Statistical Relational Learning with Nonparametric Bayesian Models

Dissertation im Fach Informatik
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

von
Zhao Xu

Tag der Einreichung: 04.06.2007
Tag der mündlichen Prüfung: 25.07.2007

Berichterstatter:
Prof. Dr. Hans-Peter Kriegel, Ludwig-Maximilians-Universität München
Prof. Dr. Tobias Scheffer, Max-Planck-Institut für Informatik
Dr. Volker Tresp, Siemens AG, München

Abstract

Statistical relational learning analyzes the probabilistic constraints between the entities, their attributes and relationships. It represents an area of growing interest in modern data mining. Many leading researches are proposed with promising results. However, there is no easily applicable recipe of how to turn a relational domain (e.g. a database) into a probabilistic model. There are mainly two reasons. First, structural learning in relational models is even more complex than structural learning in (non-relational) Bayesian networks due to the exponentially many attributes an attribute might depend on. Second, it might be difficult and expensive to obtain reliable prior knowledge for the domains of interest. To remove these constraints, this thesis applies nonparametric Bayesian analysis to relational learning and proposes two compelling models: Dirichlet enhanced relational learning and infinite hidden relational learning.

Dirichlet enhanced relational learning (DERL) extends nonparametric hierarchical Bayesian modeling to relational data. In existing relational models, the model parameters are global, which means the conditional probability distributions are the same for each entity and the relationships are independent of each other. To solve the limitations, we introduce hierarchical Bayesian (HB) framework to relational learning, such that model parameters can be personalized, i.e. owned by entities or relationships, and are coupled via common prior distributions. Additional flexibility is introduced in a nonparametric HB modeling, such that the learned knowledge can be truthfully represented. For inference, we develop an efficient variational method, which is motivated by the Pólya urn representation of DP. DERL is demonstrated in a medical domain where we form a nonparametric HB model for entities involving hospitals, patients, procedures and diagnoses. The experiments show that the additional flexibility introduced by the nonparametric HB modeling results in a more accurate model to represent the dependencies between different types of relationships and gives significantly improved prediction performance about unknown relationships.

In infinite hidden relational model (IHRM), we apply nonparametric mixture modeling to relational data, which extends the expressiveness of a relational model by introducing for each entity an infinite-dimensional hidden variable as part of a Dirichlet process (DP) mixture model. There are mainly three advantages. First, this reduces the extensive structural learning, which is particularly difficult in relational models due to the huge number of potential probabilistic parents. Second, the information can globally propagate in the ground network defined by the relational structure. Third, the number of mixture components for each entity class can be optimized by the model itself based on the data. IHRM can be applied for entity clustering and relationship/attribute prediction, which are two important tasks in relational data mining. For inference of IHRM, we develop four algorithms: collapsed Gibbs sampling with the Chinese restaurant process, blocked Gibbs sampling with the truncated stick breaking construction (SBC), and mean-field inference with truncated SBC, as well as an empirical approximation. IHRM is evaluated in three different domains: a recommendation system based on the MovieLens data set, prediction of the functions of yeast genes/proteins on the data set of KDD Cup 2001, and the medical data analysis. The experimental results show that IHRM gives significantly improved estimates of attributes/relationships and highly interpretable entity clusters in complex relational data.

Acknowledgments

This thesis is a conclusion of my research work under the joint Ph.D. program between the KDD group at the Institute of Computer Science, University of Munich and the Department of Learning Systems, Corporate Technology, Siemens AG. Throughout my doctoral study, many people helped to guide me and support me during the journey that eventually led to this thesis.

First, I am deeply indebted to my supervisor, Prof. Dr. Hans-Peter Kriegel. Without his guidance, encouragement and tremendous support, this thesis would not be possible.

I owe a lot to my co-supervisor at Siemens AG, Dr. Volker Tresp, who introduced me to the fascinating area of machine learning and has profound influence on me as a researcher. The research of this thesis is carried out through his vision in statistical machine learning. I truly appreciate him for his advices, wisdom, encouragement, cheerfulness and patience. I feel extremely fortunate to have the opportunity to work with him.

I would like to thank Prof. Dr. Tobias Scheffer from the Max Planck Institute for Computer Science for his invaluable advices and insightful comments on my thesis work. I would also like to thank Prof. Dr. Alexander Knapp and Prof. Dr. Hans Jürgen Ohlbach for their patient instructions on my oral examinations.

I am very grateful to Prof. Dr. Bernd Schürmann, head of the Department of Learning Systems at Siemens AG, for his constant support to my research.

I also appreciate all the friendships, encouragements and support from colleagues, while the following list is undoubtedly incomplete: Dr. Clemens Otte, Christof Störmann, Stefan Hagen Weber, Dr. Kai Yu, Dr. Shipeng Yu, Mrs. Susanne Grienberger, Mrs. Christa Singer, Anton Maximilian Schäfer, Dr. Stefan Brecheisen, Franz Krojer, Karsten Borgwardt, Dr. Peer Kröger, Dr. Matthias Schubert, Dr. Ralph Grothmann, Dr. Christoph Tietz and Dr. Kai Heesche.

Finally, it is impossible to have my research career without the love and support from my family. This thesis is dedicated to them.

Zhao Xu
Munich, Germany
October, 2007

Contents

List of Figures

List of Tables

Part I

Preliminaries

Chapter 1

Introduction

There are many collections of relational data in diverse areas. The construction of statistical models on this kind of data has been well studied in statistics and machine learning communities. Generally these approaches are only capable of handling data in flat form, i.e. each instance has the same set of attributes and is assumed as independently and identically distributed (i.i.d.). It is obvious that there is a lack of the conceptions about entities and relationships in these approaches. However the real-word data always consists of multiple types of entities, their distinct attributes, and relationships between entities of the same/different types. In recent years, researchers realize the importance of the relational natures of the data and introduce many advanced statistical models with compelling results. Considering the central importance of relationships, these novel approaches are called *statistical relational learning* (SRL).

In the preliminary researches of SRL, the relationships are explored and encoded in an implicit way. For example, in the *probabilistic relational model* (PRM) introduced in (Friedman et al., 1999; Getoor et al., 2001), the relationships are represented as the reference slots and the information in relationships is encoded as: an attribute can probabilistically depend on not only the attributes of the same entity, but also the attributes of related entities. PRM can be viewed as a milestone in the development of SRL. In latter works, researchers start to explicitly incorporate the relationships into the statistical models. Typically Getoor et al. (2003) introduced link uncertainty models which encode the relationships with reference uncertainty and existence uncertainty mechanisms. Each relationship is represented as a multinomial variable or a binomial variable and can be involved in probabilistic models as any other attributes.

In this thesis we apply advanced Bayesian techniques to relational learning in order to form refined statistical models to capture the probabilistic dependencies between entities and relationships. The models are expected to be more expressive, more easily applicable and more computationally efficient than the previous approaches. As a result, we propose two novel and principled developments: the first one, *Dirichlet enhanced relational learning* (DERL), extends nonparametric hierarchical Bayesian modeling to relational learning, the second one, *infinite hidden relational model* (IHRM), extends nonparametric mixture modeling to relational learning. To perform fast computation, various inference methods are explored. We demonstrate their performance on real-world applications and provide some evidence that our hope has been met.

1.1 Motivations and First Discussion of Our Models

In statistical relational learning, entities are individuals, which are distinct from one another despite the common features. Thus it makes more sense to represent the model parameters as personalized to entities, than to represent them as global. Assume a simple medical domain, where patients take procedures conditioned on their prime complaints. The parameters θ expressing the probability of prescribing a procedure given prime complaint, are modeled as global quantities. There are two important implications in the representation. First, the probability of a procedure is identical for all patients with the same prime complaint. Second, procedures for a patient are modeled as independent given his prime complaint and the global parameters θ, such that knowledge about prescribed procedures does not influence the selection of subsequent procedures. Both implications are not realistic. Patients are truly unique which might be obvious to the attending physicians but which is impossible to be represented in a probabilistic model with global parameters. Given a prime complaint, a physician might select a personalized treatment strategy. Additionally, the procedures taken by a patient are related. The prescribed procedures influence the selection of future procedures since the physician often make decision of the coming procedures based on the previous ones. A typical solution for the problem in Bayesian analysis is hierarchical Bayesian (HB) framework, i.e. each patient i has his own parameters θ_i, which share a common prior distribution.

In a HB model, the parameterized prior distribution obtains central importance since it must not only be able to represent ones' prior belief but also be flexible enough to represent the learned posterior, which might not be in the same family of distributions. Thus it is advantageous to specify the prior distribution in a flexible nonparametric form, technically as a sample distribution from a Dirichlet process (DP). Although we can still implement our vague prior belief in form of the parameters of the DP, the learned posterior can be very rich. Due to the central importance of DP in the model, we name it as *Dirichlet enhanced relational learning* (DERL), which can be viewed as a relational extension of nonparametric hierarchical Bayesian modeling. As an extra advantage, DERL provides an elegant way to capture the semantic information about hierarchical classes in relational data: the super-class can be represented as prior and sub-class can be represented as samples drawn from its super-class, thus each subclass is distinct, but shares some common features via the prior distribution. The class-instance semantics can be modeled in an equivalent way.

The other model presented in the thesis is *infinite hidden relational model* (IHRM), which is motivated by some constraints in existing relational models. First, there is no easily applicable recipe of how to turn a relational domain (e.g. a database) into a probabilistic model. One has to implement extensive structural learning, which is even more expensive than in (non-relational) Bayesian networks due to the huge number of potential parents. For example, whether I get a disease might depend on genetic disposition of my great-grandfather. Secondly, in some statistical relational approaches it is necessary to have reliable prior knowledge, for example, the clauses in Markov logic networks. However this kind of information is generally expensive and not easy to obtain. Thirdly, in present relational approaches the inference is computationally expensive over the data with missing values, complex inference approaches are required, e.g. loopy belief propagation.

To remove these constraints, we first propose a finite hidden relational model. It introduces for each entity a latent variable, which is the only parent of the attributes of the entity, and is a parent of relationships the entity participates. The ground network for the instantiated entities and relationships forms a network of latent variables, across which information can propagate. For example, the information about genetic disposition of my great-grandfather can propagate to me via the latent variables of my father and my grandfather. Since each entity class has the different number of states in its latent variable and the number varies with increasing entities, it is natural to expect the model to determine the number of latent states in a self-organized way. This is possible by embedding the model in Dirichlet process (DP) mixture modeling, which can be simply interpreted as a mixture model with an infinite number of mixture components. The term *infinite* does not mean there are infinite latent states for each entity class, but the number is not specified in advance. The model, based on the data, automatically reduces the complexity to an appropriate finite number of components. The combination of the hidden relational model and the DP mixture model is infinite hidden relational model (IHRM). For inference of IHRM, we first develop a collapsed Gibbs sampler with Chinese restaurant process (CRP). Considering the slow mixing of Markov chain in CRP, we propose other inference methods, including blocked Gibbs sampler with truncated stick-breaking construction (SBC) and mean-field approximation with truncated SBC, as well as a memory-based empirical method.

1.2 Thesis Overview

The thesis is organized as follows:

Part 1 deals with the preliminaries.

Chapter 1 gives a brief introduction to the thesis, including motivations, main contributions, outline and so on.

Chapter 2 provides a short overview of statistical relational learning (SRL), including: the motivations, the major types of modeling approaches, some leading frameworks (probabilistic relational models, directed acyclic probabilistic entity relationship models, Bayesian logic programming), and prime tasks (object identification, object ranking, object classification/clustering, relationship prediction, relationship classification and so on).

Part 2 introduces nonparametric hierarchical models to relational learning.

Chapter 3 reviews the Bayesian and hierarchical Bayesian models. We first discuss the principles of these models. Then the differences from the classical statistical models are introduced. The following topic is the exponential family distributions and their major properties. The definition of exchangeability of samples is also introduced. Finally, some inference approaches are discussed.

Chapter 4 introduces the nonparametric hierarchical Bayesian models. First, we discuss the limitations of parametric Bayesian modeling. Then we provide a brief introduction to nonparametric Bayesian modeling. Next we discuss a well-known nonparametric technology, Dirichlet process (DP), and present its application in the hierarchial Bayesian framework. Finally, the inference methods (Gibbs sampling and variational approximation) are described using the Pólya urn representation of DP.

Chapter 5 extends the nonparametric hierarchical Bayesian modeling to relational learning and introduces Dirichlet enhanced relational model (DERL). We first discuss the limitations of present relational models. Then a hierarchical Bayesian modeling method is introduced with the goal of removing these shortcomings. To involve additional flexibility to the model, we embed it in a nonparametric framework in order that the learned knowledge can be truthfully represented via nonparametric priors. After that, we discuss the method of how to model the dependencies between different types of relationships and introduce a smoothing technology to overcome the issue of overfitting. Next, an efficient variational inference method is provided for probabilistic reasoning. Finally experimental analysis on medical data is given to evaluate the performance of DERL.

Part 3 presents nonparametric mixture models to relational learning.

Chapter 6 introduces finite mixture models. We first review the motivations of the mixture models and their major applications. Then a finite mixture model is introduced in empirical Bayesian framework. Parameter estimation and predictive inference are discussed. After that, we introduce the finite mixture models in full-Bayesian framework. The corresponding inference approaches are discussed, including Gibbs sampling method and variational approximation. Finally, we discuss parameter estimation and predictive inference methods.

Chapter 7 presents the Dirichlet process (DP) mixture model. First, we introduce the definition of the model, and then describe the different representations of Dirichlet process, including the Chinese restaurant process (CRP) and the stick breaking construction (SBC). Next we discuss two Gibbs sampling algorithms for inference, one is collapsed Gibbs sampling with CRP, the other is blocked Gibbs sampling with SBC. To improve the computational efficiency, the mean-field based inference method is also discussed. Finally, the prediction methods are introduced with the corresponding inference techniques.

Chapter 8 extends nonparametric mixture modeling to relational learning and proposes infinite hidden relational model (IHRM). First, we discuss the motivations. Then, we introduce a finite hidden relation model with an example of movie recommendation system. Next, the expressive power of the finite model is enhanced by combining it with nonparametric Bayesian modeling. After that, we present diverse inference methods, including collapsed Gibbs sampling with CRP, blocked

Gibbs sampling with truncated SBC, mean-field based approximation with truncated SBC, and an relational memory-based approximation. The corresponding predictive inference methods are also provided. Finally, the performance of IHRM is demonstrated in three applications: movie recommendation system, function prediction of genes and medical data analysis.

Part 4 concludes the thesis.

Chapter 9 summarizes the major results of the thesis and discusses some future research directions.

Chapter 2

Statistical Relational Learning

2.1 Introduction

Statistical relational learning (SRL) has recently received increasing attention (Dzeroski & Lavrac, 2001; Raedt & Kersting, 2003) and plays an important role in modern data mining (Wrobel, 2001). The reason is that relevant information is not only contained in attributes describing properties of objects but also in relationships between objects. In particular, a typical domain of interest might consist of objects, their attributes and their relationships. Most machine learning approaches have tried to select a representation in which a relational representation could be avoided by constructing appropriate derived features (propositionalization). It is obvious that the full information contained, e.g. in a relational data base, could not be completely represented and exploited by propositionalization. To solve these limitations, statistical relational models are developed to encode relational information in a principled way, which combines with various knowledge representations to model multi-relational, heterogeneous and semi-structured data.

Statistical relational learning is the intersection of research in graphic models, logic representations and probabilistic theories. There are mainly four categories of modeling approaches (Getoor, 2005).

1. The first family of approaches concerns the combination of relational database models and graphical models. For example, the *probabilistic relational model* (PRM) introduced in (Friedman et al., 1999; Getoor et al., 2001), formulates a probabilistic framework for database relational model. The *directed acyclic probabilistic entity relationship model* (DAPER) introduced in (Heckerman et al., 2004), integrates Bayesian analysis with entity-relationship database representation. The *relational Markov network* (RMN) introduced in (Taskar et al., 2002), proposes a discriminative model for the relational data via integrating Markov network.

2. The second family of modeling approaches deals with the combination of first-order logic languages and graphical models. For example: the *Bayesian logic program* introduced in (Kersting & Raedt, 2000), is an extended Bayesian network with definite clause logic (i.e. pure Prolog). The *Markov logic network* proposed in (Richardson &

Domingos, 2006), is an undirected graphical model. It can be viewed as a first-order knowledge base, where each clause is associated with a weight.

3. The third family of approaches concerns functional programming with stochastic execution.

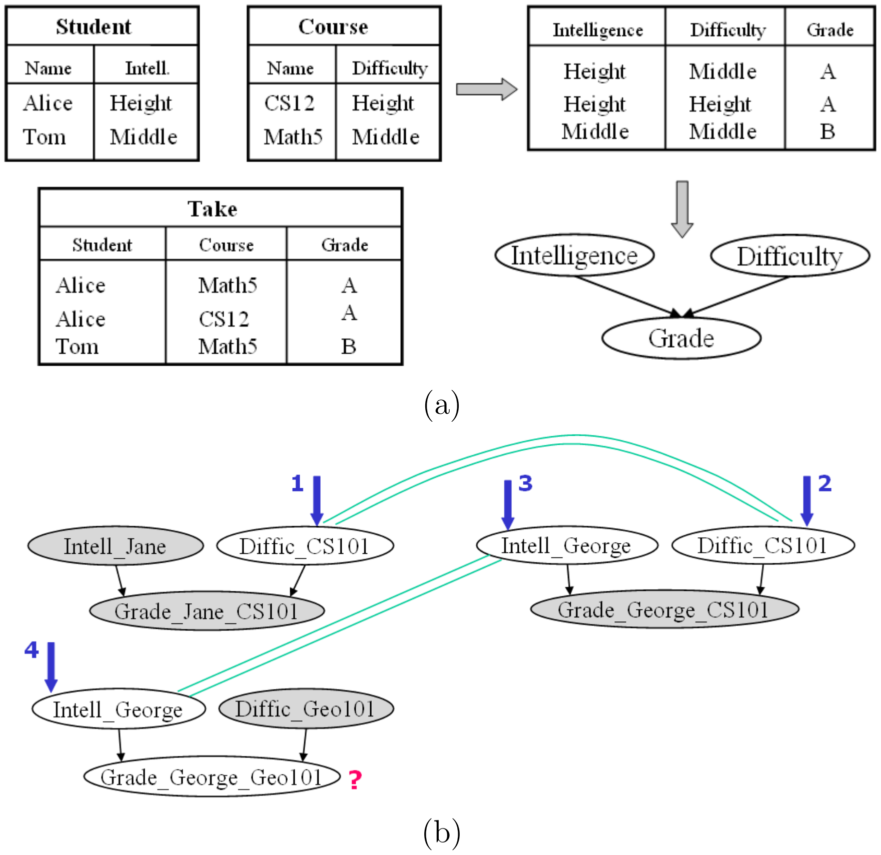
4. Fourth, the combination of dynamic probabilistic models and logic representations, e.g. the *dynamic probabilistic relational model* in (Sanghai et al., 2003) and the *relational Markov model* in (Anderson et al., 2002).

All these modeling technologies attempt to combine knowledge representation languages with statistical models such that the full information in the domain of interest can be modeled in an elegant way. As a result, robust and accurate probabilistic reasoning can be performed. These SRL approaches are widely applied in the contexts of text mining, web mining, gene analysis, customer service, social network and natural language processing as well as other complex domains. The major tasks of SRL include object identification, object ranking, object classification/clustering, relationship prediction, relationship classification and attribute prediction, as well as subgraph discovery, graph classification and so on.

2.2 Motivation

There are mainly two motivations for statistical relational learning. From the data mining point of view, typical data mining approaches look for patterns in a database only within a single relation. However in real-world cases, the domain of interest generally consists of many classes of objects and relations, e.g. multiple tables in a database, thus it is necessary to integrate data from multiple relations into a single table before executing a particular data mining approach. Unfortunately, this integration requires much thought and effort, more important, some essential patterns may be missed after flattening the data. Therefore it is better to analyze the data directly from a multi-relation database, without the need to transfer the data into a single table. Secondly, from the statistical machine learning point of view, most statistical methods assume that the data are independently and identically distributed (except for the dynamic models, e.g. hidden Markov model), but actually, the samples may depend on each other. For example, if a student obtains high grade in a course (say data mining), then he very likely obtains high grade in a related course (say machine learning). Statistical relational learning is a solution for the two limitations, which attempts to combine knowledge representation languages with statistical models in order to directly model the data with complex relations.

Let us illustrate via a particular example on school domain. There are three tables in a school database: Student, Course, and Take. The table Student stores the information about student intelligence. The table Course stores the information about the course difficulty. The table Take stores the information about student grades. Typically, one student takes several courses. A general machine learning approach, e.g. Bayesian network, first combines the three tables into a single one, then the records in the table



Student	
Name	Intell.
Alice	Height
Tom	Middle

Course	
Name	Difficulty
CS12	Height
Math5	Middle

Intelligence	Difficulty	Grade
Height	Middle	A
Height	Height	A
Middle	Middle	B

Take		
Student	Course	Grade
Alice	Math5	A
Alice	CS12	A
Tom	Math5	B

(a)

(b)

Figure 2.1: Motivations of statistical relational learning illustrated on a school domain. (a) The learning process of Bayesian network. (b) Ground network of three additional records. The numbers show the procedure of probabilistic reasoning about George's grade at the course Geo101.

are viewed as i.i.d. samples and a Bayesian network is learned on the flattened data. The procedure is shown as Figure 2.1(a). Assume that there is additional information about: Jane takes course CS101; George takes courses CS101 and Geo101. Furthermore, assume that Jane's intelligence, Jane and George's grades at the course CS101 are known, and we are interested in George's grade at the course Geo101. In the traditional Bayesian network approach, it is impossible to infer the probability of George's grade at the course Geo101, since George's intelligence is unknown. However, in relational learning, we notice the fact that both Jane and George take the course CS101, thus the three records are linked in a single network. The probabilistic inference can be implemented in the way shown as Figure 2.1(b): from Jane's intelligence and Jane's grade at the course CS101, we obtain the probability of difficulty of the course CS101. Then the information is transferred to the right subnetwork, and the probability of intelligence of George is obtained, finally, George's grade at the course Geo101 is inferred via integrating all information, not only including himself, but also including the related persons, e.g. Jane.

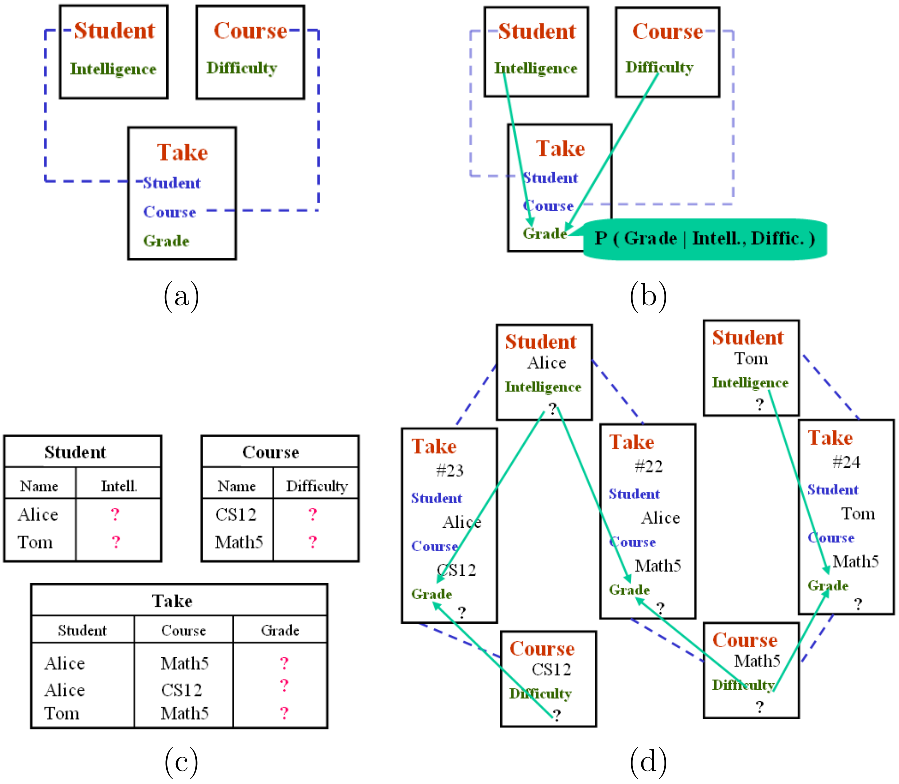
In a word, statistical relational learning integrates the strength of relational logic representations such that the resulting models can perform robust and accurate reasoning and learning in complex domains.

2.3 SRL Models

During the past decade, many SRL models are introduced in various applications. The leading frameworks include: probabilistic relational model (Friedman et al., 1999; Getoor et al., 2001) and its extension with link uncertainty (Getoor et al., 2003), directed acyclic probabilistic entity relationship model (Heckerman et al., 2004), multi-relational data mining approaches (Liu et al., 2005), Bayesian logic programming (Kersting & Raedt, 2000), and Markov logic network (Richardson & Domingos, 2006), etc. In this section, we briefly introduce some of the leading researches.

2.3.1 Probabilistic Relational Model

Probabilistic relational model (PRM) (Friedman et al., 1999; Getoor et al., 2001) describes a probabilistic formulation for a relational data base. It integrates Bayesian network with the database structure representation *relational model* (Ullman & Widom, 1997). PRM is a milestone in the development of statistical relational learning. Koller and Pfeffer (1997) proposed object-oriented Bayesian network, which extends the Bayesian network with the concepts of *classes*, *objects*, and their *attributes*. The model can be viewed as the initial work of PRM. Koller and Pfeffer (1998) introduced the *probabilistic frame-based system*, which combines the frame-based knowledge representation with Bayesian network to model organizational structure of a large complex domain. It provides more expressive power than traditional Bayesian network. With these early researches, Friedman et al. (1999) developed *probabilistic relational model*, which is a compact and effective language to describe a statistical formulation over a typed relational domain. A PRM models the probabilistic uncertainty over the attributes of objects and relationships between objects.



Student	
Name	Intell.
Alice	?
Tom	?

Course	
Name	Difficulty
CS12	?
Math5	?

Take		
Student	Course	Grade
Alice	Math5	?
Alice	CS12	?
Tom	Math5	?

Figure 2.2: An example of PRM over school domain from (Getoor et al., 2001). (a) Relational schema specifying the classes, descriptive attributes and reference slots. (b) Dependency structure and local probability model. (c) An example skeleton instantiating objects and relationships. (d) Ground Bayesian network which is obtained by applying the PRM template in (b) to the example skeleton in (c).

An attribute of an object depends on not only other attributes of the same object, but also the attributes of related objects. PRM provides a new perspective for data mining.

Probabilistic relational model is motivated from Bayesian network (BN). A BN is a graphical model to encode the probabilistic dependencies between variables, which provides an elegant formalism for representing and reasoning probabilistic uncertainty. The major advantage is that it exploits the underlying structure of the domain knowledge to represent the joint distribution in an effective way. However, BN lacks the concepts of objects and relationships. In many real-world applications, the domain of interest typically consists of objects, their attributes and relationships between them. This kind of underlying information can not be captured by a traditional Bayesian network. Generally, BN pre-processes the data into a flat representation, and then, the probabilistic dependencies are learned and reasoned. It is obvious that some important patterns are missing in the procedure of flattening the data. In addition, there is an important assumption in BN, i.e. the samples are independently and identically-distributed (i.i.d.). However, in more cases than not, the samples are linked together into a ground network via relationships. The information about relationships is helpful in making decision/prediction. For example, in a social network, the friendship between two persons influences the frequency and mode of communication between them. PRM is a framework integrating relational logic to overcome these limitations in a compact and natural way.

A probabilistic relational model consists of three components: relational schema, dependency structure and local probability model. Figure 2.2 shows an example on school domain. A relational schema describes data structure of the domain of interest. It consists of a set of classes, e.g. Student and Take. Each class is associated with a set of descriptive attributes and a set of reference slots. A descriptive attribute represents a particular property of objects in the class, e.g. Student.Intellignce, which specifies intelligence of a student. A reference slot describes a relationship between two classes, e.g. Take.Student, which specifies an instance in the class Take is related with an instance in the class Student. Figure 2.2(b) shows dependency structure and local probability model for the running example. The probabilistic dependencies are specified by the solid directed arcs, e.g. the arc from Student.Intellignce to Take.Grade specifies the fact that student's grade depends on his intelligence. An attribute can depend on other attributes of the same class, or the attributes of the related classes, e.g. a probabilistic parent of Take.Grade is Student.Intelligence, which is an attribute of the class Student which is related to the class Take. The local probability model can be a conditional probabilistic table for a discrete attribute, or a conditional probabilistic density function for a continuous attribute. In the school example, the local probability model is $P(Take.Grade|Student.Intelligence, Course.Difficulty)$.

A PRM is a probabilistic template on the domain of interest. It will be replicated on a particular skeleton. A skeleton specifies a possible relational structure of the domain and is an instantiation of objects and relationships for a schema. Figure 2.2(c) shows an example skeleton. Note, that the particular values of attributes in a skeleton can be unknown. Applying dependency structure defined by a PRM to an example skeleton, we obtain a ground Bayesian network, e.g. Figure 2.2(d), which represents the joint probability over all attributes and relationships in the skeleton. The probabilistic inference is finally

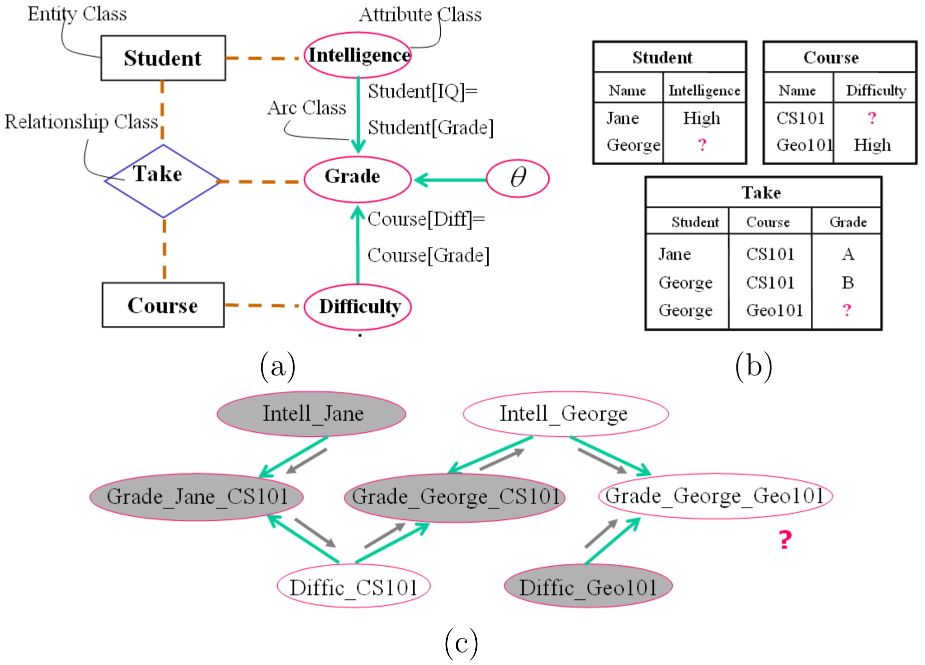


Figure 2.3: An example of DAPER model on the school domain from (Heckerman et al., 2004). (a) DAPER model. (b) Instantiated objects and relationships. (c) Ground Bayesian network. All information propagates to the attribute of interest, i.e. George's grade at the course Geo101. The grey arrows show the procedure of probabilistic inference.

performed on the ground network.

In summary, PRM integrates the relational database model with Bayesian network. An attribute can probabilistically depend on not only the attributes of the same object, but also the attributes of related objects. PRM is an important contribution in the development of statistical relational learning.

2.3.2 Directed Acyclic Probabilistic Entity Relationship Model

Directed acyclic probabilistic entity relationship model DAPER, introduced by Heckerman et al. (2004), is another leading framework in statistical relational learning, which extends Bayesian analysis with the database structure representation *entity-relationship model* (Ullman & Widom, 1997). The DAPER framework — the focus of this thesis— is particularly elegant in a Bayesian context since it encourages an explicit representation of model parameters and hyperparameters.

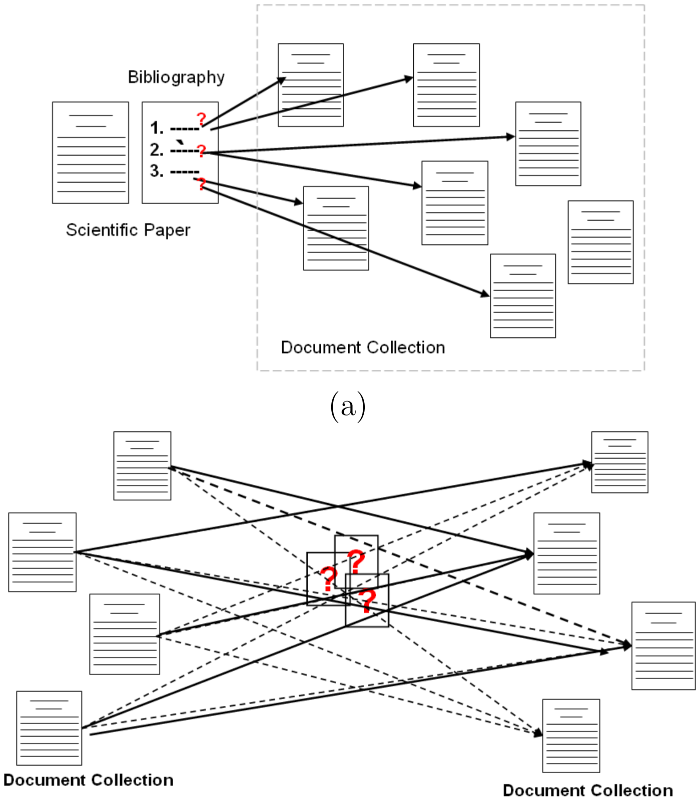
DAPER formulates a probabilistic framework for an entity relationship database representation. DAPER makes relationships first class objects in the modeling language, and encourages an explicit representation of conditional probabilistic distributions. A DAPER model consists of entity classes, relationship classes, attribute classes and arc classes, as well as local distribution classes and constraint classes. Figure 2.3(a) shows an example of a DAPER model on the school domain. The entity classes specify classes of objects in the real world, e.g. Student and Course shown as rectangles in Figure 2.3(a). The relationship class represents interaction among entity classes. It is shown as a diamond-shaped node with dashed lines linked to the related entity classes. For example, the relationship class Take(s, c) indicates that a student s takes a class c. Note, that the DAPER model assigns relationships the same importance as the entities. Attribute classes describe properties of entities or relationships. Attribute classes are connected to the corresponding entity/relationship class by a dashed line. For example, associated with courses is the attribute class Course.Difficulty and associated with the relationship class Take is the attribute class Take.Grade. The attribute class θ in Figure 2.3(a) represents the parameters specifying the probability of student's grade in different configurations (i.e. difficulty of courses and intelligence of students). It denotes a *global* attribute class, and is not associated with any entity class or relationship class. The arc classes shown as solid arrows from *parents* to *children* represent probabilistic dependencies among corresponding attributes. For example, the solid arrow from Student.Intelligence to Course.Grade specifies the fact that student's grade probabilistically depends on student's intelligence. A local distribution class for an attribute class is a specification from which local distributions for the attribute class can be constructed. As an example, the probabilistic distribution of Take.Grade conditioned on its parents is specified by a local distribution class (not shown in the figure) with the global parameters θ.

Based on the DAPER model (e.g. Figure 2.3(a)) and the instantiated entities and relationships (e.g. Figure 2.3(b)), a ground Bayesian network (e.g. Figure 2.3(c)) is generated in which probabilistic inference (e.g. belief propagation) can be performed. In the running example shown as Figure 2.3(c), all known information propagates to the unknown attribute of interest, i.e. George's grade at the course Geo101. Constraint classes specify how to derive ground Bayesian network from the corresponding DAPER model over the instantiated domain, e.g. the constraint $course[Difficulty] = course[Grade]$ indicates that in the ground network an arc should be introduced between attribute c.Difficulty and attribute Takes(s, c').Grade, only when $c = c'$. Thus it is forbidden to add a solid arrow from CS101.Difficulty to Take(George,Geo101).Grade.

In summary, DAPER framework makes relationships first class objects in the modeling language, and encourages an explicit representation of parameters and hyperparameters. It is particularly suited in a Bayesian context.

2.3.3 Relational Models with Structure Uncertainty

In some real-world applications, the relational structure itself is uncertain. Thus it is necessary to incorporate the relationships into the probabilistic models. Explicitly modeling the relationships can improve the expressive power of SRL, which make possible to build *full probabilistic models* on the domains of interest. We can not only predict the unknown relationships based on the other information, but also explicitly exploit the known relationships to predict unknown attributes of entities or other variables. Getoor et al. (2003) proposed two mechanisms to represent the relational uncertainty: one is *reference uncertainty*, the other is *existence uncertainty*. Figure 2.4 describes an example on the paper-citation domain. Assume that there is a scientific paper with three references, but the specific information about the references is unknown. Given a document collection,



(a)

(b)

Figure 2.4: Structure uncertainty on the paper-citation domain (Getoor et al., 2003). (a) Reference uncertainty modeling. (b) Existence uncertainty modeling.

it is natural to model each reference as a multinomial variable with as many states as the number of papers in the document collection. The value of the reference variable specifies which paper is cited. This kind of modeling strategy is called reference uncertainty. Assume another situation that the number of references of a paper is also unknown, and we only know that each paper can cite any other papers in a given document collection. Thus it is natural to associate with each paper $(N-1)$ Bernoulli variables, where N is the number of papers in the collection. The possible states for each variable are *exist* and *not-exist*, which specifies whether the paper represented by the variable is cited or not. This kind of modeling strategy is called existence uncertainty. Reference uncertainty is generally used in situations where one part of a relationship is certain, only the other part of the relationship is uncertainty. It is obvious that the complexity of reference uncertainty is much less than existence uncertainty, but the flexibility of existence uncertainty is much more than reference uncertainty.

Figure 2.5 describes reference uncertainty and existence uncertainty in the DAPER framework via a medical example, where Patient.PrimeComplaint is an attribute describing the prime complaint of the patient, Procedure.Id describes the identifier of the procedure, the relationship class Take(pa, pr) represents the fact that a patient pa receives a procedure pr. In existence uncertainty, a relationship class is associated with an auxiliary

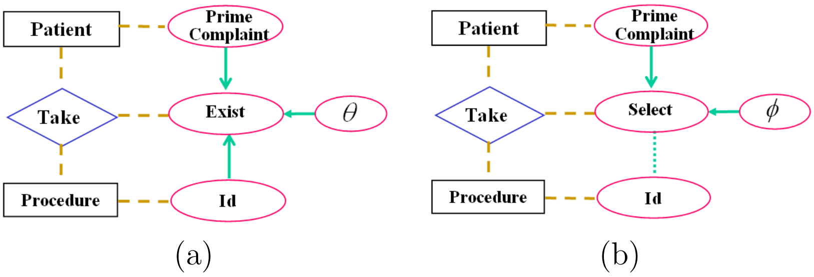
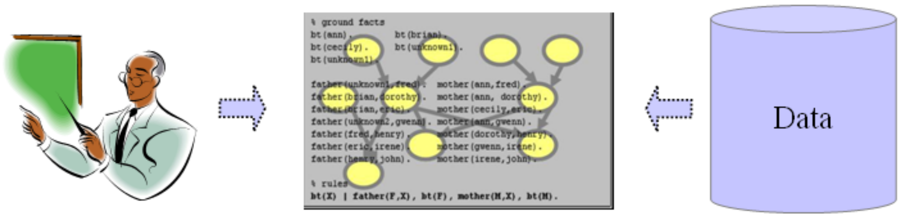


Figure 2.5: DAPER model with structure uncertainty over medical domain. (a) Existence uncertainty modeling. The auxiliary attribute Take.Exist is modeled as a binomial variable. $\theta_{e|pc,Id}$ denotes the parameters of the binomial distribution conditioned on Patient.PrimeComplaint and Procedure.Id. (b) Reference uncertainty modeling. The auxiliary attribute Take.Select is modeled as a multinomial variable with as many states as there are procedures. $\phi_{s|pc}$ denotes the parameters of the multinomial distribution conditioned on Patient.PrimeComplaint.

attribute *Exist* (e.g. Figure 2.5(a)) with two states, Yes/No. The attribute can be modeled as a binomial variable to represent the uncertainty of whether a procedure is taken by a patient. The global attribute $\theta_{e|pc,Id}$ represents the parameters of the distribution of Exist conditioned on prime complaint pc and procedure Id. In reference uncertainty, a relationship class is associated with an auxiliary attribute *Select* (e.g. Figure 2.5(b)) with as many states as there are possible procedures. The attribute Select is generally modeled as a multinomial variable. The global attribute $\phi_{s|pc}$ represents the parameters of the distribution of Select conditioned on prime complaint pc.

Existence uncertainty and reference uncertainty introduced by Getoor et al. (2003) are two important strategies to model the uncertainty about the relational structure. By explicitly incorporating the relationships into the probabilistic models, we can not only predict relationships themselves, but also use the relationship information to predict other variables of interest.

2.3.4 Bayesian Logic Programming



```
% ground facts
bt(ann).           bt(brian).
bt(cecily).        bt(unknown1).
bt(unknown1).

father(unknown1,fred).   mother(ann,fred).
father(brian,dorothy).   mother(ann, dorothy).
father(brian,eric).      mother(cecily,eric).
father(unknown2,gwenn).  mother(ann,gwenn).
father(fred,henry).      mother(dorothy,henry).
father(eric,irene).      mother(gwenn,irene).
father(henry,john).      mother(irene,john).

% rules
bt(X) | father(F,X), bt(F), mother(M,X), bt(M).
```

Figure 2.6: Bayesian logic programming (Kersting & Raedt, 2000) integrating domain expert knowledge and the data into probabilistic models.

Bayesian logic programming (BLP), introduced by Kersting and Raedt (2000), is another compelling framework to extend the expressive power of Bayesian network by intro-

Table 2.1: Associated conditional probabilistic distributions for the example Bayesian clause about children's height, where h, h_y and h_z denote the height of X, Y and Z (Kersting & Raedt, 2000).

mother(Y, X)	father(Z, X)	cpd
true	true	$N(h, \frac{1}{2}(h_y + h_z), 50)$
true	false	$N(h, \frac{1}{2}(h_y + 175), 50)$
false	true	$N(h, \frac{1}{2}(h_z + 175), 50)$
false	false	$N(h, 175, 50)$

ducing the concepts of objects and relationships. BLP integrates definite clause logic with Bayesian network by establishing a one-to-one mapping between ground atoms and random variables. There are mainly four advantages in BLP. First, it is easy to incorporate domain expert knowledge and data into the model (as Figure 2.6). The prior knowledge can be explicitly represented as pre-defined clauses. Second, BLP might have more expressive power than PRM, since the definite clause logic is able to represent more complex relationships than the relational database logic. Third, the resulting logical structure provides a deep insight into the domain of interest. Last, the model is more comprehensible whether in the reasoning process or in the final result.

Bayesian logic programming consists of a finite set of *Bayesian clauses* (BC), each of which can be intuitively viewed as a logic clause associated with a conditional probabilistic distribution, e.g. a Bayesian clause about children's height consists of a logic clause

$$height(X)|mother(Y, X), height(Y), father(Z, X), height(Z)$$

and an associated conditional probabilistic distribution shown as Table 2.1. This Bayesian clause specifies the probability distribution of height of X conditioned on the height of his parents Y and Z. More formally, Bayesian clauses are defined as:

$$A|A_1, \ldots, A_n; \quad P(A|A_1, \ldots, A_n).$$

Where A and $A_1, \ldots, A_n$ are Bayesian atoms and all atoms are (implicitly) universally quantified. The major differences between Bayesian and definite clauses include:

1. Each Bayesian predicate/atom r has an associated domain $D(r)$, e.g., $D(father) = D(mother) = \{true, false\}$ and $D(height) = \mathbb{R}$.

2. The symbol :- in definite clause is replaced with | to capture the idea of conditional probabilistic distribution.

Another important component in Bayesian logic programming is *combining rule*, which is used to integrate a finite set of conditional probabilistic distributions into a single one. Formally, it is defined as:

Definition 2.1 *A combining rule is an algorithm that integrates conditional probabilistic distributions*

$$\{P(A|A_{i,1}, \ldots, A_{i,n_i})\}_{i=1}^{M}$$

associated with a finite set of Bayesian clauses into a combined conditional probabilistic distribution

$$P(A|B_1, \ldots, B_N).$$

Where $\{B_1, \ldots, B_N\} = \cup \{A_{i,1}, \ldots, A_{i,n_i}\}_{i=1}^{M}$.

Note, that the combined Bayesian clauses should have the same head atom. Combining rules can be viewed as a generalization of the idea of canonical distributions. In summary, a Bayesian logic program is formally defined as:

Definition 2.2 *A Bayesian logic program $\mathcal{B}$ consists of a (finite) set of Bayesian clauses. For each Bayesian clause c there is exactly one conditional probability distribution $cpd(c)$ associated, and for each Bayesian predicate r there is exactly one combining rule $comb(r)$ associated.*

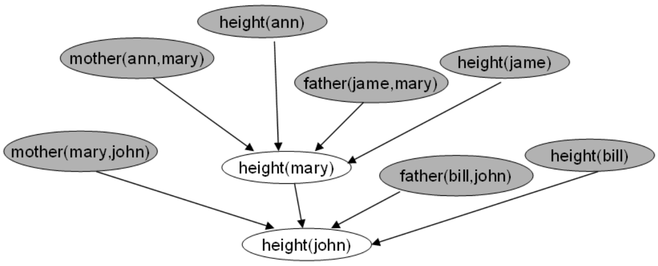


Figure 2.7: The dependency graph for the example about children's height with constants: ann, jame, mary, bill and john.

Given a Bayesian logic program $\mathcal{B}$ and a set of constants, the declarative semantics is formalized using a dependency graph $DB(\mathcal{B})$. $DB(\mathcal{B})$ is a directed network, where

1. Each node is a ground atom in the least Herbrand model $LH(\mathcal{B})$,
2. Each edge represents a direct influence relationship over the random variables in $LH(\mathcal{B})$. A direct influence relationship between random variables X and Y exists if and only if
   (a) $X, Y \in LH(\mathcal{B})$,
   (b) There is a Bayesian clause $A|A_1, \ldots, A_N$ in $\mathcal{B}$ and a substitution θ such that $X = A\theta$ and $Y = A_i\theta$, $i \in \{1, \ldots, N\}$.

For the example about children's height, assume a Herbrand universe with constants (persons) ann, jame, mary, bill and john. The ground atoms of the least Herbrand model include: father(jame, mary), mother(ann, mary), height(jame), height(ann), height(mary), father(bill, john), mother(mary, john), height(bill), height(john). Figure 2.7 shows its dependency graph, over which probabilistic queries can be answered.

In summary, Bayesian logic programming integrates Bayesian network with definite clause logic, and might provide more expressive power to model the complex relational domains, since the definite clause logic is able to represent more delicate relationships than the relational database logic.

2.4 SRL Tasks

In the context of statistical relational learning, the related information includes objects, attributes, relationships and graphs, thus various new tasks are brought. The major tasks in the context include: object identification, object ranking, object classification/clustering, relationship prediction, relationship classification and attribute prediction, as well as subgraph discovery, graph classification and so on (Getoor & Diehl, 2005; Han & Kamber, 2006). The implementation of these tasks invokes many challenges due to the heterogeneity, multi-relations and semi-structure of the data. In the section, we give brief introduction for some prime tasks.

2.4.1 Object Identification

Object identification is to find the different identifiers which map to the same real-world object. For example, in bibliography context, the references to the same paper may be described in different words, object identification is to find these references to build more reasonable citation network, over which accurate and compact inference can be performed. The problem of object identification first arises in database domain and is called entity resolution, which happens when a real-world object is distributed in multiple databases.

Traditionally, object identification is viewed as a pair-wise resolution problem, where each pair of references is independently resolved via comparing their attributes. In statistical relational learning, the performance of object identification is expected to be improved via integrating the information about relationships between objects, for example, the relationships of co-author in bibliographic data; the interaction between genes in information extraction of biology text. In statistical relational learning, what is the most interesting and promising might be the collective object identification strategy, which do not make match decision independently, in contrast, one match decision is made over others if the involved objects are related. These relationship-based object identification approaches can be widely applied in database domain for deduplication and integration, natural language extraction domain for co-reference resolution and object consolidation, social network domain for actor identification.

The existing researches include (Bhattacharya & Getoor, 2004; Bhattacharya & Getoor, 2005; Dong et al., 2005; Singla & Domingos, 2005b) and so on.

2.4.2 Object Ranking

Object ranking may be a primary focus in statistical relational learning due to the well known PageRank algorithm and its successful application in web information retrieval. The goal of object ranking is to order a set of objects based on their attributes, relationships and all other information.

PageRank algorithm is introduced by Page et al. (1998), which is a probabilistic model to estimate the likelihood that a user arrives a particular web page with a sequence of random clicks. In particular, PageRank models web surfing as a random walk where the user randomly selects the next page to browse given the structure of the network, i.e. the outgoing links from the current page. In terms of Markov theory, the likelihood of a web page can be computed as the steady-state probability of the random process. Then the web pages are ranked in the order of their likelihood. Another well-known approach for web information retrieval is HITS (Kleinberg, 1999), which divides the web pages into two categories: *hubs* and *authorities*, and then two independent random walks are performed in the two categories. For each web page, hub score and authority score are computed respectively as the steady-state probabilities of the two random processes. Finally, the web pages are ranked in the order of their values about hubs and authorities.

Another core application of object ranking is social network analysis. In the context, object ranking is used to order individuals in a given social network based on their importance/centrality. Note, that the social network can be static or dynamic. In a dynamic social network, additional information is available, e.g. the events between objects, including emails, telephone calls, messages and so on. In this situation, dynamic relational models are expected to capture the underly patterns in the complex network.

