative of $\log p(D|\mu)$ with respect to μ equal to zero:

$$\hat{\mu}_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{2.17}$$

Here $\sum_{n=1}^{N} x_n$ is a sufficient statistic for the observations under Bernoulli distribution.

Beta Priors

The **beta** distribution is the conjugate prior for the Bernoulli distribution and the binomial distribution. The beta distribution with hyperparameters a, b can be written as follows:

$$\mathrm{Beta}(\mu|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \tag{2.18}$$

The normalization constant of beta distribution involves a ratio of gamma functions ($\Gamma(x)$). The mean and variance of the beta distribution has the following simple form

$$\mathbb{E}[\mu] = \frac{a}{a+b} \tag{2.19}$$

$$\mathrm{var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)} \tag{2.20}$$

Fig. 2.1 illustrates several beta distributions. When $a = b = 1$, it assigns equal probability to all possible value of μ. Small hyperparameters indicates biased priors. Large hyperparameters lead to unimodal priors concertrated on the chosen mean.

Conjugate Posteriors and Predictions

The posterior distribution of μ given observations $D = \{x_1, \ldots, x_N\}$ and hyperparameters a, b is derived as follows

$$p(\mu|D, a, b) = \mu^C(1-\mu)^{N-C}\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \tag{2.21}$$

$$\propto \mu^{a+C-1}(1-\mu)^{b+N-C-1} \tag{2.22}$$

$$\propto \text{Beta}(a+C-1, b+N-C-1) \tag{2.23}$$

where $C = \sum_{n=1}^{N}$. We see that through conjugacy, the posterior (Eq. 2.21) has the same functional dependence on μ as the prior distribution.

The predictive function of a future observation x^* (as in Eq. 2.6) is often of interest. For binomial-beta, this takes the form

$$p(x=1|D, a, b) = \int p(x=1|\mu)p(\mu|D, a, b)d\mu = \mathbb{E}[\mu|D, a, b] \tag{2.24}$$

$$= \frac{a+C}{a+b+N} \tag{2.25}$$

which can be interpreted as the fraction of the total observations corresponding to $x = 1$ (including both real observations and prior believed observations). Comparing this posterior mean (Eq. 2.24) to the ML estimate as in Eq. 2.17, we see that in the limit of an infinitely large data set, the result reduces to the MLE. the raw frequencies underlying the ML estimate. The raw frequencies underlying ML estimate have been smoothed by the fictitious counts contributed by the Beta prior.

2.2.2 Distribution on Multinomial Variables

Multinomial Distribution

Consider a random variable x taking one of K mutually exclusive discrete values in $\{1, \ldots, K\}$. If we denote the probability of $x_k = 1$ by the parameter μ_k, then the probability distribution of $p(x)$ is

$$p(x|\mu_1, \cdots, \mu_K) = \prod_{k=1}^{K} \mu_k^{\delta(x,k)} \qquad \delta(x,k) = \begin{cases} 1 & \text{if } x = k \\ 0 & \text{if } x \neq k \end{cases} \tag{2.26}$$

It is convenient to adopt the 1-of-K scheme in which the variable is represented by a K-dimensional vector $\boldsymbol{x}$ such that if the variable is in state k, then $x_k = 1$ and $x_i = 0$ for $i \neq k$. As a result,the distribution $p(x)$ in Eq. 2.26 is also represented as

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k} \tag{2.27}$$

where $\mu = (\mu_1, \ldots, \mu_K)^{\mathrm{T}}$

Given N independent observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the corresponding likelihood function is

$$p(D|\boldsymbol{\mu}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\mu}) = \prod_{n=1}^{N}\prod_{k=1}^{K} \mu_k^{x_{nk}} = \prod_{k=1}^{K} \mu_k^{C_k} \qquad C_k = \sum_{n=1}^{N} x_{nk} \tag{2.28}$$

We see C_k is the sufficient statistics for this distribution. From this likelihood function we can derive that the maximum likelihood estimates of the parameters equal the empirical frequencies of the states over the N observations:

$$\hat{\mu}_k^{ML} = \frac{C_k}{N} \tag{2.29}$$

The **multinomial** distribution considers the joint distribution of the quantities $C_1, \ldots, C_K$, parameterized by μ and the total number N of observations. Derived from the likelihood function (Eq. 2.28) we have

$$\text{Multinomial}(C_1, \ldots, C_K)|\boldsymbol{\mu}, N) = \frac{N!}{C_1! \ldots C_K!} \prod_{k=1}^{K} \mu_k^{C_k} \tag{2.30}$$

Dirichlet Prior

The **Dirichlet** distribution (Fig. 2.2) is the conjugate prior for the multinomial distribution. The Dirichlet distribution with hyperparameters $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$ has the form:

$$\text{Dirichlet}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\ldots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1} \qquad \alpha_0 = \sum_{k=1}^{K} \alpha_k \tag{2.31}$$

Note that the Dirichlet distribution's normalization constant involves a ratio of gamma functions. When $K = 2$, the Dirichlet distribution is equivalent to the beta distribution. Denoting the beta distribution's two hyperparameters by a and b, then the beta density $\mu \sim \text{Beta}(a, b)$ as in Eq. 2.18 is equivalent to the Dirichlet density $(\mu, 1-\mu) \sim \text{Dirichlet}((\mu, 1-\mu)|(a, b))$. As in beta distribution,

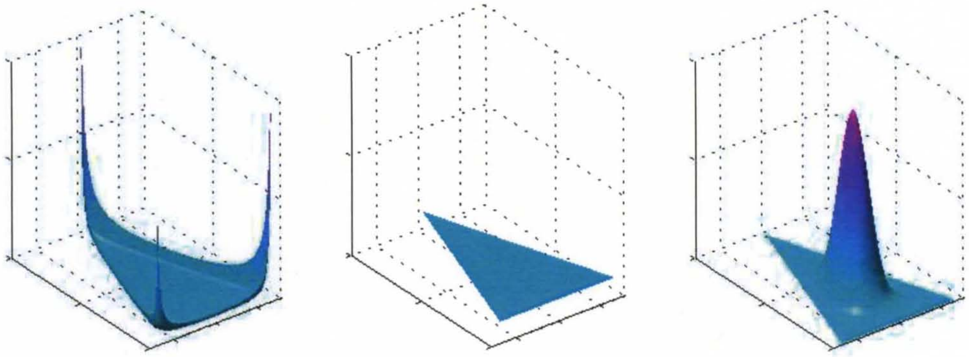


Figure 2.2: Examples of Dirichlet distributions over three variables. The two horizontal axes form the plane of the simplex and the vertical axis represents the value of the density. (Left) $\{\alpha_k\} = 0.1$ favors sparse multinomial distributions. (Center) $\{\alpha_k\} = 1$ lead to a uniform prior. (Right) $\{\alpha_k\} = 10$ represent an unbiased unimodal prior. Figures are from [Bishop, 2006, chap. 2] with permission granted from the author [Bishop].

the mean and variance of the Dirichlet distribution has very simple forms, written as:

$$\mathbb{E}[\mu_k] = \frac{\alpha_k}{\alpha_0} \tag{2.32}$$

$$\mathrm{var}[\mu_k] = \frac{\alpha_k(\alpha_0 - \alpha_k)}{\alpha_0^2(\alpha_0 + 1)} \tag{2.33}$$

The Dirichlet distribution pertains an aggregation property which can be useful when it is appealing to combine a subset of the categories for multinomial data. The aggregation property states that if $\boldsymbol{\mu} \sim \mathrm{Dirichlet}(\boldsymbol{\mu}|\boldsymbol{\alpha})$, the vector of parameters attained by aggregation are also Dirichlet [Gelman et al., 2003]. For example, combining the first two categories gives

$$(\mu_1 + \mu_2, \mu_3, \cdots, \mu_K) \sim \mathrm{Dirichlet}(\mu_1 + \mu_2, \mu_3, \cdots, \mu_K | \alpha_1 + \alpha_2, \alpha_3, \cdots, \alpha_K) \tag{2.34}$$

This also suggests that the marginal distribution of any single component of a Dirichlet distribution follows a special case of Dirichlet distribution with two components, i.e. a beta density:

$$\mu_k \sim \mathrm{Beta}(\mu_k | \alpha_k, \alpha_0 - \alpha_k) \tag{2.35}$$

This representation leads to an alternative, sequential procedure for drawing random samples from Dirichlet density [Gelman et al., 2003].

Conjugate Posteriors and Predictions

Multiplying the likelihood function (Eq.2.30) by the Dirichlet prior (Eq.2.31 distribution, we obtain the posterior distribution for the parameters μ as

$$p(\boldsymbol{\mu}|D,\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0 + N)}{\Gamma(\alpha_1 + C_1)\cdots\Gamma(\alpha_K + C_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k + C_k - 1} = \text{Dirichlet}(\boldsymbol{\mu}|\boldsymbol{\alpha} + \boldsymbol{C}) \tag{2.36}$$

where $\boldsymbol{C} = (C_1, \ldots, C_K)$.

The predictive probability of future observation x^* (as in Eq.2.6) is often of interest. For multinomial-Dirichlet, this takes the form

$$p(x = k|D,\alpha) = \int \text{Multinomial}(x = k|\mu)\text{Dirichlet}(\mu|D,\alpha)d\mu \tag{2.37}$$

$$= \mathbb{E}[\mu_k|D,\alpha_k] = \frac{\alpha_k + C_k}{\alpha_0 + N} \tag{2.38}$$

2.3 Clustering and Mixutre Model

Clustering takes a set of objects and aims to put them into groups that are similar to each other. There are several motivations for clustering. First, a good clustering allows for prediction on unobserved features of an object. Second, clusters facilitate description and communication. A third motivation is that clustering can help highlight interesting objects that deserve special attention. Clustering can be viewed as mixture density modeling, which construct complex probability distributions through combinations of simpler distributions.

Well known clustering algorithms include K-means clustering and hierarchical clustering [Hastie et al., 2009]. The distance metrics used in these algorithms include Euclidean distance, Pearson correlation and squared Pearson correlation. Besides distance-based algorithms, self organized map (SOM), spectral clustering and model-based clustering are also common [Hastie et al., 2009]. This section focuses on K-means algorithms, and the closely related finite mixture model which are the foundations for more complex mixture-density based modeling.

2.3.1 K-means Clustering

The K-means algorithm is a clustering method where each cluster is parameterized by a vector called its mean or center. Each observation is assigned to the cluster whose centers are the nearest to the observation. Since cluster centers are unknown *a prior*, the algorithm starts with some randomly (or

Algorithm 1: K-means Clustering.

```
1 Initialize the means $\mu_k$, and evaluate the initial value of the distortion measure $J$ (Eq. 2.43).;
2 Evaluate the assignments $r_{nk}$ (Eq. 2.39) using the current mean values.
      $$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j ||x_n - \mu_j||^2 \\ 0 & \text{otherwise.} \end{cases} \qquad (2.39)$$
3 Re-estimate the means $\mu_k$ (Eq. 2.42) using the current assignments.
      $$\frac{dJ}{d\mu_k} = 0 \Rightarrow \qquad (2.40)$$
      $$2\sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0 \Rightarrow \qquad (2.41)$$
      $$\mu_k = \frac{\sum_n r_{nk}x_n}{\sum_n r_{nk}} \qquad (2.42)$$
4 Evaluate the distortion measure and check for convergence of either the means or the
  distortion measures. If the convergence criterion is not satisfied return to step 2.
```

carefully) chosen centers. It then iteratively assigns observations to the nearest centers and updates the centers according to the assignment.

Let $X = \{x_1, \ldots, x_N\}$ where x_n $(n = 1, \ldots, N)$ is an observation of a D-dimensional random variable. Let K $(K \leq N)$ be a pre-specified number of clusters. The assignment of observations to clusters can be represented by a set of binary indicator variables $r_{nk} \in \{0, 1\}$ $(n = 1, \ldots, N; k = 1, \ldots, K)$, so that $r_{nk} = 1$ and $r_{nj} = 0$ for $j \neq k$ if observation x_n is assigned to cluster k. Let μ_k $(k = 1, \ldots, K)$ be a D-dimensional vector associated with the kth cluster. The goal of clustering is estimating $\{r_{nk}\}$ and $\{\mu_k\}$ to minimize the loss function, or distortion measure, defined by

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n - \mu_k||^2 \qquad (2.43)$$

That is,

$$\{\widehat{r_{nk}}\}, \{\widehat{\mu_k}\} = \underset{\{r_{nk}\},\{\mu_k\}}{\arg\min}\ J \qquad (2.44)$$

Algorithm 1 shows the iterative procedure for the estimation of these parameters. Fig. 2.3 illustrates the K-means algorithm on a sample data set.

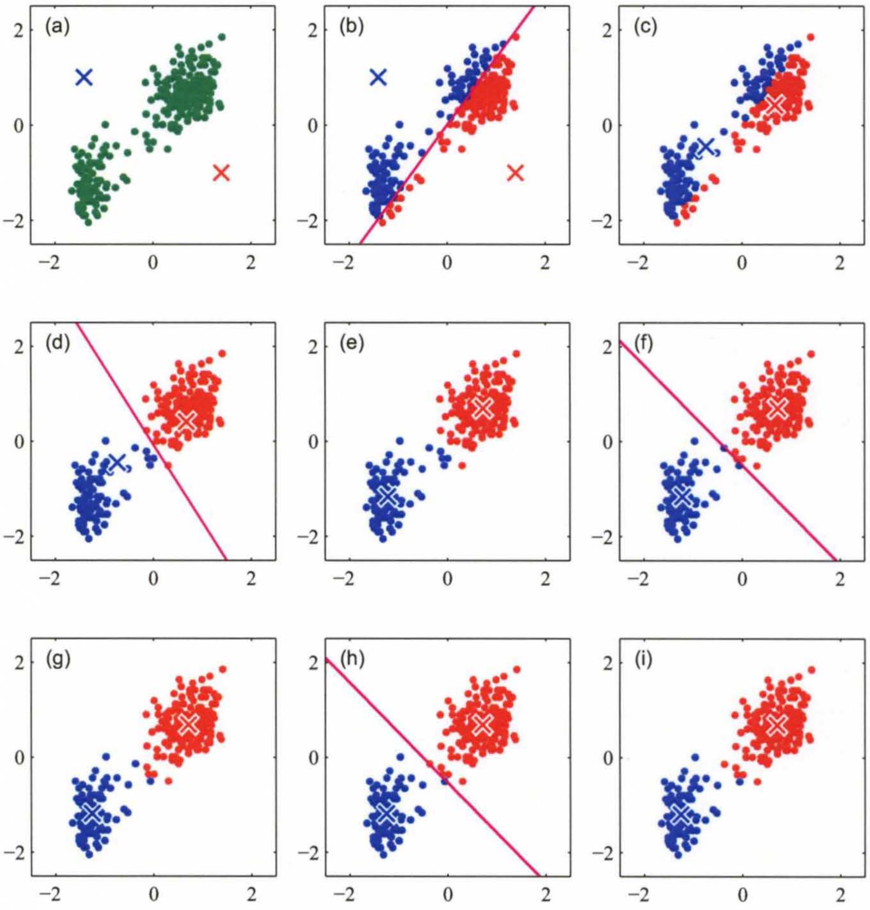


Figure 2.3: Illustration of K-means clustering. (a) Green points denote the observations in a two-dimensional Euclidean space. Red and blue crosses represent the initial choice of centers. (b) Using the current mean values, each data point is assigned either to the red cluster or to the blue cluster, according to its distance to the centers. (c) Cluster centers are updated according to the cluster assignment. (d)-(i) Repeat the two steps until convergence. Figures are from [Bishop, 2006, chap. 9] with permission granted from the author [Bishop].

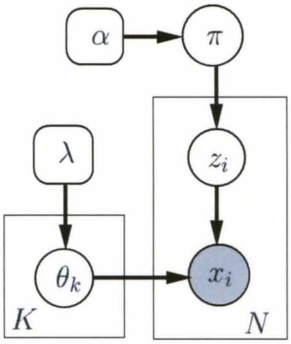


Figure 2.4: A directed graphical representation of a finite mixture model. Grey circle node denotes observed variables, white circle nodes represent latent variables, and square nodes denote hyperparameters. Plates denote repeated variables over the N observations and K components. Specifically, $\{x_i\}$ are the observations, $\{z_i\}$ are the latent component variables, and $\{\theta_i\}$ are component model parameters. The observations are conditionally independent given the cluster assignment.

2.3.2 Finite Mixture Model

Mixtures of distributions is considered as a useful extension to the "standard" probability distributions. A mixture density is formed by taking linear combinations of other distributions. A **mixture** of distributions can be described as

$$p(x) = \sum_{k=1}^{K} \pi_k f_k(x), \qquad \sum_{k=1}^{K} \pi_k = 1, K > 1 \tag{2.45}$$

In most cases, f_k are from a parametric family such as Gaussians or Bernoulli distributions, with unknown parameter θ_k, leading to the parametric mixture model

$$p(x) = \sum_{k=1}^{K} \pi_k p_k(x|\theta_k) \tag{2.46}$$

Given N independent observations $D = \{x_1, \ldots, x_N\}$, the likelihood function has the form

$$p(D|\boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k p(x_n|\theta_k) \tag{2.47}$$

Eq. 2.46 formulates the mixture model as a linear combination of other distributions. We see from this representation that the maximum likelihood estimation for the parameters does not have a closed-form analytical solution. Alternatively, mixtures can also be formulated in terms of discrete **latent** (or auxiliary) **variables**, which motivates the **expectation-maximization (EM)** algorithm [Bishop, 2006]. It is possible to associate to a random variable x in Eq. 2.46 another random variable z such that

$$\begin{aligned} x_n | z_n = k &\sim p(x_n|\theta_k) \\ z_n|\pi_1, \cdots, \pi_k &\sim \text{Multinomial}(z_n|\pi_1, \cdots, \pi_k) \end{aligned} \tag{2.48}$$

Here the latent variable z_n is a multinomial random variable identifying to which of the K components the observation x_n belongs. The EM algorithm considers the problem of maximizing the likelihood for the "complete" data set $\{D, \boldsymbol{z}\}$. The likelihood function takes the form

$$p(D, \boldsymbol{z}|\boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{k=1}^{K} \prod_{\{n|z_n=k\}} \pi_k p(x_n|\theta_k) \tag{2.49}$$

Figure 2.4 shows a probabilistic graphical representation of the finite mixture model. Equivalently, we can also represent it using a binary variable $z_{nk} \in \{0, 1\}$ and let $z_{nk} = 1$ if and only if $c_n = k$.

$$p(D, Z|\theta, \pi) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} p(x_n|\theta_k) \tag{2.50}$$

Comparison with the likelihood function in 2.47 for the partial observation D shows that the summation over K components have been replaced by multiplication. The logarithm of the new likelihood function now acts directly on the probability distribution $p(x_n|\theta_k)$, which if being a member of the exponential family, leads to a much simpler solution. Algorithm 2 describes the EM algorithm applied on a finite Gaussian mixture model where each mixture component is represented by a Gaussian distribution.

2.4 Dirichlet Process

Realistic datasets are usually generated from a complex world which can not be adequately described by most standard parametric models. A mixture model is usually formulated as a type of approximation of unknown distributions. **Nonparametric** modeling methods avoid assuming restricted functional forms and allow defining flexible models with unbounded complexity. The focus of this section will be on Dirichlet Process and mixtures [Ferguson, 1973, Aldous, 1985], an important technique in the Bayesian nonparametric family.

A **Dirichlet Process** defines a distribution on random probability measures, or equivalently non-negative functions which integrate to one. A Dirichlet process is parameterized by a **base measure** H over a measure space Θ, and a positive scalar **concentration parameter** α. Consider

Algorithm 2: EM for Finite Gaussian Mixtures.

```
1 Initialize the means $\mu_k$, covariances $\Sigma_k$ and mixing coefficients $\pi_k$, and evaluate the initial
  value of the log likelihood in Eq. 2.47.
2 Evaluate the responsibilities $\gamma(z_{nk})$ (Eq. 2.51) using the current parameter values.
      $\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}.$   (2.51)
3 Re-estimate the parameters $\mu_k, \Sigma_k, \pi_k$ (Eq. 2.52, 2.53, 2.55) using the current responsibilities.
      $\frac{d \ln \mathcal{L}}{d \mu_k} = 0 \Rightarrow \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n,$   (2.52)
      $\frac{d \ln \mathcal{L}}{d \Sigma_k} = 0 \Rightarrow \Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T,$   (2.53)
      $\frac{d \ln \mathcal{L}}{d \pi_k} = 0 \Rightarrow \pi_k = \frac{N_k}{N},$   (2.54)
      $\text{where } N_k = \sum_{n=1}^{N} \gamma(z_{nk}).$   (2.55)
4 Evaluate the log likelihood and check for convergence of either the parameters or the log
  likelihood. If the convergence criterion is not satisfied return to step 2.
```

a finite partition $(A_1, \cdots, A_K)$ of Θ:

$$\bigcup_{k=1}^{K} A_k = \Theta \qquad A_k \cup A_l = \emptyset \qquad k \neq l \tag{2.56}$$

A random probability distribution G on Θ is drawn from a Dirichlet process if its measure on every finite partition follows a Dirichlet distribution:

$$(G(A_1), \cdots, G(A_K)) \sim \text{Dirichlet}(\alpha H(A_1), \cdots, \alpha H(A_K)) \tag{2.57}$$

2.4.1 Stick-Breaking Construction

The definition of Dirichlet process in Eq. 2.56 suggests several implicit properties. However, it does not provide a mechanism for sampling from Dirichlet processes, or predicting future observations. Sethuraman [1994] provides an explicit definition of the Dirichlet process, called the **stick-breaking** construction, which shows that Dirichlet measures are discrete with probability one. This construction definition simplifies the definitions of Dirichlet process and also leads to a simple model for predictive distributions known as the Chinese restaurant process. The stick-breaking construction generates a random measure G from Dirichlet process, $G \sim \mathrm{DP}(\alpha, H)$, through the following construction procedure:

- First, construct mixture weights π from the stick-breaking process:

$$\beta_k \sim \mathrm{Beta}(1, \alpha) \qquad k = 1, 2, \cdots \tag{2.58}$$

$$\pi_k = \beta_k \prod_{l=1}^{k-1} (1 - \beta_k) \tag{2.59}$$

- Then, construct random measure G as an infinite sum of weighted point masses:

$$G(\theta) = \sum_{k=1}^{\infty} \pi_k \delta(\theta, \theta_k) \qquad \theta_k \sim H \tag{2.60}$$

The stick-breaking distribution over π is sometimes written $\pi \sim$ GEM, contributed by Griffiths, Engen and McCloskey [Pitman, 2006].

2.4.2 Blackwell-MacQueen Urn Scheme

The stick-breaking construction shows that Dirichlet measures G produced by Dirichlet processes are discrete with probability one. Therefore, the probability of multiple observations (drawn from G) taking identical values is positive. The stick-breaking construction enables a simpler derivation of the predictive probability of new observations. Consider a random probability measure G drawn from a Dirichlet process: $G \sim \mathrm{DP}(\alpha, H)$, where the base measure H has density $h(\theta)$. Consider a set of N observations $\theta_n \sim G$, the predictive distribution for a new observation θ_{N+1}, conditioned on $\theta_1, \cdots, \theta_N$ and with G marginalized out, is written as:

$$p(\theta_{N+1} = \theta | \theta_1, \cdots, \theta_N, \alpha, H) = \frac{1}{\alpha + N} \left(\alpha h(\theta) + \sum_{n=1}^{N} \delta(\theta, \theta_n) \right) \tag{2.61}$$

This generative process can be interpreted metaphorically via an extended Polya urn model called Blackwell-MacQueen Urn Scheme [Blackwell and MacQueen, 1973]. Consider each value in the measure space Θ is a unique color, each value θ_n drawn from the process is a ball with the color corresponding to the value. Consider an urn containing one ball for each proceeding observations. For the drawing steps, we will either, with probability $\frac{\alpha}{\alpha+N}$, pick a new color (draw $\theta_{N+1} \sim H$) to paint a ball and put the ball into the urn, or, with probability $\frac{N}{\alpha+N}$ pick a ball from the urn, paint a new ball with the same color as the picked ball and drop both balls into the urn. The Blackwell-

MacQueen urn scheme has been used to show the existence of the Dirichlet process [Blackwell and MacQueen, 1973]. The procedure can also be used to sample observations from a Dirichlet process, without explicitly constructing the underlying mixture $G \sim \mathrm{DP}(\alpha, H)$ and marginalize it.

2.4.3 Chinese Restaurant Process

We see from Eq. 2.61 that draws from a Dirichlet process have the properties of discreteness and clustering. Since the values of draws are repeated, let $\theta_1^*, \cdots, \theta_K^*$ be the distinct values assigned to the observations $\theta_1, \cdots, \theta_N$. The predictive distribution of Eq. 2.61 can be equivalently written as

$$p(z_{N+1} = z | z_1, \cdots, z_N, \alpha) = \frac{1}{\alpha + N} \left(\alpha \delta(z, K+1) + \sum_{k=1}^{K} \delta(z, k) \right) \tag{2.62}$$

where

$$z_n = k \text{ if } \theta_n = \theta_k^* \qquad n = 1, 2, \ldots, N+1 \qquad k = 1, 2, \ldots, K$$

$$z_{N+1} = k + 1 \text{ if } \theta_{N+1} = \theta_{K+1}^* \text{ is a new value sampled from } H$$

and

$$C_k = \sum_{n=1}^{N} \delta(z_n, k) \tag{2.63}$$

The distribution over partitions is called by Pitman and Dubins the **Chinese Restaurant Process** (CRP) [Pitman, 2006]. The name comes a metaphor useful in interpreting Eq. 2.62. In this metaphor there is a Chinese restaurant with an infinite number of tables, each of which can seat an infinite number of customers. The first customer enters the restaurant and sits at the first table. The remaining customers enter the restaurant one by one and sit at a table with other customers, or a new table by itself. In general, the $N+1$th customer either sits at an already occupied table k with probability proportional to the number of already seated diners C_k, or sits at an empty new table with probability proportional to α. There is not an *a priori* distinction between the unoccupied tables.

CRP defines an exchangeable distribution on partitions such that the joint distribution is invariant to the order in which observations are assigned to clusters. Another important clustering property induced by CRP is the expected number of clusters among N observations. Since the nth observation has the probability $\frac{\alpha}{\alpha+i-1}$ taking on a new value (being assigned to a new cluster), the expected K is:

$$\mathbb{E}[K|N] = \sum_{n=1}^{N} \frac{\alpha}{\alpha+n-1} \in O(\alpha \log n) \tag{2.64}$$

That is, α controls the number of clusters *a priori*: a larger α indicates a larger number of clusters *a priori*. The number of clusters grows only logarithmically with respect to the number of observations.

2.4.4 Dirichlet Process Mixture Model

By far the most common application of the Dirichlet process is as a nonparametric prior over components in a mixture model. The nonparametric nature of the Dirichlet process allows for an infinite number of components within a mixture model. Consider a set of observations $D = \{x_1, \ldots, x_N\}$. Each x_i has a corresponding latent parameter θ_i, so that x_i is drawn from some parameterized family $F(\theta_i)$. Each θ_i is sampled independently and identically from G. This describes the **Dirichlet process mixture model**:

$$\begin{aligned} x_i|\theta_i &\sim F(\theta_i) \\ \theta_i|G &\sim G \\ G|\alpha, H &\sim DP(\alpha, H) \end{aligned} \tag{2.65}$$

As discussed, the stick-breaking construction guarantees a random measure sampled according to a Dirichlet process. This implies the following representation of a Dirichlet process mixture model:

$$\begin{aligned} x_i|\theta_i &\sim F(\theta_i) \\ \theta_i|G &\sim G \\ G(\theta) &= \sum_{k=1}^{\infty} \pi_k \delta(\theta, \theta_k) \\ \theta_k &\sim H \\ \pi &\sim \mathrm{GEM}(\alpha) \end{aligned} \tag{2.66}$$

To highlight the mixture perspective of the Dirichlet process mixture model, there is another useful representation. As in the description on CRP, each observation x_i is associated with a latent variable z_i indicating to which mixture component the observation is assigned to. Then Eq.2.66 is equivalently

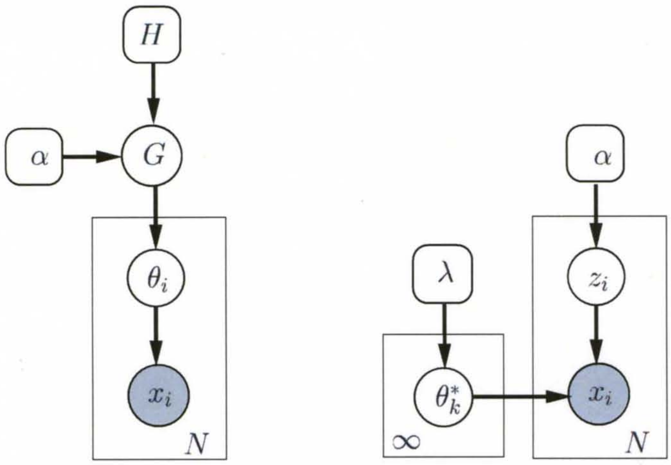


Figure 2.5: Directed graphical representations of the Dirichlet process mixture model. (Left) Standard Dirichlet process representation. Each observation x_i is generated from a directly associated model parameter θ_i, which is drawn from G, an infinite discrete distribution on the parameter space. (Right) Latent variable representation. Each x_i is generated according to its component assignment, i.e. $x_i \sim F(\theta^*_{z_i})$, where z_i is generated from a multinomial distribution, i.e. $z_i \sim \text{Multinomial}(\boldsymbol{\pi})$, and $\theta^*_{z_i} \sim H(\lambda)$. The mixture weights $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$ follow a stick-breaking process. The number of mixture components is not fixed *a priori.*

expressed as

$$\begin{aligned} x_i|z_i, \{\theta^*_k\} &\sim F(\theta^*_{z_i}) \\ \theta^*_k|H &\sim H \\ z_i|\boldsymbol{\pi} &\sim \text{Multinomial}(\boldsymbol{\pi}) \\ \boldsymbol{\pi}|\alpha &\sim \text{GEM}(\alpha) \end{aligned} \tag{2.67}$$

Equivalently, we may use the Chinese restaurant process (or the Blackwell-MacQueen Urn scheme) to express the model:

$$\begin{aligned} x_i|z_i, \{\theta^*_k\} &\sim F(\theta^*_{z_i}) \\ \theta^*_k|H &\sim H \\ z_i|z_1, \cdots, z_{i-1}, \alpha &\sim \text{CRP}(\alpha, \{C_k\}) \end{aligned} \tag{2.68}$$

where $\{C_k\}$ is as defined in Eq.2.63.

Rather than fixing the number of clusters as in finite mixture model (Eq. 2.46), the Dirichlet process mixture model allows for a countably **infinite** number of clusters, thus an infinite mixture model. In the Dirichlet process mixture model, the actual number of clusters is not fixed, and can be inferred from data using Bayesian posterior inference. An alternative derivation of the Dirichlet process mixture model is through the finite mixture model and takes the number of components K to an infinite limit, as an infinite mixture model [Rasmussen, 2000]. In this limit, predictions based on the finite mixture model approach those of the corresponding Dirichlet process. Figure 2.5 shows directed graphical representations of the Dirichlet process mixture model.

Inference for DPMs

Exact computation of the posterior distribution for a DPM model is infeasible. Approximate inference methods are required for estimating posterior quantities under DPM. One of the methods of choice for DPM is Markov chain Monte Carlo (MCMC) sampling [Escobar and West, 1995, Neal, 1998], which samples from the posterior distribution of the model parameters by simulating a Markov chain which has this as its equilibrium distribution. The simplest such methods are based on Gibbs sampling, which repeatedly and iteratively draws values for each parameter θ_i from its conditional distribution given both the observations D and the model parameters θ_{-i} ($-i$ denotes $\{1, \cdots, i-1, i+1, \cdots N\}$). As pointed out by [Neal, 1998], although the algorithm produces an ergodic Markov chain, convergence to the posterior distribution may be very slow and sampling may be inefficient. The problem lies in the fact the algorithm cannot change the θ for more than one observation simultaneously.

The problem is alleviated if Gibbs sampling is instead applied to the model formulated through latent indicator variable as in Eq. 2.67 or Eq. 2.68. Each Gibbs sampling scan consists of drawing a new value for the latent indicator variable z_i from its conditional distribution given the data D, the model parameters $\boldsymbol{\theta}^*$, and z_{-i} ($-i$ is defined as $\{1, \ldots, i-1, i+1, \cdots, K+1\}$ where K is the number of mixture components so far) and then drawing a new value for each θ_k^* from its conditional distribution given x_i for which $z_i = k$. When a new value is chosen for $\boldsymbol{\theta}^*$, the values of $\theta_i = \theta_k^*$ will change automatically and simultaneously for all observations associated with component k (all x_i such that $z_i = k$).

Finally, in a conjugate context where the distribution of each mixture components has a conjugate prior, we can often integrate analytically over the component parameters $\boldsymbol{\theta}^*$. The state of the Markov chain then consists only of the latent indicator variables z_i. Note that due to the incremental update nature of algorithm, the Gibbs sampling method can still become trapped in isolated modes corresponding to an inappropriate clustering of the data points. Split-merge Markov chain algorithms have been proposed to further alleviate the problem of inefficient sampling. The Metropolis-Hastings procedure aims to escape such local modes by splitting or merging mixture component according to proposals obtained from a restricted Gibbs sampling scan.

Variational inference provides an alternative, deterministic methodology for approximating likelihoods and posterior. The approach is based on reformulating the problem of posterior distribution estimation as an mathematical optimization problem, relaxing this object function, and then derive computational tractable algorithms which bound or approximate the statistics of interest [Wainwright and Jordan, 2008]. Blei and Jordan [2006] developed a mean-field variational algorithm for the DP mixture, based on the stick-breaking representation of the DPM. The Bayesian Hierarchical Clustering (BHC) by Heller and Ghahramani [2005a] introduces another approximation strategy to DPM. The method represents clustering through a tree structure with each node associated with a set of data point and posterior of merging, and computes the marginal likelihood and posterior prediction by summarizing over all the tree-consistent partitions where the number of them is exponential in the number of data point for balanced binary trees.

CHAPTER 3

MOLECULAR BIOLOGY AND MOLECULAR EVOLUTION

This chapter provides a brief review on the basic concepts in molecular biology (Sec. 3.1) and molecular evolution (Sec. 3.3) relevant to our applications. For a detailed treatment of the concepts, readers may consult Watson et al. [2008], Graur and Li [2000].

3.1 Molecular Biology

Heredity of every living organism is controlled by its **genome**, or DNA. The sequence of the individual subunits, or bases, of the DNA determines the development of the organism. Through a complex series of interactions, the DNA sequence produces all of the proteins of an organism. Proteins serve a variety of roles in an organism's development and function. Physically, the genome may be divided into a number of **chromosomes**. Functionally, the genome is divided into **genes**, each of which encodes a single type of RNA or polypeptide. All complex cellular characteristics are under the control of many genes, rather than a one gene-one characteristic mapping. When two genes are on the same chromosome, they tend to be inherited together. Genes affecting different characteristics are sometimes inherited independently of each other, since they are located on different chromosomes. Normally, genes are extremely stable and are copied exactly during chromosome duplication; inheritable changes (mutations) in genes, occur at infrequent rates and can have harmful consequences.

3.1.1 The Central Dogma of Molecular Biology

The **Central Dogma of Molecular Biology** was coined by Francis Crick in 1958 [Crick, 1958] and re-stated in 1970 [Crick, 1970], and refers to the hypothesis that chromosomal DNA acts as the template for RNA molecules, which subsequently move to the cytoplasm, where they determine the arrangement of amino acids within proteins. Figure 3.1 illustrates the Central Dogma of Molecular Biology. The arrows denotes the directions assumed for the transfer of genetic information. The arrow encircling DNA indicates that DNA is the template for its self-replication. The arrow from DNA to RNA indicates that RNA synthesis (**transcription**) is directed by a DNA template. Correspondingly, protein synthesis (**translation**) is directed by an RNA template. Note that the last two arrows are unidirectional, signifying that RNA is never made from protein templates, nor is DNA made from RNA templates. The Central Dogma still remains the dominant paradigm of molecular biology after its original proclamation about 60 years ago. It is still true that proteins never act as templates for RNA, though it has been discovered that, in very rare cases, RNA sequences can serve as templates for DNA chains of complementary sequence. As a result, in rare occasions, information from a cellular RNA is converted into DNA and inserted into the genome.

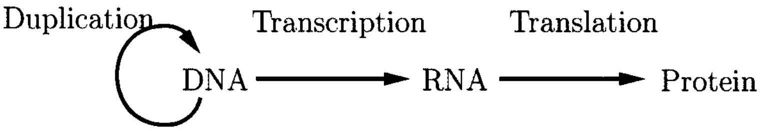


Figure 3.1: The Central Dogma of Molecular Biology.

3.1.2 Genes are DNA

It was known that chromosomes possessed a unique molecular material, deoxyribonucleic acid (DNA), but it was initially unclear that this material carried genetic information. That DNA might be the key genetic molecule emerged when Frederick Griffith observed that nonvirulent strains of the pneumonia-causing bacteria became virulent when mixed with their heat-killed pathogenic counterparts [Griffith, 1928]. The observations of such **transformations** supports the genetic interpretation and motivates the search for the chemical identity of the transformation agent. However, at that time, the belief that genes were proteins was still held by the majority of biochemists. In 1944, through their purification of the **transformation principle**, Oswald T. Avery, Colin M. MacLeod and Maclyn McCarty announced that the active genetic material was DNA [Avery et al., 1944]. Further experiments demonstrated that, besides bacteria, all known organisms and many viruses use DNA as it genetic material. Some viruses, though, use RNA acts as the genetic material.

The underlying structure of DNA remained a puzzle. By the 1950s, the observation by Erwin Chargaff led to the concept that genetic information is carried in the form of a sequence of

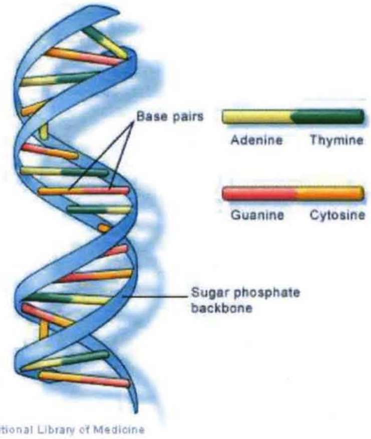


Figure 3.2: Illustration of the DNA structure. DNA is a double helix formed by base pairs attached to a sugar-phosphate backbone. (Image courtesy of the U.S. National Library of Medicine [U.S. National Library of Medicine])

bases [Chargaff, 1950]. The experiments of Chargaff also showed that regardless of the absolute amounts of each base, the relative ratios of the four bases were not random. The number of adenines (A) in all DNA samples was equal to the number of thymines (T), and the number of guanines (G) was always the same as the number of cytosines (C). High-quality X-ray diffraction photographs taken by Maurice Wilkins and Rosalind Franklin [Franklin and Gosling, 1953] suggested that the underlying DNA structure was helical and consisted of more than one polynucleotide chain. Building on these observations, in 1953, James D. Watson and Francis H. Crick announced the elegant and correct solution, a complementary double helix [Watson and Crick, 1953]. In the double helix, as illustrated in Fig. 3.2, the two DNA strands are held together by hydrogen bonds between pairs of bases on the opposing chains. This base pairing is very specific: the purine adenine only hydrogen bonds to the pyrimidine thymine, and the purine guanine only hydrogen bonds to the pyrimidine cytosine. The two intertwined strands of complementary bases suggested the exciting hypothesis that one strand acts as the specific template that directs the synthesis of the other strand. The experiments by Arthur Kornberg demonstrated that DNA is the direct template for its own formation [Lehman et al., 1958, Bessman et al., 1958]. The research by Matthew Meselson and Frank W. Stahl in 1958 showed that DNA replication was a semiconservative process where each individual strands of the double helix remain intact and is distributed into one of the two daughter DNA [Meselson and Stahl, 1958].

