Learning Latent Theories of Relations and Individuals

by

Michael Chi-Hao Chiang

B. Eng., University of Melbourne, 2000

M. Eng. Sci., University of Melbourne, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES

(Computer Science)

The University Of British Columbia

(Vancouver)

April 2011

Abstract

Inductive learning of statistical models from relational data is a key problem in artificial intelligence. Two main approaches exist for learning with relational data, and this thesis shows how they can be combined in a uniform framework.

The first approach aims to learn dependencies amongst features (relations and properties), e.g. how users' purchases of products depend on users' preferences of the products and associated properties of users and products. Such models abstract over individuals, and are compact and easy to interpret.

The second approach learns latent properties of individuals that explain the observed features, without modelling interdependencies amongst features. Latent-property models have demonstrated good predictive accuracy in practise, and are especially useful when few properties and relations are observed. Interesting latent groupings of individuals can be discovered.

Our approach aims to learn a unified representation for dependency structures for both observed features *and* latent properties. We develop a simple approximate expectation maximisation (EM) algorithm for learning the unified representation, and experiments demonstrate cases when our algorithm can generate models that predicts better than dependency-based models of observed features as well as a state-of-the-art latent-property model.

We extend our approximate EM algorithm to handle uncertainty about the number of values for latent properties. We search over the number of values and return error bounds, as an alternative to existing proposals based on sampling in the posterior distribution over the number of values.

We also solve a specific case where dependencies involve functional relations, which induces a verbose model with many parameters. In comparison, the stan-

dard solution of aggregating over all values of the function yields a simple model that predicts poorly. We show how to learn an optimal intermediate-size representation efficiently by clustering the values of the function. The proposed method generates models that capture interesting clusters of function values, dominates the simple model in prediction, and can surpass the verbose model using much fewer parameters.

Table of Contents

List of Tables

List of Figures

Glossary

AI	artificial intelligence
FOL	first-order logic
FOPL	first-order probabilistic logic
SRL	statistical relational learning
ILP	inductive logic programming
CPT	conditional probability table
MCMC	Monte Carlo Markov chain
ML	maximum likelihood
BP	belief propagation
EM	expectation maximisation
IRM	infinite relational model
BCTF	Bayesian clustered tensor factorisation
LRM	latent relational model
ILRM	infinite-size latent relational model
IHRM	infinite hidden relational model
CRP	Chinese restaurant process

DP Dirichlet process

PRM probabilistic relational model

ICL independent choice logic

PHA probabilistic Horn abduction

BLP Bayesian logic programs

RBN relational Bayesian network

MLN Markov logic network

Acknowledgements

During my time as a doctoral student I have gained more knowledge and understanding than I could have imagined; there are many individuals to thank. First and foremost I wish to thank my supervisor David Poole, whose depth of knowledge and willingness to push the boundaries of conventional ideas has been challenging and greatly inspiring. Ideas presented in this thesis have most often been the result of David's insight and intuition.

Secondly, I wish to also thank Professor Joel Friedman, whose graduate course on Markov chains proved to be a key component in my development. He has also been an utmost reliable member of my supervisory and have provided some valuable guidance. On a similar note, I thank Dr. Kevin Murphy and Dr. Nando de Freitas for not only being helpful in providing feedback for my work, but also for their interesting research and teachings that I often refer to.

Fellow students have also helped me along tremendously. The have shown great generosity with their time and knowledge. In particular, I have gained a lot from interacting with Jacek Kisynski, Mark Crowley, Peter Carbonetto, Frank Hutter, Emtiyaz Khan, and Mark Schmidt.

I could not have survived without the support of great friends. I thank Ed, Ross, Kerrie, Shannon, Sandra, Dieter, Christian, Nolwenn, Daylen, Bjoern & Britta, Frank, Roman, Matt, Alana, Nora, Minako, Corinne, Clarice, Ali, Andrew, Aline, Julie, Grace, Phillip, and too many others to mention. These individuals have been my source of perspective and encouragement throughout this time.

Finally, deepest gratitude to my family, who gave me the impetus and desire to pursue higher goals. My parents Robert and Sharon have made too many sacrifices on my account, and their continued show of pride and support is of great comfort.

Thanks also to my younger sister Michelle, who has transformed from a thorn in my side ;-) to being a great friend. Often, it is her that I look up to. Last but not least, to my late grandparents, who played a central role in my upbringing. Their influence on me will be no doubt be lasting.

Chapter 1

Introduction

Learning is an ability to improve one's performance through experience (Russell and Norvig, 2009), and is a central problem studied in artificial intelligence (AI). In order for an intelligent agent to operate successfully in the world, a model of its surrounding world is generally sought. The agent can evaluate a learned model in terms of the accuracy of inferences made using the model.

The general problem addressed in this thesis is learning in domains that are *relational*. A relational domain is a collection of objects or *individuals* where *relational ties* exist between individuals. A relational tie indicates the value of a relation amongst some collection of individuals. For instance, if Andy and John are *friends*, then a *friends* tie exists between Andy and John, where the value of the tie is "true". Or, if Ken obtained the *grade* A- in CS101 in 2009, then a *grade* tie exists for Ken, CS101, and 2009, where the value of the tie is A-. Relational data consist of examples of such ties, along with properties of individuals, e.g. Joe's height, if observed.