2.4.3 Object Classification/Clustering

The goal of object classification is to classify each object into a finite set of known groups. SRL approaches have an advantage over the traditional approaches since they collectively infer the category labels of all objects linked in a ground network. The enhanced hypertext classifier introduced by Chakrabarti et al. (1998) is among the first to notice the challenge, which explores the potential via using neighborhood class information to improve the classification accuracy for a hypertext document. Lafferty et al. (2001) introduced conditional random field to segment and label sequence data, which avoids the fundamental limitation of maximum entropy Markov models. However the model is restricted that data structure should be a chain. Taskar et al. (2002) extended the work of Lafferty et al. (2001) to data with arbitrary topology structures and applies the model to hypertext classification with promising results. Other related researches include (Lu & Getoor, 2003; Neville & Jensen, 2000) and so on.

Object clustering is also called group detection. The goal of the task is to cluster the objects into groups in terms of their attributes and relationships. A well-known application is identification of web communities, which cluster the web pages with the similar topics. The block modeling for social network is another well known application, which reduces a large, potentially incoherent social network to a small and comprehensible structure which can be interpreted more readily.

2.4.4 Relationship Prediction

Relationship prediction is to predict existence of relationships (links) based on the attributes and other relationships of the involved objects. Examples include whether there exists a hyperlink between two web pages, whether a user buys a book, and whether two persons make friends. The main challenge of the task is the sparsity and bias of the data. For example, the really existent hyperlinks in the internet is considerably sparse, relative to an extremely large number of potentially existent hyperlinks (Rattigan & Jensen, 2005). The possible solutions to improve the prediction performance are to make predictions collectively, or to incorporate discriminative modeling techniques into these prediction approaches. There are various compelling models developed for the task, such as the extension PRM model (Getoor et al., 2003), relational Markov network (Taskar et al., 2002), Markov logic network (Richardson & Domingos, 2006), nonparametric Bayesian models (Xu et al., 2005; Xu et al., 2006) and so on.

2.5 Summary

Statistical relational learning is a promising and booming research area. It explicitly exploits and models the data with objects, attributes and relationships by integrating various knowledge representations with probabilistic theories, such that the patterns in multi-relational, heterogeneous and semi-structured data can be discovered in an elegant and compact way. The related researches can be found in the following special conferences:

1. SIGKDD Workshop on Multi-Relational Data Mining (2002-2006);
2. Workshop on Mining and Learning with Graphs (ECML/PKDD 2003-2006);
3. Workshop on Learning Statistical Models from Relational Data (AAAI-2000, IJCAI 2003, and ICML 2004, ICML 2006);
4. Dagstuhl workshops on Probabilistic, Logical and Relational Learning (2005, 2007).

Part II

Relational Learning with Nonparametric Hierarchical Models

Chapter 3

Bayesian and Hierarchical Bayesian Models

3.1 Bayesian Models

3.1.1 Introduction

Bayesian data analysis is an important branch of statistical learning, which explicitly uses probability to quantify uncertainty in inferences. Let illustrate Bayesian analysis with a simple example. Assume that there is a set of N observations $D = \{y_1, y_2, ..., y_N\}$. The observation y_i can be discrete or continuous. In Bayesian framework, there is an underlying assumption that the N observations in the data set D are independently and identically distributed, which is denoted as $y_i \overset{\text{i.i.d.}}{\sim} P(\cdot|\theta)$, where θ represents the unknown parameters of the distribution. In the Bayesian framework, the unknown parameters themselves are random variables and are drawn from a distribution $P(\theta|\alpha)$ with *hyperparameters* α. Generally, the distribution $P(\theta|\alpha)$ is called *prior*, which represents our *uncertainty* about parameters θ before we see the data. Based on Bayes' rule, we obtain the *posterior* distribution of θ given data D and hyperparameters α:

$$P(\theta|D, \alpha) = \frac{P(D|\theta)P(\theta|\alpha)}{P(D|\alpha)}, \tag{3.1}$$

which reflects our uncertainty about parameters θ is updated after seeing the observations D. The factor $P(D|\theta)$ is referred to as *likelihood*, and represents the probability that the model generates the data D given parameters θ. Due to the underlying assumption of Bayesian analysis that each observation is i.i.d., the likelihood of the data D can be unfolded:

$$P(D|\theta) = \prod_{i=1}^{N} P(y_i|\theta). \tag{3.2}$$

Another factor $P(D|\alpha)$ in Equation 3.1 is called *marginal likelihood* or *evidence*

$$P(D|\alpha) = \int P(D|\theta)P(\theta|\alpha)d\theta, \tag{3.3}$$

which can be viewed as a normalization factor to ensure $\int P(\theta|D,\alpha)d\theta = 1$. Note, that with an increasing size of the data set, the posterior distribution of θ becomes increasingly localized and eventually converges to a point mass.

For a new observation y_{new}, the *predictive distribution* about its value is computed given data D and hyperparameters α:

$$\begin{aligned} P(y_{new}|D,\alpha) &= \int P(y_{new}|\theta)P(\theta|D,\alpha)d\theta \\ &\equiv \mathbb{E}_{P(\theta|D,\alpha)}\left[P(y_{new}|\theta)\right], \end{aligned} \tag{3.4}$$

where $\mathbb{E}_{P(\theta|D,\alpha)}[P(y_{new}|\theta)]$ denotes the expectation of $P(y_{new}|\theta)$ with respect to the posterior distribution $P(\theta|D,\alpha)$. The posterior distribution $P(\theta|D,\alpha)$ now plays a role of the *learned prior*, i.e., the available knowledge before the arrival of new data.

In summary, a Bayesian approach sets up a *full probability model* to learn the model parameters given data. First, prior distributions of model parameters are assumed to represent our initial uncertainty (knowledge) before the arrival of data, which might be obtained from expert experience. And then the posterior distributions are computed in terms of Bayes' rule, which reflect that our uncertainty (knowledge) is updated after seeing the data. With respect to the learned posteriors, we predict the variables of interest via averaging over all possible values of model parameters.

3.1.2 Exchangeability

A tacit assumption in statistic learning is that the N observations $D = \{y_1, y_2, ..., y_N\}$ are *exchangeable*, i.e. the joint distribution $P(y_1, \ldots, y_N)$ of the data is invariant if the indices of the variables are permuted. Let $\nu = \{\nu(1), \nu(2), \ldots, \nu(N)\}$ denote a permutation of the indies from 1 to N, the exchangeability assumption yields:

$$P(y_1, y_2, \ldots, y_N) = P(y_{\nu(1)}, y_{\nu(2)}, \ldots, y_{\nu(N)}). \tag{3.5}$$

Furthermore, when the number of the variables is infinite, i.e. $N \rightarrow \infty$, the variables are *infinite exchangeable*, if any finite subset of variables are exchangeable. Based on the exchangeability assumption, it is natural to model the data as independently and identically distributed given model parameters θ,

$$P(y_1, y_2, \ldots, y_N|\theta) = \prod_{i=1}^{N} P(y_i|\theta). \tag{3.6}$$

The exchangeability relations in a model can be illustrated in a graphical representation, referred to as *plate*, which is a template that allows the subgraphs can be replicated. Figure 3.1(a) shows the model discussed in Section 3.1.1. Figure 3.1(b) shows the equal model in a plate. In the plate language, variables (not random) are represented directly by their names, e.g. the hyperparameters α. Random variables, e.g. θ, are represented as circles with their names. The N exchangeable variables $\{y_1, \ldots, y_N\}$ are represented as a single variable y_i in a rectangle. The number N at the corner specifies the number of

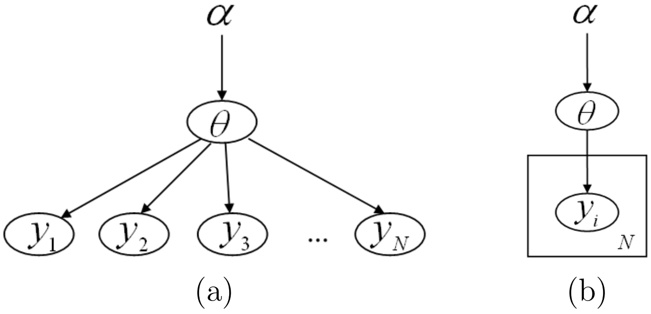


Figure 3.1: (a) A simple Bayesian model. (b) An equal model with a plate representation.

the variables. An arrow, e.g. from α to θ denotes that the probability distribution of θ is conditioned on α. Note, that the arrow from θ to y_i specifies each of the N variables y_i depends on θ. The plate representation is often used to illustrate probability models. It clarifies the exchangeability relations in a compact and elegant way.

3.1.3 Inference and Parameter Learning

Probabilistic inference means the computation of the probability of a quantity given a model, potentially under some observations. We distinguish between two kinds of inference. First, the computation of the probabilities of potentially observable quantities, such as a missing observation y', $P(y'|D,\alpha)$. Second, the computation of the probabilities of quantities that are not observable, e.g. the model parameters θ, $P(\theta|D,\alpha)$, which is sometimes called *parameter learning.* Note, that in Bayesian framework what we learn is the distributions of the unknown quantities, not the quantities themselves, since unknown variables are random in Bayesian modeling. An exception is empirical Bayesian modeling we will introduce in the next section.

Especially, the inference about a new observable is called *predictive inference.* We consider two situations: one is to predict without any data, the other is to predict with some known data. The former is denoted as *prior predictive inference*:

$$P(y_{new}|\alpha) = \int P(y_{new}|\theta)P(\theta|\alpha)d\theta, \tag{3.7}$$

which is conducted before the arrival of the data, thus the uncertainty about the model parameters is represented by the prior distributions. In the latter situation when the data D is given, we predict a new observation as:

$$P(y_{new}|D,\alpha) = \int P(y_{new}|\theta)P(\theta|D,\alpha)d\theta. \tag{3.8}$$

It is referred to as *posterior predictive inference.* The prediction is performed after the arrival of the data, thus the uncertainty about the model parameters is represented as posterior distributions.

3.1.4 Exponential Family and Conjugate Prior

Regardless of parameter learning or predictive inference, the marginal distribution $P(\theta|D, \alpha)$ or $P(y_{new}|D, \alpha)$ is computationally expensive in more cases than not. To solve the limitation, a class of distributions known as *exponential family* are introduced so that the computation can be efficient and in closed form. Members of this family include discrete and continuous distributions, such as Bernoulli, binomial, multinomial, Poisson distributions; and Gaussian, Gamma, Beta, Dirichlet distributions.

The distributions in exponential family take a certain form:

$$P(\mathbf{y}|\theta) = \mathrm{H}(\mathbf{y}) \exp\left[\theta^T \mathrm{T}(\mathbf{y}) - \mathrm{A}(\theta)\right]. \tag{3.9}$$

Where θ denotes the parameters of the distribution and is called *natural parameters.* $\mathrm{H}(\mathbf{y})$, $\mathrm{T}(\mathbf{y})$ and $\mathrm{A}(\theta)$ are different functions. $\mathrm{H}(\mathbf{y})$ is the underlying measure with respect to which $P(\mathbf{y}|\theta)$ is a density function. $\mathrm{T}(\mathbf{y})$ is a *sufficient statistic* of the distribution. Generally, a sufficient statistic is a function of the samples that contains all information to estimate the natural parameters θ, e.g., for a Gaussian distribution, the mean and covariance of the samples are the sufficient statistics to estimate the true mean and covariance of the distribution. The function $\mathrm{A}(\theta)$ is defined in terms of the other two functions:

$$\mathrm{A}(\theta) = \log \int \mathrm{H}(\mathbf{y}) \exp\left[\theta^T \mathrm{T}(\mathbf{y})\right] d\mathbf{y}, \tag{3.10}$$

which can be viewed as the logarithm of a normalization factor. $\mathrm{A}(\theta)$ is used to ensure $\int P(\mathbf{y}|\theta) d\mathbf{y} = 1$.

Each member of exponential family has a simple *conjugate prior*, which is an important property for Bayesian analysis. In Bayesian probability theory, a conjugate prior is a prior distribution which posterior distribution also takes the same mathematic form. For example, if the data are i.i.d. drawn from a multinomial distribution with unknown parameters θ and a conjugate prior (i.e. Dirichlet distribution) is assumed, then the posterior distribution of parameters θ is still Dirichlet. If the likelihood distribution $P(D|\theta)$ of data D belongs to the exponential family, then there exists a conjugate prior, which is also in the exponential family. Not all likelihood distributions are associated with conjugate priors. In general, an arbitrary likelihood distribution, not being the exponential family, has no conjugate prior. In the case, the computation about the posterior distribution might be expensive and has to be approximated via numerical methods. Therefore, for computational convenience, it is common to assume a conjugate prior for model parameters, since the assumption reduces the computation from the function approximation to the parameter approximation. Table 3.1 lists the commonly-used exponential family distributions and the corresponding conjugate priors.

3.1.5 Differences from Classical Statistical Approaches

Although Bayesian and classical statistical approaches obtain nearly identical results in many applications, the underlying mechanisms are completely different. The major difference is that: the unknown parameters θ are viewed to be random in Bayesian approaches,

Table 3.1: Some exponential family distributions $P(y|\theta)$ and their conjugate priors

Name	Distribution Function	Parameters	Conjugate Prior
Bernoulli	$\theta^y(1-\theta)^{1-y}, \quad y = 0, 1$	$0 < \theta < 1$	Beta
Binomial	$\frac{m!}{y!(m-y)!}\theta^y(1-\theta)^{m-y},$ $y = 0, 1, \ldots, m$	$0 < \theta < 1$	Beta
Multinomial	$\frac{m!}{\prod_s y_s!}\prod_s \theta_s^{y_s},$ $y_s = 0, 1, \ldots, m;$ $\sum_s y_s = m$	$0 < \theta_s < 1$ $\sum_s \theta_s = 1$	Dirichlet
Poisson	$\frac{\theta^y}{y!}e^{-\theta}, \quad y = 0, 1, \ldots$	$\theta > 0$	Gamma
Beta	$\begin{cases} \frac{\Gamma(\theta_1+\theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)}y^{\theta_1-1}(1-y)^{\theta_2-1} \\ \qquad\qquad \text{if } 0 \leq y \leq 1 \\ 0 \qquad \text{Otherwise.} \end{cases}$	$\theta_1 > 0$ $\theta_2 > 0$	—
Dirichlet	$\begin{cases} \frac{\Gamma(\sum_s \theta_s)}{\prod_s \Gamma(\theta_s)}\prod_s y_s^{\theta_s-1} \\ \text{if } 0 \leq y_s \leq 1, \sum_s y_s = 1 \\ 0 \qquad \text{Otherwise.} \end{cases}$	$\theta_s > 0$	—
Gamma	$\begin{cases} \frac{\theta_2^{\theta_1}}{\Gamma(\theta_1)}y^{\theta_1-1}e^{-\theta_2 y} & y \geq 0 \\ 0 & y < 0 \end{cases}$	$\theta_1 > 0$ $\theta_2 > 0$	—
Exponential	$\begin{cases} \theta e^{-\theta y} & y \geq 0 \\ 0 & y < 0 \end{cases}$	$\theta > 0$	Gamma
Gaussian	$\frac{1}{\sqrt{2\pi\theta_2}}\exp(-\frac{(y-\theta_1)^2}{2\theta_2})$ $-\infty < y < +\infty$	$\theta_2 > 0$	θ_1: Gaussian; θ_2: scaled inverse chi square.

but to be fixed in classical statistical approaches. In Bayesian modeling, the parameters θ have a prior distribution, which expresses our uncertainty about the values of the parameters before the arrival of the data. Given the data, the uncertainty is still remained but updated and represented as the posterior.

In Bayesian modeling, the predictive inference computes the following equation:

$$P(y_{new}|D,\alpha) = \int P(y_{new}|\theta)P(\theta|D,\alpha)d\theta, \tag{3.11}$$

which is an average over the posterior distribution of θ. In contrast, the classical statistical approaches yield the following prediction process. First the parameters are estimated in terms of some criteria, e.g. maximum-likelihood estimation:

$$\theta^{ML} = \arg\max_{\theta} P(D|\theta). \tag{3.12}$$

Then the learned parameters are viewed as the real ones, the prediction is directly performed as $P(y_{new}|\theta^{ML})$. The classical statistical approaches do not consider the uncertainty in estimating θ given data D.

3.1.6 Example

In this section, we discuss the Bayesian inference in a particular example. Assume that there is a data set $D = \{y_1, \ldots, y_N\}$. y_i is a discrete variable with S possible states. $y_i = s$ if the s'th state is taken. Let y_i be exchangeable and follow multinomial distribution with parameters $\theta = (\theta_1, \theta_2, \ldots, \theta_S)$, $0 < \theta_s < 1$ and $\sum_s \theta_s = 1$. We have

$$P(y_i = s|\theta) = \theta_s. \tag{3.13}$$

In terms of the exchangeability assumption, the data can be summarized by the number of observations at each state. Let N_s denote the number of observations with state s. We have $\sum_s N_s = N$. The likelihood of the data D can be written as:

$$P(D|\theta) = \prod_{i=1}^{N} P(y_i|\theta) = \prod_{s=1}^{S} \theta_s^{N_s}. \tag{3.14}$$

To carry out the Bayesian inference, a prior need to be specified to the unknown multinomial parameters θ. As discussed in Section 3.1.4, a conjugate prior is assumed, i.e., the parameters follow a Dirichlet distribution with hyperparameters α^*, denoted as $\theta \sim \mathrm{Dir}(\cdot|\alpha^*)$ with a density:

$$P(\theta|\alpha^*) = \frac{\Gamma(\alpha_0)}{\prod_{s=1}^{S}\Gamma(\alpha_s^*)} \prod_{s=1}^{S} \theta_s^{\alpha_s^*-1}, \tag{3.15}$$

where α_s^* is a positive real number and $\alpha_0 = \sum_{s=1}^{S} \alpha_s^*$, $\frac{\Gamma(\alpha_0)}{\prod_{s=1}^{S}\Gamma(\alpha_s^*)}$ is a normalization factor. It is common to re-parameterize

$$\alpha_s = \frac{\alpha_s^*}{\alpha_0}, \ s = 1, \ldots, S, \tag{3.16}$$

and the hyperparameters become $\alpha = \alpha_0(\alpha_1, \ldots, \alpha_S)$, where α_s represents our prior belief about the probability of the state $P(y = s|\alpha) = \alpha_s$. α_0 is a scale indicating how strongly we believe that the prior distribution is true. The larger the value is, the more confidently we can make claims for the prior distribution.

Based on the Bayes' rule, the posterior distribution is computed as:

$$\begin{aligned}
P(\theta|D, \alpha) &= \frac{P(D|\theta)P(\theta|\alpha)}{\int P(D|\theta)P(\theta|\alpha)d\theta} \\
&= \frac{\prod_{s=1}^{S} \theta_s^{\alpha_0\alpha_s + N_s - 1}}{\int \prod_{s=1}^{S} \theta_s^{\alpha_0\alpha_s + N_s - 1} d\theta_1 \cdots d\theta_S} \\
&= \frac{\Gamma(\alpha_0 + N)}{\prod_{s=1}^{S} \Gamma(\alpha_0\alpha_s + N_s)} \prod_{s=1}^{S} (\theta_s)^{\alpha_0\alpha_s + N_s - 1}. \qquad (3.17)
\end{aligned}$$

It is clear that the posterior distribution $P(\theta|D, \alpha)$ is also Dirichlet distribution with new parameters

$$\begin{aligned}
\alpha_{post} &= (\alpha_0\alpha_1 + N_1, \ldots, \alpha_0\alpha_S + N_S) \\
&= (\alpha_0 + N)(\frac{\alpha_0\alpha_1 + N_1}{\alpha_0 + N}, \ldots, \frac{\alpha_0\alpha_S + N_S}{\alpha_0 + N}). \qquad (3.18)
\end{aligned}$$

The computation of posterior distribution is based on two components: the prior belief represented as α and the known data represented as N_s. With an increasing size of the data set, the prior plays a smaller and smaller role, the posterior distribution $P(\theta|D, \alpha)$ comes to be dominated by information from the data and converges to a point distribution, when $N \to \infty$, it becomes eventually $P(\theta|D, \alpha) \approx \delta_{\theta^*}(\theta)$, where $\delta_{\theta^*}(\theta)$ is a distribution with a point mass on $\theta^* = (\frac{N_1}{N}, \ldots, \frac{N_S}{N})$. The Figure 3.2 shows the posterior distributions for three data sets with different size but identical proportion $(0.3, 0.5, 0.2)$ at each state. The prior is a Dirichlet distribution with hyperparameters $\alpha = 3(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, the prior variances about model parameters θ are $(0.0556, 0.0556, 0.0556)$. The size of the first data set is 10, the number of samples at each state is $(3, 5, 2)$. The parameters of posterior distribution are $\alpha_{post}^1 = 13(0.3077, 0.4615, 0.2308)$. The posterior variances about θ become $(0.0152, 0.0178, 0.0127)$. The second data set has more samples, $N = 50$. The number of samples at each state is $(15, 25, 10)$. The parameters of posterior distribution are $\alpha_{post}^2 = 53(0.3019, 0.4906, 0.2075)$. The posterior variances are $(0.0039, 0.0046, 0.0030)$. The third data set has the most samples, $N = 100$, the number of samples at each state is $(30, 50, 20)$. The parameters of posterior distribution are $\alpha_{post}^3 = 103(0.3010, 0.4951, 0.2039)$. The posterior variances are the smallest, $(0.0020, 0.0024, 0.0018)$. From the example, we first find that with the increasing number of samples, the posterior distribution really approximates the point mass $(0.3, 0.5, 0.2)$, which is exactly the proportion of samples at each state, although we assume an unbiased prior, i.e. the prior probability of taking each state s is equal. In addition, we also find that the variances of the parameters decrease with the increasing data. It claims the expected result that posterior distribution becomes less variable as additional information is available. There is a statement in Bayesian analysis that the posterior variances are on

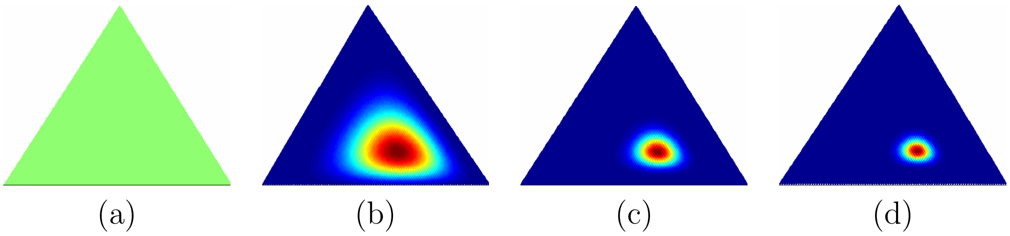


Figure 3.2: The posterior distributions for three data sets with different size but identical proportion at each state. (a) Dirichlet prior with hyperparameters $\alpha = (1, 1, 1)$. (b) posterior distribution given a small data set, the number of samples at each state is $(3, 5, 2)$. (c) posterior distribution given a middle size of data set with 50 samples. (d) posterior distribution given a data set with 100 samples. It is clear that the posterior distribution concentrates on a point mass with increasing data. The phenomenon claims the expected result that posterior distribution becomes less variable as additional information is available.

average smaller than the prior variances. Of course, the statement is made in terms of the expectations, and in particular situations the posterior variances might be similar or even larger than the prior variances.

For a new observation y_{new}, the posterior predictive inference is performed

$$\begin{aligned} P(y_{new} = s|D, \alpha) &= \int P(y_{new} = s|\theta)P(\theta|D, \alpha)d\theta \\ &= \int \theta_s P(\theta|D, \alpha)d\theta \\ &\equiv \mathbb{E}_{P(\theta|D,\alpha)}(\theta_s) = \frac{\alpha_0\alpha_s + N_s}{\alpha_0 + N}. \end{aligned} \tag{3.19}$$

The prediction combines together the prior information represented as hyperparameters α, and the data information represented as the sufficient statistics N_s. With an increasing size of the data, the prediction is dominated by the information from data. When $N \to \infty$, the predictive inference converges to $P(y_{new} = s|D, \alpha) = N_s/N$.

3.2 Hierarchical Bayesian Models

3.2.1 Introduction

In Section 3.1, we introduce Bayesian modeling in which the hyperparameters, i.e. the parameters of prior distribution, are known. However in many real-world applications, the information is not available. Thus *hierarchical Bayesian model* (HB) is introduced to solve the problem. From a broadest point of view, hierarchical model means a model with many levels and structured in terms of a sequence of conditional distributions. Figure 3.3 shows a typical hierarchical Bayesian model. The observations y_i for $i = \{1, \ldots, N\}$ are i.i.d. drawn from a distribution with unknown parameters θ. The unknown parameters are drawn from a prior distribution with unknown hyperparameters α, which themselves are random variables and are drawn from a distribution with parameters η. The observations are modeled in a hierarchical structure, which has one more level than the Bayesian model in Figure 3.1. The additional level represents the uncertainty about the unknown hyperparameters in the form $\alpha \sim P(\cdot|\eta)$, which is sometimes referred to as *hyperprior.* Given the data $D = \{y_1, \dots, y_N\}$, the posterior distribution of the unknown hyperparameters α can be written as

$$\begin{aligned} P(\alpha|D,\eta) &\propto P(\alpha|\eta)P(D|\alpha) \\ &= P(\alpha|\eta)\int P(\theta|\alpha)P(D|\theta)d\theta. \end{aligned} \tag{3.20}$$

Equation 3.20 might be analytically intractable, but can be approximated via some advanced computational methods. Hierarchical model supplies a mechanism to estimate the hyperparameters from the data. Many applied analysis approaches are developed under hierarchical Bayesian framework. In this thesis, we focus on an extended *object-oriented hierarchical Bayesian modeling* (OOHB), which is designed to analyze the parallel data sources.

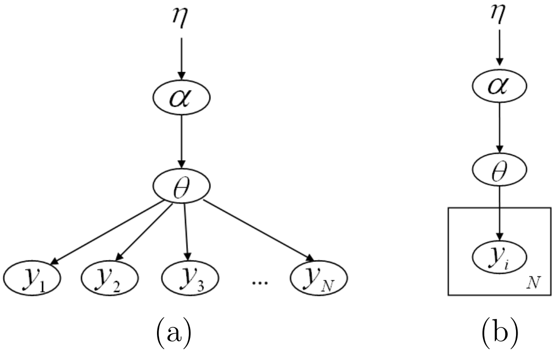


Figure 3.3: (a) A typical hierarchical Bayesian model. (b) An equal model with a plate representation.

Object-oriented hierarchical Bayesian modeling is specific for the situation where the known multiple data sets are generated from *different but related settings*, therefore the model parameters for each data set are also different but closely connected. The relation is modeled in a natural way that the parameters are independently sampled from a common but unknown prior. There are two strategies to encode the unknown prior. The first strategy assumes that the parameters of the prior (hyperparameters) are *unknown, but not random*, which is sometimes referred to as *empirical Bayesian* method. It is however *not full Bayesian* treatment. In the second strategy, we assume that the unknown hyperparameters are random variables, which uncertainty is represented as $\alpha \sim P(\alpha|\eta)$.

Let illustrate the object-oriented hierarchical Bayesian modeling with a movie example. In a survey about the popularity of a movie, user ratings ranged from 1 to 5 are investigated in different cities. Assume that we collect data sets $D = \{D_1, \dots, D_M\}$ from M cities. It is obvious that the data sets come from related but not identical scenarios. A reasonable assumption is that the observations in data set D_j are drawn from a distribution with parameters θ_j. The parameters θ_j for the setting j are distinct, but related to each other. Thus it is natural to assume that θ_j's are generated from a common prior with hyperparameters α, and the hyperparameters can be learned from the M data sets. In empirical Bayesian strategy, we perform point estimation to approximate hyperparameters α. In full Bayesian framework, the posterior distribution of α is computed. No matter which strategy we employ, for a new data set D_{M+1} collected in a new city, the posterior distribution of parameters θ_{M+1} will be estimated not only based on the current data set D_{M+1}, but also based on the historical data sets from other cities. The historical information is transferred via the common prior.

The object-oriented hierarchical Bayesian modeling is specially appropriate for the data with hierarchical structure. It needs fewer parameters to model the common and different properties in the parallel data sources. Each data source is distinguished via personalized parameters, but is closely connected with the common prior. A non-hierarchical model may achieve the similar performance with more parameters, but may tend to cause the problem of overfitting, i.e. the current data sets are modeled well but the prediction to new data is inferior.

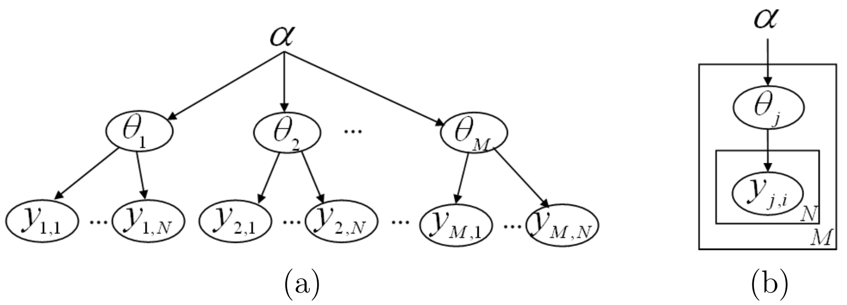
3.2.2 Exchangeability

In the object-oriented hierarchical models, the exchangeability is defined at each level of elements. In the first level, each data point $y_{j,i}$ in a data set D_j is exchangeable. In the second level, each data set D_j is exchangeable. Let $\nu = \{\nu(1), \nu(2), \ldots, \nu(M)\}$ denote a permutation of the indies of data sets from 1 to M, and $\varsigma = \{\varsigma(1), \varsigma(2), \ldots, \varsigma(N_j)\}$ denote a permutation of the indies of samples from 1 to N_j in data set D_j. The exchangeability assumption yields:

$$P(Y_{j,1}, Y_{j,2}, \ldots, Y_{j,N_j}) = P(Y_{j,\varsigma(1)}, Y_{j,\varsigma(2)}, \ldots, Y_{j,\varsigma(N_j)}) \tag{3.21a}$$

$$P(D_1, D_2, \ldots, D_M) = P(D_{\nu(1)}, D_{\nu(2)}, \ldots, D_{\nu(M)}). \tag{3.21b}$$

3.2.3 Empirical Bayesian Models



(a) (b)

Figure 3.4: (a) An empirical object-oriented hierarchical model. (b) An equal model with a plate representation.

Figure 3.4 shows an example about empirical Bayesian framework for the object-oriented hierarchical modeling. The principle idea is that the hyperparameters α are *unknown, but not random.* Although the empirical solution is not full Bayesian, it is mathematically easier and includes major properties of object-oriented hierarchical modeling. Assume that there are M parallel data sources $D = \{D_1, \ldots, D_M\}$, and in data set D_j, there are N_j observations $D_j = \{y_{j,1}, \ldots, y_{j,N_j}\}$. In the empirical Bayesian framework, the unknown hyperparameters α are approximated via point estimation methods, say *maximum likelihood* estimation:

$$\begin{aligned}\alpha^{ML} &= \arg\max_{\alpha} P(D|\alpha) \\ &= \arg\max_{\alpha} \prod_{j=1}^{M} \int P(\theta_j|\alpha) \prod_{i=1}^{N_j} P(y_{j,i}|\theta_j) d\theta_j. \end{aligned} \tag{3.22}$$

This is an optimization problem, which can be solved via, e.g. gradient descent method or Newton's method. For more details about the related algorithms, please refer to (Papalambros & Wilde, 2000). The computation of the marginal distribution $P(D|\alpha)$ is straightforward if we assume the prior and likelihood distributions are of manageable form. For example, when $P(\theta_j|\alpha)$ is assumed a conjugate prior, then the integration is analytically computed. If we have to calculate the integration numerically, the efficiency depends on the dimension of θ_j, since the integration is over individual θ_j not all of θ_j's. Therefore, the computation is typically a low dimensional integral, we can consider, say Gaussian quadrature method.

After getting the estimation α^{ML} of hyperparameters, the learned prior is viewed as true prior, by which the information in the M historical data sets is propagated to a new data set D_{M+1}, which is generated from a setting $M+1$. The property is clarified in computation of posterior distribution of new parameters θ_{M+1}:

$$P(\theta_{M+1}|D_{M+1}, \alpha^{ML}) \propto P(D_{M+1}|\theta_{M+1}) P(\theta_{M+1}|\alpha^{ML}), \tag{3.23}$$

which is proportional to the product of the likelihood $P(D_{M+1}|\theta_{M+1})$ and the prior $P(\theta_{M+1}|\alpha^{ML})$. It is clear that the new model not only explains the current data set D_{M+1}, but also implicitly reflects the previous data sets. There is a practical problem in the computation. If we want to estimate the posterior distribution of θ_j, $j \leq M$, then the data set D_j will be used twice. First, it is used with other historical data sets to estimate the hyperparameters α^{ML}. Second, it is used to estimate the distribution $P(\theta_j|D_j, \alpha^{ML})$ for the parameters of interest. It might cause overestimation. Despite the problem, it clearly makes more sense to first estimate the hyperparameters from all data sources and then estimate θ_j, than to estimate θ_j separately.

For a new observation $y_{j,new}$ in the jth scenario, $j = \{1, \ldots, M+1\}$, the predictive distribution is computed as:

$$P(y_{j,new}|D_j, \alpha^{ML}) = \int P(y_{j,new}|\theta_j) P(\theta_j|D_j, \alpha^{ML}) d\theta_j. \tag{3.24}$$

3.2.4 Example

Now we discuss the computational details in empirical HB model with a particular example. Assume that $y_{j,i}$ for $j = \{1, \ldots, M\}$, $i = \{1, \ldots, N_j\}$, is a discrete variable with S

possible states, and follows multinomial distribution with parameters θ_j, we have:

$$P(y_{j,i} = s|\theta_j) = \theta_{j,s}. \tag{3.25}$$

Where the multinomial parameters θ_j for each scenario are i.i.d. drawn from Dirichlet distribution with parameters $\alpha^* = (\alpha_1^*, \ldots, \alpha_S^*)$, $\alpha_s^* > 0$ and $\alpha_0 = \sum_{s=1}^S \alpha_s^*$. In data sets D_j, there are $N_{j,s}$ samples with state s and $\sum_s N_{j,s} = N_j$. We compute the hyperparameters α^* with maximum log-likelihood estimation method:

$$\begin{aligned}
\alpha^{ML} &= \arg\max_{\alpha^*} \log\left[\prod_{j=1}^{M} \int P(\theta_j|\alpha^*) \prod_{i=1}^{N_j} P(y_{j,i}|\theta_j) d\theta_j\right] \\
&= \arg\max_{\alpha^*} \log\left[\prod_{j=1}^{M} \int \frac{\Gamma(\alpha_0)}{\prod_{s=1}^S \Gamma(\alpha_s^*)} \prod_{s=1}^{S} \theta_{j,s}^{\alpha_s^*-1} \prod_{s=1}^{S} \theta_{j,s}^{N_{j,s}} d\theta_j\right] \\
&= \arg\max_{\alpha^*} \log\left[\prod_{j=1}^{M} \frac{\Gamma(\alpha_0)}{\prod_{s=1}^S \Gamma(\alpha_s^*)} \int \prod_{s=1}^{S} \theta_{j,s}^{\alpha_s^*+N_{j,s}-1} d\theta_j\right] \\
&= \arg\max_{\alpha^*} \Bigg[M \log\Gamma(\alpha_0) - \sum_{s=1}^{S} M \log\Gamma(\alpha_s^*) + \sum_{s=1}^{S}\sum_{j=1}^{M} \log\Gamma(\alpha_s^* + N_{j,s}) \\
&\qquad - \sum_{j=1}^{M} \log\Gamma(\alpha_0 + N_j)\Bigg].
\end{aligned} \tag{3.26}$$

It is not easy to get analytical solution to the optimization problem, but some numerical methods can be considered, e.g. coordinate ascent algorithm. The optimization approach was developed by D'Esopo (1959), which maximizes the target function by iteratively optimizing in each of the coordinate directions. In particular, coordinate ascent algorithm optimizes each α_s^* given all the others at one iteration, i.e.

$$0 = \frac{\partial}{\partial \alpha_s^*}\left[M \log\Gamma(\alpha_0) - \sum_{s=1}^{S} M \log\Gamma(\alpha_s^*) + \sum_{s=1}^{S}\sum_{j=1}^{M} \log\Gamma(\alpha_s^* + N_{j,s}) - \sum_{j=1}^{M} \log\Gamma(\alpha_0 + N_j)\right]. \tag{3.27}$$

It yields:

$$0 = M\Big[\Psi(\alpha_0) - \Psi(\alpha_s^*)\Big] + \sum_{j=1}^{M}\Big[\Psi(\alpha_s^* + N_{j,s}) - \Psi(\alpha_0 + N_j)\Big]. \tag{3.28}$$

Where $\Psi(\cdot)$ is digamma function, and comes from the first derivative of the logarithm of the gamma function $\Gamma(\cdot)$. It is clear that the equation can not be solved analytically. We consider numerical methods, say the Newton's method, which is widely used to approximate the roots of a function. The Equation 3.28 can be solved efficiently via Newton's method, since there is only a single variable α_s^*, i.e. the root-finding problem is one-dimensional. The Newton's method yields the following equations to update α_s^* at each iteration:

$$\alpha_s^{*(t+1)} = \alpha_s^{*(t)} - \frac{f(\alpha_s^{*(t)})}{f'(\alpha_s^{*(t)})}, \tag{3.29}$$

where

$$f(\alpha_s^{*(t)}) = M\left[\Psi(\alpha_0^{(t)}) - \Psi(\alpha_s^{*(t)})\right] + \sum_{j=1}^{M}\left[\Psi(\alpha_s^{*(t)} + N_{j,s}) - \Psi(\alpha_0^{(t)} + N_j)\right] \tag{3.30a}$$

$$f'(\alpha_s^{*(t)}) = M\left[\Psi'(\alpha_0^{(t)}) - \Psi'(\alpha_s^{*(t)})\right] + \sum_{j=1}^{M}\left[\Psi'(\alpha_s^{*(t)} + N_{j,s}) - \Psi'(\alpha_0^{(t)} + N_j)\right]. \tag{3.30b}$$

In summary, the unknown hyperparameters α^{ML} are optimized in the following steps:

1. Randomly initialize $(\alpha_1^{*(n)}, \ldots, \alpha_S^{*(n)})$, $n = 0$.
2. Iterate the following steps for $n = 1, 2, \ldots$
   - Update $\alpha_s^{*(n)}$ given $(\alpha_1^{*(n)}, \ldots, \alpha_{s-1}^{*(n)}, \alpha_s^{*(n-1)}, \ldots, \alpha_S^{*(n-1)})$.
   - Let $\alpha_s^{*(t)} = \alpha_s^{*(n-1)}$, $t = 0$
   - Iteratively compute Equation 3.29 for $t = 1, 2, \ldots$, where

     $$\alpha_0^{(t)} = \alpha_1^{*(n)} + \cdots + \alpha_{s-1}^{*(n)} + \alpha_s^{*(t)} + \alpha_{s+1}^{*(n-1)} + \cdots + \alpha_S^{*(n-1)} \tag{3.31}$$

   - Stop until $\alpha_s^{*(t)}$ reaches a stationary point, and let

     $$\alpha_s^{*(n)} = \alpha_s^{*(t)} \tag{3.32}$$

   - $s \leftarrow s + 1$, go to update the next $\alpha_s^{*(n)}$.
3. Stop until the convergence achieves.

Note that the optimized hyperparameters α^{ML} are not Bayesian computation, since the empirical solution is not a full probability model, the uncertainty in estimating α^* is not considered. Despite the limitation, the empirical Bayesian estimation is a good starting point from which a full Bayesian solution can be explored.

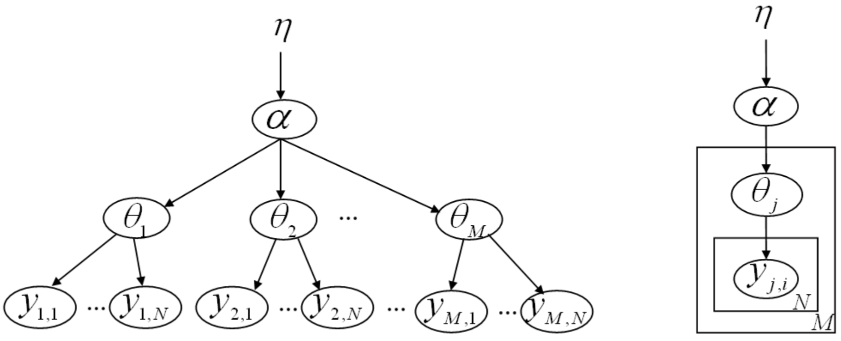
After obtaining the maximum log-likelihood estimation α^{ML}, it is straightforward to compute the posterior distribution and the predictive distribution. For $j = \{1, \ldots, M+1\}$, the posterior distribution is derived as:

$$\begin{aligned} P(\theta_j|D_j, \alpha^{ML}) &= \mathrm{Dir}(\cdot|\alpha^{post}) \\ \alpha^{post} &= (\alpha_1^{ML} + N_{j,1}, \ldots, \alpha_S^{ML} + N_{j,S}), \end{aligned} \tag{3.33}$$

the predictive distribution $P(y_{j,new} = s|D_j, \alpha^{ML})$ for a new observation is:

$$\begin{aligned} & \int P(y_{j,new} = s|\theta_j)P(\theta_j|D_j, \alpha^{ML})d\theta_j \\ = & \int \theta_{j,s}P(\theta_j|D_j, \alpha^{ML})d\theta_j \\ \equiv & \ \mathbb{E}_{P(\theta_j|D_j,\alpha^{ML})}\theta_{j,s} = \frac{\alpha_s^{ML} + N_{j,s}}{\sum_{s=1}^{S} \alpha_s^{ML} + N_{j,s}}. \end{aligned} \tag{3.34}$$

3.2.5 Hierarchical Models in Full Bayesian Framework



(a) (b)

Figure 3.5: (a) A hierarchical model in full Bayesian framework. (b) An equal model with a plate representation.

The full Bayesian hierarchical model is shown as Figure 3.5. In contrate with the empirical Bayesian hierarchical model, the unknown hyperparameters α are random variables, which uncertainty is represented as $\alpha \sim P(\cdot|\eta)$. The distribution is referred to as *hyperprior*. In the full Bayesian framework, the samples are generated from the following procedure:

$$\begin{aligned} \alpha|\eta &\sim P(\alpha|\eta). \\ \theta_j|\alpha &\sim P(\theta_j|\alpha) \text{ for } j = \{1, \ldots, M\}. \\ y_{j,i}|\theta_j &\sim P(y_{j,i}|\theta_j) \text{ for } i = \{1, \ldots, N_j\}. \end{aligned}$$

The joint probability is defined as:

$$P(\alpha, \{\theta_j\}_{j=1}^M, D|\eta) = P(\alpha|\eta) \prod_{j=1}^{M} P(\theta_j|\alpha) \prod_{i=1}^{N_j} P(y_{j,i}|\theta_j). \tag{3.36}$$

In the full Bayesian model, the unknown parameters include α and $\theta_1, \ldots, \theta_M$. Based on the Bayes' rule, the joint posterior distribution is computed as:

$$P(\alpha, \{\theta_j\}_{j=1}^M|D, \eta) \propto P(\alpha|\eta) \prod_{j=1}^{M} P(\theta_j|\alpha) \prod_{i=1}^{N_j} P(y_{j,i}|\theta_j). \tag{3.37}$$

The unnormalized distribution is a product of the hyperprior $P(\alpha|\eta)$, the prior $P(\theta_j|\alpha)$ and the likelihood $P(y_{j,i}|\theta_j)$. In different situations, we may be interested in a specific marginal posterior distribution, say $P(\alpha|D,\eta)$ or $P(\theta_j|D,\eta)$ for the parameters α or θ_j. The marginal distribution can be obtained by integrating the joint posterior distribution over α and θ_j's:

$$P(\alpha|D,\eta) = \int P(\alpha, \{\theta_j\}_{j=1}^M|D,\eta)d\theta_1,\ldots,d\theta_M. \tag{3.38a}$$

$$P(\theta_j|D,\eta) = \int P(\alpha, \{\theta_j\}_{j=1}^M|D,\eta)d\alpha d\theta_1,\ldots,d\theta_{j-1},d\theta_{j+1},\ldots,d\theta_M. \tag{3.38b}$$

For a new scenario, the computation about the distribution of parameters θ_{M+1} can be performed in two different situations: first, no observations in the scenario are available; second, some observations D_{M+1} are available. In the first situation, the distribution is estimated in terms of prior knowledge and the data from other scenarios:

$$P(\theta_{M+1}|D,\eta) \propto \int P(\theta_{M+1}|\alpha)P(\alpha|D,\eta)d\alpha, \tag{3.39}$$

which can be viewed as the *prior* of the new parameters θ_{M+1}. In the second situation, i.e., the observations D_{M+1} are available, the distribution of the parameters is computed as:

$$P(\theta_{M+1}|D_{M+1},D,\eta) \propto P(D_{M+1}|\theta_{M+1})\int P(\theta_{M+1}|\alpha)P(\alpha|D,\eta)d\alpha, \tag{3.40}$$

which exploits not only the current data set D_{M+1}, but also the historical data sets $D = \{D_1,\ldots,D_M\}$. In empirical Bayesian framework, the historical information is propagated via the learned hyperparameters α^{ML}, in the full Bayesian framework, the propagation is implemented via the marginal posterior distribution $P(\alpha|D,\eta)$. The full Bayesian framework has an advantage over the empirical Bayesian framework since it considers the uncertainty in estimating α. For a new observation $y_{M+1,new}$ in the scenario $M+1$, the predictive distribution is computed as:

$$P(y_{M+1,new}|D_{M+1},D,\eta) = \int P(y_{M+1,new}|\theta_{M+1})P(\theta_{M+1}|D_{M+1},D,\eta)d\theta_{M+1}. \tag{3.41}$$

Now we discuss some computational details in the full Bayesian framework. The key inference problem is the computation of the posterior distribution $P(\alpha,\{\theta_j\}_{j=1}^M|D,\eta)$, unfortunately, it is analytically intractable with respect to a large number of unknown parameters. A typical solution for the problem is the Markov chain Monte Carlo algorithm, e.g., Gibbs sampling (GS) which is applicable when the variables have a small finite set of states, or are easily sampled from their conditional distributions. We now briefly introduce the main idea of the GS method. Suppose the whole of variables to be sampled are ξ, we divide them into m subsets $\xi = \{\xi_1,\ldots\xi_m\}$. In each iteration, the Gibbs sampler draws each subset of variables conditioned on all others. The procedure at the iteration t is executed as:

- Sample $\xi_1^{(t)}$ conditioned on $\xi_2^{(t-1)}$, $\xi_3^{(t-1)}$,..., $\xi_m^{(t-1)}$.
- Sample $\xi_2^{(t)}$ conditioned on $\xi_1^{(t)}$, $\xi_3^{(t-1)}$,..., $\xi_m^{(t-1)}$.
- ...
- Sample $\xi_m^{(t)}$ conditioned on $\xi_1^{(t)}$, $\xi_2^{(t)}$,..., $\xi_{m-1}^{(t)}$.

The above steps are performed $W + w$ iterations. The first w iterations are discarded as *burn-in* period. The last W members of the sequence are collected and are averaged over in order to get the desired distributions. For more details about MCMC algorithms, please refer to (Andrieu et al., 2003; Gilks et al., 1995). In the full-Bayesian hierarchical model, the Gibbs sampler yields the following steps:

1. Draw the hyperparameters α from the distribution $P(\alpha|\{\theta_j\}_{j=1}^M, \eta)$.
2. Draw the i.i.d. parameters θ_j's from $P(\theta_j|D_j, \alpha)$.
3. If the predictive distribution about a new observation $y_{j,new}$ is desired, draw $y_{j,new}$ from the distribution $P(y_{j,new}|\theta_j)$.

After the procedure converges, the desired distributions, say $P(\alpha|D, \eta)$ and $P(y_{j,new}|D, \eta)$, can be approximated as:

$$P(\alpha|D, \eta) \approx \frac{1}{W} \sum_{t=w+1}^{W+w} P(\alpha^{(t)}|\{\theta_j^{(t-1)}\}_{j=1}^M, \eta). \tag{3.42}$$

$$P(y_{j,new}|D, \eta) \approx \begin{cases} 0, & y_{j,new} < y_{min}, \\ \frac{N_\ell}{W}, & y_\ell < y_{j,new} < y_{\ell+1}, \\ 0, & y_{j,new} > y_{max}. \end{cases} \tag{3.43}$$

Where y_{min} and y_{max} are the minimum and maximum in the sequence of samples $y_{j,new}^{(t)}$. N_ℓ is the number of samples in the interval $[y_\ell \;\; y_{\ell+1}]$.

3.3 Summary

Bayesian analysis sets up a full probability model to fit a set of observations and to summarize the results about the model parameters or other unobserved quantities. What distinguishes Bayesian analysis from other statistic analysis is that the unknown parameters are represented as random variables. The advantage is that Bayesian analysis can explicitly use the probability to quantify uncertainty in inferences. Estimation or prediction is performed in terms of both prior knowledge and known data. More details about Bayesian analysis can be obtained in (Berry, 1996; Congdon, 2001; Congdon, 2003; Gelman et al., 2004).

Hierarchical Bayesian (HB) modeling is designed for the situations where the hyperparameters are unknown. An early thorough introduction to HB modeling is provided by

Good (1965). Deely and Lindley (1981) extended empirical Bayesian framework to full Bayesian framework. HB is widely applied to the meta analysis in machine learning area.