3.1.3 RNA

Although DNA must carry the genetic material for protein synthesis, it does not serve as a direct role for this process because it had been shown that protein synthesis occurs at sites (cytoplasm for all eukaryotic cells) where DNA is absent. There had to be another type of molecule which obtains its genetic information from DNA, and carries it to the cytoplasm to serve as the template for protein synthesis. This second information-carrying molecule is ribonucleic acid (RNA). The chemical structure of RNA is very similar to DNA: it also is a long, unbranched molecule containing four types of nucleotides linked together by 3'→ 5' phosphodiester bonds. Two differences distinguish RNA from DNA: first, the sugar of DNA is deoxyribose, while in RNA it is ribose. The second is that RNA contains no thymine, but the closely related pyrimidine uracil. Unlike DNA, RNA is typically found in the cell as a single-stranded molecule. The RNA that carries information from DNA to the ribosomal sites of protein synthesis is called **messenger RNA (mRNA)**. Only 4% of total cellular RNA is mRNA. About 10% of cellular RNA is **transfer RNA (tRNA)** molecules, which serves as an adaptor between the four-letter genetic code in mRNA and the twenty-letter code of amino acids in proteins. **Ribosomal RNA (rRNA)** accounts for 85% of all cellular RNA found in ribosomes. Although initially considered as the template for ordering amino acids, the rRNA, together with about fifty different ribosomal proteins binding to them, function as the factories for protein synthesis. They bring together the tRNA and amnio acid precursors into position where they decode the information provided by the mRNA templates.

RNA molecules plays an essential role in gene expression regulation. Short RNAs have a distinct role in gene regulation: they can repress the expression of genes with homology to them. This repression, called **RNA interference (RNAi)**, can function to inhibit translation of the mRNA or degradation of the mRNA. **Short interfering RNAs (siRNAs)** are short double-stranded fragments about 23 nucleotides long. Once a given siRNA has been assembled within a complex called **RISC** (RNA-induced silencing complex), they together inhibit expression of a homologous gene in three possible ways: destruction of its target mRNA; inhibition of the translation of its target mRNA; or induce of chromatin modifications within the promoter which silence the gene. **microRNAs (miRNAs)** are another type of RNAs that repress gene expression in a similar way as siRNAs.

3.1.4 Gene Expression

The Central Dogma states the unidirectional information flow from genes to proteins, beginning with the form of linear nucleotides sequence in a polynucleotide chain, to the form of linear amino acids sequence in a polypeptide chain. Understanding the detailed mechanism of the flow, or how the gene is expressed, remains an exceptionally challenging task. Transcription (Fig. 3.3) refers to the process by which genetic information in the form of a nucleotide sequence is transferred from DNA to RNA. Translation refers to the process by which genetic information contained in an mRNA in the form of nucleotide sequence is used to direct the ordering of amino acids into a polypeptide chain of a protein. Chemically and enzymatically, transcription is very similar to DNA replication. However, the two processes have profound differences and serve completely different purposes. In DNA replication, the entire genome is copied once during cell division. In transcription, only certain parts of the genome are selected and the number of copies out of the parts can range from one to thousands. The choices of which regions to transcribe, and of what extent, can be regulated. The protein products of genes represent an enormous variety of structures and functions, including structural, enzymatical, and regulatory functions. A gene encoding a protein or RNA involved in regulating the expression of other genes is called a regulatory gene.

The coding sequence of a gene is a series of three-nucleotide codons. In many eukaryotic genes, these blocks of coding sequences are separated from each other by blocks of noncoding sequences. The coding sequences are called **exons** and the intervening sequences are called **introns**. A gene, part of the entire genome, is transcribed into a single RNA copy. This RNA copy is the primary transcript (or **pre-mRNA**) and contains the same introns and exons as the orignial gene. The protein-synthesizing factory of the cell is only able to translate mRNAs containing a contiguous stretch of codons; It cannot recognizing and skipping over a block of intron. Therefore, the pre-mRNA must have their introns removed before they can be translated into protein. The process by which introns are removed from the pre-mRNA is called **RNA splicing** (Fig. 3.4). Some pre-mRNAs can be spliced in more than one way, for example, through different combinations of exons, generating alternative mRNAs. This process is called **alternative splicing**, through which a gene can lead to more than one form of a polypeptide product. It is estimated that about 95% of the genes in the human genome are spliced in alternative ways and allow for different protein isoforms per gene. The number of different mature mRNA that a given gene can encode through alternative splicing varies from two to even thousands.

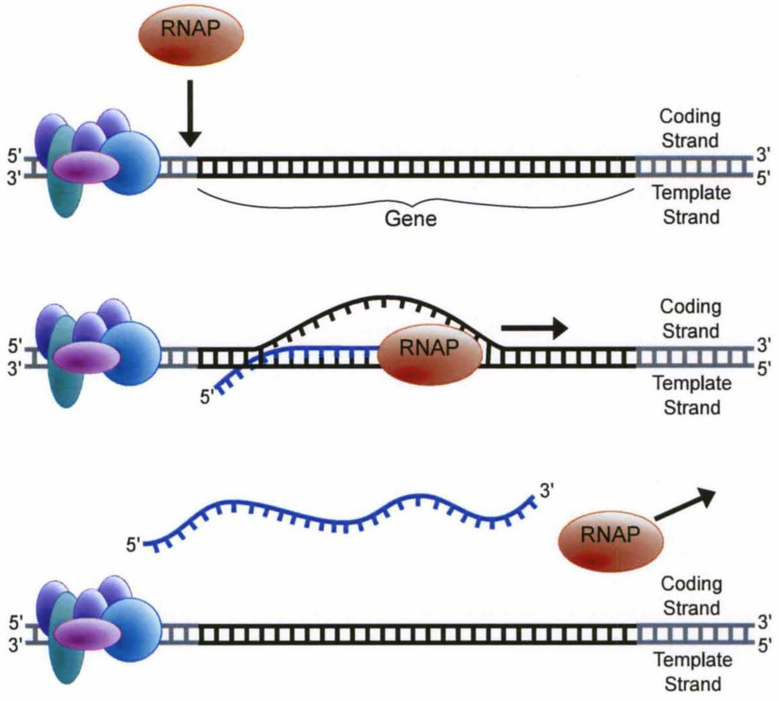


Figure 3.3: An illustration of transcription, which is the process by which genetic information in the form of nucleotide sequence is transferred from DNA to RNA. (Top) Initiation, by which RNA polymerase (RNAP) binds to a promoter in DNA. (Center) Elongation, by which one strand of the DNA is used as a template for RNA synthesis. (Bottom) Termination, by which the newly synthesized RNA is released from the elongation complex. Figure from wikipedia with permission granted from the author [wikipedia, d].

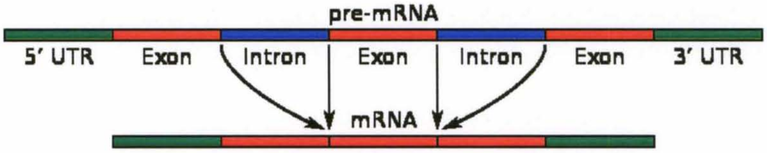


Figure 3.4: An illustration of RNA splicing, by which introns are removed from the pre-mRNA and exons are linked to form a mature mRNA. Figure from wikipedia with permission granted from the author [wikipedia, b].

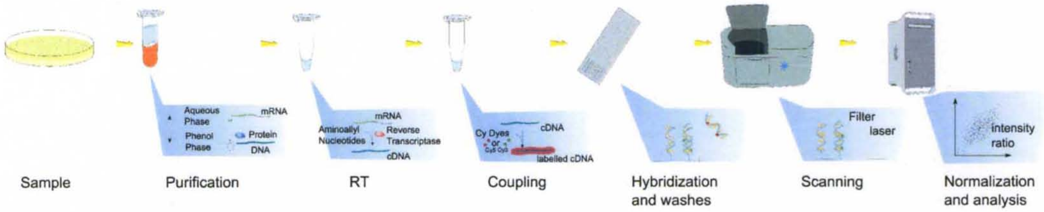


Figure 3.5: An illustration of the procedure for DNA microarray experiment. Figure from wikipedia with permission granted from the author [wikipedia, a].

3.1.5 DNA Microarray Technology

It is often desirable to determine the expression level of a specific mRNA in two different cell types, or the same cell type of different conditions. This type of information can be obtained based on the process of hybridization. Because the two strands of the double helix are held together by relatively weak (noncovalent) forces, DNA strands can separate and reassociate. **Hybridization** refers to the process where two complementary single-stranded nucleic acids meet together and reform regular double helices. Hybrids can be formed between complementary strands of DNA and/or RNA. Hybridization has been the basis for many important techniques in molecular biology, such as Southern blots [Southern, 1975], as well as DNA microarrays. A Southern blot allows for the identification of the amount of a specific gene. In this procedure, thousands of DNA fragments are generated by cutting the genome into discretely visible bands. The cut DNA are then separated by gel electrophoresis, and the double strands of each fragment is separated in alkali. These strands are then incubated with a **probe** which is a defined DNA sequence of interest-either a purified fragment or a chemically synthesized DNA molecule. Probing is performed under specific conditions of salt concentration and temperature such that the probe DNA will only hybridize tightly to its exact complement. This hybridization activity can be detected by a variety of medias such as films that are sensitive to the light or electrons emitted by the labeled DNA. A northern blot is a similar procedure used to identify a particular mRNA in a population of RNAs, where hybridizations are between complement strands of RNA and DNA.

DNA microarray (Fig. 3.5) is based on the same principles of Southern and northern blots, but allows for massive and parallel experiments. Advanced technology allows tens of thousands of probes matching all known mRNAs in a cell to be attached to a small surface of one square centimeter, forming a DNA micoarray. Samples of mRNA from cells, tissues, and other biological sources are labeled with fluorescence and added to the array for hybridization. Molecules in the fluorescent sample react with probes on the chip, causing each spot to glow with an intensity proportional

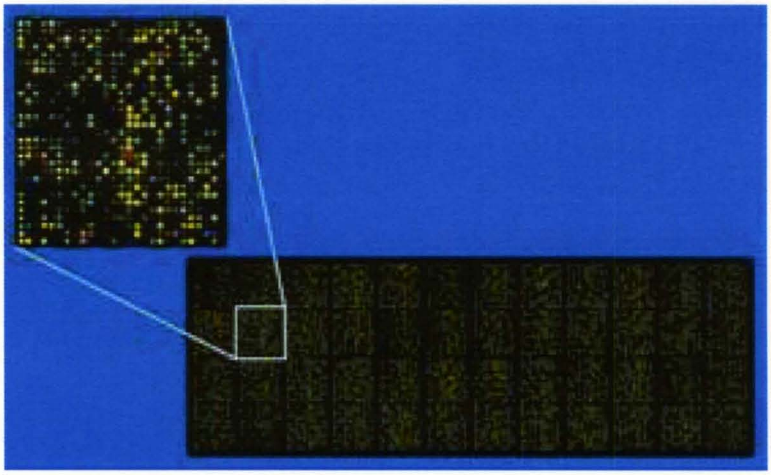


Figure 3.6: An example of an approximatedly 40,000 probe spotted oligonucleotide array. Figure from wikipedia with permission granted from the author [wikipedia, c].

to the abundance of the mRNA. After scanning the surface of the chip, the amount of sample hybridized to each of the probes can be quantified from the image. High throughput experimental data generated from microarray enable gene expression profiling on a genomic scale. Gene expression profiling through microarray technology are also used for identification of microRNAs involved in specific cellular process.

Image Analysis

Image analysis of microarray experiments aims to extract intensity information for each spot on the scanned array (Fig. 3.6). It can have a substantial effect on subsequent microarray data analysis. The analysis of the image consists of several steps. **Gridding** seeks to identify each spot on the array by aligning a grid to the spots, which are arranged in columns and rows on the array. **Segmentation** aims to separate the identified spots from the background. **Intensity extraction** summarizes each segmented spot with an intensity value, which will be used for statistical analysis. Typical intensity measure are the mean or median intensity from the distribution of all pixels within the spot. **Background correction** seeks to adjust the intensity according to the background signal and spatial bias on an array.

Normalization

The extracted intensities enable the comparison of gene expression profiles from different cells or tissues, or under different conditions. This comparison seeks to identify each gene or RNA whose expression varies in response to genetic and environmental differences. This type of variation is referred to as **interesting variation**. However, before applying any statistical analysis techniques to the intensity values, it is essential to assure that the values extracted from multiple arrays are comparable. This is due to **obscuring variation** or **systematic bias**, i.e. variation introduced during the sample preparation, manufacture of the arrays, and the processing of the arrays (labeling, hybridization, and scanning). The comparison and analysis cannot be trusted unless arrays are appropriately **normalized**, i.e. systematic bias are removed. Stafford [2008] provies a review on various methods for normalizing gene expression arrays. One strategy of normalization is based on the assumption that some genes are **invariant** or non-differentially expressed across samples. Normalization is then achieved by mathematically adjusting the intensities of the arrays until the expressions of the invariant genes are equal in all the arrays. In the past, housekeeping control genes were assumed expressed at a constant level and were frequently used for normalization. However, it has been reported that the expression levels of housekeeping genes can vary significantly. Another method is to assume that the total mass of mRNA in a cell is constant, and therefore the intensities are adjusted such that the masses of different arrays are equal. Alternatively to these approaches, **spike control** is a technique that introduces an external RNA to the mRNA samples during preparation. The spiked transcript is amplified and labeled the same way as the other transcripts and hybridized with a unique probe on the arrays. The mRNA intensities are then normalized according to the external controls on different arrays being equal to each other. The spiked transcript must not match any gene in the RNA samples. One could also use many spiked control RNAs with different concentrations to improve normalization accuracy. The mathematical adjustment can be linear, such as scaling (multiplying) a constant to the intensities; or non-linear, such as fitting smoothing splines.

When there is no *a priori* knowledge about the choice of invariant genes, these genes may be inferred from the arrays using mathematical techniques. One such method is to first rank genes according to their intensities, and then consider the genes whose rank does not differ more than a threshold value between arrays as invariant genes. **Quantile normalization** is also frequently used, which assumes that the probe intensity distributions between arrays should be the same.

3.2 Gene Expression Data and Clustering

Genome wide gene expression information generated from various high throughput biological experiments, such as microarrays, provide unprecedented opportunities for understanding the associational and causal relationships among genes, diseases, and environmental factors. A wide range of statistical and computational methods have been designed and applied to gene expression studies. For example, statistical hypothesis testing, in conjunction with some correction for chance, are used to select genes differentially expressed across various conditions under study (e.g. treatment vs. control). Supervised learning, such as Fisher discriminant analysis, logistic regression, and support vector machine (SVM), are used to predict phenotypes (e.g. cancer vs. normal) [Jelizarow et al., 2010]. Bayesian networks, ordinary differential equations, and clustering are used to infer the regulatory relationship [Friedman et al., 2000, Greenfield et al., 2010a, Yip et al., 2010, Reiss et al., 2006, Bar-Joseph et al., 2003, Segal et al., 2001, 2003].

Clustering [Duda et al., 2000, Bishop, 2006, Hastie et al., 2009], introduced in Sec. 2.3, partitions a set of observations into mutually exclusive subsets. In microarray data analysis, genes may be clustered into biological meaningful groups according to their pattern of expression across all experimental conditions. Genes within a gene cluster have similar expression patterns and are said to be coexpressed (Fig. 3.7). Rather than clustering genes, experimental conditions may also be clustered into groups where genes express similarly across conditions within each group.

Yeung et al. [2001a] demonstrated the potential usefulness of the Gaussian mixture model-based approach to gene expression data clustering. They also used a variety of multivariate normality assessment tests suggested by Aitchison [1982] to explore the extent to which different transformations of gene expression data satisfy the Gaussian assumption. They showed in [Yeung et al., 2001b] that, with proper data transformation, the approach performed well on gene expression data. The "correct" number of mixture components of a finite mixture model, which is usually uncertain, can be estimated through model selection methods [Robert, 2007], such as Bayes factor [Kass and Raftery, 1995], Akaike's information criterion (AIC) [Akaike, 1974], and Bayesian information criterion (BIC) [Schwarz, 1978]. An alternative approach to model selection and model comparision is to incorporate a Dirichlet process prior on the mixtures, a.k.a, Dirichlet process mixture model (DPM) [Escobar and West, 1995, Neal, 1998, Rasmussen, 2000, Jain and Neal, 2004], introduced in Sec. 2.4.4. DPM and various inference methods have been applied to cluster genes and experimental conditions, including those using Gibbs sampling [Medvedovic and Sivaganesan, 2002, Qin, 2006], variational inference [Teschendorff et al., 2005], and Bayesian hierarchical clustering [Savage et al., 2009]. These studies demonstrated the robustness of DPM at estimating the number of clusters compared to finite mixture model with model selection techniques.

An alternative type of clustering, called biclustering, has gained tremendous popularity in bioinformatics. Unlike standard clustering, biclustering algorithms attempt to uncover blocks (submatrices of a microarray data matrix) with interesting patterns. Based on the problem, the definition

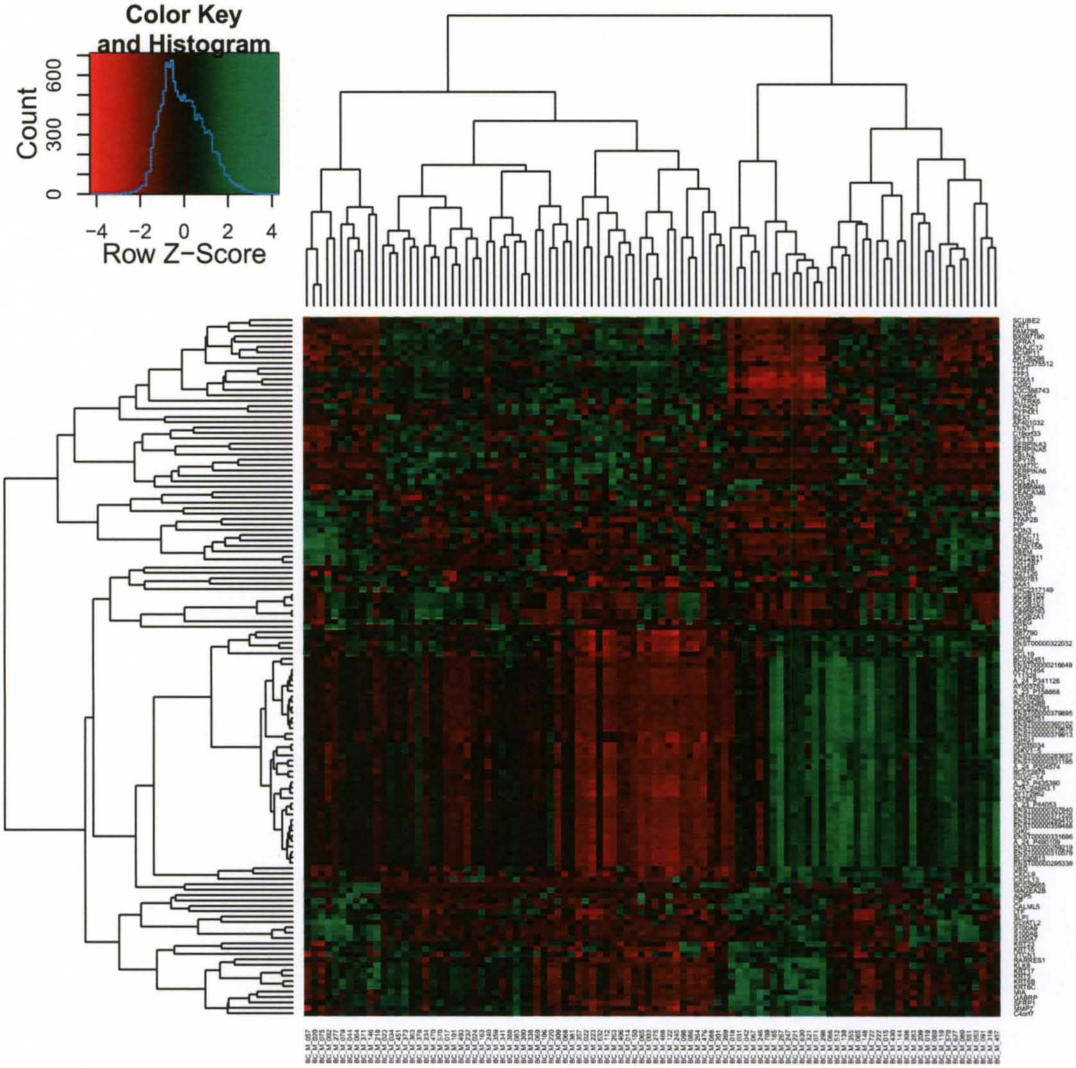


Figure 3.7: Heatmap illustrations of agglomerative hierarhical clustering on a subset of microarray gene expression data, in which each row denotes a gene, each column denotes a sample, and each entry denotes a log based and normalized expression value for the corresponding row and column. The dendrograms show the agglomerative clustering result performned independently on rows and on columns. Rows and columns are ordered according to the dendrograms. Sec. 5.1.2 provides more detailed description of the data.

of "interesting pattern" may vary enormously. In one instance, it may denote a submatrix having constant or similar values on rows and/or columns. In another, it may mean a submatrix having coherent patterns, such as correlation, across rows or columns. Algorithms also differ in their definition of "structure" based on which interesting blocks are organized. Some only allowed mutually exclusive blocks, such as a microarray data matrix divided into checkerboard blocks. Others allowed arbitrarily positioned and overlapping blocks. This section revies two representive biclustering algorithms. For a more complete review on the topic, readers are referred to Madeira and Oliveira [2004].

Biclustering by Cheng and Church

Cheng and Church [2000] defined the concept of a bicluster as a subset of genes and a subset of conditions with a high similarity score, which measures the coherence of the genes and conditions in the bicluster. They developed mean squared residue, a similarity score suitable to expression data transformed by a logarithm and augmented by the additive inverse. Let X be the set of genes and Y be the set of conditions. Let a_{ij} be the element of the expression matrix A representing the logarithm of the relative abundance of the mRNA of the ith gene under the jth condition. Let $I \subset X$ and $J \subset Y$ be subsets of genes and conditions. The pair (I, J) specifies a submatrix A_{IJ} with mean squared residue score defined by

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2, \tag{3.1}$$

where

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}, a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}, \tag{3.2}$$

and

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij} = \frac{1}{|I|} \sum_{i \in I} a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{Ij} \tag{3.3}$$

are the row and column means and the mean in the submatrix A_{IJ}. A_{IJ} is called a *δ-bicluster* if $H(I, J) \leq \delta$ for some $\delta \leq 0$. The goal is to find one or more potentially overlapping δ-biclusters with a large number of rows and columns. They showed the NP-hardness of the problem and developed greedy algorithms to find interesting biclusters. To find a bicluster, at the *node deletion* step, it recursively removes either one (or multiple) rows or columns to receive a better mean squared residual score (Eq. 3.1) until a certain criteria is reached; after node deletion, the resulting δ-bicluster may not be maximal, in the sense that some rows and columns may be added without increasing the score. This is the *node addition* step. In order to find multiple biclusters, repeated runs of the two steps will not be satisfactory, as the deterministic characteristic of the algorithm. Before each repetition, at the *node mask* step, the elements in the submatrix were replaced by random numbers. In the following the computational complexity is discussed. For details of the algorithm, please refer to the original paper by Cheng and Church [2000].

The Plaid Model

Lazzeroni and Owen [2000] introduced plaid models for decomposing gene expression data. The generative models assumed that each expression value is a sum over multiple components, called layers, each of which may represent the presence of a particular set of biological processes. Let K be the number of layers. Let $\rho_{ik} = 1$ if the ith gene is affected by the kth layers and $\rho_{ik} = 0$ otherwise. Let $\kappa_{jk} = 1$ if the jth condition is affected by the kth layers and $\kappa_{jk} = 0$ otherwise. Let θ_{ijk} be a quantity afforded by the kth layer to the ith gene and jth condition. Let a_{ij} be the element of the expression matrix A representing the ith gene under the jth condition. The plaid models are expressed as

$$\begin{aligned} a_{ij} &= \theta_{ij0} + \sum_{k=1}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk} && (3.4) \\ &= \sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk} && (3.5) \end{aligned}$$

where the 0th layer ($k = 0$) is the base layer affecting on all genes and conditions ($\rho_{ik} = 1, \kappa_{jk} = 1$). The situation of the biclusters being mutually exclusive or overlapping is expressed by ρ_{ik} and κ_{jk}. $\sum_{k=1}^{K} \rho_{ik} = 1$ indicates that the ith gene to exactly one bicluster. $\sum_{k=1}^{K} \rho_{ik} \leq 2$ indicates that the ith gene belongs to more than one biclusters. $\sum_{k=1}^{K} \rho_{ik} \leq 0$ indicates that the ith gene belongs to none of the interesting biclusters.

In a plaid model, the concept of coherent pattern in a bicluster is expressed by θ_{ijk}. To define coherence as having constant elements throughout a bicluster, the plaid models set $\theta_{ijk} = \mu_k$ for all i, j of the bicluster. For a bicluster k with similar, though not constant, expression values across all experimental conditions (or across all genes, or across all genes and conditions) in that bicluster,

θ_{ijk} are constricted by Eq. 3.6, Eq. 3.7, and Eq. 3.8 respectively.

$$\theta_{ijk} = \mu_k + \alpha_{ik}, \tag{3.6}$$

$$\theta_{ijk} = \mu_k + \beta_{ik}, \tag{3.7}$$

$$\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}. \tag{3.8}$$

A biclustering problem can be considered as inferring a plaid model with a small value of

$$\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(a_{ij} - \sum_{k=0}^{K}\theta_{ijk}\rho_{ik}\kappa_{jk}\right)^2, \tag{3.9}$$

for an expression A with M genes and N samples. The problem is NP-hard, and the authors proposed a greedy strategy which, if given $K-1$ layers, seek the Kth layer to minimize the sum of squared errors:

$$Q = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}(Z_{ij} - \theta_{ijK}\rho_{iK}\kappa_{jK})^2 \tag{3.10}$$

where

$$Z_{ij} = a_{ij} - \sum_{k=0}^{K-1}\theta_{ijk}\rho_{ik}\kappa_{jk} \tag{3.11}$$

is the residual from the first $K-1$ layers. They developed an iterative algorithm with each cycle updating θ, ρ and κ respectively. The algorithm is briefly shown in algorithm 3. For details of the estimation procedure, please refer to the original paper [Lazzeroni and Owen, 2000].

3.3 Molecular Evolution

Molecular evolution focuses mainly on two areas of study: the characterization of changes (including to what pattern and rates such changes occur) in nucleic acids and proteins during evolutionary time, as well as the reconstruction of evolutionary history of organisms from molecular data. The two disciplines are closely related, and progress in one area facilitates progress in the other: knowledge for the pattern and rate of evolution is essential in reconstructing phylogenetic trees statistically; and in the meantime, phylogenetic knowledge is essential in determining the arrangement of DNA sequences of organisms and guiding the estimation of evolution patterns and rates.

Algorithm 3: Estimation for Plaid Model

```
foreach $k = 1, \ldots, K$ do
    foreach $s = 1, \ldots, S$ do
        Initialization. ;
        Update $\theta_{ijk}^{(s)} : \mu_k^{(s)}, \alpha_{ik}^{(s)}, \beta_{jk}^{(s)}$. ;
        Update $\rho_{ik}^{(s)}, \kappa_{jk}^{(s)}$. ;
    end
    Update $Z_{ijk}$. ;
    Estimate importance of layer $k$: $\sigma_k^2 = \sum_{i=1}^{n} \sum_{j=1}^{p} \rho_{ik}\kappa_{jk}\theta_{ijk}^2$. ;
    Convergence test based on permutation test. If convergence is reached, jump out of the
    loop. ;
end
foreach $k = 1, \ldots, K$ do
    backfitting: $\theta_{ijk}$;
end
```

3.3.1 Models of Nucleotide Substitution

One basic process in the evolution of DNA sequences is the substitution of one nucleotide for another during evolutionary time. This process are complex and often cannot be directly observed due to the slow changes. Therefore, in order to study the dynamics of nucleotide substitution, one must make simplified and abstracted assumptions regarding to the characteristics of the process. It is a standard practice in model-based phylogenetic methods to assume that character substitution occurs according to a continuous-time Markov chain [Huelsenbeck et al., 2004]. In a Markov chain, the rate of change between states is represented by a transition matrix. Phylogenetic analyses using DNA sequence data assume four states (the nucleotides A, C, G, T/U) and thus the rate matrix is a 4×4 matrix representing the 12 possible nucleotide substitutions. Some well recognized nucleotide substitution models include Jukes-Cantor (JC69) [Jukes and Cantor, 1969], two parameter models (K80) [Kimura, 1980], F81 [Felsenstein, 1981], HKY85 [Hasegawa et al., 1985], TN93 [Tamura and Nei, 1993], and GTR [Tavaré, 1986]. This section reviews JC69 and K80.

Discrete-Time Jukes-Cantor Model

The discrete-time Jukes-Cantor model is discussed by Ewens and Grant [2004] and is a simpler and discrete-time version of the Jukes-Cantor model. It is a Markov chain with four states $A, G, C,$ and

T. The transition matrix P for this model, in the order $AGCT$, is

$$P = \begin{pmatrix} 1-3\alpha & \alpha & \alpha & \alpha \\ \alpha & 1-3\alpha & \alpha & \alpha \\ \alpha & \alpha & 1-3\alpha & \alpha \\ \alpha & \alpha & \alpha & 1-3\alpha \end{pmatrix}, \tag{3.12}$$

where the parameter α depends on the timescale chosen. The spectral expansion of P^n is

$$P^n = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} + (1-4\alpha)^n \begin{pmatrix} 3/4 & -1/4 & -1/4 & -1/4 \\ -1/4 & 3/4 & -1/4 & -1/4 \\ -1/4 & -1/4 & 3/4 & -1/4 \\ -1/4 & -1/4 & -1/4 & 3/4 \end{pmatrix}. \tag{3.13}$$

Thus, the probability that a site has the same nucleotide at time n as the one at time 0 is

$$\frac{1}{4} + \frac{3}{4}(1-4\alpha)^n, \tag{3.14}$$

and the probability that a site has a different nucleotide at time n to the one at time 0 is

$$\frac{1}{4} - \frac{1}{4}(1-4\alpha)^n \tag{3.15}$$

The stationary distribution of the model is

$$\pi = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}). \tag{3.16}$$

Continuous-Time Jukes-Cantor Models

Let i, j,and k be arbitary states from $\{A, G, C, T\}$. The instantaneous transition rates q_{ij} are defined as

$$q_{ij} = \alpha \text{ for all } i \neq j \tag{3.17}$$

Thus, $q_i = \sum_{i \neq j} q_{ij} = 3\alpha$. Following the Markov assumption and time-homogeneity assumption, the probability of the state at time $t + \varepsilon$ being j given the state at time t being i is

$$P_{ij}(t+\varepsilon) = \sum_k P_{ik}(t) P_{kj}(\varepsilon), \tag{3.18}$$

where the sum is taken over all possible states in the set $\{A, G, C, T\}$. To solve the equation, a system of differential equations, called *forward Kolmogrorov equations* of the system, can be derived:

$$\frac{d}{dt}P_{ij}(t) = -3\alpha P_{ij}(t) + \alpha \sum_{k \neq j} P_{ik}(t) \tag{3.19}$$

$$= -3\alpha P_{ij}(t) + \alpha \sum_{k \neq j} (1 - P_{ij}(t)) \tag{3.20}$$

$$= \alpha - 4\alpha P_{ij}(t). \tag{3.21}$$

With boundary conditions $P_{ii}(0) = 1$, $P_{ij} = 0$ for $i \neq j$, the solution of the linear differential equation is

$$\begin{aligned} P_{ii}(t) &= \frac{1}{4} + \frac{3}{4}e^{-4\alpha t}, \\ P_{ij}(t) &= \frac{1}{4} - \frac{1}{4}e^{-4\alpha t}, \text{for} j \neq i. \end{aligned} \tag{3.22}$$

Discrete-Time Kimura Models

The assumption that all nucleotide substitutions occur with equal probability, as in the one parameter model of JC69, is unrealistic in most cases. For example, a *transition*, that is, the replacement of one purine or pyrimidine by another (i.e., replacement between A and G or between C and T) is generally more frequent than a *transversion*, that is, the replacement of one purine by a pyrimidine or of one pyrimidine by a purine. To take this fact into account, Kimura [1980] proposed a two-parameter model (K80). The transition matrix for the discrete-time version of K80, in the order $AGCT$, is

$$P = \begin{pmatrix} 1-\alpha-2\beta & \alpha & \beta & \beta \\ \alpha & 1-\alpha-2\beta & \beta & \beta \\ \beta & \beta & 1-\alpha-2\beta & \alpha \\ \beta & \beta & \alpha & 1-\alpha-2\beta \end{pmatrix}, \tag{3.23}$$

where α is the probability of a transition in one time unit and β is the probability of a transversion in one unit. The spectral expansion of P^n is

$$P^n = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} + (1-4\beta)^n \begin{pmatrix} 1/4 & 1/4 & -1/4 & -1/4 \\ 1/4 & 1/4 & -1/4 & -1/4 \\ -1/4 & -1/4 & 1/4 & 1/4 \\ -1/4 & -1/4 & 1/4 & 1/4 \end{pmatrix}$$
$$+ (1-2(\alpha+\beta))^n \begin{pmatrix} 1/2 & -1/2 & 0 & 0 \\ -1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & -1/2 \\ 0 & 0 & -1/2 & 1/2 \end{pmatrix}. \tag{3.24}$$

Continuous-Time Kimura Models

Let i, j,and k be arbitary states from $\{A, G, C, T\}$. The instantaneous transition rates q_{ij} for K80 is

$$q = \begin{pmatrix} 1-\alpha-2\beta & \alpha & \beta & \beta \\ \alpha & 1-\alpha-2\beta & \beta & \beta \\ \beta & \beta & 1-\alpha-2\beta & \alpha \\ \beta & \beta & \alpha & 1-\alpha-2\beta \end{pmatrix}, \tag{3.25}$$

and

$$q_i = \sum_{i \neq j} q_{ij} = \alpha + 2\beta. \tag{3.26}$$

Recall that the forward Kolmogorov equations of the continuous-time Markov model take the form

$$\frac{d}{dt} P_{ij}(t) = -q_j P_{ij}(t) + \sum_{k \neq j} P_{ik}(t) q_{kj}, \text{ for } j = 1, 2, \cdots, s. \tag{3.27}$$

With boundary conditions $P_{ii}(0) = 1$ and $P_{ij}(0) = 0$ for $j \neq i$, the solutions of the differential equations are

$$P_{ii}(t) = \frac{1}{4} + \frac{1}{4} e^{-4\beta t} + \frac{1}{2} e^{-2(\alpha+\beta)t}, \tag{3.28}$$

and for $(i,j) \in \{(A,G),(G,A),(C,T),(T,C)\}$

$$P_{ij}(t) = \frac{1}{4} + \frac{1}{4}e^{-4\beta t} - \frac{1}{2}e^{-2(\alpha+\beta)t}, \tag{3.29}$$

and for $(i,j) \in \{(A,C),(A,T),(G,C),(G,T),(C,A),(C,G),(T,A),(T,G)\}$

$$P_{ij}(t) = \frac{1}{4} - \frac{1}{4}e^{-4\beta t}. \tag{3.30}$$

3.3.2 Molecular Phylogenetics

The objectives of phylogenetics are to reconstruct the genealogical relationships among biological species, to estimate the time of divergence between species, and to chronicle events along evolutionary lineages. Molecular phylogenetics study evolutionary relationships through molecular data such as nucleotide sequences and amino acid sequences, as opposed to traditional information such as anatomical, morphological, and palaeontological data. Comparing to these traditional data, there are several advantages of DNA and proteins in phylogenetic studies: they are heritable, unambiguous, and abundant.

The reconstruction requires proper assumptions on the process of DNA evolution, and the structure of the evolutionary relationships among taxonomic units. Sec. 3.3.1 introduced nucleotide substitution, a basic process in the evolution of DNA. In phylogenetics, the evolutionary relationships among organisms are presented as a **phylogenetic tree** (3.8). It is worthing mentioning that alternative structures such as **phylogenetic networks** have also been studied, e.g. [Huson et al., 2011], though we shall focus on tree structure in this study. The nodes of the tree represent taxonomic units (e.g. species, individuals, or genes), and the branches define the ancestry-descent relationships among the taxonomic units and the number of changes occurred during the event. **Operational taxonomic units (OTUs)** refers to the leaf nodes of the tree and represent the extant taxonomic units under comparison. Internal nodes represent inferred ancestral units, and are sometimes referred to as **hypothetical taxonomic units (HTUs)**. A **bifurcating** node is defined as one which has exactly two immediate descendant lineages, while a **multifurcating** node can have more than two. In practical reconstruction process, it is often assumed that speciation is a bifurcating process. A **rooted tree** is a tree with a particular internal node (the root) representing the most recent common ancestor of all the taxonomic units under study. An **unrooted tree** is a tree without specifying the root, though strictly speaking, an unrooted tree may not be considered a phylogenetic tree.

Edwards and Cavalli-Sforza [1964] found that the number of possible bifurcating unrooted trees with n leaves is

$$N_R = \frac{(2n-5)!}{2^{n-3}(n-3)!}, \text{ for } n \geq 3, \tag{3.31}$$

and Felsenstein [1978] showed that the number of possible bifurcating rooted trees with n leaves is

$$N_R = \frac{(2n-3)!}{2^{n-2}(n-2)!}, \text{ for } n \geq 2. \tag{3.32}$$

Table 3.3.2 lists N_R for $n = 1, 2, \cdots, 20$.

Table 3.1: The number of possible rooted evolutionary trees for a given number of OTUs

n	N_R
2	1
3	3
4	15
5	105
6	954
7	10,395
8	135,135
9	2,027,025
10	34,459,425
15	213,458,046,676,875
20	8,200,794,532,637,891,559,375

We introduce in the following the basic phylogenetic tree inference methods. For a detailed treatment of these methods, readers may consult Felsenstein [2003]. The simplest method for tree construction is called **UPGMA**, which stands for unweighted pair group method using arithmetic averages [Sokal and Michener, 1958]. It is used when the rates of evolution are approximately constant among the different lineages. UPGMA is a greedy algorithm essentially the same as agglomerative hierarchical clustering Duda et al. [2000]. It starts with each OTU in its own cluster and repeatedly merge clusters greedily until there is only one cluster left. There are many different ways of defining distance, which quantifies the evolutionary difference between OTUs. One can take the distance to be the fraction of mismatches between two sequences, where a **mismatched** base pair is defined as pair in which different nucleotides are found in the two **aligned** sequences. This gives a sensible definition for small fractions. But for two unrelated sequences, random substitutions will cause the fraction to approach a value expected by chance. Markov models of nucleotide substitution, discussed in Sec. 3.3.1, can be used to define evolutionary distances with large value as the fraction approaches the expected value. In UPGMA, the distance between two clusters is defined as the average distance between all pairs of sequences from each cluster. Besides average distance, the minimum or maximum of all the distances can also be used to define distance between two clusters.