Statistical machine learning increasingly emphasise the importance of learning in relational domains, driven by important relational problems that include collaborative filtering (Hofmann and Puzicha, 1999; Koren, 2008; Marlin and Zemel, 2004; Ungar and Foster, 1998), analysis of academic citation networks (McCallum et al., 2000), document networks (Chang and Blei, 2010), webpage linkage (Newman, 2003), social networks (Airoldi et al., 2008; Handcock et al., 2007; Hofman and Wiggins, 2008; Newman, 2003), or biological networks (Airoldi et al., 2008).

Given relational data representing different relations amongst individuals, a traditional approach to for relational learning – with roots in the inductive logic programming (ILP) and statistical relational learning (SRL) communities (De Raedt, 2008; Getoor and Taskar, 2007; Muggleton and De Raedt, 1994; Nienhuys-Cheng and de Wolf, 1997) – aims to discover how each relation are depend (probabilistically) on other relations. The result is a dependency structure over relations that best fits the data, where each dependency may be logical or probabilistic. For instance, observing *co-worker* ties and *collaboration* ties amongst individuals, one may learn that for every pair (x, y), *collaborates*(x, y) implies *co-worker*(x, y) with some probability p. Models learned in this setting quantify over individuals, i.e. they abstracting over individuals, and afford a desirable property of small representation size and ease of interpretation.

Rather than searching for the best dependency structure over observed relations, an alternative approach is to postulate and infer latent properties or relations (a property is a unary relation) that explain the observed relations. This approach has been useful in analysis of networks and collaborative filtering (Airoldi et al., 2008; Handcock et al., 2007; Hofman and Wiggins, 2008; Hofmann and Puzicha, 1999; Kemp et al., 2006; Kok and Domingos, 2007; Koren, 2008; Marlin and Zemel, 2004; Neville and Jensen, 2005; Newman, 2003; Taskar et al., 2001; Ungar and Foster, 1998; Xu et al., 2006). Suppose customers' ratings of products are given, a latent-property model describes how the *rating* relation depend on the latent properties of users and products. The learning goal is to infer the values of latent properties of each individual that best fits the data (observed ratings). This approach has lead to improved predictive accuracy over models which do not use latent properties (Jensen et al., 2004; Neville and Jensen, 2005; Taskar et al., 2001; Xu et al., 2006).

The aim of this thesis is a unifying framework for learning dependency-based models that include both observed and latent relations. Existing work have either focused on dependency structures over observed relations only, or latent properties of individuals as explanations of all observed relations. The combination of the two settings can directly exploit the advantages of both representations.

The remainder of this introduction covers some prominent relational domains and existing proposals for learning in these domains (Sec. 1.1). In addition, in

Sec. 5.4 philosophical difficulties associated with models that represent only observed relations are highlighted, which serves to motivate our proposal to combine observed and latent relations in a single representation.

1.1 Learning in Relational Domains

There are several fields, from statistics to social sciences, which address relational data. In these fields, relations are often viewed as networks whose nodes are individuals and edges represent relational ties amongst individuals. The simplest networks are defined over a single type of individuals, for example friendship networks over people, protein or gene networks, network of citations amongst research papers, or hyperlink network of webpages. Figure 1.1 shows some of these examples.

A common strategy to analysing such networks is to model latent properties (unary relation) of individuals in the network (Airoldi et al., 2008; Handcock et al., 2007; Hofman and Wiggins, 2008; Hofmann and Puzicha, 1999; Koren, 2008; Marlin and Zemel, 2004; Newman, 2003; Ungar and Foster, 1998). That is, each individual has a latent property, and relational ties are probabilistically dependent on the latent properties of individuals implicated by the relational tie. The general idea is to postulate latent properties to explain relational ties.

Learning latent-property models involves computing the latent property values (from the observed relational ties) for each individual in the network, and how the values of relational ties depend on the latent properties. Inferring values of latent properties is commonly known as *clustering*, where each latent property value correspond to a cluster, and learning the value of a latent property for an individual amounts to assigning the individual to the corresponding cluster. For instance, we can represent friendship as being probabilistically dependent on the latent properties of participating individuals, with the conditional statement

$$\forall X, Y \quad P\left(\textit{friends}(X,Y) = true \mid \alpha(X) = i \wedge \alpha(Y) = j\right) = p$$

where α is the name of a latent property, and i, j are values of α. The meaning of the rule is that for every pair of individuals (x, y), (x, y) are friends with probability p if $\alpha(x) = i$ and $\alpha(y) = j$ holds.