In Bayesian analysis, for computational efficiency, we often assume a conjugate prior, i.e. the prior and the posterior distributions are of the same mathematic form. The advantage is that the assumption reduces the function approximation to parameter approximation in computing the posterior distribution. The constraint is that the mathematic form of the prior distribution is expected to be flexible enough not only to represent one's vague prior belief, but also to represent the learned posterior. However, a parametric model is often too strict to come up the expectation. To solve the limitation, nonparametric Bayesian modeling is considered, which will be discussed in Chapter 4.

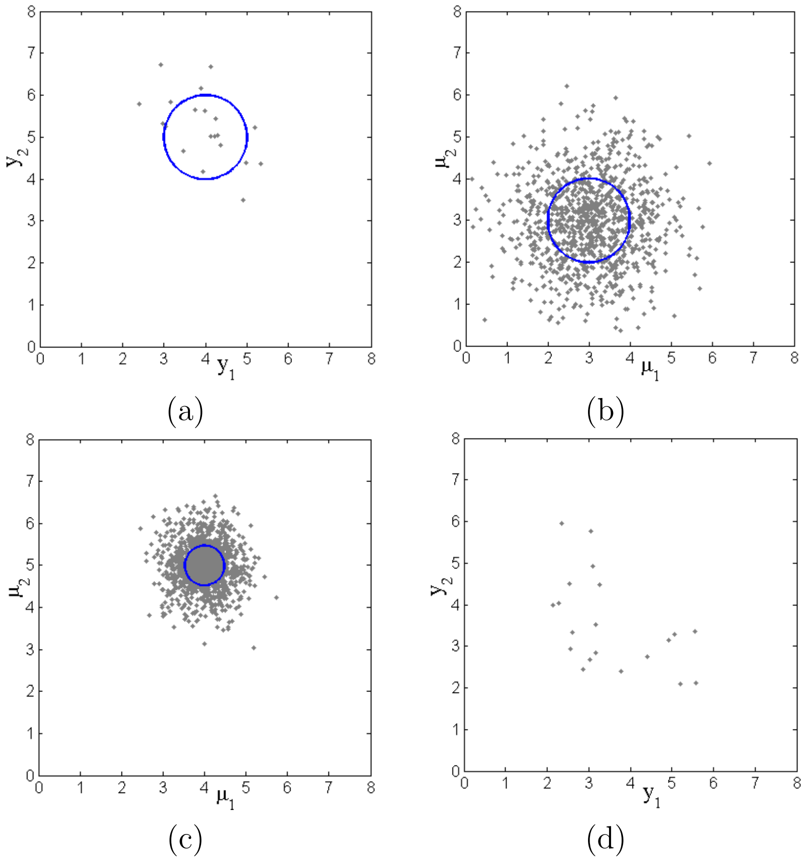
Chapter 4

Nonparametric Hierarchical Bayesian Models

4.1 Introduction

The models in Chapter 3 belong to the *parametric* Bayesian models, since they are strongly dependent on the parametric assumption. However, the assumption is not always practical. For example, in more cases than not, it is not easy to known the mathematic form of the likelihood distribution in advance, thus it is impossible to built a model with a finite set of parameters. Furthermore, if a wrong mathematic form is specified to the likelihood distribution, then the estimation will be completely divergent from the real situation, since the inference methods in the parametric models are closely connected with the specific functional forms of the distributions, if the functional forms change, then the methods with good effect in the original models might lead to inferior result. To remove the constraints, the *nonparametric* Bayesian models are developed to learn the functions of interest directly from the data, e.g. the probability distribution in the task of density estimation. The term *nonparametric* does not mean there are no parameters in the models, but that the number and properties of the parameters are flexible and not fixed in advance. Nonparametric models are therefore also called *distribution free*. The meanings of the two terms are slightly different, but are often used interchangeably. There are different definitions for the nonparametric Bayesian models. For example, Bernardo and Smith (1994) defined nonparametric Bayesian models as probability models with infinite parameters. Mueller and Quintana (2004) defined nonparametric Bayesian models as probability models on function spaces. The two definitions are equivalent, but focus on the different perspectives.

Many nonparametric Bayesian models are developed for various statistical problems, for example, density estimation, regression, survival time analysis and so on. In statistic machine learning, the well-known nonparametric Bayesian models include *Dirichlet process* and *Gaussian process*. The term *process* means that the degrees of freedom of the model are infinite. Generally, Dirichlet process is used in density estimation, clustering; Gaussian process is used in regression, classification, and so on. In this chapter, we introduce the application of Dirichlet process (DP) in the hierarchical Bayesian modeling.



(a) (b)

(c) (d)

Figure 4.1: (a) A set of N observations, which are i.i.d. drawn from a Gaussian distribution with unknown mean μ and known covariance matrix Σ. (b) A conjugate prior distribution of the unknown parameter μ. (c) The learned posterior distribution of μ. So far the Bayesian inference is performed in an ideal situation where the data really follows Gaussian distribution as we assume. However, in many cases, the observations are not distributed as assumed. (d) The data is an arbitrary distribution, which can not be represented by a Gaussian with any parameters. Then prediction based on the Gaussian model will be divergent from the real situation.

4.2 Model Description

In this section, we will first introduce application of the nonparametric framework on Bayesian models and then extend it to the hierarchical Bayesian (HB) models. Assume that there are N observations $D = \{y_1, y_2, \ldots, y_N\}$. Each observation y_i is a continuous two-dimensional random variable, and the two dimensions are independent with each other, i.e. the covariance is 0. In Bayesian modeling, we assume that the observations are i.i.d. drawn from a multivariate Gaussian distribution with mean $\mu = (\mu_1, \mu_2)$ and covariance matrix $\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$. To simplify the model, we further assume that Σ is known, but μ is unknown. As usual we assume a conjugate prior, i.e. μ is drawn from a multivariate Gaussian distribution, $\mu \sim N(\mu_{prior}, \Sigma_{prior})$. The hyperparameters

$\mu_{prior} = (\mu_{prior,1}, \mu_{prior,2})$ and $\Sigma_{prior} = \begin{bmatrix} \sigma^2_{prior,1} & 0 \\ 0 & \sigma^2_{prior,2} \end{bmatrix}$ are mean and covariance matrix of the prior distribution, respectively. Given the Bayesian model and the observations D, the computation of the posterior distribution is straightforward, which is still a Gaussian distribution with parameters $\mu_{post} = (\mu_{post,1}, \mu_{post,2})$ and $\Sigma_{post} = \begin{bmatrix} \sigma^2_{post,1} & 0 \\ 0 & \sigma^2_{post,2} \end{bmatrix}$, where

$$\mu_{post,1} = \frac{\frac{\mu_{prior,1}}{\sigma^2_{prior,1}} + \frac{\sum_{i=1}^{N} y_i^1}{\sigma_1^2}}{\frac{1}{\sigma^2_{prior,1}} + \frac{N}{\sigma_1^2}}; \quad \sigma^2_{post,1} = (\frac{1}{\sigma^2_{prior,1}} + \frac{N}{\sigma_1^2})^{-1},$$

$\mu_{post,2}$ and $\sigma^2_{post,2}$ are computed in an equivalent way. Figure 4.1(a) shows the known data, which is distributed as Gaussian. The prior and the posterior distributions of μ are shown as Figure 4.1(b) and (c). So far the Bayesian inference is performed in an ideal situation where the data really follows Gaussian distribution as we assume. However, in many cases, the observations y_i's are not exactly Gaussian, but an arbitrary distribution, e.g. a distribution shown as Figure 4.1(d), which can not be approximated by a Gaussian distribution with any parameters. To solve this problem, it is nature to embed the Bayesian model in a nonparametric framework, i.e., consider the likelihood distribution itself, rather than the parameters, as a random variable. That means we do not specify the functional form of the likelihood distribution in advance. Therefore, what we learn from the data is the probability distribution itself, rather than the parameters. Note, that the prior distribution in the nonparametric model is not a distribution over parameter space, but a distribution over a set of distributions. Furthermore, the data in the nonparametric model can be any arbitrary distribution without the limitation about scope and type. Figure 4.2(b) shows the nonparametric model. In contract with the parametric model shown as Figure 4.2(a), the likelihood is an arbitrary distribution G drawn from $P(G)$, rather than a distribution with a specific mathematic form and unknown parameters. From the figure, it is clear how the samples are generated in the nonparametric Bayesian model. Given a prior $P(G)$, specifying the probability of the likelihood, a sample distribution G is drawn and then the samples y_i are i.i.d. drawn from G.

Now we introduce how to apply the nonparametric framework to the hierarchical Bayesian (HB) model. In HB model, the common prior of the parameters is of central importance. It is expected to be flexible enough to represent the true situation. However, in many cases, a parametric prior is often too strict to meet the expectation. Therefore we consider to embed the hierarchical Bayesian modeling in the nonparametric framework, i.e. the unknown prior G is a sample distribution drawn from a probability model $P(G)$, such that G can be of any mathematic form to truthfully represent the learned knowledge. Assume that there are M parallel data sets $D = \{D_1, D_2, \ldots, D_M\}$, and in the data set D_j, there are N_j observations $D_j = \{y_{j,1}, y_{j,2}, \ldots, y_{j,N_j}\}$. Assume that the likelihood distribution of each data set D_j is of the same functional form but distinct parameters θ_j. The θ_j's share a common prior. Figure 4.3 shows a parametric HB model and a nonparametric HB model for the example. What distinguishes nonparametric model from parametric model is that in the nonparametric model the prior can be any arbitrary distribution, not a distribution with specific form. The generative process of the

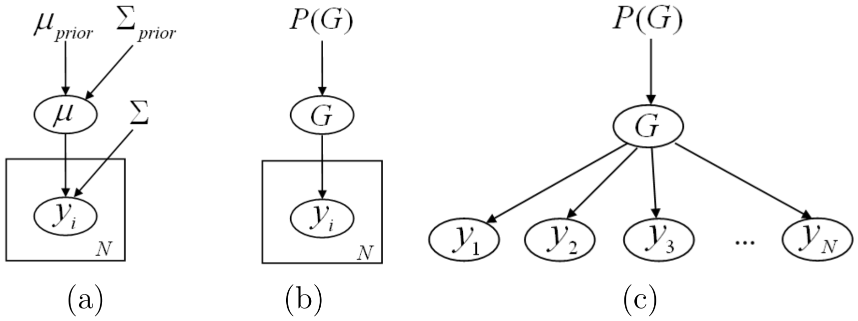


Figure 4.2: (a) A parametric Bayesian model for $D = \{y_1, y_2, \ldots, y_N\}$. The observations are i.i.d. drawn from a Gaussian distribution with parameters μ and Σ. We assume Σ is known but μ is unknown and follows a Gaussian distribution with hyperparameters μ_{prior} and Σ_{prior}. (b) A nonparametric Bayesian model in the same setting. In contract with the parametric model, the likelihood is an arbitrary distribution G drawn from $P(G)$, rather than a distribution with specific mathematic form and unknown parameters. (c) The equal model to (b).

nonparametric HB model is as follows:

$$\begin{aligned} G|P(G) &\sim P(G). \\ \theta_j|G &\sim G(\theta_j) \text{ for } j = \{1, \ldots, M\}. \\ y_{j,i}|\theta_j &\sim P(y_{j,i}|\theta_j) \text{ for } i = \{1, \ldots, N_j\}. \end{aligned}$$

Of central importance in nonparametric framework are the unknown distribution G and its probability model $P(G)$. Generally G is called *random probability distribution (RPD)*. Ferguson (1973) and Antoniak (1974) stated two desirable properties of $P(G)$. First, it should be largely supported, i.e. $P(G)$ is expected to cover most of the probability distributions on a given sample space. Second, the posterior inference should be computationally manageable, since the integral on the infinite function space is difficult. So far, many probabilistic models about $P(G)$ have been developed, including Dirichlet Process (DP), invariant DP, Pólya Trees, Bernstern Polynomials, logistic normal process and so on, in which DP is commonly used in the area of statistic machine learning. Dirichlet process is generally denoted as $\text{DP}(\alpha_0, G_0)$, where α_0 and G_0 are the parameters. The strategy, replacing the parametric prior distribution with a sample from DP, is called *Dirichlet enhancement* (Escobar & West, 1998), which extends the flexibility of the parametric Bayesian modeling by encoding the additional uncertainty about the functional form of the prior. As an important result, Dirichlet enhanced models not only represent one's prior knowledge via the parameters of DP, i.e. α_0 and G_0, but also make the prior G (i.e. a sample distribution from DP) as complex as necessary to model the real situation. In the next section, we introduce some detailed information about DP.

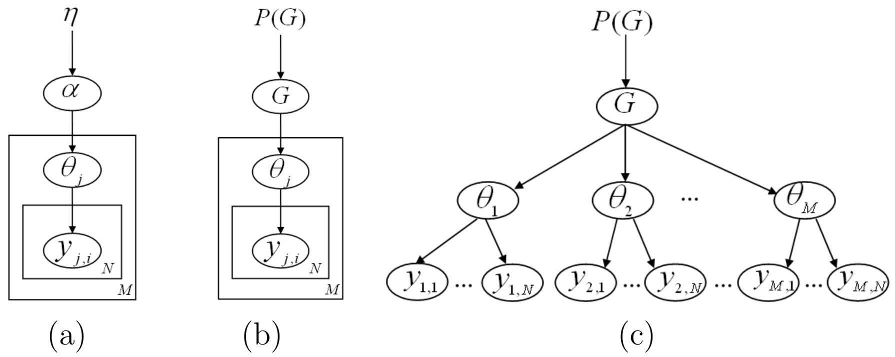


Figure 4.3: (a) A parametric hierarchical Bayesian (HB) model. Assume that there are M parallel data sets $D = \{D_1, D_2, \dots, D_M\}$, and in D_j, there are N_j observations $D_j = \{y_{j,1}, y_{j,2}, \dots, y_{j,N_j}\}$. (b) A nonparametric HB model in the same setting. The difference is that in the nonparametric model the prior can be any arbitrary distribution as complex as necessary, rather than a distribution with assumed form. (c) The equal model to (b).

4.3 Dirichlet Process

Dirichlet process, introduced by Ferguson (1973), defines a prior for random probability distributions. It can be viewed as an extension of Dirichlet distribution from finite dimensions to infinite dimensions.

4.3.1 Dirichlet Distribution

Dirichlet distribution is known as a conjugate prior of a multinomial distribution. It is closely connected with Gamma distribution. Assume that each dimension of a random vector $\nu = (\nu_1, \dots, \nu_K)$ is i.i.d. drawn from a Gamma distribution with a shape parameter α_k and a scale parameter 1, i.e. $\nu_k \sim \text{Gamma}(\alpha_k, 1)$. Then the random vector $\theta = (\theta_1, \dots, \theta_K)$, where

$$\theta_k = \frac{\nu_k}{\sum_{k=1}^{K} \nu_k}, \tag{4.2}$$

follows Dirichlet distribution with parameters $\alpha = (\alpha_1, \dots, \alpha_K)$. Let $\alpha_0 = \sum_{k=1}^{K} \alpha_k$, the Dirichlet distribution is defined as:

$$P(\theta_1, \dots, \theta_K | \alpha) = \frac{\Gamma(\alpha_0)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1}. \tag{4.3}$$

Note, that θ is a $K-1$ dimensional random vector, since $\sum_{k=1}^{K} \theta_k = 1$. When $K = 2$, a Dirichlet distribution reduces to a Beta distribution. Some major properties of Dirichlet distribution are listed as follows:

1. Let $y_1, \dots, y_N$ be discrete samples with S states. y_i is i.i.d. drawn from a multinomial distribution with parameters θ, which prior is a Dirichlet distribution with

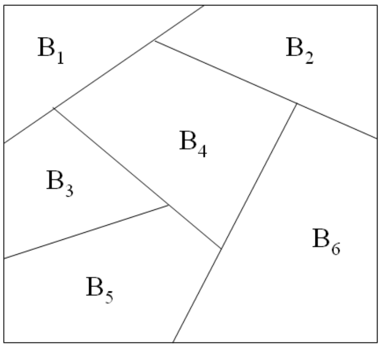


Figure 4.4: An example partition $\{B_1, \ldots, B_6\}$ on a 2-dimensional continuous space B. Let G_0, G be a specific distribution and a random distribution on B. If $G \sim \mathrm{DP}(\alpha_0, G_0)$, then the random vector $(G(B_1), \ldots, G(B_6))$ is drawn from a Dirichlet distribution with parameters $(\alpha_0 G_0(B_1), \ldots, \alpha_0 G_0(B_6))$.

hyperparameters α. Then posterior distribution of θ given the N samples is still Dirichlet with parameters $\alpha_{post} = (\alpha_1 + N_1, \ldots, \alpha_S + N_S)$, where N_s is the number of samples with state s.

2. Let random vector $\theta = (\theta_1, \ldots, \theta_K)$ follow Dirichlet distribution with parameters α. $I_1, \ldots, I_L$ denote L integers, and $0 < I_1 < \cdots < I_L = K$, then the random vector $\theta' = (\sum_{k=1}^{I_1} \theta_k, \sum_{k=I_1+1}^{I_2} \theta_k, \ldots, \sum_{k=I_{L-1}+1}^{I_L} \theta_k)$ is still Dirichlet with parameters $(\sum_{k=1}^{I_1} \alpha_k, \sum_{k=I_1+1}^{I_2} \alpha_k, \ldots, \sum_{k=I_{L-1}+1}^{I_L} \alpha_k)$.

3. If a random vector $\theta = (\theta_1, \ldots, \theta_K)$ follows a Dirichlet distribution, then marginal distribution of θ_k is $\mathrm{Beta}(\alpha_k, \alpha_0 - \alpha_k)$, and the expectation of θ_k is $\mathbb{E}(\theta_k) = \alpha_k/\alpha_0$.

4.3.2 Basic Properties of DP

Dirichlet process is a distribution on a set of distributions. DP is indexed by two parameters: the base distribution G_0 and concentration parameter α_0. Let θ be random variables. G_0 is a probability distribution over the space of θ. α_0 is a positive real-value scalar. A random measure G is distributed according to a Dirichlet process with parameters G_0 and α_0, if for all positive integer K and any partition $\{B_1, \ldots, B_K\}$ on the space of θ, the random probabilities $(G(B_1), \ldots, G(B_K))$ follows a Dirichlet distribution:

$$(G(B_1), \ldots, G(B_K)) \sim \mathrm{Dir}(\alpha_0 G_0(B_1), \ldots, \alpha_0 G_0(B_K)). \tag{4.4}$$

To make the definition easy to understand, we illustrate with a simple example. Assume that there is a 2-dimensional continuous space B. G_0 is a Gaussian distribution over B. G is a random probability distribution on B and is drawn from a Dirichlet process, $G \sim \mathrm{DP}(\alpha_0, G_0)$. For any partition of the space B, e.g. a partition $\{B_1, \ldots, B_6\}$ as Figure 4.4, we have $G_0(B_1) + \cdots + G_0(B_6) = 1$ and $G(B_1) + \cdots + G(B_6) = 1$. Note, that the vector $(G(B_1), \ldots, G(B_6))$ is random due to the randomness of the distribution

G. Since G is drawn from $\mathrm{DP}(\alpha_0, G_0)$, the random vector $(G(B_1), \ldots, G(B_6))$ follows a Dirichlet distribution with parameters $(\alpha_0 G_0(B_1), \ldots, \alpha_0 G_0(B_6))$:

$$(G(B_1), \ldots, G(B_6)) \sim \mathrm{Dir}(\alpha_0 G_0(B_1), \ldots, \alpha_0 G_0(B_6)).$$

The two parameters of DP can be explained intuitively. G_0 represents one's prior belief, α_0 measures the strength of one's belief in G_0. For large values of α_0, a sampled G is likely to be close to G_0. For small values of α_0, a sampled G is likely to put most of its probability mass on just a few atoms. Walker et al. (1999) provided more detailed discussion about the two parameters.

The properties of DP were explored by Ferguson (1973). Here we only introduce a fundamental theorem about posterior updating, which is important for DP inference. The theorem is comparable with the posterior computation of Dirichlet distribution.

Theorem 4.1 *Given a Dirichlet process* $\mathrm{DP}(\alpha_0, G_0)$ *and a set of samples* $\theta_1, \ldots, \theta_N$. *The posterior Dirichlet process is* $\mathrm{DP}(\alpha_0 + N, G_0^{post})$, *where* $G_0^{post} \propto G_0 + \sum_{i=1}^{N} \delta_{\theta_i}$, *and* δ_{θ_i} *is a point mass at* θ_i.

4.3.3 Pólya Urn Representation

According to the definition of DP, introduced in last section, it is difficult to draw the random probability distribution G directly, since the probability function space is infinite. To remove the computational constraint, Blackwell and MacQueen (1973) introduced the Pólya urn representation. Intuitively, the urn process is performed as follows. Assume that there are many balls with different colors in an urn. One draws balls with probability distribution G_0. If a ball is drawn, one puts back the ball and an additional ball with the same color, thus after a sequence of draws, balls with a color already encountered become more likely to be drawn again. The essential property of urn process is that if a state is sampled previously, the probability that the state is sampled again is increased. Note, that there is no need to draw G directly. Formally, the urn process is defined as follows:

1. The first sample θ_1 is drawn from the base distribution G_0.

2. Conditioned on previous $N-1$ samples $\theta_1, \ldots, \theta_{N-1}$, the sample θ_N is drawn from the distribution:

$$P(\theta_N|\theta_1, \ldots, \theta_{N-1}, \alpha_0, G_0) = \frac{\alpha_0 G_0 + \sum_{i=1}^{N-1} \delta_{\theta_i}}{\alpha_0 + N - 1}, \tag{4.5}$$

where δ_{θ_i} is a distribution with a point mass on θ_i. The distribution can be viewed as a *mixed* distribution (in analogy to a discrete or continuous distribution). It consists of one continuous component G_0 and $N-1$ discrete components δ_{θ_i}. At the points $\{\theta_1, \theta_2, \ldots, \theta_{N-1}\}$, the distribution is discrete with probability $\frac{1}{\alpha_0+N-1}$, on the left space, the distribution is continuous with density G_0. Assume that in the sequence of $N-1$ samples, there are $K \leq N-1$ distinct values $\{\theta_1^*, \ldots, \theta_K^*\}$. Let N_k denote the number of

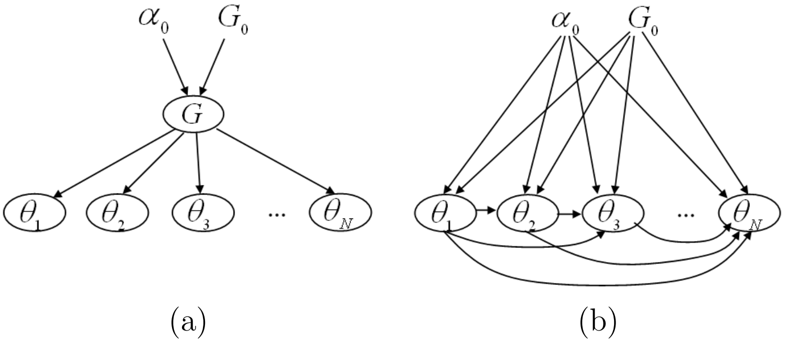


Figure 4.5: (a) Graphic representation of DP, where the random probability distribution G is explicitly drawn from the DP. (b) Graphic representation of Pólya Urn process of DP, where G is integrated out.

times the value θ_k^* occurs in the sequence and $\sum_k N_k = N-1$. The probability of θ_N is simplified as:

$$P(\theta_N|\theta_1,\ldots,\theta_{N-1},\alpha_0,G_0) = \begin{cases} \frac{N_k}{\alpha_0+N-1}, & \text{if } \theta_N = \theta_k^* \\ \frac{\alpha_0}{\alpha_0+N-1}, & \text{else.} \end{cases} \tag{4.6}$$

In particular, with probability $\frac{N_k}{\alpha_0+N-1}$, the new sample takes on an existing value θ_k^*; with probability $\frac{\alpha_0}{\alpha_0+N-1}$, the new sample draws a new value from the distribution G_0. Figure 4.5 shows the Pólya Urn process in a graphic representation, note that in Figure 4.5(b) G is integrated out.

In terms of Equation 4.6, the effect of the concentration parameter α_0 is clarified. The larger α_0 is, the more likely new atoms are drawn. In the limiting case $\alpha_0 \to \infty$, the distribution $P(\theta_N|\theta_1,\ldots,\theta_{N-1},\alpha_0,G_0)$ approaches the base distribution G_0. As α_0 is very small, the distribution $P(\theta_N|\theta_1,\ldots,\theta_{N-1},\alpha_0,G_0)$ is largely based on existing θ_k^*. Figure 4.6 shows the samples drawn from DPs with the same base distribution G_0 but different concentration parameter α_0. Figure 4.6(e) is the histogram of the samples drawn from a DP with $\alpha_0 = 100000$, which closely approximates to G_0 shown as Figure 4.6(f).

4.3.4 Other Representations

Besides the urn representation, there are some other representations for Dirichlet process, including Chinese restaurant process introduced by Aldous (1985) and stick breaking construction introduced by Sethuraman (1994). The two representations focus on the discreteness property of DP and are often applied to the mixture models, more details can be found in Part III. Of course, the two representations are also applicable in hierarchical models, since they are just alternative methods to draw samples from DP.

4.4 Inference

The key inferential problem in the Dirichlet enhanced HB model is to compute the joint posterior distribution of the unknown variables given the parallel data sets $D =$

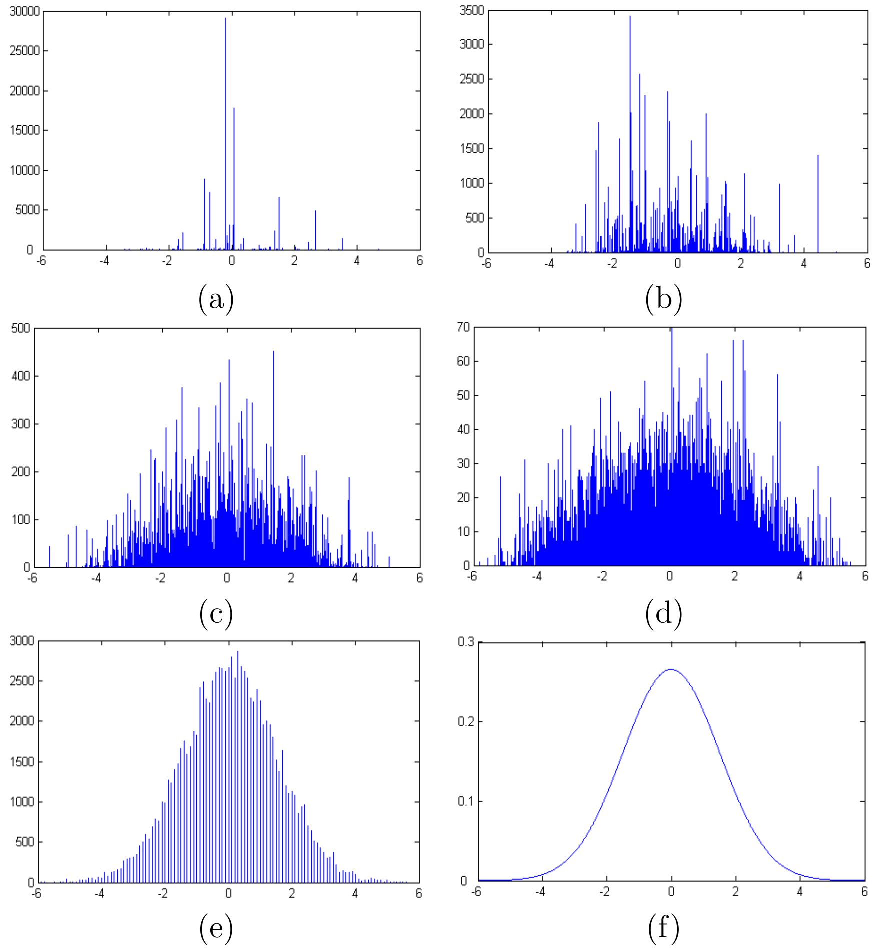


Figure 4.6: The 100000 samples drawn from DPs with the same base distribution G_0 and different concentration parameter α_0. G_0 is a Gaussian distribution with mean 0 and standard deviation 1.5. More atoms are drawn with increasing α_0: (a) $\alpha_0 = 10$ (b) $\alpha_0 = 100$ (c) $\alpha_0 = 1000$ (d) $\alpha_0 = 10000$. (e) The histogram of the samples drawn from a DP with $\alpha_0 = 100000$. (f) The base distribution G_0. It is obvious that with a large concentration parameter α_0, the samples are distributed closely as G_0.

$\{D_1, \ldots, D_M\}$. As discussed in Section 4.2, the unknown variables in the model include: the prior G and the parameters $\{\theta_1, \ldots, \theta_M\}$, one for each data set. Thus the posterior distribution is defined as:

$$P(G, \theta_1, \ldots, \theta_M | D, \alpha_0, G_0) = \frac{P(G|\alpha_0, G_0) \prod_{j=1}^{M} P(\theta_j|G) P(D_j|\theta_j)}{P(D|\alpha_0, G_0)}. \tag{4.7}$$

It is clear that the equation is intractable, since direct computation of hyperprior $P(G|\alpha_0, G_0)$ is impossible. To solve the problem, we consider to integrate out G via Pólya urn representation, such that Equation 4.7 is simplified as:

$$\begin{aligned} P(\theta_1, \ldots, \theta_M | D, \alpha_0, G_0) &= \frac{\prod_{j}^{M} P(\theta_j|\alpha_0, G_0, \{\theta_{j'}\}_{j'<j}) P(D_j|\theta_j)}{P(D|\alpha_0, G_0)} \\ &= \prod_{j}^{M} \frac{\alpha_0 G_0(\theta_j) + \sum_{j'<j} \delta_{\theta_{j'}}(\theta_j)}{(\alpha_0 + j - 1) P(D|\alpha_0, G_0)} P(D_j|\theta_j). \end{aligned} \tag{4.8}$$

The computation in Equation 4.8 is still analytically intractable. A typical solution for the problem is the Markov chain Monte Carlo method, e.g. (West et al., 1994; Escobar & West, 1998). For computational efficiency, an alternative solution is introduced by Tresp and Yu (2004), where an approximation to $P(\theta_j|\alpha_0, G_0, \{\theta_{j'}\}_{j'<j})$ is computed via variational inference method. Other solutions include sequential importance sampling-based methods, predictive recursion, and so on. In this chapter, we focus on Gibbs sampling and variational approximation algorithms. More details about other methods please refer to, e.g., (Liu, 1996; MacEachern et al., 1999; Newton & Zhang, 1999; Quintana & Newton, 2000).

Additionally, Equation 4.7 can also be reduced via other representations of DP, e.g. stick breaking construction and Chinese restaurant process. The corresponding inference methods please refer to, e.g., (MacEachern, 1994; Escobar & West, 1995; Ishwaran & James, 2001; Gelfand & Kottas, 2002; Blei & Jordan, 2005).

4.4.1 Inference with Gibbs Sampling

Traditionally, posterior inference in nonparametric Bayesian models is performed via Markov Chain Monte Carlo (MCMC) methods. There are many advanced approaches proposed, such as (Escobar & West, 1995; Escobar & West, 1998; Tresp & Yu, 2004). The main challenge in the sampling process is how to sample the prior G directly from $P(G|\{\theta_j\}_{j=1}^{M}, \alpha_0, G_0)$. Although there is Theorem 4.1 about posterior updating of DP, the sampling is still not easy to perform. However, if integrating out the random probability distribution G as shown in Equation 4.8, the sampling process is simplified and the variables to be sampled are only parameters $\{\theta_1, \ldots, \theta_M\}$. If θ_i can be easily sampled from $P(\theta_i|G_0)$, we appeal to Gibbs sampling (GS) method to approximate the posterior of interest. In particular, at each iteration the GS method draws each θ_j conditioned on the samples of other parameters $\{\theta_{j'}\}_{j' \neq j}$, the distribution $P(\theta_j|\{\theta_{j'}\}_{j' \neq j}, D_j, \alpha_0, G_0)$ is proportional to:

$$P(D_j|\theta_j)\left[\alpha_0 G_0(\theta_j)+\sum_{j'\neq j}\delta_{\theta_{j'}}\right]. \tag{4.9}$$

The equation is the product of the prior distribution $P(\theta_j|\{\theta_{j'}\}_{j'\neq j},\alpha_0,G_0)$ represented as urn process and the likelihood distribution $P(D_j|\theta_j)$. Particularly, the parameter θ_j is assigned:

1. An existing value $\theta_{j'}$ with probability proportional to
$$P(D_j|\theta_{j'}) \tag{4.10}$$
2. A new value with probability proportional to
$$\alpha_0\int G_0(\theta_j)P(D_j|\theta_j)d\theta_j. \tag{4.11}$$
   The new value is drawn from the distribution
$$\frac{1}{C}G_0(\theta_j)P(D_j|\theta_j), \tag{4.12}$$
   where $C=\int G_0(\theta_j)P(D_j|\theta_j)d\theta_j$ is a normalization constant.

To perform the GS method, the integration $\int G_0(\theta_j)P(D_j|\theta_j)d\theta_j$ needs to be computed. It is tractable if $P(D_j|\theta_j)$ and $G_0(\theta_j)$ are assumed to be of manageable form. For example, we assume that $P(D_j|\theta_j)$ is a distribution in the exponential family and $G_0(\theta_j)$ is conjugated with $P(D_j|\theta_j)$. The assumption is widely used in Bayesian modeling. In the nonparametric hierarchical models, the assumption is not so strong, since G_0 is just a parameter of Dirichlet process, even if the mathematic form of G_0 is specified, the prior G can be arbitrary distribution as complex as necessary. If $P(D_j|\theta_j)$ and $G_0(\theta_j)$ are not of manageable form, the integration might be difficult. A possible strategy to calculate the integration $\int G_0(\theta_j)P(D_j|\theta_j)d\theta_j$ is numerical method. Since there is only one variable θ_j, the dimension of the integral is the dimension of θ_j. Therefore, it is potentially a low dimensional integral problem and can be performed efficiently via e.g. Gaussian quadrature (GQ) method. For more details about GQ method, please refer to (Naylor & Smith, 1982; Evans & Swartz, 1995). In the case that the integral can not be calculated efficiently, we consider some alternative sampling algorithms, which perform MCMC without computing the integral, e.g. (MacEachern & Mueller, 1998).

In summary, the unknown parameters $\{\theta_1,\ldots,\theta_M\}$ are sampled in the following steps:

1. For each data set D_j, initialize $\theta_j^{(t)}$ from G_0, $t=0$.
2. Iterate the following steps for $t=1,2,\ldots$.
   - Update $\theta_j^{(t)}$ given $\{\theta_1^{(t)},\ldots,\theta_{j-1}^{(t)},\theta_{j+1}^{(t-1)},\ldots,\theta_M^{(t-1)}\}$.

- Assign $\theta_j^{(t)}$ an existing value $\theta_{j'}^{(t-1)}$ with probability proportional to

$$P(D_j|\theta_{j'}^{(t-1)}) \tag{4.13}$$

- Instead, a new value is generated with probability proportional to

$$\alpha_0 \int G_0(\theta_j^{(t)})P(D_j|\theta_j^{(t)})d\theta_j^{(t)}. \tag{4.14}$$

  The new value is drawn from the distribution $\frac{1}{C}G_0(\theta_j^{(t)})P(D_j|\theta_j^{(t)})$, where $C = \int G_0(\theta_j^{(t)})P(D_j|\theta_j^{(t)})d\theta_j^{(t)}$ is a normalization constant. Note, that the sampling process might be cheaply implemented, since integration in Equation 4.14 and normalization constant C are not changeable between iterations, thus we only compute the value once for each data set D_j.

- $j \leftarrow j + 1$, go to update the next $\theta_j^{(t)}$.

3. Stop until a stationary point reaches.

4.4.2 Inference with Variational Method

MCMC methods are successful solutions to approximate a conditional probability distribution. Such methods are accurate, however there are two main limitations. First, the efficiency can be low, especially when the data are large-scale, multivariate or highly-correlated. Second, it is not easy to diagnose the convergence. To remove these constraints, variational inference methods are considered. There are two main classes of variational algorithms: *sequential* and *block*. In the chapter, we focus on the block variational approaches, which are particularly suitable in the situations where the subsets of variables are amenable to exact inference. Suppose the distribution of interest is $P(\xi)$ and the exact computation of $P(\xi)$ is intractable. Thus we expect to find a distribution $q(\xi)$, referred to as a *variational distribution*, to approximate $P(\xi)$ as close as possible. Assume $q(\xi)$ can be any distribution over the domain of ξ. The difference between $q(\xi)$ and $P(\xi)$ can be measured via *Kullback-Leibler* (KL) divergence:

$$KL(q(\xi)||P(\xi)) = \sum_{\xi} q(\xi)\log q(\xi) - \sum_{\xi} q(\xi)\log P(\xi), \tag{4.15}$$

which is sometimes referred to as *variational free energy*. The minimum is 0 when $P(\xi) = q(\xi)$. The larger the divergence is, the more different the two distributions are. Thus the probabilistic inference problem (i.e. computing $P(\xi)$) is converted into an optimization problem: minimize the KL divergence with respect to the variational distribution. It is clear that the optimization problem is not easy to solve. Many efforts are made to find suitable forms of $q(\xi)$ to make the problem computationally tractable. For more details about variational inference methods, please refer to (Jordan et al., 1998). In the nonparametric HB model, Yu et al. (2004) and Tresp and Yu (2004) proposed a variational method to approximate the posterior distribution of $\{\theta_j\}_{j=1}^M$ in Equation 4.8.

For computational efficiency, a family of fully-factorized distributions are assumed, $q(\theta_1, \ldots, \theta_M) = \prod_{j=1}^{M} q(\theta_j)$, and for each θ_j, the variational distribution is defined as:

$$\begin{aligned} q(\theta_j) &= \sum_{k=1}^{M} \omega_{j,k} \delta_{\theta_k^{ML}}, \qquad j = 1, \ldots, M; \\ \theta_k^{ML} &= \arg\max_{\theta_k} P(D_k|\theta_k). \end{aligned} \tag{4.16}$$

Where θ_k^{ML} denotes maximum-likelihood estimation of the parameters of the data set D_k. It is obvious that $q(\theta_j)$ is a discrete distribution at points $\{\theta_1^{ML}, \ldots, \theta_M^{ML}\}$, and θ_j equals θ_k^{ML} with probability $\omega_{j,k}$, i.e. $P(\theta_j = \theta_k^{ML}) = \omega_{j,k}$ and $\sum_{k=1}^{M} \omega_{j,k} = 1$. The KL divergence between the variational distribution and real distribution is now written as:

$$\begin{aligned} KL(q||P) = \mathbb{E}_q[\log \prod_{j=1}^{M} q(\theta_j)] - \mathbb{E}_q[\log P(\{D_j, \theta_j\}_{j=1}^{M}|\alpha_0, G_0)] \\ + \log P(\{D_j\}_{j=1}^{M}|\alpha_0, G_0). \end{aligned} \tag{4.17}$$

The posterior inference problem is now converted to minimize the KL divergence with respect to the variational parameters $\omega_{j,k}$, $j, k = \{1, \ldots, M\}$. Permuting the above equation, we get an inequality about the log-likelihood of the data:

$$\begin{aligned} &\log P(\{D_j\}_{j=1}^{M}|\alpha_0, G_0) \\ &= \mathbb{E}_q[\log P(\{D_j, \theta_j\}_{j=1}^{M}|\alpha_0, G_0)] - \mathbb{E}_q[\log \prod_{j=1}^{M} q(\theta_j)] + KL(q||P) \\ &\geq \mathbb{E}_q[\log P(\{D_j, \theta_j\}_{j=1}^{M}|\alpha_0, G_0)] - \mathbb{E}_q[\log \prod_{j=1}^{M} q(\theta_j)] \end{aligned} \tag{4.18}$$

The right terms define a *lower bound* of the log-likelihood of the data. The difference between the lower bound and the log-likelihood is the KL divergence. Alternatively, the lower bound can also be derived via the *Jensen's inequality*:

$$\begin{aligned} &\log P(\{D_j\}_{j=1}^{M}|\alpha_0, G_0) \\ &= \log \sum_{\theta_1, \ldots, \theta_M} P(\{D_j, \theta_j\}_{j=1}^{M}|\alpha_0, G_0) \\ &= \log \sum_{\theta_1, \ldots, \theta_M} \frac{\prod_{j=1}^{M} q(\theta_j) P(\{D_j, \theta_j\}_{j=1}^{M}|\alpha_0, G_0)}{\prod_{j=1}^{M} q(\theta_j)} \\ &\geq \sum_{\theta_1, \ldots, \theta_M} \prod_{j=1}^{M} q(\theta_j) \log P(\{D_j, \theta_j\}_{j=1}^{M}|\alpha_0, G_0) - \sum_{\theta_1, \ldots, \theta_M} \prod_{j=1}^{M} q(\theta_j) \log \prod_{j=1}^{M} q(\theta_j) \\ &= \mathbb{E}_q\big[\log P(\{D_j, \theta_j\}_{j=1}^{M}|\alpha_0, G_0)\big] - \mathbb{E}_q\big[\log \prod_{j=1}^{M} q(\theta_j)\big] \end{aligned} \tag{4.19}$$

It is clear that the larger the lower bound is, the smaller the KL divergence is. Thus the posterior inference is now converted to maximize the lower bound with respect to the variational parameters. Many optimization approaches can be considered to solve the problem, e.g. the coordinate ascent approach mentioned in Section 3.2.4. In particular, the coordinate ascent algorithm optimizes each variational variable $\omega_{j,k}$ given all the others at one iteration. Note, that the constraint, i.e., $\sum_k \omega_{j,k} = 1$ for $j = \{1, \dots, M\}$, should be satisfied. The coordinate ascent algorithm yields the following equation:

$$\begin{aligned}
0 &= \frac{\partial}{\partial \omega_{j,k}} \mathbb{E}_q\big[\log P(\{D_j, \theta_j\}_{j=1}^M | \alpha_0, G_0)\big] - \mathbb{E}_q\big[\log \prod_{j=1}^M q(\theta_j)\big] + \lambda[\sum_{k=1}^M \omega_{j,k} - 1] \\
&= \sum_{j=1}^M \mathbb{E}_q[\log P(D_j|\theta_j)] + \sum_{j=1}^M \mathbb{E}_q[\log P(\theta_j|\{\theta_{j'}\}_{j' \neq j}, \alpha_0, G_0)] \\
&- \sum_{j=1}^M \mathbb{E}_q[\log q(\theta_j)] + \lambda[\sum_{k=1}^M \omega_{j,k} - 1]. \qquad (4.20)
\end{aligned}$$

Where the additional term $\lambda[\sum_k \omega_{j,k} - 1]$ is the Lagrange multiplier λ with the constraint $\sum_k \omega_{j,k} = 1$. Since the variational distributions $q(\theta_j)$ are discrete distributions at points $\{\theta_1^{ML}, \dots, \theta_M^{ML}\}$, the expectations in Equation 4.20 are sum over the M points. We have:

$$\begin{aligned}
\mathbb{E}_q[\log P(\theta_j|\{\theta_{j'}\}_{j' \neq j}, \alpha_0, G_0)] &= \sum_{k=1}^M \omega_{j,k} \log P(\theta_j = \theta_k^{ML}|\{\theta_{j'}\}_{j' \neq j}, \alpha_0, G_0) \\
\mathbb{E}_q[\log P(D_j|\theta_j)] &= \sum_{k=1}^M \omega_{j,k} \log P(D_j|\theta_k^{ML})] \\
\mathbb{E}_q[\log q(\theta_j)] &= \sum_{k=1}^M \omega_{j,k} \log \omega_{j,k}. \qquad (4.21)
\end{aligned}$$

Now we compute Equation 4.20 and obtain the updating expression for the variational parameter $\omega_{j,k}$:

$$\omega_{j,k} \propto P(D_j|\theta_k^{ML}) P(\theta_j = \theta_k^{ML}|\{\theta_{j'}\}_{j' \neq j}, \alpha_0, G_0), \qquad (4.22)$$

where

$$P(\theta_j = \theta_k^{ML}|\{\theta_{j'}\}_{j' \neq j}, \alpha_0, G_0) = \frac{\alpha_0}{\alpha_0 + M - 1} G_0 + \sum_{j' \neq j} \frac{1}{\alpha_0 + M - 1} \delta_{\theta_{j'}}. \qquad (4.23)$$

Note again, that the assumed variational distributions are discrete, $\theta_{j'}$ can only takes the values $\{\theta_k^{ML}\}_{k=1}^M$, thus:

$$P(\theta_j|\{\theta_{j'}\}_{j' \neq j}, \alpha_0, G_0) = \frac{\alpha_0}{\alpha_0 + M - 1} G_0 + \sum_{k=1}^M \frac{\omega_k}{\alpha_0 + M - 1} \delta_{\theta_k^{ML}}, \qquad (4.24)$$

where $\omega_k = \sum_{j' \neq j} \omega_{j',k}$.

In summary, the variational-inference coordinate ascent algorithm yields the updating steps as follows:

1. For each data set D_j, compute θ_j^{ML} and $P(D_j|\theta_k^{ML})$ for $j, k = \{1, \ldots, M\}$.
2. Initialize $\omega_{j,k}^{(0)}$ with constraints $\sum_k \omega_{j,k} = 1$.
3. Iterate the following steps for $t = 1, 2, \ldots$.
   - Update $\omega_{j,k}^{(t)}$ given $\{\omega_{1,:}^{(t)}, \ldots, \omega_{j-1,:}^{(t)}, \omega_{j+1,:}^{(t-1)}, \ldots, \omega_{M,:}^{(t-1)}\}$.

     $$\omega_{j,k}^{(t)} \propto P(D_j|\theta_k^{ML}) \left(\frac{\alpha_0}{\alpha_0 + M - 1} G_0 + \sum_{k=1}^{M} \frac{\omega_k}{\alpha_0 + M - 1} \delta_{\theta_k^{ML}} \right), \tag{4.25}$$

     where $\omega_k = \omega_{1,k}^{(t)} + \ldots + \omega_{j-1,k}^{(t)} + \omega_{j+1,k}^{(t-1)} + \ldots + \omega_{M,k}^{(t-1)}$. Note, that the computation might be implemented cheaply, since many terms, such as $P(D_j|\theta_k^{ML})$ for each j and k, does not change in iterations.
   - $k \leftarrow k + 1$, go to update the next $\omega_{j,k}^{(t)}$. When all variational parameters about θ_j are updated, $j \leftarrow j + 1$, and go to update for the next θ_j.
4. Stop until a stationary point reaches.

In practical computation, we need to choose appropriate starting values for the variational parameters, since poor initialization points may lead to local extreme that yields poor bound. To solve the problem, we can run the method several times and choose the final result with the best bound.

4.5 Predictive Inference

In the section, we introduce how to compute the predictive distribution $P(y_{j,new}|D, \alpha_0, G_0)$ for a new observation $y_{j,new}$ based on the results of the two inference methods.

In Gibbs sampling framework, unknown parameters θ_j's are drawn for $j = \{1, \ldots, M\}$ at each iteration. When the MCMC sequence converges, the predictive distribution is approximated. In particular, the first w members of the sequence are discarded as *burn-in* period, the predictive distribution is computed as the average over the last W members of the sequence:

$$P(y_{j,new}|D, \alpha_0, G_0) \approx \frac{1}{W} \sum_{t=w+1}^{W+w} P(y_{j,new}|\theta_j^{(t)}) \tag{4.26}$$

In variational inference framework, we obtain optimized variational parameters when the updating process reaches a stationary point. The corresponding variational distribution is a close approximation to the posterior of the unobservable variables $\{\theta_1, \ldots, \theta_M\}$, over which the predictive distribution of $y_{j,new}$ is estimated as:

$$\begin{aligned} P(y_{j,new}|D, \alpha_0, G_0) &\approx \int P(y_{j,new}|\theta_j)q(\theta_j)d\theta_j \\ &= \sum_{k=1}^{M} \omega_{j,k} P(y_{j,new}|\theta_k^{ML}). \end{aligned} \tag{4.27}$$

4.6 Summary

Nonparametric Bayesian modeling extends the flexibility of the parametric Bayesian modeling by encoding the additional uncertainty about the functional forms of the distributions. It makes more sense to view the distribution of interest itself as random variable and then to learn it directly from the data, than to only learn the parameters of the distribution with specified functional form.

Dirichlet enhanced modeling embeds the hierarchical models in nonparametric Bayesian framework (known as Dirichlet process), so that the common prior of the parallel data sources can be flexible enough to truthfully represent the learned knowledge. The model not only represents one's prior knowledge via the parameters of DP, but also makes the prior (i.e. a sample distribution from DP) as complex as necessary to model the real situation. The other applications of nonparametric Bayesian models please refer to (Dey et al., 1998; Mueller & Quintana, 2004; Tresp & Yu, 2004; Tresp, 2006; Jordan, 2005).

In the next chapter, we discuss the applications of nonparametric HB model in relational learning. Relational learning plays an important role in modern data mining. It not only encodes the information in object attributes but also models the information in relations between objects, thus the learned knowledge might be too complex to be represented by a parametric model. To solve the problem, we apply nonparametric HB modeling to relational learning, such that the learned models can represent rich probability structures and parameter dependencies in complex relational domains.

Chapter 5

Dirichlet Enhanced Relational Models

5.1 Introduction

Statistical relational learning (SRL) extends traditional machine learning methods by introducing the concepts of objects, attributes and relationships. In a relational system, objects are viewed to be distinct from each other, and are linked together in a ground network via the relationships among them, rather than being independently and identically distributed. There are various leading models developed for relational learning (see Chapter 2), in which the *directed acyclic probabilistic entity relationship* (DAPER) framework (Heckerman et al., 2004) is particularly elegant in a Bayesian context, since it encourages an explicit representation of parameters and hyperparameters. A Bayesian approach is well suited for relational modeling, the reason is that parameters, instead of being global, can be personalized to objects and relations leading to a hierarchical Bayesian (HB) framework (Gelman et al., 2004).