An alternative strategy to the greedy approach in UPGMA is **maximum parsimony**, which involves the identification of a topology that can explain the observed sequences (or the observed differences among the sequences) with a minimal number of substitutions. A maximum parsimony algorithm thus consists of two components: an objective function (or optimality criterion) which is a value assigned to a phylogeny, and a searching algorithm which explore "all" trees and pick the one with the best value according to the optimality criterion. The objective function of parsimony often treats each base independently, and then adds the substitutions for all bases. **Unweighted parsimony** considers all the different nucleotide substitutions are given equal weight during summation, and **weighted parsimony** refers to the case where the various character state changes are assigned different weights. An exhaustive search, i.e. enumerate all possible topologies and compare them, is computationally infeasible because the searching space increase rapidly with the number of OTUs. Heuristic search methods that can balance the computation resources effectively between exploration and exploitation of the searching space is essential to a maximum parsimony algorithm. **Branch swapping** is also an important technique which is used to generate topologically similar trees from an initial one.

The **neighbor-joining** method [Saitou and Nei, 1987] is a greedy approximate algorithm with the same objective as the maximum parsimony algorithm, i.e. to find the minimum evolution tree. The algorithm starts with a star-like tree where all the OTUs are connected to a central node. At each stage of clustering, it finds and merges the pairs of neighbors that minimize the **total branch length**. The method defines a pair of **neighbors** as a pair of taxonomic units (can be OTUs or HTUs) connected through a single interior node in an unrooted, bifurcating tree. The combined pair of taxonomic units is regarded as a single taxonomic units (HTU) and can serve as neighbor to other taxonomic units as well. The merging procedure is continued until all interior branches are found.

The Likelihood of a Phylogenetic Tree

From a statistical point view, a phylogenetic tree and its substitution pattern describe a **generative model** about how the observations (i.e. DNA sequences) were evolved from their common ancestors. Given contemporary sequences derived from contemporary organisms, the model parameters, i.e. the rates of substitution, the number of changes, and the topology can be inferred using a variety of parameter estimation methods, such as maximum likelihood estimation (MLE) and Bayesian inference.

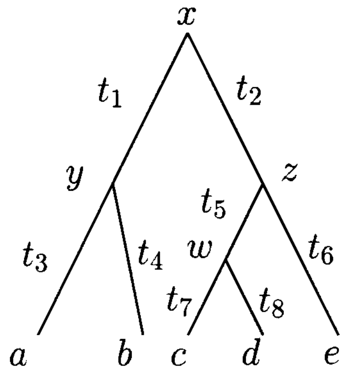


Figure 3.8: A phylogenetic tree for five species.

Assume the evolutionary tree of five species shown in Fig. 3.8. Let τ be the tree topology, $\beta = \{t_1, t_2, \cdots, t_8\}$ be the branch lengths measuring the expected number of substitutions, Θ be whatever substitution model chosen (i.e. JC69, K80). Let $\mathsf{F} = \{a, b, c, d, e\}$ at the leaves be the observed nucleotides at a particular site. Let $\mathsf{I} = \{x, y, z, w\}$ at non-leaf nodes be the unobserved ancestral nucleotides at that site. The likelihood of τ, β, Θ given the observation F, is

$$P(\mathsf{F}|\tau, \beta, \Theta) = \sum_{\mathsf{I}} \pi_x P_{xy}(t_1) P_{xz}(t_2) P_{ya}(t_3) P_{yb}(t_4) P_{zw}(t_5) P_{ze}(t_6) P_{wc}(t_7) P_{wd}(t_8) \tag{3.33}$$

$$= \sum_x \sum_y \sum_z \sum_w \pi_x P_{xy}(t_1) P_{xz}(t_2) P_{ya}(t_3) P_{yb}(t_4) P_{zw}(t_5) P_{ze}(t_6) P_{wc}(t_7) P_{wd}(t_8) \tag{3.34}$$

$$= \sum_x \pi_x \Big\{ \sum_y P_{xy}(t_1) P_{ya}(t_3) P_{yb}(t_4) \Big\} \Big\{ \sum_z P_{xz}(t_2) P_{ze}(t_6) \sum_w P_{zw}(t_5) P_{wc}(t_7) P_{wd}(t_8) \Big\} \tag{3.35}$$

The likelihood sums over all possible assignments of nucleotide state to the ancestral nodes. For a phylogenetic tree with N leaf nodes, the number of assignments is 4^{N-1} for nucleotide sequence and 20^{N-1} for amino acid. Felsenstein [1981] developed a *pruning algorithm* which reformed the likelihood using Horner's rule (i.e. reform Eq. 3.33 into Eq. 3.35) and calculated the likelihood in a bottom up way using a dynamic programming strategy.

Let Y be an $N \times S$ matrix representing a set of aligned sequences over N loci and S sites. Let Y_s be the sth column of Y. Let τ, β, Θ be the topology, branch lengths and substitution parameters respectively. If it is assumed that the sites evolve independently, then the likelihood of the model parameters given the observation Y is

$$\mathcal{L}(\tau, \beta, \Theta \; ; \; \mathsf{Y}) \overset{def.}{=} P(\mathsf{Y}|\tau, \beta, \Theta) = \prod_{s=1}^{S} P(\mathsf{Y}_s|\tau, \beta, \Theta) \tag{3.36}$$

Maximum Likelihood Estimation

Let $\Phi = (\tau, \beta, \Theta)$, The *maximum likelihood* (ML) estimator, denoted by $\hat{\Phi}_{\mathrm{MLE}}$, is

$$\hat{\Phi}_{\mathrm{ML}} = \arg\max_{\tau,\beta,\Theta} \mathcal{L}(\tau, \beta, \Theta; \mathsf{Y}) \tag{3.37}$$

$$= \arg\max_{\tau,\beta,\Theta} \sum_{s=1}^{S} P(\mathsf{Y}_s|\tau, \beta, \Theta) \tag{3.38}$$

The parameters to be estimated include the topology τ , branch lengths β and substitution parameters Θ. Given the values of these parameters, the likelihood can be calculated using the pruning algorithm. The ML estimation normally consists of two tasks. The first task is to, given a fixed topology τ, estimate the branch lengths and substitution parameters. The likelihood function

$$\mathcal{L}(\beta, \Theta; \mathsf{Y}, \tau) \overset{def.}{=} P(\mathsf{Y}|\tau, \beta, \Theta) \tag{3.39}$$

has the same joint density as of the likelihood function $\mathcal{L}(\tau, \beta, \Theta; \mathsf{Y})$, except that τ is treated as fixed and known value. The estimation is then

$$\hat{\beta}_{\mathrm{ML}}, \hat{\Theta}_{\mathrm{ML}} = \arg\max_{\beta,\Theta} \sum_{s=1}^{S} P(\mathsf{Y}_s|\tau, \beta, \Theta) \tag{3.40}$$

This estimation is normally done through iterative methods such as expectation maximization [Felsenstein, 1981, Friedman et al., 2002, Dempster et al., 1977] or numerical optimization [Olsen et al., 1994, Yang, 2000, Nocedal and Wright, 2000]. An optimization procedure often cycles through two steps. In the first step, branch lengths are optimized one by one through Newton's algorithm while keeping substitution parameters fixed. In the second step, substitution parameters are optimized through numerical optimization methods such as BFGs while keeping branch lengths fixed.

The second task is to estimate the topology, as in Eq. 3.37. For a given topology τ, the optimized likelihood ($\mathcal{L}(\hat{\beta}_{ML}, \hat{\Theta}_{ML}; \mathsf{Y}, \tau)$) can be estimated using numerical methods. Therefore, a brute force approach would enumerate all possible topologies, estimate the maximum likelihood with respect to branch lengths and substitution parameters for each topology, and choose that topology which yielded the highest likelihood ($\hat{\tau}_{\text{ML}}$). This exhaustive search strategy is computationally infeasible because of the large topology space. To effectively explore the topology space and estimate the maximum likelihood, algorithms based on heuristic paradigms have been proposed. Felsenstein [1981] developed a search algorithm, which builds the tree by successively adding species to it, beginning with a two-species tree. When the kth species is being added to the tree, there will be $2k - 5$ branches from which it could arise. Each of these topologies is tried and their maximum likelihood with respect to branch lengths and substitution parameters are estimated. The placement yielding the highest likelihood is accepted. If $k \leq 5$, before trying adding the species to the tree, local rearrangements of taxa in the tree are conducted to explore alternative topologies. If any of these topologies improves the likelihood, it is accepted and the hill climbing process continues until no local rearrangement can improve the likelihood.

Guindon and Gascuel [2003] developed an efficient estimation algorithm which starts with an initial tree constructed by a distance-based algorithm such as BIONJ [Gascuel, 1997] and iteratively refines the tree through simultaneous branch length optimization and subtree swapping.

Lewis [1998], Lemmon and Milinkovitch [2002], independently, developed genetic algorithms for the maximum likelihood estimation. The algorithm in [Lewis, 1998] starts with a population of n individuals, each of which is a phylogenetic tree with arbitrarily specified topology, branch lengths and substitution parameters. The algorithm then explores the searching space through generations, which follow a simulated natural selection process. At each generation, individuals with higher likelihood will get higher chances to survive and reproduce; meanwhile, individuals are subjected to branch length mutation, topology mutation and substitution model mutation; at the end of the generation, recombinations between individuals are carried out.

The standard maximum likelihood formulation for phylogenetic trees (as in Eq. 3.37) seeks parameters yielding the highest likelihood. An alternative maximum likelihood formulation for estimating tree topology, which we introduce here, is to sum over all possible branch and substitution parameters:

$$\hat{\tau}_{\text{ML}} = \arg\max_{\tau} \sum_{\beta,\Theta} \mathcal{L}(\tau, \beta, \Theta \; ; \; \mathsf{Y}) \tag{3.41}$$

Bayesian Inference

According to Bayes' theorem, the posterior probability of τ, β, Θ given the aligned sequences Y is

$$P(\tau, \beta, \Theta|\mathsf{Y}) = \frac{P(\tau, \beta, \Theta)\mathcal{L}(\tau, \beta, \Theta; \mathsf{Y})}{\int P(\tau, \beta, \Theta)\mathcal{L}(\tau, \beta, \Theta; \mathsf{Y}) d\ \tau\ \beta\ \Theta} \tag{3.42}$$

Here, for ease of representation, symbols $\sum$ and $\int$ are used interchangeably.

$$P(\tau|\mathsf{Y}) = \int P(\tau, \beta, \Theta|\mathsf{Y}) d\ \beta\ \Theta \tag{3.43}$$

The study of Bayesian approaches to molecular phylogenetics using MCMC begins with several independent works in [Yang and Rannala, 1997, Mau et al., 1999]. Rannala and Yang [1996] developed an algorithm based on empirical Bayesian analysis, where the parameters of the substitution model and the prior distribution of phylogenetic trees were estimated using maximum likelihood, and these estimates were then used to replace the true parameters to evaluate the posterior probabilities of trees. The calculations involve a sum over all topologies and, for each topology, an integral over branch lengths using numerical integration. Recognizing that the computational procedure is suitable only for very small trees, they incorporated Monte Carlo integration to evaluate the integral over branch lengths and avoided the sum over all topologies by sampling the posterior distribution using MCMC [Yang and Rannala, 1997]. According to the Metropolis-Hastings algorithm, the probability of accepting a proposed new state Φ^* from the current state Φ is

$$r = \min\left(1, \frac{P(\Phi^*|\mathsf{Y})q(\Phi^*, \Phi)}{P(\Phi|\mathsf{Y})q(\Phi, \Phi^*)}\right) \tag{3.44}$$

the sampler uses a Metropolis-within-Gibbs [Tierney, 1994] algorithm that cycles through blocks of model parameters within Φ. For example, the probability of accepting the proposed new substitution parameters Θ^* from the current state Θ is

$$r_\Theta = \min\left(1, \frac{P(\Theta^*|\mathsf{Y}, \tau, \beta)q_\Theta(\Theta^*, \Theta)}{P(\Theta|\mathsf{Y}, \tau, \beta)q_\Theta(\Theta, \Theta^*)}\right) \tag{3.45}$$

Larget and Simon [1999] developed in BAMBE two algorithms for proposing new trees. The first one, a GLOBAL algorithm, modifies all branch lengths and potentially changes the tree topology simultaneously; The second, a LOCAL algorithm, modifies the tree only in a small neighborhood of a randomly chosen internal branch, leaving the rest of the tree unchanged. A version relaxing the assumption on a molecular clock was also proposed in the same paper.

To reduce the chance that Markov chain simulations remain in the neighborhood of a single mode for a long period of time, Huelsenbeck and Ronquist [2001] implemented in MRBAYES a variant of Metropolis-coupled MCMC ($(MC)^3$) [Geyer, 1991], also known as simulated tempering [Neal, 1996]. $(MC)^3$ runs m Markov chains in parallel, having different, but related, stationary distributions, $f_1, \cdots, f_m$. After all m chains have gone one step, a swap is attempted between two randomly chosen chains i and j according to a Metropolis update with the odds ratio defined as

$$r = \frac{f_i(\tau_j)f_j(\tau_i)}{f_i(\tau_i)f_j(\tau_j)} \tag{3.46}$$

Here, τ_i and τ_j are the tree topologies drawn from chain i and j respectively. The stationary distribution $f_i(\tau)$ is defined as

$$f_i(\tau) = P(\tau|\mathsf{Y})^{\beta_i} \tag{3.47}$$

MRBAYES uses incremental heating which defines β_i for the ith chain as

$$\beta_i = \frac{1}{1 + (i-1)T} \tag{3.48}$$

where T is a user-set temperature. Samples for the target posterior distribution are drawn only from the first chain (cold chain, $i = 1, \beta_i = 1$). The heated chains ($\beta_i < 1$) can more effectively explore the searching space because heating makes the peaks (modes or local optima) lower and valleys higher.

Rate Heterogeneity Across Sites

It is realistic to assume that substitution parameters vary across different sites [Tateno et al., 1994, Yang, 1996]. A nuanced model would allow using one set of substitution parameters for each site. That is, for an aligned sequence data Y with S sites, the substitution parameters takes the form

$$\Theta = (\Theta_1, \Theta_2, \cdots, \Theta_S),$$

where Θ_s is the substitution parameters specific to site s ($s = 1, 2, \cdots, S$). This, however, results in too many parameters to estimate given a limited number of observations. A more practical approach is to model the rate variation using a probabilistic distribution.

In the gamma-rates model [Gu et al., 1995, Kelly and Rice, 1996], it is assumed that the substitution parameters Θ take the form

$$\Theta = (r_1 Q, r_2 Q, \cdots, r_S Q).$$

Here all the sites share a common set of substitution parameters Q but vary in rates by a proportion (r_s). $r_1, r_2, \cdots, r_S$ are independent and identically distributed (i.i.d.) random variables from a gamma distribution (denoted $g(r|\alpha_1, \alpha_2)$ with α_1, α_2 being prior parameters. The likelihood of model parameters given information on site s is

$$P(\mathsf{Y}_s|\tau, \beta, Q, \alpha_1, \alpha_2) = \int P(\mathsf{Y}_s|\tau, \beta, r, Q) g(r|\alpha_1, \alpha_2) dr \tag{3.49}$$

This marginal likelihood does not have an analytical form and thus numerical methods or Monte Carlo methods are necessary. Mayrose et al. [2005] used numerical integration which approximate the integrand with simpler functions and a reduced finite set of r. Nevertheless, the computational intensive integrand and the large domain of integration prohibit the marginal likelihood from practical usage. A more practical and widely accepted method is to use finite mixture modeling techniques, where sites are partitioned and all sites within each partition share a common rate. For example, under discrete-gamma model[Yang, 1996, Mayrose et al., 2005] with C equal probable rate classes, The likelihood of model parameters given Y_s is

$$P(\mathsf{Y}_s|\tau, \beta, Q, \alpha_1, \alpha_2) = \sum_{c=1}^{C} P(\mathsf{Y}_s|\tau, \beta, r_c, Q) g(r_c|\alpha_1, \alpha_2). \tag{3.50}$$

CHAPTER 4

INFINITE RELATIONAL MODEL WITH FEATURE SELECTION

4.1 Introduction

Infinite relational models (IRMs) [Kemp and Tenenbaum, 2006, Xu et al., 2006, Friedman et al., 1999, Airoldi et al., 2008] are generalizations of Dirichlet process mixture (DPM) models [Escobar and West, 1995] (also known as infinite mixture models [Rasmussen, 2000]) to the relational domain, where the observations include not only the object-feature data representing entity properties, but also one or more relations involving one or more types, representing object-object relationships. The goal of the IRMs is to partition each type into clusters, where a good set of partitions allows entity features and relationships between entities to be predicted by their cluster assignments. IRM associates each entity with a latent variable representing cluster assignment, and defines a generative model for the observations and latent variables. In the Bayesian hierarchical model, the relations are conditionally independent given the cluster assignments, and the prior assigns some probability mass to all possible partitionings of the type, often through Bayesian nonparametric techniques [Ferguson, 1973] and the Chinese restaurant process (CRP) [Aldous, 1985, Pitman, 2006].

A fundamental assumption of an IRM approach to clustering is that all features are interesting in describing the underlying structure. There are many cases in which, however, the structure of greatest interest may be best represented using only a selected subset of features. This may result in structures with more intuitive interpretation, or better prediction accuracy. In general, removing unnecessary features (and variables) may also improve the precision of parameter estimation.

In this paper, we introduce joint feature selection and infinite relational model (FIRM) which allows for more robust collective inference and often results in significant performance gains. Traditional IRMs define the same form of mixture density functions over all features, and feature selection, an essential part of clustering, is ignored or must be done prior to the application of the methods. In contrast with IRMs, FIRM incorporates latent variables of features to explicitly represent feature saliency in a relational context and results in structures with more intuitive interpretation and better prediction accuracy. By recasting the feature selection problem as parameter estimation, FIRM is able to efficiently avoid the combinatorial search through the space of all feature subsets.

Although many mixture-based methods for joint feature selection and clustering have been proposed and applied to a wide range of domains [Law et al., 2003, 2004, Tadesse et al., 2005, Constantinopoulos et al., 2006, Chang et al., 2005], these previous methods have focused exclusively on feature (or attribute) data. Feature selection in clustering, however, is inherently a harder problem than in supervised learning, due to the absence of class labels and prior knowledge of the underlying structure which would guide the search for relevant features. Relationships between objects of the same or different domains account for a large proportion of semantic knowledge. The studies in statistical relational learning have demonstrated that learning joint models can often lead to better performance. FIRM differs from these previous feature selection methods in that it incorporates an arbitrary system of relations and entity types in a domain of interest. By conditioning the multiple probability density functions on latent component variables, the model allows for information exchange between features and relations of the same entity type, and also leads to information propagation among different entity types through the entire multi-relational network.

4.2 Background

The general form of a finite mixture model with K mixture components or clusters can be written as:

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}), \tag{4.1}$$

where $\boldsymbol{x}$ is a multivariate variable, π_k is the mixing proportion of the kth component and represents the prior probability of an observation being generated from the kth component, and $f_k(\cdot)$ is the probability density function of the kth component. The components are often modeled by members of the same parametric density family so the finite mixture model can be written as:

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{x}|\boldsymbol{\theta}_k), \tag{4.2}$$

where $\boldsymbol{\theta}_k$ is the parameter vector for the kth component.

The feature saliency is defined as the probability that a feature is relevant to the clustering.

Assuming the features are independent for each mixture component, the mixture density function with feature saliency is then written as:

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \prod_{d=1}^{D} (\rho_d f(x_d|\theta_{kd}) + (1 - \rho_d) g(x_d|\phi_d)), \tag{4.3}$$

where D is the number of features, x_d is the observation of the dth feature, and θ_{kd} is the parameter specific to the kth component and the dth feature, ρ_d is the prior probability of the dth feature being relevant to data clustering, and $g(x_d|\phi_d)$ is the density function of the dth feature when it is irrelevant to the mixtures. The form of $g(\cdot|\cdot)$ can be any uni-variate distribution, e.g. Gaussian or even mixture distributions.

Given a set of observations, the model parameters can be estimated with different type of inference algorithms, including expectation-maximization (EM) for maximizing the likelihood function (MLE) or the posterior probability (MAP), Markov chain Monte Carlo (MCMC) for drawing samples of the posterior probability distributions, or Bayesian variational approximations for maximizing the marginal likelihood function. The choice of the number of mixture components K, a key task in the application of finite mixture models, can be handled with different type of model selection techniques, including a complexity criterion approach using an extended minimum message length (MML) criterion for finite mixtures [Law et al., 2004, Figueiredo and Jain, 2002], a Bayesian sampling approach using reversible jump MCMC embedded within a Gibbs sampler [Tadesse et al., 2005, Richardson and Green, 1997], or a Bayesian variational model selection approach [Constantinopoulos et al., 2006, Corduneanu and Bishop, 2001].

The Dirichlet process mixture model (DPM, also known as infinite mixture model) [Escobar and West, 1995, Rasmussen, 2000] is a Bayesian nonparametric alternative for density estimation using mixtures of Dirichlet processes [Ferguson, 1973]. DPM allows a countably infinite number of component and this number can be automatically and implicitly inferred from data using the usual Bayesian posterior inference framework such as MCMC for DPM [Escobar and West, 1995, Neal, 1998, Jain and Neal, 2004], or tree-based approximating for DPM [Heller and Ghahramani, 2005a]. The feature saliency strategy for feature selection shown in (4.3) can be smoothly incorporated into the framework of DPM, as proposed in [Kim et al., 2006].

4.3 Model

We shall restrict the discussion to the situation where the observed data includes a feature data X and a relation data R, though the model applies to situations of multiple feature datasets and relations. Specifically, suppose the feature data X contains a vector of feature observations for each entity of type T_1, and the relational data R contains a relation observation for each pair of entities of type T_1 and T_2. FIRM aims to model X and R by the following generative process,

$$\begin{aligned}
\boldsymbol{z} &\sim \text{CRP}(\alpha_z), && (4.4)\\
s_d &\sim \text{Bernoulli}(\rho_d), && (4.5)\\
\theta_{k,d} &\sim f_0(\lambda_\theta), && (4.6)\\
\phi_d &\sim g_0(\lambda_\phi), && (4.7)\\
X_{i,d} &\sim f(X_{i,d}|\theta_{z_i,d})^{s_d} g(X_{i,d}|\phi_d)^{1-s_d}, && (4.8)\\
\boldsymbol{u} &\sim \text{CRP}(\alpha_u), && (4.9)\\
\psi_{k,l} &\sim h_0(\lambda_\psi), && (4.10)\\
R_{i,j} &\sim h(R_{i,j}|\psi_{z_i,u_j}). && (4.11)
\end{aligned}$$

where a discrete latent variable, $z_i \in \{1, \cdots, K\}$, is introduced for each entity i of T_1 to encode which component has generated the observation; i.e. $z_i = k$ indicates that the ith observation is generated by the kth component, also introduced is a binary latent variable, $s_d \in \{0, 1\}$ for each feature d to encode whether the feature is relevant to the clustering, and f, f_0, h, h_0, g are functions described in the subsections.

4.3.1 Prior on Partitions

While the Dirichlet process implicitly partitions the observations, the Chinese restaurant process (CRP) [Pitman, 2006, Aldous, 1985] explicitly defines a predictive distribution over partitions and provides a convenient interpretation of the partitions induced by the Dirichlet process. Imagine building a partition over n observations: under the CRP, there is an infinite number of clusters; and the probability that the ith observation is assigned to the kth cluster is proportional to n_k, the number of observations already assigned to the kth cluster; the observation can also be assigned to a new (and empty) cluster with probability proportional to the positive scalar concentration parameter

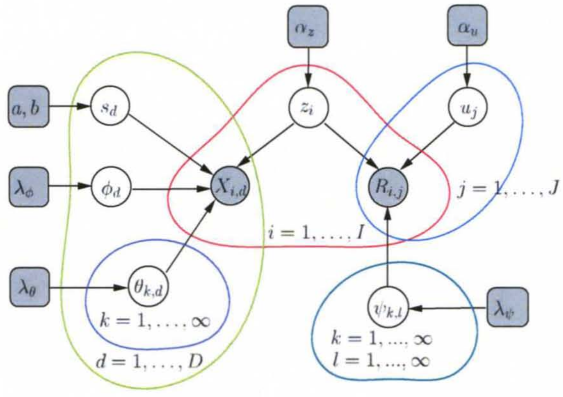


Figure 4.1: The basic graphical model for FIRM with feature data X and relation data R. The observation of each entity i includes a length D vector ($X_{i,\cdot}$) representing the features (or properties) of i, and a length J vector ($R_{i,\cdot}$) representing i's relationships with J entities. The observations are conditionally independent given the cluster assignment. White and grey circle nodes denote latent and observed variables respectively, gray square nodes denote hyperparameters, and arrows indicate probabilistic dependencies. Plates denote repeated variables over the indicated indices. ∞ denotes countably infinite number of components and grows with the number of entities. Specifically, $\boldsymbol{z}$ and $\boldsymbol{u}$ are latent component indicator variables, $\boldsymbol{s}$ are latent feature saliency variables, and $\boldsymbol{\theta}, \boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are component model parameters.

α; note there is no *a priori* distinction between the empty clusters, i.e.

$$p(z_i = k|\boldsymbol{z}_{-i}, \alpha) = \begin{cases} \frac{n_k}{n-1+\alpha} & n_k > 0 \\ \frac{\alpha}{n-1+\alpha} & k\text{: new cluster} \end{cases} \tag{4.12}$$

Here the subscript $-i$ indicates all indices except i. The CRP induces an exchangeable distribution on partitions, so that the joint distribution is invariant to the order in which observations are assigned to clusters. One particular combinatorial characterization of the partition structure produced by the CRP is that the number of non-empty clusters almost surely approaches $\alpha \log(n)$ as $N \to \infty$. This shows that being a nonparametric prior, the CRP favors models whose complexity grows with the number of observations. Importantly, the simple predictive distributions induced by the CRP lead to efficient Monte Carlo algorithms for learning and inference.

4.3.2 Prior on Feature Saliency

We assume that s_d's, the latent indicator variable for feature selection, are independent Bernoulli random variables given feature saliency variable ρ_d's:

$$p(s_d|\rho_d) = \text{Bernoulli}(\rho_d) = \rho_d^{s_d}(1-\rho_d)^{1-s_d}, \tag{4.13}$$

and conjugate prior

$$p(\rho_d|a,b) = \mathrm{Beta}(a,b) = \frac{\rho_d^{a-1}(1-\rho_d)^{b-1}}{\mathrm{B}(a,b)}, \tag{4.14}$$

where the hyperparameters a, b are common to all features, and $\mathrm{B}(a,b)$ is the beta function and a normalization constant of the beta distribution.

4.3.3 Choices for the Likelihood Functions and Their Priors

The likelihood terms in 4.8 and 4.11 describe the dependency of observations (features and relations) on the latent variables. These models should be tailored to the specific types of the observations. The most common choices in practice are to use a Gaussian noise distribution for a numerical variable and a Bernoulli distribution for a binary one. Independently of each other, features and relations can take on either continuous or discrete values. For simplicity of discussion, we only describe the situation where all of the observations are numerical and are modeled by Gaussian distribution (denoted $\mathcal{N}$).

In slight abuse of annotation, we use a univariate variable x to represent an observation, $X_{i,d}$, and use μ_k, σ_k^2 to represent the *component and feature* specific mean and variance, then the likelihood function $f(\cdot|\cdot)$ in (4.8) is written as:

$$f(x|\mu_k, \sigma_k^2) = \mathcal{N}(\mu_k, \sigma_k^2). \tag{4.15}$$

μ_k and σ_k^2 are given conjugate Gaussian priors and conjugate inverse-gamma priors (denoted $\mathcal{IG}$) respectively:

$$f_0(\mu_k, \sigma_k^2|\lambda_\theta) = \mathcal{N}(\mu|m, \sigma^2 v)\ \mathcal{IG}(\sigma^2|\alpha, \beta), \tag{4.16}$$

where $\lambda_\theta = \{m, v, \alpha, \beta\}$ are hyperparameters common to all components. Likewise, the likelihood function $h(\cdot|\cdot)$ and the prior function h_0, which generate the relation R, are defined analogously.

4.3.4 Posterior Distribution

Having specified the model and the prior densities, we can now write the posterior, which is the distribution of the parameters conditioned on the observed data and hyperparameters. The posterior is given by

$$p(\Phi|X, R, \Lambda) \propto p(X|\Phi)p(R|\Phi)p(\Phi|\Lambda) \tag{4.17}$$

Algorithm 4: Collapsed Gibbs sampling for FIRM.

```
1. initialize latent indicator variables $\boldsymbol{z}^1, \boldsymbol{u}^1, \boldsymbol{s}^1$
2. For $t = 1, \ldots, T$
    – For each $i = 1, \ldots, I$ sample latent indicator variables $z_i$'s in sequential:
          $z_i^{t+1} \sim p(z_i | X, R, \Omega^* \backslash z_i^t, \Lambda)$
    – For each $j = 1, \ldots, J$ sample latent indicator variables $u_i$'s in sequential:
          $u_j^{t+1} \sim p(u_j | X, R, \Omega^* \backslash u_j^t, \Lambda)$
    – For each $s = 1, \ldots, D$ sample latent feature saliency variables $s_i$'s in sequential:
          $s_d^{t+1} \sim p(s_d | X, R, \Omega^* \backslash s_d^t, \Lambda)$
```

Note Ω^* denotes Ω^t except that those latent variables already being sampled at iteration t are updated with the newest values, e.g. when sampling z_i^{t+1},
$\Omega^* = \{z_1^{t+1}, \ldots, z_{i-1}^{t+1}, z_i^t, \ldots, z_I^t, \boldsymbol{u}^t, \boldsymbol{s}^t\}$. The notation $\backslash$, e.g. $A \backslash B$, denotes the set-theoretic difference.

where $\Phi = \{\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}\}$ denotes all latent variables in the model, and $\Lambda = \{\alpha_z, \alpha_u, a, b, \lambda_\theta, \lambda_\phi, \lambda_\psi\}$ denotes all hyperparameters in the model. A graphical representation of the model is given in Fig. 4.1.

4.4 Inference with Gibbs Sampling

In a fully Bayesian framework, the interest is in computing the posterior distribution over the parameters, $p(\Phi | X, R, \Lambda)$. The posterior, given in 4.17, is only known up to a normalization constant, $p(X, R | \Lambda)$, whose computation involves integrating over the unnormalized posterior, which is not analytically tractable. Instead, we approximate the posterior distribution using Markov chain Monte Carlo (MCMC).

We propose an inference procedure based on collapsed Gibbs sampling. Gibbs sampling is applicable when the joint density of the parameters is not known, but the parameters can be partitioned into groups whose posterior conditional densities are known. The collapsing is appealing when part of the parameter groups can be integrated out while the resulting marginalized posterior conditional densities are still computational feasible. The collapsed Gibbs sampler iteratively sweeps through the groups of parameters (while skipping the ones that are integrated out) and generate a random sample for each, conditioned on the current value of the others. This procedure forms a homogeneous Markov chain and its stationary distribution is exactly the joint posterior.

Algorithm 5: Sampling for the component parameters.

```
For samples $\{\boldsymbol{z}, \boldsymbol{u}, \boldsymbol{s}\}'s$ from the equillibruim distribution
    • For each $k = 1, \ldots, K$ and $d = 1, \ldots, D$ where $s_d = 1$, sample component parameters $\boldsymbol{\theta}$ in parallel:
        $\theta_{k,d} \sim \mathcal{P}(\theta_{k,d} | X_{i:z_i=k,d}, \lambda_\theta)$
    • For each $k = 1, \ldots, K$ and $l = 1, \ldots, L$ sample component parameters $\boldsymbol{\psi}$ in parallel:
        $\psi_{k,l} \sim \mathcal{P}(\psi_{k,l} | R_{i:z_i=k,j:u_j=l}, \lambda_\psi)$
    • For each $d = 1, \ldots, D$ sample parameters $\boldsymbol{\phi}$ in parallel:
        $\phi_d \sim \mathcal{P}(\phi_d | X_{\cdot,d}, \lambda_\phi)$
Note the density function $\mathcal{P}(\cdot|\cdot)$, denotes the posterior probability density function of the model with Gaussian likelihood and Normal-inverse-Gamma prior.
```

The collapsed Gibbs sampler keeps only the latent indicator variables ($\Omega = \{\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}\}$) while marginalize over the latent component model parameters ($\{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi}\}$), so the posterior density is given by

$$p(\Omega | X, R, \Lambda) \propto p(X | \Omega, \Lambda) p(R | \Omega, \Lambda) p(\Omega | \Lambda) \tag{4.18}$$

Because of the choice of conjugate priors (4.15, 4.16), the marginalized likelihood $p(X|\Omega, \Lambda)$ and $P(R|\Omega, \Lambda)$ which condition on the latent indicator variables, have analytical tractable forms:

$$\begin{aligned} p(X | \Omega, \Lambda) &= p(X | \boldsymbol{z}, \boldsymbol{s}, \lambda_\theta, \lambda_\phi) && (4.19) \\ &= \int p(X | \boldsymbol{z}, \boldsymbol{s}, \boldsymbol{\theta}, \boldsymbol{\phi}) p(\boldsymbol{\theta} | \lambda_\theta) p(\boldsymbol{\phi} | \lambda_\phi) \, d\boldsymbol{\theta}\boldsymbol{\phi} && (4.20) \\ &= \prod_{d=1}^{D} \Big[\prod_{k=1}^{\max(\boldsymbol{z})} \mathcal{M}(X_{i:z_i=k,d}) \Big]^{s_d} \Big[\mathcal{M}(X_{\cdot,d}) \Big]^{1-s_d} && (4.21) \end{aligned}$$

and

$$\begin{aligned} p(R | \Omega, \Lambda) &= p(R | \boldsymbol{z}, \boldsymbol{u}, \lambda_\psi) && (4.22) \\ &= \int p(R | \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{\psi}) p(\boldsymbol{\psi} | \lambda_{psi}) \, d\boldsymbol{\psi} && (4.23) \\ &= \prod_{k=1}^{\max(\boldsymbol{z})} \prod_{l=1}^{\max(\boldsymbol{u})} \mathcal{M}(R_{i:z_i=k,j:z_j=l}) && (4.24) \end{aligned}$$

where the subscript $i : z_i = k$ indicates all indexes such that $z_i = k$, and the density function $\mathcal{M}(\cdot)$, described in Appendix, denotes the marginal likelihood of a model with Gaussian likelihood and normal-inverse-gamma prior.

The Gibbs sampler (Algorithm 4) draws samples of z_i's, u_i's, and s_d's sequentially from their conditional densities in 4.25, 4.26, and 4.27 respectively.

$$p(z_i|X, R, \Omega \backslash z_i, \Lambda) \propto p(X, R|\Omega, \Lambda) p(z_i|\boldsymbol{z}_{-i}, \alpha_z) \tag{4.25}$$

$$p(u_j|X, R, \Omega \backslash u_j, \Lambda) \propto p(X, R|\Omega, \Lambda) p(u_j|\boldsymbol{u}_{-j}, \alpha_u) \tag{4.26}$$

$$p(s_d|X, R, \Omega \backslash s_d, \Lambda) \propto p(X, R|\Omega, \Lambda) p(s_d|a, b) \tag{4.27}$$

After the chain converges, the component parameters, which are independent conditioned on the sampled latent indicator variables, are drawn from the corresponding posterior density functions (Algorithm 5).

4.5 Experiments

We demonstrate the proposed method on an image clustering problem, and compare it to three other Bayesian nonparametric techniques: infinite relational model (IRM), Dirichlet process mixture (DPM) model, and joint feature selection and clustering with DPM (FDPM).

4.5.1 Data

We consider the setting of two domains (T_1 and T_2) and one relation between them. Domain T_1 contains the digits 0-9, and domain T_2 consists of the alphabetic characters A-J. Digit samples obtained from the MNIST dataset which consists of 28 × 28 pixel gray scale images of handwritten digits (Fig. 4.2). 100 images of each digit were randomly selected, yielding 1000 unique images. The images were rescaled so that pixel intensities are in the [0, 1] interval, and the vectorized images were arranged as the rows of the 1000 × 787 matrix X. Character samples were obtained from the Simon Lucas dataset which consists of 20 × 16 pixel binary images of handwritten letters. All 39 images of the capital letters A-J were selected, resulting in 390 unique images. The vectorized images were arranged as the rows of the 390 × 320 matrix Y. Both datasets are available online [Roweis] in Matlab format courtesy of Sam Roweis.

We assume, between the ten digits and the ten letters, there only exists 1:1 mappings in the pairs of (1, A), (2, B), (3, C)..., (9, I), and (0, J). We call these pairs the *digit:character assumption.* The pairwise relations between the 1000 samples of digits and the 390 samples of character are

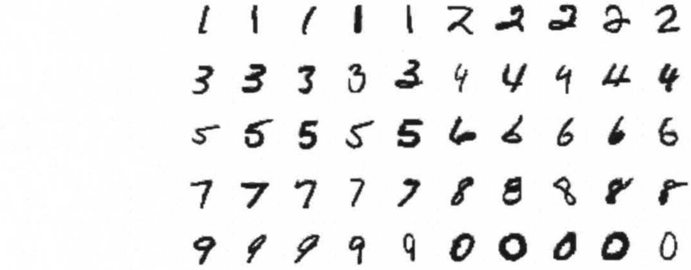


Figure 4.2: Sample images from the MNIST digit dataset

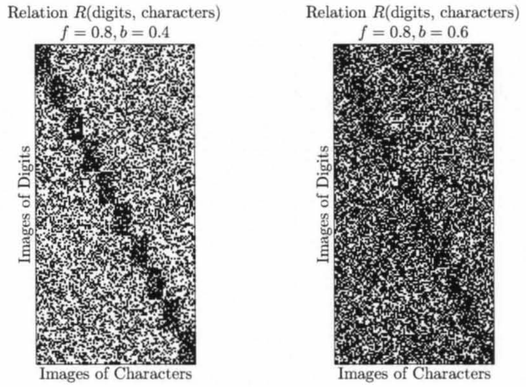


Figure 4.3: Illustration of synthetic relational data with random noise, in which each row corresponds to an image of digit, each column corresponds to an image of characters, and each entry represents whether a relation exits between the corresponding digit image and character image. The binary values are generated from Bernoulli distributions, with parameter b for background, and f for foreground. Here foreground is defined as those pairs of digit and character satisfying our digit:character assumption.

represented by a 1000 × 390 binary matrix R, where a value 1 denotes an existing mapping and 0 denotes no mapping (Fig. 4.3). The observation R is noisy, and each entry $R_{i,j}$ is drawn from a Bernoulli distribution, whose parameter is chosen according to some desired noise rate. If the digit represented by the ith digit image and the letter represented by the jth letter image follows the digit:character assumption, the noise rate should be low; otherwise, the noise rate should be high.