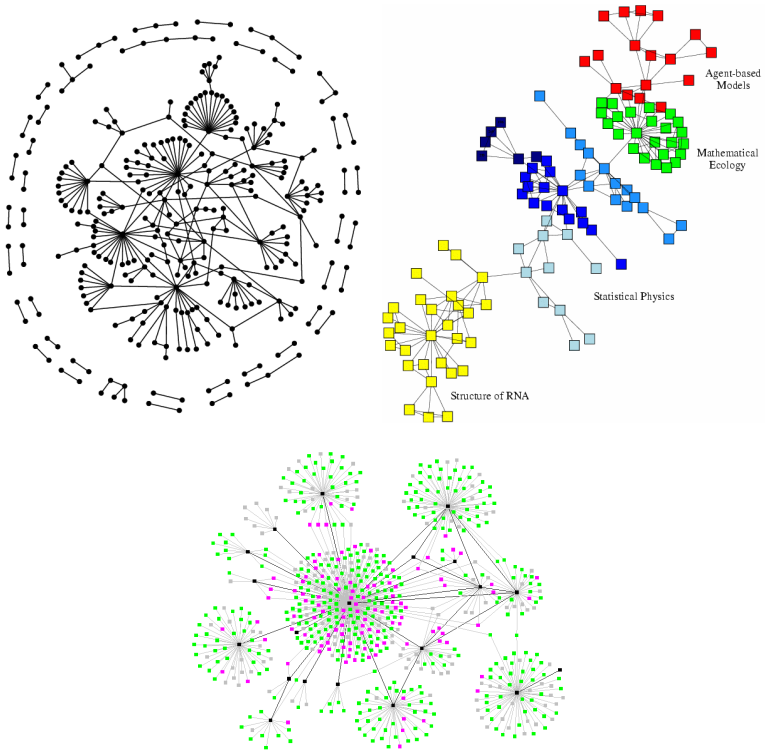


Figure 1.1: (left) a yeast protein network; (center) a collaboration network amongst researchers in various disciplines; (right) a disease contagion network. Network nodes denote individuals, e.g. proteins or researchers, and edges denote relational ties, e.g. research x *collaborates* with researcher y. (Images sourced from http://www-personal.umich.edu/∼mejn/networks)

A network of relational ties can be large and complex, where much information is available to be exploited. Learning latent properties of individuals to explain the observed ties is one way to exploit the available information. Even in the simple case of having only one network, the problem of inferring latent property values is already computationally intractable (Ungar and Foster, 1998).

In general, different relations may be observed amongst individuals, e.g. *friends*, *likes*, or *respects*. Learning latent-property models with multiple observed relations has been proposed by (Kemp et al., 2006; Sutskever et al., 2010; Xu et al., 2006).

To illustrate such a model, suppose we observe *friendship* and *respect* ties amongst individuals, we may have a latent-property model as follows:

$$\forall X, Y \quad P\left(\textit{friends}(X,Y) = true \mid \alpha(X) = i \wedge \alpha(Y) = j\right) = p$$
$$\forall X, Y \quad P\left(\textit{respect}(X,Y) = true \mid \alpha(X) = i' \wedge \alpha(Y) = j'\right) = p'$$

The main learning goal here is to infer values of α for all domain individuals that explain all observed *friend* and *respect* ties. The common assumption amongst current latent-property models of relational data is that the latent-properties are sufficient to explain all observed relations, and dependencies amongst relations are thus not needed Xu et al. (2006). Whilst these models achieve good predictive accuracy and reveals interesting latent groupings of individuals, Kemp et al. (2006) notes that a desired extension entails interpretable models about relations, and points to more traditional frameworks of learning models based on first-order logic (FOL).

The traditional relational learning approach naturally handles multiple relations, where the aim is to learn a set of dependencies amongst the observed relations. The underlying motivation is that each relation can be explained by other relations in the observed set. The ILP community (De Raedt, 2008; Muggleton and De Raedt, 1994) has contributed a vastly to learning dependency-based models represented in FOL. Algorithms such as Progol (Muggleton, 1995a) and Golem (Muggleton, 1992) and have been successful in discovering interesting first-order logic (FOL) programs in various domains. Examples include modelling mutagenicity of chemical compounds based on knowledge about its molecular structure (Srinivasan et al., 1996), modelling London's tube (train) network (Bratko and Muggleton, 1995), or learning to recognise illegal chess moves (Muggleton et al., 1989), amongst others.

The SRL community has a similar goal, but with particular emphasis on probabilistic theories and accounting for domain uncertainty. Many languages have been proposed for representing first-order knowledge with uncertainty[1]. Such lan-

[1]SRL is not a full extension of ILP towards probabilistic representations, as ILP deals with a greater range of dependency structures than is possible with SRL algorithms.

guages are generally known as first-order probabilistic logic (FOPL) (Friedman et al., 1999; Getoor et al., 2002; Kersting and De Raedt, 2002; Kok and Domingos, 2005; Muggleton, 2003; Richardson and Domingos, 2006)), and learning entails finding the best dependency structure over the known relations, as well as parameters that represent the uncertainty of each dependency. Consider the UW-CSE domain[2] where data about properties of students and faculty members are given, along with various relations. A possible FOPL rule for this domain is

$$\forall X, Y \quad P\left(\textit{advised\_by}(X, Y) \mid \textit{taught\_by}(X, Y), \textit{stu}(X), \textit{prof}(Y)\right) = p \quad (1.1)$$

where all properties and relations in the above rule are Boolean-valued and are observed[3]. The meaning of the statement is that for any pair (x, y), x is advised by y with probability p if x is a student and y is a professor, and that x was taught by y. Most current FOPLs models comprise of a set of such rules, and are coherent joint probability models of the domain (Friedman et al., 1999; Kersting and De Raedt, 2000; Muggleton, 1995b; Poole, 1993b, 1997; Richardson and Domingos, 2006).