In a HB framework, parameterization of the prior distribution obtains central importance since it must be able not only to represent ones prior belief but also be flexible enough to represent the learned posterior, which might not be of the same mathematical form as the prior. Thus it makes sense to specify the prior distribution in a flexible nonparametric form, technically as a sample from a Dirichlet process (DP). Although we can still implement our vague prior belief in form of the base distribution of DP, the learned posterior can be very rich. Due to the central importance of the Dirichlet process, the re-parameterization of a prior distribution in form of a nonparametric highly flexible representation is sometimes referred to as Dirichlet enhancement (Escobar & West, 1998), thus we name the proposed model *Dirichlet enhanced relational learning* (DERL), which is one of main contributions in the thesis. The work was published in (Xu et al., 2005).

We apply DERL model in a medical context. Objects in the domain include hospitals, patients, diagnoses and procedures. The existence of a diagnosis or a procedure is dependent on patient attributes and hospital attributes and is modeled as reference uncertainty, which is a mechanism to represent the uncertainty in the relational structure itself (see (Getoor et al., 2003) and Chapter 2). The prior distributions for the multinomial parameters describing the reference uncertainties are now Dirichlet enhanced and are learned in the nonparametric HB framework. As an important result, parameters for diagnoses and procedures depend on each other, which allows inference from diagnoses to procedures and vice versa. We investigate the task of predicting additional procedures and diagnoses based on hospital and patient attributes, the prime complaint and on previously administered procedures and diagnoses, thus emulate the process of a clinical workflow. Compared with PRM (see Chapter 2), the performance of DERL is promising.

5.2 Model Description

5.2.1 Hierarchical Bayes for Relational Models

Figure 5.1(a) shows a relational model on a medical domain, where the objects include patients and procedures. Patient.PrimeComplaint is an attribute describing the prime complaint of the patient. Procedure.Id specifies the identifier of the procedure. The relationship class Take models the fact that patients receive procedures. The uncertainty of which procedure is taken by a patient is modeled as reference uncertainty (Getoor et al., 2003), by which the relationship class Take is associated with an additional attribute Select with as many states as there are possible procedures. The relationship attribute Select follows multinomial distribution with global parameters θ conditioned on prime complaint and hyperparameters h. In this relational model, parameters and hyperparameters are explicitly modeled as global attributes. There are two important implications. First, the probability for taking a procedure is identical for all patients with the same prime complaint. Second, procedures are modeled as independent given prime complaint such that information about prescribed procedures does not influence the selection of subsequent procedures. Both implications are not realistic. Patients are truly unique, which might be obvious to the attending physician but which is impossible to be represented in a probabilistic model. Thus, given prime complaint a physician might select a personalized treatment strategy. Additionally, the procedures taken by a patient are related with each other. The prescribed procedures influence the selections of later procedures, the physician often makes decision of the next procedure based on the previous ones. A principled approach to solve these limitations is hierarchical Bayesian modeling (Section 3.2) where it is assumed that each patient should be an individual requiring individual procedure probabilities. As shown in Figure 5.1(b), in HB modeling the procedure probabilities are additional attribute of a patient i and are represented as θ_i. Naturally, we will almost never have sufficient data to estimate the individual parameters for each patient; this dilemma is solved by assuming that all parameters originate from a common prior distribution which can be learned and shared between patients: the hyperparameters are still modeled as global attributes, since they are still shared by all patients, not individual for each patient. Thus a common prior distribution can be learned and a new patient inherits an informed prior distribution biasing the model in a sensible manner.

To fix the HB relational model we now introduce the parameters. The relationship attribute Select follows multinomial distribution with personalized parameters θ_i. For the patient i, we have $P(Select = m|\theta_i) = \theta_{i,m}$, $\theta_{i,m} > 0$ and $\sum_{m=1}^{M} \theta_{i,m} = 1$. Where M

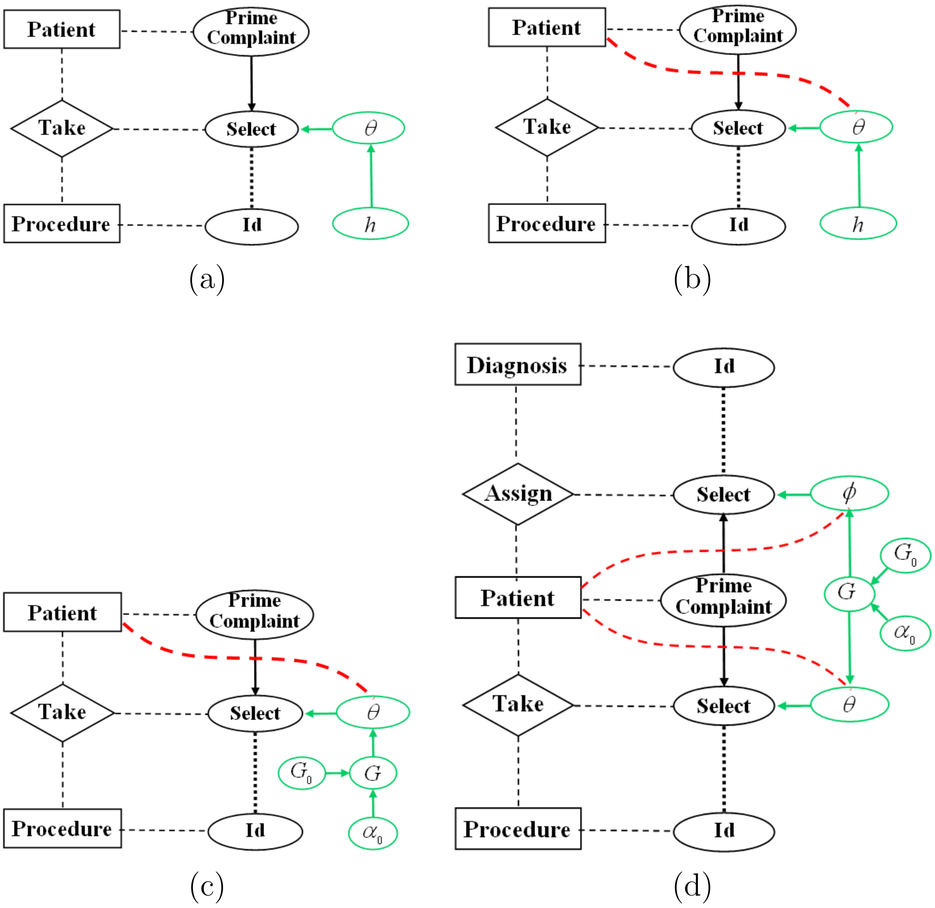


Figure 5.1: (a) A relational model explicitly incorporating the relations into the probabilistic model with reference uncertainty (see (Getoor et al., 2003) and Chapter 2). The attribute Take.Select is modeled as a multinomial variable with as many states as there are procedures. θ represents multinomial parameters for selecting procedures conditioned on Patient.PrimeComplaint. h denotes parameters of prior distribution. Note, that θ's are global parameters, which means the probability of selecting a procedures is identical for all patients with the same prime complaint. (b) Hierarchical Bayesian (HB) model, where the multinomial parameters θ_i are owned by the patient himself. (c) Nonparametric HB model. The prior G is a sample distribution from a Dirichlet process. (d) Nonparametric HB model with multi-relations.

denotes the total number of procedures, and m denotes index of a procedure. The individual parameters θ_i's share a common prior with hyperparameters h. For computational efficiency, we assume a conjugate prior, i.e. θ_i's are generated from a Dirichlet distribution with parameters $h = \{\beta_0, \beta\}$:

$$\text{Dir}(\theta_i|\beta_0, \beta) = \frac{1}{C} \prod_{m=1}^{M} \theta_{i,m}^{\beta_0 \beta_m - 1}. \tag{5.1}$$

Where C is a normalization constant given by integration over all possible θ_i. $\beta = (\beta_1, \ldots, \beta_M)$, $\beta_m > 0$, $\sum_m \beta_m = 1$, which represents our prior belief about procedure probabilities, i.e. $\mathbb{E}[P(\mathit{Select} = m|\beta_0, \beta)] = \beta_m$. $\beta_0 > 0$ is a confidence parameter. The larger the value is, the more confident our prior belief is.

In HB relational model, each patient obtains personalized procedure probabilities and shares a common parametric prior such that the two unrealistic assumptions are released. In more cases than not, the learned posterior distribution will not fall into the class of prior distributions and can not be described by $P(\cdot|h_{post})$ for any h_{post} (Chapter 4). One solution to the problem is to assume the prior distribution a very flexible nonparametric form which leads to nonparametric Bayesian framework.

5.2.2 Nonparametric Hierarchical Bayes and Dirichlet Enhancement

Figure 5.1(c) shows a nonparametric HB relational model on the medical domain. The common prior is a sample distribution drawn from a Dirichlet process:

$$G \sim \text{DP}(G_0, \alpha_0), \tag{5.2}$$

where G_0 is the base distribution, by which we can implement our vague prior belief. $\alpha_0 > 0$ is the concentration parameter specifying the degree of certainty in our prior belief. The nice feature of this approach is that, although we can still implement our vague prior belief in form of the parameters (G_0 and α_0) of the DP, the prior G can be very rich, i.e. any arbitrary distribution in the underlying sample space. For more details about Dirichlet process, please refer to Chapter 4. After sampling the prior G, the multinomial parameters θ_i for the patient i are i.i.d. drawn from G. It is difficult to drawn G directly from a DP given parameters α_0 and G_0, since the probability function space is infinite. To remove the computational constraint, we can integrate out G and directly sample the parameters θ_i via Pólya urn process (see Chapter 4):

1. For the first patient, the parameters θ_1 are drawn from the base distribution G_0.

2. Conditioned on parameters $\theta_1, \ldots, \theta_{N-1}$ of previous $N-1$ patients, the parameter θ_N for the patient N is drawn from the distribution:

$$P(\theta_N|\theta_1, \ldots, \theta_{N-1}, \alpha_0, G_0) = \frac{\alpha_0 G_0 + \sum_{i=1}^{N-1} \delta_{\theta_i}}{\alpha_0 + N - 1}, \tag{5.3}$$

where δ_{θ_i} is a distribution concentrated at a single point θ_i. Assume that in the sequence of $N-1$ parameters, there are $K \leq N-1$ distinct values $\{\theta_1^*, \dots, \theta_K^*\}$. Let N_k denote the number of times the value θ_k^* occurs in the sequence and $\sum_k N_k = N-1$. The probability of θ_N is simplified as:

$$P(\theta_N|\theta_1, \dots, \theta_{N-1}, \alpha_0, G_0) = \begin{cases} \frac{N_k}{\alpha_0+N-1} \text{ if } \theta_N = \theta_k^* \\ \frac{\alpha_0}{\alpha_0+N-1} \text{ else.} \end{cases} \tag{5.4}$$

In particular, with probability $\frac{N_k}{\alpha_0+N-1}$, the new patient takes on existing values θ_k^*; with probability $\frac{\alpha_0}{\alpha_0+N-1}$, the new patient draws new values from the distribution G_0. Note that despite the continuous nature of the base distribution G_0 (a Dirichlet distribution in the running example), a sample distribution G from a Dirichlet process is discrete in nature. After sampling θ_i for each patient, his procedures can be i.i.d. drawn from Mult$(\cdot|\theta_i)$.

5.2.3 DERL with Multi-relationships

Figure 5.1(d) shows the DERL model with multi-relations on the medical domain, where we introduce additional objects: diagnoses. A patient not only receives procedures, but also obtains diagnoses. The uncertainty of which diagnosis is assigned to the patient is also modeled as reference uncertainty, where the relationship class Assign is associated with an auxiliary relationship attribute Select with as many states as there are possible diagnoses. Again the relationship attribute Assign.Select follows multinomial distribution. The personalized attribute ϕ_i represents the parameters of distribution of Assign.Select for the patient i given his prime complaint pc. The two parameters θ_i and ϕ_i share a prior G, which is a sample distribution drawn from a DP. Note that the base distribution G_0 of the DP is a product of two independent Dirichlet distributions:

$$G_0 = \text{Dir}(\theta_i|\beta_0^{pr}, \beta^{pr}) \times \text{Dir}(\phi_i|\beta_0^{di}, \beta^{di}). \tag{5.5}$$

The Pólya urn process is extended as:

1. For the first patient, the parameters θ_1 and ϕ_1 are drawn from the base distribution G_0, i.e. $\theta_1 \sim \text{Dir}(\cdot|\beta_0^{pr}, \beta^{pr})$ and $\phi_1 \sim \text{Dir}(\cdot|\beta_0^{di}, \beta^{di})$.

2. For the second patient, the parameters θ_2 and ϕ_2 inherit the existing values θ_1 and ϕ_1 with probability $\frac{1}{\alpha_0+1}$ or draw new values with probability $\frac{\alpha_0}{\alpha_0+1}$: $\theta_2 \sim \text{Dir}(\cdot|\beta_0^{pr}, \beta^{pr})$ and $\phi_2 \sim \text{Dir}(\cdot|\beta_0^{di}, \beta^{di})$.

3. Assume that $N-1$ pairs of parameters $\{(\theta_1, \phi_1); \dots; (\theta_{N-1}, \phi_{N-1})\}$ are sampled, and there are K distinct pairs $\{(\theta_1^*, \phi_1^*); \dots; (\theta_K^*, \phi_K^*)\}$. N_k denote the number of times the pair (θ_k^*, ϕ_k^*) occurs in the sequence. Then for the patient N, the parameter pair (θ_N, ϕ_N) is assigned:

   (a) Existing values (θ_k^*, ϕ_k^*) with probability $\frac{N_k}{\alpha_0+N-1}$,

   (b) New values with probability $\frac{\alpha_0}{\alpha_0+N-1}$. The new values are drawn from the base distribution:

$$\theta_N \sim \text{Dir}(\cdot|\beta_0^{pr}, \beta^{pr}); \;\; \phi_N \sim \text{Dir}(\cdot|\beta_0^{di}, \beta^{di}).$$

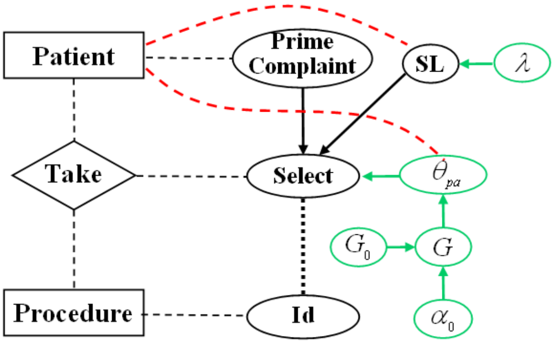


Figure 5.2: Nonparametric HB relational model with smoothing technique on the medical example. λ is the smoothing parameter. SL is an auxiliary variable, one for each patient, to smooth the probability.

From the sampling process, it is clear that the parameters for a patient are always coupled together, although our prior belief is independent distributions for the two types of relations. The DERL model does represent the probabilistic dependencies between different types of relations.

5.2.4 Smoothing

In more cases than not, the relations might be conditioned on some other attributes, thus we need to introduce multiple prior distributions, one for each configuration of the probabilistic parents. For example, in the medical example we have to specify separate G_{pc} for each configuration of Patient.PrimeComplaint. This immediately brings up the issue of overfitting, since for any particular state of Patient.PrimeComplaint, there might be only few or no data in the training data set. For example, if there is no patient with the prime complaint *circulatory* in training data, then we have $G_{circulatory} = 0$, which means the procedure probability is always zero for any new patient with prime complaint *circulatory*. It is obviously incorrect. A typical solution to deal with this problem is to smooth the probability, i.e., assigning positive value to the probability no matter whether the configuration occurs in the training data. Thus we employ linear-interpolation-smoothing method introduced in language modeling (Jelinek, 1997). The probability of selecting a procedure s conditioned on prime complaint pc is:

$$\widehat{P}(s|pc) = \lambda P(s) + (1-\lambda)P(s|pc). \tag{5.6}$$

The (conditional) probabilities $P(s)$ and $P(s|pc)$ are represented as separate prior distributions, and the probability $\widehat{P}(s|pc)$ is averaged over the two distributions with mixture weight λ. LM-smoothing can be implemented in the DERL model with an additional hidden variable LS shown as Figure 5.2. In the running example, LS follows binomial distribution with a parameter λ:

$$\begin{aligned}\widehat{P}(s|pc) &= \sum_{LS} P(LS)P(s|LS,pc)\\ &= P(LS=1)P(s|LS=1,pc) + P(LS=0)P(s|LS=0,pc)\\ &= \lambda P(s) + (1-\lambda)P(s|pc).\end{aligned}$$

5.3 Approximate Inference and Learning

Traditionally, learning in nonparametric Bayesian modeling is performed via Markov Chain Monte Carlo (MCMC) methods. The most common samplers include the Pólya urn or Chinese restaurant sampling approaches (Teh et al., 2004; Tresp & Yu, 2004). Unfortunately, these approaches are computationally involved; to remove the constraint, we focus on variational method introduced in Section 4.4.2 and extend it to relational domain.

The key inferential problem in the DERL model is to estimate the posterior distribution $\hat{G}_{pc}$ for each configuration pc of $Patient.PrimeComplaint$. Let N^{pc} denote the number of patients with prime complaint pc. $s^{pr}_{i,m}$ denotes the m'th procedure taken by patient i and M^{pr}_i denotes the total number of procedures received by the patient. Equivalently, $s^{di}_{i,\ell}$ denotes the ℓ'th diagnosis assigned to patient i and M^{di}_i denotes the total number of diagnoses assigned to the patient. The posterior $\hat{G}_{pc}$ is proportional to:

$$\mathrm{DP}(\hat{G}_{pc}|G_0,\alpha_0)\int \prod_i^{N^{pc}} \hat{G}_{pc}(\theta_i,\phi_i) \prod_m^{M_i^{pr}} P(s^{pr}_{i,m}|\theta_i) \prod_\ell^{M_i^{di}} P(s^{di}_{i,\ell}|\phi_i) d\theta_i d\phi_i. \tag{5.7}$$

Unfortunately, it is clear that the equation is analytically intractable. To solve the problem, variational inference method is considered, which target is to find a variational distribution q_{pc} to approximate the posterior distribution $\hat{G}_{pc}$ as close as possible. For computational efficiency, we assume a family of fully-factorized variational distributions

$$q_{pc}(\theta_1,\ldots,\theta_{N^{pc}},\phi_1,\ldots,\phi_{N^{pc}}) = \prod_i^{N^{pc}} q(\theta_i)q(\phi_i), \tag{5.8}$$

and for each θ_i the variational distribution is assumed as

$$q(\theta_i) = \sum_{k=1}^{N^{pc}} \omega_{i,k}\delta_{\theta_k^{MAP}}, \tag{5.9}$$

which is a discrete distribution at points θ_k^{MAP}, $k = \{1,\ldots,N^{pc}\}$. θ_i equals θ_k^{MAP} with probability $\omega_{i,k}$, i.e. $P(\theta_i = \theta_k^{MAP}) = \omega_{i,k}$ and $\sum_{k=1}^M \omega_{i,k} = 1$. θ_k^{MAP} is the *maximum a posteriori* (MAP) estimate of the parameters of patient k given his procedures,

$$\theta_k^{MAP} = \arg\max_{\theta_k} \mathrm{Dir}(\theta_k|\beta_0^{pr},\beta^{pr}) \prod_m^{M_k^{pr}} \mathrm{Mult}(s^{pr}_{k,m}|\theta_k). \tag{5.10}$$

Equivalently, for each ϕ_i the variational distribution is assumed as

$$q(\phi_i) = \sum_{k=1}^{N^{pc}} \omega_{i,k} \delta_{\phi_k^{MAP}}. \tag{5.11}$$

Note, that the variational parameters $\omega_{i,k}$ for ϕ_i are the same as θ_i, since θ_i and ϕ_i are always coupled together.

Now the inference problem is transferred to an optimization problem, i.e. we need to minimize the deference between the variational distribution q_{pc} and the posterior $\hat{G}_{pc}$ with respect to the variational parameters $\omega_{i,k}$, $i, k = \{1, \ldots, N^{pc}\}$. Extending the variational-inference coordinate ascent algorithm in Section 4.4.2 to relational data, we obtain the updating steps as follows:

1. For each patient i, compute θ_i^{MAP} and ϕ_i^{MAP} as Equation 5.10. And compute

$$P(D_i^{pr}|\theta_k^{MAP}) = \prod_{m}^{M_i^{pr}} P(s_{i,m}^{pr}|\theta_k^{MAP})$$
$$P(D_i^{di}|\phi_k^{MAP}) = \prod_{\ell}^{M_i^{di}} P(s_{i,\ell}^{di}|\phi_k^{MAP}),$$

   where D_i^{pr} and D_i^{di} denote the procedures and diagnoses taken by the patient i. $k = \{1, \ldots, N^{pc}\}$.

2. Initialize $\omega_{i,k}^{(0)}$ with constraints $\sum_k \omega_{i,k}^{(0)} = 1$. In practice, we can choose $\omega_{i,k}^{(0)} = \frac{1}{N^{pc}}$. If it leads to local extreme, we can run the algorithm several times with random initialization and choose the best result.

3. Iterate the following steps for $t = 1, 2, \ldots$.

   - Update $\omega_{i,k}^{(t)}$ given $\{\omega_{1,:}^{(t)}, \ldots, \omega_{j-1,:}^{(t)}, \omega_{j+1,:}^{(t-1)}, \ldots, \omega_{N,:}^{(t-1)}\}$:

$$\omega_{i,k}^{(t)} \propto P(D_i^{pr}|\theta_k^{MAP}) P(D_i^{di}|\phi_k^{MAP}) \times$$
$$\left\{ \frac{\alpha_0}{\alpha_0 + N^{pc} - 1} G_0 + \sum_{k=1}^{M} \frac{\omega_k^{(t)}}{\alpha_0 + N^{pc} - 1} \delta_{\theta_k^{MAP}} \delta_{\phi_k^{MAP}} \right\}, \tag{5.12}$$

     where $\omega_k^{(t)} = \omega_{1,k}^{(t)} + \ldots + \omega_{i-1,k}^{(t)} + \omega_{i+1,k}^{(t-1)} + \ldots + \omega_{N^{pc},k}^{(t-1)}$. Note, that the computation of Equation 5.12 might be implemented cheaply, since many terms, such as $P(D_i^{pr}|\theta_k^{MAP})$ and $P(D_i^{di}|\phi_k^{MAP})$, do not change in iterations.

   - $k \leftarrow k + 1$, go to update the next $\omega_{i,k}^{(t)}$. When all variational parameters about patient i are updated, go to update for the next patient $i \leftarrow i + 1$.

4. Stop until a stationary point reaches.

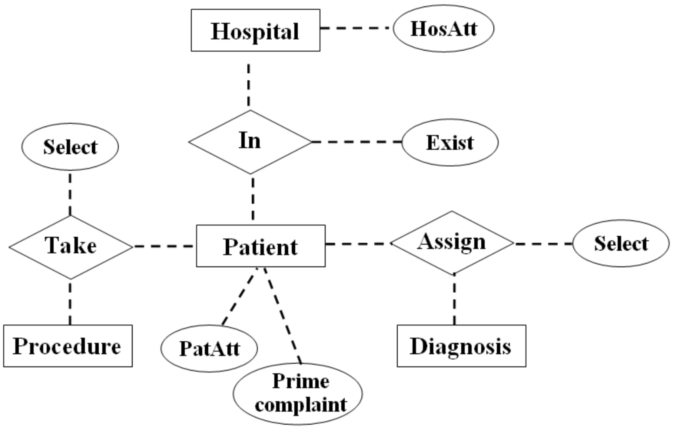


Figure 5.3: The structure of a medical data base represented by entity relational model.

After convergence, the posterior distribution assumes the form

$$\hat{G}_{pc}(\theta_i, \phi_i) = \frac{\alpha_0}{\alpha_0 + N^{pc} - 1} G_0 + \sum_{k=1}^{N^{pc}} \frac{\omega_k}{\alpha_0 + N^{pc} - 1} \delta_{\theta_k^{MAP}} \delta_{\phi_k^{MAP}}. \tag{5.13}$$

With $\alpha_0 \to \infty$ the posterior corresponds to the uninformed prior. With a finite α_0 we obtain a nonparametric hierarchical Bayesian solution.

5.4 Experimental Analysis

5.4.1 Clinical Data Description

Clinical decision support system is a branch of medical informatics, which is the intersection of research in machine learning, data mining and clinical science. The system is designed to assist the physicians in making diagnoses or delivering clinical care. In the experimental analysis, we apply DERL model in the clinical decision support system to provide the physicians case-specific suggestions. In particular, DERL model is used to predict additional procedures and diagnoses for patients based on hospital and patient attributes, the prime complaints and on previously administered procedures and diagnoses, thus the clinical workflow is emulated.

The medical domain is shown as Figure 5.3 with entity-relationship model, which is a commonly used representation for the structure of a database (Ullman & Widom, 1997). The domain includes four entity classes (Hospital, Patient, Diagnosis and Procedure) and three relationship classes (In: patient being in a hospital, Assign: patient assigning a diagnosis and Take: patient taking a procedure). A patient i is in exactly one hospital and typically has multiple procedures and diagnoses. Hospital class has attribute classes such as hospital bedsize, teaching status (teaching/nonteaching), hospital location (urban/rural), etc. Patient class has attribute classes including gender, age, admission source, etc. To reduce complexity of the Figure 5.3, hospital and patient attributes are grouped together as HosAtt and PatAtt respectively (these attributes are not aggregated in learning and inference). In addition, Patient class has the attribute class PrimeComplaint, which states the prime complaint of the patient at the time of admission. The procedures are codes in ICD-9-CM system, for example, *07.42* means *division of nerves to adrenal glands*. We use data from 9980 patients for training and 4082 patients for testing. In the data, there are 703 diagnoses and 367 procedures.

5.4.2 Experiment Result

The DERL model on the clinical data is shown as Figure 5.4(a). For both the hospital attributes and patient attributes we learned multinomial mixture models using hidden mixture attributes Hospital.Z^{ho} and Patient.Z^{pa}. The system was optimized to have 60 patient clusters and 3 hospital clusters. Both the relations between patients and procedures and the relations between patients and diagnoses are modeled as reference uncertainty. Thus the two relationship classes have additional attributes $Select^{pr}$ and $Select^{di}$, respectively. The values of $Select^{pr}$ and $Select^{di}$ indicate which procedure, resp. diagnosis is given by the physician to the patient. $Select^{pr}$ and $Select^{di}$ follow multinomial distributions with parameters θ_i and ϕ_i, which are individual for each patient. The two parameters share a prior $G_{pc,Z^{ho}}$ [1], which is a sample from a Dirichlet process. Note that the base distribution G_0 of the Dirichlet process is a product of two independent Dirichlet distributions as Equation 5.5. In the experiments we assume that

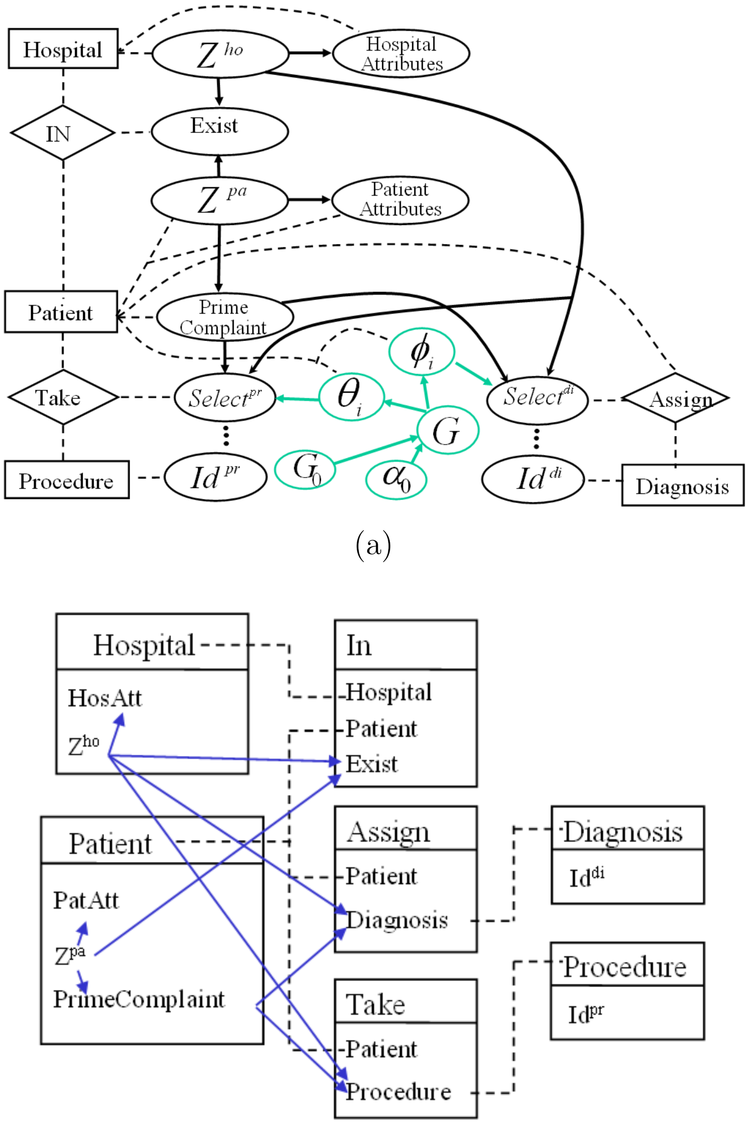
$$\beta^{pr} = (\frac{1}{M^{pr}}, \frac{1}{M^{pr}}, \ldots, \frac{1}{M^{pr}});$$
$$\beta^{di} = (\frac{1}{M^{di}}, \frac{1}{M^{di}}, \ldots, \frac{1}{M^{di}}).$$

Where M^{pr} and M^{di} denote the number of procedures and diagnoses, respectively (i.e. 367 and 703 in the case). The base distribution states unbiased priors, i.e. we believe that each procedure, resp. diagnosis has the same probability before the arrival of the data. It also specifies a priori, procedures and diagnoses are modeled as independent. However a posterior learned by the Dirichlet enhanced model is able to represent *dependencies* between procedures and diagnoses. The confidence parameters β_0^{pr} and β_0^{di} for G_0 are optimized via v-folder cross-validation method. Since the relations are dependent on *Patient.PrimeComplaint* and *Hospital.*Z^{ho}, we implement separate prior distribution for each configuration of the parents. As mentioned in Section 5.2.4, it will bring up the issue of overfitting. To remove the constraint we employ linear-interpolation-smoothing technique. In this case, it yields:

$$\hat{P}(s^{pr}|Z^{ho}, pc) = \lambda_0 P(s^{pr}) + \lambda_1 P(s^{pr}|Z^{ho}) + \lambda_2 P(s^{pr}|pc) + \lambda_3 P(s^{pr}|Z^{ho}, pc)$$

and a corresponding expression for diagnosis selections s^{di}. The weights $\lambda_\ell > 0$, $\sum_\ell \lambda_\ell = 1$ can be estimated using EM algorithm. We did not show the smoothing variables in Figure 5.4(a) due to the readability of the figure.

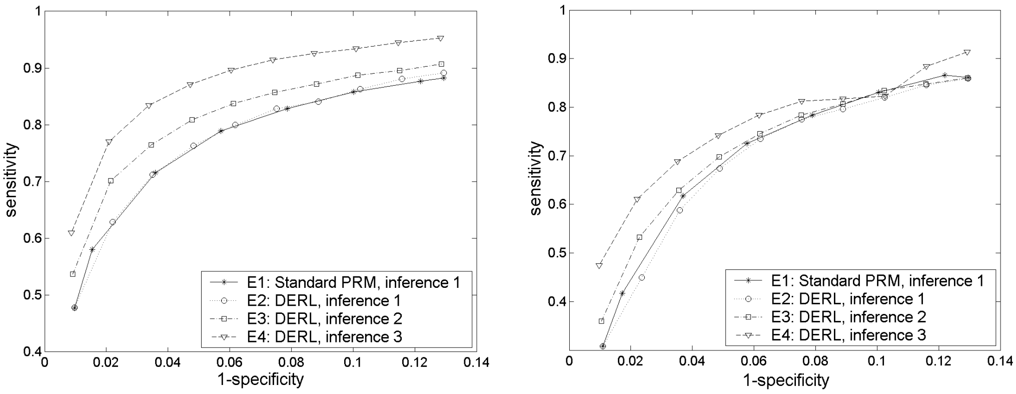
[1] Model selection showed that we obtain a better predictive model by using prime complaint as a parent and not Z^{pa}.



(a)

(b)

Figure 5.4: (a) DERL model for the medical application, where the model parameters θ_i and ϕ_i are owned by each patient himself. (b) PRM model for the same application, where the model parameters are global.



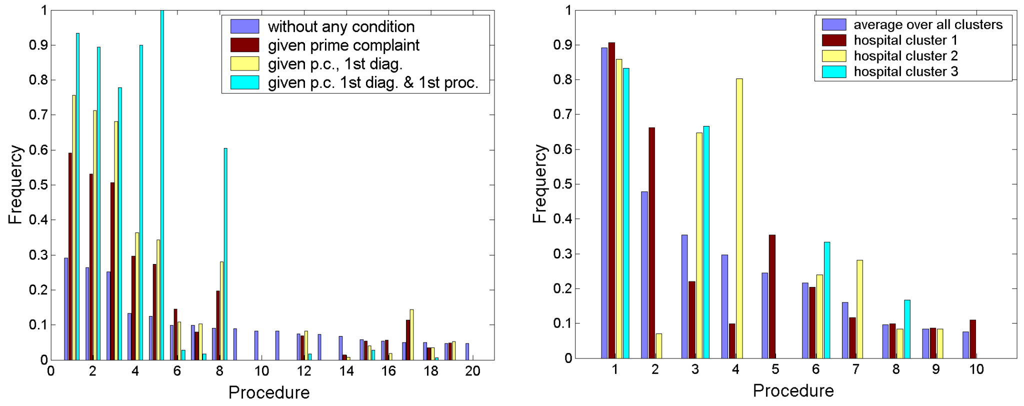
(a) (b)

Figure 5.5: (a) ROC curves for predicting procedures, given prime complaint and patient and hospital attributes. The plots are average over all test patients. (b) ROC curves for predicting procedures given prime complaint *respiratory problem* and patient and hospital attributes.

The DERL model is compared with standard PRM, which is shown in Figure 5.4(b). For more details about PRM, please refer to e.g. Friedman et al. (1999) and Chapter 2. The difference from DERL model is that the multinomial distributions of selecting procedures (and diagnoses) are global, not individual for each patient.

We evaluate model performances by predicting the application of procedures. In the first experiment we predicted any of the procedures that a patient has received given hospital attributes, patient attributes and given prime complaint. The corresponding ROC curve (averaged over all patients) for DERL model is shown as E2 in Figure 5.5(a). In the experiment we selected the top C procedures recommended by the model. Sensitivity indicates how many percents of the actually being performed procedures were correctly proposed by the model. (1-specificity) indicates how many percents of the procedures that were not actually performed were recommended by the model. Along the curves, the C was varied from left to right as $C = 5, 10, \dots, 50$. $E1$ in Figure 5.5(a) shows the experimental result of the standard PRM model given the same information as E2. It is essentially identical to the result of E2. The situation changes when additional information is available such as past procedures or diagnoses: the standard PRM model would not change the proposal probabilities. In contrast, the prediction of a subsequent procedure is improved for DERL model if the first diagnosis is available (E3) or both the first diagnosis and the first procedure are available (E4). We can see, for example, that if we would propose 15 procedures, after we know the prime complaint, the first diagnosis, and the first procedure, we would cover approximately 83% of the actually prescribed procedures. Figures 5.5(b) shows the corresponding plots for patients with prime complaint *respiratory problem* exhibiting similar trends.

In the second set of experiments, we investigated how the procedure probabilities sequentially change when additional information becomes available. Figure 5.6(a) shows the selection probabilities for 20 procedures which are relevant for *myocardial infarction.* The top ten procedures are listed in Table 5.1. The first column indicates the predicted



(a) (b)

Figure 5.6: (a) Procedure probabilities conditioned on increasing information. (b) Procedure probabilities for different hospital clusters.

Table 5.1: The most frequent procedures for disease No. 410.71.

Rank	Code	Description
1	88.56	coronary arteriography using two catheters
2	37.22	left heart cardiac catheterization
3	88.53	angiocardiography of left heart structures
4	36.06	insertion of coronary artery stent(s)
5	36.01	single vessel percutaneous transluminal coronary angioplasty
6	99.20	injection or infusion of platelet inhibitor
7	36.15	single internal mammary-coronary artery bypass
8	39.61	extracorporeal circulation auxiliary to open heart surgery
9	88.72	diagnostic ultrasound of heart
10	99.04	transfusion of packed cells

probabilities for the case that only patient attributes and hospital attributes are available. The second column shows the procedure probabilities when, in addition, the prime complaint *circulatory problem* becomes available. The third column shows the situation when, in addition, the first diagnosis *acute myocardial infarction* becomes available. The fourth column shows the situation when, in addition, the procedure *single vessel percutaneous transluminal coronary angioplasty* has been performed. One sees that the probabilities for procedures relevant for myocardial infarction increase when prime complaint becomes available. The tendency is that if more information becomes available, the model becomes more certain about coming procedures for a patient. Figure 5.6(b) shows probabilities of selecting procedures given the diagnosis *single live-born in hospital* in deferent hospital clusters. One can see that the probabilities vary significantly. It demonstrates that hospital attributes are quite relevant for the procedure prediction. In the experiment, the hospitals are assigned to the most likely cluster based on a mixture model.

5.5 Summary

In this chapter we give some analysis how nonparametric hierarchical Bayesian modeling can be very useful in relational learning and propose a new DERL model, which is one of the major contributions of the thesis. In DERL, model parameters can be attributes of entities or relations and can thus be non-global. These individual parameters share a common nonparametric prior, technically as a sample distribution from a Dirichlet process. As an important result, the posterior learned by DERL can exhibit a rich structure and parameter dependencies which are impossible to be represented in a parametric formulation. We demonstrated the performance of DERL model using data from a medical database. The relations are explicitly incorporated into probabilistic models with reference uncertainty (Getoor et al., 2003) and DERL model is used to encode the dependencies between patients and diagnoses and patients and procedures. Despite the fact that the base distribution (prior belief) exhibits parameter independence, the learned posterior does display parameter dependencies. The couplings between diagnoses and procedures could truthfully be modeled.

Part III

Relational Learning with Infinite Mixture Models

Chapter 6

Finite Mixture Models

6.1 Introduction

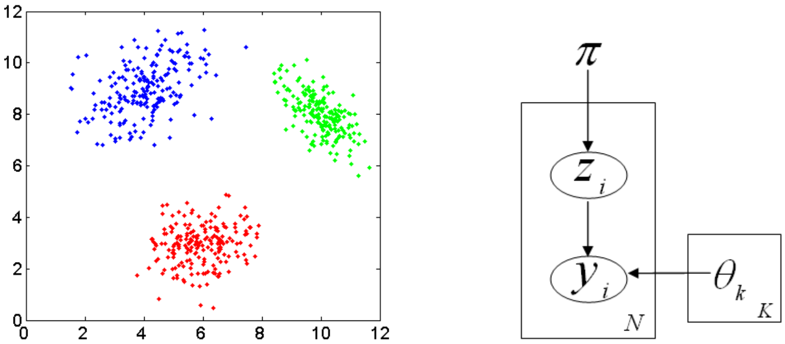
Mixture model is a very common modeling tool, which is well suited in the situations where the samples are generated under different conditions. For example, we want to make a survey about the reaction time of people when driving. It is better to divide the people into two subsets in which ones do or do not drink alcohol, then the reaction time is modeled as separate distributions conditioned on the situation of alcohol or non-alcohol, rather than building a single bimodal distribution with two different peaks. Let y denote the reaction time of a person, θ_1 and θ_2 are the distribution parameters in the two situations, respectively. π is the probability of a person drinking alcohol. Then the distribution of the reaction time is represented as

$$P(y|\pi, \theta_1, \theta_2) = \pi P(y|\theta_1) + (1-\pi)P(y|\theta_2). \tag{6.1}$$

The atom distributions $P(y|\theta_1)$ and $P(y|\theta_2)$ are referred to as *mixture components*. When the atom distribution is parameterized, we can directly refer to the parameters (θ_1 and θ_2) as mixture components. The parameter π, referred to as *mixture proportion* or *mixture weight*, specifies the proportion in which the atom distributions are mixed. The finite mixture model can be viewed as a special case of a more general specification *continuous mixture model*:

$$\begin{aligned} P(y) &= \int \pi(\theta)P(z|\theta)d\theta \\ &= \sum_{k=1}^{\infty} \pi_k P(y|\theta_k). \end{aligned} \tag{6.2}$$

In the running example about reaction time, if the atom distributions are conditioned on the extent ones drink alcohol, rather than the binary variable alcohol/non-alcohol, then we obtain a continuous mixture model. Furthermore, from the mathematical form of the density function point of view, the hierarchical Bayesian model introduced in Chapter 4 can be thought of as a variant of the continuous mixture model. The two models are however applied in different situations. The mixture model is applied when the samples



(a) (b)

Figure 6.1: (a) Samples drawn from a population equally mixing 3 Gaussian distributions. (b) The graphical representation of an empirical finite mixture model. $\Theta = \{\theta_1, \ldots, \theta_K\}$ are K mixture components, π are mixture proportions. The parameters Θ and π are unknown but not random. Each observation y_i is associated with an auxiliary variable z_i, which specifies the mixture component from which the observation y_i is generated.

belong to a single data set and are generated under different conditions. It is widely used in the clustering and classification problems. In comparison, hierarchical model is widely used in the situations where multiple parallel data sets are available and these data sets come from different but related settings. The parameters for each data set are distinct but share a common prior, by which the learned knowledge from previous data sets can be transferred to the new data sets. Hierarchical model is widely used in meta-learning.

6.2 Empirical Mixture Models

6.2.1 Model Description

Mixture models supply a method to learn the population consisting of several subpopulations within each of which a relatively simple distribution applies. Figure 6.1(a) shows samples drawn from a population equally mixing 3 Gaussian distributions. It makes more sense to build a distribution in the form of $P(y) = \frac{1}{3}\sum_{k=1}^{3} \mathrm{N}(\mu_k, \sigma_k^2)$, than to build a single distribution with three peaks. In the section we discuss mixture model in empirical Bayesian framework. Although the empirical solution is not full Bayesian, it is mathematically easier and includes the main properties of finite mixture modeling. Figure 6.1(b) shows the empirical model in a plate representation. The model consists of K components $\Theta = \{\theta_1, \ldots, \theta_K\}$, which are mixed in the proportions $\pi = (\pi_1, \ldots, \pi_K)$ and $\pi_k > 0, \sum_{k=1}^{K} \pi_k = 1$. The parameters Θ and π are *unknown, but not random.* For each observation y_i, an auxiliary variable z_i, referred to as *indicator*, is introduced, which specifies the mixture component from which the observation y_i is generated. z_i can be modeled as a discrete random variable with K states and is generally assumed a multinomial distribution with parameters π, i.e. $P(z_i = k|\pi) = \pi_k$. Assume that there are N

observations $D = \{y_1, \ldots, y_N\}$, then N hidden variables $Z = \{z_1, \ldots, z_N\}$ are associated, one for each observation. The generative process of the empirical mixture model is defined as:

$$\begin{aligned} z_i|\pi &\overset{\text{i.i.d.}}{\sim} \text{Mult}(\cdot|\pi), \\ y_i|z_i, \Theta &\overset{\text{i.i.d.}}{\sim} P(\cdot|z_i, \Theta). \end{aligned}$$

The joint probability of the model is defined as:

$$P(D, Z|\pi, \Theta) = \prod_{i}^{N} P(z_i|\pi)P(y_i|z_i, \Theta). \tag{6.3}$$

In the mixture model, the auxiliary variables Z are unobservable and can be viewed as *missing data*, the observations D are thus viewed as *incomplete-data* and the combination $\{D, Z\}$ are viewed as *complete-data*.

In practical computation, we need to choose an appropriate value K for the number of mixture components. In many cases, the value is not available in advance. The simplest solution for the problem is the v-folder cross validation method (Miloslavsky & van der Laan, 2003). An alternative solution is to represent the number K as another unknown parameter of the model and to learn it in the way like other parameters (Richardson & Green, 1997).

In real-world applications, each mixture component is often viewed as a group, the observations generated from the same component are viewed as members in the same group. The hidden variable z_i indicates the group of y_i. The observations are similar in the identical group and are dissimilar to the observations belonging to other groups. The mixture model is commonly used in the clustering and classification problems.

6.2.2 Parameters Estimation with EM Algorithm

In the empirical mixture model, the main learning problem is to estimate the unknown parameters Θ and π. Since they are unknown but not random, we can approximate their values via point estimation methods, say *maximum likelihood* estimation. Given observations $D = \{y_1, \ldots, y_N\}$, the maximum-(log)likelihood estimations of the parameters π and Θ are defined as:

$$\begin{aligned} \pi^{ML}, \Theta^{ML} &= \arg\max_{\pi,\,\Theta} \log P(D|\pi, \Theta) \\ &= \arg\max_{\pi,\,\Theta} \sum_{i}^{N} \left[\log \sum_{k=1}^{K} P(z_i = k|\pi)P(y_i|z_i = k, \Theta) \right]. \end{aligned} \tag{6.4}$$

It is obvious that the equation is analytical intractable due to the log of the sum, which is introduced by the missing data, i.e. the hidden variables $Z = \{z_1, \ldots, z_N\}$. To solve the problem, we consider elaborate techniques, e.g. *Expectation-Maximization* (EM) method.

EM method is introduced by Dempster et al. (1977), which is a principal method to find the maximum-likelihood estimation of parameters given data with missing values.

The method is mainly used in the following situations. First, there are really missing observations in the given data set. Second, the data is complete, but there are hidden variables introduced in modeling process, e.g. the auxiliary variables Z in the mixture model. The latter application is more common in the data mining area. Principally, EM method is to optimize the expected value of the complete-data log-likelihood, since the likelihood is in fact a random variable in terms of randomness of the missing values. In particular, the following two steps are repeated until convergence. In E step, the expectation of the complete-data log-likelihood is calculated given the current parameter estimations. In M step, the expectation calculated in the last step is maximized with respect to parameters, then the optimized parameters are used in the next iteration. EM method guarantees that the log-likelihood increases at each iteration and converges to a local maximum of the likelihood function. For more details about EM method, please refers to (Redner & Walker, 1984; Ghahramani & Jordan, 1994; Bishop, 1994).

For the mixture model, EM method optimizes the expectation $\mathbb{E}[\log P(D, Z|\pi, \Theta)]$ with respect to the mixture proportions π and the mixture components Θ. Given the observations $D = \{y_1, \ldots, y_N\}$, the expected complete-data log-likelihood is computed in the E step:

$$\begin{aligned}\mathcal{Q}^{(t)} &= \mathbb{E}_{P(Z|D,\pi^{(t-1)},\Theta^{(t-1)})}\left[\log P(D, Z|\pi^{(t)}, \Theta^{(t)})\right] \\ &= \sum_i \sum_k P(z_i = k|y_i, \pi^{(t-1)}, \Theta^{(t-1)}) \left[\log P(z_i = k|\pi^{(t)}) + \log P(y_i|\theta_k^{(t)})\right]. \end{aligned} \tag{6.5}$$

Where $\pi^{(t-1)}$ and $\Theta^{(t-1)}$ denote the mixture weights and mixture components computed in the last iteration. $P(z_i|y_i, \pi^{(t-1)}, \Theta^{(t-1)})$ is the posterior distribution of hidden variable z_i given observation y_i and learned parameters $\pi^{(t-1)}$ and $\Theta^{(t-1)}$. $\pi^{(t)}$ and $\Theta^{(t)}$ are unknown parameters and will be optimized in M step as:

$$\pi^{(t)*}, \Theta^{(t)*} = \arg\max_{\pi^{(t)},\, \Theta^{(t)}} \mathcal{Q}^{(t)}. \tag{6.6}$$

Note, that the constraint $\sum_k \pi_k = 1$ should be satisfied at each iteration.

Now let us use a particular example to illustrate the specification. Assume that the observations $D = \{y_1, \ldots, y_N\}$ are discrete variables with S possible states. They are generated from K multinomial distributions. θ_k are the parameters of the k'th distribution. Under the mixture modeling, the auxiliary variable z_i is introduced, one for each observation y_i. Then we use EM algorithm to estimate the parameters π and $\Theta = \{\theta_1, \ldots, \theta_K\}$. The expected complete-data log-likelihood at iteration t is written as:

$$\mathcal{Q}^{(t)} = \sum_i \sum_k P(z_i = k|y_i, \pi^{(t-1)}, \theta^{(t-1)})\left[\log \pi_k^{(t)} + \log \theta_{k,y_i}^{(t)}\right]. \tag{6.7}$$

Thus the updating steps are defined as:

1. Take some initial value for $\pi^{(0)}$ and $\Theta^{(0)}$, and alternatively run the E step and M step until convergence.