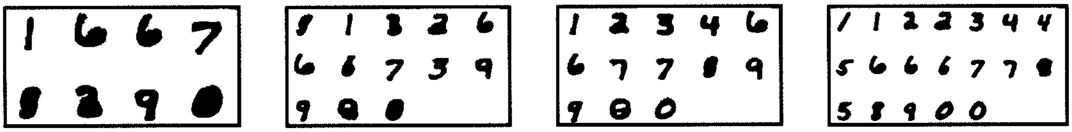
4.5.2 Task

The objective is to categorize the images and to predict missing pixels in the images. Although the digit images corresponds to the ten digits, we cannot expect the models to find exactly ten clusters with each corresponding to a digit. This is due to the facts that, first, there are high variations as to how each digit is written; and second, there exists a large proportion of low variation features surrounding all images regardless of their corresponding digit. Bayesian nonparametric models can more effectively account for the variations in the data and can infer the number of parameters automatically. To illustrate how well each of the models avoids over-fitting and under-fitting, we compare their performances on predicting missing pixels in testing images. To simply performance comparison, we focus on the setting where the digit images X and the relationship data R are made available, thus FIRM can be applied exactly as shown in Fig. 4.1.

4.5.3 Methods

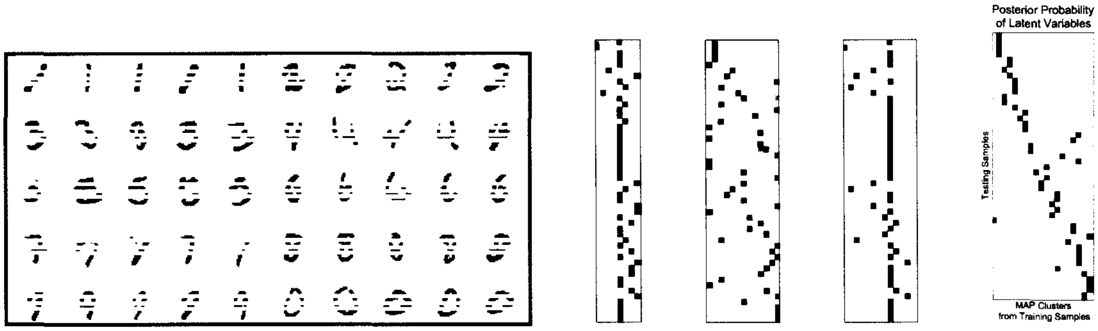
In each experiment, the collapsed Gibbs sampling algorithms for the models (DPM, FDPM, IRM, and FIRM) were performed for 100 iterations [1], the first half was used for burn-in and every fifth sample from the second half was used to estimate posterior quantities. Each iteration consisted of sampling all the latent mixture component variables and latent saliency variables. To facilitate mixing, we interleaved split-merge proposals [Jain and Neal, 2004] between Gibbs sweeps. To improve the mixing, Metropolis coupled MCMC [Geyer, 1991] was performed in which two chains were run, one of which was "heated" by raising the posterior probability to a power 0.8. Same component hyperparameters (Fig. 4.1) are used for all experiments: $\lambda_\theta = \lambda_\phi = \{m = 0, v = 1, \alpha = 5, \beta = 0.5\}$, $\lambda_\psi = \{\alpha = 0.2, \beta = 0.2\}$, $\alpha_z = \alpha_u = 1$.

[1] Tests showed that increasing the number of iterations did not lead to better predictive accuracy.



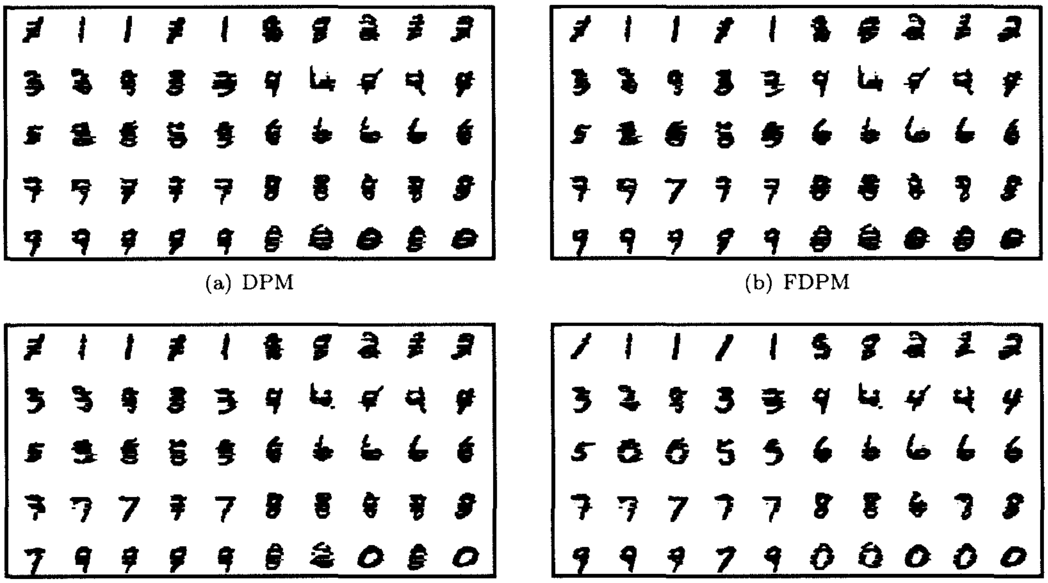
(a) DPM (b) FDPM (c) IRM (d) FIRM

Figure 4.4: Clustering 1000 images of digits. Shown is the cluster mean inferred from each of the four methodologies. Gibbs sampling algorithms were run and samples with the highest posterior probability among the Gibbs samples were selected. The number of clusters is automatically inferred from the data. The relational data R used by IRM and FIRM are generated with a moderate level of noise ($f = 0.8$ and $b = 0.2$).



(a) Testing images with missing pixels (b) DPM (c) FDPM (d) IRM (e) FIRM

Figure 4.5: Testing image and cluster prediction. (a) 50 randomly selected testing images with the same 10 out of the 28 rows missing. (b-e) Posterior probability of the latent indicator variables for testing images. Rows represent the 50 images, columns represent the MAP clusters (see Fig. 4.4), and each value represents the predictive probability of a testing image being assigned to a cluster. Each row vector sums to 1.



(a) DPM (b) FDPM

(c) IRM (d) FIRM

Figure 4.6: Missing value prediction. We run the models on the 1000 training images and predict the missing values in the 50 testing images (see Fig. 4.5) using all the Gibbs samples. The predicted missing values are shown along with the rest of the images.

4.5.4 Cluster Mean

To illustrate the performance, we first show the means of the clustering inferred from each model (Fig. 4.3). Although DPM successfully discovers four clean groups of images representing the digits 1, 6, and 7, it fails at differentiating the other images into cleaner groups. This is due to both the high variation within images of each digit, and the large proportion of low variation features around all digit images. This latter problem is effectively addressed by FDPM which, by incorporating feature selection mechanism into the clustering process, discover five new and cleaner groups representing the digits 2, 3, 6, and 9.

The artificially generated relational data has a strong bias towards the 10 digits. Therefore, by incorporating the relation, IRM would ideally have the advantage over DPM and FDPM when the goal is to find clusters congruent to the known digits 0-9. This is indeed the case as shown in the figure. Images of digit 4, though similar to those of digit 9 and assigned to the same cluster by FDPM, are assigned to a separate group by IRM; similarly, IRM separates some images of digit 7 from images of digit 9. Addressing both high and low variation feature problems, FIRM is able to produce clusters with very clear and intuitive interpretation. Image groups of digit 5 and 8, which are lost in the other models, are found by FIRM. Preferring clear interpretation, the model also split group into multiple ones even if they correspond to the same digit. For example, digits 1, 2, 5, and 8 each has two image clusters.

4.5.5 Cluster Prediction

Clustering with less interpretability often leads to poor intuitive prediction. We illustrate the performance in Fig. 4.5, where 50 testing images (not in the 1000 training images) were randomly selected and 10 out of the 28 rows (same for all images) were set as missing values. We then compute, for each testing image, the predictive probability of it belonging to the known clusters (in this case, the known clusters are inferred from the 1000 training images and we use the MAP ones (Fig. 4.4) for illustration purpose). DPM and IRM tends to classify images into black blob clusters (the fifth one in DPM and the tenth one in IRM). Regardless of the missing values, FIRM aligns almost all of the 50 images very well with the corresponding image clusters of high intuitive interpretation.

4.5.6 Missing Value Prediction

We further illustrate the predictive performance in Fig. 4.6, where missing values in testing images (Fig. 4.5) are predicted by summarizing over all possible clusterings weighted by their posterior

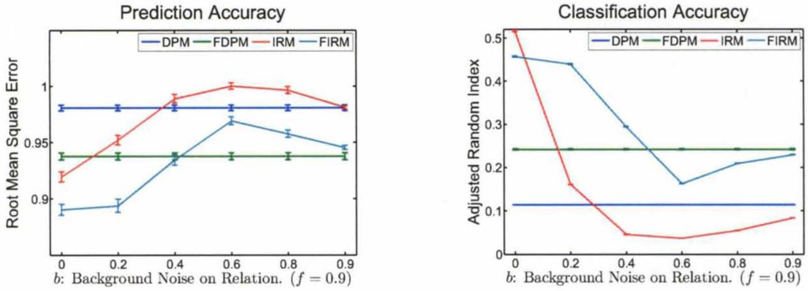


Figure 4.7: Prediction accuracy. Gibbs samplers are run on the 1000 training images. 1000 testing images (100 for each digit) are randomly selected and 10 rows are set as missing (same as in Fig. 4.5). (Top) Root mean square error (RMSE) between the real values of the missing data and their posterior expectations approximated from Gibbs samples. (Bottom) Adjusted random index between the known testing images partitioning (corresponding to digit 0-9) and their predicted clusters from the MAP of the Gibbs samples.

probability given the training images. This posterior expectation is approximated through Gibbs sampling with half burn-in. Although less intuitive interpretation does not necessarily result in a less accurate prediction on missing values, clustering algorithms which do not sufficiently address variation problems often yield poor predictions. With about 35% of rows missing, images could be confused with each other. For example, some images of digit 2 and 3 could be confused with digit 8 (Fig. 4.5). Therefore, we cannot hope to predict missing values and form images exactly as they were. Nevertheless, the predictions from FIRM complement the images very well to form clear ones than the other models, which produce smearing edges and/or lost connectivity.

4.5.7 Varying Noise Rate in Relation

The effectiveness of relational models requires reliable data sources to integrate. One would expect that the more noise embedded in the relational data, the less accurate the models will be when clustering the images. We generate the synthetic relational data using a variety of background noise rate from 0 to 0.9, while holding foreground to 90% of true positive information. We run the experiment (the same procedure as described in Sec.4.5.2-Sec.4.5.6) using these different relational data and quantify its prediction accuracy on missing values in Fig. 4.7. DPM and FDPM, indifferent to the relational data, provides two baselines of prediction accuracy. As expected, the performance of both IRM and FIRM is affected by the noise: without background noise, both methods predict more accurate than FDPM; with increasing noise, their performance decreases. However, FIRM is able to hang around much closer to the baseline of FDPM than IRM. FIRM is more robust to random noise in the relational data.

4.6 Conclusion

We described a joint feature selection and infinite relational model (FIRM), a novel approach for learning system of categories and predicting missing values from multiple data sources. We have contrasted predictive performance on the MNIST image dataset, with standard clustering model (DPM), extension to DPM with feature selection (FDPM), infinite relation model (IRM), and FIRM. Our results show that FIRM, by incorporating feature selection into relational learning, are more robust to noise, and adjust better to model complexity. This results in a more intuitive interpretation and greater predictive accuracy. We consider these results promising, and future work will explore applications of the model to other real-world data sets, and extensions to more richly structured models.

CHAPTER 5

APPLICATION OF FIRM TO BIOLOGICAL PROBLEMS

5.1 MicroRNA and mRNA Module Discovery

The cell can be represented as an overlay of at least several types of networks, which describes protein-DNA, miRNA-mRNA, protein-protein, and protein-metabolite interactions (Fig.5.1). These networks are composed of hundreds to thousands of molecules interacting via nonlinear and potentially more complex processes. The complex and dynamic nature of these networks makes human understanding of biological systems extremely difficult. Many studies on reconstructing biological networks have relied on bioinformatics approaches on genome-wide biological data, such as time-series microarray gene expression, binding motifs sequences, and ChIP-seq experimental data, using computational and statistical methods such as clustering [Reiss et al., 2006, Bar-Joseph et al., 2003, Medvedovic and Sivaganesan, 2002, Qin, 2006], supervised learning [Yeunga et al., 2011], ordinary differential equations [Greenfield et al., 2010b], and Bayesian networks [Friedman et al., 2000].

The underlying assumption of clustering methods, and many additional approaches, is that co-regulated genes tend to be co-expressed. MicroRNA (miRNA) plays an important role in biological processes by translational repression or degradation of mRNAs. For the later case, the expression levels of genes may be substantially affected by miRNAs. It is thus interesting to discover co-expressed mRNAs and miRNAs that are potentially involved in the same regulatory network (Fig. 5.2). In this study, we apply FIRM to jointly cluster mRNAs and miRNAs using a miRNA-mRNA correlation matrix and gene annotation data (details on the data are described in Section 5.1.2). FIRM aims to find mRNA and miRNA clusterings that yield clean blocks representing co-expression and potentially co-regulatory relationships between the genes and miRNAs. The searching for clean blocks on the correlation matrix is a trade-off between merging/splitting mRNA groups and merging/splitting miRNA groups. The incorporation of Gene Ontology annotation data introduces additional infor-

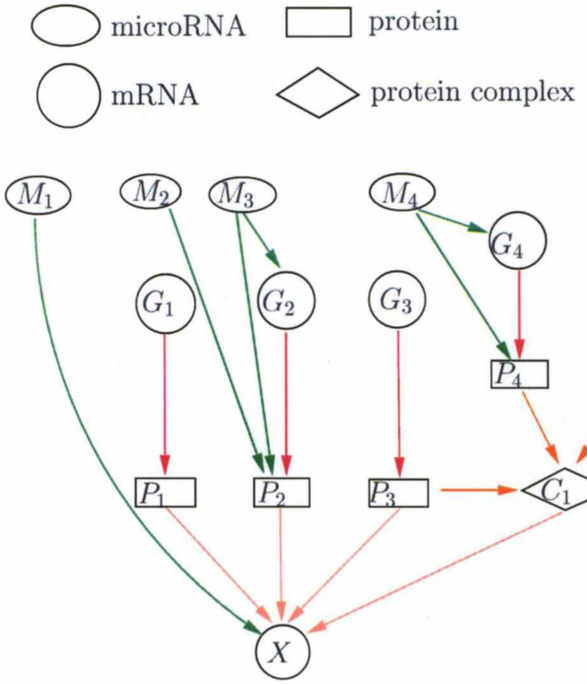


Figure 5.1: An illustration of gene regulatory network involving genes, proteins, and microRNAs. The nodes represent expression levels, circle for mRNA, ellipse for miRNA, and square for protein. The expression level of an mRNA is determined by multiple factors, including transcription factors and protein complexes which transcribe the genes, and miRNA which degrades the mRNA.

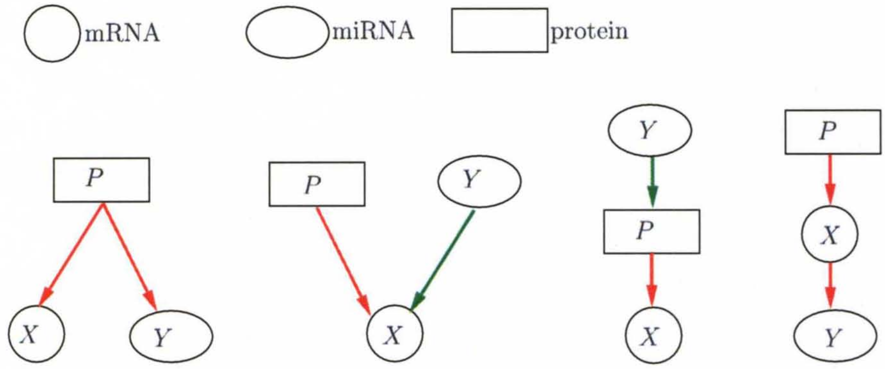


Figure 5.2: An illustration of putative regulatory relationships for a pair of positively or negatively co-expressed mRNA (denoted X) and miRNA (denoted Y). (left to right): X and Y share a common transcription factor (denoted P); P transcribes X, and Y degrades X; Y translationally represses P which transcribes X; P transcribes X which regulates Y.

mation on mRNA, which tends to result in fewer mRNA clusters due to the high degree of sparsity in the data. This in turn leads to a higher number of miRNA clusters.

5.1.1 Background

MicroRNAs (miRNAs) are a class of small non-coding RNAs involved in regulating gene expression (gene silencing) at the post-transcriptional level. These small (approximately 22 nucleotide) single-strand RNAs guide a gene silencing complex to a target mRNA by complementary base pairing, mostly at the 3' untranslated region (3' UTR) of the target mRNA. The binding of the RNA-induced silencing complex (RISC) to the conjugate mRNA causes a silence of the gene either by translational repression or by degradation of the mRNA [He and Hannon, 2004]. 1100 human miRNA has been discovered and described [http://www.microrna.org] and it has been estimated that each of the miRNAs tends to target about 100 different mRNAs [Lim et al., 2005] MiRNA plays an important role in several essential biological processes, including differentiation, cell growth, stress response and cell death [Zamore and Haley, 2005]. However, few targets of miRNAs have been experimentally validated. Attempts at identifying miRNA targets have mainly focused on bioinformatics approaches, though the procedure is very challenging because of the insufficient knowledge of microRNA biology and their targets *in vivo*.

The bioinformatics prediction of miRNA-target interactions relies on the rules of miRNA-target interactions concluded from several experimentally validated cases. Early studies show that near-perfect matches (6 to 8 continuous bases) complementarity at the 5' end of the miRNA, known as the "seed region" at positions 2 to 7, is a primary determinant of target specificity [Lai, 2002].

However, studies have shown that the presence of a perfect seed match alone is not a reliable predictor for microRNA regulation [Didiano and Hobert, 2006], due to the large number of random occurrences of any given hexamer in 3' UTRs [Betel et al., 2010], resulting in a high false positive rate. Meanwhile, studies have also found that some target sites, despite of the presence of a mismatch or a G:U wobble in the seed region, display a noticeable regulatory effect [Didiano and Hobert, 2006]. Therefore, a perfect seed match is neither necessary nor sufficient for microRNA regulation. Rather, additional factors, such as sequence-dependent 3' UTR accessibility [Didiano and Hobert, 2006], AT-richness [Robins and Press, 2005], evolutionary conservation [Lau et al., 2001, Lee and Ambros, 2001, Lim et al., 2003], and/or specific RNA- or protein-based cofactors, may be major determinants of 3' UTR responsiveness to a seed-matched miRNA [Grimson et al., 2007].

There has been a wide array of miRNA-target prediction algorithms proposed, ranging from those based on perfect seed complementarity [Krek et al., 2005], those allowing for G:U wobbles or mismatches in the seed region [John et al., 2004], to those considering secondary structure [Kruger and Rehmsmeier, 2006, Kertesz et al., 2007]. False positive predictions, which form a large portion of the predictions from these methods, are filtered through evolutionary conservation [Lewis et al., 2003, 2005], which eliminates poorly conserved candidate sites from consideration based on the observations that the phylogenetic conservation of miRNAs is very strong within mammals and often extends to invertebrate homologs [Lim et al., 2003].

Assuming that changes in mRNA expression following microRNA transfections are reasonable indicators for microRNA regulation, Betel et al. [2010] proposed a model for predicting mRNA expression changes after microRNA transfections, by incorporating target site information, contextual features, conservation, and mRNA expressions into a single support vector regression (SVR) model.

5.1.2 Description of Data

Expression profiling of mRNA and miRNA has been used to characterize various tissues and tumors. In breast cancer, mRNA and miRNA profiling has been used to associate them with clinical and pathological characteristics. Abnormal expression levels of several mRNAs and miRNAs have been shown to be associated with multiple cancer types including breast cancer. In this work, we downloaded the expression profiling of miRNAs and mRNAs in 101 human primary breast tumor samples from Gene Expression Omnibus (GEO) with accession ID GSE19536 [GSE19536].

According to the data contributors [GSE19536], the miRNA profiling from total RNA was performed using Agilent Technologies. miRNA signal intensities for replicate samples were averaged and log2 transformed. The expression levels were normalized to the 75th percentile. MiRNAs that were detected in less than 10% of the samples were filtered out, resulting in 469 miRNAs considered to be expressed in this set of human breast tumors and used in further analysis. The mRNA profiling from total RNA on the same samples were performed on an Agilent catalog design whole human genome 4x44K single channel oligo array. Scanning was performed on Agilent Scanner G2565A and signals were extracted using Feature Extraction v9.5. Data were log2 transformed, non-uniform spots were excluded. Population outliers were excluded when averaging replicated probes. Probes that are missing on more than 10 arrays were excluded. Quantile normalization was performed in R Bioconductor using Limma [Smyth, 2005] and missing values were imputed using LLS imputation [Kim et al., 2005]. We used the top $10,000$ probes with the highest expression variance across the samples for further analysis. Spearman correlation is calculated for each pair of mRNA and miRNA, yielding a $10,000 \times 469$ correlation matrix.

The Gene Ontology (GO) project [Ashburner et al., 2000] provides an explicitly defined vocabularies to describe the biological properties of genes. GO consists of two components: the GO ontology, which defines terms along with the structured relationships between the terms; and GO annotation, which map the associations between gene products and the terms. GO provides both ontologies and annotations for three areas of cell biology: molecular function, biological process, and cellular component. A GO term consists of a name, an identifier, a definition with cited sources, among others. The GO ontology is structured as a directed acyclic graph (DAG), where the nodes represent the terms and the edges represent relationships among them. In a DAG, the parent terms represent more general entities than their children terms, and a term may have multiple parents. A GO annotation associates a gene with terms in the ontologies and is generated either by experiment or by computational prediction. To represent the knowledge of a gene, the annotation associates it with as many terms as appropriate, and with the most specific terms available. Once a gene is annotated to a term, by inheritance, it is also associated with all the terms on the path from this term to the root term (in the DAG). The annotation relationships of a gene and its (inherited) terms must all be accurate or the ontology must be revised. The GO ontology and annotations are continually updated to correct errors and reflect current knowledge. In this study, we transformed the GO annotations to a flattened vector representation, i.e. each gene is represented by a binary vector of length N (the number of all GO terms), and a value of 1 denotes an association between the gene and the term. Filtering the genes not associated with any term, and filtering the terms with fewer than 10 associated genes, resulted in a binary matrix containing association information for 6128 genes and 3325 GO terms, which is referred to the flattened GO matrix.

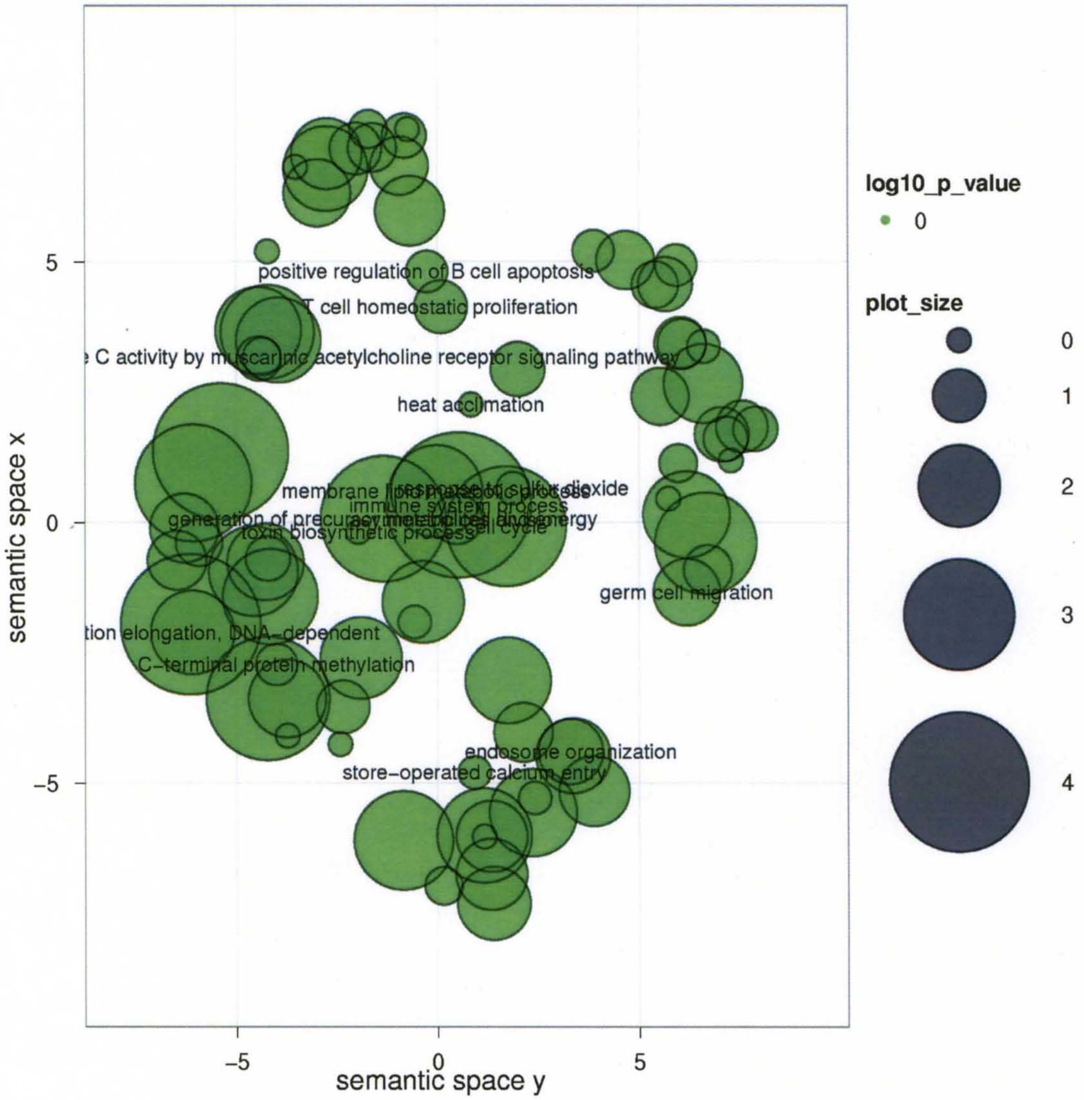


Figure 5.3: A visualization of the GO terms selected by FIRM using REVIGO. Cluster representatives (i.e. terms with redundant ones filtered out) are plotted in a two dimensional space derived by applying multidimensional scaling to a matrix of pairwise semantic similarities between GO terms. Bubble size indicates the frequency of the GO terms in the underlying GO annotation database. The more general a term, the larger its corresponding bubble.

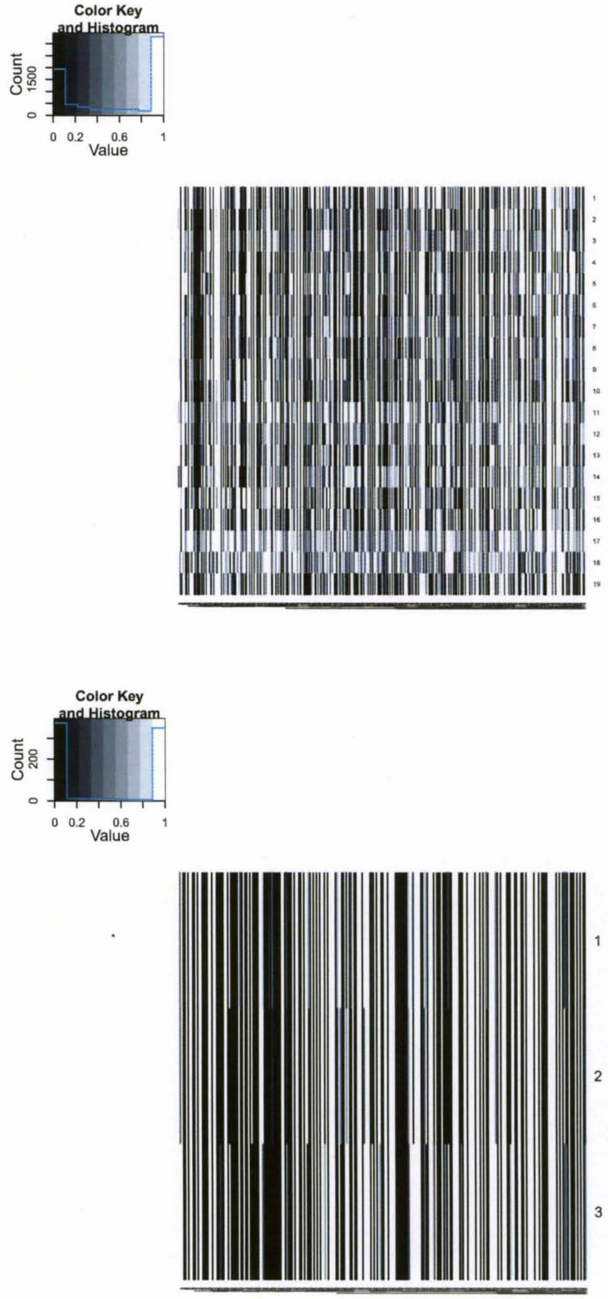


Figure 5.4: Heatmaps of the adjusted P-values, where each row denotes an mRNA cluster, and each column denotes an miRNA cluster. The null hypothesis states that a correlation submatrix has mean 0, and the alternative hypothesis states that the mean is less than 0. (Top) Results from FIRM. (Bottom) Results from IRM.

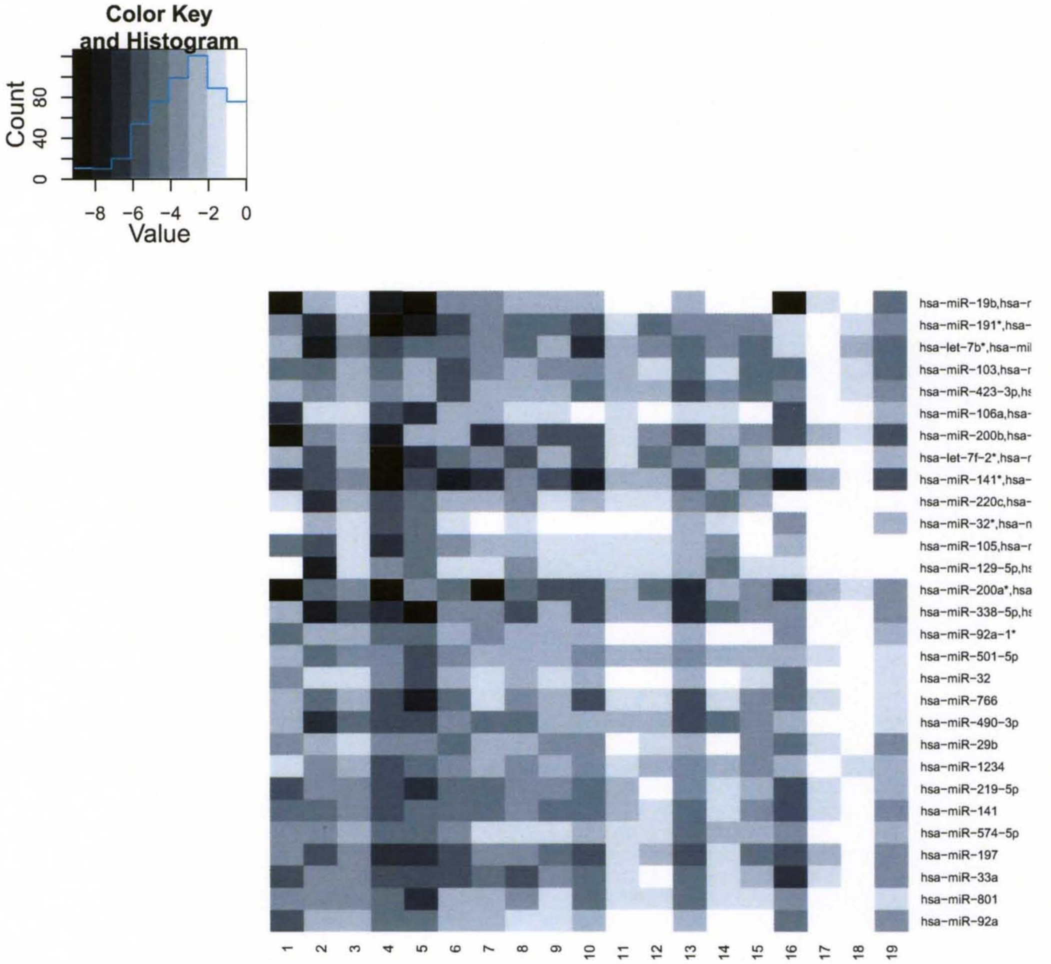


Figure 5.5: Heatmaps of the adjusted P-values for representative miRNA clusters, where each row denotes an miRNA cluster, each column denotes an mRNA cluster, and each entry is the log 10 based P-values. Only the 29 highly negatively correlated miRNA clusters (P-value less than 10^5) with at least one mRNA groups are chosen.

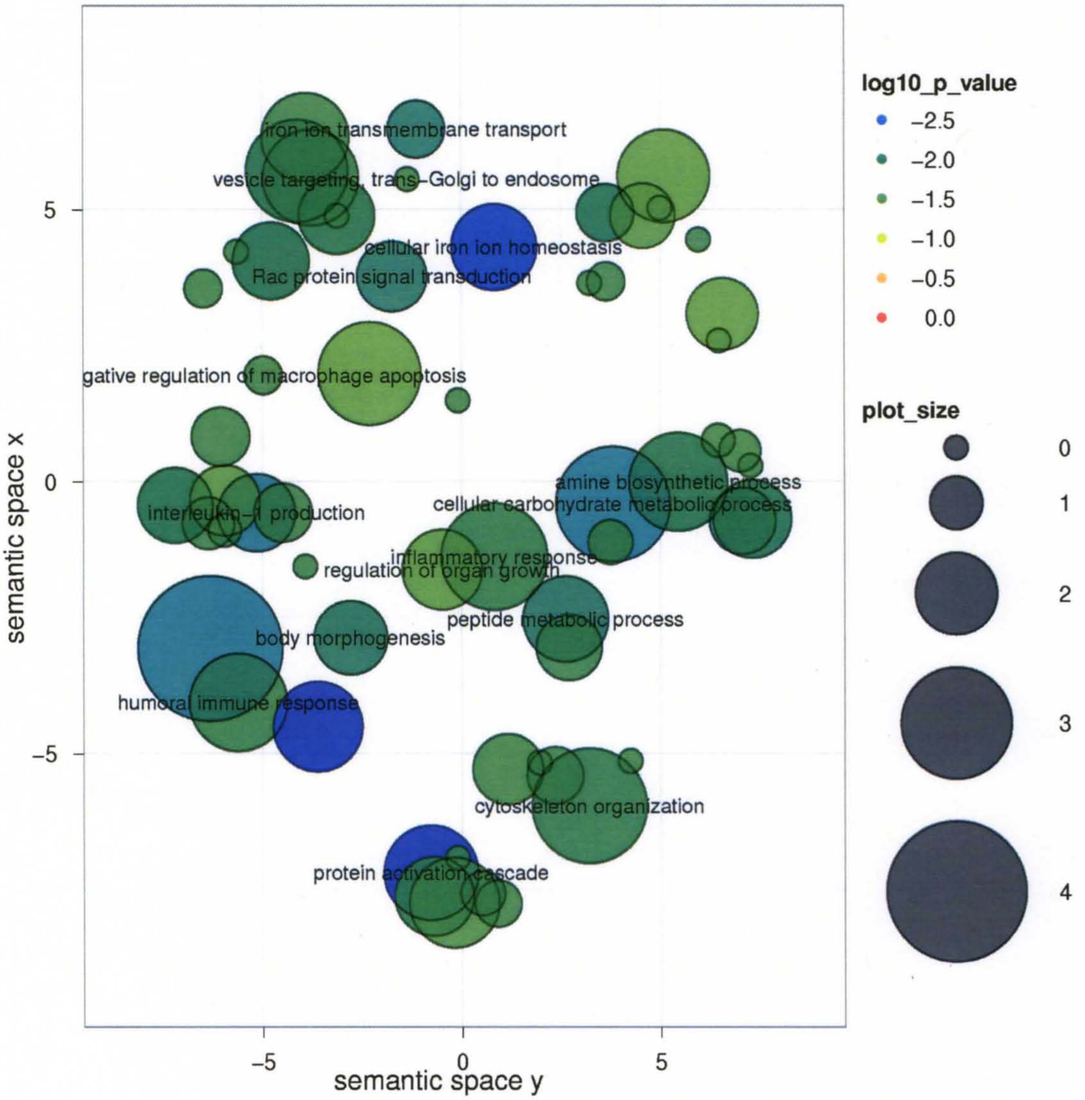


Figure 5.6: A visualization of the GO terms enriched by mRNA cluster 4, using REVIGO. Cluster representatives (i.e. terms with redundant ones filtered out) are plotted in a two dimensional space derived by applying multidimensional scaling to a matrix of pairwise semantic similarities between GO terms. Bubble color indicates the P-value from hypergeometric test. Bubble size indicates the frequency of the GO terms in the underlying GO annotation database. The more general a term, the larger its corresponding bubble.

5.1.3 Results

We applied various clustering algorithms on the data. The flattened GO matrix is extremely sparse (with only about 2.2% of the entries for known associations) The infinite mixture (or DPM) model inferred three gene clusters, indicating it favors a very low number of clusters due to the limited differentiating factors within the data. Infinite relational models combine the flattened GO matrix and the correlation matrix, and is presumed to be able to balance the two types of data when clustering. However, it inferred three gene clusters as well, and thus was also subjected to the sparsity issue. Feature selection tends to filter out those features which do not provide significant differentiating power. Applying FDPM to the GO matrix led to five gene clusters and 1747 selected GO terms, while applying FIRM on the GO matrix and the correlation matrix yielded 19 gene clusters and 106 selected GO terms. To further investigate the GO terms selected by FIRM, we use REVIGO [Supek et al., 2011] (with default parameters) to clustering the GO terms selected by FIRM, relying on semantic similarity measures, and to visualize the list of GO terms in scatterplots. The results indicate that FIRM selected a set of GO terms with very low degree of redundancy, and the set includes several groups of GO terms crucial to tumor control (e.g. regulation of mature B cell apoptosis, and immune system process) (Fig. 5.3).

FIRM and IRM categorize mRNAs and miRNAs in a way such that the clusterings would result in relatively clean blocks of correlations representing co-expressing relationship. The clustering of mRNAs and miRNAs are thus depending on each other. Both FIRM and IRM estimated a large number of clusters (373 from FIRM and 256 from IRM) for the 469 miRNAs due to the low correlations among miRNA and mRNA expressions. One-sided (less than) student's t-test are performed on each of the correlation submatrices yielded from the clustering, where the null hypothesis states that the sample correlations have mean zero (Fig. 5.4). The P-values are adjusted for multiple comparisons through the control of the false discovery rate [Benjamini and Hochberg, 1995]. Several submatrices have significantly high negative correlations between the corresponding mRNA and miRNA clusters (Fig. 5.5). The miRNA cluster consisting of two miRNAs (hsa-miR-141*, hsa-miR-200c) are highly negatively correlated (P-value less than 10^5) with ten gene groups, and there are 29 miRNA clusters, with a total of 49 miRNAs, that are highly negatively correlated (P-value less than 10^5) with at least one gene group (Fig. 5.5).