The following section describes particular motivations for introducing latent relations, by examining the technical and philosophical issues that are confronted by models comprised of only observed relations when predictions are concerned.

1.2 Relational Models and Probability Prediction

The conditional probability associated with rules such as Eq. 1.1 represent a statistic obtained about student-professor pairs. Using such a rule for prediction means that every student-professor pair that satisfies the condition is assigned the same probability. This mode of reasoning leads to some philosophical as well as technical difficulties outlined in this section. We start with an example.

Example 1

To treat tuberculosis (TB), Dr. Smith prescribes a long and costly course of antibiotic medication. Skin tests are performed to determine whether a patient has TB. According to his records, 53 out of 100 (53%) previous positive skin tests have

[2] http://alchemy.cs.washington.edu/data/uw-cse

[3] Relations where some cases are unobserved are still referred to as observed relations. Relations where all cases are unobserved are latent relations.

correctly indicated the presence of TB (true-positive). On this basis Dr. Smith's initial hypothesis is

$$\mathbf{I} : \forall X, P(\textit{tb}(X) \mid \textit{pos\_skin\_test}(X)) = 0.53$$

which means that the probability that any patient returning a positive skin test has 0.53 probability of carrying TB. Based on this hypothesis, in prescribing medication for all positive skin test patients he expects to err in 47% (false-positive) of positive skin test cases. Dr. Smith also knows that prior vaccinations for TB triggers positive skin tests, and has observed that 40 of the positive skin tests were attributable to prior vaccinations. Accounting for this fact, he calculates that for vaccinated cases, a positive skin test means a false-positive, whilst for non-vaccinated cases, there are now 53 of 60 (88%) skin tests that correctly indicate the presence of TB. He thus forms a second (more specific) hypothesis

$$\mathbf{IIa} : \forall X, P(\textit{tb}(X) \mid \textit{pos\_skin\_test}(X), \textit{vacc}(X)) = 0$$
$$\mathbf{IIb} : \forall X, P(\textit{tb}(X) \mid \textit{pos\_skin\_test}(X), \neg\textit{vacc}(X)) = 0.88$$

Thus, by checking for prior vaccinations, Dr. Smith expects to at least reduce his error from 47% to 12% on treated patients.

The example exposes two important issues. First, using hypothesis **I** and expecting 53% success (88% for hypothesis **II**) for all impending cases, the doctor has effectively extrapolated that all positive skin tests have a 0.53 probability (0.88 for hypothesis **II**) of correctly revealing the presence of TB.

Using sample statistics as the probability for some proposition about an individual is called *direct inference*. In philosophy, the validity of direct inference has been the subject of a long-standing philosophical debate (see McGrew (2001) for a survey), closely related to the *problem of induction* in philosophy (see Vickers (2009)). The basic issue, first articulated by Hume (1748), is that inductions, e.g. hypotheses **I** and **II**, lacks logical justification. Namely, any deductive justification of the hypothesis implies the existence of an entailing theory that is itself subject to the same question of justification. Arguments for a deductive justification thus confront the problem of infinite regress. Arguing that inductions can be justified inductively, on the other hand, renders the argument circular. Thus, concluding that

the true probability of a new case being a true-positive based on observed rates on past samples is an induction, and is therefore not logically justifiable.

The second issue questions whether a more specific hypothesis warrants discarding more general hypotheses. For example, observing that some new patient is not vaccinated may not be sufficient grounds for discarding the hypothesis **I**. Hypothesis **II** is more specific to the query, and is expected to be a more precise hypothesis than **I**. However, narrower samples carry degraded statistical confidence. Hypothesis **I**, on the other hand, holds greater statistical confidence but is less precise.

Selecting the most specific sample is a principle supported by philosophers such as Reichenbach (1949) and Kyburg (1983, 1974). The key argument is that the true probability for a proposition is a matter of finding the "right" statistical sample – the *right reference class* – and that the right reference class is one that is the most specific. This argument answers both issues raised here; that the right reference class provides the correct sample statistic corresponding to the true probability for the proposition in question, and so relinquishes the need for hypotheses that are more general (less specific). In response to the problem of diminished statistical confidence associated with narrow reference classes, one can choose the most specific reference class which yields acceptable statistical confidence (Kyburg, 1983; Reichenbach, 1949). This approach is *ad hoc*, however.

In general, the problem of identifying the right reference class from a set of competing reference classes is known as the *reference class problem* (Reichenbach, 1949). There are, however, practical concerns about whether it is possible to find the right reference class at all, given that complete data is never available. In other words, we may never observe an adequate set of attributes required to define a reference class specific enough that encapsulates the correct probability. Nonetheless, the specificity criterion highlights the general idea that learning as much as possible about individuals can help yield the correct probability.

Where reference class methods are based on sample statistics, and thus reliant solely on observed data, we argue that attempting to model latent properties of individuals can further improve accuracy. Methods for learning latent properties mentioned in Sec. 1.1 are examples of this approach. We show in Ch. 3 that reference classes alone may lead to systematic prediction errors, and the inclusion

of latent properties can rectify this bias.

The following example demonstrates the importance that the properties of individuals play, by showing that unintuitive predictions can result from using the reference class method.