2. In the *Expectation* (E) step, compute the probability of $z_i = k$, for $i = \{1, \ldots, N\}$, $k = \{1, \ldots, K\}$, given the estimates of the parameters optimized at last step. The E step can be viewed as soft class assignment for each observation.

$$
\begin{aligned}
P(z_i = k | y_i, \pi^{(t-1)}, \Theta^{(t-1)}) &= \frac{P(z_i = k|\pi^{(t-1)})P(y_i|z_i = k, \Theta^{(t-1)})}{\sum_k P(z_i = k|\pi^{(t-1)})P(y_i|z_i = k, \Theta^{(t-1)})} \\
&= \frac{\pi_k^{(t-1)}\theta_{k,y_i}^{(t-1)}}{\sum_k \pi_k^{(t-1)}\theta_{k,y_i}^{(t-1)}}. \qquad (6.8)
\end{aligned}
$$

3. In the *Maximization* (M) step, update the estimates of the parameters to maximize the expected complete-data log-likelihood $\mathcal{Q}^{(t)}$. Note, that the constraints $\sum_k \pi_k^{(t)} = 1$ and $\sum_s \theta_{k,s}^{(t)} = 1$ should be satisfied.

$$
\begin{aligned}
0 &= \frac{\partial \mathcal{Q}^{(t)}}{\partial \pi_k^{(t)}} = \frac{\partial}{\partial \pi_k^{(t)}} \sum_i \left[P(z_i = k | y_i, \pi^{(t-1)}, \Theta^{(t-1)}) \log \pi_k^{(t)} \right] + \lambda(\sum_k \pi_k^{(t)} - 1) \\
&\Rightarrow \pi_k^{(t)} \propto \sum_i P(z_i = k | y_i, \pi^{(t-1)}, \Theta^{(t-1)}). \qquad (6.9)
\end{aligned}
$$

Where the term $\lambda(\sum_k \pi_k^{(t)} - 1)$ is the Lagrange multiplier λ with the constraint $\sum_k \pi_k^{(t)} = 1$.

$$
\begin{aligned}
0 &= \frac{\partial \mathcal{Q}^{(t)}}{\partial \theta_{k,s}^{(t)}} = \frac{\partial}{\partial \theta_{k,s}^{(t)}} \sum_i \left[P(z_i = k | y_i, \pi^{(t-1)}, \Theta^{(t-1)}) \log \theta_{k,s}^{(t)} \delta_s(y_i) \right] + \lambda(\sum_s \theta_{k,s}^{(t)} - 1) \\
&\Rightarrow \theta_{k,s}^{(t)} \propto \sum_i P(z_i = k | y_i, \pi^{(t-1)}, \theta^{(t-1)}) \delta_s(y_i). \qquad (6.10)
\end{aligned}
$$

Where $\delta_s(y_i)$ equals to 1 if the observation y_i takes the state s, and 0 otherwise.

6.2.3 Predictive Inference

In mixture model, there are mainly two prediction tasks, one is to predict a new observation $P(y_{new}|\pi, \Theta)$, the other is to predict the hidden variable given the new observation $P(z_{new} = k|y_{new}, \pi, \Theta)$. The two predictive inferences are performed as:

$$
\begin{aligned}
P(y_{new}|\pi, \Theta) &= \sum_k P(z_{new} = k|\pi) P(y_{new}|z_{new} = k, \Theta) \\
&= \sum_k \pi_k P(y_{new}|\theta_k) \qquad (6.11a) \\
P(z_{new} = k|y_{new}, \pi, \Theta) &= \frac{\pi_k P(y_{new}|\theta_k)}{\sum_k \pi_k P(y_{new}|\theta_k)}. \qquad (6.11b)
\end{aligned}
$$

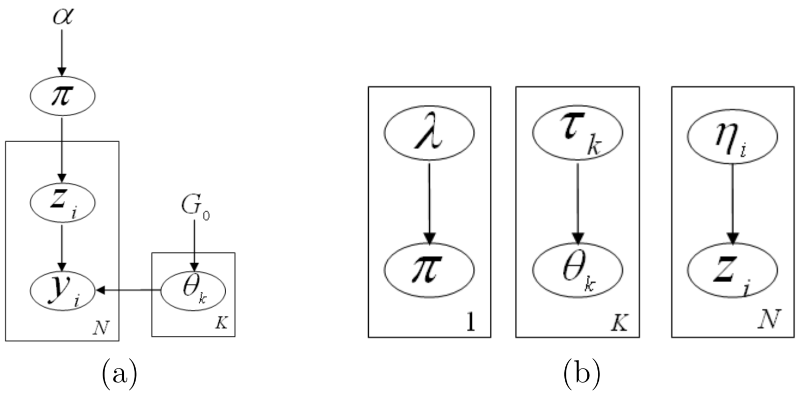


Figure 6.2: (a) A finite mixture model in full Bayesian framework. (b) The graphic representation of the variational distribution assumed to approximate the posterior distribution of the parameters. λ, τ_k and η_i are variational parameters.

6.3 Mixture Models in Full Bayesian Framework

6.3.1 Model Description

In empirical mixture model, the uncertainty in estimating the unknown parameters π and $\Theta = \{\theta_1, \dots, \theta_K\}$ is not considered. To remove the limitation, we embed the finite mixture model in the full Bayesian framework, i.e. the unknown parameters themselves are viewed as random variables. The mixture proportions π are multinomial parameters, for computational efficiency, we assume a conjugate Dirichlet prior with hyperparameters $\alpha = (\alpha_1, \dots, \alpha_K)$, i.e. $\pi \sim \mathrm{Dir}(\cdot|\alpha)$. The mixture component θ_k denote the parameters of the distribution of observations with hidden state k. All θ_k's share a common prior G_0. Given the priors $\mathrm{Dir}(\pi|\alpha)$ and G_0, the generative process of full-Bayesian mixture model is defined as follows:

$$\begin{aligned} \pi|\alpha &\sim \mathrm{Dir}(\cdot|\alpha) \\ \theta_k|G_0 &\sim G_0(\cdot),\ k = 1, \dots, K \\ z_i|\pi &\sim \mathrm{Mult}(\cdot|\pi),\ i = 1, \dots, N \\ y_i|z_i, \Theta &\sim P(\cdot|z_i, \Theta),\ i = 1, \dots, N \end{aligned}$$

The graphical representation of the full Bayesian model is shown as Figure 6.2(a), which has one more level than the empirical model in Figure 6.1(b). The additional level represents the uncertainty about the unknown parameters Θ and π. The joint probability of the full Bayesian model is:

$$P(D, Z, \pi, \Theta|\alpha, G_0) = P(\pi|\alpha) \prod_{k=1}^{K} P(\theta_k|G_0) \prod_{i=1}^{N} P(z_i|\pi) P(y_i|z_i, \Theta). \tag{6.12}$$

6.3.2 Inference

The key inferential problem in the full Bayesian mixture model is to compute the joint posterior distribution of the unobservable variables given observations $D = \{y_1, \ldots, y_N\}$. As discussed in last section, the unobservable variables in the model include: mixture proportions π, mixture components $\Theta = \{\theta_1, \ldots, \theta_K\}$ and the indicators $Z = \{z_1, \ldots, z_N\}$. Thus the joint posterior distribution is defined as:

$$\begin{aligned} P(\pi, \Theta, Z | D, \alpha, G_0) &= \frac{P(\pi, \theta, Z, D|\alpha, G_0)}{P(D|\alpha, G_0)} \\ &= \frac{P(\pi|\alpha) \prod_k P(\theta_k|G_0) \prod_i P(z_i|\pi) P(y_i|z_i, \Theta)}{P(D|\alpha, G_0)}, \end{aligned} \tag{6.13}$$

where the normalization term $P(D|\alpha, G_0)$ is:

$$\int P(\pi|\alpha) \prod_k \int P(\theta_k|G_0) \prod_i \sum_k P(z_i = k|\pi) P(y_i|\theta_k) d\pi d\theta_1 \ldots d\theta_K. \tag{6.14}$$

Unfortunately, it is clear that the posterior distribution is analytically intractable due to the coupling between π and Θ in the summation over the hidden variables. Although the exact inference is impossible, many approximate inference algorithms can be considered, e.g. Gibbs sampling and variational approximation.

Inference with Gibbs Sampling

Gibbs sampling (GS) is the simplest Markov chain Monte Carlo method, which obtains a Markov chain via iteratively sampling each unknown variables conditioned on the data and the previous samples of all other unknown variables. For more details about GS method, please refer to Section 3.2.5. In the full Bayesian mixture model, if the prior distribution G_0 is assumed to be of manageable form, Gibbs sampling is straightforward. In particular, the following three steps are repeated until convergence.

- In the first step, the indicator variable z_i for each observation is sampled with the probability

  $$\begin{aligned} P(z_i^{(t)} = k|y_i, \pi^{(t-1)}, \theta^{(t-1)}) &\propto P(z_i^{(t)} = k|\pi^{(t-1)}) P(y_i|\theta_k^{(t-1)}) \\ &= \pi_k^{(t-1)} P(y_i|\theta_k^{(t-1)}). \end{aligned} \tag{6.15}$$

  Since the number of hidden states is finite, the computation can be cheaply implemented.

- In the second step, the mixture proportions π are sampled from:

  $$\begin{aligned} \pi^{(t)} &\sim P(\cdot|Z^{(t)}, \alpha) = \mathrm{Dir}(\cdot|\alpha_{post}^{(t)}) \\ \alpha_{post}^{(t)} &= (\alpha_1 + N^{(t)}(1), \ldots, \alpha_K + N^{(t)}(K)). \end{aligned} \tag{6.16}$$

  Where $N^{(t)}(k)$ is a *sufficient statistic* about the data, which denotes the number of observations with hidden state k at the iteration t.

- At last, the mixture components $\Theta = \{\theta_1, \ldots, \theta_K\}$ are drawn from:

$$\theta_k^{(t)} \sim P(\cdot|D_k^{(t)}, G_0). \tag{6.17}$$

Where $P(\cdot|D_k^{(t)}, G_0)$ is the posterior of mixture component θ_k at iteration t. $D_k^{(t)}$ denotes the observations with hidden state k at iteration t. It is clear that the computation about the posterior $P(\theta_k^{(t)}|D_k^{(t)}, G_0)$ is tractable if $P(D_k^{(t)}|\theta_k^{(t)})$ and $G_0(\theta_k^{(t)})$ are assumed to be of manageable form. For example, we assume that $P(D_k^{(t)}|\theta_k^{(t)})$ is a distribution in the exponential family and $G_0(\theta_k^{(t)})$ is conjugated with $P(D_k^{(t)}|\theta_k^{(t)})$.

Inference with Variational Method

MCMC algorithms supply a successful solution to approximate the posterior distributions, however it is computationally quite involved. To remove the constraint, many alternative solutions are introduced, e.g. variational inference algorithms. The principle of these algorithms is to find a variational distribution $q(\xi)$ to approximate the distribution of interest $P(\xi)$, where ξ denotes a set of variables. The difference between $q(\xi)$ and $P(\xi)$ is measured via *Kullback-Leibler* (KL) divergence. The smaller the divergence is, the more approximate the two distributions are. For more information about variational methods, please refer to Section 4.4.2. For the full Bayesian mixture model, we focus on the simplest variational method, *mean-field* approximation, in which $q(X)$ is restricted to a family of fully-factorized distributions for computational efficiency (Jordan et al., 1998).

In the full Bayesian mixture model, the distribution of interest is the joint posterior distribution of the unobservable variables, $P(Z, \pi, \Theta|D, \alpha, G_0)$. Let $q(Z, \pi, \Theta)$ denote the variational distribution. The KL divergence between them is defined as:

$$\begin{aligned}
&\sum_{Z,\,\pi,\,\Theta} q(Z, \pi, \Theta) \log q(Z, \pi, \Theta) - \sum_{Z,\,\pi,\,\Theta} q(Z, \pi, \Theta) P(Z, \pi, \Theta|D, \alpha, G_0) \\
&= \mathbb{E}_q\big[\log q(Z, \Theta, \pi)\big] - \mathbb{E}_q\big[\log P(D, Z, \Theta, \pi|\alpha, G_0)\big] + \log P(D|\alpha, G_0).
\end{aligned} \tag{6.18}$$

We permute the equation and obtain:

$$\begin{aligned}
&\log P(D|\alpha, G_0) \\
&= \mathbb{E}_q\left[\log P(D, Z, \Theta, \pi|\alpha, G_0)\right] - \mathbb{E}_q\left[\log q(Z, \Theta, \pi)\right] + KL(q||P) \\
&\geq \mathbb{E}_q\left[\log P(D, Z, \Theta, \pi|\alpha, G_0)\right] - \mathbb{E}_q\left[\log q(Z, \Theta, \pi)\right].
\end{aligned} \tag{6.19}$$

Equation 6.19 defines a *lower bound* of the log-likelihood of the observations. It can also be derived via the *Jensen's inequality*:

$$
\begin{aligned}
&\log P(D|\alpha, G_0) \\
&= \log \sum_{Z,\,\pi,\,\Theta} P(D, Z, \Theta, \pi|\alpha, G_0) \\
&= \log \sum_{Z,\,\pi,\,\Theta} \frac{q(Z, \Theta, \pi) P(D, Z, \Theta, \pi|\alpha, G_0)}{q(Z, \Theta, \pi)} \\
&\geq \sum_{Z,\,\pi,\,\Theta} q(Z, \Theta, \pi) \log P(D, Z, \Theta, \pi|\alpha, G_0) - \sum_{Z,\,\pi,\,\Theta} q(Z, \Theta, \pi) \log q(Z, \Theta, \pi) \\
&= \mathbb{E}_q \left[\log P(D, Z, \Theta, \pi|\alpha, G_0)\right] - \mathbb{E}_q \left[\log q(Z, \Theta, \pi)\right]. \qquad (6.20)
\end{aligned}
$$

It is clear that the larger the lower bound is, the smaller the KL divergence is. Thus the posterior inference problem is now converted to an optimization problem, i.e. to maximize the lower bound with respect to the variational parameters. For computational convenience, we select a fully-factorized family of variational distributions:

$$q(Z, \Theta, \pi) = q(\pi|\lambda) \prod_{k}^{K} q(\theta_k|\tau_k) \prod_{i}^{N} q(z_i|\eta_i). \qquad (6.21)$$

λ, τ_k and η_i are variational parameters. Note, that there is one τ_k for each mixture component and one η_i for each observation. $q(\pi|\lambda)$ is a Dirichlet distribution, $q(\theta_k|\tau_k)$ is of the same mathematic form as G_0, $q(z_i|\eta_i)$ is a multinomial distribution. The variational distributions decouple some probabilistic dependencies, e.g. the mixture components θ_k's no longer share a common prior, which is shown in Figure 6.2(b) with a plate representation. Given the fully-factorized variational distributions, the lower bound $\mathcal{L}$ of the log-likelihood of observations, i.e. Equation 6.19, is now written as:

$$
\begin{aligned}
\mathcal{L} = \mathbb{E}_q \left[\log P(\pi|\alpha)\right] + \sum_{k}^{K} \mathbb{E}_q \left[\log P(\theta_k|G_0)\right] + \sum_{i}^{N} \mathbb{E}_q \left[\log P(z_i|\pi)\right] \\
+ \sum_{i}^{N} \mathbb{E}_q \left[\log P(y_i|z_i, \Theta)\right] - \mathbb{E}_q \left[\log q(Z, \Theta, \pi)\right]. \qquad (6.22)
\end{aligned}
$$

Many optimization approaches can be considered to maximize the equation, e.g. the coordinate ascent approach mentioned in Section 3.2.4. In particular, the coordinate ascent algorithm optimizes each variational variable λ, τ_k and η_i given all the others at one iteration. Note that the constraints, $\sum_k \eta_{i,k} = 1$ for $i = \{1, \ldots, N\}$, should be satisfied.

We now illustrate the specification with an example. Assume that the observations are discrete variables with S states and are generated from K multinomial distributions. The prior distribution G_0 is conjugate Dirichlet distribution with hyperparameters $\beta = \beta_0(\frac{1}{S}, \ldots, \frac{1}{S})$, which represents our prior belief that each observation state appears with equal probability. The other hyperparameters α are assumed as $\alpha = \alpha_0(\frac{1}{K}, \ldots, \frac{1}{K})$, which represents our prior belief that each hidden state appears with equal probability. α_0 and β_0 indicate how strongly we believe that the prior distributions should be true. The larger the values are, the stronger our belief is. Of course, the assumptions do not mean the following algorithms are only applicable to the discrete variables, they are easily extended to other situations.

Following we describe the computation of Equation 6.22. Let us start from the first term.

$$\begin{aligned}\mathbb{E}_{q(\pi|\lambda)} \log P(\pi|\alpha) &= \mathbb{E}_q \left[\log\Gamma(\alpha_0) - \sum_k \log\Gamma(\alpha_k) + \sum_k (\alpha_k - 1)\log\pi_k \right] \\ &= \log\Gamma(\alpha_0) - \sum_k \log\Gamma(\alpha_k) + \sum_k (\alpha_k - 1)\mathbb{E}_q[\log\pi_k]. \qquad (6.23)\end{aligned}$$

The computation of $\mathbb{E}_q[\log\pi_k]$ exploits the property of Dirichlet distribution as a member of exponential family: the first derivative of normalization factor is the expectation of sufficient statistic of the distribution. As an exponential family of distribution, the variational distribution $q(\pi|\lambda)$ can be written as:

$$q(\pi|\lambda) = \exp\left[\sum_k (\lambda_k - 1)\log\pi_k + \log\Gamma(\sum_k \lambda_k) - \sum_k \log\Gamma(\lambda_k) \right], \qquad (6.24)$$

and we have:

$$\begin{aligned}&\text{Lebesgue-Stieltjes integrator:} && H(\pi) = 1, \\ &\text{Natural parameter:} && \zeta^T = \lambda - 1, \\ &\text{Sufficient statistic:} && T(\pi) = \log\pi^T = (\log\pi_1, \ldots, \log\pi_K)^T, \\ &\text{Normalization factor:} && A(\zeta) = \sum_k \log\Gamma(\lambda_k) - \log\Gamma(\sum_k \lambda_k) \\ &&& = \sum_k \log\Gamma(\zeta_k + 1) - \log\Gamma(\sum_k \zeta_k + K)\end{aligned}$$

Thus the first derivative of normalization factor is computed as:

$$\begin{aligned}\frac{\partial A}{\partial\zeta_k} &= \frac{\partial}{\partial\zeta_k}\left[\sum_k \log\Gamma(\zeta_k + 1) - \log\Gamma(\sum_k \zeta_k + K) \right] \\ &= \frac{\partial}{\partial\zeta_k}\left[\log\Gamma(\zeta_k + 1) - \log\Gamma(\sum_k \zeta_k + K) \right] \\ &= \Psi(\zeta_k + 1) - \Psi(\sum_{k'} \zeta_{k'} + K) \\ &= \Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'}). \qquad (6.25)\end{aligned}$$

Where Ψ is the digamma function, which is the first derivative of the log Gamma function. Then we have

$$\mathbb{E}_q[\log \pi_k] = \Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'}). \tag{6.26}$$

Thus the first term of Equation 6.22 is computed as:

$$\begin{aligned}\mathbb{E}_{q(\pi|\lambda)}[\log P(\pi|\alpha)] &= \log \Gamma(\sum_k \alpha_k) - \sum_k \log \Gamma(\alpha_k) \\ &+ \sum_k (\alpha_k - 1) \left[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'}) \right]. \end{aligned} \tag{6.27}$$

Equivalently, the second term of Equation 6.22 is computed as:

$$\begin{aligned}\mathbb{E}_{q(\theta_k|\tau_k)}[\log P(\theta_k|G_0)] &= \log \Gamma(\sum_s \beta_s) - \sum_s \log \Gamma(\beta_s) \\ &+ \sum_s (\beta_s - 1) \left[\Psi(\tau_{k,s}) - \Psi(\sum_{s'} \tau_{k,s'}) \right]. \end{aligned} \tag{6.28}$$

The computation of the third term $\mathbb{E}_q[\log P(Z_i|\pi)]$ is different with the first two terms, since both involved variables (z_i and π) are unobservable. The expectation is computed as:

$$\begin{aligned}\mathbb{E}_{q(z_i|\eta_i)q(\pi|\lambda)}[\log P(z_i|\pi)] &= \sum_k \int q(z_i = k|\eta_i) q(\pi|\lambda) \log \pi_k d\pi \\ &= \sum_k \eta_{i,k} E_q \left[\log \pi_k \right] \\ &= \sum_k \eta_{i,k} \left[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'}) \right]. \end{aligned} \tag{6.29}$$

Equivalently, the fourth term is computed as:

$$\begin{aligned}\mathbb{E}_{q(z_i|\eta_i)\prod_k q(\theta_k|\tau_k)}[\log P(y_i|z_i, \Theta)] &= \sum_k \int q(z_i = k|\eta_i) q(\theta_k|\tau_k) \log \theta_{k,y_i} d\theta_k \\ &= \sum_k \eta_{i,k} E_q \left[\log \theta_{k,y_i} \right] \\ &= \sum_k \eta_{i,k} \left[\Psi(\tau_{k,y_i}) - \Psi(\sum_s \tau_{k,s}) \right]. \end{aligned} \tag{6.30}$$

At last, the negative entropy term is computed as:

$$\begin{aligned}
&\mathbb{E}_q \log q(Z, \pi, \Theta) \\
&= \mathbb{E}_q \log q(\pi|\lambda) + \sum_k \log q(\theta_k|\tau_k) + \sum_i \log q(z_i|\eta_i) \\
&= \log \Gamma(\sum_k \lambda_k) - \sum_k \log \Gamma(\lambda_k) + \sum_k (\lambda_k - 1) \left[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'}) \right] \\
&+ \sum_k \log \Gamma(\sum_s \tau_{k,s}) - \sum_s \log \Gamma(\tau_{k,s}) + \sum_s (\tau_{k,s} - 1) \left[\Psi(\tau_{k,s}) - \Psi(\sum_{s'} \tau_{k,s'}) \right] \\
&+ \sum_i \sum_k \eta_{i,k} \log \eta_{i,k}. \qquad (6.31)
\end{aligned}$$

After computing each term of Equation 6.22, we now optimize it with respect to the variational parameters via a coordinate ascent algorithm, which maximizes Equation 6.22 by iteratively optimizing each variational parameter given all others. Let us start from the optimization of λ.

$$\begin{aligned}
0 = \frac{\partial \mathcal{L}}{\partial \lambda_k} = \frac{\partial}{\partial \lambda_k} \Bigg[&(\alpha_k - 1)\Psi(\lambda_k) - \sum_k (\alpha_k - 1)\Psi(\sum_{k'} \lambda_{k'}) + \sum_i \eta_{i,k} \Psi(\lambda_k) \\
&- \sum_i \sum_k \eta_{i,k} \Psi(\sum_{k'} \lambda_{k'}) - \log \Gamma(\sum_k \lambda_k) + \log \Gamma(\lambda_k) - (\lambda_k - 1)\Psi(\lambda_k) \\
&+ \sum_k (\lambda_k - 1)\Psi(\sum_{k'} \lambda_{k'}) \Bigg]. \qquad (6.32)
\end{aligned}$$

After simplification, it yields:

$$0 = \Psi'(\lambda_k) \left[\alpha_k + \sum_i \eta_{i,k} - \lambda_k \right] - \Psi'(\sum_{k'} \lambda_{k'}) \sum_k \left[\alpha_k + \sum_i \eta_{i,k} - \lambda_k \right]. \qquad (6.33)$$

Thus the variational parameter λ_k is updated as:

$$\lambda_k = \frac{\alpha_0}{K} + \sum_i \eta_{i,k}. \qquad (6.34)$$

Equivalently, $\tau_{k,s}$ is updated as:

$$\tau_{k,s} = \frac{\beta_0}{S} + \sum_i \eta_{i,k} \delta_s(y_i). \qquad (6.35)$$

Where $\delta_s(y_i) = 1$ if $y_i = s$, 0 otherwise.

At last, the lower bound is optimized with respect to $\eta_{i,k}$ for $k = 1, \ldots, K$ and $i = 1, \ldots, N$.

$$\begin{aligned}
0 = \frac{\partial \mathcal{L}}{\partial \eta_{i,k}} = \frac{\partial}{\partial \eta_{i,k}} &\eta_{i,k} \left[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'}) \right] + \eta_{i,k} \left[\Psi(\tau_{k,x_i}) - \Psi(\sum_s \tau_{k,s}) \right] \\
&- \eta_{i,k} \log \eta_{i,k} + \mu(\sum_k \eta_{i,k} - 1), \qquad (6.36)
\end{aligned}$$

which yields:

$$\eta_{i,k} \propto \exp\left[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'}) + \Psi(\tau_{k,x_i}) - \Psi(\sum_{s} \tau_{k,s})\right]. \tag{6.37}$$

In summary, the coordinate ascent algorithm yields the following steps:

1. Initialize variational parameters $\eta_{i,k}$ for $i = 1, \ldots, N$ and $k = 1, \ldots, K$. In practice, we can assume $\eta_{i,k}^{(0)} = 1/K$. If it leads to local extreme values, we can run the algorithm with different random initializations and choose variational parameters with the best lower bound.

2. Repeat the following computation until convergence.

$$\lambda_k = \frac{\alpha_0}{K} + \sum_i \eta_{i,k}; \tag{6.38a}$$

$$\tau_{k,s} = \frac{\beta_0}{S} + \sum_i \eta_{i,k}\delta_s(y_i); \tag{6.38b}$$

$$\eta_{i,k} \propto \exp\left[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'}) + \Psi(\tau_{k,x_i}) - \Psi(\sum_{s} \tau_{k,s})\right]. \tag{6.38c}$$

The values for the unknown quantities must be statistically and graphically summarized to monitor convergence. There are mainly two methods to diagnose convergence. First, since the algorithm is to optimize the lower bound of the likelihood, we can monitor the sequence of lower bounds. If it does not change much with updating variational parameters, then we can say the process reaches stationary point. Alternatively, we can directly monitor the output. If the difference of the variational parameters between the two iterations is small enough, then we believe the convergence occurs. When the process reaches stationary point, the variational distribution $q(Z, \Theta, \pi)$ with the optimized variational parameters is an approximation to the posterior of the unobservable variables, by which the predictive inference can be performed. In addition, with the optimized variational parameters, Equation 6.22 provides a lower bound for the log-likelihood of the observations.

6.3.3 Parameter Estimate

In this section, we introduce empirical Bayesian methods to estimate hyperparameters in the finite Bayesian mixture model. In particular, given N observations $D = \{y_1, \ldots, y_N\}$, we wish to find the unknown hyperparameters α and β which maximize the expected complete-data log-likelihood:

$$\begin{aligned}\alpha^{ML}, \beta^{ML} &= \arg\max_{\alpha,\,\beta} \mathbb{E}\big[\log P(D, Z, \pi, \Theta|\alpha, \beta)\big] \\ &= \arg\max_{\alpha,\,\beta} \int \sum_Z P(Z, \pi, \Theta|D, \alpha, \beta) \log P(D, Z, \pi, \Theta|\alpha, \beta) d\pi d\Theta.\end{aligned}$$

As discussed in Section 6.3.2, the posterior distribution $P(Z, \pi, \Theta|D, \alpha, \beta)$ is computationally intractable. To solve the problem, we introduce two solutions: stochastic EM and variational EM.

Stochastic EM

Stochastic EM method was introduced by Celeux and Diebolt (1985) and Wei and Tanner (1990). The main idea of the method is to approximate the intractable (log)likelihood expectation with a sum over the samples generated from a MCMC method. If only one sample is drawn at each iteration (Celeux & Diebolt, 1985), then it is known as stochastic EM (SEM). If several samples are drawn (Wei & Tanner, 1990), then it is known as Monte Carlo EM (MCEM). In this section, we focus on SEM, which repeats the following two steps until convergence. In the *S-step*, a single sample is drawn for each of the unknown variables from its posterior distribution given the current estimations of the parameters. S-step provides us with pseudo-complete data. In the *M-step*, we maximize the log likelihood of the pseudo-complete data with respect to the parameters. Iteratively performing S-step and M-step, we obtain a Markov chain about the model parameters which converges to a stationary point. The sequence of samples generated by Stochastic EM method provides a region for the parameters of interest, which is often called *plausible region*. The mean of the samples approximates the maximum likelihood estimations of the parameters, and the variance intuitively implies the information loss due to the missing data.

For the full Bayesian mixture model, stochastic EM method draws samples for the unobservable variables Z, π and Θ in S-step, and optimizes the hyperparameters α and β in the M-step. Let illustrate SEM method with the example in Section 6.3.2, which yields following steps.

1. Initialize hyperparameters $\alpha_0^{(0)}$, $\beta_0^{(0)}$ and parameters $\pi^{(0)}$ and $\Theta^{(0)}$.

2. Iterate the following steps until convergence.

   (a) S step: draw samples for the unobservable variables $Z^{(t)}$, $\pi^{(t)}$ and $\Theta^{(t)}$ based on Equation 6.15, 6.16 and 6.17.

   (b) M step: view the pseudo-data as the real data, and maximize the log likelihood of the pseudo-complete data with respect to the hyperparameters via coordinate ascent method, i.e. alternatively optimize the two hyperparameters α_0 and β_0 until convergence.

$$\begin{aligned} 0 &= \frac{\partial}{\partial \alpha_0} \log P(D, Z, \pi, \Theta|\alpha_0, \beta_0) = \frac{\partial}{\partial \alpha_0} \log P(\pi|\alpha_0) \\ &= \Psi(\alpha_0) - \Psi(\alpha_0/K) + \log \pi_k \end{aligned} \tag{6.39}$$

   It is obvious that the above equation can not be solved analytically. Thus we consider the Newton's method, which is widely used to approximate the roots of a function. We repeat the following computation until convergence.

$$\alpha_0^{(t')} = \alpha_0^{(t'-1)} - \frac{\Psi(\alpha_0^{(t'-1)}) - \Psi(\alpha_0^{(t'-1)}/K) + \log \pi_k}{\Psi'(\alpha_0^{(t'-1)}) - \frac{1}{K}\Psi'(\alpha_0^{(t'-1)}/K)}. \tag{6.40}$$

Equivalently, β_0 is iteratively updated as:

$$\beta_0^{(t')} = \beta_0^{(t'-1)} - \frac{\Psi(\beta_0^{(t'-1)}) - \Psi(\beta_0^{(t'-1)}/S) + \log \theta_{k,s}}{\Psi'(\beta_0^{(t'-1)}) - \frac{1}{S}\Psi'(\beta_0^{(t'-1)}/S)}. \tag{6.41}$$

Variational EM

Due to possible low efficiency of MCMC methods, an alternative solution, variational EM, is considered, which is introduced in (Neal & Hinton, 1998; Jordan et al., 1998). The method combines the lower bound of incomplete-data log-likelihood with parameter estimation via EM algorithm. As discussed in Section 6.2.2, EM algorithm maximizes the expected complete-data log-likelihood. Let ξ denote the unobservable variables, D denote the observations, φ denote the model parameters, the expected complete-data log-likelihood is written as:

$$\mathcal{Q} = \sum_{\xi} P(\xi|D, \varphi) \log P(D, \xi|\varphi). \tag{6.42}$$

When the posterior distribution $P(\xi|D, \varphi)$ of unobservable variables is intractable, we can not apply EM algorithm directly. To solve the problem, variational EM is introduced. We define the lower bound $\mathcal{L}(q, \varphi)$ of the incomplete-data log-likelihood via *Jensen's inequality*:

$$\mathcal{L}(q, \varphi) = \mathbb{E}_q\big[\log P(D, \xi|\varphi)\big] - \mathbb{E}_q\big[\log q(\xi)\big]. \tag{6.43}$$

We iterate the following two steps until convergence. In the *E-step*, we maximize the lower bound with respect to the variational distribution q. As discussed in Section 4.4.2, the step actually optimizes the variational distribution to approximate the real posterior given the current estimation of model parameters φ. In the *M-step*, we maximize the lower bound with respect to φ. The step is equivalent to the traditional presentation of the EM algorithm, since the partial derivative of lower bound $\mathcal{L}$ with respect to φ for fixed q equals to the partial derivative of the function

$$\mathbb{E}_q\big[\log P(D, \xi|\varphi)\big],$$

which is the expectation of complete-data log-likelihood. More formally, the variational EM algorithm is defined as:

1. Initialize $q^{(0)}$ and $\varphi^{(0)}$.

2. Iterate the following steps until convergence.

$$\text{E step:} \qquad q^{(t)} = \arg\max_{q} \ \mathcal{L}(q, \varphi^{(t-1)}) \tag{6.44}$$

$$\text{M step:} \qquad \varphi^{(t)} = \arg\max_{\varphi} \ \mathcal{L}(q^{(t)}, \varphi), \tag{6.45}$$

which can be viewed as coordinate ascent update over lower bound $\mathcal{L}$.

Now we introduce the variational EM for finite Bayesian mixture model. For computational efficiency, we consider the fully-factorized variational distribution defined in Equation 6.21, thus the lower bound of the incomplete-data log-likelihood is defined as Equation 6.22. In E-step, the lower bound is optimized with respect to the variational parameters given the current hyperparameters α and β, which has been discussed in Section 6.3.2. In M-step, the lower bound is optimized with respect to the hyperparameters. Following we discuss the computation in M-step via the running example in Section 6.3.2. In this case, the hyperparameters α and β are reduced to two positive real-value scalars, α_0 and β_0. We appeal to coordinate ascent algorithm to optimize the two hyperparameters.

$$\begin{aligned} 0 = \frac{\partial \mathcal{L}}{\partial \alpha_0} &= \frac{\partial}{\partial \alpha_0} \log \Gamma(\alpha_0) - \log \Gamma(\alpha_0/K) + (\alpha_0/K - 1)\big[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'})\big] \\ &= \Psi(\alpha_0) - \frac{1}{K}\Psi(\alpha_0/K) + \frac{1}{K}[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'})]. \end{aligned} \tag{6.46}$$

It is obvious that the equation is not easy to solve. We again employ the Newton's method to find the roots. It yields:

$$\alpha_0^{(t')} = \alpha_0^{(t'-1)} - \frac{f(\alpha_0^{(t'-1)})}{f'(\alpha_0^{(t'-1)})} \tag{6.47}$$

where

$$f(\alpha_0^{(t'-1)})) = \Psi(\alpha_0^{(t'-1)}) - \frac{1}{K}\Psi(\alpha_0^{(t'-1)}/K) + \frac{1}{K}[\Psi(\lambda_k) - \Psi(\sum_{k'} \lambda_{k'})] \tag{6.48}$$

$$f'(\alpha_0^{(t'-1)}) = \Psi'(\alpha_0^{(t'-1)}) - \frac{1}{K^2}\Psi'(\alpha_0^{(t'-1)}/K). \tag{6.49}$$

Equivalently, β_0 is updated as:

$$\beta_0^{(t')} = \beta_0^{(t'-1)} - \frac{f(\beta_0^{(t'-1)})}{f'(\beta_0^{(t'-1)})}, \tag{6.50}$$

where

$$f(\beta_0^{(t'-1)}) = K \times [\Psi(\beta_0^{(t'-1)}) - \Psi(\beta_0^{(t'-1)}/K)] + \sum_k \left[\Psi(\tau_{k,s}) - \Psi(\sum_{s'} \tau_{k,s'})\right] \tag{6.51}$$

$$f'(\beta_0^{(t'-1)}) = K \times \Psi'(\beta_0^{(t'-1)}) - \Psi'(\beta_0^{(t'-1)}/K). \tag{6.52}$$

6.3.4 Predictive Inference

In this section, we discuss the predictive inference of mixture model in full Bayesian framework. There are two main prediction tasks: one is to predict the new observations $P(y_{new}|D, \alpha, \beta)$, the other is to predict the cluster assignments of new observations $P(z_{new} = k|y_{new}, D, \alpha, \beta)$. As the unknown parameters π and Θ themselves are random variables in full Bayesian framework, the predictive probabilities can not be directly computed as Section 6.2.3.

Predictive inference with Gibbs sampling

At each iteration of the MCMC method z_i, θ_k and π are drawn for $i = \{1, \ldots, N\}$ and $k = \{1, \ldots, K\}$. When the sequence converges, the predictive distributions are approximated over the samples. In particular, the first w members are discarded as *burn-in* period and the last W members of the sequence are collected to estimate the distributions of interest.

$$\begin{aligned}
P(y_{new}|D, \alpha, \beta) &= \frac{1}{W} \sum_{t=w+1}^{W+w} P(y_{new}|\pi^{(t)}, \Theta^{(t)}) \\
&= \frac{1}{W} \sum_{t=w+1}^{W+w} \sum_{k}^{K} \pi_k^{(t)} P(y_{new}|\theta_k^{(t)}) \\
P(z_{new} = k|y_{new}, D, \alpha, \beta) &= \frac{1}{W} \sum_{t=w+1}^{W+w} \frac{\pi_k^{(t)} P(y_{new}|\theta_k^{(t)})}{\sum_k^K \pi_k^{(t)} P(y_{new}|\theta_k^{(t)})}
\end{aligned} \tag{6.53}$$

Predictive inference with variational approximation

When the coordinate ascent procedure reaches stationary point, the variational inference method yields the optimized variational parameters and the corresponding variational distribution is a close approximation to the true posterior of the unobservable variables (Z, Θ and π), over which the prediction probabilities are computed.

$$\begin{aligned}
P(y_{new}|D, \alpha, \beta) &= \sum_{z_{new},\, \pi,\, \Theta} P(y_{new}, z_{new}, \pi, \Theta|D, \alpha, \beta) \\
&= \sum_k \int P(\pi, \Theta|D, \alpha, \beta) \pi_k P(y_{new}|\theta_k) d\pi d\Theta \\
&\approx \sum_k \int \pi_k q(\pi|\lambda) d\pi \int P(y_{new}|\theta_k) q(\theta_k|\tau_k) d\theta_k \\
&= \sum_k \mathbb{E}_q(\pi_k) \mathbb{E}_q \left[P(y_{new}|\theta_k)\right].
\end{aligned}$$

$$P(z_{new} = k|y_{new}, D, \alpha, \beta) = \frac{\mathbb{E}_q(\pi_k) \mathbb{E}_q \left[P(y_{new}|\theta_k)\right]}{\sum_k \mathbb{E}_q(\pi_k) \mathbb{E}_q \left[P(y_{new}|\theta_k)\right]}. \tag{6.54}$$

6.4 Summary

In this chapter, we introduce the finite mixture model, which is widely used to solve the clustering/classification problems. But there is one limitation in applying mixture models, i.e. it is difficult to decide the number of mixture components in advance. A principal solution for the problem is to embed the finite mixture model into a nonparametric Bayesian framework, such that the number of mixture components can be very flexible and can be optimized by the model itself based on the complexity of the data. In the next chapter we will introduce the infinite mixture models and the corresponding inference methods.

Chapter 7

Infinite Mixture Models

7.1 Introduction

The mixture models introduced in Chapter 6 are *finite* models, which assume that there are a finite number of mixture components from which the samples are generated. However the assumption is not always practical, since in more cases than not it is difficult to known the number of components in advance. If a wrong number is specified to the model, the estimation will be completely divergent from the real situation. To remove the constraint, many methods are developed. For example, Richardson and Green (1997) introduced a hierarchical Bayesian method which encodes the uncertainty about the number of components via a new random variable, and then optimizes its value via a reversible jump MCMC sampler. Unfortunately, the method might be ineffective when new data generated from new components are available. In this situation, the model parameters have to be trained again, the information learned in previous analysis can not be used to accelerate the future learning process. A possible solution for the problem is to embed the finite mixture model in a nonparametric Bayesian framework, e.g. Dirichlet process (DP), such as the number of mixture components is not restricted and will be optimized with respect to the data in a self-organized way. The new model is called *infinite mixture model.* The term *infinite* does not mean the number of mixture components are infinite, but the number is flexible and not fixed in advance. Due to combining with Dirichlet process, the infinite mixture model is also referred to as *Dirichlet process mixture model.*

In a DP mixture model, the underlying sample θ_i generated from a DP is treated as the parameters of the distribution of the observation y_i, i.e. $y_i \sim P(\cdot|\theta_i)$. The model takes advantage of the discreteness property of Dirichlet process, in particular, a distribution drawn from a DP places its probability mass on a countably infinite subset of the underlying sample (θ_i) space. The parameters θ_i's are viewed as the hidden variables, one for each observation. They are indicators to specify which components the observations are generated. The observations with identical parameter values are assumed to be the members in the same cluster. Thus Dirichlet process provides a clustering effect for the observations. Furthermore, the parameters for a new observation may take on existing values or new values, i.e. the new observation is a member of an existing cluster or a member of a new cluster. That means new mixture components continue to emerge with additional data

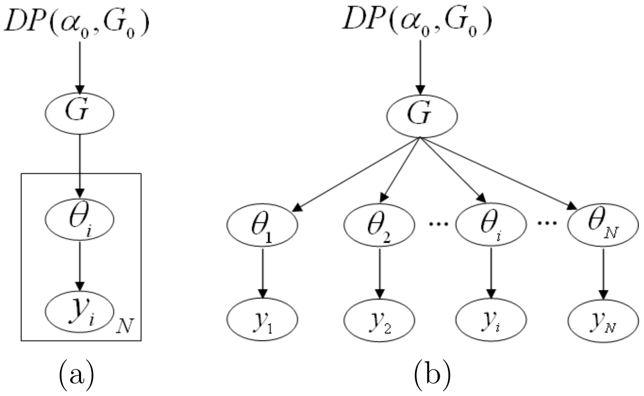


Figure 7.1: Dirichlet process mixture model.

as many as necessary. Therefore the DP mixture model might have an infinite number of clusters and infer the structure of the data automatically and increasingly.

For inference computation in a DP mixture model, the traditional solution is Markov chain Monte Carlo (MCMC) simulation methods, including collapsed Gibbs sampling (Neal, 2000), blocked Gibbs sampling (Ishwaran & James, 2001) and so on. For computational efficiency, Blei and Jordan (2005) introduced a variational inference method, which is motivated by the truncated stick-breaking construction. While less accurate than MCMC sampling methods, the variational approximation provides a fast solution for inference in a DP mixture model.

7.2 Model Description

Dirichlet Process mixture model was introduced by Antoniak (1974), which embeds the finite mixture model in a nonparametric Bayesian framework as shown in Figure 7.1. The generative process of the model is defined as:

$$\begin{aligned} G|G_0, \alpha_0 &\sim DP(G_0, \alpha_0) \\ \theta_i|G &\sim G(\cdot),\ i = 1, \dots, N \\ y_i|\theta_i &\sim P(\cdot|\theta_i). \end{aligned}$$

In particular, we draw a prior distribution G from a DP with hyperparameters α_0 and G_0, draw parameters $\Theta = \{\theta_1, \dots, \theta_N\}$ from G, and draw observations conditioned on the corresponding parameters. The observations with identical parameter values are generated from the same mixture components, thus have the same cluster assignments. Suppose there are $K \leq N$ distinct values $\{\theta_1^*, \dots, \theta_K^*\}$ in the N parameters. Then the DP mixture model partitions the observations into K groups in a nature way. The joint probability of the model is defined as:

$$P(G|\alpha_0, G_0) \prod_{i=1}^{N} P(\theta_i|G) P(y_i|\theta_i) \qquad (7.1)$$

There is a practical problem in the DP mixture model, i.e. it is difficult to drawn G directly from a DP given hyperparameters α_0 and G_0, since the probability function space is infinite. There are mainly three approaches to remove the computational constraint. The first approach is the Pólya Urn process, we have discussed in Section 4.3.3. The second approach is the *Chinese restaurant process (CRP)* (Aldous, 1985), which directly draws the parameters θ_i with an auxiliary variable z_i. This solution supplies an explanation about the clustering effect of DP. The third approach is the *stick breaking construction* (SBC) (Sethuraman, 1994), which explicitly draws a random distribution G from a DP as a sum of infinite weighted components.

7.2.1 Chinese Restaurant Process

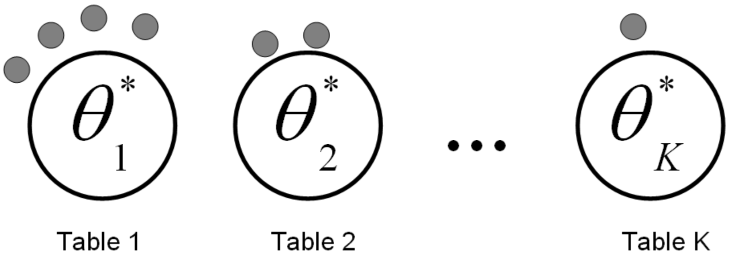


Figure 7.2: Chinese Restaurant Process (Aldous, 1985). After N customers have entered a restaurant, K tables are occupied. Associated with the table k are parameters θ_k^*.

Chinese restaurant process (CRP) was introduced by Aldous (1985), which provides a sampling method to integrate out the random distribution G and directly draws underlying samples from a DP. In Chinese restaurant process it is assumed that customers sit down in a Chinese restaurant with an infinite number of tables. An auxiliary discrete variable z_i is introduced for each custom, the fact that $z_i = k$ means that customer i sits at table k. Associated with each table k are parameters θ_k^*, that are independently drawn from the base distribution G_0. The Chinese restaurant process can be shown as Figure 7.2. In detail:

1. The first customer sits at the first table, $z_1 = 1$; and θ_1^* are generated from G_0 for the table.

2. With probability $1/(1+\alpha_0)$, the second customer also sits at the first table, $z_2 = 1$, and inherits θ_1^*; with probability $\alpha_0/(1+\alpha_0)$ the customer sits at the second table, $z_2 = 2$, and new parameters are generated, $\theta_2^* \sim G_0$, for the second table.

3. We continue this process, after N customers have entered the room, K tables are occupied, N_k customers occupy table k.

4. Customer $N+1$ enters the restaurant, he sits with probability

$$\frac{N_k}{N+\alpha_0} \tag{7.2}$$

at a previously occupied table k and inherits θ_k^*. Thus: $z_{N+1} = k$, $N_k \leftarrow N_k + 1$

5. With probability
$$\frac{\alpha_0}{N + \alpha_0} \tag{7.3}$$
the customer sits at a new table $K + 1$. Thus: $z_{N+1} = K + 1$, $N_{K+1} = 1$.

6. For the new table, new parameters are generated, $\theta_{K+1}^* \sim G_0$ and $K \leftarrow K + 1$.

When the auxiliary variable z_i is sampled for each observation (customer), their parameters $\theta_{z_i}^*$ are decided, and then the observation is sampled from $P(\cdot|\theta_{z_i}^*)$. The Chinese restaurant process clearly exhibits clustering effect of DP. The observations are randomly partitioned according to their hidden variables. The observations with the identical hidden state are in the same cluster. With probability proportional to N_k (Equation 7.2), a new observation is assigned to an existing cluster, with probability proportional to α_0 (Equation 7.3), the new observation is assigned to a new cluster. It is obvious that the larger α_0 is, the more likely new clusters emerge. The larger the size of a cluster is, the more likely a new observation is assigned to the cluster. From this point of view, DP mixture model can be viewed as a flexible mixture model where the number of mixture components is optimized by the model itself and might increase when new observations are available.

7.2.2 Stick Breaking Construction

Stick breaking construction (SBC) is another well-known representation of DP, which was introduced by Sethuraman (1994). The method explicitly draws the random distribution G from DP(G_0, α_0) as an infinite sum:
$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}. \tag{7.4}$$

Where $\delta_{\theta_k^*}$ is a distribution with a point mass on θ_k^*. $\pi_k > 0$ and $\sum_{k=1}^{\infty} \pi_k = 1$. The SBC representation highlights the view of *infinite mixture*. θ_k^* can be interpreted as mixture components, and $\theta_k^* \overset{\text{i.i.d.}}{\sim} G_0$ are independently drawn from the base distribution G_0. $\pi = (\pi_1, \ldots, \pi_\infty)$ can be viewed as mixture weights, and are generated via the stick breaking procedure, denoted as Stick(α_0):
$$v_k \sim Beta(1, \alpha_0); \quad \pi_1 = v_1, \;\; \pi_k = v_k \prod_{k'=1}^{k-1} (1 - v_{k'}). \tag{7.5}$$

For the DP mixture model with stick breaking representation shown as Figure 7.3, it is convenient to introduce an auxiliary hidden variable z_i with an infinite number of states for each observation. Samples from G can now equivalently be generated by selecting a state of z_i with probability defined by Stick(α_0) and by then again inherit the corresponding mixture components specified by the state of z_i. The generative model can be defined as:

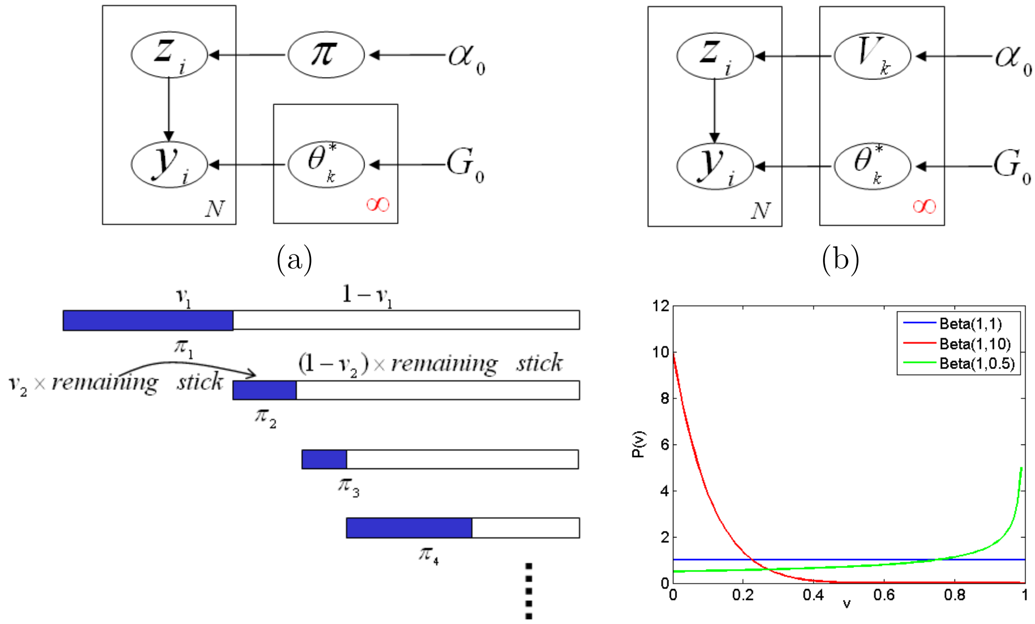


Figure 7.3: Dirichlet process mixture model with stick-breaking construction. (a) The mixture weights are shown as π with infinite dimensions. (b) The mixture weights are shown as an infinite number of v_k's. (c) The process of how to generate π from v_k's, i.e. Equation 7.5. (d) Beta distributions with different parameters.