The pairs of mRNA and miRNA groups that have the largest P-value ($\sim 6.67e - 10$) are (mRNA group 1, "hsa-miR-19b,hsa-miR-20a,hsa-miR-20b"), (mRNA group 16, "hsa-miR-19b,hsa-miR-20a,hsa-miR-20b"), (mRNA group 4, "hsa-miR-191*,hsa-miR-411*,hsa-miR-425*"), (mRNA group 4, "hsa-miR-141*,hsa-miR-200c"). Hypergeometric test on the enrichment of GO terms from the biological process category is performed for each mRNA group using GOstats [Falcon and Gentleman, 2007]. Several GO terms important to tumor control, including negative regulation of macrophage apoptosis, humoral immune response, and inflammatory response, among others, are significantly enriched by mRNA group 4 (Fig. 5.3).

5.2 Drug-Target Interaction

The identification of protein function and the prediction of ligand-target interaction is an active research field facilitated by means of categorizing ligands and proteins into biologically sensible groups. Because of the pharmacological fact that related drugs can bind to receptors without obvious sequence or structural similarity, it is appropriate to categorize proteins based not only on their sequence or structures but also on the chemical structure and the phenotypic side-effect of their ligands. In chemogenomic studies where the complete set of ligands for a protein is not known *a priori*, integrating the *de novo* detection of interacting ligand and protein groups into the categorization process can guide the process towards more biologically sensible solutions.

Various data types have been collected for drug-target interaction prediction, including chemical compound descriptors, protein sequences, ligand-target bindings and pharmaceutical effects. We apply a special case of FIRM (and a special case of IRM as well, where all input data are relational) to jointly detect biologically sensible ligand groups and protein groups, and to predict drug-target interactions. The method takes advantage of the Bayesian nonparametric paradigm for integrating multiple data types, for allowing for missing values (e.g. unknown ligand-target interaction) in the data, for automatically inferring the number of clusters without explicit model comparison, and for predicting the ligand-target interactions.

5.2.1 Background

The identification of interactions between ligands (chemical compounds, drugs) and proteins (receptors, targets) is an important step of drug discovery. A protein's chemical conformation, i.e. its three dimensional shape, determines its functional state. When a **ligand** (usually a small molecule) binds to a site on a target protein, it may cause a conformational change in the receptor protein, resulting in and altered behavior of the receptor and triggering physiological response. The binding occurs by intermolecular forces, such as ionic bonds and hydrogen bonds, and is usually reversible. The potency of a drug depends on both the **binding affinity** which characterize the intermolecular forces, and the ligand efficacy which refers to its ability to produce a biological response and the degree of this response.

Traditionally, drug discovery has been an effort of screening (or selecting) ligands according to its properties for a single protein target. When a compound interacts with a target in a productive way, that compound is considered as having the potential of becoming a drug. Compounds fail the initial screen may be screened later against other targets. This is an extremely time consuming effort, and among the estimated 3000 druggable proteins [Russ and Lampel, 2005] in human, only approximately 800 has been investigated by the pharmaceutical industry [Paolini et al., 2006]. To address the issue, advances in assay and instrument technologies have been made for high-throughput miniaturization and parallelization of compound synthesis. This high-throughput screening process allows for batches (millions) of compounds being tested for binding activity or biological activity against batches of target molecules [Macarron et al., 2011]. Despite all of the technology advancement, however, drug discovery requires tremendous chemical and biological insight, and the experimental effort requires a high quality compound library and target library, and reliable high-throughput binding and functional assay. As a result, our current knowledge about the compound and protein interactions is relatively limited. Among over 10 million non-redundant chemical structures in the chemical space, only a small fraction of them has been tested on a fraction of the entire target space and only approximately 1000 have been approved as drugs [Rognan, 2007].

In silico prediction (or virtual screening [Rognan, 2007]) methods complementing existing experimental approach. One such *in silico* method is molecular **docking**, which, predicts the preferred orientation and affinity of a binding between a pair of ligand and receptor. Based on the knowledge of a protein's 3-D structure, the prediction is made either by matching the molecular surface descriptions of the pair, or by simulating a chemical environment for the pair to bind. The information on a proteins' structure are usually determined through biophysical techniques such as x-ray crystallography. Docking also requires an efficient algorithm for searching through all possible orientations and conformations, as well as a sensible scoring function for binding affinity estimation. An alternative strategy is through supervised machine learning and can be categorized into three categories [Rognan, 2007]: **ligand-based**, **target-based**, and **ligand-target based**. The idea underlying ligand-based *in silico* screening is that similar ligands are likely to have similar binding profiles, i.e. targeting the same proteins. For example, structural descriptions on a set of chemical compounds are collected; from these compounds, there is a subset, considered as training set, whose binding status against a protein of interest are known *a priori*. The supervised model aims to predict, for the remaining compounds (testing set), their probability and/or affinity of binding against the protein, by building a functional mapping between the structural description space and the binding status space using the training information [Xia et al., 2004, Nidhi et al., 2006]. The functional mapping can be linear or nonlinear, and often a wide array of supervised models, such as logistic regression, Fisher discriminant analysis, and SVM, are applied and their performance are compared or combined. In ligand-based approach, the proteins are considered independent and a separate model is built for each protein of interesting. Analogically, a target-based learning approach would build the model based on protein similarity in the target space.

A ligand-target based approach jointly takes into account all three types of information, i.e. drug chemical structures, target protein descriptions and the currently known drug-target interactions, therefore allows for information propagation in a "single" learning process. The essential idea of the approach is to unify the structural information in the original spaces (ligand and protein) into a **joint ligand×target space**. The approach would then search for a functional mapping from the joint space to the binding status. Kernel learning methods for machine learning [Cristianini and Shawe-Taylor, 2000, Schölkopf and Smola, 2002, Bishop, 2006] work directly with pairwise distances or similarities between observations, rather than an explicit feature representation which is not readily available. The learning problem is thus casted into one of defining a kernel that captures the intrinsic nature of the problem. Erhan and L'Heureux [2006] defines a kernel over the joint ligand × target space by first defining similarity measures between pairs of targets, then between pairs of compounds, and then combining the two measures into a kernel function of the desired type. The method first defines the kernel on proteins as a linear combination of four standard kernels, including an identity kernel, a Gaussian kernel, a correlation kernel, and a quadratic kernel. The kernel on compounds is defined exactly the same way. Then the kernel on the joint space is defined by the product of the protein kernel and compound kernel. This factorization dramatically reduces the computational complexity of working with tensor products in large dimensions. Ivanciuc [2007] reviews a wide range of compound kernels used in chemoinformatics, including those based on 2D or 3D fingerprints of compounds, and those based directly on detecting common substructures in the compounds' 2D or 3D structure. A variety of approaches have also been proposed to design kernels for proteins, ranging from those based on the amino-acid sequence of a protein [Jaakkola et al., 2000] to those based on the 3D structures of proteins [Borgwardt et al., 2005]. Jacob and Vert [2008] proposes a new protein kernel which incorporates information on protein categorization with respect to ligand binding.

Another ligand-target based approach is called bipartite graph learning method [Yamanishi et al., 2008], which models the interactions using a bipartite graph between proteins and ligands, and predicts interactions between a ligand-protein pair based on the distance between the two on the learned graph. Bleakley and Yamanishi [2009] proposes the bipartite local models, which first adapts the bipartite graph strategy to ligand-based and target-based learning, independently. It then combines the prediction results from the two in a straightforward way, e.g. the predicted interaction score for a pair is the higher one between scores from ligand-based and target-based. Van Laarhoven et al. [2011] further explores the relevance of the topology of ligand-protein interaction bipartite graph as a source of information for predicting interactions. Besides compound fingerprints, pharmaceutical effect information has also been used to quantify compound similarity in prediction [Yamanishi et al., 2010].

5.2.2 Description of Data

This section briefly describes the various representations of ligands and proteins, and introduces the experimental dataset collected by Yamanishi et al. [2010], which contains the drug-target interaction matrix, and the similarity matrices between ligands and between proteins.

Chemical Data

Chemical compounds can be described in different details and dimensionalities, and are usually classified as one, two, or three dimensional (1-D, 2-D, 3-D) descriptors [Rognan, 2007]. A basic 1-D descriptor accounts for the compounds' global properties (for example, molecular weight, atom and bond counts), which can be derived from the chemical formulae. A popular representation of this kind is the "Simplified Molecular Input Line Entry System" or SMILES [Weininger, 1988]. Fingerprint-based methods [Willett, 2006] are widely used in both 2-D and 3-D descriptors. Fingerprints are easy to derive, represent, and computationally easy to compare. For 2-D descriptor, a bit string, called a "fingerprint", encodes the occurrence of predefined structural events (atoms, fragments, rings, substructures and 2-D pharmacophores). A fingerprint-based 3-D descriptor encodes conformation-specific properties into a bit string, including the occurrence of all possible pharmacophore tuplets (doublets, triplets, and quadruplets), their corresponding features (e.g., H-bond acceptor, and positively ionisable atom), and interfeature distances.

Yamanishi et al. [2010] collected structures of chemical compounds from the KEGG DRUG and KEGG LIGAND databases [Kanehisa et al., 2008], and computed the chemical structure similarities between compounds using SIMCOMP [Hattori et al., 2003], a program that finds the common substructures between two compounds and outputs the global similarity score based on a graph alignment algorithm. The similarity between two compound structures x and y is evaluated by Tanimoto coefficient defined as $S_{chem}(x, y) = |x \cap y|/|x \cup y|$. As in Yamanishi et al. [2010], the similarity score is referred to as chemical structure similarity.

Pharmacological Data

Yamanishi et al. [2010] also collected pharmacological effect keywords for drugs (pharmaceutical molecules) from the JAPIC (Japan Pharmaceutical Information Center) database. The key word was then translated into English followed by the unification of synonymous words. There are total of 17109 keywords tagged "pharmaceutical effect". Each drug is represented by a binary vector of length 17109 in which 1 denotes the occurrence of the pharmaceutical effect keyword for the drugs, and 0 otherwise. The "pharmacological similarity" between two drugs is evaluated by the weighted cosine correlation coefficient between vectors of their keywords, where weights are introduced to emphasize infrequent keywords rather than frequent keywords across different drug package inserts.

Genomic Data

Yamanishi et al. [2010] collected amino acid sequences of the human genome from the KEGG GENES database [Kanehisa et al., 2008], and computed the normalized version of Smith-Waterman score [Smith and Waterman, 1981] which quantifies the amino acid sequence similarities.

Drug-Target Interaction Data

The interactions between drugs and target proteins are collected by Yamanishi et al. [2010] from several online drug-target databases. The numbers of known drugs with pharmacological information in JAPIC are 212, 99, 105 and 27, for their targets enzymes, ion channels, GPCRs and nuclear receptors, respectively. The numbers of the corresponding target proteins in these classes are 664, 204, 95 and 26, respectively. The numbers of the corresponding interactions are 1515, 776, 314 and 44, respectively.

5.2.3 Results Using Drug-Target Interaction Data Alone

To illustrate the performance, we first show the categorization result inferred by using the drug-target interaction data alone. The hypothesis is that both ligands and proteins can be put into sensible groups within each of which the elements share a common binding activity. Unlike traditional feature-based clustering algorithms which assume features are independent when clustering observations (for example, proteins are independent when clustering drugs (or vice versa)), relational clustering algorithms (e.g. FIRM and IRM) aim to find clear blocks by jointly clustering both dimensions. By randomly selecting entries of the interaction matrix as missing values (with various percentage), we are able to simulate the real-world case where a great deal of drug-target interaction relationships are undetermined yet. The clean blocks from applying FIRM (Fig. 5.7) suggest interesting drug and target groups. Even with half of the information missing from the original data, the blocks are still homogeneous, suggesting a clustering that is able to capture the category-level interactions.

Clusters not only facilitate description and communication, they also allow for prediction on unobserved drug-target interactions. For example, it is very likely that missing entries in the blocks filled with black and red represent interactions (Fig. 5.7). To systematically test the intuition, we perform N-fold cross validation (N=10, 5, 3, 2) and plot the ROC curves (Fig. 5.8). For example, in 2-fold cross-validation, half of the entries in the matrix are randomly selected and used as training data, while the remaining entries are used for testing. Gibbs sampling was performed for 100 iterations, with the first half used for burn-in and every fifth sample from the second half used to estimate posterior quantities.

5.2.4 Results Using Various Data Sources

Although the gold standard drug-target interaction data provides the most direct interaction information (Fig. 5.7), and can be used alone to predict unobserved interactions (Fig. 5.8), there are several motivations for incorporating other data sources. First, the drug-target data may contain no information on a protein or compound, and therefore, sequences or structures are the only information available. Second, the drug-target data may consist of only a very few entries for a protein or compound. To apply amino sequence and other similarity data for interaction prediction, it is helpful to first visualize how congruent they are to the drug-target data. One way to do so is drawing the heatmap of a similarity matrix, with the rows and columns ordered according to the clustering structure inferred by using the drug-target data alone (Fig. 5.9). Ideally, the blocks on the diagonal of the heatmap (as defined by the clustering) would have higher similarity values, indicating that proteins (or drugs) in the same interaction groups (as inferred with the drug-target data alone) have similar sequence (or structures). We cannot expect a perfect match though, because of the fact that similarity measures are simplified indirect representations of interaction activity. Nevertheless, some similarity matrices may have a higher congruence to the drug-target data than others. For example,

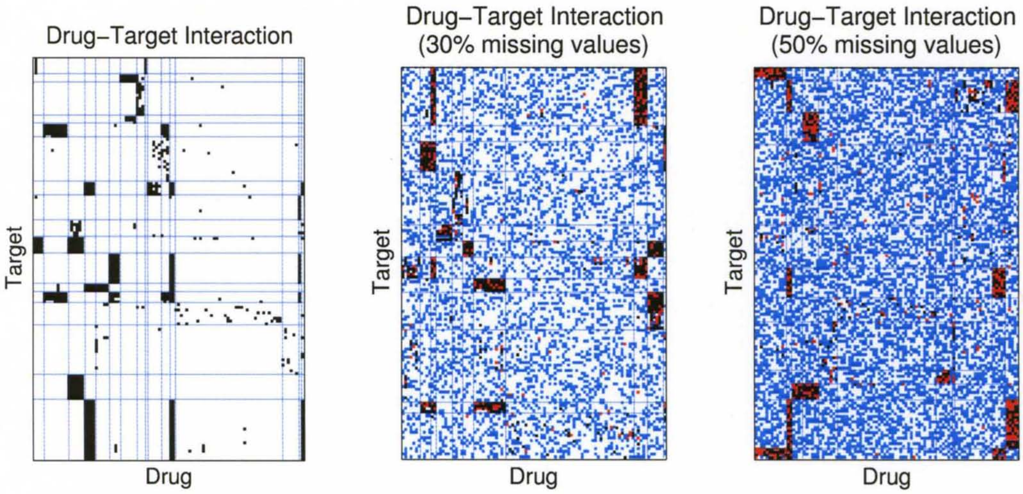


Figure 5.7: Biclustering of drug-target interaction gold standard data, with varying degree of missing values (unknown interaction status). Row represents target, column represents drug, and each entry in the binary matrix represents whether there exists an interaction between the corresponding drug and target. Gibbs sampling algorithm was run and samples with the highest posterior probability among the Gibbs samples were selected. The orders of the drugs and targets in the heatmaps are according to the biclustering. Black and white show known interactions and non-interactions, respectively. We randomly select a percentage of entries and mark them as missing values using either red (for existing interactions) or blue (for non-interactions). (Left) The complete data. (Center) Randomly select 30% of the data as missing values. (Right) Randomly select half of the data as missing values.

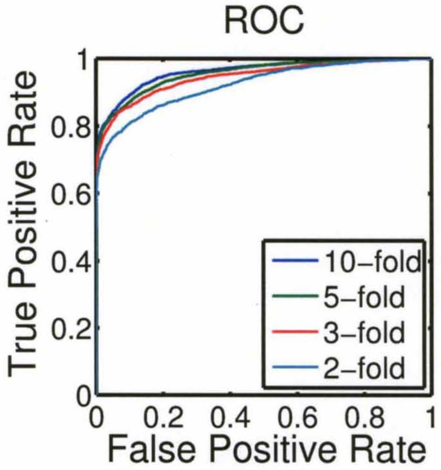


Figure 5.8: Prediction on unobserved drug-target interactions using the partially observed drug-target interaction data alone. Gibbs sampling was run, and posterior expectation of each missing entry was estimated, on 5 times N-fold cross validation (N=10, 5, 3, 2).

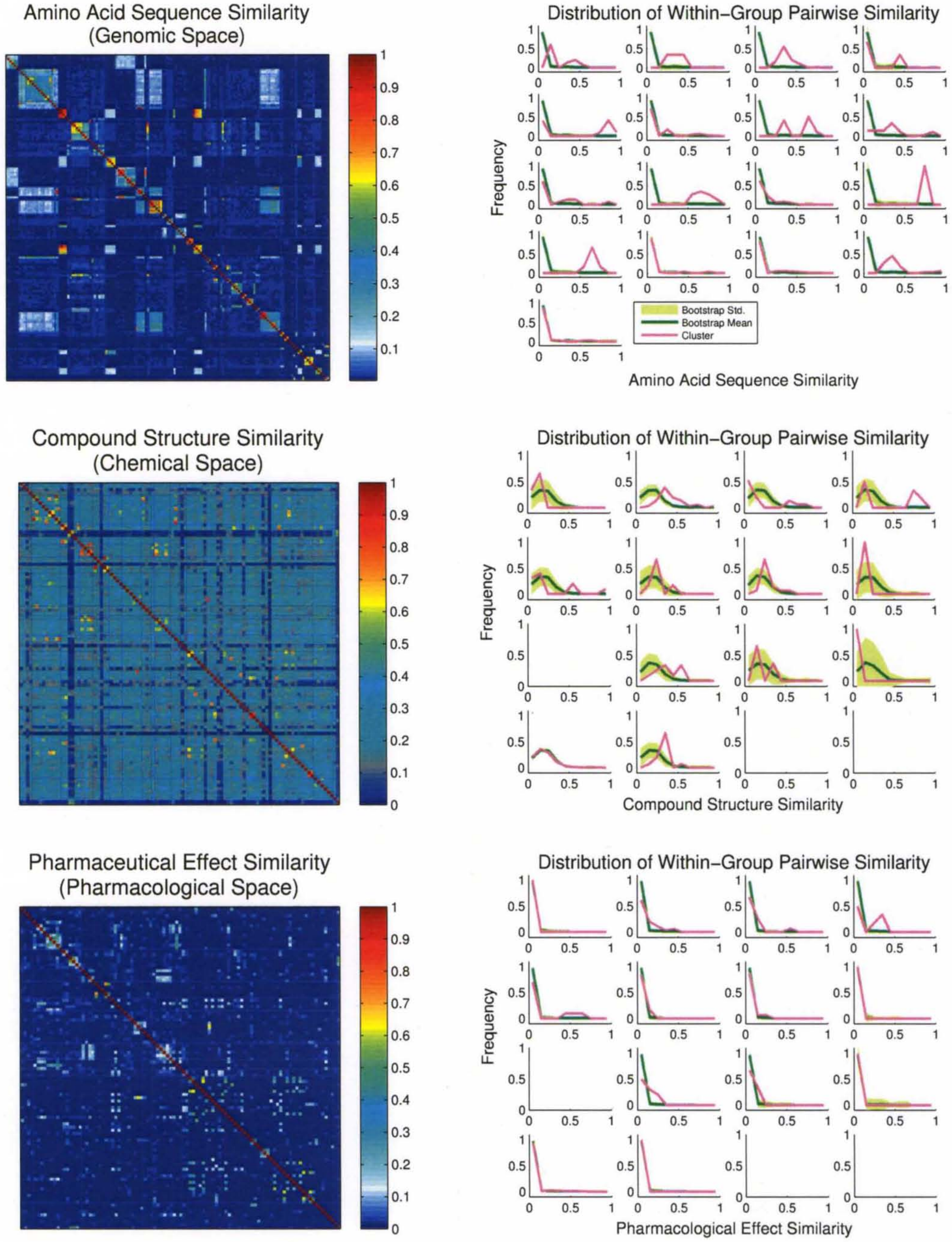


Figure 5.9: An illustration of congruence between drug-target interaction data and similarity matrices. (Top) Genomic space. (Center) Chemical space. (Bottom) Pharmacological space. The left column shows heatmaps with row and column ordered by clusters inferred from drug-target interaction data, and the right column demonstrates distribution of within-cluster pairwise similarity (in red) against bootstrap ones (in green).

as shown in Fig. 5.9, in genomic space, the blocks have relatively high similarity values and are likely not due to randomness, while the pharmacological space has relatively low similarities in the blocks.

The prediction accuracy of an algorithm is affected by random error, data error (e.g. biased sample), and methodology error (e.g. over-fitting or under-fitting). While we cannot avoid random error, it is possible to improve performance by collecting data more congruent to the prediction, and devising more accurate algorithms. Here we study the effectiveness of the different data sources on the prediction task, when applying the same prediction framework of relational clustering. For convenience, we define a ligand as a **known drug** if its interaction status with many proteins has already been observed, and **unknown drug** otherwise. Similarly, a **known target** refers to a protein whose interaction status with many drugs has been observed, and **unknown protein** refers to otherwise. Depending on the characteristic of the prediction tasks, there are several different ways of using the data sources for clustering:

- For an unobserved status between a pair of known drug and known target, the drug-target interaction data alone contains the most relevant information for the prediction (Sec. 5.2.3, Fig. 5.8).

- For a pair of known drug and unknown protein, the prediction requires both the genomic data and the drug-target data. In this case, we first simultaneously cluster the known drugs and known targets, using both data (Fig. 5.10). Then for each unknown protein, we predict its cluster membership to the inferred clustering based on the genomic information. Their interactions with unknown drugs are then predicted based on the inferred cluster assignment. Note that the procedure is done in the Bayesian nonparametric framework with predictions averaged over uncertainties on clusterings.

- For a pair of unknown drug and known protein, besides the drug-target data, it needs information from either the chemical data or the pharmacological data. The prediction procedure is analogical to the case of known drug and unknown protein as described above.

- For a pair of unknown drug and unknown protein, besides the drug-target data, the interaction prediction requires genomic information, as well as ligand information from either the chemical data or the pharmacological data. The prediction procedure is in the same fashion as the other cases, i.e. we first simultaneously clustering the known drugs and known targets, using all three types of data covering genomic space, ligand space, and the drug-target space (Fig. 5.10). Each unknown protein (and unknown ligand) is then categorized into the inferred clustering, which

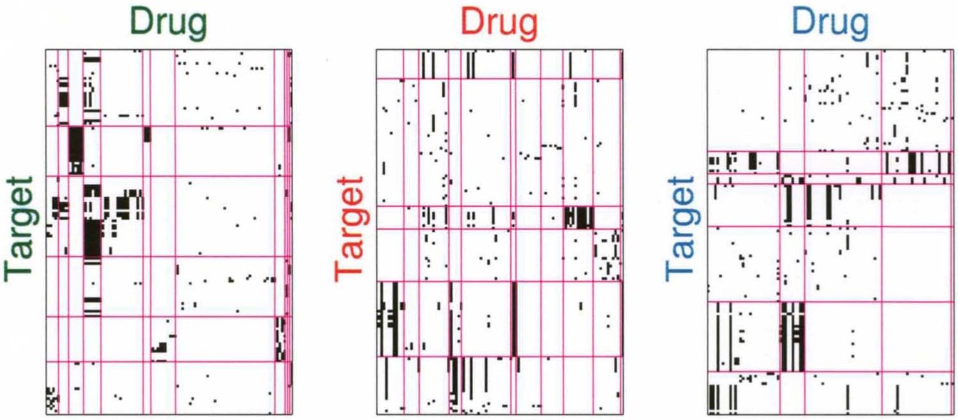


Figure 5.10: Illustration of the integrative biclustering of the drug-target interaction data and other data sources. The drug-target matrix are shown with drugs and targets ordered according to the clustering. (Left) Integrated with genomic data. (Center) Integrated with chemical data. (Right) Integrated with pharmacological data.

is then used to infer the interaction relationship.

We conduct a 10-time 5-fold cross validation for each of the various scenarios 5.11. For example, in the case of known drugs and unknown proteins, the proteins (and as a result, the drug-target data and the genomic data) are randomly partitioned into five sets, each of which is used as testing set (unknown proteins) once while the remaining is considered as training set. Gibbs sampling is run on the training data set to jointly cluster the drugs and known targets. The sampled distribution for the clustering is then used to predict interactions for the testing set of proteins. This cross validation procedure is repeated 10 times for the case. Ideally, the gold-standard drug-target data would contain the observation for any drug-target pair belongs to the set of known drugs and known proteins. We cannot expect so, however, in a real-world case because a great deal of the interactions has not yet been determined. Therefore, as in Sec. 5.2.3 and illustrated in Fig. 5.7, we randomly select entries of the interaction matrix as missing values (with a variety of percentages). The drug-target matrix consists of the most direct information on interaction, and thus results in the best prediction performance when the percent of missing values is low (Fig. 5.11, 30% missing). Meanwhile, interaction prediction for unknown proteins, through the genomic data, yields better performance than prediction for unknown ligands, through the chemical or pharmacological data. This is consistent with the analysis that the genomic space is more congruent to the drug-target data than the other spaces do (Fig. 5.9). As the missing observations in the gold-standard interaction matrix increase, the performance of all cases is affected. Nevertheless, due to the introduction of genomic data which is highly relevant to the interaction, the prediction for unknown proteins using genomic data remains accurate while the performance of using interaction matrix alone dropped dramatically.

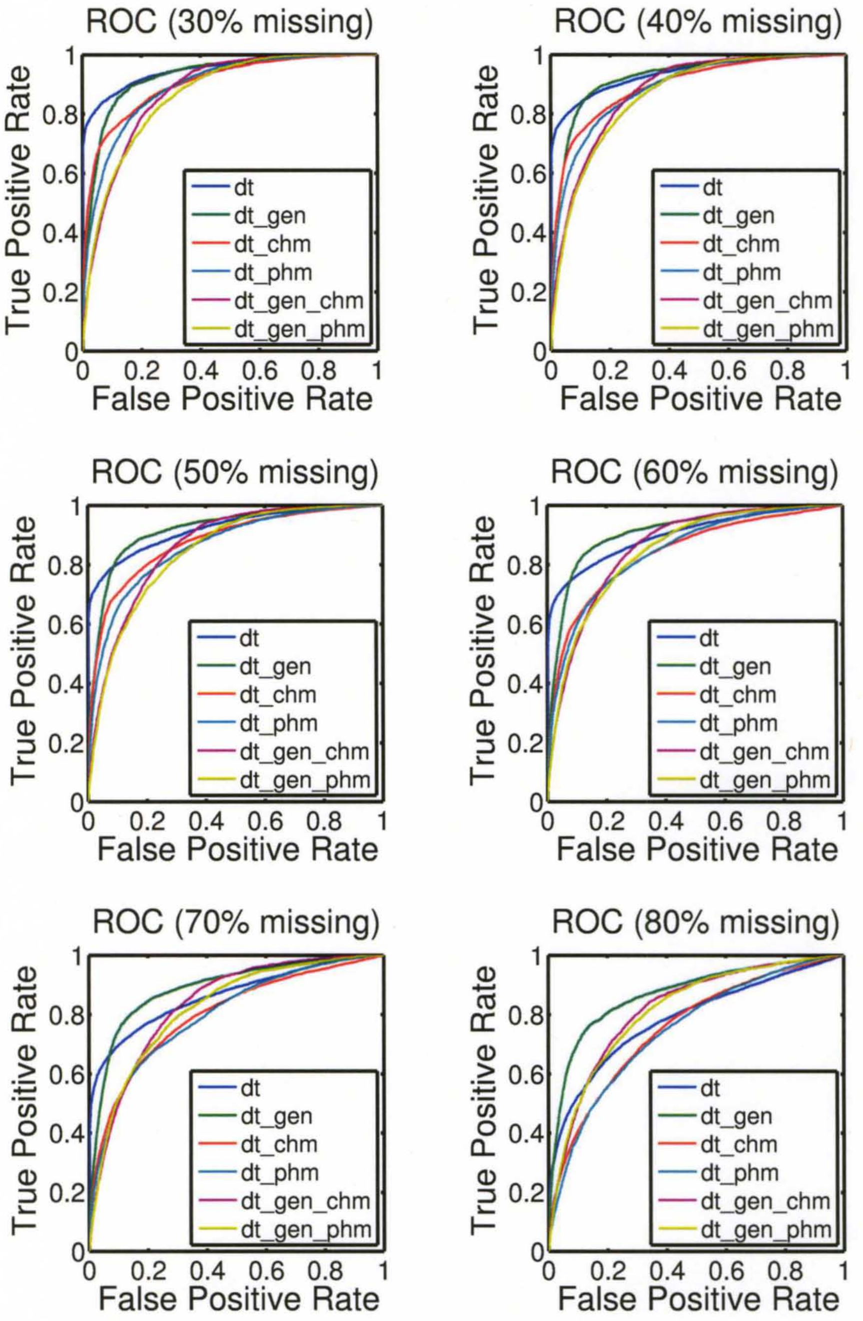


Figure 5.11: Illustration of the prediction performance on a variety of prediction scenarios: (dt) prediction on known drug and known target; (dt_gen) prediction on known drug and unknown protein, through genomic data; (dt_chm / dt_phm) prediction on unknown ligand and known target, through chemical/pharmacological data; (dt_gen_chm / dt_gen_phm) prediction on unknown ligand and unknown target, through genomic data and chemical/pharmacological data. The drug-target interaction data (with a variety of randomly selected missing entries) is used in all scenarios. A 10-times 5-fold cross validation is performed, and the receiver operating characteristic (ROC) curves are plotted.

CHAPTER 6

BAYESIAN HIERARCHICAL CROSS-CLUSTERING

Most clustering algorithms assume that all dimensions of the data can be described by a single structure. Cross-clustering (or multi-view clustering) allows multiple structures, each applying to a subset of the dimensions. We present a novel approach to cross-clustering, based on approximating the solution to a Cross Dirichlet Process mixture (CDPM) model [Shafto et al., 2006, Mansinghka et al., 2009]. Our bottom-up, deterministic approach results in a hierarchical clustering of dimensions, and at each node, a hierarchical clustering of data points. We also present a randomized approximation, based on a truncated hierarchy, that scales linearly in the number of levels. Results on synthetic and real-world data sets demonstrate that the cross-clustering based algorithms perform as well or better than the clustering based algorithms, our deterministic approaches models perform as well as the MCMC-based CDPM, and the randomized approximation provides a remarkable speed-up relative to the full deterministic approximation with minimal cost in predictive error.

6.1 Introduction

Standard approaches to clustering assume that there is a single clustering that describes all of the data. Consider, for example the Dirichlet process mixture (DPM) model, a widely used model for density estimation and for clustering. The model takes as input a set of data points, and their values on a set of dimensions. For each data point, the DPM infers a latent variable indicating an assignment of the data to a mixture component. A fundamental assumption underlying this approach is that all of the dimensions of the data are described from a single view, i.e., the data points were generated by a single underlying DPM.

However, there are many cases in which a single view does not describe all aspects of the data points. In some cases, we might expect some dimensions to be described by one model while others are merely "noise". More generally, any given data set may be generated by multiple different models, each applying to subsets of the observed dimensions. In these contexts, clustering algorithms typically identify a single dominant structure and the dimensions better explained by other models appear to be weakly structured.

Cross-clustering (or multi-view clustering) relaxes the single-DPM assumption, allowing the possibility that a data set may have multiple different views. Consider, for example, a generalization of the DPM, the Cross Dirichlet Process Mixture model (CDPM) [Shafto et al., 2006, Mansinghka et al., 2009]. This model allows that a single data set may be composed of data generated by multiple different DPMs. The model therefore infers, for each dimension, a latent variable indicating an assignment of that dimension to a view, and w.r.t. each view, an assignment of data points to mixture components of DPM. This provides the capability to separate structured features from noisy features and the ability to identify cases where different dimensions of the data are best described by different DPMs. Because the CDPM is a generalization of the DPM, this approach should lead to improved predictive performance on previously unobserved values. However, cross-clustering models admit a very large number of possible latent structures, and their success depends on reliable, efficient inference algorithms.

In this paper, we propose Bayesian Hierarchical Cross-Clustering (BHCC), a deterministic approach to approximate inference for a CDPM. We also propose Randomized BHCC (RBHCC), a much faster alternative approximation to the CDPM. Building off the work by Heller and Ghahramani [2005a], BHCC builds a hierarchical clustering of dimensions, where the posterior probability of merging dimensions in different views is estimated based on the marginal likelihood that the data are generated by a DPM.

6.2 Related Work

In addition to the work by Shafto et al. [2006], Mansinghka et al. [2009], there has been growing interest in the problem of multi-view clustering. Rodriguez et al. [2008] proposed a very similar approach which they call the Nested Dirichlet Process. In terms of other approaches, there are those that allow for two views [Qi and Davidson, 2009, Gondek and Hofmann, 2004, Dang and Bailey, 2010], and those that allow many views. Because we typically do not know how many views there are *a priori*, approaches that allow potentially many views, and infer the correct number for a given data set are more appealing. Cui et al. [2007] use a sequential approach, iteratively clustering in subspaces that are orthogonal to existing solutions. Guan et al. [2010] propose a deterministic, variational approximation to CDPM. Their model differs in that they use a DP prior on categories via the stick-breaking construction. Unlike in their work, our approaches result in hierarchical clusterings of dimensions, which may be desirable in some situations. Additionally, we provide results on real-world prediction problems to provide objective validation for the approach.

6.3 Cross Dirichlet Process Mixture Model

The problem of learning cross-cutting category structure can be approached by generalizing standard category-learning approaches. Shafto et al. [2006] introduced the CDPM (which they called *CrossCat*), a generalization of standard DPMs [Neal, 1998]. The CDPM was formalized by assuming that dimensions are assigned to mixtures via the Chinese Restaurant Process (CRP) [Aldous, 1985].

Let X be an $I \times J$ data matrix, where the ith row $X_{i,\cdot}$ represents data point i and the jth column $X_{\cdot,j}$ represents dimension j. Let $\boldsymbol{u}$ be a vector of latent variables representing the partitioning of dimensions into views, where $u_j = v$ indicates that dimension j is assigned to view v. Let Z be a matrix of latent variables representing the partitioning of data points w.r.t. all views, where $Z_{i,v} = c$ indicates that, in view v, data point i is assigned to component c. The generative model for a CDPM is then,

$$\boldsymbol{u} \sim \mathrm{CRP}(\alpha), \tag{6.1}$$

$$Z_{\cdot,v} \sim \mathrm{CRP}(\alpha), \tag{6.2}$$

$$\Theta_{c,v} \sim H(\delta), \tag{6.3}$$

$$X_{i,\boldsymbol{u}=v} \sim F(X_{i,\boldsymbol{u}=v}|\Theta_{Z_{i,v},v}), \tag{6.4}$$

where α is the concentration hyperparameter of the CRPs (using a single parameter for simplicity), H is the prior distribution over component parameters $\Theta_{c,v}$, F is the component distribution (e.g. F is Binomial distribution and H is Beta distribution), and $\boldsymbol{u} = v$ returns a vector of indices: $(j|u_j = v \text{ for } j = 1, \ldots, J)$. Alternatively, by adopting conjugate models, one may substitute Equation 6.3 and 6.4 with

$$X_{Z_{\cdot,v}=c,\boldsymbol{u}=v} \sim G(X_{Z_{\cdot,v}=c,\boldsymbol{u}=v}|\delta). \tag{6.5}$$

As with the DPMs, analytic inference is intractable, but simple Gibbs sampling algorithms are no longer possible. Because mixing over possible views requires potentially creating new DPMs on data points, special-purpose MCMC algorithms are required (see [Mansinghka et al., 2009]). Developing computationally efficient samplers that mix well is time-consuming and challenging, and it is desirable to have alternatives to sampling-based methods.

6.4 Bayesian Hierarchical Cross Clustering

The BHCC algorithm takes the data matrix X and produces tree T, a hierarchical clustering of the dimensions. Each subtree in Tree T is represented by a 4-tuple $T_c = (c, T_a, T_b, r_c)$ where c is the identification number for the root node of T_c, T_a and T_b are the left and right subtrees of c, and r_c is the posterior probability of merging T_a and T_b to form T_c. Each node in T is associated with a set of dimensions and forms a view w.r.t. which the data are generated from a DPM.

Definition 6.4.1. Define $\mathcal{L}(T)$ as a function returning identification numbers for all the leaf nodes in tree T, i.e., if $T = (c, T_a, T_b, r_c)$, then

$$\mathcal{L}(T) = \begin{cases} \{c\} \text{ if } T_a = T_b = \emptyset \\ \mathcal{L}(T_a) \cup \mathcal{L}(T_b) \text{ otherwise.} \end{cases}$$

BHCC is described in Algorithm 6. The algorithm is initialized with each dimension forming a view by itself: it starts with J trees: $T_j = (j, \emptyset, \emptyset, 1)$ for $j = 1, \ldots, J$. The algorithm proceeds by repeatedly merging the pair of subtrees that, when joined, have the highest probability, and continues until all dimensions are joined in the same view. To estimate the posterior probability of merging trees T_a and T_b, BHCC considers two hypotheses $\mathcal{H}_1^c$ and $\mathcal{H}_2^c$. The null hypothesis $\mathcal{H}_1^c$ states that the set of dimensions $\mathcal{L}(T_a) \cup \mathcal{L}(T_b)$ form one view, i.e., data points in $X_{\cdot, \mathcal{L}(T_a) \cup \mathcal{L}(T_b)}$ were generated by the same DPM,

$$p(X_{\cdot, \mathcal{L}(T_a) \cup \mathcal{L}(T_b)} | \mathcal{H}_1^c) = p(X_{\cdot, \mathcal{L}(T_a) \cup \mathcal{L}(T_b)} | \text{DPM}).$$

We follow [Heller and Ghahramani, 2005a] in approximating the marginal likelihood of the data under a DPM using BHC.

The alternative hypothesis states that the dimensions $\mathcal{L}(T_a) \cup \mathcal{L}(T_b)$ form two or more views, i.e., data points in $X_{\cdot, \mathcal{L}(T_a) \cup \mathcal{L}(T_b)}$ were generated by two or more DPMs. The number of possible ways of dividing n dimensions into two or more dimension clusters is $B_n - 1$ where B_n is the Bell number: $B_n = \sum_{k=0}^{n-1} \binom{n-1}{k} B_k$ and $B_0 = 1$. Thus summing over these possibilities is intractable. BHCC restricts itself to dimension partitionings consistent with the subtrees T_a and T_b (see Definition 6.5.1).

The marginal likelihood of this restricted alternative hypothesis, $\mathcal{H}_2^c$, given the data, is a product over the subtrees:

$$p(X_{\cdot,\mathcal{L}(T_a)\cup\mathcal{L}(T_b)}|\mathcal{H}_2^c) = p(X_{\cdot,\mathcal{L}(T_a)}|T_a)p(X_{\cdot,\mathcal{L}(T_b)}|T_b),$$

where each term on the right-hand side of the equation is a probability of a data under a BHCC tree defined recursively as follows.