Example 2
Principal Jones wishes to assess the likelihood of Timothy passing grade 9 maths. Mr. Jones proposes the following theory:

$$\begin{aligned}
\textbf{I: } & \forall S, C \quad P(\textit{pass}(S, C)) = 0.77 \\
\textbf{II: } & \forall S, C \quad P(\textit{pass}(S, C) \mid C = \textit{grade_9_math}) = 0.84 \\
\textbf{III: } & \forall S, C \quad P(\textit{pass}(S, C) \mid S = \textit{timothy}) = 0.85
\end{aligned}$$

The specificity criterion yields **II** and **III** as valid candidates, thus Mr. Jones would need to decide on which is the most appropriate statistic. To decide between **II** and **III**, Mr. Jones chooses **II** as it is the statistically stronger reference class (since the number of grade 9 math students outnumber courses Timothy took), and ascribes 0.84 as the probability that Timothy will pass grade 9 maths.

In this example, choosing **II** is unintuitive as it effectively assumes that Timothy is a typical student, and prediction only takes into account the effect of the course. Similarly, if we choose **III**, then the effect of the course would be marginalised. A more general question is how all of the known reference classes can be combined to form a weighted prediction. A straightforward approach would be to interpolate between $\mathbf{I} \sim \mathbf{III}$. However, suppose Timothy is an above average student and grade 9 mathematics is a particularly easy course, then it is expected that the probability of Timothy passing grade 9 mathematics would be higher than any interpolated probability. It is unclear whether there is a principled to extrapolative combination. In this work we propose to use latent properties as a way to alleviate this problem.

The preceding discussion highlights the theme that in addition to exploring dependency structures amongst observed relations, hidden knowledge about individuals (i.e. via latent properties) can help to further improve predictions about relations. That is, whilst reference class methods rely on conditioning to obtain

predictions and demonstrably leads to systematic errors, we support the idea that latent knowledge about individuals plays a central role in restoring accuracy. In this thesis we examine representations that incorporate latent knowledge about individuals in a model where complex dependency structures over relations are allowed, and provide algorithmic procedures towards learning such representations.

1.3 Thesis Contributions and Outline

The contributions of the thesis, in relation to existing work, can be explained using Fig. 1.2.

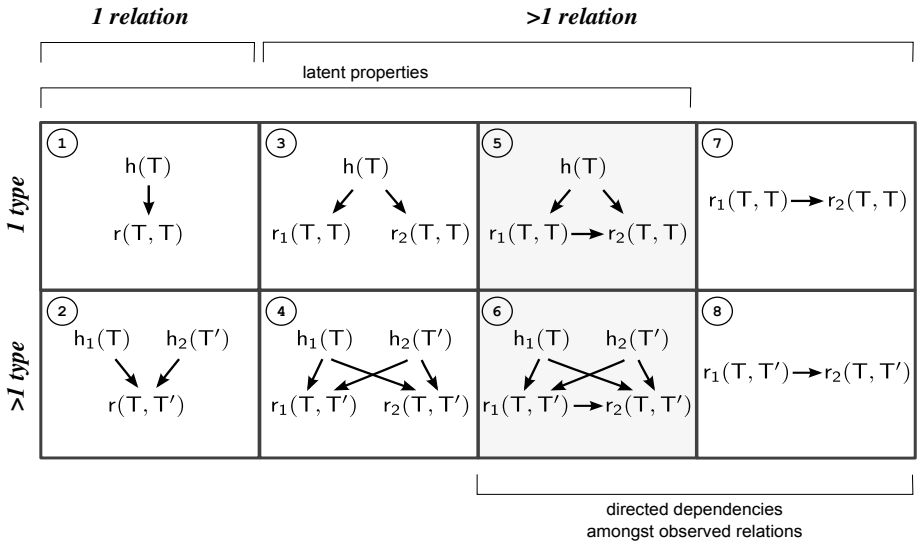


Figure 1.2: Schemas for relational models: (top row) settings where the domain contains only a single type of individuals; (bottom row) domains with multiple types of individuals; (leftmost column) settings where only one relation is observed; (other columns) settings where multiple relations are observed. T, T' denote different types, and h denotes a latent property whilst $r_{(\cdot)}$ denotes an observed relation. The main contribution of this thesis correspond to schemas 5 and 6 (shaded), which subsumes all other schemas shown.

The left-most column of Fig. 1.2 corresponds to problem settings where only one relation is observed, represented by $r(\cdot)$ in the figure. Analysis of networks (Airoldi et al., 2008; Breiger et al., 1975; Chang and Blei, 2010; Handcock et al., 2007; Hofman and Wiggins, 2008; Kok and Domingos, 2007; Xu et al., 2009) as well as collaborative filtering (Hofmann and Puzicha, 1999; Koren, 2008; Marlin and Zemel, 2004; Salakhutdinov and Mnih, 2008; Ungar and Foster, 1998) typically operate in such a setting. A common approach is to model latent properties of individuals to explain the observed relation – an approach often called *clustering* – where the latent property is represented by $h(\cdot)$ in Fig. 1.2. Aside from net-

work analysis and collaborative filtering, similar models have been studied in SRL (Neville and Jensen, 2005; Taskar et al., 2001).