1. Draw mixing weights $\pi \sim \text{Stick}(\alpha_0)$ as Equation 7.5.
2. Draw i.i.d. parameters $\theta^*_k \sim G_0$, $k = 1, 2, \ldots$.
3. For each observation y_i,
   (a) Draw $z_i \sim \text{Mult}(\pi)$,
   (b) Draw $y_i \sim P(\cdot|\theta^*_{z_i})$.

Ishwaran and James (2001) developed *truncated Dirichlet process* (TDP), which is equivalent to Dirichlet process except that there are only K distinct mixture components. We can obtain TDP by setting $v_K = 1$. The truncated DP closely approximates a true Dirichlet process when K is large enough. The value of K is related to the number of observations.

7.3 Inference

The key inferential problem in a DP mixture model is to compute the joint posterior distribution of the unobservable variables given observations $D = \{y_1, \ldots, y_N\}$. The unobservable variables in the model include: random distribution G and parameters $\Theta =$

$\{\theta_1, \ldots, \theta_N\}$, one for each observation. Thus the joint posterior distribution is defined as:

$$P(G, \Theta|D, \alpha_0, G_0) = \frac{P(G|\alpha_0, G_0) \prod_{i=1}^{N} P(\theta_i|G)P(y_i|\theta_i)}{P(D|\alpha_0, G_0)}, \tag{7.6}$$

where the normalization term $P(D|\alpha_0, G_0)$ is:

$$\int P(G|\alpha_0, G_0) \prod_{i=1}^{N} P(\theta_i|G)P(y_i|\theta_i) d\theta_1, \ldots, d\theta_N dG. \tag{7.7}$$

Unfortunately, the computation of the posterior is analytically intractable. To solve the problem, we consider to represent DP via Chinese restaurant process or stick breaking construction. Although the exact inference is impossible, many approximate methods can be considered, e.g. collapsed Gibbs sampling with CRP (Neal, 2000), blocked Gibbs sampling with truncated SBC (Ishwaran & James, 2001) and variational inference with truncated SBC (Blei & Jordan, 2005). Note, that the unobservable variables are different when using different DP representations.

In the following sections, we will illustrate these inference methods with a particular example. Assume that there are N observations $D = \{y_1, \ldots, y_N\}$ generated from a DP mixture model with parameters α_0 and G_0. y_i is a discrete variable with S possible states and follows multinomial distribution with parameters θ_i. For computational efficiency, we assume G_0 is a conjugate Dirichlet distribution. The assumption is not so strict, since the prior G sampled from a DP can be of arbitrary mathematic form despite a Dirichlet base distribution G_0. The parameters of base distribution G_0 are (β_0, β), where $\beta_0 > 0$, $\beta = (\beta_1, \ldots, \beta_S)$, $\beta_s > 0$ and $\sum_{s=1}^{S} \beta_s = 1$. We assume $\beta_s = 1/S$, which means we believe that each state occurs with equal probability. Thus there are only two hyperparameters α_0 and β_0 in the running example. β_0 represents our confidence about the prior belief that the multinomial parameters should be equal. α_0 represents how strongly the parameters of observations should be coupled. If α_0 is chosen to be small, only few clusters are generated and the parameters tend to be highly coupled. If α_0 is chosen to be large, the coupling is loose and more clusters are formed.

7.3.1 Collapsed Gibbs Sampling with CRP

Neal (2000) introduced a collapsed Gibbs sampling method for inference in the DP mixture model, which integrates out all random variables except for the auxiliary variables $Z = \{z_1, \ldots, z_N\}$. The Markov chain of the method is thus defined only on the auxiliary variables. The main idea of the method is to iteratively sample each hidden variables z_i conditioned on the others Z_{-i} until the procedure reaches a stationary point. In particular, z_i is updated as follows:

1. $z_i = k$ and the observation y_i inherits the parameters θ_k^* assigned to the component k:

$$P(z_i = k|D, Z_{-i}, \alpha_0, G_0) \propto N_k \; P(y_i|D_{-i}, z_i = k, Z_{-i}, G_0), \tag{7.8}$$

where D_{-i} denotes all observations except for y_i.

2. Instead, a new state $K+1$ is generated with probability

$$P(z_i = K+1|D, Z_{-i}, \alpha_0, G_0) \propto \alpha_0 \; P(y_i|G_0). \tag{7.9}$$

   Accordingly, new parameters θ^*_{K+1} are drawn from G_0.

3. Each term in Equation 7.8 and Equation 7.9 is computed as:

$$\begin{aligned} P(y_i|D_{-i}, z_i = k, Z_{-i}, G_0) &= \int P(\theta^*_k|D_{-i}, Z_{-i}, G_0)P(y_i|\theta^*_k)d\theta^*_k \\ &= \mathbb{E}_{P(\theta^*_k|D_{-i},Z_{-i},G_0)}\left[P(y_i|\theta^*_k)\right]. \end{aligned} \tag{7.10}$$

$$\begin{aligned} P(y_i|G_0) &= \int P(\theta^*_{new}|G_0)P(y_i|\theta^*_{new})d\theta^*_{new} \\ &= \mathbb{E}_{P(\theta^*_{new}|G_0)}\left[P(y_i|\theta^*_{new})\right]. \end{aligned} \tag{7.11}$$

To perform the Gibbs sampling, the integrations in Equation 7.10 and 7.11 need to be computed. It is tractable if $P(\theta^*_k|G_0)$ and $P(y_i|\theta^*_k)$ are assumed to be of manageable form. For example, we assume that $P(y_i|\theta^*_k)$ is a distribution in the exponential family and $P(\theta^*_k|G_0)$ is conjugated with $P(y_i|\theta^*_k)$. If $P(\theta^*_k|G_0)$ and $P(y_i|\theta^*_k)$ are not of manageable form, we can consider a numerical method to implement the integrations, e.g. Gaussian quadrature method (Naylor & Smith, 1982; Evans & Swartz, 1995).

We now discuss the computational details in the sampling process via the running example. In detail:

1. For each observation y_i, initialize $z_i^{(0)}$. In practice, we can assume that each observation in its own cluster, i.e. $z_i^{(0)} = i$.

2. Iterate the following steps for $t = 1, 2, \ldots$.

   - Update $z_i^{(t)}$ conditioned on $\{z_1^{(t)}, \ldots, z_{i-1}^{(t)}, z_{i+1}^{(t-1)}, \ldots, z_N^{(t-1)}\}$.
   - Assign $z_i^{(t)}$ an existing value with probability proportional to

$$N_k^{(t)} \int P(\theta^*_k|D_{-i}, Z_{-i}^{(t)}, G_0)P(y_i|\theta^*_k)d\theta^*_k. \tag{7.12}$$

   Where $N_k^{(t)}$ is the number of observations with hidden state k at iteration t. $P(y_i|\theta^*_k) = \theta^*_{k,y_i}$. $P(\theta^*_k|D_{-i}, Z_{-i}^{(t)}, G_0)$ is the posterior distribution of θ^*_k. Since we assume a conjugate distribution for G_0, $P(\theta^*_k|D_{-i}, Z_{-i}^{(t)}, G_0)$ is still a Dirichlet distribution, but the parameters become

$$\beta_{post} = (\frac{\beta_0}{S} + N^{(t)}(k,1), \ldots, \frac{\beta_0}{S} + N^{(t)}(k,S)).$$

Where $N^{(t)}(k,s)$ is the *sufficient statistic* about the observations, which is the number of observations with hidden state k and value s at iteration t, $N_k^{(t)} = \sum_s N^{(t)}(k,s)$. In summary, Equation 7.12 equals to

$$N_k^{(t)} \frac{\frac{\beta_0}{S} + N^{(t)}(k, y_i)}{\beta_0 + N_k^{(t)}}.$$

- Instead, a new value is generated with probability proportional to

$$\begin{aligned} &\alpha_0 \int P(\theta^*_{new}|G_0)P(y_i|\theta^*_{new})d\theta^*_{new} \\ &= \alpha_0 \mathbb{E}_{P(\theta^*_{new}|G_0)}(\theta^*_{new,y_i}) = \frac{\alpha_0}{S}. \end{aligned} \tag{7.13}$$

- $i \leftarrow i+1$, go to update the next $z_i^{(t)}$.

3. Stop until a stationary point reaches.

7.3.2 Blocked Gibbs Sampling with Truncated DP

In the collapsed Gibbs sampling method, the hidden variables are updated one at a time which potentially slows down the method. In this section, we introduce another Gibbs sampling method, which exploits the truncated stick breaking process (Ishwaran & James, 2001). The method is sometimes referred to as blocked Gibbs sampler (BGS). In BGS, the posterior distribution for hidden variables can be explicitly drawn from DP via a truncated stick-breaking construction. The advantage is that: given the posterior, one can independently sample the auxiliary variables in a block. In each iteration of the Markov chain, the sampled variables include not only the hidden variables $Z = \{z_1, \dots, z_N\}$, but also the parameters, π and $\Theta^* = \{\theta_1^*, \dots, \theta_K^*\}$. In detail:

1. Sample the hidden variable z_i independently:

$$\begin{aligned} P(z_i = k|D, Z_{-i}, \pi, \Theta, \alpha_0, G_0) &= P(z_i = k|y_i, \pi) \\ &\propto \pi_k P(y_i|\theta_k^*). \end{aligned} \tag{7.14}$$

2. Update π as follows:

   (a) Sample v_k independently from the distribution Beta$(\lambda_{k,1}, \lambda_{k,2})$ for $k = 1, \dots, K-1$ with

$$\lambda_{k,1} = 1 + \sum_{i=1}^{N} \delta_k(z_i), \quad \lambda_{k,2} = \alpha_0 + \sum_{k'=k+1}^{K} \sum_{i=1}^{N} \delta_{k'}(z_i), \tag{7.15}$$

   and set $v_K = 1$. Where $\delta_k(z_i)$ equals to 1 if $z_i = k$ and 0 otherwise.

   (b) Compute $\pi_1 = v_1$; $\pi_k = v_k \prod_{k'=1}^{k-1}(1 - v_{k'}), k > 1$.

3. Update the parameters $\Theta^* = \{\theta_1^*, \ldots, \theta_K^*\}$ conditioned on Z and D:

$$\theta_k^* \sim P(\cdot|D, Z, G_0). \tag{7.16}$$

The main computation in the blocked Gibbs sampler is to compute the posterior distribution $P(\theta_k^*|D, Z, G_0)$ in Equation 7.16. Again if G_0 and $P(y_i|\theta_k^*)$ are of manageable form, the computation is analytically tractable, e.g. $P(\theta_k^*|G_0)$ is conjugate with $P(y_i|\theta_k^*)$. In addition, the blocked Gibbs sampling method exploits the truncated Dirichlet process, thus we need to decide the truncation parameter K in advance. When K is large enough, the resulting distribution can be closely approximate the true distribution G. The value of K is decided by the complexity of the data. In practice, we can set K as the number of observations, and it will be automatically optimized in the sampling process.

For the running example, the blocked GS yields the following steps:

1. Take some initial values for $\Theta^{*(0)}$ and $\pi^{(0)}$. Note that the constraint $\sum_k \pi_k = 1$ should be satisfied. In practice, we can assume $\pi_k^{(0)} = 1/K$.

2. Repeat for $t = 1, 2, \ldots$

   (a) The hidden variable z_i is independently sampled with probability

   $$P(z_i^t = k|y_i, \pi^{(t)}, \theta_k^{*(t)}) \propto \pi_k^{(t)} P(y_i|\theta_k^{*(t)}) = \pi_k^{(t)} \theta_{k,y_i}^{*(t)}. \tag{7.17}$$

   (b) $\pi^{(t)}$ are updated as Equation 7.15.

   (c) The parameters $\theta_k^{*(t)}$ are sampled from the posterior distribution

   $$P(\theta_k^{*(t)}|D, Z, G_0) = \mathrm{Dir}(\cdot|\beta_{post}) \tag{7.18}$$

   Where $\beta_{post} = (\frac{\beta_0}{S} + N^{(t)}(k,1), \ldots, \frac{\beta_0}{S} + N^{(t)}(k,S))$. $N^{(t)}(k,s)$ is defined in Section 7.3.1.

3. Stop until the joint distribution of Z, π and Θ^* converges.

The blocked GS method for the DP mixture model is similar to the GS method for the finite mixture model (Section 6.3.2), except for the sampling procedure of mixture weights π. For the DP mixture model, π is sampled from a truncated stick-breaking construction as Equation 7.15, but in the finite mixture model π are directly drawn from a posterior Dirichlet distribution. In particular situations, the two models might give similar results, but the underlying mechanisms are completely different.

7.3.3 Mean Field with Truncated DP

Although the blocked GS method is faster than the collapsed GS method by independently sampling the hidden variables Z in a block, efficiency of the block GS method might be still lower than our expectation, especially when the data is multi-variate, large-scale or highly-correlated. Thus Blei and Jordan (2005) introduced a mean-field inference method

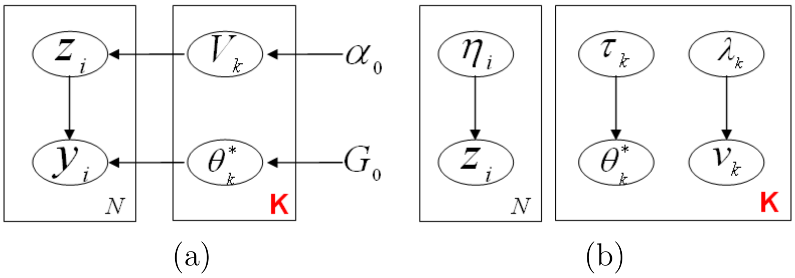


Figure 7.4: (a) DP mixture model with truncated stick-breaking construction. (b) Graphic representation of the variational distribution assumed to approximate the posterior of the unobservable variables in the DP mixture model. η_i, τ_k and λ_k are variational parameters. Note, that there are one η_i for each observation, one τ_k and one λ_k for each mixture component.

for the DP mixture model, which exploits the truncated stick-breaking construction of DP. Mean-field is a simplest variation inference method, which assumes a fully-factorized family of variational distributions to approximate the distribution of interest, Section 6.3.2 describes the main idea of mean field method. For more details, please refer to (Jordan et al., 1998).

Figure 7.4(a) shows the DP mixture model with truncated stick-breaking construction, where the unobservable variables include $Z = \{z_1, \ldots, z_N\}$, $V = \{v_1, \ldots, v_K\}$ and $\Theta^* = \{\theta_1^*, \ldots, \theta_K^*\}$. Based on Jensen's inequality, we obtain the lower bound of the log likelihood of the observations:

$$\log P(D|\alpha_0, G_0) \geq \mathbb{E}_q\big[\log P(D, Z, V, \Theta^*|\alpha_0, G_0)\big] - \mathbb{E}_q\left[\log q(Z, V, \Theta^*)\right], \tag{7.19}$$

where $q(Z, V, \Theta^*)$ denotes the variational distribution. The larger the lower bound is, the closer the variational distribution $q(Z, V, \Theta^*)$ approximates to the joint posterior $P(Z, V, \Theta^*|D, \alpha_0, G_0)$. Thus the posterior inference problem is now converted to an optimization problem: to maximize the lower bound with respect to the variational distribution. For computational efficiency, a fully-factorized family of variational distributions is assumed:

$$q(Z, V, \Theta^*) = \prod_k^K q(v_k|\lambda_k) q(\theta_k^*|\tau_k) \prod_i^N q(z_i|\eta_i). \tag{7.20}$$

λ_k, τ_k and η_i are variational parameters. Note, that there is one τ_k and one λ_k for each mixture component and one η_i for each observation. $q(v_k|\lambda_k)$ is a Beta distribution, $q(\theta_k^*|\tau_k)$ is of the same mathematic form as G_0, $q(z_i|\eta_i)$ is a multinomial distribution. The variational distributions decouple some probabilistic dependencies, e.g. the hidden variables $Z = \{z_1, \ldots, z_N\}$ are no longer drawn from a common multinomial prior, shown as Figure 7.4(b). Given the fully-factorized variational distribution, the lower bound of the log-likelihood of observations is now defined as:

$$\begin{aligned}\log P(D|\alpha_0, G_0) \geq & \sum_k^K \mathbb{E}_q \left[\log P(v_k|\alpha_0)\right] + \sum_k^K \mathbb{E}_q \left[\log P(\theta_k^*|G_0)\right] \\ & + \sum_i^N \mathbb{E}_q \left[\log P(z_i|V)\right] + \sum_i^N \mathbb{E}_q \left[\log P(y_i|z_i, \Theta)\right] \\ & - \mathbb{E}_q \left[\log q(Z, V, \Theta)\right]. \end{aligned} \tag{7.21}$$

Following, we discuss computation of each term in Equation 7.21 via the running example. Let us start from the first term.

$$\begin{aligned} & \mathbb{E}_{q(v_k|\lambda_k)} \left[\log P(v_k|\alpha_0)\right] \\ &= \mathbb{E}_{q(v_k|\lambda_k)} \left[\log \frac{\Gamma(1+\alpha_0)}{\Gamma(\alpha_0)}(1-v_k)^{\alpha_0-1}\right] \\ &= \log\Gamma(1+\alpha_0) - \log\Gamma(\alpha_0) + (\alpha_0-1)\mathbb{E}_{q(v_k|\lambda_k)} \left[\log(1-v_k)\right]. \end{aligned} \tag{7.22}$$

As discussed in Section 6.3.2, for an exponential family of distribution, the first derivative of normalization factor is the expectation of sufficient statistic of the distribution. Since the variational distribution $q(v_k|\lambda_k)$ is a Beta distribution $\text{Beta}(\lambda_{k,1}, \lambda_{k,2})$, we obtain:

$$\mathbb{E}_{q(v_k|\lambda_k)} \left[\log(1-v_k)\right] = \Psi(\lambda_{k,2}) - \Psi(\lambda_{k,1}+\lambda_{k,2}). \tag{7.23}$$

Thus the first term is computed as

$$\begin{aligned} & \mathbb{E}_{q(v_k|\lambda_k)} \left[\log P(v_k|\alpha_0)\right] \\ &= \log\Gamma(1+\alpha_0) - \log\Gamma(\alpha_0) + (\alpha_0-1)\left[\Psi(\lambda_{k,2}) - \Psi(\lambda_{k,1}+\lambda_{k,2})\right]. \end{aligned} \tag{7.24}$$

The second and fourth terms in Equation 7.21 are computed in an equivalent way like Section 6.3.2.

$$\begin{aligned}\mathbb{E}_{q(\theta_k^*|\tau_k)}[\log P(\theta_k^*|G_0)] = & \log\Gamma(\sum_s \beta_s) - \sum_s \log\Gamma(\beta_s) \\ & + \sum_s(\beta_s-1)\left[\Psi(\tau_{k,s}) - \Psi(\sum_{s'} \tau_{k,s'})\right]. \end{aligned} \tag{7.25}$$

$$\mathbb{E}_{q(z_i|\eta_i)\prod_k q(\theta_k|\tau_k)}[\log P(y_i|z_i, \Theta)] = \sum_k \eta_{i,k}\left[\Psi(\tau_{k,y_i}) - \Psi(\sum_s \tau_{k,s})\right]. \tag{7.26}$$

The computation of the third term $\mathbb{E}_q\left[\log P(z_i|V)\right]$ is a little different from Section 6.3.2, since the mixing weights π are a function of V in the DP mixture model, $\pi_k = v_k \prod_{k'=1}^{k-1}(1-$

$v_{k'}$). We have

$$\begin{aligned}
&\mathbb{E}_q\left[\log P(z_i|V)\right] \\
&= \sum_{k=1}^{K} \eta_{i,k} E_q\left[\log \pi_k\right] \\
&= \sum_{k=1}^{K} \eta_{i,k} \left\{ \mathbb{E}_q\left[\log v_k\right] + \sum_{k'=1}^{k-1} \mathbb{E}_q[\log(1-v_{k'})] \right\} \\
&= \sum_{k=1}^{K} \eta_{i,k} \left\{ \Psi(\lambda_{k,1}) - \Psi(\lambda_{k,1}+\lambda_{k,2}) + \sum_{k'=1}^{k-1} \left[\Psi(\lambda_{k',2}) - \Psi(\lambda_{k',1}+\lambda_{k',2})\right] \right\}. \qquad (7.27)
\end{aligned}$$

The negative entropy term $\mathbb{E}_q\left[\log q(Z,V,\Theta^*)\right]$ is computed as:

$$\begin{aligned}
&\mathbb{E}_q\left[\sum_{k=1}^{K} \log q(v_k|\lambda_k) + \sum_{k=1}^{K} \log q(\theta_k^*|\tau_k) + \sum_{i=1}^{N} \log q(z_i|\eta_i)\right] \\
&= \sum_{k=1}^{K} \left\{ \log\Gamma(\lambda_{k,0}) - \sum_{s=1}^{2} \log\Gamma(\lambda_{k,s}) + \sum_{s=1}^{2} (\lambda_{k,s}-1)\left[\Psi(\lambda_{k,s}) - \Psi(\lambda_{k,0})\right] \right\} \\
&+ \sum_{k=1}^{K} \left\{ \log\Gamma(\tau_{k,0}) - \sum_{s=1}^{S} \log\Gamma(\tau_{k,s}) + \sum_{s=1}^{S} (\tau_{k,s}-1)\left[\Psi(\tau_{k,s}) - \Psi(\tau_{k,0})\right] \right\} \\
&+ \sum_{i=1}^{N} \sum_{k=1}^{K} \eta_{i,k} \log \eta_{i,k}. \qquad (7.28)
\end{aligned}$$

Where $\lambda_{k,0} = \lambda_{k,1} + \lambda_{k,2}$ and $\tau_{k,0} = \sum_{s=1}^{S} \tau_{k,s}$.

After computing each term in Equation 7.21, we now discuss optimization of the equation with respect to the variational parameters. It is obvious that the optimization problem is analytically intractable, thus again we consider the coordinate ascent algorithm as Section 6.3.2. It yields the following steps for the running example:

1. Randomly initialize the variational parameters. Note that the following constraints should be satisfied. λ_k is Beta parameter, thus $\lambda_{k,1} > 0$ and $\lambda_{k,2} > 0$. τ_k are the parameters of variational distribution $q(\theta_k^*|\tau_k)$ being of the same functional form as G_0. In the running example, G_0 is a Dirichlet distribution, thus τ_k are Dirichlet parameters and $\tau_{k,s} > 0$. η_i are multinomial parameters, thus $\eta_{i,k} \geq 0$ and $\sum_k \eta_{i,k} = 1$. In practice, we assume $\eta_{i,k}^{(0)} = 1/K$ for $i = 1, \ldots, N$ and $k = 1, \ldots, K$.

2. Repeat the following computation until convergence.

$$\lambda_{k,1} = 1 + \sum_{i=1}^{N} \eta_{i,k}, \quad \lambda_{k,2} = \alpha_0 + \sum_{i=1}^{N} \sum_{k'=k+1}^{K} \eta_{i,k'}, \tag{7.29a}$$

$$\tau_{k,s} = \frac{\beta_0}{S} + \sum_{i=1}^{N} \eta_{i,k} \delta_s(y_i), \tag{7.29b}$$

$$\eta_{i,k} \propto \exp\left(\mathbb{E}_q \left[\log v_k\right] + \sum_{k'=1}^{k-1} \mathbb{E}_q \left[\log(1 - v_{k'})\right] + \mathbb{E}_q \left[\log \theta^*_{k,y_i}\right] \right). \tag{7.29c}$$

Where $\delta_s(y_i)$ equals to 1 if $y_i = s$ and 0 otherwise. The convergence is monitored in an equivalent way like Section 6.3.2.

So far we have discussed three inference methods to approximate the posterior distribution of unobservable variables in the DP mixture model. Generally, mean-field method is much faster than the two GS methods, but GS methods provide more accurate prediction results. For more comparison of the three methods, please refer to (Blei & Jordan, 2005). In this chapter we assume that the hyperparameters, α_0 and G_0, are known, in some particular applications, the hyperparameters are unknown. For parameter learning in the DP mixture model, please refer to (McAuliffe et al., 2006), which introduced an empirical solution for the problem.

7.4 Predictive Inference

As finite mixture model, there are also two main prediction tasks in the DP mixture model, one is to predict a new observation $P(y_{new}|D, \alpha_0, G_0)$, the other is to predict the cluster assignment of a new observation $P(z_{new} = k|y_{new}, D, \alpha_0, G_0)$. They can be computed in the three inference methods introduced in the last section.

7.4.1 Collapsed Gibbs sampling

In collapsed GS method, the Markov chain consists of only the samples of hidden variables $Z^{(t)} = \{z_1^{(t)}, \ldots, z_N^{(t)}\}$. When the sequence converges, the predictive distributions are approximated over the samples. In particular, the first w members are discarded as *burn-in* period and the last W members of the sequence are collected to estimate the predictive distributions. Note, that the new observation y_i may be generated from a new mixture components in collapsed GS, thus z_{new} takes a value from $\{1, \ldots, K, K+1\}$. For the running example, the predictive probabilities are computed as:

$$\begin{aligned}
&P(y_{new}|D,\alpha_0,G_0) \\
&\approx \frac{1}{W}\sum_{t=w+1}^{W+w}\sum_{k=1}^{K+1}\int P(z_{new}=k|Z^{(t)},\alpha_0)P(y_{new}|\theta_k^*)P(\theta_k^*|Z^{(t)},D,G_0)d\theta_k^* \\
&=\frac{1}{W}\sum_{t=w+1}^{W+w}\sum_{k=1}^{K+1}P(z_{new}=k|Z^{(t)},\alpha_0)\mathbb{E}_{P(\theta_k^*|Z^{(t)},D,G_0)}\left[P(y_{new}|\theta_k^*)\right] \qquad (7.30a)
\end{aligned}$$

$$\begin{aligned}
&P(z_{new}=k|y_{new},D,\alpha_0,G_0) \\
&\propto \frac{1}{W}\sum_{t=w+1}^{W+w}\int P(z_{new}=k|Z^{(t)},\alpha_0)P(y_{new}|\theta_k^*)P(\theta_k^*|Z^{(t)},D,G_0)d\theta_k^* \\
&=\frac{1}{W}\sum_{t=w+1}^{W+w}P(z_{new}=k|Z^{(t)},\alpha_0)\mathbb{E}_{P(\theta_k^*|Z^{(t)},D,G_0)}\left[P(y_{new}|\theta_k^*)\right] \qquad (7.30b)
\end{aligned}$$

The terms in Equation 7.30 are defined as follows:

1. If $k = 1, \ldots, K$

$$P(z_{new}=k|Z^{(t)},\alpha_0)=\frac{N_k^{(t)}}{N+\alpha_0} \qquad (7.31a)$$

$$\mathbb{E}_{P(\theta_k^*|Z^{(t)},D,G_0)}\left[P(y_{new}|\theta_k^*)\right]=\frac{\frac{\beta_0}{S}+N^{(t)}(k,y_{new})}{\beta_0+N_k^{(t)}} \qquad (7.31b)$$

Where $N^{(t)}(k,s)$ is the *sufficient statistic* about observations at the iteration t, which denotes the number of observations with value s and hidden state k. $N_k^{(t)}$ denotes the number of observations with hidden state k at iteration t, $N_k^{(t)} = \sum_s N^{(t)}(k,s)$.

2. If $k = K + 1$

$$P(z_{new}=k|Z^{(t)},\alpha_0)=\frac{\alpha_0}{N+\alpha_0} \qquad (7.32a)$$

$$\mathbb{E}_{P(\theta_k^*|G_0)}\left[P(y_{new}|\theta_k^*)\right]=\frac{1}{S}. \qquad (7.32b)$$

7.4.2 Blocked Gibbs sampling

In analogy to collapsed GS method, the prediction inference in block GS method is much easier, since in each iteration the blocked sampler draws not only the hidden variables $Z^{(t)} = \{z_1^{(t)}, \ldots, z_N^{(t)}\}$, but also the mixture weights $\pi^{(t)}$ and mixture components $\Theta^{*(t)} =$

$\{\theta_1^{*(t)}, \ldots, \theta_K^{*(t)}\}$. For the running example, the predictive probabilities are computed as:

$$\begin{aligned}
P(y_{new}|D, \alpha_0, G_0) &\approx \frac{1}{W}\sum_{t=w+1}^{W+w}\sum_{k=1}^{K} P(z_{new}=k|\pi^{(t)})P(y_{new}|\theta_k^{*(t)}) \\
&= \frac{1}{W}\sum_{t=w+1}^{W+w}\sum_{k=1}^{K} \pi_k^{(t)}\theta_{k,y_{new}}^{*(t)} \qquad (7.33a) \\
P(z_{new}=k|y_{new}, D, \alpha_0, G_0) &\approx \frac{1}{W}\sum_{t=w+1}^{W+w} \frac{P(z_{new}=k|\pi^{(t)})P(y_{new}|\theta_k^{*(t)})}{\sum_k^K P(z_{new}=k|\pi^{(t)})P(y_{new}|\theta_k^{*(t)})} \\
&= \frac{1}{W}\sum_{t=w+1}^{W+w} \frac{\pi_k^{(t)}\theta_{k,y_{new}}^{*(t)}}{\sum_k^K \pi_k^{(t)}\theta_{k,y_{new}}^{*(t)}} \qquad (7.33b)
\end{aligned}$$

Note, that the predictive probabilities are averaged over K (not $K+1$) hidden states, since block GS method exploits truncated stick-breaking construction of DP, the new observation can only be generated from an existing mixture component, i.e. z_{new} takes a value from $\{1, \ldots, K\}$.

7.4.3 Mean field

When the updating process converges, mean-field inference method yields optimized variational parameters η, τ, λ, with which the variational distribution $q(Z, V, \Theta^*)$ closely approximates the posterior distribution $P(Z, V, \Theta^*|D, \alpha_0, G_0)$. Thus the prediction inference is implemented with the optimized distribution $q(Z, V, \Theta^*)$. For the running example, the predictive probabilities are computed as:

$$\begin{aligned}
&P(y_{new}|D, \alpha_0, G_0) \\
&= \sum_{k=1}^{K}\int P(y_{new}, z_{new}=k, \pi, \Theta^*|D, \alpha_0, G_0)d\pi d\Theta^* \\
&= \sum_{k=1}^{K}\int \pi_k P(\pi|D, \alpha_0, , G_0)d\pi \int P(y_{new}|\theta_k^*)P(\theta_k^*|D, \alpha_0, G_0)d\theta_k^* \\
&\approx \sum_{k=1}^{K}\int \pi_k q(\pi)d\pi \int P(y_{new}|\theta_k^*)q(\theta_k^*)d\theta_k^* \\
&= \sum_{k=1}^{K}\mathbb{E}_q(\pi_k)\mathbb{E}_q\left[P(y_{new}|\theta_k^*)\right]. \qquad (7.34)
\end{aligned}$$

The prediction of the hidden state for a new observation is computed as:

$$P(z_{new}=k|y_{new}, D, \alpha_0, G_0) \propto \mathbb{E}_q(\pi_k)\mathbb{E}_q\left[P(y_{new}|\theta_k^*)\right]. \qquad (7.35)$$

Let $\lambda_{k,0} = \lambda_{k,1} + \lambda_{k,2}$ and $\tau_{k,0} = \sum_{s=1}^{S} \tau_{k,s}$. Each term in Equation 7.34 and 7.35 is computed as follows.

$$\mathbb{E}_q(\pi_k) = \mathbb{E}_q \left[v_k \prod_{k'=1}^{k-1} (1 - v_{k'}) \right] = \frac{\lambda_{k,1}}{\lambda_{k,0}} \prod_{k'=1}^{k-1} \frac{\lambda_{k',2}}{\lambda_{k',0}}. \tag{7.36}$$

Since the observations are multinomial in the running example, we can compute the expectation $\mathbb{E}_q \left[P(y_{new}|\theta_k^*) \right]$ as:

$$\mathbb{E}_q \left[P(y_{new}|\theta_k^*) \right] = \mathbb{E}_q(\theta_{k,y_{new}}^*) = \frac{\tau_{k,y_{new}}}{\tau_{k,0}}. \tag{7.37}$$

In summary, the predictive probability is computed as:

$$P(y_{new}|D, \alpha_0, G_0) \approx \sum_{k=1}^{K} \frac{\tau_{k,y_{new}} \lambda_{k,1}}{\tau_{k,0} \lambda_{k,0}} \prod_{k'=1}^{k-1} \frac{\lambda_{k',2}}{\lambda_{k',0}} \tag{7.38}$$

$$P(z_{new} = k|y_{new}, D, \alpha_0, G_0) \propto \frac{\tau_{k,y_{new}} \lambda_{k,1}}{\tau_{k,0} \lambda_{k,0}} \prod_{k'=1}^{k-1} \frac{\lambda_{k',2}}{\lambda_{k',0}}. \tag{7.39}$$

7.5 Summary

In this chapter we introduce Dirichlet process mixture model, which extends the flexibility of the finite mixture model by encoding the uncertainty about the number of mixture components in an elegant way. The DP mixture model assumes that the parameters, one for each observation, share a common prior G drawn from a Dirichlet process $DP(\alpha_0, G_0)$. The model can be viewed as an extension of the nonparametric hierarchical Bayesian model introduced in Chapter 4 by focusing on the discreteness property of random distribution G. The advantage of the nonparametric mixture model is that the number of states of the hidden variables can be optimized by the model itself based on the complexity of the data and can increase when additional data is available.

In the next chapter, we will discuss the applications of DP mixture model in relational learning. Typically, structural model selection in a relational system is extensive due to the exponentially many attributes an attribute might depend on. To solve the problem, we apply DP mixture modeling to relational data, such that the attributes and relations only depend on the corresponding hidden variables, i.e. the cluster assignments. Although the probability structures and parameter dependencies are specified in advance, the learned models are still flexible enough to approximate the posterior distributions of variables of interest by propagating information in the whole ground network defined by the relational structure.

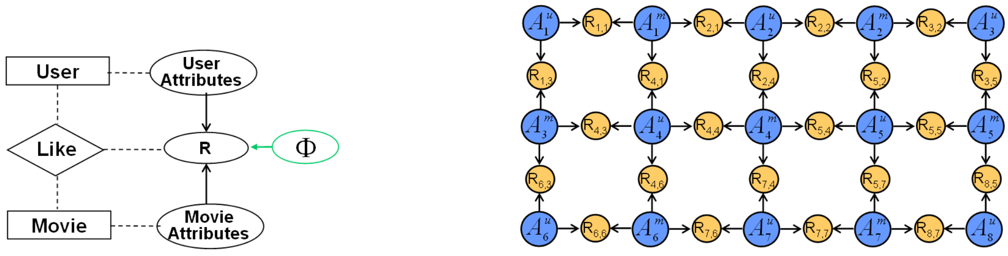
Chapter 8

Infinite Hidden Relational Models

8.1 Introduction

Relational learning is an object oriented approach that clearly distinguishes between objects, attributes and relationships. The learned dependencies encode probabilistic constraints in the relational domain. A simple example of a relational system is a movie recommendation system. There are two entity classes (User and Movie) and one relationship class (Like: whether a user likes a movie). User class has attribute classes, e.g. age, gender, occupation and so on. Movie class has attribute classes, e.g. genre, year and so on. Like class has attribute class R, which can be yes/no, or rating (to which extent a user likes a movie). Based on the attributes of the two entities, i.e. of the user and the movie, a recommendation system wants to predict the relationship attribute R. Figure 8.1 shows a relational model in a directed acyclic probabilistic entity relationship (DAPER) representation (Heckerman et al., 2004), which is our preferred representation of relational models. The table at the bottom of Figure 8.1 lists known relationships between 8 users and 7 movies. Figure 8.1(c) shows the ground Bayesian network given the model and the relationships. It is clear that entity attributes locally predict the probability of a relationship attribute. Whether a user likes a movie is decided by the user and movie attributes. Thus given the parent attributes all relational attributes are independent. That means the known ratings from the user of interest do not influence the prediction of future ratings of the user. To solve the limitation, structural learning might be involved to obtain non-local dependency. But structural model selection should be extensive due to the exponentially many attributes an attribute might depend on.

From this point of view it can make sense to introduce hidden variables representing unknown attributes of the entities, e.g. the preference of users in the movie recommendation example. Entity attributes are now children of hidden variables of the corresponding entities and relationship attributes are children of hidden variables of the entities participating in the relationships. Since the central importance the hidden variables are, we refer it to *hidden relational model* (HRM). In HRM, the ground Bayesian network forms a network of hidden variables via the relational structure. It can be viewed on as a direct generalization of hidden Markov model used in speech recognition or hidden Markov random field used in computer vision (Yedidia et al., 2005). As in those models, information


User
User Attributes
Like
R
Φ
Movie
Movie Attributes
A^u_1 $R_{1,1}$ A^m_1 $R_{2,1}$ A^u_2 $R_{2,2}$ A^m_2 $R_{3,2}$ A^u_3
$R_{1,3}$ $R_{4,1}$ $R_{2,4}$ $R_{5,2}$ $R_{3,5}$
A^m_3 $R_{4,3}$ A^u_4 $R_{4,4}$ A^m_4 $R_{5,4}$ A^u_5 $R_{5,5}$ A^m_5
$R_{6,3}$ $R_{4,6}$ $R_{7,4}$ $R_{5,7}$ $R_{8,5}$
A^u_6 $R_{6,6}$ A^m_6 $R_{7,6}$ A^u_7 $R_{7,7}$ A^m_7 $R_{8,7}$ A^u_8

(a) (c)

Like											
User ID	Movie ID	R	User ID	Movie ID	R	User ID	Movie ID	R	User ID	Movie ID	R
U_1	M_1	*Yes*	U_3	M_5	*No*	U_5	M_4	*Yes*	U_7	M_6	*Yes*
U_1	M_3	*No*	U_4	M_1	*No*	U_5	M_5	*No*	U_7	M_7	*No*
U_2	M_1	*Yes*	U_4	M_3	*Yes*	U_5	M_7	*Yes*	U_8	M_5	*No*
U_2	M_2	*Yes*	U_4	M_4	*Yes*	U_6	M_3	*No*	U_8	M_7	*Yes*
U_2	M_4	*No*	U_4	M_6	*Yes*	U_6	M_6	*Yes*			
U_3	M_2	*Yes*	U_5	M_2	*No*	U_7	M_4	*Yes*			

(b)

Figure 8.1: (a) A relational model on the movie recommendation system.(b) A table about known relationships between 8 users and 7 movies. (c) Ground Bayesian network applying the relational model (a) on the data set (b).

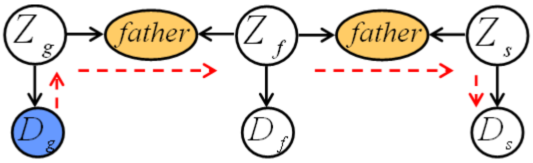


Figure 8.2: A ground Bayesian network about heart condition in a family. The information about heart condition of grandfather propagates to the son via the hidden variables. The dashed line specifies how the information flows to the variable of interest.

can propagate across the network of hidden variables. Figure 8.2 gives an example of how information propagates. The fact that a person's grandfather had a heart condition is reflected in his hidden variable, which then influences the hidden variable of her father (who might not have a heart condition) which influences her own hidden variable, which then changes the probability for her obtaining a heart condition. HRM can also be interpreted as a relational mixture model, which provides clustering effect for the entities in a natural way. The cluster assignments of entities depend not only on the entity attributes, but also on the relationships between entities. It can be viewed on as a generalization of co-clustering model (Hofmann & Puzicha, 1999).

Since each entity class might have the different number of states in its hidden variable, it is natural to allow the model to determine the appropriate number of hidden states in a self-organized way. This is possible by embedding HRM in Dirichlet process (DP) mixture model, which can be interpreted as a mixture model with an infinite number of mixture components but where the model, based on the data, automatically reduces the complexity to an appropriate finite number of components. The combination of the hidden relational model and the DP mixture model is the *infinite hidden relational model* (IHRM), which can be viewed as an extension of nonparametric hierarchical Bayesian modeling to relational data. The difference from the Dirichlet enhanced relational model introduced in Chapter 5 is that IHRM focuses on the discreteness property of DP and incorporates the relationships into the probabilistic model by existence uncertainty mechanism (Getoor et al., 2003). IHRM is one of the main contributions in the thesis and was published in (Xu et al., 2006) and (Xu et al., 2007).

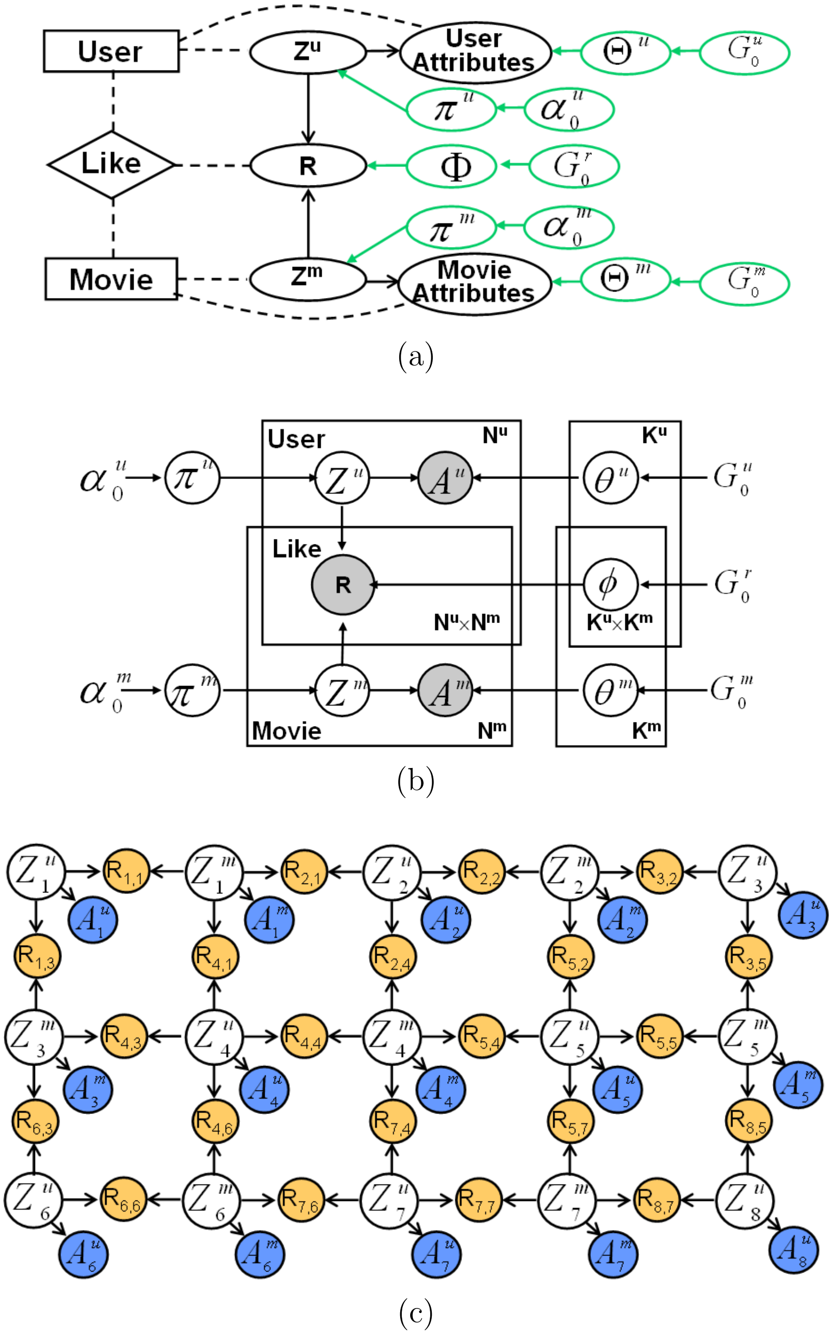
As in other relational models, inference in IHRM is executed in a large interconnected ground network. Thus being able to perform efficient inference is critical for the success of the model. To solve the problem, we propose collapsed Gibbs sampling, blocked Gibbs sampling, mean-field method and an empirical approximation, which can be viewed as an relational extension of the inference methods in (Escobar & West, 1995; Escobar & West, 1998; Ishwaran & James, 2001; Blei & Jordan, 2005). For experimental analysis, we apply IHRM model in three domains, including: the medical recommendation system, the movie recommendation system and the function prediction of genes. The promising results demonstrate the performance of IHRM.

8.2 Model Description

Infinite hidden relational model (IHRM) is a new development in the thesis which tends to set up a general and effective framework to model the relational data. An IHRM can be viewed as a template, which specifies the probabilistic dependencies and distributions for types of entities and relationships. Given an IHRM and instantiated entities and relationships, a ground Bayesian network is formed, over which the probabilities of variables of interest can be inferred. In this section, we first introduce the finite hidden relational model (HRM) and then extend it to infinite version (IHRM), at last we provide two generative models describing how to generate data given an IHRM.

8.2.1 Hidden Relational Model

Figure 8.3 shows a HRM on the movie recommendation system. (a) and (b) describe the model in the DAPER and plate representations, respectively. The first innovation of HRM is to introduce for each entity a hidden variable, in the example denoted as Z^u and Z^m. They can be thought of as unknown attributes of the entities and are the parents of both the entity attributes and the relationship attributes. The underlying assumption is that if the hidden variable was known, both entity attributes and relationship attributes can be well predicted. The most important result from introducing the hidden variables is that now information can propagate through the ground Bayesian network via interconnected hidden variables. For example, given a ground Bayesian network of HRM shown as Figure 8.3(c), let us consider a predictive inference about the relationship attribute, say $R_{2,7}$ between user 2 and movie 7. The probability is calculated on the evidence about (1)



(a)

(b)

(c)

Figure 8.3: Hidden relational model (HRM) for movie recommendation system. (a) DAPER representation. (b) Plate representation. (c) Ground Bayesian network given users, movies and relationships in Figure 8.1(b).

the attributes of the immediately related entities, i.e. of user 2 and movie 7, (2) the other relationships associated with the entities, i.e. the ratings $R_{2,1}$, $R_{2,2}$, $R_{2,4}$ from the user 2 and the ratings $R_{5,7}$, $R_{7,7}$, $R_{8,7}$ about the movie 7, (3) *high-order* information transferred via the hidden variables in the ground Bayesian network, e.g., the information about A^u_4 and $R_{4,*}$ propagated through Z^u_4 and Z^m_4. Via collecting more evidence, HRM potentially provides more accurate prediction results than the traditional relational models. From the figure, it is clear that the hidden states of the entities are decided not only by their attributes, but also by their relationships. If both the associated users and movies have strong known attributes, those will determine the states of the hidden variables and the prediction for relationship attribute $R_{i,j}$ is mostly based on the entity attributes. In terms of a recommender-system terminology we would obtain a content-based recommendation system. Conversely, if the known attributes are weak, then the states of the hidden variables for the users might be determined by the relationships to other movies and the states of those movies' hidden variables. With the same argument, the states of the hidden variables for the movies might be determined by the relationships to other users and the states of those users' hidden variables. Again in terms of a recommender-system terminology we would obtain a (item-based) collaborative-filtering system. As an extra advantage, HRM provides an elegant way to combine the content-based recommendation methods with collaborative-filtering methods.

In summary, by introducing the hidden variables, information can globally flow in the ground Bayesian network defined by the relationship structure. This reduces the need for extensive structural learning, which is particularly difficult in relational models due to the huge number of potential parents. Note that a similar propagation of information can be observed in hidden Markov models used in speech recognition or in the hidden Markov random fields used in image analysis (Yedidia et al., 2005). In fact the HRM can be viewed as a direct generalization of both models for relational data. Additionally, the HRM naturally provides clustering effect as a mixture model. The assignments of hidden variables Z^u and Z^m specify the clusters of the corresponding entities.

We now complete the model by introducing the variables. First we consider the variables in User class. There is a hidden variable Z^u_i with K^u states for each user. The assignment $Z^u_i = k$ specifies the mixture component of the user i. The mixing weights $\pi^u = (\pi^u_1, \ldots, \pi^u_{K^u})$ are multinomial parameters with $P(Z^u = k) = \pi^u_k$ ($\pi^u_k > 0, \sum_k \pi^u_k = 1$) and are drawn from a conjugated Dirichlet prior, $\pi^u \sim \mathrm{Dir}(\cdot|\alpha^u_0, \alpha^u_1, \ldots, \alpha^u_{K^u})$. $\alpha^u_k > 0$, $\sum_{k=1}^{K^u} \alpha^u_k = 1$. α^u_k represents our prior expectation about the probability of a user taking hidden state k. In practice, we can assume a neutral prior with $\alpha^u_k = 1/K^u$, which represents our prior belief in the fact that the probabilities of hidden states should be equal. $\alpha^u_0 > 0$ is a confidence parameter indicating how strongly we believe that the prior brief represented by α^u_k should be true. The larger the value is, the stronger our belief is.

All user attributes are assumed to be discrete and independent given Z^u. Thus a particular user attribute A^u_i with S states is a sample from a multinomial distribution conditioned on Z^u_i, $P(A^u_i = s|Z^u_i = k) = \theta^u_{k,s}$ ($\theta^u_{k,s} > 0, \sum_s \theta^u_{k,s} = 1$) and

$$(\theta^u_{k,1}, \ldots, \theta^u_{k,S}) \sim G^u_0 = \mathrm{Dir}(\cdot|\beta^u_0, \beta^u_1, \ldots, \beta^u_S). \tag{8.1}$$

$\sum_{s=1}^{S} \beta^u_s = 1$, $\beta^u_s > 0$. Again β^u_s represents our prior expectation about the probability of user attribute. $\beta^u_0 > 0$ is a confidence parameter about our prior brief β^u. The parameters for the entity class Movie are defined in an equivalent way.