Let $T_c = (c, T_a, T_b, r_c)$ be the merged tree. The marginal probability of the data in tree T_c is a weighted sum of the probability of the data under both hypothesis:

$$\begin{aligned} p(X_{\cdot,\mathcal{L}(T_c)}|T_c) = \pi_c p(X_{\cdot,\mathcal{L}(T_c)}|\mathcal{H}_1^c) + \\ (1-\pi_c)p(X_{\cdot,\mathcal{L}(T_a)}|T_a)p(X_{\cdot,\mathcal{L}(T_b)}|T_b), \end{aligned} \tag{6.6}$$

where the weight π_c is the prior probability of this merge, i.e., $\pi_c \overset{\text{def.}}{=} p(\mathcal{H}_1^c)$. Heller and Ghahramani [2005a] proposed a bottom-up method for computing the prior under the CRP:

$$\begin{array}{lll} \pi_c = 1 & d_c = \alpha & \text{if } T_c \text{ is a leaf} \\ \pi_c = \frac{\alpha\Gamma(N_c)}{d_c} & d_c = \alpha\Gamma(N_c) + d_a d_b & \text{otherwise,} \end{array} \tag{6.7}$$

where α is the concentration hyperparameter, $N_c \overset{\text{def.}}{=} |\mathcal{L}(T_c)|$, and $\Gamma(\cdot)$ is the Gamma function.

The posterior probability of the merged hypothesis given the data is computed using Bayes rule,

$$r_c \overset{\text{def.}}{=} p(\mathcal{H}_1^c|X_{\cdot,\mathcal{L}(T_c)}) = \frac{\pi_c p(X_{\cdot,\mathcal{L}(T_c)}|\mathcal{H}_1^c)}{p(X_{\cdot,\mathcal{L}(T_c)}|T_c)}. \tag{6.8}$$

The quantity r_c is used to decide greedily which two trees to merge at the stage of inferring the BHCC tree; it also allows one to define posterior predictive distributions as discussed in Section 6.6.

6.5 Approximate Inference in a Cross Dirichlet Process Mixture Model

In this section, we show that the BHCC algorithm is an approximate inference algorithm for CDPM.

Definition 6.5.1. Define Ptns(T) as a function returning the set of tree-consistent partitionings of

Algorithm 6: Bayesian Hierarchical Cross-Clustering (BHCC) algorithm.

```
input      : An $I \times J$ data matrix $X$
output     : The final merged tree $T$
initialize : Each dimension $j$ forms a view by itself, i.e. $T_j \leftarrow (j, \emptyset, \emptyset, 1)$ for $j = 1, \ldots, J$,
             where the 4-tuple has the format: (root node, left subtree, right subtree,
             posterior of merge). $S \leftarrow \{T_j | j = 1, \ldots, J\}$. The current largest root node id
             $c \leftarrow J$
while $|S| > 1$ do
    $c \leftarrow c + 1$
    Find the pair of $T_a, T_b$ with the highest probability of the merged hypothesis:
        $r_c \leftarrow \frac{\pi_c p(X_{\cdot,\mathcal{L}(T_a)\cup\mathcal{L}(T_b)}|\text{DPM})}{\pi_c p(X_{\cdot,\mathcal{L}(T_a)\cup\mathcal{L}(T_b)}|\text{DPM}) + (1-\pi_c)p_a p_b}$
    where $p_a \leftarrow p(X_{\cdot,\mathcal{L}(T_a)}|T_a)$, and $p_b \leftarrow p(X_{\cdot,\mathcal{L}(T_b)}|T_b)$
    $T_c \leftarrow (c, T_a, T_b, r_c)$, i.e. join trees $T_a$ and $T_b$ to form $T_c$ with root node $c$, and the
    posterior of merge $r_c$
    $S \leftarrow S \cup \{T_c\} - \{T_a, T_b\}$
end
$T \leftarrow S$
```

the set $\mathcal{L}(T)$, i.e., if $T = (c, T_a, T_b, r_c)$, then

$$\text{Ptns}(T) = \begin{cases} (c) \text{ if } T_a = T_b = \emptyset \\ (\mathcal{L}(T)) \cup \text{Ptns}(T_a) \times \text{Ptns}(T_b) \text{ else.} \end{cases}$$

For example, assume a binary tree T with 3 leaf nodes: $T = (6, T_4, T_3, r_6), T_4 = (4, T_1, T_2, r_4), T_1 = (1, \emptyset, \emptyset, 1), T_2 = (2, \emptyset, \emptyset, 1), T_3 = (3, \emptyset, \emptyset, 1)$, then $\text{Ptns}(T) = \{(1,2,3), (1,2)(3), (1)(2)(3)\}$.

Lemma 6.5.2. *Let $\boldsymbol{u}$ be a vector of indicator variables representing a partitioning of N elements, $p(\boldsymbol{u})$ be the probability of $\boldsymbol{u}$ in a Dirichlet-Multinomial model, i.e., $p(\boldsymbol{u}) = \int p(\boldsymbol{u}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\alpha)\, d\boldsymbol{\theta}$ where $p(\boldsymbol{u}|\boldsymbol{\theta})$ is the Multinomial distribution and $p(\boldsymbol{\theta}|\alpha)$ is the Dirichlet distribution, then*

$$p(\boldsymbol{u}) = \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}\, \alpha^{\max(\boldsymbol{u})} \prod_{v=1}^{\max(\boldsymbol{u})} \Gamma(|\boldsymbol{u} = v|),$$

where $\max(\boldsymbol{u})$ *is the number of clusters in partitioning* $\boldsymbol{u}$.

Lemma 6.5.3. *The marginal likelihood of a CDPM is:*

$$p(X_{\cdot,\mathcal{L}(T_c)}|\text{CDPM}) = \sum_{\boldsymbol{u}\in\mathcal{U}} p(\boldsymbol{u}) \prod_{v=1}^{\max(\boldsymbol{u})} p(X_{\cdot,\boldsymbol{u}=v}|\text{DPM}),$$

where $\mathcal{U}$ is the set of all possible partitionings of the dimensions $\mathcal{L}(T_c)$.

Lemma 6.5.2 follows from a standard Dirichlet integral. Lemma 6.5.3 follows from the definition of CDPM. Here the explicit dependence on N, α and δ has been dropped for simplicity.

Definition 6.5.4. Following from Equation 6.6, we define $p(\boldsymbol{u}|T_c)$ as

$$p(\boldsymbol{u}|T_c) = \frac{1}{d_c}\,\alpha^{\max(\boldsymbol{u})} \prod_{v=1}^{\max(\boldsymbol{u})} \Gamma(|\boldsymbol{u} = v|).$$

Theorem 6.5.5. *The quantity in Equation 6.6 computed by the BHCC algorithm is:*

$$\begin{aligned} & p(X_{\cdot,\mathcal{L}(T_c)}|T_c) \\ = & \sum_{\boldsymbol{u}\in \mathrm{Ptns}(T_c)} p(\boldsymbol{u}|T_c) \prod_{v=1}^{\max(\boldsymbol{u})} p(X_{\cdot,\boldsymbol{u}=v}|\mathrm{BHC}) \end{aligned}$$

Theorem 6.5.5 can be proven by induction, starting from the base case where T_c is a leaf node; proceeding to an arbitrary non-leaf node T_c, with inductive hypothesis that the Theorem holds for both subtrees T_a and T_b. The essential techniques for the proof are the same as in Heller and Ghahramani [2005a] and we omit the details here.

Corollary 6.5.6. *For any BHCC tree $T_c = (c, T_a, T_b, r_c)$, the following is a lower bound on the marginal likelihood of a CDPM:*

$$\frac{d_c\Gamma(\alpha)}{\Gamma(N_c + \alpha)} p(X_{\cdot,\mathcal{L}(T_c)}|T_c) \leq p(X_{\cdot,\mathcal{L}(T_c)}|\mathrm{CDPM})$$

recalling that $N_c = |\mathcal{L}(T_c)|$.

Proof. Notice $\mathrm{Ptns}(T_c) \subseteq \mathcal{U}$, and $p(X|\mathrm{DPM}) \geq p(X|\mathrm{BHC})$, based on Lemma 6.5.2-6.5.3, Definition 6.5.4, and Theorem 6.5.5 we have

$$\begin{aligned}
&p(X_{\cdot,\mathcal{L}(T_c)}|\mathrm{CDPM})\\
&\geq \sum_{\boldsymbol{u}\in\mathrm{Ptns}(T_c)} p(\boldsymbol{u}) \prod_{v=1}^{\max(\boldsymbol{u})} p(X_{\cdot,\boldsymbol{u}=v}|\mathrm{DPM})\\
&\geq \sum_{\boldsymbol{u}\in\mathrm{Ptns}(T_c)} p(\boldsymbol{u}) \prod_{v=1}^{\max(\boldsymbol{u})} p(X_{\cdot,\boldsymbol{u}=v}|\mathrm{BHC})\\
&= \frac{d_c\Gamma(\alpha)}{\Gamma(N_c+\alpha)} \sum_{\boldsymbol{u}\in\mathrm{Ptns}(T_c)} p(\boldsymbol{u}|T_c) \prod_{v=1}^{\max(\boldsymbol{u})} p(X_{\cdot,\boldsymbol{u}=v}|\mathrm{BHC})\\
&= \frac{d_c\Gamma(\alpha)}{\Gamma(N_c+\alpha)}\, p(X_{\cdot,\mathcal{L}(T_c)}|T_c)
\end{aligned}$$

□

6.6 Prediction

In this section we define approximate inference for the posterior predictive distribution of a new data point $p(\boldsymbol{x}|X,T)$, a new dimension $p(\boldsymbol{y}|X,T)$, and a missing value $p(X_{i,j}|X,T)$, for any BHCC tree T. Throughout this section, $\boldsymbol{x}$, a length-J row vector, represents a new data point; and $\boldsymbol{y}$, a length-I column vector, represents a new dimension.

To estimate the predictive distribution of new observation ($\boldsymbol{x}, \boldsymbol{y}$ or $X_{i,j}$) given X, the model sums the predictions for each of the possible CDPM hypotheses, weighted by the posterior probability of these hypothesis given the data. Our approach is to approximate these predictions by considering only tree-consistent hypotheses. We exploit the previously-built BHCC tree, and approximate the sum using only tree-consistent hypotheses. We present predictive distributions in Section 6.6.1 with recursive definitions, and we offer inductive proofs in Sec. 6.10.

6.6.1 Predictive Distributions

For any tree $T_c = (c, T_a, T_b, r_c)$, all three types of predictive distribution are approximated recursively by summing over the probability of the new observation ($\boldsymbol{x}$, $\boldsymbol{y}$, or $X_{i,j}$) conditioned on the two hypothesis ($\mathcal{H}_1^c$ and $\mathcal{H}_2^c$) weighted by the posterior probability of the hypothesis given the data:

$$\begin{aligned}&p(\text{new observation}|X_{\cdot,\mathcal{L}(T_c)}, T_c)\\ &\overset{\text{def.}}{=} p(\mathcal{H}_1^c|X_{\cdot,\mathcal{L}(T_c)})p(\text{new observation}|\mathcal{H}_1^c)+\\ &(1-p(\mathcal{H}_1^c|X_{\cdot,\mathcal{L}(T_c)}))p(\text{new observation}|\mathcal{H}_2^c).\end{aligned} \tag{6.9}$$

The base case of the recursion, for which T_c is a leaf node, is also defined by Equation 6.9 with $p(\mathcal{H}_1^c|X_{\cdot,\mathcal{L}(T_c)}) = r_c = 1$ by definition.

To predict a novel data point, note that while recursing down the tree, the new data point $\boldsymbol{x}$ is divided into two *independent* pieces according to the tree partitioning: $\boldsymbol{x}_{\mathcal{L}(T_a)}$ and $\boldsymbol{x}_{\mathcal{L}(T_b)}$. Thus for $\boldsymbol{x}$, Equation 6.9 becomes

$$\begin{aligned}&p(\boldsymbol{x}|X_{\cdot,\mathcal{L}(T_c)}, T_c) = r_c p(\boldsymbol{x}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM}) + (1-r_c)\\ &p(\boldsymbol{x}_{\mathcal{L}(T_a)}|X_{\cdot,\mathcal{L}(T_a)}, T_a)p(\boldsymbol{x}_{\mathcal{L}(T_b)}|X_{\cdot,\mathcal{L}(T_b)}, T_b).\end{aligned} \tag{6.10}$$

When predicting a novel dimension, the new dimension $\boldsymbol{y}$ can be generated by each of the subtrees T_a and T_b. Thus for $\boldsymbol{y}$, Equation 6.9 becomes

$$\begin{aligned}&p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_c)}, T_c) = r_c p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM})+\\ &(1-r_c)\Big(p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_a)}, T_a) + p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_b)}, T_b)\Big).\end{aligned} \tag{6.11}$$

We define $p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})$ as the probability of the dimension $\boldsymbol{y}$ given that $X_{\cdot,\mathcal{L}(T_k)}$ is generated from a DPM w.r.t. the view containing the set of dimensions $\mathcal{L}(T_k)$:

$$\begin{aligned}&p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})\\ &\overset{\text{def.}}{=} \sum_{\boldsymbol{z}\in\mathcal{V}} p(\boldsymbol{z}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})p(\boldsymbol{y}|\boldsymbol{z})\\ &= \sum_{\boldsymbol{z}\in\mathcal{V}} p(\boldsymbol{z}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM}) \prod_{c=1}^{\max(\boldsymbol{z})} p(\boldsymbol{y}_{\boldsymbol{z}=c}),\end{aligned} \tag{6.12}$$

where $\mathcal{V}$ denotes the set of all partitionings of data points in the data, each partitioning $\boldsymbol{z} \in \mathcal{V}$ is represented in the form of a vector of indicator variables. $|\mathcal{V}|$ follows the Bell number and the quantity in Equation 6.12 is intractable. Again, it is approximated by recursing through the BHC tree.

For a missing value, $X_{i,j}$, those views (a.k.a. dimension clusters) in the tree not including dimension j will not contribute to the prediction. Thus Equation 6.9 becomes

$$
\begin{aligned}
& p(X_{i,j}|X_{\cdot,\mathcal{L}(T_c)}, T_c) \overset{\text{def.}}{=} \\
& r_c p(X_{i,j}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM})|j \cap \mathcal{L}(T_c)| + (1 - r_c) \\
& \Big(p(X_{i,j}|X_{\cdot,\mathcal{L}(T_a)}, T_a) + p(X_{i,j}|X_{\cdot,\mathcal{L}(T_b)}, T_b)\Big).
\end{aligned}
\tag{6.13}
$$

6.7 Randomized BHCC

Computational complexity is the primary limitation of the BHCC algorithm, which takes $O(I^2J^2)$ computation time to build the tree given a $I \times J$ data. This section presents one method that can dramatically decrease the complexity, based on a randomized filtering approach [see Heller and Ghahramani, 2005b].

The Randomized BHCC (RBHCC) algorithm is described in Algorithm 7. It takes in a data set X and randomly selects a subset of dimensions V_0 from the whole set of dimensions V. The original BHCC algorithm is run on $X_{\cdot,V_0}$, obtaining a tree T. Based on the priors of the two top subtrees of T, along with the predictive probabilities that a dimensions belongs to the left subtree and the right subtree (defined in Equation 6.11), the remaining dimensions, $V - V_0$ are then filtered individually down the tree.

For RBHCC, the cost is composed of three parts. The upper levels of the tree are constructed using randomized BHCC, which includes recursively running normal BHCC to construct the initial tree and then filtering the remaining dimensions based on the tree. The lower levels (the threshold number B in Algorithm 7 is reached) are built using normal BHCC. The total number of dimension comparisons can be expressed recursively as:

$$
Comp(J) = |V_0|^2 + J + Comp(aJ) + Comp((1-a)J),
$$

recalling that V_0 is the set of dimensions randomly chosen for running BHCC, $0 \leq a \leq 1$ is the proportion of dimensions on one side of the tree. Note that $|V_0|$ can be considerably smaller than J when J is large; Meanwhile, the depth of a balanced binary tree is $\log(J)$. Thus the number of dimension comparisons $comp(J)$ has the computational complexity $O(J\log(J))$. If the DPM w.r.t. the views is approximated by the randomized BHC algorithm, the overall complexity of RBHCC is $O(IJ\log I \log J)$.

Generally, capturing higher level structures is sufficiently informative. Thus we could restrict the

Algorithm 7: Randomized Bayesian Hierarchical Cross-Clustering (RBHCC) algorithm

```
input  : An $I \times J$ data matrix $X$. A threshold number $B$ determining when to stop
         RBHCC
output : The final merged tree $T$
if $J < B$ then return $T \leftarrow$ BHCC($X$)
$V \leftarrow \{1, 2, \ldots, J\}, V_a = V_b = \emptyset$
Pick $V_0 \subset V$ randomly where $|V_0| \ll J$
$T \leftarrow$ BHCC($X_{\cdot, V_0}$), which, as defined in Algorithm 6, returns a 4-tuple $(c, T_a, T_b, r_c)$ with
  $c$ the root node id, $T_a$ and $T_b$ the left and right subtrees of the root, and $r_c$ the posterior
  of merging $T_a$ and $T_b$
foreach $j \in V - V_0$ do
    $\boldsymbol{y} \leftarrow X_{\cdot, j}$
    $p_a \leftarrow p(\boldsymbol{y} | X_{\cdot, \mathcal{L}(T_a)}, T_a)$, $p_b \leftarrow p(\boldsymbol{y} | X_{\cdot, \mathcal{L}(T_b)}, T_b)$
    if $\pi_a p_a > \pi_b p_b$ then
        $V_a \leftarrow V_a \cup \{j\}$
    else
        $V_b \leftarrow V_b \cup \{j\}$
    end
end
$T_a \leftarrow$ RBHCC($X_{\cdot, V_a \cup \mathcal{L}(T_a)}, B$)
$T_b \leftarrow$ RBHCC($X_{\cdot, V_b \cup \mathcal{L}(T_b)}, B$)
$p_a \leftarrow p(X_{\cdot, \mathcal{L}(T_a)} | T_a)$, $p_b \leftarrow p(X_{\cdot, \mathcal{L}(T_b)} | T_b)$,

    $r_c \leftarrow \frac{\pi_c p(X_{\cdot, \{\mathcal{L}(T_a), \mathcal{L}(T_b)\}} | \mathrm{DPM})}{\pi_c p(X_{\cdot, \{\mathcal{L}(T_a), \mathcal{L}(T_b)\}} | \mathrm{DPM}) + (1 - \pi_c) p_a p_b}$

$T \leftarrow (c, T_a, T_b, r_c)$, where $c$ is a node id unique to $T$
```

algorithm to running only the top L levels, either a priori or interactively. With these much smaller dimensions and data points cut-off levels (L for dimensions and K for data points), the truncated RBHCC algorithm is linear, $O(IJLK)$.

6.8 Results

6.8.1 An Illustration

We illustrate the performance of the algorithms using a synthetic dataset. We generated a Binomial dataset with 100 data points and 200 dimensions. The data set has four views of data point clusterings. i.e. a CDPM with four DPMs each w.r.t. a subset of the original 200 dimensions. The number of dimensions within Views 1-4 are 30, 50, 50 and 70 respectively. The number of Binomial mixture components [1] under Views 1-4 are 4, 5, 6 and 8 respectively.

[1] Let $\boldsymbol{x}_k$ be a data point in the kth component, and $\boldsymbol{\theta}_k$ be the mean of the kth component. Then $\boldsymbol{x}_k \sim \mathrm{Binomial}(\boldsymbol{N}, \boldsymbol{\theta}_k)$ and $\boldsymbol{\theta}_k \sim \mathrm{Beta}(\boldsymbol{\alpha}, \boldsymbol{\beta})$. We set $\boldsymbol{N} = \mathbf{50}, \boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbf{0.5}$.

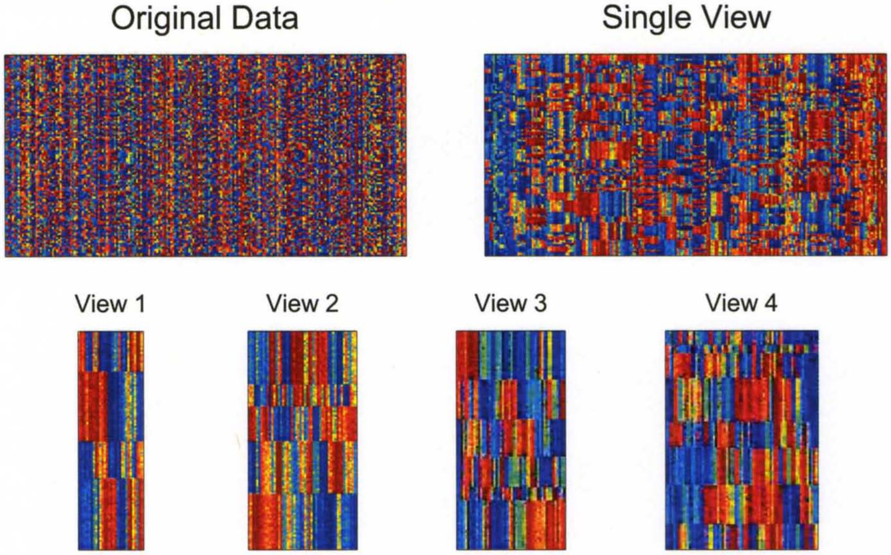


Figure 6.1: Comparison of the clustering(s) between BHC and BHCC. The data matrix and the results from BHC (Single View) and BHCC (View 1-4) are shown in heatmaps, where each row denotes a data point and each column denotes a dimension. Note that the number of views and the partitioning of dimensions are unknown *a priori* to BHCC and are inferred from the data.

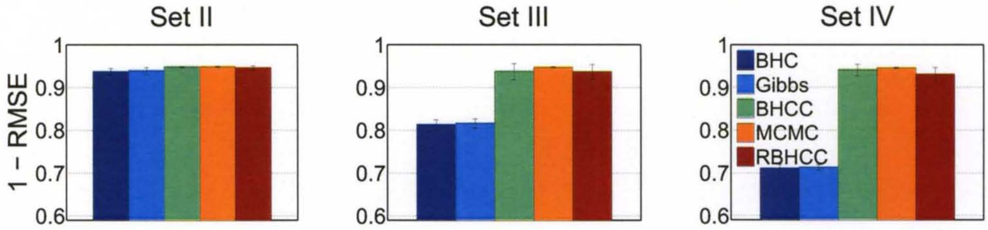


Figure 6.2: Comparison of predictive performance among DPM approximations (BHC and Gibbs sampling) and CDPM approximations (BHCC, MCMC and RBHCC) on the same synthetic datasets. Set II has 2 views; Set III has 3 views and Set IV has 4 views. Each DPM w.r.t. its view contains four well separated Binomial mixture components.

We applied BHC and BHCC on the data. Figure 6.1 shows the original data, the result from BHC (Single View) and the results from BHCC (Views 1 - 4). Heatmaps are used to display the data matrix and the results where each row denotes a data point and each column denotes a dimension. For BHC and BHCC, the data points are rearranged according to the inferred hierarchical tree over data points to reflect the clustering.

Note that in BHCC, the partitioning of dimensional space is represented by an inferred hierarchical tree over dimensions. It yields four dimension subsets when choosing 0.5 as the threshold of posterior probability of merging. In each view, clear category structure is evident by the horizontal striations.

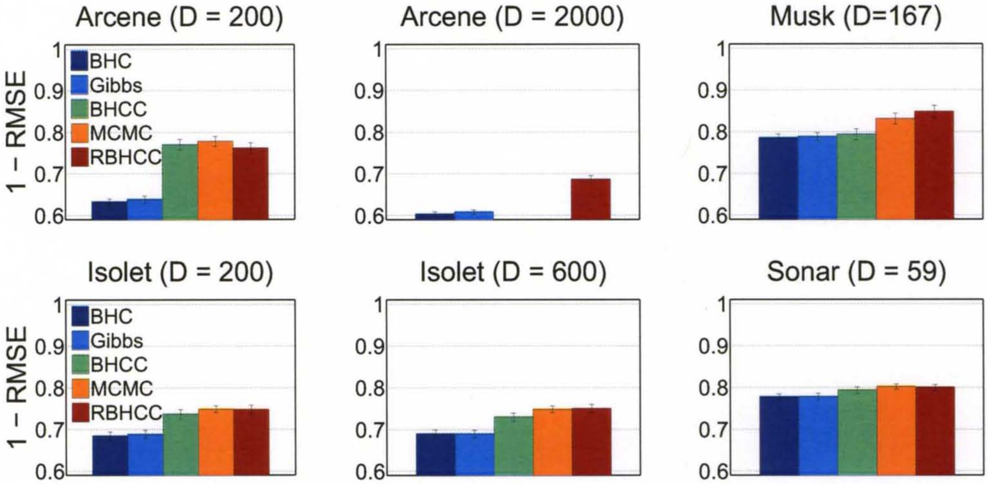


Figure 6.3: Comparison of predictive performance among DPM approximations (BHC and Gibbs sampling) and CDPM approximations (BHCC, MCMC and RBHCC) on the same real datasets. The 4 real datasets are Arcene, Isolet, Musk and Sonar. The number of dimensions used for the test is D (e.g. D = 200) and the number of data points chosen is fixed to 100. For test Arcene (D=2000), the results from BHCC and MCMC are not available because of long running time.

6.8.2 Comparison of Predictive Performance

We compared the predictive performance of BHCC and RBHCC to those from Markov Chain Monte Carlo (MCMC) for CDPM, and two inference methods for DPM, i.e. the BHC approximation and collapsed Gibbs sampling [Neal, 1998]. We first compared these algorithms on three synthetic datasets. We then compared them on four real-world datasets. In each experiment, a one time 10-fold cross-validation is performed on predicting missing values, i.e., the set of entries in the data matrix is randomly partitioned into 10 subsets, of which each subset is hold out once as the validation data and the remaining 9 subsets are used as training data.

The number of filtering levels for RBHCC is set to $L = 6$. In each experiment, the collapsed Gibbs sampling for DPM was performed for 100 iterations [2], the first half was used for burn-in and every fifth sample from the second half was used to estimate posterior quantities. Each iteration consisted of sampling all the latent indicator variables associating mixture components with data points. To facilitate mixing, we interleaved split-merge proposals [Jain and Neal, 2004] between Gibbs sweeps. In each experiment, MCMC for CDPM was performed for 100 iterations, first half of which were used for burn-in and every fifth of the second half for estimating posterior quantities. To improve the mixing, Metropolis coupled MCMC [Geyer, 1991] was performed in which two chains were run, one of which was "heated" by raising the posterior probability to a power 0.8. Due to

[2] Tests showed that increasing the number of iterations did not lead to better predictive accuracy.

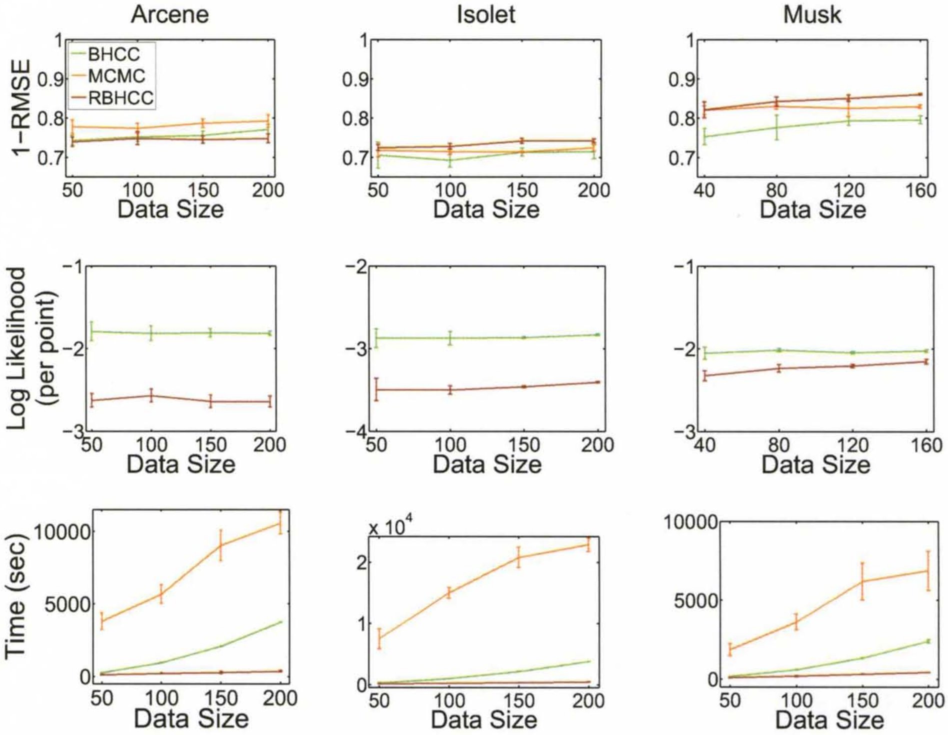


Figure 6.4: Comparison of the prediction accuracy, the estimated marginal likelihood lower bound (not applicable to MCMC), and the runtime among RBHCC, BHCC and MCMC. X-axis represents the number of dimensions used. The number of data points is fixed to 100.

space limitations, we do not review the MCMC for CDPM and the collapsed Gibbs sampling for DPM here.

We generated three sets of 200-dimensional Binomial datasets, each with 100 data points. Set II has two views; Set III has three views and Set IV has four views. Each view contains a DPM with four well separated Binomial mixture components. Figure 6.2 shows the prediction results. We found that the predictive accuracy of inference methods for DPM decreased dramatically as the number of views in the data increased, while methods for CDPM maintained a consistently high accuracy regardless of the number of views. Meanwhile, RBHCC, BHCC and MCMC yield equally high accuracy.

The real datasets used were the Arcene data (900 data points, 10000 dimensions) from NIPS 2003 feature selection challenge, Isolet (7797 data points, 617 dimensions), Musk Version 1 (476 data points, 168 dimensions), and Sonar mines vs. rocks (208 data points, 60 dimensions) data. All data are from the UCI repository [Asuncion and Newman, 2007]. In each experiment, we chose a subset of 100 data points and varied the number of dimensions between 60 and 2000. Figure 6.3 shows the results. The performance of each algorithm varied across datasets. Comparing

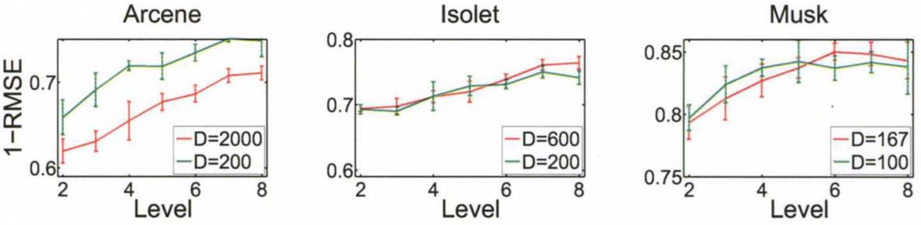


Figure 6.5: Comparison of prediction accuracy among RBHCC with different filtering levels. The number of dimensions used for the test is D (e.g. $D = 2000$) and the number of data points are fixed to 100.

the predictive accuracy between inference algorithms for CDPM and for DPM, we found the former gave a substantial improvement on the Arcene and the Isolet data and a slight improvement on the Musk and the Sonar data. Meanwhile, RBHCC, BHCC and MCMC yield generally equal accuracy.

6.8.3 Prediction, Estimated Marginal Likelihood, and Runtime Comparisons among CDPM Inference Algorithms

To investigate the relationship between speed and accuracy among the CDPM algorithms, we contrasted prediction accuracy, the estimated log marginal likelihood (per point, not applicable to MCMC) and the runtime among RBHCC, BHCC and MCMC on the real datasets of varying sizes. To do so, we selected a subset of the total dimensions, varying the number of dimensions between 40 and 200 (see Figure 6.4). Although the estimated marginal likelihood lower bounds differ for RBHCC and BHCC, the algorithms yield quite similar prediction accuracy. The runtime of MCMC depends on the structure (i.e. number of views and DPMs) underlying the data, which leads to quite variable runtime. Furthermore, we note that, consistent with our complexity analyses, RBHCC offers a significant speedup compared to BHCC: RBHCC scales linearly w.r.t. the number of dimensions in the data, while BHCC is quadratic.

6.8.4 Varying the Filtering Levels for RBHCC

To investigate the tradeoff between runtime and prediction accuracy in the RBHCC, we varied the number of filtering levels between 2 and 8 for RBHCC and ran it on dataset Arcene, Isolet and Musk datasets. Figure 6.5 shows the prediction accuracy. For all tests, the accuracy increases as the level increases. Further, the gain in accuracy gets smaller as L reaches to a certain level, indicating the performance reaches a potential upper bound and it is not necessary to continue increasing L.

6.9 Conclusions

We described Bayesian Hierarchical Cross-Clustering (BHCC), a novel approach for approximate inference of multi-view data. The algorithm provides a deterministic, agglomerative, approximate approach to inference in a Cross Dirichlet Process Mixture (CDPM) model. We have also introduced a fast, randomized algorithm (RBHCC) that scales linearly in the number of levels of the hierarchy. We have contrasted predictive performance on synthetic and real-world data, with clustering models that adopt a single view of the data, the DPM with Gibbs sampling-based inference and Bayesian Hierarchical Clustering, and cross-clustering models that adopt multiple views of the data, the CDPM with MCMC-based inference, BHCC, and RBHCC. Our results show that algorithms based on inferring multiple views have greater predictive accuracy, that the deterministic approaches perform comparably to MCMC-based inference, and the RBHCC provides a remarkable speed-up relative to BHCC, with little cost in predictive accuracy. We consider these results promising, and future work will explore applications of the model to other real-world data sets, and extensions to more richly structured models.

6.10 Inductive Proofs of Predictive Distributions

We begin with some preliminary definitions, then proceed to prove the consistency of the predictive distributions.

Definition 6.10.1. Define Nodes(T) as a function returning the identification numbers of all the nodes in tree T, i.e., if $T = (c, T_a, T_b, r_c)$, then

$$\text{Nodes}(T) = \begin{cases} \emptyset \text{ if } T = \emptyset, \\ \{c\} \cup \text{Nodes}(T_a) \cup \text{Nodes}(T_b) \text{ else.} \end{cases}$$

Definition 6.10.2. Define Parent(T, k) as a function returning identification number of the immediate parent of node k in a BHCC tree T, i.e.,

$$\text{Parent}(T, k) =$$
$$\begin{cases} \emptyset \text{ if } k \text{ is root of } T, \\ \{c | T_c = (c, T_k, T_b, r_c) \vee T_c = (c, T_a, T_k, r_c)\} \text{ else.} \end{cases}$$

Definition 6.10.3. Define Path(T, k) as a function returning identification numbers of all the nodes in the path from the root to node k in a BHCC tree T, i.e.,

$$\mathrm{Path}(T,k) = \begin{cases} \emptyset \text{ if } \{k\} \cap \mathrm{Nodes}(T) = \emptyset, \\ \{k\} \cup \mathrm{Path}(T, \mathrm{Parent}(T,k)) \text{ else.} \end{cases}$$

Definition 6.10.4. Define $\omega(T,k)$ as the posterior probability that $\mathcal{L}(T_k)$ forms a view and merging other subtrees to T_k does not yield a view:

$$\omega(T,k) = r_k \prod_{c \in \mathrm{Path}(T,k) - \{k\}} (1 - r_c).$$

Lemma 6.10.5. *The quantity defined in Equation 6.10 has a lower bound:*

$$\sum_{\boldsymbol{u} \in \mathrm{Ptns}(T_c)} \prod_{v=1}^{\max(\boldsymbol{u})} \omega(T_c, k) p(\boldsymbol{x}_{\boldsymbol{u}=v} | X_{.,\boldsymbol{u}=v}, \mathrm{DPM}), \tag{6.14}$$

where the partitioning $\boldsymbol{u}$ *is represented in the form of a vector of indicator variables,* $\max(\boldsymbol{u})$ *is the number of clusters in partitioning* $\boldsymbol{u}$*, and* k *is the node in* T_c *such that the two vectors* $\boldsymbol{u} = v$ *and* $\mathcal{L}(T_k)$ *have the same set of dimensions.*

Proof. We show a proof by induction. If c is the leaf node, $\mathrm{Ptns}(T_c) = (c)$, $\max(\boldsymbol{u}) = 1$, $k = c$ and $\omega(T_c, c) = r_c = 1$, thus Equation 6.14 becomes $p(\boldsymbol{x}|X_{.,\mathcal{L}(T_c)}, \mathrm{DPM})$ which is equal to the quantity in Equation 6.10. Thus the lemma is true in the base case.

Our inductive hypothesis is that the lemma holds for the two subtrees T_a and T_b. That is,

$$p(\boldsymbol{x}_{\mathcal{L}(T_a)} | X_{\mathcal{L}(T_a)}, T_a) \geq \sum_{\boldsymbol{u}' \in \mathrm{Ptns}(T_a)} \prod_{v'=1}^{\max(\boldsymbol{u}')} \omega(T_a, k') p(\boldsymbol{x}_{\boldsymbol{u}'=v'} | X_{.,\boldsymbol{u}'=v'}, \mathrm{DPM})$$

and same for $p(\boldsymbol{x}_{\mathcal{L}(T_b)} | X_{\mathcal{L}(T_b)}, T_b)$; Also note that

$$\omega(T_a, k') \geq (1 - r_c)\omega(T_a, k') = \omega(T_c, k'),$$

and same for $\omega(T_b, k'')$. Therefore

$$\begin{aligned}
&(1-r_c)p(\boldsymbol{x}_{\mathcal{L}(T_a)}|X_{\mathcal{L}(T_a)}, T_a)p(\boldsymbol{x}_{\mathcal{L}(T_b)}|X_{\mathcal{L}(T_b)}, T_b) \geq \\
&\sum_{\boldsymbol{u}'\in\text{Ptns}(T_a)} \prod_{v'=1}^{\max(\boldsymbol{u}')} \omega(T_c, k')p(\boldsymbol{x}_{\boldsymbol{u}'=v'}|X_{\cdot,\boldsymbol{u}'=v'}, \text{DPM}) \times \\
&\sum_{\boldsymbol{u}''\in\text{Ptns}(T_b)} \prod_{v''=1}^{\max(\boldsymbol{u}'')} \omega(T_c, k'')p(\boldsymbol{x}_{\boldsymbol{u}''=v''}|X_{\cdot,\boldsymbol{u}''=v''}, \text{DPM}) \\
&= \sum_{\boldsymbol{u}\in\text{Ptns}(T_a)\times\text{Ptns}(T_b)} \prod_{v=1}^{\max(\boldsymbol{u})} \omega(T_c, k)p(\boldsymbol{x}_{\boldsymbol{u}=v}|X_{\cdot,\boldsymbol{u}=v}, \text{DPM})
\end{aligned} \tag{6.15}$$

Meanwhile, for the trivial partitioning $(\mathcal{L}(T_c))$ (recalling that $(\mathcal{L}(T_c))$ represents the partitioning where all dimensions in $\mathcal{L}(T_c)$ are assigned to the same cluster), we have $\max(\boldsymbol{u}) = 1$, $k = c$ and $\omega(T_c, c) = r_c$. Thus for $\boldsymbol{u} = (\mathcal{L}(T_c))$

$$\begin{aligned}
&\prod_{v=1}^{\max(\boldsymbol{u})} \omega(T_c, k)p(\boldsymbol{x}_{\boldsymbol{u}=v}|X_{\cdot,\boldsymbol{u}=v}, \text{DPM}) \\
&= r_c p(\boldsymbol{x}_{\mathcal{L}(T_c)}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM})
\end{aligned} \tag{6.16}$$

By definition, $\text{Ptns}(T_c) = (\mathcal{L}(T_c) \cup \text{Ptns}(T_a) \times \text{Ptns}(T_b)$. Therefore combining the results from Equation 6.15 and 6.16, we see the lemma is true. □

Lemma 6.10.6. *The quantity defined in Equation 6.11 is equal to the quantity:*

$$\sum_{k\in\text{Nodes}(T_c)} \omega(T_c, k)p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM}) \tag{6.17}$$

which sums over the prediction w.r.t. all the nodes in T_c weighted by the posterior of the nodes.