A distinction between models for network analysis and models for collaborative filtering is that network models pertain to a single type of individuals (illustrated by schema 1), e.g. people, whilst collaborative filtering models pertain to two types (illustrated by schema 2), e.g. users and movies. For each different type a latent property can be introduced.

Schemas $3 \sim 8$ apply for the setting where there are multiple observed relations. Traditional approaches for this setting derives from ILP and SRL (De Raedt, 2008; Getoor and Taskar, 2007; Muggleton and De Raedt, 1994; Nienhuys-Cheng and de Wolf, 1997), where the aim is to obtain the best dependency structure over the observed relations that explains the observed relations. This approach pertains to schemas 7 and 8, and algorithms proposed under this approach can generally handle both single and multiple-type domains.

An alternative approach is to model latent properties that simultaneously explains all observed relations, without directed dependencies amongst observed relations. Such an approach corresponds to schemas 3 and 4, and the main proposal implementing these schemas is the infinite relational model (IRM) (Kemp et al., 2006) (a similar model, the infinite hidden relational model (IHRM), was independently proposed by Xu et al. (2006)).

Overall, schemas $1 \sim 4$ denote *hierarchical* models, whilst schemas $5 \sim 8$ are not. Schemas 5 and 6 represent a union of all other schemas in Fig. 1.2, and are to date not implemented in literature. It is argued in Kemp et al. (2006); Sutskever et al. (2010) that modelling dependencies amongst observed relations in addition to latent properties is a promising extension. In this thesis we contribute towards this extension by presenting a unifying representation and learning framework that captures schemas 5 and 6. The proposed framework underpins specific contribution of this thesis, which I outline below along with other contributions.

1. *A unified framework for relational modelling*
   The main contribution of this thesis is a unifying representation and learning framework that can address schemas 5 and 6 shown in Fig. 1.2, and therefore all schemas shown.

Relational models can be naturally expressed using languages based on FOL, e.g. the independent choice logic (ICL) (Poole, 1997), including latent-property models such as well as Kemp et al. (2006) and Xu et al. (2006). In this thesis shows we derive a simple relational probabilistic language – called the latent relational model (LRM) (described in Ch. 2) – and used it to represent all of these models. The LRM captures relations and latent properties, and allows complex dependency structures over the relations and properties.

Given a unifying representational language, algorithms are developed in Ch. 4 to 6 that can learn models in the language. Since the language is rich enough to express models conforming to all schemas shown, and that we can learn any model in the language, our proposed framework enables learning of models that conform to all schemas shown in Fig. 1.2. The combination of latent properties with a dependency-based representation over observed relations enables us to solve modelling problems tackled in the clustering literature (e.g. Airoldi et al. (2008); Hofmann and Puzicha (1999); Kemp et al. (2006); Xu et al. (2006)) as well as the relational learning literature (see De Raedt (2008); Getoor and Taskar (2007)).

Experiments show that our learning algorithm can find models which outperform state-of-the-art latent-property models such as the IRM (Kemp et al., 2006) in probability prediction, as well as dependency-based models without latent properties.

A manuscript relating the general modelling framework proposed is yet to be submitted for publication.

2. *A formal argument for latent relational models*
   In the Sec. 5.4, we have argued for the inclusion of latent properties for better probability predictions, largely from a philosophical standpoint. Also, contributions from the literature have argued for latent properties for explaining relational data, based on intuitive and empirical arguments (Airoldi et al., 2008; Hofman and Wiggins, 2008; Hofmann and Puzicha, 1999; Kemp et al., 2006; Taskar et al., 2001; Ungar and Foster, 1998; Xu et al., 2006). In Ch. 3 we make a contribution by providing a formal argument for why modelling

latent properties can help improve predictive accuracy. Empirical evidence obtained from experiments with synthetic and real-world data supports the theoretical result.

The result is essentially an argument for LRMs, and a journal paper on this result has been accepted subject to minor revisions in the International Journal of Approximation Reasoning.

3. *An approximate expectation maximisation (EM) algorithm for learning latent relational models*
   In Ch. 4 we present a learning algorithm for LRMs. Since learning involves estimation latent properties as well as probability parameters of the model, the standard EM (Dempster et al., 1977) framework is appropriate. However, LRMs typically represent large, complex probabilistic models with many densely-correlated latent random variables. Learning such models is computationally intractable.

   Our algorithm is an approximation of EM that recover computational tractability. Our algorithm also directly ties the computation of latent properties and dependency parameters, which is required for learning models that simultaneously represent the two. The proposed algorithm was presented at the International Workshop for Statistical Relational Learning, July 2009.

4. *Learning latent relational models with unknown size of latent properties*
   In Ch. 5, a search-based method is proposed that extends the algorithm of Ch. 4. The extension accounts for uncertainty over the number of values for latent properties. The proposed procedure searches over the unbounded number of values of latent properties, and bounds the error on the likelihood corresponding at each stage of the search. In turn, bounds on the posterior of the number of values are derived, allowing for averaged predictions that are also accompanied with error bounds. We show how the averaging leads to accurate predictions. Existing relational models that account for uncertainty of the number of latent property values are often known as *nonparametric* relational models (Kemp et al., 2006; Xu et al., 2006), and learning such representations generally rely on Monte Carlo sampling. Our search-based

approach is an alternative to existing approaches. A manuscript for the results of this chapter has yet to be submitted.