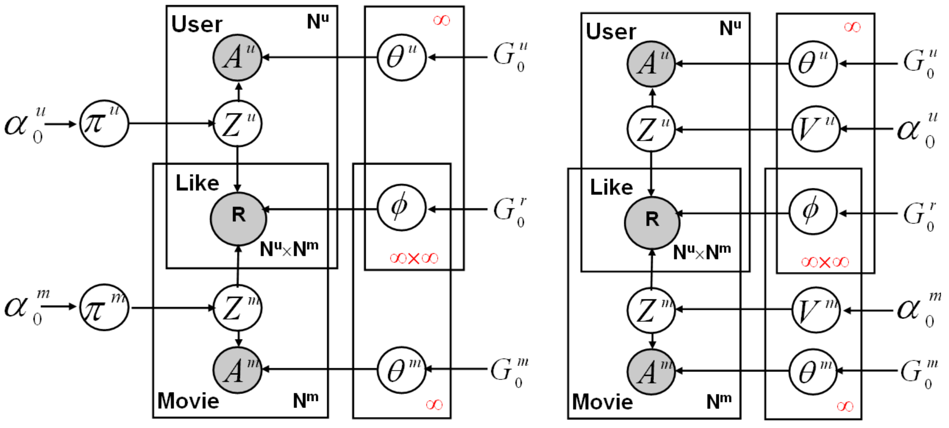
We now consider the variables in the relationship class. In the running example, the relationship attribute R is assumed to be discrete with S^r states. A particular relationship is a sample drawn from a multinomial distribution conditioned on Z^u and Z^m, $P(R_{i,j} = s|Z^u_i = k, Z^m_j = \ell) = \phi_{k,\ell,s}$ $(\phi_{k,\ell,s} > 0, \sum_s \phi_{k,\ell,s} = 1)$. Note, that there are $K^u \times K^m$ parameters $\phi_{k,\ell}$ and they are drawn:

$$(\phi_{k,\ell,1}, \ldots, \phi_{k,\ell,S^r}) \sim G^r_0 = \mathrm{Dir}(\cdot|\beta^r_0, \beta^r_1, \ldots, \beta^r_{S^r}), \tag{8.2}$$

where $\beta^r_s > 0$, $\sum_{s=1}^{S^r} \beta^r_s = 1$ and $\beta^r_0 > 0$. If the entity attribute, resp. relationship attribute is continuous, we only need to assume the prior G^u_0 resp. G^r_0 a suitable form, e.g. a Gaussian distribution.

From mixture model point of view, the most interesting term in HRM is $\phi_{k,\ell}$, which can be interpreted as a *correlation mixture component*. It makes the two distinct mixture systems coupled. If a user i is assigned to a cluster k, i.e. $Z^u_i = k$, then he inherits not only θ^u_k, but also $\phi_{k,\ell}, \ell = \{1, \ldots, K^m\}$. If a new user cluster is generated, then a new mixture component $\theta^c_{K^c+1}$ will be sampled, and a set of new correlation mixture components will accordingly be sampled , $\phi_{K^u+1,\ell}, \ell = \{1, \ldots, K^m\}$.

8.2.2 Infinite Hidden Relational Model



(a) (b)

Figure 8.4: Infinite hidden relational model (IHRM) for movie recommendation system in the plate representation. (a) and (b) describe the same model, but (b) explicitly specifies how to generate the mixing weights π via the *stick breaking construction*. The DAPER representation of IHRM looks the same as the finite hidden relational model in Figure 8.3(a). However, the definitions of variables are different.

The hidden variables play a key role in HRM, we would expect that HRM might require a large number of states for the hidden variables. Consider again the movie recommendation system. With little information about past ratings all users might look the same (movies are globally liked or disliked), with more information available, one might discover certain clusters in the users (action movie aficionados, comedy aficionados, ...) but with an increasing number of past ratings the clusters might show increasingly detailed structure ultimately indicating that everyone is an individual. It thus makes sense to permit an arbitrary number of hidden states by integrating with a Dirichlet process mixture model. This permits the model to decide itself about the optimal number of hidden states for each entity class. For our discussion it suffices to say that we obtain an infinite hidden relational model by simply letting the number of hidden states approach infinity, $K^u \to \infty$, $K^m \to \infty$. Figure 8.4 shows the infinite model in the plate representation. The DAPER representation of IHRM looks the same as the finite HRM in Figure 8.3. However, the definitions of variables are different. For example, the hidden variable Z_i^u has infinite states, and thus there are infinite mixture component θ_k^u. The mixing weights π^u are also infinite dimensional, which are generated not from a Dirichlet prior, but from a *stick breaking construction* $\text{Stick}(\cdot|\alpha_0^u)$ (more details in the next section). Although a model with the infinite number of states and parameters cannot be represented, sampling in such model is elegant and simple (see the next section). In the relational Dirichlet process mixture model, α_0^u, resp. α_0^m determines the tendency of the model to either use a large number or a small number of states in the hidden variables, which is apparent from the sampling procedures described below. If α_0^u, resp. α_0^m, is chosen to be small, only few clusters are generated and the parameters tend to be highly coupled. If α_0^u, resp. α_0^m, is chosen to be large, the coupling is loose and more clusters are generated. From Figure 8.4, it is clear that we introduce multiple DPs, one for each entity class, and these DPs are coupled together by relationship attributes R and correlation mixture components $\phi_{k,\ell}$.

8.2.3 Generative Models

Now we describe the generative models for IHRM. There are mainly two common methods to generate samples from a Dirichlet process (DP) mixture model, i.e., Chinese restaurant process (CRP) (Aldous, 1985) and stick breaking construction (SBC) (Sethuraman, 1994). We will introduce how to extend them to IHRM. To describe the generative models, we need some notation. Let the number of entity classes be C, and let G_0^c and α_0^c denote the base distribution and concentration parameter for entity class c. In an entity class c, there are N^c entities e_i^c indexed by i, and K^c mixture components θ_k^c indexed by k. θ_k^c denotes the parameters of distribution of the entity attributes. The number of relationship classes is denoted by B. G_0^b denotes the base distribution of a relationship class b. In order to avoid a cluttering of notation, we only describe relationships between two entity classes. The generalization to relationships involving multiple entity classes is straightforward. For a relationship class b between two entity classes c_i and c_j, there are $K^{c_i} \times K^{c_j}$ correlation mixture components $\phi_{k,\ell}^b$ indexed by hidden states k for c_i and ℓ for c_j. $\phi_{k,\ell}^b$ denotes the parameters of distribution of relationship attributes. Here we restrict ourselves that the entity and relationship attributes are drawn from exponential family distributions with parameters θ_k^c and $\phi_{k,\ell}^b$, respectively. The base distributions G_0^c and G_0^b are the conjugate priors with the hyperparameters β^c and β^b. In the following sections, we discuss the computation based on these assumptions, for computation in more complex situations, e.g. non-conjugated base distributions, please refer to (Liu, 1996; MacEachern et al., 1999; Newton & Zhang, 1999; Quintana & Newton, 2000; Blei & Jordan, 2005).

Generative Model with Chinese Restaurant Process

Chinese restaurant process (CRP) (Aldous, 1985) integrates out the random distributions of parameters and directly draws the underlying samples sequentially. Extending CRP to the IHRM, the future entities are generated on previously sampled entities. In detail:

1. Initialization:

   (a) The first entity e_1^c in each of C entity classes is assigned to the first cluster, the first mixture component θ_1^c is sampled, then the entity attribute is drawn.

   $$Z_1^c = 1; \quad \theta_1^c \sim G_0^c; \quad A_1^c \sim P(\cdot|\theta_1^c). \tag{8.3}$$

   (b) For each relationship class b between entity classes c and c', the correlation mixture component $\phi_{1,1}^b$ is drawn from G_0^b. And then the relationship attribute $R_{1,1}^b$ between the entities e_1^c and $e_1^{c'}$ is drawn:

   $$\phi_{1,1}^b \sim G_0^b; \quad R_{1,1}^b \sim P(\cdot|\phi_{1,1}^b). \tag{8.4}$$

2. Iteration: for a new entity e_i^c in the entity class c:

   (a) Assign the new entity to an existing cluster $Z_i^c = k$ with probability $N_k^c/(N^c + \alpha_0^c)$. The entity inherits all parameters associated with the cluster k, then its attribute and relationships are sampled:

   $$A_i^c \sim P(\cdot|\theta_k^c); \quad R_{i,j}^{b'} \sim P(\cdot|\Phi^{b'}, Z_i^c, Z_j^{c_j}), \tag{8.5}$$

   where N_k^c denotes the number of entities in class c with hidden state k. b' denotes a relationship class involving entity class c. j denotes the index of an entity possibly having a relationship of class b' with the entity e_i^c.

   (b) Instead, the new entity is assigned a new cluster with probability $\alpha_0^c/(N^c + \alpha_0^c)$ and accordingly new parameters are sampled, conditioned on which the attributes and relationships are sampled for the new entity:

   $$\theta_{K^c+1}^c \sim G_0^c; \quad \phi_{K^c+1,\ell}^{b'} \sim G_0^{b'}, \tag{8.6a}$$
   $$A_i^c \sim P(\cdot|\theta_{K^c+1}^c); \quad R_{i,j}^{b'} \sim P(\cdot|\Phi^{b'}, Z_i^c, Z_j^{c_j}). \tag{8.6b}$$

   Where ℓ denotes the hidden state of the entity class involved in the relationship class b', $\ell = \{1, \ldots, K^{c_j}\}$. Then $K^c \leftarrow K^c + 1$.

   (c) $i \leftarrow i + 1$, go to sample the next entity.

Generative Model with Stick Breaking Construction

The stick breaking construction (SBC) (Sethuraman, 1994) is a representation of DP, by which we can explicitly samples the distributions of attribute parameters and relationship parameters. Following we describe the generative model of IHRM in terms of SBC.

1. For each entity class c,

   (a) Draw mixing weights $\pi^c \sim \text{Stick}(\cdot|\alpha_0^c)$, where $\text{Stick}(\cdot|\alpha_0)$ denotes the stick breaking construction defined as

   $$V_k^c \overset{\text{i.i.d.}}{\sim} \text{Beta}(1, \alpha_0^c); \quad \pi_1^c = V_1^c, \quad \pi_k^c = V_k^c \prod_{k'=1}^{k-1}(1 - V_{k'}^c),\ k > 1. \tag{8.7}$$

   (b) Draw i.i.d. mixture components $\theta_k^c \sim G_0^c$, $k = 1, 2, \ldots$

2. For each relationship class b between two entity classes c_i and c_j, draw $\phi_{k,\ell}^b \overset{\text{i.i.d.}}{\sim} G_0^b$ with component indices k for c_i and ℓ for c_j.

3. For each entity e_i^c,

   (a) Draw cluster assignment $Z_i^c \sim \text{Mult}(\cdot|\pi^c)$;

   (b) Draw entity attributes $A_i^c \sim P(\cdot|\Theta^c, Z_i^c)$.

4. For entities $e_i^{c_i}$ and $e_j^{c_j}$ with a relationship of class b, draw

   $$R_{i,j}^b \sim P(\cdot|\Phi^b, Z_i^{c_i}, Z_j^{c_j}). \tag{8.8}$$

8.3 Inference

Markov chain Monte Carlo (MCMC) sampling methods have been used to approximate posterior distribution with a DP mixture prior. In the section, we extend these MCMC methods to IHRM. We first introduce a Gibbs sampler with the Chinese restaurant process (CRP), which is a collapsed version of Pólya urn sampling. Unfortunately CRP sampler exhibits slow mixing of the Markov chain. Blocked sampling typically shows better mixing. Thus we extend the efficient blocked Gibbs sampling with truncated stick breaking construction (TSB) to IHRM. These MCMC inference are motivated by (Escobar & West, 1995; Escobar & West, 1998; Ishwaran & James, 2001; Blei & Jordan, 2005).

While an unquestioned success, MCMC sampling methods are not expected techniques due to the low efficiency. In particular, IHRM has multiple DPs that interact through the relationships, the exchange of information between DPs is slow in the two Gibbs samplers, thus it needs more time to reach stationary points. To solve the constraint, we explore an alternative solution by variational inference method, which is motivated by (Blei & Jordan, 2005). The method assumes a specific form for the posterior and maximizes the lower bound of log-likelihood via coordinate ascent algorithm.

Additionally we propose an empirical approximation for the inference of IHRM, which can be viewed as an extension of memory-based inference method to relational data.

8.3.1 Collapsed Gibbs Sampling with CRP

The collapsed Gibbs sampling (GS) integrates out the posterior distributions of parameters of entity attributes and relationship attributes. The Markov chain is thus defined only on the hidden variables of all entities. The collapsed GS iteratively samples each hidden variable Z_i^c, for $c = \{1, \ldots, C\}$ and $i = \{1, \ldots, N^c\}$, conditioned on the other hidden variables Z_{-i} until the procedure converges. In particular, Z_i^c is updated as:

1. The entity e_i^c is assigned to an existing cluster $Z_i^c = k$ and inherits the parameters assigned to component k with probability proportional to

$$N_k^c \, P(A_i^c|A^c, Z_i^c = k, Z_{-i}^c, G_0^c) \prod_{b'} \prod_{j'} P(R_{i,j'}^{b'}|R^{b'}, Z_i^c = k, Z_{-i}, G_0^{b'}), \tag{8.9}$$

   where b' denotes a relationship class involving the entity class c. A^c and $R^{b'}$ denote the known attributes of entity class c and relationships of class b'. j' denotes an entity having an known relationship of class b' with the entity e_i^c. Note, that for a binary case, the *nonexistent* relationships can also be considered, e.g., the relationships that a user *dislikes* some movies in the running example.

2. Instead, the entity is assigned to a new cluster with probability proportional to

$$\alpha_0^c \, P(A_i^c|G_0^c) \prod_{b'} \prod_{j'} P(R_{i,j'}^{b'}|G_0^{b'}) \tag{8.10}$$

In Equation 8.9 and Equation 8.10, we use the following definitions:

$$P(A_i^c|A^c, Z_i^c = k, Z_{-i}^c, G_0^c) = \mathbb{E}_{P(\theta_k^c|A^c, Z_{-i}^c, G_0^c)} \left[P(A_i^c|\theta_k^c)\right], \tag{8.11}$$

$$P(R_{i,j'}^{b'}|R^{b'}, Z_i^c = k, Z_{-i}, G_0^{b'}) = \mathbb{E}_{P(\phi_{k,\ell'}^{b'}|R^{b'}, Z_{-i}, G_0^{b'})} \left[P(R_{i,j'}^{b'}|\phi_{k,\ell'}^{b'})\right], \tag{8.12}$$

$$P(A_i^c|G_0^c) = \mathbb{E}_{P(\theta_{new}^c|G_0^c)} \left[P(A_i^c|\theta_{new}^c)\right], \tag{8.13}$$

$$P(R_{i,j'}^{b'}|G_0^{b'}) = \mathbb{E}_{P(\phi_{new,\ell'}^{b'}|G_0^{b'})} \left[P(R_{i,j'}^{b'}|\phi_{new,\ell'}^{b'})\right]. \tag{8.14}$$

Where ℓ' denotes the component assignment of the entity j'. The first two equations are the posterior expectations of the probabilities of attributes and relationships of the entity e_i^c conditioned on the samples of hidden variables of other entities. The last two equations are the prior expectations of the corresponding quantities. Since we assume conjugated distributions as the base distributions G_0^c and G_0^b, the computation can be implemented analytically.

From the update steps, one can see that the update of Z_i^c is conditioned on both the hidden variables Z_{-i}^c of the same entity class and the hidden variables in the related entity classes. Via the relationships, the DPs are coupled together. When the process converges, the predictive distributes are approximated as an average across the Monte Carlo samples. The details are described in Section 8.4.

8.3.2 Blocked Gibbs Sampling with Truncated SBC

In the collapsed GS, the hidden variables are updated one at a time, which potentially slows down the method. For computational efficiency, we extend the blocked GS (Ishwaran & James, 2001) (see also Chapter 7) to IHRM. In the method, the posterior distributions of parameters of entity attributes and relationship attributes are explicitly sampled in the form of truncated stick breaking construction (TSB). The advantage is that given the posterior, we can independently sample the hidden variables in a block, which highly accelerates the computation. The Markov chain is thus defined not only on the hidden variables Z_i^c, but also the parameters, including π^c, Θ^c and Φ^b, for $c = \{1, \dots, C\}$, $i = \{1, \dots, N^c\}$ and $b = \{1, \dots, B\}$. Note, that there are additional parameters K^c in block GS, which specify the positions to truncate the DPs. In practice, we set K^c as the number of entities in class c, K^c will be automatically reduced to a suitable value based on the data in the sampling process. Taking some initial values for the unobservable variables Z, π^c, Θ^c and Φ^b, the following steps are repeated until convergence:

1. For each entity class c,

   (a) Update hidden variable Z_i^c for each entity i independently:

$$P(Z_i^c = k | D_i^c, Z_{-i}, \pi^c, \Theta^c, \{\Phi^{b'}\}_{b'=1}^{B'}) \\ \propto \pi_k^c P(A_i^c | Z_i^c = k, \Theta^c) \prod_{b'} \prod_{j'} P(R_{i,j'}^{b'} | Z_i^c = k, Z_{j'}^{c_{j'}}, \Phi^{b'}). \qquad (8.15)$$

   Where D_i^c denotes all information about the entity i, including attributes and relationships. $c_{j'}$ denotes the class of the entity j'. $Z_{j'}^{c_{j'}}$ denotes the component assignment of j'.

   (b) Update π^c as follows:

      i. Sample v_k^c independently from Beta$(\lambda_{k,1}^c, \lambda_{k,2}^c)$ for $k = \{1, \dots, K^c - 1\}$ with

$$\lambda_{k,1}^c = 1 + \sum_{i=1}^{N^c} \delta_k(Z_i^c), \quad \lambda_{k,2}^c = \alpha_0^c + \sum_{k'=k+1}^{K^c} \sum_{i=1}^{N^c} \delta_{k'}(Z_i^c), \qquad (8.16)$$

      and set $v_{K^c}^c = 1$. Where $\delta_k(Z_i^c)$ equals to 1 if $Z_i^c = k$ and 0 otherwise.

      ii. Compute $\pi_1^c = v_1^c$, $\pi_k^c = v_k^c \prod_{k'=1}^{k-1} (1 - v_{k'}^c), k > 1$.

2. Update the parameters

$$\theta_k^c \sim P(\cdot | A^c, Z^c, G_0^c), \qquad \phi_{k,\ell}^b \sim P(\cdot | R^b, Z, G_0^b). \qquad (8.17)$$

   The parameters are drawn from their posteriors based on the sampled hidden states. Again, since we assume conjugated priors as the base distributions G_0^c and G_b^b, the simulation can be implemented cheaply. For more complex situations, e.g. when it is difficult to draw samples from the posterior, please refer to (Escobar & West, 1998; MacEachern & Mueller, 1998).

8.3.3 Mean Field with Truncated SBC

Blei and Jordan (2005) introduced a variational inference method for nonparametric Bayesian mixture model, which approximates the posterior of unobserved variables using a factorized distribution for all the stick lengths and component parameters (see also Chapter 7). We now extend it to IHRM. The unobservable variables in IHRM include Z^c, V^c, Θ^c and Φ^b. The posterior distribution is $P(\{Z^c, V^c, \Theta^c\}_{c=1}^C, \{\Phi^b\}_{b=1}^B | D, \{\alpha_0^c, G_0^c\}_{c=1}^C, \{G_0^b\}_{b=1}^B)$, which is obviously untractable. In the variational inference framework, we first define a variational distribution $q(\{Z^c, V^c, \Theta^c\}_{c=1}^C, \{\Phi^b\}_{b=1}^B | \xi)$ with variational parameters ξ, then we minimize Kullback-Leibler (KL) divergence between $q(\{Z^c, V^c, \Theta^c\}_{c=1}^C, \{\Phi^b\}_{b=1}^B | \xi)$ and true posterior distribution with respect to the variational parameters. As a result, the variational distribution with optimized parameters ξ is an approximation to the true posterior distribution. In IHRM, the KL divergence is defined as:

$$\begin{aligned}&\log P(D|\{\alpha_0^c, G_0^c\}_{c=1}^C, \{G_0^b\}_{b=1}^B) + \mathbb{E}_q[\log q(\{Z^c, V^c, \Theta^c\}_{c=1}^C, \{\Phi^b\}_{b=1}^B | \xi)]\\ &- \mathbb{E}_q[\log P(\{Z^c, V^c, \Theta^c\}_{c=1}^C, \{\Phi^b\}_{b=1}^B, D|\{\alpha_0^c, G_0^c\}_{c=1}^C, \{G_0^b\}_{b=1}^B)].\end{aligned} \tag{8.18}$$

The minimization of KL divergence can be cast as the maximization of a lower bound $\mathcal{L}$ of the log likelihood of the data:

$$\begin{aligned}&\log P(D|\{\alpha_0^c, G_0^c\}_{c=1}^C, \{G_0^b\}_{b=1}^B)\\ &\geq \mathbb{E}_q[\log P(\{Z^c, V^c, \Theta^c\}_{c=1}^C, \{\Phi^b\}_{b=1}^B, D|\{\alpha_0^c, G_0^c\}_{c=1}^C, \{G_0^b\}_{b=1}^B)]\\ &- \mathbb{E}_q[\log q(\{Z^c, V^c, \Theta^c\}_{c=1}^C, \{\Phi^b\}_{b=1}^B | \xi)].\end{aligned} \tag{8.19}$$

The Equation 8.19 can also be derived from *Jensen's inequality.* For the computational efficiency, we choose a family of fully-factorized distributions breaking all dependencies between the unobservable variables:

$$q(\{Z^c, V^c, \Theta^c\}_{c=1}^C, \{\Phi^b\}_{b=1}^B) = \left[\prod_c^C \prod_i^{N^c} q(Z_i^c|\eta_i^c) \prod_k^{K^c} q(V_k^c|\lambda_k^c) q(\theta_k^c|\tau_k^c)\right] \left[\prod_b^B \prod_k^{K^{c_i}} \prod_\ell^{K^{c_j}} q(\phi_{k,\ell}^b|\rho_{k,\ell}^b)\right]. \tag{8.20}$$

Where c_i and c_j denote the entity classes involved in the relationship class b. k and ℓ denote the component indices for c_i and c_j. Variational parameters are $\xi = \{\eta_i^c, \lambda_k^c, \tau_k^c, \rho_{k,\ell}^b\}$. Note, that there are one η_i^c for each entity e_i^c, one λ_k^c and one τ_k^c for each entity component, one $\rho_{k,\ell}^b$ for each correlation component. $q(Z_i^c|\eta_i^c)$ is a multinomial distribution. $q(V_k^c|\lambda_k^c)$ is a Beta distribution. $q(\theta_k^c|\tau_k^c)$ and $q(\phi_{k,\ell}^b|\rho_{k,\ell}^b)$ are the distributions with the same form as G_0^c and G_0^b, respectively. Figure 8.5 illustrates the variational distribution via the running example. It is clear that some probabilistic dependencies between these variables are removed, which can be found via comparing with Figure 8.4(b).

Given the variational distribution as Equation 8.20, the lower bound of the log likeli-

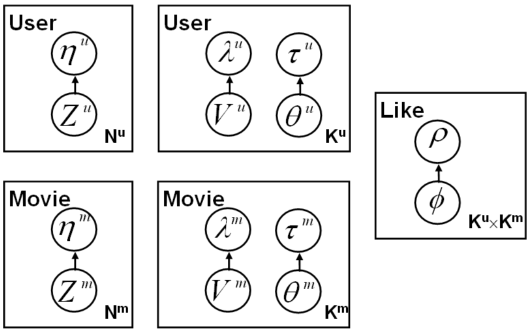


Figure 8.5: Graphical representation of the variational distribution for the movie recommendation system.

hood in Equation 8.19 is computed as:

$$\begin{aligned}
\mathcal{L} = & \sum_{c}^{C}\sum_{k}^{K^c}\Big\{\mathbb{E}_q[\log P(V_k^c|\alpha_0^c)] + \mathbb{E}_q[\log P(\theta_k^c|G_0^c)]\Big\} + \sum_{b}^{B}\sum_{k}^{K^{c_i}}\sum_{\ell}^{K^{c_j}}\Big\{\mathbb{E}_q[\log P(\phi_{k,\ell}^b|G_0^b)]\Big\} \\
& + \sum_{c}^{C}\sum_{i}^{N^c}\Big\{\mathbb{E}_q[\log P(Z_i^c|V^c)]\Big\} \\
& + \sum_{c}^{C}\sum_{i}^{N^c}\Big\{\mathbb{E}_q[\log P(A_i^c|Z_i^c,\theta^c)]\Big\} + \sum_{b}^{B}\sum_{i}^{N^{c_i}}\sum_{j}^{N^{c_j}}\Big\{\mathbb{E}_q[\log P(R_{i,j}^b|Z,\phi^b)]\Big\} \\
& - \mathbb{E}_q[\log q(\{Z^c,V^c,\Theta^c\}_{c=1}^{C},\{\Phi^b\}_{b=1}^{B})]
\end{aligned} \tag{8.21}$$

The first line includes the terms related to the parameters $\{\Phi^b\}_{b=1}^B$ and $\{V^c,\Theta^c\}_{c=1}^C$. The terms in the second line is related to the hidden variables $\{Z^c\}_{c=1}^C$. The terms about the observed attributes and relationships are in the third line, where the first sum is about entity attributes, the second sum is about relationships. The expectation in the last line is the entropy term.

Now we discuss the computation of Equation 8.21. Some terms, such as $\mathbb{E}_q[\log P(V_k^c|\alpha_0^c)]$, $\mathbb{E}_q[\log P(\theta_k^c|G_0^c)]$ and $\mathbb{E}_q[\log P(Z_i^c|V^c)]$ as well as $\mathbb{E}_q[\log P(A_i^c|Z_i^c,\Theta^c)]$, involve standard computation in Section 7.3.3. The terms $\mathbb{E}_q[\log P(\phi_{k,\ell}^b|G_0^b)]$ about relationship parameters are computed in an equivalent way like the terms $\mathbb{E}_q[\log P(\theta_k^c|G_0^c)]$ about entity attribute parameters. The terms $\mathbb{E}_q[\log P(R_{i,j}^b|Z,\Phi^b)]$ about relationships are computed in a different way, since all involved variables $Z_i^{c_i}$, $Z_j^{c_j}$ and Φ^b are unobservable.

$$\begin{aligned}
\mathbb{E}_q[\log P(R_{i,j}^b|Z,\phi^b)] &= \sum_{k}^{K^{c_i}}\sum_{\ell}^{K^{c_j}}\int q(Z_i^{c_i}=k|\eta_i^{c_i})q(Z_j^{c_j}=\ell|\eta_j^{c_j})q(\phi_{k,\ell}^b|\rho_{k,\ell}^b)\log P(R_{i,j}^b|\phi_{k,\ell}^b)d\phi_{k,\ell}^b \\
&= \sum_{k}^{K^{c_i}}\sum_{\ell}^{K^{c_j}}\eta_{i,k}^{c_i}\eta_{j,\ell}^{c_j}\mathbb{E}_q[\log P(R_{i,j}^b|\phi_{k,\ell}^b)]
\end{aligned} \tag{8.22}$$

After computing each term in Equation 8.21, we optimize the lower bound in a coordinate ascent algorithm, which yields the following update steps:

1. Randomly initialize the variational parameters η_i^c with the constraints $\sum_k \eta_{i,k}^c = 1$.

2. Iteratively update the variational parameters until convergence.

$$\lambda_{k,1}^c = 1 + \sum_{i=1}^{N^c} \eta_{i,k}^c, \qquad \lambda_{k,2}^c = \alpha_0^c + \sum_{i=1}^{N^c} \sum_{k'=k+1}^{K^c} \eta_{i,k'}^c, \tag{8.23}$$

$$\tau_{k,1}^c = \beta_1^c + \sum_{i=1}^{N^c} \eta_{i,k}^c \mathrm{T}(A_i^c), \qquad \tau_{k,2}^c = \beta_2^c + \sum_{i=1}^{N^c} \eta_{i,k}^c, \tag{8.24}$$

$$\rho_{k,\ell,1}^b = \beta_1^b + \sum_{i,j} \eta_{i,k}^{c_i} \eta_{j,\ell}^{c_j}\ \mathrm{T}(R_{i,j}^b), \qquad \rho_{k,\ell,2}^b = \beta_2^b + \sum_{i,j} \eta_{i,k}^{c_i} \eta_{j,\ell}^{c_j}, \tag{8.25}$$

$$\eta_{i,k}^c \propto \exp\left(\mathbb{E}_q[\log V_k^c] + \sum_{k'=1}^{k-1} \mathbb{E}_q[\log(1 - V_{k'}^c)] + \mathbb{E}_q[\log P(A_i^c|\theta_k^c)] \right.$$
$$\left. + \sum_b \sum_j \sum_\ell \eta_{j,\ell}^{c_j} \mathbb{E}_q[\log P(R_{i,j}^b|\phi_{k,\ell}^b)] \right). \tag{8.26}$$

Where λ_k^c denotes parameters of Beta distribution $q(V_k^c|\lambda_k^c)$, thus λ_k^c is a two-dimensional vector. τ_k^c denotes parameters of exponential family distributions $q(\theta_k^c|\tau_k^c)$. We decompose τ_k^c such that $\tau_{k,1}^c$ contains the first $dim(\theta_k^c)$ components and $\tau_{k,2}^c$ is a scalar. $\rho_{k,\ell,1}^b$ and $\rho_{k,\ell,2}^b$ are defined equivalently. $\mathrm{T}(A_i^c)$ denotes the *sufficient statistic* of the exponential family distribution $P(A_i^c|\theta_k^c)$. It is clear that Equation 8.23 and Equation 8.24 correspond to the updates for variational parameters of entity class c, and they follow equations in (Blei & Jordan, 2005). Equation 8.25 represents the updates of variational parameters of relationships, which is computed on the involved entities. The most interesting updates are Equation 8.26, where the posteriors of entity assignment are *coupled together*. These essentially connect the DPs together. As can be understood intuitively, in Equation 8.26 the posterior updates for $\eta_{i,k}^c$ include a prior term (first two expectations), the likelihood term of entity attributes (third expectation), and the likelihood terms of relationships (last term). To calculate the last term we need to sum over all the relationships of the entity e_i^c weighted by $\eta_{j,\ell}^{c_j}$ of assignments of the other entities involved in the relationships. At convergence all updates in Equation 8.26 will not change the posterior assignments.

8.3.4 Empirical Approximation

In this section, a very basic inference method is proposed for IHRM. For each entity class, we assume the number of mixture components to be the number of entities. Thus each entity is assumed to only contribute to its own class. Based on this simplification the parameters for the attributes and relationships can be learned very efficiently. Note that this approximation can be interpreted as a relational memory-based learning method. The corresponding predictive inference is introduced in Section 8.4.

8.4 Predictive Inference

One key predictive inference in relational learning is that of predicting the relationships between entities of interest. In this section, we will illustrate the predictive computation with the running example. The following three situations are considered:

- Situation 1: both the user and the movie of interest are in the training set.
- Situation 2: the movie exists in the training set. The user is new, but his information D^u_{new}, including attribute A^u_{new} and previous ratings $R_{new,*}$, is available. $R_{new,j'}$ denotes a known rating between the new user and a known movie j'.
- Situation 3: both the user and the movie are new, the information (D^u_{new}, D^m_{new}) about new user and new movie is available, including user attribute A^u_{new}, movie attribute A^m_{new}, and user ratings $R_{new,*}$, move ratings $R_{*,new}$. $R_{i',new}$ denotes a known rating between a known user i' and the new movie.

The predictive distribution $P(R_{i,j} = s|A^u, A^m, R, \alpha^u_0, \alpha^m_0, G^u_0, G^m_0, G^r_0)$ will be computed in the four inference methods introduced in Section 8.3.

8.4.1 Collapsed Gibbs Sampling with CRP

At each iteration of the collapsed GS method, the hidden variables Z^u, Z^m are sampled for all users and movies in training data set. After the sampling procedure reaches stationary, the predictive distribution is approximated over the samples $Z^{u(t)}$ and $Z^{m(t)}$.

$$\begin{aligned}&P(R_{i,j} = s|A^u, A^m, R, \alpha^u_0, \alpha^m_0, G^u_0, G^m_0, G^r_0)\\ \approx\ &\frac{1}{W}\sum_{t=w+1}^{W+w} P(R_{i,j} = s|A^u, A^m, R, Z^{u(t)}, Z^{m(t)}, \alpha^u_0, \alpha^m_0, G^u_0, G^m_0, G^r_0). \qquad (8.27)\end{aligned}$$

Where the first w members of the MCMC sequence are discarded as *burn-in* period, the last W members are stored to approximate the predictive distribution. In each situation, $P(R_{i,j} = s|A^u, A^m, R, Z^{u(t)}, Z^{m(t)}, \alpha^u_0, \alpha^m_0, G^u_0, G^m_0, G^r_0)$ is computed in different way.

- Situation 1:

$$\begin{aligned}&P(R_{i,j} = s|A^u, A^m, R, Z^{u(t)}, Z^{m(t)}, \alpha^u_0, \alpha^m_0, G^u_0, G^m_0, G^r_0)\\ =\ &P(R_{i,j} = s|R, Z^{u(t)}, Z^{m(t)}, G^r_0)\\ =\ &\int P(R_{i,j} = s, \phi^{(t)}_{k^*,\ell^*}|R, Z^{u(t)}, Z^{m(t)}, G^r_0)\ d\phi^{(t)}_{k^*,\ell^*}\\ =\ &\int \phi^{(t)}_{k^*,\ell^*,s} P(\phi^{(t)}_{k^*,\ell^*}|R, Z^{u(t)}, Z^{m(t)}, G^r_0)\ d\phi^{(t)}_{k^*,\ell^*} \equiv \mathbb{E}_{\hat{P}}\left[\phi^{(t)}_{k^*,\ell^*,s}\right]. \qquad (8.28)\end{aligned}$$

Where k^* and ℓ^* denote the component assignments of the user i and the movie j at the iteration t. $\hat{P}$ denotes the posterior of relationship parameters, $\hat{P} = P(\phi^{(t)}_{k^*,\ell^*}|R, Z^{u(t)}, Z^{m(t)}, G^r_0)$, which is still a Dirichlet distribution with parameters

$$\beta^{r(t)}_{post} = (\beta^r_0 \times \beta^r_1 + N^{r(t)}(k^*, \ell^*, 1), \ldots, \beta^r_0 \times \beta^r_S + N^{r(t)}(k^*, \ell^*, S)).$$

$N^{r(t)}(k^*, \ell^*, s)$ is a *sufficient statistic* about the relationships at the iteration t, which is the number of relationships with the value s, user-component assignment k^* and movie-component assignment ℓ^*. $N^{r(t)}(k^*, \ell^*) = \sum_s N^{r(t)}(k^*, \ell^*, s)$. We have:

$$\begin{aligned} &P(R_{i,j} = s | A^u, A^m, R, Z^{u(t)}, Z^{m(t)}, \alpha_0^u, \alpha_0^m, G_0^u, G_0^m, G_0^r) \\ &= \mathbb{E}_{\hat{P}}\left[\phi^{(t)}_{k^*,\ell^*,s}\right] = \frac{\beta_0^r \times \beta_s^r + N^{r(t)}(k^*, \ell^*, s)}{\beta_0^r + N^{r(t)}(k^*, \ell^*)}. \end{aligned} \tag{8.29}$$

- Situation 2:

$$\begin{aligned} &P(R_{i,j} = s | D^u_{new}, A^u, A^m, R, Z^{u(t)}, Z^{m(t)}, \alpha_0^u, \alpha_0^m, G_0^u, G_0^m, G_0^r) \\ &= P(R_{new,j} = s | D^u_{new}, A^u, R, Z^{u(t)}, Z^{m(t)}, \alpha_0^u, G_0^u, G_0^r) \\ &\propto \sum_{k=1}^{K^u+1} P(R_{new,j} = s | R, Z^u_{new} = k, Z^{u(t)}, Z^{m(t)}, G_0^r) P(Z^u_{new} = k | Z^{u(t)}, \alpha_0^u) \\ &\times P(A^u_{new} | A^u, Z^u_{new} = k, Z^{u(t)}, G_0^u) \prod_{j'} P(R_{new,j'} | R, Z^u_{new} = k, Z^{u(t)}, Z^{m(t)}, G_0^r). \end{aligned} \tag{8.30}$$

Note, that in collapsed GS, a new entity might be assigned into a new cluster, thus the predictive probability is averaged over $K^u + 1$ clusters. The terms in Equation 8.30 are computed as:

1. If $k = 1, \ldots, K^u$

$$P(Z^u_{new} = k | Z^{u(t)}, \alpha_0^u) = \frac{N^{u(t)}(k)}{\alpha_0^u + N^u}, \tag{8.31a}$$

$$\begin{aligned} &P(A^u_{new} | A^u, Z^u_{new} = k, Z^{u(t)}, G_0^u) \\ &= \mathbb{E}_{P(\theta_k^{u(t)} | A^u, Z^{u(t)}, G_0^u)}\left[\theta^{u(t)}_{k,s_{new}}\right] = \frac{\beta_0^u \times \beta^u_{s_{new}} + N^{u(t)}(k, s_{new})}{\beta_0^u + N^{u(t)}(k)}, \end{aligned} \tag{8.31b}$$

$$\begin{aligned} &P(R_{new,j'} | R, Z^u_{new} = k, Z^{u(t)}, Z^{m(t)}, G_0^r) \\ &= \mathbb{E}_{P(\phi^{(t)}_{k,\ell'} | R, Z^{u(t)}, Z^{m(t)}, G_0^r)}\left[\phi^{(t)}_{k,\ell',s'}\right] = \frac{\beta_0^r \times \beta_{s'}^r + N^{r(t)}(k, \ell', s')}{\beta_0^r + N^{r(t)}(k, \ell')}. \end{aligned} \tag{8.31c}$$

Where s_{new} and s' denote the values of A^u_{new} and $R_{new,j'}$, respectively. ℓ' denotes the component assignment of the movie j'. $N^{u(t)}(k, s)$ is a *sufficient statistic* about user attributes at the iteration t, which represents the number of users with attribute value s and component assignment k, $N^{u(t)}(k) = \sum_s N^{u(t)}(k, s)$.

2. If $k = K^u + 1$

$$P(Z^u_{new} = k | Z^{u(t)}, \alpha^u_0) = \frac{\alpha^u_0}{\alpha^u_0 + N^u}, \tag{8.32a}$$

$$P(A^u_{new} | A^u, Z^u_{new} = k, Z^{u(t)}, G^u_0) = \mathbb{E}_{P(\theta^{u(t)}_k | G^u_0)}\left[\theta^{u(t)}_{k,s_{new}}\right] = \beta^u_{s_{new}}, \tag{8.32b}$$

$$P(R_{new,j'} | R, Z^u_{new} = k, Z^{u(t)}, Z^{m(t)}, G^r_0) = \mathbb{E}_{P(\phi^{(t)}_{k,\ell'} | G^r_0)}\left[\phi^{(t)}_{k,\ell',s'}\right] = \beta^r_{s'}. \tag{8.32c}$$

- Situation 3:

$$\begin{aligned}
&P(R_{new,new} = s | D^u_{new}, D^m_{new}, A^u, A^m, R, Z^{u(t)}, Z^{m(t)}, \alpha^u_0, \alpha^m_0, G^u_0, G^m_0, G^r_0) \\
&\propto \sum_{k=1}^{K^u+1} \sum_{\ell=1}^{K^m+1} P(R_{new,new} = s | R, Z^u_{new} = k, Z^m_{new} = \ell, Z^{u(t)}, Z^{m(t)}, G^r_0) \\
&\times P(Z^u_{new} = k | Z^{u(t)}, \alpha^u_0) P(A^u_{new} | A^u, Z^u_{new} = k, Z^{u(t)}, G^u_0) \\
&\times \prod_{j'} P(R_{new,j'} | R, Z^u_{new} = k, Z^{u(t)}, Z^{m(t)}, G^r_0) \\
&\times P(Z^m_{new} = \ell | Z^{m(t)}, \alpha^m_0) P(A^m_{new} | A^m, Z^m_{new} = \ell, Z^{m(t)}, G^m_0) \\
&\times \prod_{i'} P(R_{i',new} | R, Z^m_{new} = \ell, Z^{u(t)}, Z^{m(t)}, G^r_0).
\end{aligned} \tag{8.33}$$

Where the terms about users are computed as Equation 8.31 and 8.32. The corresponding terms about movies are computed equivalently.

8.4.2 Blocked Gibbs Sampling with Truncated SBC

At each iteration of block GS, we sample not only hidden variables Z^u, Z^m for all users and movies in training data set, but also the parameters π^u, π^m, Θ^u, Θ^m, Φ. Thus the MCMC sequence is defined by the mixture assignments and the mixture components. After the procedure of GS reaches stationary point, the probability of interest is approximated over the sampled values, including $Z^{u(t)}$, $Z^{m(t)}$, $\pi^{u(t)}$, $\pi^{m(t)}$, $\Theta^{u(t)}$, $\Theta^{m(t)}$, $\Phi^{(t)}$.

$$\begin{aligned}
&P(R_{i,j} = s | A^u, A^m, R, \alpha^u_0, \alpha^m_0, G^u_0, G^m_0, G^r_0) \\
&\approx \frac{1}{W} \sum_{t=w+1}^{W+w} P(R_{i,j} = s | Z^{u(t)}, Z^{m(t)}, \pi^{u(t)}, \pi^{m(t)}, \theta^{u(t)}, \theta^{m(t)}, \phi^{(t)}).
\end{aligned} \tag{8.34}$$

Where W and w are defined as Section 8.4.1. In each situation, we compute the distribution $P(R_{i,j} = s | Z^{u(t)}, Z^{m(t)}, \pi^{u(t)}, \pi^{m(t)}, \theta^{u(t)}, \theta^{m(t)}, \phi^{(t)})$ in different way. In detail:

- Situation 1:

$$P(R_{i,j} = s | Z^{u(t)}, Z^{m(t)}, \pi^{u(t)}, \pi^{m(t)}, \theta^{u(t)}, \theta^{m(t)}, \phi^{(t)}) = \phi^{(t)}_{k^*,\ell^*,s} \tag{8.35}$$

- Situation 2:

$$
\begin{aligned}
&P(R_{i,j} = s|D^u_{new}, Z^{u(t)}, Z^{m(t)}, \pi^{u(t)}, \pi^{m(t)}, \theta^{u(t)}, \theta^{m(t)}, \phi^{(t)}) \\
&\propto \sum_{k=1}^{K^u} \Big[P(R_{new,j} = s|Z^u_{new} = k, Z^{m(t)}_j, \phi^{(t)}) P(Z^u_{new} = k|\pi^{u(t)}) \\
&\times P(A^u_{new}|Z^u_{new} = k, \theta^{u(t)}) \prod_{j'} P(R_{new,j'}|Z^u_{new} = k, Z^{m(t)}_{j'}, \phi^{(t)}) \Big] \\
&= \sum_{k=1}^{K^u} \Big[\phi^{(t)}_{k,\ell^*,s} \pi^{u(t)}_k \theta^{u(t)}_{k,s_{new}} \prod_{j'} \phi^{(t)}_{k,\ell',s'} \Big]
\end{aligned} \tag{8.36}
$$

- Situation 3:

$$
\begin{aligned}
&P(R_{new,new} = s|D^u_{new}, D^m_{new}, Z^{u(t)}, Z^{m(t)}, \pi^{u(t)}, \pi^{m(t)}, \theta^{u(t)}, \theta^{m(t)}, \phi^{(t)}) \\
&\propto \sum_{k=1}^{K^u} \sum_{\ell=1}^{K^m} \Big[P(R_{new,new} = s|Z^u_{new} = k, Z^m_{new} = \ell, \phi^{(t)}) \\
&\times P(Z^u_{new} = k|\pi^{u(t)}) P(A^u_{new}|\theta^{u(t)}_k) \prod_{j'} P(R_{new,j'}|\phi^{(t)}_{k,\ell'}) \\
&\times P(Z^m_{new} = \ell|\pi^{m(t)}) P(A^m_{new}|\theta^{m(t)}_\ell) \prod_{i'} P(R_{i',new}|\phi^{(t)}_{k',\ell}) \Big] \\
&= \sum_{k=1}^{K^u} \sum_{\ell=1}^{K^m} \Big[\phi^{(t)}_{k,\ell,s} \pi^{u(t)}_k \theta^{u(t)}_{k,s_{new}} \prod_{j'} \phi^{(t)}_{k,\ell',s'} \pi^{m(t)}_\ell \theta^{m(t)}_{\ell,s_{new}} \prod_{i'} \phi^{(t)}_{k',\ell,s'} \Big]
\end{aligned} \tag{8.37}
$$

8.4.3 Mean Field with Truncated SBC

Mean field inference method minimizes the KL-divergence between the variational distribution and the posterior with respect to the variational parameters. At the convergence point, we obtain the optimized variational distribution with parameters λ^u, λ^m, η^u, η^m, τ^u, τ^m and ρ, by which the predictive distribution $P(R_{i,j} = s|A^u, A^m, R, \alpha^u_0, \alpha^m_0, G^u_0, G^m_0, G^r_0)$ is approximated.

- Situation 1:

$$
\begin{aligned}
&P(R_{i,j} = s|D^u_i, D^m_j, \eta^u, \eta^m, \tau^u, \tau^m, \rho) \\
&\propto \sum_{k=1}^{K^u} \sum_{\ell=1}^{K^m} \Big[P(R_{i,j} = s|\rho_{k,\ell}) \\
&\times P(Z^u_i = k|\eta^u_i) P(A^u_i|\tau^u_k) \prod_{j'} \sum_{\ell'} \eta^m_{j',\ell'} P(R_{i,j'}|\rho_{k,\ell'}) \\
&\times P(Z^m_j = \ell|\eta^m_j) P(A^m_j|\tau^m_\ell) \prod_{i'} \sum_{k'} \eta^u_{i',k'} P(R_{i',j}|\rho_{k',\ell}) \Big].
\end{aligned} \tag{8.38}
$$

- Situation 2:

$$\begin{aligned}
&P(R_{new,j} = s | D^u_{new}, D^m_j, \lambda^u, \eta^m, \tau^u, \tau^m, \rho) \\
\propto &\sum_{k=1}^{K^u} \sum_{\ell=1}^{K^m} P(R_{new,j} = s | \rho_{k,\ell}) \\
\times &P(Z^u_{new} = k | \lambda^u) P(A^u_{new} | \tau^u_k) \prod_{j'} \sum_{\ell'} \eta^m_{j',\ell'} P(R_{new,j'} | \rho_{k,\ell'}) \\
\times &P(Z^m_j = \ell | \eta^m_j) P(A^m_j | \tau^m_\ell) \prod_{i'} \sum_{k'} \eta^u_{i',k'} P(R_{i',j} | \rho_{k',\ell}). \qquad (8.39)
\end{aligned}$$

- Situation 3:

$$\begin{aligned}
&P(R_{new,new} = s | D^u_{new}, D^m_{new}, \lambda^u, \lambda^m, \tau^u, \tau^m, \rho) \\
\propto &\sum_{k=1}^{K^u} \sum_{\ell=1}^{K^m} P(R_{new,new} = s | \rho_{k,\ell}) \\
\times &P(Z^u_{new} = k | \lambda^u) P(A^u_{new} | \tau^u_k) \prod_{j'} \sum_{\ell'} \eta^m_{j',\ell'} P(R_{new,j'} | \rho_{k,\ell'}) \\
\times &P(Z^m_{new} = \ell | \lambda^m) P(A^m_{new} | \tau^m_\ell) \prod_{i'} \sum_{k'} \eta^u_{i',k'} P(R_{i',new} | \rho_{k',\ell}). \qquad (8.40)
\end{aligned}$$

The main difference of the computations in the three situations is how to estimate the probability of component assignment of the entity of interest. If the entity of interest, e.g. a user i, exists in the training data set, the variational parameters η^u_i is available, thus the probability can be represented as $P(Z^u_i = k | \eta^u_i)$; if the user is a new one, the probability will be represented as $P(Z^u_{new} = k | \lambda^u)$, which is conditioned on the variational parameters λ^u for the stick lengths. The terms in above equations are computed as:

$$P(Z^u_i = k | \eta^u_i) = \eta^u_{i,k} \qquad (8.41a)$$

$$P(Z^u_{new} = k | \lambda^u) = \mathbb{E}_{q(V^u | \lambda^u)} \left[V^u_k \prod_{k'=1}^{k-1} (1 - V^u_{k'}) \right] \qquad (8.41b)$$

$$P(A^u_i = s | \tau^u_k) = \mathbb{E}_{q(\theta^u_{k,s} | \tau^u_k)} \left[\theta^u_{k,s} \right] \qquad (8.41c)$$

$$P(R_{i,j} = s' | \rho_{k,\ell}) = \mathbb{E}_{q(\phi_{k,\ell,s'} | \rho_{k,\ell})} \left[\phi_{k,\ell,s} \right] \qquad (8.41d)$$

The corresponding terms about movies are computed in an equivalent way.