Proof. We show a proof by induction. If c is a leaf node, then $T_a = T_b = \emptyset$ and $r_c = 1$. By definition, $p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_c)}, T_c) = p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM})$; Meanwhile, $\text{Nodes}(T_c) = \{c\}$, $\omega(T_c, c) = r_c = 1$, $\mathcal{L}(T_c) = \{c\}$, thus Equation 6.17 becomes $p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM})$. Thus the lemma is true in the base case.

Our inductive hypothesis is that Equation 6.17 holds for the two subtrees T_a and T_b, i.e.,

$$p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_a)}, T_a) = \sum_{k\in\text{Nodes}(T_a)} \omega(T_a, k)p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})$$

and same for T_b. Meanwhile, by definition, $(1 - r_c)\omega(T_a, k) = \omega(T_c, k)$ and same for T_b; and

$r_c = \omega(T_c, c)$. Therefore,

$$\begin{aligned}
&p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_c)}, T_c) = \omega(T_c, c)p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM})+\\
&\sum_{k\in\text{Nodes}(T_a)} \omega(T_c, k)p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})+\\
&\sum_{k\in\text{Nodes}(T_b)} \omega(T_c, k)p(\boldsymbol{y}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})
\end{aligned}$$

Further notice $\text{Nodes}(T_c) = \{c\} \cup \text{Nodes}(T_a) \cup \text{Nodes}(T_b)$, thus the lemma is true. □

Lemma 6.10.7. *The quantity defined in Equation 6.13 is equal to the quantity:*

$$\sum_{k\in\text{Path}(T_c,j)} \omega(T_c, k)p(X_{i,j}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM}) \tag{6.18}$$

which sums over the prediction w.r.t. the nodes on the path from root to dimension j, weighted by the posterior of the nodes.

Proof. We show a proof by induction. If c is a leaf node, $\text{Path}(T_c, j) = \{c\} \cap \{j\}$, $\omega(T_c, c) = r_c = 1$, $\mathcal{L}(T_c) = \{c\}$, thus Equation 6.18 becomes $p(X_{i,j}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM}) \times |\{c\} \cap \{j\}|$ which is also the case in Equation 6.13 ($T_a = T_b = \emptyset$ and $r_c = 1$). Thus the lemma is true in the base case.

Our inductive hypothesis is that Equation 6.18 holds for the two subtrees T_a and T_b, i.e.,

$$\begin{aligned}
&p(X_{i,j}|X_{\cdot,\mathcal{L}(T_a)}, T_a) =\\
&\sum_{k\in\text{Path}(T_a,j)} \omega(T_a, k)p(X_{i,j}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})
\end{aligned}$$

and same for T_b. Meanwhile, by definition, $(1 - r_c)\omega(T_a, k) = \omega(T_c, k)$ and same for T_b; and $r_c = \omega(T_c, c)$. Therefore,

$$\begin{aligned}
&p(X_{i,j}|X_{\cdot,\mathcal{L}(T_c)}, T_c) =\\
&\omega(T_c, c)p(X_{i,j}|X_{\cdot,\mathcal{L}(T_c)}, \text{DPM}) \times |\{j\} \cap \mathcal{L}(T_c)|+\\
&\sum_{k\in\text{Path}(T_a,j)} \omega(T_c, k)p(X_{i,j}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})+\\
&\sum_{k\in\text{Path}(T_b,j)} \omega(T_c, k)p(X_{i,j}|X_{\cdot,\mathcal{L}(T_k)}, \text{DPM})
\end{aligned}$$

Assume j is a leaf node of subtree T_a, then $\text{Path}(T_b, j) = \emptyset$, $\text{Path}(T_c, j) = \{c\} \cup \text{Path}(T_a, j)$ and $|\{j\} \cap \mathcal{L}(T_c)| = 1$, thus the lemma is true. □

CHAPTER 7

BAYESIAN CONGRUENCE MEASURING IN PHYLOGENOMICS

7.1 Introduction

The availability of genome-scale data provides unprecedented opportunities for phylogenetic analyses (phylogenomics). However, molecular phylogenies inferred from individual loci may conflict with each other (incongruence). The incongruence between genes can be the result of random and systematic errors in phylogenetic tree reconstruction, but can also be caused by the underlying biological processes, including population genetic processes [Hartl and Clark], within-species genetic recombination (e.g., chromosomes crossover and gene conversion) [Meselson and Radding, 1975] and horizontal gene transfer [Jain et al., 1999].

Techniques for assessing the significance of phylogenetic incongruence are particularly important to systematic biology on a genome-scale. Due to various heterogeneities caused by the biological processes, however, measuring phylogenetic incongruence has been a statistically and computationally challenging task. Nevertheless, several methods have been proposed (Planet [2006] provides an excellent review). An intuitive framework for measuring incongruence is the incongruence length difference (ILD) test [Farris et al., 1994], initially developed in a parsimony context, and later adapted to a distance-based method [Zelwer and Daubin, 2004]. The test statistic is defined by $d = L_C - \sum_{i=1}^{N} L_i$ where L_i and L_C denote the lengths of the most parsimonious trees calculated for the ith individual loci and for the combined loci, respectively. However, studies have suggested that the test performs poorly when a substantial rate or pattern heterogeneity exists among sites [Dolphin et al., 2000, Darlu and Lecointre, 2002].

In a maximum likelihood context, Huelsenbeck and Bull [1996] described a method based on a likelihood ratio test with the ratio $d = L_1/L_0$ where L_0 is the maximum likelihood assuming that all the genes share identical trees while allowing rate heterogeneity to vary across sites, and L_1 is the maximum likelihood assuming that all the genes have different trees and different evolutionary rates. The null distribution for the test is calculated using the bootstrapping resampling technique. Based on hierarchical clustering and the likelihood ratio test, Leigh et al. [2008] described a method to identify congruent subsets of genes. However, there are several concerns with a maximum-likelihood and bootstrap based approach. To calculate P-values using nonparametric bootstrap, the maximum likelihood estimation must be repeated typically 100 to 1000 times. It therefore can be prohibitively slow [Larget and Simon, 1999]. In addition, the empirical test of Hillis and Bull [1993] suggested that the bootstrap proportion varied too much among replicate data sets to be used as a measure of repeatability.

Bayesian approaches typically model uncertainty in a more interpretable style than maximum likelihood approaches. Although Bayesian analyses have been successfully applied to estimate phylogeny, to our knowledge, very few of these works can explicitly test incongruence between genes or identify congruent gene subsets. Most of these analyses assumed that all genes evolved under the same phylogenetic tree [Larget and Simon, 1999, Huelsenbeck and Ronquist, 2001, Pagel and Meade, 2004, Lartillot and Philippe, 2004]. Suchard et al. [2003] proposed a Bayesian hierarchical model which allowed partitions to have different trees. However, it did not explicitly measure the degree of incongruence among genes. At the same time, it assumed that partitions were known in advance and thus failed in identifying congruent gene subsets. Ané et al. [2007] analyzed each gene separately using Bayesian analysis and constructed a gene-to-tree map which is, in turn, used to estimate the posterior probability of pairwise gene dissimilarity. A drawback of this method is that gene trees, exclusively inferred separately, may not resolve well.

7.2 Methods

The analysis begins with aligned molecular sequence data Y over N loci, primarily DNA or protein sequences. Data $\mathsf{Y} = (Y_1, \dots, Y_N)$ consists of N disjoint alignments with Y_n $(n = 1, \dots, N)$ corresponding to loci n. Data $\mathsf{Y}_k = (Y_{k1}, \dots, Y_{kN_k})$ denotes a subset of Y $(\mathsf{Y}_k \subseteq \mathsf{Y})$ consisting of N_k disjoint alignments, where each Y_{kg} $(g = 1, \dots, N_k)$ refers to some Y_n $(n = 1, \dots, N)$.

The null hypothesis, denoted H_0, states that the interesting alignments are congruent. The alternative hypothesis, denoted H_1, states that at least some part of the interesting alignments are incongruent to the others. According to Bayes' theorem, the posterior probability of all the N_k

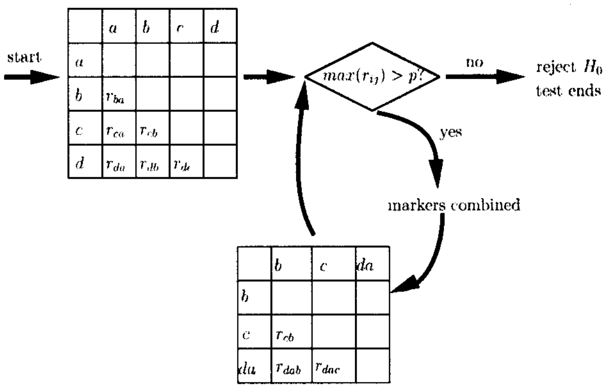


Figure 7.1: Hierarchical clustering algorithm using posterior probability of gene clusters being congruent as merging criteria. a, b, c, d denote markers.

markers in Y_k being congruent given the alignment is

$$p(H_0|\mathsf{Y}_k) = \frac{\pi_k p(\mathsf{Y}_k|H_0)}{\pi_k p(\mathsf{Y}_k|H_0) + (1 - \pi_k)p(\mathsf{Y}_k|H_1)} \tag{7.1}$$

where π_k is the prior probability of all the N_k markers being congruent. The larger $p(H_0|\mathsf{Y}_k)$ is, the more confidence we have in H_0 to believe that the N_k markers are congruent.

The algorithm starts with measuring the degree of congruence for all pairs of loci, and the pair with the highest posterior probability (denoted r) is selected. The value r is compared with a threshold p, ($p = .5$ in this work), if $r > p$, the test continues, treating this pair as a congruent gene cluster consisting of two genes. If $r \leq p$, none of the pairs are congruent and the test ends. The algorithm is shown in Figure 7.1, where congruent information between pair of genes or gene clusters are represented. In Section 7.2.1, the formal definition of topological congruence and branch length congruence are described. In Section 7.2.2, a greedy algorithm is proposed to estimate the likelihood quantities involved in the evaluation of the posterior probability defined in Equation 7.1.

7.2.1 Likelihood of Congruence

For an aligned set of sequences $\mathsf{Y}_k = (Y_{k1}, \ldots, Y_{kN_k})$ over N_k loci, topological congruence defines all N_k genes as having identical evolution topology but with various branch lengths and substitution processes. Thus the marginal likelihood that the N_k markers are *topologically congruent* given alignments Y_k is

$$p(\mathsf{Y}_k|H_0) = \int \prod_{g=1}^{N_k} p(Y_{kg}|\tau_k, \beta_{kg}, \Theta_{kg}) p(\tau_k, \beta_{kg}, \Theta_{kg}) \\ d\,\tau_k\, \beta_{kg}\, \Theta_{kg} \tag{7.2}$$

where τ_k is the topology shared by these N_k genes, β_{kg} is the branch length of sequence Y_{kg}, and Θ_{kg} is the substitution model of sequence Y_{kg}.

For branch-length congruence, all N_k genes have identical branch lengths in addition to identical topology, so the marginal likelihood that these N_k loci are *branch-length congruent* given the alignments Y_k is

$$p(\mathsf{Y}_k|H_0) = \int \prod_{g=1}^{N_k} p(Y_{kg}|\tau_k, \beta_k, \Theta_{kg}) p(\tau_k, \beta_k, \Theta_{kg}) \\ d\,\tau_k\, \beta_k\, \Theta_{kg}$$

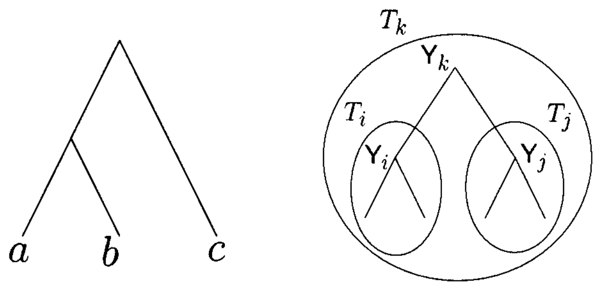
The rest of the chapter focuses on topological congruence. However, the same algorithm can be applied to branch-length congruence with minor modifications. In Section 7.2.3, the form of the likelihood function given a *single* gene and the strategies on prior distribution are presented. The evaluation of the marginal likelihood (Equation 7.2) is discussed in Section 7.2.4.

7.2.2 Likelihood of Incongruence

A main difficulty when evaluating the marginal likelihood of incongruence comes from the hypothesis' combinatorial nature. For example, assume we have three markers (a, b, c). Hypothesis H_1, stating that at least some of the markers are incongruent given the alignments, allows four possibilities: $\{a|b|c,\ ab|c,\ ac|b,\ a|bc\}$, where symbol | separates incongruent markers from congruent markers. Thus the marginal likelihood of H_1 given sequence alignments Y_a, Y_b, Y_c is

$$\begin{aligned} p(Y_a, Y_b, Y_c|H_1) &= w_1 p(Y_a|H_0)p(Y_b|H_0)p(Y_c|H_0) \\ &+ w_2 p(Y_a, Y_b|H_0)p(Y_c|H_0) + w_3 p(Y_a, Y_c|H_0)p(Y_b|H_0) \\ &+ w_4 p(Y_b, Y_c|H_0)p(Y_a|H_0) \end{aligned}$$

where w_i $(i = 1, \ldots, 4)$ is the weight for each case and $\sum_{i=1}^{4} w_i = 1$. A brute force estimation of $p(\mathsf{Y}_k|H_1)$ requires enumerating all possible incongruent clusterings over these N_k markers. Notice



(a) (b)

Figure 7.2: (a) An example tree with three genes. Tree-consistent partitions are $a|b|c$ and $ab|c$. (b) A portion of a tree showing T_i and T_j are merged into T_k.

that the number of possible clusterings over n elements is the nth Bell number: $B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$, ($B_0 = 1$), which prohibits the use of the brute force approach. Instead, we follow the approximation approach developed by Heller and Ghahramani [2005a] and restrict to clusterings that partition the genes in a manner consistent with the subtrees of the merging algorithm described in Figure 7.1. For example, if three genes a, b, c are merged according to Figure 7.2(a), then we only consider two clusterings: $\{a|b|c,\ ab|c\}$. So,

$$
\begin{aligned}
p(Y_a, Y_b, Y_c|H_1) \approx \{&\pi p(Y_a, Y_b|H_0) + \\
&(1-\pi)p(Y_a|H_0)p(Y_b|H_0)\}p(Y_c|H_0)
\end{aligned}
$$

More generally, assume gene cluster k is merged from two mutually exclusive subsets of genes i and j. That is, $\mathsf{Y}_k = \mathsf{Y}_i \cup \mathsf{Y}_j$ and $\mathsf{Y}_i \cap \mathsf{Y}_j = \emptyset$. Equipped with the restricted hypothesis, which we denote $\tilde{H}_1$, the likelihood of incongruence is

$$p(\mathsf{Y}_k|H_1) \approx p(\mathsf{Y}_k|T_k, \tilde{H}_1) = p(\mathsf{Y}_i|T_i)p(\mathsf{Y}_j|T_j) \tag{7.3}$$

and

$$p(\mathsf{Y}_k|T_k) = \pi_k p(\mathsf{Y}_k|H_0) + (1-\pi_k)p(\mathsf{Y}_k|T_k, \tilde{H}_1) \tag{7.4}$$

where T_i, T_j, T_k are binary trees expressing the merging processes as shown in Figure 7.2(b). Restricting to tree-consistent clusterings and assigning different prior probability to them, the method provides a reasonable approximation to the brute force approach which averages over all possible clusterings.

7.2.3 Likelihood Function and Priors

All sites from an individual gene sequence (e.g., an aligned sequence Y_{kg}) are assumed to evolve under identical topology. Assuming the same substitution rate across sites, however, can be unrealistic. A more nuanced model would allow using one set of substitution parameters for each site. This, however, results in too many parameters to estimate given a limited number of observations. A more practical approach is to model the rate variation using a probabilistic distribution. We use the discrete-gamma model [Yang, 2006].

In the discrete-gamma model, a finite mixture model is used to model across-site rate heterogeneity. All sites within a gene are assumed to share a substitution pattern (based composition or transition-transversion rate), but fall into several classes with different rates. Thus, a site with rate r_c and pattern Q has the substitution-rate matrix r_cQ, with r_c calculated using a gamma function. As it is not known to which rate class each site belongs, we average over all the site classes. Incorporating this into the likelihood function, given a sequence alignment Y_{kg} of gene kg, we have

$$\begin{aligned} & p(Y_{kg}|\tau_k, \beta_{kg}, \Theta_{kg}) \\ & = \prod_{s=1}^{S_{kg}} \sum_{c=1}^{C} p(Y_{kgs}|\tau_k, \beta_{kg}, r_cQ_{kg})p(r_c) \end{aligned} \tag{7.5}$$

where Y_{kgs} denotes the sth site in sequence Y_{kg}, S_{kg} is the number of sites in Y_{kg}, and Q_{kg} is the substitution pattern shared by all sites within Y_{kg}. The summation is a weighted average over all C site-rate classes. $p(r_c)$ is the prior probability that a site's rate falls in rate class c. For equally likely rate classes, $p(r_c) = 1/C$.

For the general time-reversible (GTR) model of nucleotide substitution, the matrix is normally written as the product of a symmetric matrix R representing substitution rate, and a diagonal matrix Π representing a stationary distribution:

$$Q_{kg}^{\text{GTR}} = R_{kg}\Pi_{kg} =$$
$$\begin{pmatrix} \cdot & a_{kg}\pi_{kgC} & b_{kg}\pi_{kgA} & c_{kg}\pi_{kgG} \\ a_{kg}\pi_{kgT} & \cdot & d_{kg}\pi_{kgA} & e_{kg}\pi_{kgsG} \\ a_{kg}\pi_{kgT} & d_{kg}\pi_{kgC} & \cdot & f_{kg}\pi_{kgG} \\ c_{kg}\pi_{kgT} & e_{kg}\pi_{kgC} & f_{kg}\pi_{kgA} & \cdot \end{pmatrix}$$

Once the tree topology, branch lengths, and site-specific rates are chosen, the likelihood at each site ($p(Y_{kgs}|\tau_k, \beta_{kg}, r_cQ_{kg})$) and the likelihood for each gene (see Equation 7.5) are computed using Felsenstein's pruning algorithm [Felsenstein, 1981].

The stationary distribution requires summation to one and so is modeled by a Dirichlet prior distribution,

$$\mathrm{diag}(\Pi_{kg}) \sim \mathrm{Dirichlet}(\alpha_{kg}).$$

The tree topology is sampled from a multinomial distribution,

$$\tau_k \sim \mathrm{Multinomial}(p_1, \ldots, p_E).$$

where $E = (2M-5)!/2^{M-3}(M-3)!$, p_i $(i = 1, \ldots, E)$ is the probability of the ith topology being sampled over the E possible M-taxon topologies. Without bias, these E topologies are assumed to be equally probable, so $p_i = 1$ $(i = 1, \ldots, E)$.

The prior information for branch lengths within a gene is modeled by an exponential distribution with an average branch length $1/\lambda_{kg}$,

$$\beta_{kg} \sim \mathrm{Exponential}(1/\lambda_{kg}).$$

The prior belief on a set of genes being congruent is expressed using π_k (as in Equation 7.1). $\pi_k = 0$ expresses a strong belief that alignments in Y_k are incongruent, while $\pi_k = 1$ says they are congruent. The Dirichlet process prior [Aldous, 1985] is used to model the prior belief. Assume a set of genes are partitioned into congruent gene clusters of various sizes (here size means the number of genes in a cluster). For a new gene not in this set, a Dirichlet process prior, in general, says that this new gene is more likely to be congruent with gene clusters of larger size. Heller and Ghahramani [2005a] proposed a prior for agglomerative clustering, which has similar property to Dirichlet Process prior:

$$\begin{array}{lll} \pi_k = 1 & d_k = \eta & \text{if } T_k \text{ is a leaf} \\ \pi_k = \frac{\eta\Gamma(N_k)}{d_k} & d_k = \eta\Gamma(N_k) + d_i d_j & \text{else} \end{array}$$

where η is the concentration hyperparameter, and $\Gamma(\cdot)$ is the Gamma function.

In this work, $\alpha_{kg} = (1, 1, 1, 1)$, $\lambda_{kg} = 10$ for all k and g, and $\eta = 0.5$, though a Bayesian hierarchical model can be easily built such that the uncertainty on hyperparameters α_{kg}, λ_{kg}, and η are incorporated into the model.

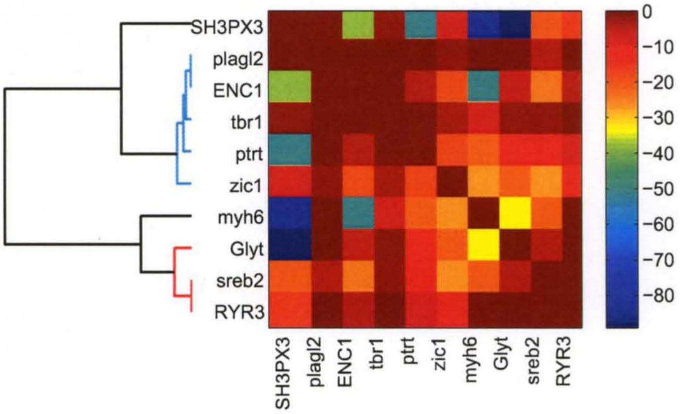


Figure 7.3: The dendrogram shows the hierarchical clustering structure of genes based on their posterior probability of being congruent. The square heatmap shows the congruence relationships between pairs of genes. The warmer the color is in a cell, the more congruent the corresponding pair of genes are. The colormap shows values of posterior probability (in logarithm) represented by colors.

7.2.4 Estimation of Marginal Likelihood

A key computation component of the model described in Section 7.2.1 is the calculation of the marginal likelihood defined in Equation 7.2, which is a highly variable function over a high dimensional parameter space. The integral is analytically intractable (e.g. due to lack of conjugate priors), and the parameter space is too high-dimensional for numerical integration. In this work, the approach by Newton and Raftery [1994] using Monte Carlo sampling from the posterior is used. Notice that marginal likelihood can be expressed as an expectation with respect to the posterior distribution of the parameters:

$$\frac{1}{p(\mathsf{Y}_k|H_0)} = \int \frac{p(\Omega_k|\mathsf{Y}_k)}{p(\mathsf{Y}_k|\Omega_k)} d\Omega_k = E\left\{ \frac{1}{p(\mathsf{Y}_k|\Omega_k)} \middle| \mathsf{Y}_k \right\} \tag{7.6}$$

where $\Omega_k = (\tau_k, \beta_{kg}, \Theta_{kg}), g = 1, \dots, N_k$ are model parameters, and $p(\mathsf{Y}_k|\Omega_k)$ are the likelihood function, as indicated in Equation 7.2. From here the harmonic mean identity can be used to approximate the marginal likelihood $p(\mathsf{Y}_k|H_0)$:

$$\hat{p}(\mathsf{Y}_k|H_0) = \left\{ \frac{1}{S} \sum_{t=1}^{S} \frac{1}{p(\mathsf{Y}_k|\Omega_k^t)} \right\}^{-1} \tag{7.7}$$

where $\Omega_k^1, \dots, \Omega_k^S$ are S samples drawn from the posterior distribution $p(\Omega_k|\mathsf{Y}_k)$.

MCMC has been widely used in phylogenetic inference to sample model parameters [Larget and Simon, 1999, Huelsenbeck and Ronquist, 2001, Suchard et al., 2001]. The approach in MRBAYES [Huelsenbeck and Ronquist, 2001] is adapted in this work. To draw from $p(\Omega_k|\mathrm{Y}_k)$, the sampler uses a Metropolis-within-Gibbs [Tierney, 1994] algorithm that cycles through blocks of model parameters within Ω_k, updating them via a Metropolis-Hastings proposal. For example, to sample the substitution model parameters for the first markers in Y_k, the acceptance probability is:

$$r = min\left(1, \frac{p(\Theta^*_{k1})}{p(\Theta_{k1})} \frac{p(\mathrm{Y}_{k1}|\tau_k, \beta_{k1}, \Theta^*_{k1})}{p(\mathrm{Y}_{k1}|\tau_k, \beta_{k1}, \Theta_{k1})} \frac{q(\Theta_{k1}|\Theta^*_{k1})}{q(\Theta^*_{k1}|\Theta_{k1})}\right) \tag{7.8}$$

where Θ^*_{k1} stands for the proposed values for the substitution model parameters. Simulated tempering [Neal, 1996], also known as Metropolis-coupled MCMC [Geyer, 1991], is used to reduce the chance that Markov chain simulations remain in the neighborhood of a single model for a long period of time.

It is worth noting that estimation of marginal likelihood remains a central problem in Bayesian inference. The decision of using the harmonic mean estimator is due to its simplicity. However, the estimator can have infinite variance. Raftery et al. [2007] described a stabilized version of the estimator. Gelman and Meng [1998] proposed path sampling which generalizes the thermodynamic integration originated from theoretical physics and involves a sequence of intermediate distributions bridging prior and posterior. Lartillot and Philippe [2006] applied thermodynamic integration to phylogenetic analysis.

7.3 Results

The method proposed herein is used to estimate the phylogeny relationships amongst ray-finned fish (*Actinopterygii*) with 10 alignments of protein-coding genes assembled by Li et al. [2008]. Twenty species, out of 52 ray-finned fish, are randomly selected, and mouse (*Mus musculus*) is used as the outgroup to root the phylogeny tree. Li et al. [2008] defined one data block for each codon position and each gene, yielding 30 data blocks (3 codon positions × 10 genes). For each data block, substitution parameters (GTR $+\,\Gamma$) were estimated using maximum likelihood and Bayesian inference method. They defined the distance between data blocks using their estimated substitution parameters. Then data blocks were clustered by hierarchical clustering with centroid linkage. As expected, the three major clusters discovered by their method corresponded exactly to codon positions. The trees inferred from each individual gene by the Bayesian phylogenetic method (MRBAYES GTR$+\Gamma$) either are poorly resolved star-like trees or exhibit an obviously different topology (data not shown

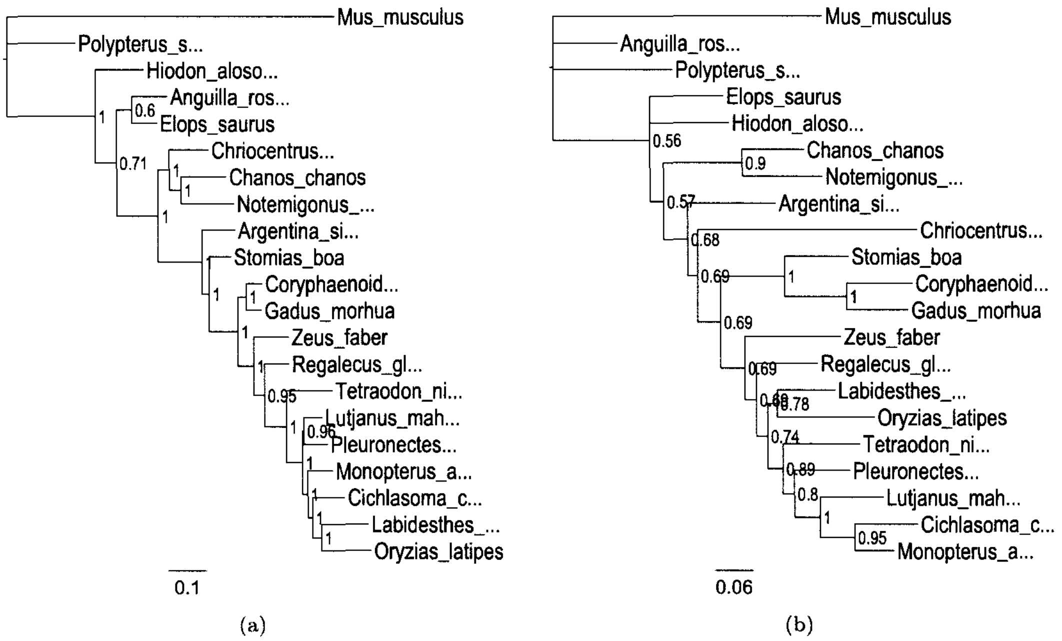


Figure 7.4: 50% majority-rule consensus trees inferred from congruent set 1 (a) and congruent set 2 (b). Posterior probabilities for branches are indicated.

here), indicating that a systematic way of combining these genes is desirable in order to accurately analyze the data set.

The Bayesian topological congruence method proposed herein is applied to identify congruent sets of genes using a Dirichlet process prior with concentration parameter $\eta = .5$. From this test, four mutually incongruent sets of genes were identified, containing 5, 3, 1, and 1 genes, respectively. The pairwise gene congruence is shown in a square matrix in Figure 7.3. The warmer (e.g., red is warmer than blue) the color is in a cell, the more congruent the corresponding pair of genes are. The colorbar maps color to values of posterior probability (on a logarithmic scale). The degree of congruence between genes ranges from extremely congruent to extremely incongruent. Gene pairs such as (*plagl2, ENC1*), (*tbr1, ptrt*) are highly congruent, with posterior probabilities near 1; gene pairs such as (*myh6, SH3PX3*), (*ENC1, myh6*) are highly incongruent, with posterior probabilities smaller than e^{-50}. It also indicates that some genes, such as *SH3PX3* and *myh6* are incongruent to most of the other genes.

Genes are further clustered into congruent subsets, shown in a dendrogram in Figure 7.3. Branch lengths in the dendrogram correspond to the posterior probability of congruence between gene subsets connected by the branch. The shorter the branch, the more congruent they are. The cut point value is $p = 0.5$. Branches having $r \leq 0.5$ are in black and $r > 0.5$ are in lighter colors. The tree shows two main congruent subsets: set1=(*plagl2, ENC1, tbr1, ptrt, zic1*) and set2=(*RYR3,* *sreb2, Glyt*). Notice that although *SH3PX3* is congruent to *plagl2* and *tbr1*, it is not included in congruent set 1 since that merge has the posterior probability $r = e^{-67}$. This is also indicated in the square matrix, where the first column shows that *SH3PX3* is incongruent to *ENC1* and *ptrt.* Similarly, although *myh6* is highly congruent with *RYR3*, the gene is not included in congruent set 2 because the merge has posterior probability $r = e^{-50}$.

Bayesian phylogenies inferred from each of the two congruent sets are shown in Fig. 7.4. Branches of the 50% majority-rule consensus tree from congruent set 1 have high posterior probability, providing strong support for the topology. The main branch with low probability is the pair (*Anguilla_rostrata, Elops_saurus*). Although the 50% majority-rule consensus tree from congruent set 2 has an overall similar topology as the one from congruent set 1, its branches have relatively low posterior probabilities. However, one interesting result comes from analysis of congruent set 2. In this set, there are three levels of ancestral nodes from the *Chriocentrus_dorab* group to the (*Chanos_chanos, Notemigonus_crysoleucas*) group, while in congruent set 1, *Chriocentrus_dorab* and the (*Chanos_chanos, Notemigonus_crysoleucas*) group share an immediate common ancestor.

7.4 Discussion

Bayesian methods of multigene analysis correspond to various ways of partitioning the genome Pagel and Meade [2004], Lartillot and Philippe [2004], Rannala and Yang [2008], Nylander et al. [2004]. Gene topological congruence analysis can be considered as partitioning genes according to the underlying gene topology while allowing branch length and substitution heterogeneity within a partition. To infer gene partitioning based on topological congruence, a mixture model is proposed:

$$p(\mathsf{z}|\mathsf{Y}) = \frac{p(\mathsf{Y}|\mathsf{z})p(\mathsf{z})}{p(\mathsf{Y})} = \frac{\prod_{k=1}^{K} L(\mathsf{Y}_k|H_0)p(\mathsf{z})}{p(\mathsf{Y})} \tag{7.9}$$

where, if N genes are clustered into K partitions, $\mathsf{z} = (z_1, \ldots, z_N)$, z_i is the partition of the ith gene, and $L(\mathsf{Y}_k|H_0)$ is the marginal likelihood integrating over heterogeneous parameters, as defined by Equation 7.2 for topological congruence.

The posterior probability of multiple markers (for example, three markers: a, b, c) being congruent given the sequences are the posterior probability of them being assigned into one partition:

$$p(H_0|\mathsf{Y}) = \sum_{\substack{\mathbf{z} \in \mathcal{Z} \\ z_a = z_b = z_c}} p(\mathbf{z}|\mathsf{Y}) \tag{7.10}$$

where $\mathcal{Z}$ is the set of all possible clusterings over N elements. Although MCMC inference algorithm has been widely used for phylogenetic analysis, sampling over the large sample space imposed by Equation 7.9 is extremely computationally expensive.

The greedy agglomerative algorithm in Figure 7.1 can be considered as a deterministic alternative to estimating the mixture model (Equation 7.9) by a sampling method such as MCMC. However, it must be noted that this method still does not scale well with very large numbers of loci for two reasons. First, the agglomerative algorithm (Figure 7.1) has a computation time complexity of $O(N^2)$, where N is the number of genes in the data set. Second, the merging criterion still requires calculating the marginal likelihood (Equation 7.2) using an MCMC sampler. For this reason, the experiment reported in this work includes only ten genes and twenty taxa, a data set smaller than would be normally interesting to genome wide phylogenetic analysis.

In general, Bayesian phylogenomic analysis methods that account for evolutionary heterogeneity among genes, including the algorithm described in this work, can present significant computational challenges. One solution is to devise parallelizable algorithms. It is particularly interesting to point out that the algorithm presented in this work is readily parallelizable. For example, given three gene clusters i, j and k, the evaluation of $p(H_0|\mathsf{Y}_i, \mathsf{Y}_j)$ and $p(H_0|\mathsf{Y}_i, \mathsf{Y}_k)$ are independent and can therefore be computed in parallel by different machines. This can significantly speed up the computation and allow much larger scale applications of the algorithm.

7.5 Conclusion

Genomic scale data offers invaluable opportunities to solve difficult phylogenetic problems, but also imposes enormous challenges for statistical and computational methods [Rannala and Yang, 2008]. The method proposed in this work accounts for evolutionary heterogeneities and identifies congruent gene subsets using Bayesian hypothesis testing. The proposed method approximates the posterior probability of genes being congruent in a fast deterministic manner. A notable feature of the method is that it is particularly suitable for parallel computation. The test presented on the data set shows that the model recovers interesting congruence structure among genes. Future work will explore applications of the model to more interesting genome wide data.

CHAPTER 8

SUMMARY AND FUTURE WORK

8.1 Summary of Methods and Contributions

In this dissertation, we investigated several Bayesian nonparametric techniques to clustering in relational and high-dimensional settings. These include 1) infinite relational model with feature selection (FIRM) which incorporates the rich information of multi-relational data; 2) Bayesian Hierarchical Cross-Clustering (BHCC), a deterministic approximation to Cross Dirichlet Process mixture (CDPM) and to cross-clustering; 3) randomized approximation (RBHCC), based on a truncated hierarchy; and 4) Bayesian Congruence Measuring (BCM), an extension of BHCC, which measures incongruence between genes and to identify sets of congruent loci with identical evolutionary histories. As particular examples, we considered applications of these methods in solving challenging biological problems. In the study of interactions between microRNA and mRNA, FIRM was applied on both miRNA-mRNA correlation matrix and gene ontology (GO) annotation data, and discovered GO terms, and mRNA/miRNA clusterings that suggesting interesting biological functions. In the field of chemogenomics and drug-target interaction, FIRM was applied to a variety of chemogenomic data and was able to predict drug-target interaction even with high degree of missing values. In the field of phylogenomic analysis, BCM was applied to estimate the phylogeny relationships amongst ray-finned fish (*Actinopterygii*) with 10 alignments of protein-coding genes, and the result show that the model recovers interesting congruence structure among genes.

Infinite relational models (IRMs) are generalizations of Dirichlet process mixture (DPM) models to the relational domain, where the observations include both the object-feature data representing entity properties, and one or more relations involving multiple types, representing object-object relationships. In Chapter 4, we developed FIRM, a Bayesian nonparametric model which extends the infinite relational model with simultaneous feature selection. FIRM addresses many cases in which the structure of greatest interest may be best represented using only a selected subset of features, therefore results in structures with more intuitive interpretation and often better prediction accuracy. By conditioning the multiple probability density functions on latent component variables, the model allows for information exchange between features and relations of the same entity type, and also leads to information propagation among different entity types through the entire multi-relational network. Although the joint density of the parameters is not known, the parameters can be partitioned into groups whose posterior conditional densities are known. Meanwhile, part of the parameter groups can be integrated out while the resulting marginalized posterior conditional densities are still computational feasible. This leads to our proposed inference procedure based on collapsed Gibbs sampling, which iteratively sweeps through the groups of parameters (while skipping the ones that are integrated out) and generate a random sample for each, conditioned on the current value of the others. This procedure forms a homogeneous Markov chain and its stationary distribution is exactly the joint posterior.

The identification of gene and protein functions, and the prediction of interactions among biological entities is an active research field facilitated by means of categorizing the entities into biologically sensible groups. Chapter 5 applied FIRM to challenging problems in bioinformatics. We begin with the problem of discovering groups of mRNA and microRNA in the biological context of breast cancer. The model encodes latent categorization of mRNA and microRNA, and the latent saliency of gene ontology terms. The latent structures further encode the gene expressions of microRNA and mRNA, and the gene ontology annotation mappings. We also studied the prediction of drug-target interactions, by encoding in the model latent categorization structure of drugs and proteins, which in turn encodes drug-target interaction, chemical compound similarity, and amino acid sequence similarity.

DPM is a widely used Bayesian nonparametric model for clustering and for density estimation. Although it allows for unbounded number of model parameters, and is more flexible than a finite mixture model, the model assumes a single clustering structure to account for all the variability in the data. Chapter 6 developed several approximate inference algorithms for the Cross Dirichlet Process Mixture (CDPM) model, which allows for multiple views, each describing the data using a subset of the dimensions. Meanwhile, the model does not restrict the number of views *a priori*, but allows potentially an unbounded number of views that are automatically inferred for a given data set. The proposed Bayesian Hierarchical Cross-Clustering (BHCC) is a greedy and deterministic approximation to CDPM which results in a hierarchical clustering of dimensions, and at each node, a hierarchical clustering of data points. We derived the posterior predictive distribution and asymptotic lower bound for the CDPM. The randomized BHCC (RBHCC) is more efficient approximation algorithm extended from BHCC, and is based on a randomization and truncated hierarchy, that scales linearly in the number of dimensions and data points. Predictive performance on synthetic and real-world data sets demonstrated that approximation algorithms for CDPM are more efficient and effective in explaining the heterogeneity in high dimensional data than algorithms for DPM.