5. *A dynamic programming algorithm for learning with functional relations*
   In Ch. 6, we address the situation where relations can probabilistically depend on *functions*[4]. Standard solutions provide either simple models which can predict poorly, or complex models whose number of parameters scale with the number of function values. We propose an approach that optimally partitions the range of the function to trade off these two extremes to obtain a fixed-size representation. The proposed approach yields intermediate-size representations with predictive accuracy that is competitive if not better than the complex model. The results of this chapter was published in the work-in-progress track of the International Conference for Inductive Logic Programming, June 2007.

In summary, the main contribution of this thesis is to provide a framework that unifies representation and learning of latent properties of individuals and complex dependency structures over relations. The two paradigms have, to date, been largely parallel developments. Our framework includes the language (Ch. 2) and learning algorithms (Ch. 4 to 6) sufficient to subsume the two relational learning paradigms. Other contributions of this thesis include (i) a first theoretical argument that supports the inclusion of latent properties in relational models (Ch. 3) for better predictive accuracy; (ii) an empirical demonstration that combining latent properties are not sufficient to achieve the best accuracy, where dependencies amongst observed relations can yield further improvements (Ch. 4); (iii) an extension of our learning algorithm to account for uncertainty about the number of values latent properties can represent (Ch. 5); and finally (iv) a dynamic programming algorithm for learning addressing the special case where functional relations appear in the language, which yields predictive accuracy which surpasses standard solutions for this problem.

[4]Mapping between sets of individuals.

Chapter 2

A Unified Framework for Relational Modelling

This chapter covers the background and notation for this thesis, and introduces the main representational formalism that combines cluster-based and dependency-based representations.

2.1 Background

2.1.1 Notation

In this work we use both predicate logic and probability notations, where the two sets of notations have overlaps. This section develops notation used in this thesis that aims to resolve some confusion due to the overlaps.

Predicate Language

To begin with, *constants* are expressed in lower-case, e.g. *joe* or *venus*, and are used to represent individuals. A *type* is associated with each individual, e.g. *joe* is a *person*. We use $\mathcal{D}(\tau)$ to represent a domain of type τ, which is the set of individuals of type τ. Types are assumed disjoint – that for any pair $\tau_i \neq \tau_j$, $\mathcal{D}(\tau_i) \cap \mathcal{D}(\tau_j) = \emptyset$. A *logical variable* is written in upper-case (e.g. X or $Person$) and denotes some individual. A logical variable is also typed, e.g. $Person$ denotes

some member of $\mathcal{D}(\tau)$.

A *relation* is given by

$$r : \Omega_r \rightarrow V_r$$

where r is the name of the relation, $\Omega_r = \mathcal{D}(\tau_1) \times \ldots, \times \mathcal{D}(\tau_a)$ is the *domain* of the relation, and $T_r = (\tau_1, \ldots, \tau_a)$ is the *type* of the relation. $V_r = \{v_1, \ldots, v_k\}$ is the *range* of the relation – an enumerated set of values not appearing in any domain. Number a and k are positive integers denoting the *arity* and *size* of r; relation r is thus referred to as a k-valued a-ary relation. When $a = 1$, r is a *unary relation*. In this paper, a unary relation is also referred to as a *property*. When $V_r = \{\mathsf{F}, \mathsf{T}\}$, where F, T are Boolean values, r is a Boolean relation. (Note that this description of a relation is more general than defined in standard predicate logic, as we are interested in representing multi-valued relations in addition to Boolean relations.)

A *functional relation* is given by

$$f : \Omega_f \rightarrow R_f$$

where f is the name of the function, $\Omega_f = \mathcal{D}_1(\tau_1) \times \ldots \times \mathcal{D}_{n_f}(\tau_{n_f})$ is the *domain* of the function, $T_f = \{\tau_1, \ldots, \tau_{n_f}$ is the *type* of the function, and $R_f = \mathcal{D}'_1(\tau_1) \times \ldots \times \mathcal{D}'_{m_f}(\tau_{m_f})$ is the *range* of the function (cf. range of standard relation given above). Integer m_f is the arity of f. A function is a mapping between types.

An *atom* is an expression of the form $r(\sigma_1, \ldots, \sigma_a)$ where each σ_i is either a constant or logical variable. The types of $\sigma_1, \ldots, \sigma_a$ must match the type of r. If all of $\sigma_1, \ldots, \sigma_a$ are constants, $r(\sigma_1, \ldots, \sigma_a)$ is a *ground atom*.

A *literal* specifies the value of an atom, e.g. $r(X_1, \ldots, X_a) = v$ where $v \in V_r$. A literal that contains no logical variables, a ground literal, is a proposition. For a Boolean relation r, the literal $r(X_1, \ldots, X_a) = \mathsf{T}$ is written simply as $r(X_1, \ldots, X_a)$, and $r(X_1, \ldots, X_a) = \mathsf{F}$ is written as $\neg r(X_1, \ldots, X_a)$. A literal is also a *formula*.

Formulae with multiple literals are formed using connectives, e.g. $\wedge$ and/or $\vee$. Connecting literals using only $\wedge$ forms a *conjunctive formula* or *conjunction*, e.g. $\neg$*pass*$(Student) \wedge$ *difficulty*$(Course) = high$. A *disjunctive formula* or *disjunction* is formed using only $\vee$, e.g. $\neg$*pass*$(Student) = high \vee \neg$*difficulty*$(Course) = high$.