8.4.4 Empirical Approximation

In empirical approximation, each user/movie in training data is assigned to its own cluster, thus we compute the predictive probability directly. In detail:

- Situation 1:

$$
\begin{aligned}
&P(R_{i,j}=s|A^u,A^m,R)\\
&\propto \sum_{k\neq i}^{N^u}\sum_{\ell\neq j}^{N^m} P(R_{i,j}=s|Z_i^u=k,Z_j^m=\ell,R)\\
&\times P(Z_i^u=k|Z^u,\alpha_0^u)P(A_i^u|Z_i^u=k,A^u)\prod_{j'}P(R_{i,j'}|Z_i^u=k,R)\\
&\times P(Z_j^m=\ell|Z^m,\alpha_0^m)P(A_j^m|Z_j^m=\ell,A^m)\prod_{i'}P(R_{i',j}|Z_j^m=\ell,R).
\end{aligned}
$$

- Situation 2:

$$
\begin{aligned}
&P(R_{new,j}=s|D_{new}^u,A^u,A^m,R)\\
&\propto \sum_{k=1}^{N^u}\sum_{\ell\neq j}^{N^m} P(R_{new,j}=s|Z_{new}^u=k,Z_j^m=\ell,R)\\
&\times P(Z_{new}^u=k|Z^u,\alpha_0^u)P(A_{new}^u|Z_{new}^u=k,A^u)\prod_{j'}P(R_{new,j'}|Z_{new}^u=k,R)\\
&\times P(Z_j^m=\ell|Z^m,\alpha_0^m)P(A_j^m|Z_j^m=\ell,A^m)\prod_{i'}P(R_{i',j}|Z_j^m=\ell,R).
\end{aligned}
$$

- Situation 3:

$$
\begin{aligned}
&P(R_{new,new}=s|D_{new}^u,D_{new}^m,A^u,A^m,R)\\
&\propto \sum_{k=1}^{N^u}\sum_{\ell=1}^{N^m} P(R_{new,new}=s|Z_{new}^u=k,Z_{new}^m=\ell,R)\\
&\times P(Z_{new}^u=k|Z^u,\alpha_0^u)P(A_{new}^u|Z_{new}^u=k,A^u)\prod_{j'}P(R_{new,j'}|Z_{new}^u=k,R)\\
&\times P(Z_{new}^m=\ell|Z^m,\alpha_0^m)P(A_{new}^m|Z_{new}^m=\ell,A^m)\prod_{i'}P(R_{i',new}|Z_{new}^m=\ell,R).
\end{aligned}
$$

The computations in the three situations are the same except for the possible mixture components for the entity of interest. For a new user, Z_{new}^u can be any value between 1 and N^u. However, if the user of interest is in the training data set, then $Z_i^u\neq i$. The terms in above equations are computed as follows:

1. Since each training entity is of its own cluster, thus we have

$$P(Z_i^u=k|Z^u,\alpha_0^u)=\frac{1}{N^u-1};\quad P(Z_{new}^u=k|Z^u,\alpha_0^u)=\frac{1}{N^u}.$$

2. $P(A^u_i|Z^u_i = k, A^u)$ is computed over memory-based naive Bayes;

$$P(A^u_i|Z^u_i = k, A^u) = \frac{\beta^u_0 \times \beta^u_{s*} + 1}{\beta^u_0 + 1},$$

   if the attributes of the user i and the user k are the same; otherwise,

$$P(A^u_i|Z^u_i = k, A^u) = \frac{\beta^u_0 \times \beta^u_{s*}}{\beta^u_0 + 1}.$$

   Where $s*$ denotes the value of A^u_i.

3. $P(R_{i,j'}|Z^u_i = k, R)$ is also computed over memory-based naive Bayes,

$$P(R_{i,j'}|Z^u_i = k, R) = \frac{\beta^r_0 \times \beta^r_{s*} + 1}{\beta^r_0 + 1},$$

   if the user i and the user k give the same ratings to the movie j'; otherwise

$$P(R_{i,j'}|Z^u_i = k, R) = \frac{\beta^r_0 \times \beta^r_{s*}}{\beta^r_0 + 1}.$$

   Where $s*$ denotes the value of $R_{i,j'}$.

4. Similarly, we compute

$$P(R_{i,j} = s|Z^u_i = k, Z^m_j = \ell, R) = \frac{\beta^r_0 \times \beta^r_s + 1}{\beta^r_0 + 1},$$

   if the user k gives the movie ℓ a rating s; otherwise

$$P(R_{i,j} = s|Z^u_i = k, Z^m_j = \ell, R) = \frac{\beta^r_0 \times \beta^r_s}{\beta^r_0 + 1}.$$

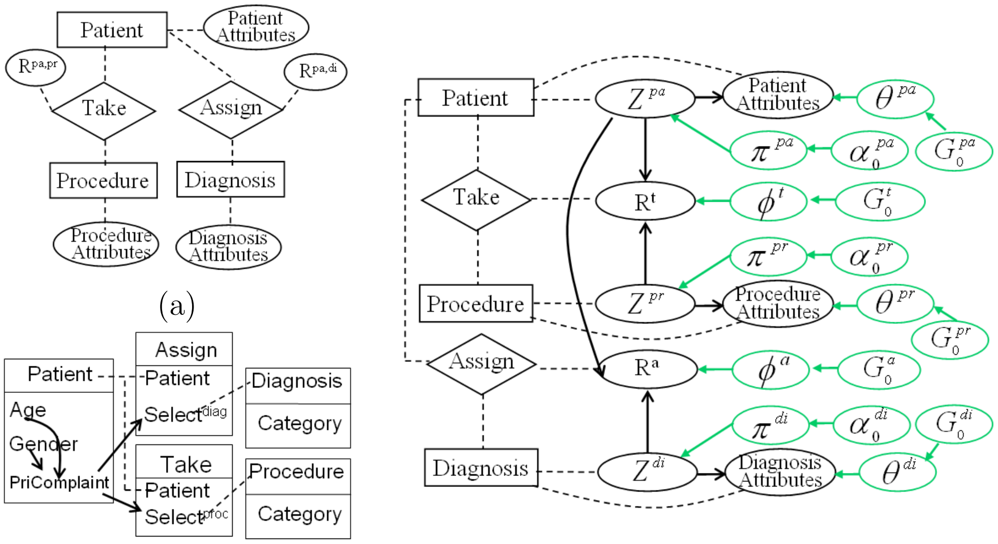
5. The corresponding terms about movies are computed equivalently.

It is clear that the predictive computation of the empirical approximation (EA) method scales in the product of the number of users and the number of movies in the training data. When the training data is large enough, the EA method might be much slower than the other inference methods. Mean field method might be an expected solution that balances the computational time and prediction accuracy for an online system.

8.5 Experimental Analysis

8.5.1 Clinical Data Analysis

The first experiment is executed on a clinical database. The target of the experiment is to demonstrate the performance of IHRM with the most simplest inference method: empirical approximation (EA). The structure of the clinical database is shown as Figure 8.6(a)



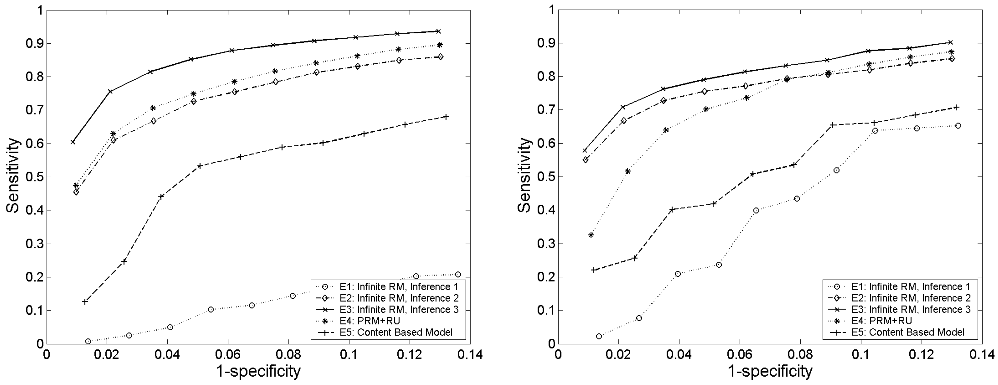
(b) (c)

Figure 8.6: (a) A clinical database represented by entity-relationship model. (b) PRM model for the clinical database. (c) Infinite hidden relational model.

with entity-relationship representation. The domain includes three entity classes (Patient, Diagnosis and Procedure) and two relationship classes (Assign: patients are assigned diagnoses. Take: patients take procedures). A patient typically has multiple procedures and multiple diagnoses. Patient class has several attribute classes, including Age, Gender, PrimeComplaint. To reduce the complexity of Figure 8.6, patient attributes are grouped together as PatientAttributes (these attributes are not aggregated in learning and inference). The DiagnosisAttributes contain the category of the diagnosis as specified in the ICD-9 code and the ProcedureAttributes contain the category of the procedure as specified in the CPT4 code. The relationships between the patients and the procedures and the relationships between the patients and the diagnoses are modeled as existence uncertainty. $R^t = 1$ if the patient takes the procedure and $R^t = 0$ otherwise. Equivalently, $R^a = 1$ if the patient is assigned the diagnosis and $R^a = 0$ otherwise. In the data, there are totally 14062 patients, 703 diagnoses and 367 procedures.

The infinite hidden relational model is shown in Figure 8.6(c). It contains three DPs, one for each entity class. We compare IHRM with two models. The first one is a relational model with reference uncertainty (Getoor et al., 2003) but without a hidden variable structure. The model is shown as Figure 8.6(b), where each relationship class is associated with an auxiliary variable *Select*. The value of *Select* specifies which procedure, resp. diagnosis is chosen for a patient. In addition, the variable *Select* conditions on *Patient.PrimeComplaint*. The parameters in the model are global, which means that patients with the same prime complaint have the same probability of taking a procedure. The second comparison model is a content-based Bayesian network. In this model, only the attributes of patients and procedures determine if a procedure is prescribed.

We test model performances by predicting future procedures for patients. ROC curve is used as evaluation criteria. In the experiment we selected the top N procedures rec-



(a) (b)

Figure 8.7: (a) ROC curves for predicting procedures on the total data. (b) ROC curves on a subset of patients with prime complaint *respiratory problem*.

ommended by the models. Sensitivity indicates how many percents of the actually being performed procedures were correctly proposed by the model. (1-specificity) indicates how many percent of the procedures that were not actually performed were recommended by the model. Along the curves, the N was varied from left to right as $N = 5, 10, \ldots, 50$.

In the experiments, we predict the following procedure of a patient *given his first procedure*. The corresponding ROC curves (averaged over all patients) for the experiments are shown in Figure 8.7(a). The infinite hidden relational model (E3) exploiting all relational information and all attributes gave best performance. When we remove the attributes of the entities, the performance degrades (E2). The results show that entity attributes are a reasonable predictor, without them, the performance of the full model cannot be achieved. If, in addition, we only consider the one-sided collaborative effect, the performance is even worse (E1). (E5) is the pure content-based approach using the Bayesian network. (E4) shows the results of relational model using reference uncertainty, which gave good results but did not achieve the performance of IHRM. Figures 8.7(b) shows the corresponding plots for a subset of patients (i.e. patients with prime complaint *respiratory problem*). The results exhibit similar trends as Figure 8.7(a).

The set of experiments verifies that the predictive probabilities of relationships can be estimated precisely by collecting all related evidence in the whole relational network. Although we do not perform expensive structure model selection, the real probabilistic dependency can be encoded in an elegant and compact way.

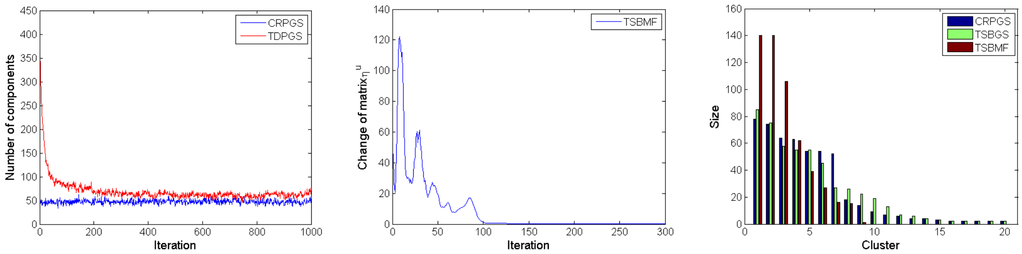
8.5.2 Movie Recommendation

Secondly, we demonstrate the performance of IHRM on the MovieLens data, which contains movie ratings from a large number of users (Sarwar et al., 2000). The task of the experiment is to evaluate the proposed inference methods. In the MovieLens data, there are two entity classes (User and Movie) and one relationship class (Like: users like movies). The User class has several attribute classes such as Age, Gender, Occupation. The Movie class has attribute classes such as Published-year, Genres and so on. The

Table 8.1: Performance of IHRM on MovieLens data.

	CRPGS	TSBGS	TSBMF	EA	Pearson
Given5	65.13	65.51	65.26	63.91	57.81
Given10	65.71	66.35	65.83	64.10	60.04
Given15	66.73	67.82	66.54	64.55	61.25
Given20	68.53	68.27	67.63	64.55	62.41
Time(s)	164993	33770	2892	-	-
Time(s/iter.)	109	17	19	-	-
$\#C.^u$	47	59	9	-	-
$\#C.^m$	77	44	6	-	-

relationship class Like has an auxiliary attribute R with two states: $R = 1$ indicates that the user likes the movie and $R = 0$ indicates otherwise. The IHRM model for the movie recommendation system is shown as Figure 8.4. In the data set, there are totally 943 users and 1680 movies. The ratings are originally recorded on a five-point scale, ranging from 1 to 5. We transfer the ratings to be binary, *yes* if a rating is higher than the average rating of the user, and *no* otherwise. The performances of all inference methods are analyzed from 3 points: prediction accuracy, convergence time and clustering effect. To evaluate the prediction performance, we execute 4 sets of experiments with respectively 5, 10, 15 and 20 randomly selected ratings as the known ratings, and predict the remaining ratings for each test user. These experiments are referred to as *given5*, *given10*, *given15* and *given20*. For testing, the relationship is predicted to *exist* (i.e. $R = 1$) if the predictive probability is larger than a threshold $\varepsilon = 0.5$, and *nonexist* (i.e. $R = 0$) otherwise.



(a) (b) (c)

Figure 8.8: (a) The traces of the number of User clusters for the runs of two Gibbs samplers. (b) The trace of the change of variational parameter matrix η^u for the run of the mean field method. (c) The sizes of the largest User clusters of the three inference methods.

We evaluate the following four inference methods: Gibbs sampling with Chinese restaurant process (CRPGS), Gibbs sampling with truncated stick-breaking (TSBGS), and the corresponding mean field method (TSBMF) as well as the empirical approximation method (EA). In TSBGS and TSBMF, the truncation parameters K^u and K^m are initially set to be the number of users and the number of movies, respectively. For TS-

BMF we consider $\alpha_0 \in \{5, 10, 100, 1000\}$, and obtain the best prediction when $\alpha_0 = 100$. For CRPGS and TSBGS α_0 is set to 100. For the variational method, the change of variational parameters between two iterations is monitored to determine the convergence. For the Gibbs samplers, the convergence was analyzed using three measures: Geweke statistic on likelihood, Geweke statistic on the number of components and autocorrelation. Figure 8.8 shows the traces for the runs of the 3 inference methods. (a) shows the traces of the number of User clusters for the runs of the 2 Gibbs samplers. (b) shows the change of variational parameters η^u in the variational method. Table 8.1 shows that the blocked Gibbs sampler TSBGS converges approximately by a factor 5 faster than the CRPGS sampler. The mean field method TSBMF is again by a factor around 10 faster than the blocked Gibbs sampler TSBGS and thus almost two orders of magnitude faster than CRPGS. CRPGS is much slower than the blocked Gibbs sampler mainly due to the large time cost per iteration shown as Table 8.1. The reason is that CRPGS samples the hidden variables one by one, which causes two additional time costs. First, the expectations of attribute parameters and relationship parameters have to be updated when sampling each user/movie assignment. Second, the posterior of hidden variables have to be computed one by one, thus we can not use fast matrix multiplication technology to accelerate the computation.

The prediction results are shown in Table 8.1. All methods under consideration achieve comparably good results. The best results are achieved by the two Gibbs sampling methods. To demonstrate the performance of IHRM, we also implement a Pearson-coefficient based collaborative filtering method (Resnick, 1994). It is clear that IHRM outperforms the traditional CF method, especially when there are few known ratings for the test user.

IHRM provides cluster assignments for all entities involved, in our case for the users and the movies. The columns $\#C.^u$ and $\#C.^m$ in Table 8.1 denote the numbers of clusters for User class and Movie class, respectively. The Gibbs samplers converge to 47-59 clusters for the users and 44-77 clusters for the movies. The mean field method has a tendency to converge to a smaller number of clusters with the same value of α_0. Further analysis shows that the clustering results of the three methods are actually similar. First, the sizes of most clusters generated by the Gibbs samplers are very small, e.g. there are 72% (75.47%) user clusters with less than 5 members in CRPGS (TSBGS). Figure 8.8(c) shows the sizes of the 20 largest User clusters of the three methods. Intuitively, the Gibbs samplers tend to assign the outliers to new clusters. Second, we compute the rand index (0-1) of the clustering results of the methods, the values are 0.8071 between CRPGS and TSBMF, 0.8221 between TSBGS and TSBMF, which also demonstrate the similarity of the clustering results between Gibbs samplers and mean field method. Table 8.2 shows the movies with highest posterior probabilities in the 8 largest clusters generated by CRPGS. The values in parentheses, e.g. 161/207, means: the number (167) of coincident movies assigned to the cluster in the last 10 iterations and the average size (207) of the cluster.As we can see from the numerical values, there is quite some fluctuation in the cluster assignments. In **cluster 1** most movies are very new and popular (the data set was collected from September 1997 through April 1998). Also they tend to be romance and comedy movies. **Cluster 2** includes many old movies, or movies produced by the non-USA countries, or drama movies. **Cluster 3** contains many comedies and **cluster 4**

Table 8.2: Clustering result of CRP-based Gibbs sampler on MovieLens data.

Cluster 1 (161/207)	**Cluster 2 (76/113)**
My Best Friend's Wedding (1997) G.I. Jane (1997) The Truth About Cats and Dogs (1996) Phenomenon (1996) Up Close and Personal (1996) Tin Cup (1996) Bed of Roses (1996) Sabrina (1995) Clueless (1995)......	Big Night (1996) Antonia's Line (1995) Three Colors: Red (1994) Three Colors: White (1994) Cinema Paradiso(1989) Henry V (1989) Jean de Florette (1986) A Clockwork Orange (1971) Citizen Kane (1941) Mr. Smith Goes to Washington (1939)......
Cluster 3 (49/98)	**Cluster 4 (32/51)**
Swingers (1996) Get Shorty (1995) Mighty Aphrodite (1995) Welcome to the Dollhouse (1995) Clerks (1994) Ed Wood (1994) The Hudsucker Proxy (1994) What's Eating Gilbert Grape (1993) Groundhog Day (1993)......	Event Horizon (1997) Batman and Robin (1997) Escape from L.A. (1996) Batman Forever (1995) Batman Returns (1992) 101 Dalmatians (1996) The First Wives Club (1996) Nine Months (1995) Casper (1995)......
Cluster 5 (16/27)	**Cluster 6 (9/15)**
Conspiracy Theory (1997) The Game (1997) Air Force One (1997) Ransom (1996) The Rock (1996) Primal Fear (1996) Crimson Tide (1995) In the Line of Fire (1993) The Abyss (1989)......	Brave Heart (1995) Forrest Gump (1994) Fugitive (1993) Terminator 2: Judgment Day (1991) Indiana Jones and the Last Crusade (1989) Die Hard (1988) Aliens (1986) Terminator (1984) Return of the Jedi (1983)
Cluster 7 (8/13)	**Cluster 8 (3/6)**
Shawshank Redemption (1994) Wrong Trousers (1993) Schindler's List (1993) Silence of the Lambs (1991) One Flew Over the Cuckoo's Nest (1975) Godfather (1972) Rear Window (1954) Casablanca (1942)	Star Wars (1977) Star Wars: The Empire Strikes Back (1980) Raiders of the Lost Ark (1981)

Table 8.3: An example gene.

Attribute	Value
Gene ID	G234070
Essential	Non-Essential
Class	1, ATPases 2, Motorproteins
Complex	Cytoskeleton
Phenotype	Mating and sporulation defects
Motif	PS00017
Chromosome	1
Function	1, Cell growth, cell division and DNA synthesis 2, Cellular organization 3, Cellular transport and transprotmechanisms
Localization	Cytoskeleton

consists of comedy and sci-fi movies. In **cluster 5** all the movies are relatively new and most movies include conspiracy and government. In **cluster 6** all the movies belong to the genre of action/thriller (except for Forrest Gump). **Cluster 7** are drama movies. The three movies in **cluster 8** are relatively old (from 1977 to 1981) and the main actor in the three movies is Harrison Ford. Overall we were quite surprised by the good interpretability of the clusters.

8.5.3 Prediction of Functions of Genes

The third evaluation is performed on the yeast genome data set of KDD Cup 2001 (Cheng et al., 2002). The goal of the experiment is to evaluate the expressive power of IHRM on the domain with multiple entity classes and multiple relationship classes.

The genomes in several organisms have been sequenced. Traditionally, the functions of genes/proteins are predicted by comparing with characterized genes/proteins in sequence similarity. But only 52% of 6449 yeast proteins have been characterized. Of the remaining, only 4% show strong similarity with the known ones at the sequence level. It is therefore necessary to integrate other information to characterize genes/proteins. In the experiment we need to predict functions of genes based on the information not only at the gene-level but also at the protein-level. The data set provided by KDD Cup 2001 consists of two relational tables. One table specifies properties of genes or proteins. These properties include *chromosome, essential, phenotype, motif, structural-category, complex and function. Chromosome* expresses the chromosome on which the gene appears. *Essential* specifies whether organisms with a mutation in this gene can survive. *Phenotype* represents the observed characteristics of organisms with differences in this gene. *Structural-category* represents the structural category of the protein for which this gene codes. *Motif* expresses the information about the amino acid sequence. *Complex* specifies how the expression of the gene can complex with others to form a larger protein. The other table in the data set contains the information about interactions between genes.

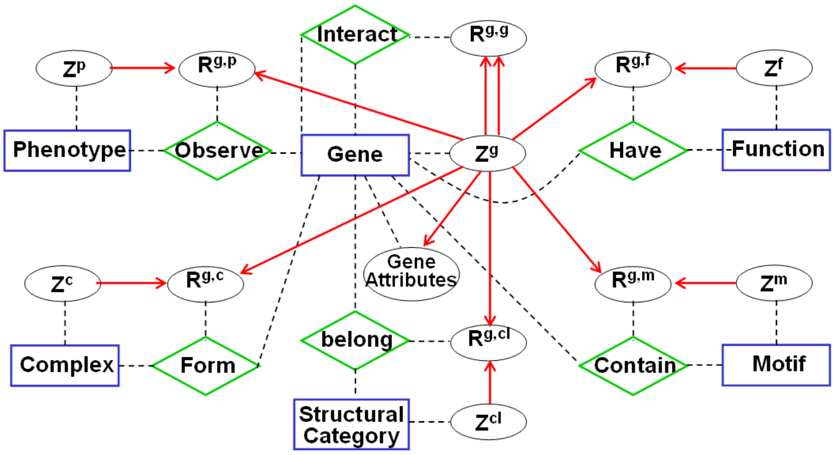


Figure 8.9: Infinite hidden relational model for a gene data set.

A gene typically has multiple complexes, phenotypes, structural-categories, motifs and functions, but only one chromosome and one essential value. An example gene is shown in Table 8.3. To keep the multi-relational nature of the data, we assume that there are six entity classes (Gene, Complex, Phenotype, Structural-category, Motif and Function) and six relationship classes (Interact: genes interact with each other, Have: genes have functions, Observe: phenotypes are observed for the genes, Form: which kinds of complex is formed for the genes, Belong: genes belong to structural-categories, Contain: genes contain characteristic motifs). Gene class has attribute classes: Essential, Chromosome. The attributes of other entity classes are not available in the data set. The data set totally contains 1243 genes. A subset (381 genes) is withheld for test in the KDD Cup 2001. The remaining 862 genes are provided to participants. In the data, there are 56 complexes, 11 phenotypes, 351 motifs, 24 structural-categories and 14 functions. There are mainly two challenges in the gene data set. First, there are many types of relationships. Second, there are large numbers of entities, but only a small number of known relationships.

Figure 8.9 shows IHRM for the gene data. A hidden variable is added to each entity, and all relationships are modeled as existence uncertainty. Thus each relationship class has an auxiliary attribute R with two states: 1 if the relationship exists, and 0 otherwise. The prediction results are shown in Table 8.4. There were 41 groups participating in the KDD Cup 2001 contest. The algorithms include decision tree, neural network, SVM, Bayesian network and so on. The performance of IHRM is comparable to the best results. The winning algorithm is based on inductive logic programming. The IHRM is only slightly worse (probably not significantly) if compared to the winning algorithm. The two Gibbs samplers do not mix well and fail to converge to a stationary distribution despite the long simulation time. The reason might be the sparsity and bias of the data.

Table 8.5 illustrates the clustering result of TSBMF with 8 largest clusters, which size is shown in parentheses. We give some brief explanation about the result. For example, for most genes in **cluster 1**, the structural category is *transcription factors*; the complex is *transcription complexes/transcriptosome*; the location is *nucleus*, and the function is *cellular organization and transcription*. We can intuitively view genes in the cluster as

Table 8.4: Prediction of gene functions

Methods	TSBMF	EA	Kdd cup winner
Accuracy (%)	92.78	93.18	93.63

Table 8.5: The largest gene clusters generated by TSBMF.

Cluster 1 (12)	Cluster 2 (7)	Cluster 3 (6)	Cluster 4 (5)
G234191 G234427 G235272 G235462 G235513 G235744 G236096 G236244 G236546 G238049 G238295 G238942	G234907 G235313 G235592 G236176 G237674 G237702 G238933	G235317 G235326 G235459 G235489 G235502 G235737	G234170 G234312 G234575 G236363 G238307
Cluster 5 (5)	Cluster 6 (5)	Cluster 7 (5)	Cluster 8 (4)
G234393 G234768 G236406 G236869 G240048	G235300 G235390 G235828 G238527 G239640	G235259 G235499 G235597 G235672 G235872	G234341 G234458 G234523 G236084

transcription genes. In **cluster 2**, all genes have the function *cellular organization (proteins are localized to the corresponding organelle)*. **Cluster 3** includes the genes which have the motif *PS01145*, and the function *cellular organization and protein synthesis*. All genes in the cluster tend to form *translation complexes* and locate at *cytoplasm*.

In the second set of experiments, we investigated the influence of a variety of relationships on the prediction of functions. We perform the EA inference method by ignoring a specific type of known relationships. The result is shown in Table 8.6. When a specific type of known relationships are ignored, the lower the prediction accuracy is, the higher the importance of this type of relationships is. One observation is that the most important relationship class is *Complex* that specifies how genes complex with another genes to form larger proteins. The second most important relationship class is: *Interact*. The result coincides with the lesson learned by KDD Cup 2001 that protein interaction information is less important in function prediction. This lesson is somewhat surprising since there is a general belief in biology that the knowledge about regulatory pathways is helpful to determine the functions of genes.

8.6 Discussion and Related Work

Kemp et al. (2006) presented an extension of their previous work (Kemp et al., 2004) which is quite close to the IHRM (they named their model infinite relational models, IRM). There are mainly three differences between the two models. First, the IRM is based on the

Table 8.6: The importance of relationship classes in predicting gene functions.

Ignored relationships	Accuracy(%)	Importance
Complex	91.13	197
Interaction	92.14	100
Structural-categories	92.61	55
Phenotype	92.71	45
Attributes of gene	93.08	10
Motif	93.12	6

predicate-based representation, whereas the IHRM is derived from the entity-relationship model. Second, the IRM introduces a hidden variable for each object and some predicates. The IHRM only introduces a hidden variable for each object. The attributes and relations are naturally associated with the involved objects. Third, in the IHRM one can specify any reasonable probability distribution for an attribute given its parent, whereas the IRM would model an attribute as a unary predicate, i.e. would need to transform the conditional distribution into a logical binary representation. This might be difficult if, for example, the attributes are continuous valued. In particular, the IHRM could model $P(A|Z)$ for example in a Gaussian distribution whereas the IRM would need to introduce an additional discrete representation. Kemp et al. (2006) described the CRPGS method also used in this chapter, exploiting the conjugacy between base distribution and likelihood distribution. Thus the blocked Gibbs sampler and its mean field solution derived here can also be used in the context of the IRM. Also related to IHRM is the refined probabilistic relational model with class hierarchies described in (Getoor et al., 2000), which specializes distinct probabilistic dependency for each subclass. The author-topic model introduced in (Rosen-Zvi et al., 2004) is another related work, which implicitly explored the document-author and document-word relations. Carbonetto et al. (2005) introduced the nonparametric BLOG model, which specifies nonparametric probabilistic distributions over possible worlds defined by first-order logic. Taskar et al. (2001) introduced a classification/clustering relational model, which associates a finite-dimensional hidden variable with each entity. The probabilistic dependency can be learned from the data or be specified in advance. Wang et al. (2005) proposed a group-topic model for text mining, which jointly discovers the latent groups in a network as well as the latent topics of events (or relations) between objects. Neville and Jensen (2005) developed a latent group model for relational data, which introduces two latent variables c_i and g_i for an object, and c_i is conditioned on g_i. The object attributes depends on c_i and relations depend on g_i of the involved objects. The limitation of the model is that only relations between members in the same group are considered. These related models demonstrate good performance in certain applications. However, most are restricted to domains with simple relationships.

8.7 Summary

In this chapter, we have introduced the infinite hidden relational model (IHRM), which is a new development in the thesis. The model extends the expressiveness of relational models by introducing for each entity an infinite-state latent variable as part of a Dirichlet process (DP) mixture model. We hope that IHRM will be a useful addition to relational learning by allowing for flexible inference in a relational network reducing the need for extensive structural model selection. In addition, IHRM also discovers the clustering structure for the domain of interest. The cluster assignment of an entity is not independently decided by its attributes, but is decided by its relationships with other entities. The experiment in the movie recommendation system demonstrates the clustering effect of IHRM by highly-interpretable results. To develop the full potential of IHRM, it is necessary to explore fast inference methods considering the slow mixing between DPs. The collapsed Gibbs sampling (CRPGS) is not capable of coming up to the expectation despite the best predictive accuracy. The blocked Gibbs sampling (TSBGS) is more than a factor of four faster than CRPGS. Another factor of 10 in speed up can be achieved by using mean field approximation (TSBMF). Thus the inference methods presented in this chapter make IHRM applicable to considerably larger domains. In the future work, it will be interesting to explore even more complex relational structures, for example by focussing on domains with hierarchical class structures (ontologies) or on domains with dynamic relationships.

Part IV

Conclusions

Chapter 9

Conclusions

In this chapter, we summarize the major results in the thesis and discuss some promising research directions for the future work.

9.1 Summary

In this thesis, we applied nonparametric Bayesian analysis on statistical relational learning and proposed two novel developments: Dirichlet enhanced relational model (DERL) and infinite hidden relational model (IHRM). The two models explicitly incorporate the relationships into probabilistic models and is capable of expressing the complex probabilistic dependencies with nonparametric Bayesian techniques, thus the relational information in a domain of interest can be truthfully exploited and encoded, which not only improves the predictive accuracy in estimating the probability of a relationship, but also improves the accuracy in estimating the probability of attributes and classifying/clustering entities.

We first presented a nonparametric hierarchical Bayesian relational model, DERL, which allows the parameters of conditional distributions are personalized to the entities and relationships, instead of being global. That means the conditional distributions themselves can be modeled as the attributes of entities. Additional flexibility is introduced by applying Dirichlet process (DP) as the nonparametric prior, which makes the posterior of the parameters of the conditional distributions as complex as necessary, although we can still implement our prior belief in the parameters of DP. As a result, the learned model can represent a rich relational structure and parameter dependencies which are impossible to be represented in a parametric formulation. For example, the coupling between different types of relationships could truthfully be modeled. In addition, DERL makes possible to represent the hierarchical class structures of entities in an elegant way: the distinct distribution for each subclass can share a common prior defined in the upper-layer class. For inference, we explore an efficient variational approximation method, which is motivated by the Pólya urn representation of DP. The performance of DERL is demonstrated in clinical data with promising results. We found that DERL improve the estimation about predictive probability of a future procedure by modeling the additional dependencies between physician's diagnoses and prescribed procedures, in contrast, the relational models with global parameters are not capable of modeling these dependencies and always provide the same prediction despite the increasing information.

The second model we proposed is infinite hidden relational model (IHRM), which reduces the extensive structural learning, a typical difficulty in current relational models. IHRM introduces for each entity a hidden variable, which is the only parent of the attributes of the entity and a parent of relationships it participates. Considering the different complexities of the classes of entities, it is better to allow the number of states of the hidden variables to be class-specific. In addition, the number should vary with the increasing data. To meet these requirements, we take advantage of the discreteness property of Dirichlet process and obtain a relational DP mixture model, which can simply be imaged as a mixture model with infinite number of states. The term *infinite* does not mean the number of states is really infinite, instead, it means the number is not specified in advance, but is decided by the data itself and as large as necessary. Given a universe of discourse describing the entities, their attributes and relationships, IHRM can be instantiated and results in a ground Bayesian network, across which all related information propagates into the variables of interest. From this point of view, IHRM can be understood as a relational generalization of hidden Markov model or hidden Markov random field. Clustering is a natural outcome of IHRM and provides interesting insight into the structure of the data. For inference, we develop four methods, including collapsed Gibbs sampler with Chinese restaurant process, blocked Gibbs sampler with truncated stick-breaking construction and the corresponding mean-field solution, as well as an memory-based empirical approximation. The blocked Gibbs sampler is more than a factor of five faster than the collapsed Gibbs sampler. Another factor of 10 in speed up can be achieved by using mean-field approximation. These fast inference algorithms make IHRM applicable to large and complex domains.

9.2 Future Work

There are a number of promising research directions for the future work.

First, it will be interesting to explore even more complex relational structures, for example domains with ontology. The ontological information provides a formal description about the domain of interest. Roughly speaking, ontology is defined by:

- A set of classes;
- A taxonomic (subclass-superclass) hierarchy;
- Slots for each class and value range for each slot.

Given ontology of a domain of interest, a knowledge base can be created by introducing individual instances of these classes and filling in specific slot values as well as describing additional slot restrictions. Bayesian analysis is particularly suited for a domain with ontology since an explicit representation of parameters and hyperparameters provides a natural way to capture the semantics about subclass-superclass and class-instance. The preliminary researches in the direction include relational Markov networks introduced by Taskar et al. (2002) and a PRM based model introduced by Getoor et al. (2000).

Taskar et al. (2002) made use of the semantic information about the relational structure of a set of web pages to improve the accuracy of link prediction/classification. Getoor et al. (2000) provided a refined probabilistic relational model with class hierarchy, which specializes distinct probabilistic dependency for each subclass.

Another interesting research direction is to extend discriminative modeling or hybrid of generative and discriminative modeling to relational learning. The work in this thesis is based on generative modeling techniques, which maximize the joint distribution of the relational data (and unobserved variables in the model) and is well suited for the situations where the data is not sufficient. The discriminative modeling techniques optimize the conditional likelihood of the data and typically provides excellent generalization performance. For example, via integrating discriminative modeling techniques, the accuracy of classification/prediction of IHRM might be improved. In addition, the hybrid of the two modeling techniques is widely used for semi-supervised learning. This type of techniques might be very promising in relational data due to the sparsity and unbalance of the relationships. The related work includes: Taskar et al. (2002), Singla and Domingos (2005a), Sutton and McCallum (2006), McCallum et al. (2006), Yu et al. (2007) and so on.

Last but not least, the direction we feel compelling is the extension of dynamic models to relational learning. In more cases than not, the domains of interest are naturally dynamic. For example, in a clinical system, there are physicians, patients, complaints, medications, diagnoses, treatments and so on. A typical workflow in the domain is a loop: examination of complaints → diagnoses from physicians → treatments/medications, which is clearly a dynamic process. Thus it makes sense to extend the dynamic models to relational data. The existing work includes: relational reinforcement learning proposed by Dzeroski et al. (1998), and dynamic PRM model proposed by Sanghai et al. (2003).

Nonparametric Bayesian analysis has created a revolution within the modern machine learning. We believe that these advanced techniques are also helpful in statistical relational learning and the results achieved in this thesis open up ample directions for future work. Our hope is that the proposed robust models can capture the probabilistic dependencies in the domains with complex relational structures more accurately and more efficiently than previous approaches.

Bibliography

Aldous, D. (1985). Exchangeability and related topics. In *Ecole d'ete de probabilities de saint-flour xiii 1983*, 1–198. Springer.

Anderson, C., Domingos, P., & Weld, D. (2002). Relational Markov models and their application to adaptive web navigation. *In Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining* (pp. 143–152). ACM Press.

Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Journal of Machine Learning*, *50*, 5–43.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, *2*, 1152–1174.

Bernardo, J. M., & Smith, A. F. M. (Eds.). (1994). *Bayesian theory.* New York: John Wiley and Sons.

Berry, D. A. (1996). *Statistics: A bayesian perspective.* Duxbury Press.

Bhattacharya, I., & Getoor, L. (2004). Iterative record linkage for cleaning and integration. *In Proceedings of SIGMOD2004 Workshop on Research Issues on Data Mining and Knowledge Discovery.*

Bhattacharya, I., & Getoor, L. (2005). *Entity resolution in graphs* (Technical Report 4758). Department of Computer Science, University of Maryland.

Bishop, C. M. (1994). *Mixture density networks* (Technical Report NCRG/94/004). Neural Computing Research Group, Aston University.

Blackwell, D., & MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, *1*, 353–355.

Blei, D., & Jordan, M. (2005). Variational inference for Dirichlet process mixtures. *Journal of Bayesian Analysis*, *1*, 121–144.

Carbonetto, P., Kisynski, J., de Freitas, N., & Poole, D. (2005). Nonparametric Bayesian logic. *In Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence.* AUAI Press.

Celeux, G., & Diebolt, J. (1985). The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, *2*, 73–82.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. *In Proceedings of ACM International Conference on Management of Data* (pp. 307–318). ACM Press.

Cheng, J., Hatzis, C., Hayashi, H., Krogel, M., Morishita, S., Page, D., & Sese, J. (2002). Kdd cup 2001 report. *SIGKDD Explorations*, *3*, 47–64.

Congdon, P. (2001). *Bayesian statistical modelling.* New York: Wiley.

Congdon, P. (2003). *Applied bayesian modelling.* New York: Wiley.

Deely, J. J., & Lindley, D. V. (1981). Bayes empirical Bayes. *Journal of the American Statistical Association*, *76*, 833–841.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

D'Esopo, D. (1959). A convex programming procedure. *Naval Logistics Quarterly*, *6*, 33–42.

Dey, D., Mueller, P., & Sinha, D. (Eds.). (1998). *Practical nonparametric and semiparametric bayesian statistics. lecture notes in statist. 133.* New York: Springer.

Dong, X., Halevy, A., & Madhavan, J. (2005). Reference reconciliation in complex information spaces. *In Proceedings of ACM International Conference on Management of Data* (pp. 85–96). ACM Press.

Dzeroski, S., & Lavrac, N. (Eds.). (2001). *Relational data mining.* Berlin: Springer.

Dzeroski, S., Raedt, L. D., & Blockeel, H. (1998). Relational reinforcement learning. *In Proceedings of International Workshop on Inductive Logic Programming* (pp. 11–22).

Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, *90*, 577–588.

Escobar, M. D., & West, M. (1998). *Computing bayesian nonparametric hierarchical models* (Technical Report ISDS 92-A20). Duke University.

Evans, M., & Swartz, T. (1995). Bayesian integration using multivariate student importance sampling. *Journal of Computing Science and Statistics*, *27*, 456–461.

Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, *1*, 209–230.

Friedman, N., Getoor, L., Koller, D., & Pfeffer, A. (1999). Learning probabilistic relational models. *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (pp. 1300–1309). Morgan Kaufmann.

Gelfand, A. E., & Kottas, A. (2002). A computational approach for full nonparametric Bayesian inference under Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, *11*, 289–305.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis, second edition.* New York: Chapman and Hall.

Getoor, L. (2005). Introduction to statistical relational learning. *Online tutorial.*

Getoor, L., & Diehl, C. (2005). Link mining: A survey. *SIGKDD Explorations*, *7*.

Getoor, L., Friedman, N., Koller, D., & Pfeffer, A. (2001). Learning probabilistic relational models. In S. Dzeroski and N. Lavrac (Eds.), *Relational data mining*, 7–35. Springer.

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2003). Learning probabilistic models of link structure. *Journal of Machine Learning Research*, *3*, 679–707.

Getoor, L., Koller, D., & Friedman, N. (2000). From instances to classes in probabilistic relational models. *In Proceedings of ICML2000 Workshop on Attribute-Value and Relational Learning.*

Ghahramani, Z., & Jordan, M. I. (1994). *Learning from incomplete data* (Technical Report 108). MIT Center for Biological and Computational Learning.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. (Eds.). (1995). *Markov chain monte carlo in practice.* New York: Chapman and Hall.

Good, I. J. (1965). *The estimation of probabilities: An essay on modern bayesian methods.* Cambridge, Mass: M.I.T. Press.

Han, J., & Kamber, M. (2006). *Data mining, second edition: Concepts and techniques.* Morgan Kaufmann.

Heckerman, D., Meek, C., & Koller, D. (2004). *Probabilistic models for relational data* (Technical Report MSR-TR-2004-30). Microsoft.

Hofmann, T., & Puzicha, J. (1999). Latent class models for collaborative filtering. *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence.*

Ishwaran, J., & James, L. (2001). Gibbs sampling methods for stick breaking priors. *Journal of the American Statistical Association*, *96*, 161–174.

Jelinek, F. (1997). *Statistical methods for speech recognition.* Cambridge, MA, USA: MIT Press.

Jordan, M. I. (2005). Dirichlet processes, Chinese restaurant processes and all that. *Tutorial at the NIPS 2005.*

Jordan, M. I., Ghahramani, Z., Jaakkola, T., & Saul, L. (1998). An introduction to variational methods for graphical models. In M. I. Jordan (Ed.), *Learning in graphical models*, 105–161. MIT Press.

Kemp, C., Griffiths, T., & Tenenbaum, J. R. (2004). *Discovering latent classes in relational data* (Technical Report AI Memo 2004-019).

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., & Ueda, N. (2006). Learning systems of concepts with an infinite relational model. *In Proceedings of the Twenty-first National Conference on Artificial Intelligence.*

Kersting, K., & Raedt, L. D. (2000). Bayesian logic programs. *In Proceedings of the Work-in-Progress Track at the Tenth International Conference on Inductive Logic Programming* (pp. 138–155).

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, *46*, 604–632.

Koller, D., & Pfeffer, A. (1997). Object-oriented Bayesian networks. *In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence* (pp. 302–313). Morgan Kaufmann.

Koller, D., & Pfeffer, A. (1998). Probabilistic frame-based systems. *In Proceedings of the Fifteenth National Conference on Artificial Intelligence* (pp. 580–587). AAAI Press.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 282–289). Morgan Kaufmann.

Liu, H., Yin, X., & Han, J. (2005). An efficient multi-relational naive Bayesian classifier based on semantic relationship graphs. *In Proceedings of the Fourth International Workshop on Multi-relational Mining* (pp. 39–48). ACM Press.

Liu, J. S. (1996). Nonparametric hierarchical Bayes via sequential imputations. *The Annals of Statistics*, *24*, 911–930.

Lu, Q., & Getoor, L. (2003). Link-based classification. *In Proceedings of Twentieth International Conference on Machine Learning* (pp. 496–503). Morgan Kaufmann.

MacEachern, S. (1994). Estimating normal means with a conjugate style Dirichlet process prior. *Communications in Statistics: Simulation and Computation*, *23*, 727–741.

MacEachern, S. N., Clyde, M., & Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: the next generation. *The Canadian Journal of Statistics*, *27*, 251–267.

MacEachern, S. N., & Mueller, P. (1998). Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, *7*, 223–238.

McAuliffe, J. D., Blei, D. M., & Jordan, M. I. (2006). Nonparametric empirical Bayes for the Dirichlet process mixture model. *Statistics and Computing*, *16*, 5–14.

McCallum, A., Pal, C., Druck, G., & Wang, X. (2006). Multi-conditional learning: generative/discriminative training for clustering and classification. *In Proceedings of the Twenty-first National Conference on Artificial Intelligence.* AAAI Press.

Miloslavsky, M., & van der Laan, M. J. (2003). Fitting of mixtures with unspecified number of components using cross validation distance estimate. *41*, 413–428.

Mueller, P., & Quintana, F. A. (2004). Nonparametric Bayesian data analysis. *Statistical Science*, *19*, 95–110.

Naylor, J. C., & Smith, A. F. M. (1982). Applications of a method for the efficient computation of posterior distributions. *The Annals of Statistics*, *31*, 214–215.

Neal, R. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, *9*, 249–265.

Neal, R., & Hinton, G. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in graphical models*, 355–368. MIT Press.

Neville, J., & Jensen, D. (2000). Iterative classification in relational data. *In Proceedings of AAAI-2000 Workshop on Learning Statistical Models from Relational Data* (pp. 13–20). AAAI Press.

Neville, J., & Jensen, D. (2005). Leveraging relational autocorrelation with latent group models. *In Proceedings of the Fourth International Workshop on Multi-relational Mining* (pp. 49–55). New York, USA: ACM Press.

Newton, M. A., & Zhang, Y. (1999). A recursive algorithm for nonparametric analysis with missing data. *Biometrika*, *86*, 15–26.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The pagerank citation ranking: Bringing order to the web* (Technical Report). Stanford University.

Papalambros, P. Y., & Wilde, D. J. (2000). *Principles of optimal design: Modeling and computation, second edition.* New York: Cambridge University Press.

Quintana, F. A., & Newton, M. A. (2000). Computational aspects of nonparametric Bayesian analysis with applications to the modeling of multiple binary sequences. *Journal of Computational and Graphical Statistics*, *9*, 711–737.

Raedt, L. D., & Kersting, K. (2003). Probabilistic logic learning. *SIGKDD Explor. Newsl.*, *5*, 31–48.

Rattigan, M., & Jensen, D. (2005). The case for anomalous link discovery. *SIGKDD Explorations*, *7*.

Redner, R., & Walker, H. (1984). Mixture densities, maximum likelihood, and the EM algorithm. *SIAM Review*, *26*, 195–39.

Resnick, P. (1994). Grouplens: An open architecture for collaborative filtering of netnews. *In Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work* (pp. 175–186). ACM.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Journal of Machine Learning Research*, *62*, 107–136.

Richardson, S., & Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, Series B*, *59*, 731–92.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. *In Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence* (pp. 487–494). AUAI Press.

Sanghai, S., Domingos, P., & Weld, D. (2003). Dynamic probabilistic relational models. *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence.* Morgan Kaufmann.

Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2000). Analysis of recommender algorithms for e-commerce. *In Proceedings of ACM E-Commerce Conference* (pp. 158–167). New York, USA: ACM Press.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, *4*, 639–650.

Singla, P., & Domingos, P. (2005a). Discriminative training of Markov logic networks. *In Proceedings of the Twentieth National Conference on Artificial Intelligence* (pp. 868–873). AAAI Press.

Singla, P., & Domingos, P. (2005b). Object identification with attribute-mediated dependences. *In Proceedings of the Ninth European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 297–308). Springer.

Sutton, C., & McCallum, A. (2006). An introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar (Eds.), *Introduction to statistical relational learning.* MIT Press.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *In Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence* (pp. 485–492). Morgan Kaufmann.

Taskar, B., Segal, E., & Koller, D. (2001). Probabilistic classification and clustering in relational data. *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (pp. 870–878).

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2004). *Hierarchical* D*irichlet processes* (Technical Report 653). UC Berkeley Statistics.

Tresp, V. (2006). Dirichlet processes and nonparametric Bayesian modelling. *Online Tutorial.*

Tresp, V., & Yu, K. (2004). An introduction to nonparametric hierarchical Bayesian modelling. In *Proceedings of hamilton summer school on switching and learning in feedback systems*, 290–312. Springer.

Ullman, J. D., & Widom, J. (1997). *A first course in database systems.* Upper Saddle River,NJ,USA: Prentice Hall.

Walker, S. G., Damien, P., Laud, P., & Smith, A. F. M. (1999). Bayesian nonparametric inference for random distributions and related functions (with discussion). *Journal of the Royal Statistical Society. Series B, statistical methodology*, *61*, 485–527.

Wang, X., Mohanty, N., & McCallum, A. (2005). Group and topic discovery from relations and text. *In Proceedings of the Third International Workshop on Link Discovery* (pp. 28–35). New York, USA: ACM Press.

Wei, G. C. G., & Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, *85*, 699–704.

West, M., Mueller, P., & Escobar, M. (1994). Hierarchical priors and mixture models with applications in regression and density estimation. In P. R. Freeman and A. F. M. Smith (Eds.), *Aspects of uncertainty*, 363–386. New York: Wiley.

Wrobel, S. (2001). Inductive logic programming for knowledge discovery in databases. In S. Dzeroski and N. Lavrac (Eds.), *Relational data mining*, 74–101. Springer.

Xu, Z., Tresp, V., Yu, K., & Kriegel, H.-P. (2006). Infinite hidden relational models. *In Proceedings of the Twenty-second Conference on Uncertainty in Artificial Intelligence.* AUAI Press.

Xu, Z., Tresp, V., Yu, K., Yu, S., & Kriegel, H.-P. (2005). Dirichlet enhanced relational learning. *In Proceedings of the Twenty-second International Conference on Machine Learning* (pp. 1004–1011). ACM Press.

Xu, Z., Tresp, V., Yu, S., Yu, K., & Kriegel, H.-P. (2007). Fast inference in infinite hidden relational models. *Proceedings of the 5th International Workshop on Mining and Learning with Graphs (MLG 2007).*

Yedidia, J., Freeman, W., & Weiss, Y. (2005). Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, *51*, 2282–2312.

Yu, K., Chu, W., Yu, S., Tresp, V., & Xu, Z. (2007). Stochastic relational models for discriminative link prediction. *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press.

Yu, K., Tresp, V., & Yu, S. (2004). A nonparametric hierarchical Bayesian framework for information filtering. *In Proceedings of the Twenty-seventh International ACM SIGIR Conference* (pp. 353–360). ACM.