Generalizing cross-clustering and the BHCC algorithm, Chapter 7 developed the Bayesian Congruence Measuring (BCM) to estimate the degree of incongruence among phylogenies of different genes, and to identify sets of congruent loci within which the evolutionary histories are identical. Analogical to cross-clustering where there exits multiple views, BCM also consists of multiple views. Rather than clustering structure as originally developed in CDPM, the intended structure of each view (congruent loci) in BCM represents evolutionary processes. The results on a gene sequence data of 10 nuclear genes from 20 ray-finned fish (*Actinopterygii*) species demonstrates the interesting properties of the algorithm.

8.2 Future Directions

There are many exciting directions for future research inspired by this dissertation. In this section, we briefly survey several potential research directions suggested by our methods.

8.2.1 Potential Applications

Biological systems often involve multiple types of elements (e.g. mRNA, miRNA, and protein) and events (e.g. gene transcription and translation, mRNA repression or degradation by miRNA, and alternative splicing). High throughput experimental techniques are able to capture snapshots of the events and elements. For example, microarrays and RNA-Seq data are used to measure gene and miRNA expression, and other transcriptome data; ChIP-chip and ChIP-seq are used to measure protein interactions with DNA. Meanwhile, various type of databases have been accumulated to represent biological information such as biological processes, molecular functions, cellular components, and pathway information. However, the high dimensional, multi-relational, and noisy nature of these data provides tremendous challenges for analysis. It is thus important to integrate multiple types of data so that they complement each other. We have applied FIRM on two problem domains (miRNA-mRNA interaction, and drug-target interaction) and generated promising results. The success of FIRM depends on how congruent the dataset are to each other, and how relevant the data is to the problem. Therefore, it is always important to involve domain expert when applying FIRM.

While we have focused on applications in bioinformatics, this approaches are also broadly useful in many other fields of interest.

8.2.2 Alternative Bayesian Inference and Learning Approach

In the Bayesian paradigm, the uncertainty of model parameters is represented by a posterior probability, which, according to Bayes' rule, is proportional to the product of the prior and the likelihood. Because of the unified view of both observations and models as random variables, the Bayesian framework and probabilistic graphical model allows for building models representing complex systems. One common theme running through many of these richer and more complex Bayesian models is that the component likelihood function may by itself be a mixture of distributions, thus does not have an efficient analytic (closed-form) solution. Estimation of marginal likelihood remains a central problem in Bayesian inference. The component likelihood function of the BHCC and randomized BHCC (Chapter 6) is by itself a DPM model. The BCM algorithm (Chapter 7) also has a component likelihood function which marginalizes over uncertainty in trees, branch length, and substitution parameters. These models require efficient and reliable approximation algorithm to estimate their component likelihood functions.

We have chosen the Bayesian hierarchical clustering (BHC) to approximate the DPM in BHCC. The BHC is an efficient and determnistic algorithm to DPM which provides a lower bound to the DPM. Alternatively, it would be interesting to use other learning algorithm to DPM, such as the variational inference algorithm, as reviewed in Sec. 2.4.4. For the likelihood function in BCM, we have chosen the harmonic mean estimator due to its simplicity. It is known that the estimator can have infinite variance. Gelman and Meng [1998] proposed path sampling which generalizes the thermodynamic integration originated from theoretical physics and involves a sequence of intermediate distributions bridging prior and posterior. Lartillot and Philippe [2006] applied thermodynamic integration to phylogenetic analysis. It would be interesting to adapt these alternative methods to the BCM algorithm.

Recent Bayesian inference community has also seen the popularity of approximate Bayesian computation (ABC), which is considered as a generally valid approximation for doing Bayesian inference in complex models [Beaumont et al., 2002]. ABC has also been applied to conduct the model choice in a wide range of phylogenetic models [Cornuet et al., 2008, Robert et al., 2011]. It is thus interesting to study and veryfy its application in the models presented in this dissertation.

8.2.3 Alternative Nonparametric Methods

The Dirichlet process allows for infinite (unknown and unbounded) number of mixture components associated with the data. It has desirable asymptotic properties, and leads to simple and effective learning algorithms. The models presented in this dissertation are direct applications or extensions of the Dirichlet process. However, it is not the only framework with such properties, it would be interesting to explore some alternative Bayesian nonparametric methods. Meanwhile, nonparametric methods alternative to Bayesian nonparametrics, such as kernel density estimation, have also proven to be useful for modeling. It would be interesting to integrate these alternative methods into the framework of probabilistic graphical model for complex problems.

8.2.4 Causal Learning and Biological Networks

Cause-effect relationships are the fundamental building blocks of both nature and human understanding of nature. Biological networks are the abstracted representations of the causal "machinery" underlying complex biological systems (such as protein-DNA, protein-protein, and protein-metabolite interactions), and capture many of the systems' essential characteristics. However, the complexity of nature and the inevitable lack of detailed observations make human understanding of biological systems extremely challenging. Despite the development of computational algorithms and the availability of high-throughput experimental methods for sequencing and binding, our understanding of system-level functions and mechanisms of a biological network is still hindered substantially by the lack of appropriate methodologies for learning complex and dynamic causal relationships even with many resources available. For instance, although Bayesian network (and dynamic Bayesian network) representation of large problems can often lead to a causal interpretation, the application of it to learn biological network is still challenging computationally, statistically, and theoretically. It would be interesting to develop novel representational and inferential methods, and observational and experimental strategies for the understanding of dynamic causal relationships inside biological systems, while borrowing the ideas of Dirichlet process, BHCC, and RBHCC for efficient exploration over the searching space. Recent findings on network motifs have offered exciting insights into systems biology. Network motifs, the recurring sub-network patterns throughout a biological network, carry out specific information-processing functions and are presumably favored by evolution, and hence are more abundant than other network patterns. While a complete understanding of the functions and mechanisms of biological networks in dynamic and large-scale settings remains grand challenges, effort on simultaneously discovering, assembling, and refining network motifs could lead to significant progress on the investigation. Therefore, it would be particularly interesting in defining potential canonical network patterns composed of multiple levels of interactions (e.g. protein-DNA, protein-protein, microRNA-mRNA, metabolic interactions, and evolutionary patterns), identifying these putative motifs on several large biological networks and estimating their probabilistic distributions, designing computational algorithms of learning network motifs in biological networks from high-throughput experimental data, and analyzing the identifiability and computational complexity of these algorithms.

References

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. 53

J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 44(2):139–177, 1982. 35

H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, January 1974. 35

D. Aldous. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII - 1983*, pages 1–198, 1985. 18, 53, 56, 92, 115

C. Ané, B. Larget, D. A. Baum, S. D. Smith, and A. Rokas. Bayesian estimation of concordance among gene trees. *Mol Biol Evol*, 24(2):412–426, 2007. 110

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene Ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–29, May 2000. 72

A. Asuncion and D. J. Newman. UCI machine learning repository, 2007. 103

O. T. Avery, C. M. MacLeod, and M. McCarty. Studies on the chemical nature of the substance inducing transformation of pneumococcal types: induction of transformation by a desoxyribonucleic acid fraction isolated from *Pneumococcus* type III. *Journal of Experimental Medecine*, 79(2): 137–158, Feb. 1944. 27

Z. Bar-Joseph, G. K. Gerber, T. I. Lee, N. J. Rinaldi, J. Y. Yoo, F. Robert, D. B. Gordon, E. Fraenkel, T. S. Jaakkola, R. A. Young, and D. K. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342, Oct. 2003. 35, 68

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden markov model. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002. 2

M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian Computation in Population Genetics. *Genetics*, 162(4):2025–2035, Dec. 2002. 125

Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. 77

M. J. Bessman, I. R. Lehman, E. S. Simms, and A. Kornberg. Enzymatic synthesis of deoxyribonucleic acid. ii. general properties of the reaction. *The Journal of Biological Chemistry*, 233:171–177, 1958. 28

D. Betel, A. Koppal, P. Agius, C. Sander, and C. Leslie. Comprehensive modeling of microRNA targets predicts functional non-conserved and non-canonical sites. *Genome Biology*, 11(8):R90+, Aug. 2010. 71

C. M. Bishop. Figures from pattern recognition and machine learning. URL `http://research.microsoft.com/en-us/um/people/cmbishop/prml/webfigs.htm`. 13, 16

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, August 2006. 13, 16, 17, 35, 80

D. Blackwell and J. B. MacQueen. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973. 20, 21

K. Bleakley and Y. Yamanishi. Supervised prediction of drug-target interactions using bipartite local models. *Bioinformatics (Oxford, England)*, 25(18):2397–2403, Sept. 2009. 81

D. M. Blei and M. I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006. 25

D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003. 2

D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, page 2003. MIT Press, 2004. 2

K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H. P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl 1):47–56, Jan. 2005. 80

S. Chang, N. Dasgupta, and L. Carin. A bayesian approach to unsupervised feature selection and density estimation using expectation propagation. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 1043–1050, 2005. 54

E. Chargaff. Chemical specificity of nucleic acids and mechanism of their enzymatic degradation. *Cellular and Molecular Life Sciences (CMLS)*, 6(6):201–209, June 1950. 28

Y. Cheng and G. M. Church. Biclustering of expression data. *International Conference on Intelligent Systems for Molecular Biology*, 8:93–103, 2000. 37, 38

C. Constantinopoulos, M. K. Titsias, and A. Likas. Bayesian feature and model selection for gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1013–1018, 2006. 54, 55

A. Corduneanu and C. M. Bishop. Variational bayesian model selection for mixture distributions. In *Artificial Intelligence and Statistics*, 2001. 55

J.-M. Cornuet, F. Santos, M. A. Beaumont, C. P. Robert, J.-M. Marin, D. J. Balding, T. Guillemaud, and A. Estoup. Inferring population history with DIY ABC: a user-friendly approach to approximate Bayesian computation. *Bioinformatics*, 24(23):2713–2719, Dec. 2008. 125

F. Crick. On protein synthesis. *Symposia of the Society for Experimental Biology*, 12:138–163, 1958. 26

F. Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, Aug. 1970. 26

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, Mar. 2000. 80

Y. Cui, X. Z. Fern, and J. G. Dy. Non-redundant multi-view clustering via orthogonalization. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 133–142, 2007. 91

X. H. Dang and J. Bailey. Generation of alternative clusterings using the cami approach. In *Proceedings of the SIAM International Conference on Data Mining*, pages 118–129, 2010. 91

P. Darlu and G. Lecointre. When does the incongruence length difference test fail? *Mol Biol Evol*, 19(4):432–437, 2002. 109

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 48

D. Didiano and O. Hobert. Perfect seed pairing is not a generally reliable predictor for miRNA-target interactions. *Nature structural & molecular biology*, 13(9):849–851, Sept. 2006. 71

K. Dolphin, R. Belshaw, Orme, and D. L. Quicke. Noise and incongruence: Interpreting results of the incongruence length difference test. *Molecular Phylogenetics and Evolution*, 17(3):401–406, 2000. 109

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience, 2nd edition, November 2000. 35, 45

A. Edwards and L. Cavalli-Sforza. Reconstruction of evolutionary trees. *Phenetic and Phylogenetic Classification, ed. V. H. Heywood and J. McNeill*, pages 67–76, 1964. 44

D. Erhan and P.-J. L'Heureux. Collaborative Filtering on a Family of Biological Targets. *Journal of Chemical Information and Modeling*, 46(2):626–635, 2006. 80

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995. 2, 24, 35, 53, 55

W. J. Ewens and G. R. Grant. *Statistical Methods in Bioinformatics: An Introduction (Statistics for Biology and Health).* Springer, 2nd edition, December 2004. 40

S. Falcon and R. Gentleman. Using GOstats to test gene lists for GO term association. *Bioinformatics*, 23(2):257–8, 2007. 78

J. S. Farris, M. Kallersjo, A. G. Kluge, and C. Bult. Testing significance of incongruence. *Cladistics*, 10(3):315–319, 1994. 109

L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *CVPR*, pages 524–531, 2005. 2

J. Felsenstein. The number of evolutionary trees. *Systematic Zoology*, 27(1):27+, March 1978. 45

J. Felsenstein. Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981. 40, 47, 48, 49, 115

J. Felsenstein. *Inferring Phylogenies.* Sinauer Associates, 2 edition, 2003. 45

T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1 (2):209–230, 1973. 18, 53, 55

M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002. 55

R. E. Franklin and R. G. Gosling. Molecular configuration in sodium thymonucleate. *Nature*, 171 (4356):740–741, Apr. 1953. 28

N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999. 53

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, Aug. 2000. 35, 68

N. Friedman, M. Ninio, I. Pe'er, and T. Pupko. A structural em algorithm for phylogenetic inference. *J Comput Biol*, 9(2):331–353, 2002. 48

O. Gascuel. Bionj: an improved version of the nj algorithm based on a simple model of sequence data. *Mol Biol Evol*, 14(7):685–695, July 1997. 49

A. Gelman and X.-L. Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statist. Sci.*, 13(2):163–185, 1998. 117, 125

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis, Second Edition.* Chapman and Hall/CRC, 2 edition, 2003. 13

C. J. Geyer. Markov chain monte carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of 23rd Symposium on the Interface*, pages 156–163. Interface Foundation, Fairfax Station, 1991. 51, 63, 102, 117

D. Gondek and T. Hofmann. Non-redundant data clustering. In *Proceedings of the 4th IEEE International Conference on Data Mining*, pages 75–82, 2004. 91

D. Graur and W.-H. Li. *Fundamentals of Molecular Evolution.* Sinauer Associates, 2 edition, 2000. 26

A. Greenfield, A. Madar, H. Ostrer, and R. Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PLoS ONE*, 5(10):e13397+, Oct. 2010a. 35

A. Greenfield, A. Madar, H. Ostrer, and R. Bonneau. DREAM4: Combining Genetic and Dynamic Information to Identify Biological Networks and Dynamical Models. *PLoS ONE*, 5(10):e13397+, Oct. 2010b. 68

F. Griffith. The significance of pneumococcal types. *The Journal of hygiene*, 27(2):113–159, Jan. 1928. 27

A. Grimson, K. K. Farh, W. K. Johnston, P. Garrett-Engele, L. P. Lim, and D. P. Bartel. MicroRNA Targeting Specificity in Mammals: Determinants beyond Seed Pairing. *Molecular Cell*, 27(1):91–105, July 2007. 71

GSE19536. URL `http://www.ncbi.nlm.nih.gov/projects/geo/query/acc.cgi?acc=GSE19536`. 71

X. Gu, Y.-X. Fu, and W.-H. Li. Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites. *Mol Biol Evol*, 12(4):546–57, 1995. 52

Y. Guan, J. G. Dy, D. Niu, and Z. Ghahramani. Variational inference for nonparametric multiple clustering. In *MultiClust: 1st International Workshop on Discovering, Summarizing and Using Multiple Clusterings held in conjunction with the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010. 91

S. Guindon and O. Gascuel. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Systematic biology*, 52(5):696–704, October 2003. 49

D. L. Hartl and A. G. Clark. *Principles of Population Genetics.* Sinauer Associates, Inc. Publishers, Sunderland, Massachusetts. 109

M. Hasegawa, H. Kishino, and T.-A. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *Journal of Molecular Evolution*, 22(2):160–174, October 1985. 40

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., 2nd edition, 2009. 14, 35

M. Hattori, Y. Okuno, S. Goto, and M. Kanehisa. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. *Journal of the American Chemical Society*, 125(39):11853–11865, Oct. 2003. 81

L. He and G. J. Hannon. MicroRNAs: small RNAs with a big role in gene regulation. *Nature Reviews Genetics*, 5(7):522–531, July 2004. 70

K. A. Heller and Z. Ghahramani. Bayesian hierarchical clustering. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM Press, 2005a. 2, 25, 55, 91, 93, 94, 96, 113, 115

K. A. Heller and Z. Ghahramani. Randomized algorithms for fast bayesian hierarchical clustering, 2005b. 99

D. M. Hillis and J. J. Bull. An empirical test of bootstrapping as a method for assessing confidence in phylogenetic analysis. *Systematic biology*, 42(2):182–192, 1993. 110

http://www.microrna.org. URL `http://www.microrna.org`. 70

J. P. Huelsenbeck and J. J. Bull. A likelihood ratio test to detect conflicting phylogenetic signal. *Systematic Biology*, 45(1):92–98, 1996. 109

J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001. 51, 110, 117

J. P. Huelsenbeck, B. Larget, and M. E. Alfaro. Bayesian phylogenetic model selection using reversible jump markov chain monte carlo. *Mol Biol Evol*, 21(6):1123–1133, June 2004. 40

D. H. Huson, R. Rupp, and C. Scornavacca. *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge University Press, 2011. 44

O. Ivanciuc. *Reviews in Computational Chemistry*, volume 23, chapter Applications of Support Vector Machines in Chemistry, pages 291–400. John Wiley & Sons, Inc., 2007. 80

T. Jaakkola, M. Diekhans, and D. Haussler. A Discriminative Framework for Detecting Remote Protein Homologies. *Journal of Computational Biology*, 7(1-2):95–114, Feb. 2000. 80

L. Jacob and J.-P. Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, Oct. 2008. 80

R. Jain, M. C. Rivera, and J. A. Lake. Horizontal gene transfer among genomes: The complexity hypothesis. *Proceedings of the National Academy of Sciences*, 96:3801–3806, 1999. 109

S. Jain and R. M. Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2004. 35, 55, 63, 102

E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003. 7

M. Jelizarow, V. Guillemot, A. Tenenhaus, K. Strimmer, and A.-L. Boulesteix. Over-optimism in bioinformatics: an illustration. *Bioinformatics*, pages 1990–1998, 2010. 35

B. John, A. J. Enright, A. Aravin, T. Tuschl, C. Sander, and D. S. Marks. Human MicroRNA targets. *PLoS biology*, 2(11):e363+, Nov. 2004. 71

T. H. Jukes and C. R. Cantor. *Evolution of Protein Molecules*. Academy Press, 1969. 40

M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic acids research*, 36(Database issue):D480–484, Jan. 2008. 81, 82

R. E. Kass and A. E. Raftery. Bayes factor and model uncertainty. *Journal of the American Statistical Association*, 90(430):773–795, 1995. 35

C. Kelly and J. Rice. Modeling nucleotide evolution: a heterogeneous rate analysis. *Math Biosci*, 133(1):85–109, 1996. 52

C. Kemp and J. B. Tenenbaum. Learning systems of concepts with an infinite relational model. In

In Proceedings of the 21st National Conference on Artificial Intelligence, 2006. 53

M. Kertesz, N. Iovino, U. Unnerstall, U. Gaul, and E. Segal. The role of site accessibility in microRNA target recognition. *Nature genetics*, 39(10):1278–1284, Oct. 2007. 71

Kim, Sinae, Tadesse, G. Mahlet, Vannucci, and Marina. Variable selection in clustering via dirichlet process mixture models. *Biometrika*, 93(4):877–893, 2006. 55

H. Kim, G. H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2):187–198, Jan. 2005. 72

M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16(2):111–120, June 1980. 40, 42

A. Krek, D. Grun, M. N. Poy, R. Wolf, L. Rosenberg, E. J. Epstein, P. MacMenamin, I. da Piedade, K. C. Gunsalus, M. Stoffel, and N. Rajewsky. Combinatorial microRNA target predictions. *Nature Genetics*, 37(5):495–500, Apr. 2005. 71

J. Kruger and M. Rehmsmeier. RNAhybrid: microRNA target prediction easy, fast and flexible. *Nucl. Acids Res.*, 34(suppl_2):W451–454, July 2006. 71

E. C. Lai. Micro RNAs are complementary to 3' UTR sequence motifs that mediate negative post-transcriptional regulation. *Nature genetics*, 30(4):363–364, Apr. 2002. 70

B. Larget and D. L. Simon. Markov chasin monte carlo algorithms for the bayesian analysis of phylogenetic trees. *Mol Biol Evol*, 16:750–759, 1999. 50, 110, 116

N. Lartillot and H. Philippe. A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Mol Biol Evol*, 21(6):1095–1109, 2004. 110, 119

N. Lartillot and H. Philippe. Computing bayes factors using thermodynamic integration. *Systematic Biology*, 55(2):195–207, 2006. 117, 125

N. C. Lau, L. P. Lim, E. G. Weinstein, and D. P. Bartel. An Abundant Class of Tiny RNAs with Probable Regulatory Roles in Caenorhabditis elegans. *Science*, 294(5543):858–862, Oct. 2001. 71

M. H. Law, A. K. Jain, and M. A. T. Figueiredo. Feature selection in mixture-based clustering. In *Advances in Neural Information Processing Systems*, pages 625–632. MIT Press, 2003. 54

M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain. Simultaneous feature selection and clustering using mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9): 1154–1166, 2004. 54, 55

L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2000. 38, 39

R. C. Lee and V. Ambros. An Extensive Class of Small RNAs in Caenorhabditis elegans. *Science*, 294(5543):862–864, Oct. 2001. 71

I. R. Lehman, M. J. Bessman, E. S. Simms, and A. Kornberg. Enzymatic synthesis of deoxyribonucleic acid. i. preparation of substrates and partial purification of an enzyme from escherichia coli. *The Journal of Biological Chemistry*, 233:163–170, 1958. 28

J. W. Leigh, E. Susko, M. Baumgartner, and A. J. Roger. Testing congruence in phylogenomic analysis. *Systematic Biology*, 57(1):104–115, 2008. 110

A. R. Lemmon and M. C. Milinkovitch. The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation. *Proc Natl Acad Sci U S A*, 99(16):10516–10521, August 2002. 49

B. P. Lewis, I.-h. H. Shih, M. W. Jones-Rhoades, D. P. Bartel, and C. B. Burge. Prediction of mammalian microRNA targets. *Cell*, 115(7):787–798, Dec. 2003. 71

B. P. Lewis, C. B. Burge, and D. P. Bartel. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microrna targets. *Cell*, 120(1):15–20, January 2005. 71

P. O. Lewis. A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. *Mol Biol Evol*, 15(3):277–283, March 1998. 49

C. Li, G. Lu, and G. Orti. Optimal data partitioning and a test case for ray-finned fishes (actinopterygii) based on ten nuclear loci. *Systematic Biology*, 57(4):519–539, 2008. 117

L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 577–584, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. 2

L. Lim, M. Glasner, S. Yekta, C. Burge, and D. Bartel. Vertebrate MicroRNA Genes. *Science*, 299 (5612):1540+, Mar. 2003. 71

L. P. Lim, N. C. Lau, P. Garrett-Engele, A. Grimson, J. M. Schelter, J. Castle, D. P. Bartel, P. S. Linsley, and J. M. Johnson. Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs. *Nature*, 433(7027):769–773, Feb. 2005. 70

R. Macarron, M. N. Banks, D. Bojanic, D. J. Burns, D. A. Cirovic, T. Garyantes, D. V. S. Green, R. P. Hertzberg, W. P. Janzen, J. W. Paslay, U. Schopfer, and G. S. Sittampalam. Impact of high-throughput screening in biomedical research. *Nature Reviews Drug Discovery*, 10(3):188–195, Mar. 2011. 79

D. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003. 7

S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE transactions on computational biology and bioinformatics*, 1(1):24–45, 2004. 37

V. K. Mansinghka, E. Jonas, C. Petschulat, B. Cronin, P. Shafto, and J. B. Tenenbaum. Cross-categorization: A method for discovering multiple overlapping clusterings. In *Advances in Neural*

Information Processing Systems, Workshop on Nonparametric Bayesian Statistics, 2009. 90, 91, 93

B. Mau, M. A. Newton, and B. Larget. Bayesian phylogenetic inference via markov chain monte carlo methods. *Biometrics*, 55(1):1–12, March 1999. 50

I. Mayrose, N. Friedman, and T. Pupko. A gamma mixture model better accounts for among site rate heterogeneity. *Bioinformatics*, 21 Suppl 2, September 2005. 52

M. Medvedovic and S. Sivaganesan. Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics*, 18(9):1194–1206, September 2002. 2, 35, 68

M. Meselson and F. W. Stahl. The replication of dna in escherichia coli. *Proceedings of the National Academy of Sciences*, 44(7):671–682, July 1958. 28

M. S. Meselson and C. M. Radding. A general model for genetic recombination. *Proceedings of the National Academy of Sciences*, 72(1):358–361, 1975. 109

R. M. Neal. Sampling from multimodal distributions using tempered transitions. *Journal Statistics and Computing*, 6(4):353–366, 1996. 51, 117

R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. Technical report, 4915 Department of Statistics University of Toronto, 1998. 24, 35, 55, 92, 102

M. A. Newton and A. E. Raftery. Approximate bayesian inference with the weighted likelihood bootstrap. *J. Roy. Statist. Soc.*, (B 56):3–48, 1994. 116

Nidhi, M. Glick, J. W. Davies, and J. L. Jenkins. Prediction of Biological Targets for Compounds Using Multiple-Category Bayesian Models Trained on Chemogenomics Databases. *Journal of Chemical Information and Modeling*, 46(3):1124–1133, May 2006. ISSN 1549-9596. 80

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, USA, August 2000. 48

J. A. Nylander, F. Ronquist, J. P. Huelsenbeck, and J. Nieves-Aldrey. Bayesian phylogenetic analysis of combined data. *Systematic biology*, 53(1):47–67, 2004. 119

G. Olsen, H. Matsuda, R. Hagstrom, and R. Overbeek. fastdnaml: a tool for construction of phylogenetic trees of dna sequences using maximum likelihood. *Computer Applications in the Biosciences*, 10(1):41–8, 1994. 48

M. Pagel and A. Meade. A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data. *Syst Biol*, 53(4):571–581, 2004. 110, 119

G. V. Paolini, R. H. Shapland, W. P. van Hoorn, J. S. Mason, and A. L. Hopkins. Global mapping of pharmacological space. *Nature biotechnology*, 24(7):805–815, July 2006. 79

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009. 7

J. Pitman. *Combinatorial stochastic processes*, volume 1875 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2006. ISBN 978-3-540-30990-1; 3-540-30990-X. 20, 21, 53, 56

P. J. Planet. Tree disagreement: Measuring and testing incongruence in phylogenies. *Journal of Biomedical Informatics*, 39(1):86–102, 2006. 109

Z. Qi and I. Davidson. A principled and flexible framework for finding alternative clusterings. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 717–726, 2009. 91

Z. Qin. Clustering microarray gene expression data using weighted chinese restaurant process. *Bioinformatics*, June 2006. 35, 68

A. E. Raftery, M. A. Newton, J. M. Satagopan, and P. N. Krivitsky. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. *Bayesian Statistics 8*, pages 1–45, 2007. 117

B. Rannala and Z. Yang. Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. *J Mol Evol*, 43(3):304–311, September 1996. 50

B. Rannala and Z. Yang. Phylogenetic inference using whole genomes. *Annual review of genomics and human genetics*, 9(1):217–231, 2008. 119, 120

C. E. Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems 12*, pages 554–560. MIT Press, 2000. 2, 24, 35, 53, 55

D. J. Reiss, N. S. Baliga, and R. Bonneau. Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. *BMC bioinformatics*, 7(1):280+, June 2006. 35, 68

S. Richardson and P. J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997. 55

C. P. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation (Springer Texts in Statistics)*. Springer Verlag, New York, 2nd edition, June 2007. 35

C. P. Robert, J.-M. M. Cornuet, J.-M. M. Marin, and N. S. Pillai. Lack of confidence in approximate Bayesian computation model choice. *Proceedings of the National Academy of Sciences of the United States of America*, 108(37):15112–15117, Sept. 2011. 125

H. Robins and W. H. Press. Human microRNAs target a functionally distinct population of genes with AT-rich 3' UTRs. *Proceedings of the National Academy of Sciences*, 102(43):15557–15562, Oct. 2005. 71

A. Rodriguez, D. B. Dunson, and A. E. Gelfand. The nested dirichlet process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008. 91

D. Rognan. Chemogenomic approaches to rational drug design. *British journal of pharmacology*, 152(1):38–52, Sept. 2007. ISSN 0007-1188. 79, 81

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelli-*

gence, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6. 2

S. Roweis. Handwritten digits. URL `http://cs.nyu.edu/~roweis/data.html`. 61

A. P. Russ and S. Lampel. The druggable genome: an update. *Drug Discovery Today*, 10(23-24): 1607–1610, Dec. 2005. 79

N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987. 46

R. Savage, K. Heller, Y. Xu, Z. Ghahramani, W. Truman, M. Grant, K. Denby, and D. Wild. R/bhc: fast bayesian hierarchical clustering for microarray data. *BMC Bioinformatics*, 10(1): 242+, August 2009. 2, 35

B. Schölkopf and A. J. Smola. *Learning with kernels : support vector machines, regularization, optimization, and beyond.* MIT Press, 1st edition, Dec. 2002. 80

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978. 35

E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Rich probabilistic models for gene expression. *Bioinformatics*, 17(suppl_1):S243–252, June 2001. 35

E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34(2):166–176, 2003. 35

J. Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639–650, 1994. 19

P. Shafto, C. Kemp, V. Mansignhka, M. Gordon, and J. B. Tenenbaum. Learning cross-cutting systems of categories. In *Proceedings of the Twenty-Eighth Annual Conference of the Cognitive Science Society*, 2006. 90, 91, 92

L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–421–I–428 Vol.1, 2004. 2

T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, Mar. 1981. 82

G. K. Smyth. Limma: linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, editors, *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, New York, 2005. 72

R. Sokal and C. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958. 45

E. M. Southern. Detection of specific sequences among dna fragments separated by gel electrophoresis. *Journal of molecular biology*, 98(3):503–517, Nov. 1975. 32

P. Stafford. *Methods in microarray normalization.* CRC Press, 2008. 34

M. A. Suchard, R. E. Weiss, and J. S. Sinsheimer. Bayesian selection of continuous-time markov chain evolutionary models. *Mol Biol Evol*, 18(6):1001–1013, 2001. 117

M. A. Suchard, J. S. Kitchen, Christina M.R.and Sinsheimer, and R. E. Weiss. Hierarchical phylogenetic models for analyzing multipartite sequence data. *Systematic Biology*, 52(5):649–664, 2003. 110

E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–605–I–612 vol.1, 2003. 2

F. Supek, M. Bosnjak, N. Skunca, and T. Smuc. Revigo summarizes and visualizes long lists of gene ontology terms. *PLoS ONE*, 6(7):e21800, 07 2011. 77

M. G. Tadesse, N. Sha, and M. Vannucci. Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association*, 100(470):602–618, 2005. 54, 55

K. Tamura and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial dna in humans and chimpanzees. *Mol Biol Evol*, 10(3):512–526, May 1993. 40

Y. Tateno, N. Takezaki, and M. Nei. Relative efficiencies of the maximum-likelihood, neighbor-joining, and maximum-parsimony methods when substitution rate varies with site. *Mol Biol Evol*, 11(2):261–77, 1994. 51

S. Tavaré. *Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences*, volume 17, pages 57–86. Amer Mathematical Society, 1986. 40

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2003. 2

A. E. Teschendorff, Y. Wang, N. L. Barbosa-Morais, J. D. Brenton, and C. Caldas. A variational bayesian mixture modelling framework for cluster analysis of gene-expression data. *Bioinformatics*, 21(13):3025–3033, July 2005. 35

L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4): 1701–1728, 1994. 50, 117

U.S. National Library of Medicine. DNA structure. URL `http://ghr.nlm.nih.gov/handbook/basics/dna`. 28

T. Van Laarhoven, S. B. Nabuurs, and E. Marchiori. Gaussian interaction profile kernels for predicting drug-target interaction. *Bioinformatics*, 27(21):3036–3043, Nov. 2011. 81

M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference.* Now Publishers Inc., Hanover, MA, USA, 2008. 25

J. D. Watson and F. H. Crick. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, Apr. 1953. 28

J. D. Watson, T. A. Baker, S. P. Bell, A. Gann, M. Levine, and R. Losick. *Molecular Biology of the Gene, Sixth Edition.* Benjamin Cummings, 6 edition, 2008. 26

D. Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, (1):31–36, Feb. 1988. URL `http://dx.doi.org/10.1021/ci00057a005`. 81

wikipedia, a. URL `http://en.wikipedia.org/wiki/DNA_microarray`. 32

wikipedia, b. URL `http://en.wikipedia.org/wiki/RNA_splicing`. 31

wikipedia, c. URL `http://en.wikipedia.org/wiki/Microarray_analysis_techniques`. 33

wikipedia, d. URL `http://en.wikipedia.org/wiki/Transcription_(genetics)`. 31

P. Willett. Similarity-based virtual screening using 2D fingerprints. *Drug discovery today*, 11(23-24): 1046–1053, Dec. 2006. 81

X. Xia, E. G. Maliski, P. Gallant, and D. Rogers. Classification of Kinase Inhibitors Using a Bayesian Model. *Journal of Medicinal Chemistry*, 47(18):4463–4470, Aug. 2004. ISSN 0022-2623. 80

Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Infinite hidden relational models. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Arlington, Virginia, 2006. AUAI Press. 53

Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa. Prediction of drugtarget interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, 24 (13):i232–i240, July 2008. 80

Y. Yamanishi, M. Kotera, M. Kanehisa, and S. Goto. Drug-target interaction prediction from chemical, genomic and pharmacological data in an integrated framework. *Bioinformatics*, 26(12): i246–i254, June 2010. 81, 82

Z. Yang. Among-site rate variation and its impact on phylogenetic analyses. *Trends in Ecology and Evolution*, 11(9):367–372, September 1996. 51, 52

Z. Yang. Maximum likelihood estimation on large phylogenies and analysis of adaptive evolution in human influenza virus a. *J Mol Evol*, 51(5):423–432, November 2000. 48

Z. Yang. *Computational Molecular Evolution (Oxford Series in Ecology and Evolution)*. Oxford University Press, USA, 2006. 114

Z. Yang and B. Rannala. Bayesian phylogenetic inference using dna sequences: a markov chain monte carlo method. *Mol Biol Evol*, 14(7):717–724, July 1997. 50

K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, October 2001a. 35

K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo. Model-based clustering and data transformations for gene expression data. Technical report, University of Washington, 2001b. 35

K. Y. Yeunga, K. M. Dombekb, K. Loa, J. E. Mittlera, J. Zhuc, E. E. Schadtd, R. E. Bumgarnera, and A. E. Raftery. Construction of regulatory networks using expression time-series data of a genotyped population. *PNAS*, 108(48):19436–19441, November 2011. 68

K. Y. Yip, R. P. Alexander, K.-K. Yan, and M. Gerstein. Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data. *PLoS ONE*, 5(1):e8121+, Jan. 2010. 35

P. D. Zamore and B. Haley. Ribo-gnome: The Big World of Small RNAs. *Science*, 309(5740): 1519–1524, Sept. 2005. 70

M. Zelwer and V. Daubin. Detecting phylogenetic incongruence using bionj: an improvement of the ild test. *Molecular phylogenetics and evolution*, 33(3):687–693, 2004. 109

CURRICULUM VITAE

Dazhuo Li

Computer Engineering and Computer Science Department
University of Louisville
Louisville, KY 40292
dazhuo.li@louisville.edu

Education

Ph.D., Computer Science and Engineering, (expected May, 2012)
University of Louisville, Louisville, KY, USA
Dissertation Title: Bayesian Nonparametric Clusterings in Relational and High-dimensional Settings with Applications in Bioinformatics
Advisor: Eric Rouchka, D.Sc.

M.SE., Computer Science and Software Engineering, 2003–2006
Zhejiang University, Hangzhou, Zhejiang, China

B.S., Mechanical Engineering and Automation, 1998–2002
Zhejiang University, Hangzhou, Zhejiang, China

Professional position

Software Engineer, Apr. 2012 (expected beginning date)
Cisco Systems, Inc. USA

Software Engineer, Spring 2006 (Brief)
Motorola, Inc. Hangzhou Research and Development Center, Hangzhou, Zhejiang, China

Software Engineer, 2004–2006
ZyXEL Communications Corp. Wuxi Research and Development Center, Jiangsu, China

Books

1. Dazhuo Li, Hai Liu. Core Eclipse: Rich Client Platform, User Interface and Web Application Development. ISBN:9787115158369. *Posts & Telecom Press*, 2007.

Refereed Publications

2. Dazhuo Li, Patrick Shafto. Bayesian Hierarchical Cross-Clustering. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Geoffrey Gordon, David Dunson, and Miroslav Dudk, eds. JMLR W&CP, vol. 15, 2011.
3. Dazhuo Li, Fahim Mohammad, Eric Rouchka. A Bayesian Nonparametric Model for Joint Relation Integration and Domain Clustering. In *Proceedings of the 9th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2010.
4. Dazhuo Li, Eric Rouchka, Patrick Shafto. Phylogenomic Analysis Using Bayesian Congruence Measuring. In *Proceedings of 2nd ISCA International Conference on Bioinformatics and Computational Biology (BICoB)*, 2010.
5. Dar-jen Chang, Nathaniel Jones , Dazhuo Li, Ming Ouyang, Rammohan Ragade. Compute Pairwise Euclidean Distances of Data Points with GPUs. In *Proceedings of the IASTED International Symposium on Computational Biology and Bioinformatics (CBB)*, 2008.

Journal and Conference Abstracts

6. Dazhuo Li, Eric Rouchka. Integrative Biclustering of Heterogeneous Datasets using a Bayesian Nonparametric Model With Application to Chemogenomics. In *BMC Bioinformatics 2011, 12(Suppl 7):A6 (5 August 2011).*
7. Dazhuo Li, Eric Rouchka. Bayesian Cross-Clustering for Microarray Data Analysis. In *The 7th Annual Conference of the MidSouth Computational Biology and Bioinformatics Society (MCBIOS)*, 2010.

Working Papers

8. Dazhuo Li, Antonio Badia. Query Language with Prefix Quantifier. In *submission.*
9. Dazhuo Li, Eric Rouchka, Patrick Shafto. Infinite Relational Model with Feature Selection. In *preparation.*
10. Dazhuo Li, Eric Rouchka, et al. Host Dependent Control of Inflammation and Colon Cancer by Gut Microbiome in a Mouse Model. In *preparation.*
11. Dazhuo Li, Eric Rouchka, et al. Detecting Potential Biomarkers for Ionizing Radiation Exposure. In *preparation.*

Conference Presentations

1. Bayesian Hierarchical Cross-Clustering. *The 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Ft. lauderdale FL. Apr. 2011
2. Integrative Biclustering of Heterogeneous Datasets using a Bayesian Nonparametric Model With Application to Chemogenomics. *UT-ORNL-KBRIN Bioinformatics Summit 2011.*, Memphis, TN. Mar. 2011
3. Bayesian Nonparametric Model for Joint Relation Integration and Domain Clustering. *The 9th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Washington DC. Dec. 2010
4. Phylogenomic Analysis Using Bayesian Congruence Measuring. *The 2nd ISCA International Conference on Bioinformatics and Computational Biology (BICoB)*, Honolulu, Hawaii. Mar. 2010

5. Bayesian Cross-Clustering for Microarray Data Analysis. *The 7th Annual Conference of the MidSouth Computational Biology and Bioinformatics Society (MCBIOS)*, Jonesboro, Arkansas. Feb. 2010

6. Compute Pairwise Euclidean Distances of Data Points with GPUs. *The IASTED International Symposium on Computational Biology and Bioinformatics (CBB)*, Orlando, Florida. Nov. 2008