A substitution is a set $\theta = \{X_1 \backslash x_1, \ldots, X_k \backslash x_k\}$ where X_i are distinct logical variables and x_i are constants. When applied to a formula f, each occurrence of X_i in f is replaced with x_i. We denote the application of substitution of θ to f as $f\theta$. For example, suppose f is $a(X) = u \wedge \neg b(X, Y)$ and $\theta = \{X \backslash x, Y \backslash y\}$, to $f\theta$ is then $a(x) = u \wedge \neg b(x, y)$. If there are no logical variables $f\theta$, θ is called a *grounding substitution*. We also allow substitutions for (sets of) atoms, e.g. for $b(X, Y)\theta$ is $b(x, y)$, and for the set g given by $\{a(X), b(X, Y)\}$, $g\theta$ is $\{a(x), b(x, y)\}$.

Given some formula f containing logical variables $X_1, \ldots, X_n$, where each X_i has type τ_i, let the domain of f be $\Omega_f = \mathcal{D}(\tau_i) \times \ldots, \times \mathcal{D}(\tau_n)$. The *substitution space* of f, Γ_f, is the set of all possible grounding substitutions for f, given by

$$\Gamma_f = \{\{X_1 \backslash x_1, \ldots, X_n \backslash x_n\} : (x_1, \ldots, x_n) \in \Omega_f\}$$

For example, if formula f is $a(X) \wedge b(X, Y)$, where $\Omega_f = \mathcal{D}(\tau_X) \times \mathcal{D}(\tau_Y)$, then Γ_f is $\{\{X \backslash x, Y \backslash y\} : (x, y) \in \Omega_f\}$.

Probability

We are generally interested in models that express uncertainty about the knowledge it represents. Uncertainty is represented by probability, and some basic notation is given as follows.

A random variable is denoted by an upper-case letter, e.g. X, which shares notation with logical variables. The distinction will be made explicit when necessary. The probability distribution for a random variable X is written as $p(X)$. The probability of a random event $X = x$ is normally given by $P(X = x)$, but will be abbreviated to $P(x)$ in this work. Sets of random variables are expressed in bold, e.g. $\mathbf{X}$, and similarly for instantiations, e.g. $\mathbf{x}$. Joint probabilities of sets events are written as $P(\mathbf{X} = \mathbf{x})$, $P(\mathbf{x})$ for short. All random variables are assumed discrete in this work unless stated otherwise.

A univariate distribution is expressed in lower-case, e.g. $p(X)$, and $p(\mathbf{X})$ for a multivariate distribution.

Models used in this work are relational probabilistic models whose random variables correspond to ground atoms of the language. Random variables and

ground atoms are used interchangeably. For example, the notation $P(a(x) = v)$ may be used to denote the probability of random event $a(x) = v$, or $P(a(X) = v)$ to denote the probability of the random event $a(X) = v$ where X denote some individual.

2.1.2 Relational Data

A *dataset* for relation r is a non-empty set $\mathbb{D}_r = \{d_1, \ldots, d_m\}$. Each d_i is a tuple of the form $\langle x_1, \ldots, x_a, v\rangle$ where $(x_1, \ldots, x_a) \in \Omega_r$ and $v \in V_r$. If $\langle x_1, \ldots, x_a, v\rangle \in \mathbb{D}_r$ and $\langle x_1, \ldots, x_a, v'\rangle \in \mathbb{D}_r$, then $v = v'$. A *database* is a set of datasets, where no more than one dataset for each relation. The following defines what a database entails.

$$\begin{array}{lllll} \text{(i)} & \mathbb{D} & \models \mathsf{T} & & \\ \text{(ii)} & \mathbb{D} & \models (r(x_1, \ldots, x_a) = v) & \text{iff } \langle x_1, \ldots, x_a, v\rangle \in \mathbb{D}_r & \\ \text{(iii)} & \mathbb{D} & \models \alpha \wedge \beta & \text{iff } \mathbb{D} \models \alpha \wedge \mathbb{D} \models \beta & \\ \text{(iv)} & \mathbb{D} & \models \alpha \vee \beta & \text{iff } \mathbb{D} \models \alpha \vee \mathbb{D} \models \beta & \end{array} \tag{2.1}$$

where α, β are formulae. We say that r is an *observed relation* if $\mathbb{D}_r \neq \emptyset$, and is a *latent relation* otherwise.

Counts can be obtained from a database $\mathbb{D}$ via logical formulae. The *count* of cases satisfying formula f with respect to $\mathbb{D}$ is given by

$$\#_{\mathbb{D}}(f) = \sum_{\theta \in \Gamma_f} \mathbb{I}(\mathbb{D} \models f\theta) \tag{2.2}$$

where $\mathbb{I}(s)$ is a characteristic function; returns 1 if s holds, and 0 otherwise.

We also define *soft counts* for a given formula $f = l_1 \wedge \ldots \wedge l_n$, where each l_i has the form $r_i(X_1, \ldots, X_{n_i}) = v_i$. Let Q be some marginal probability distribution defined for all $l_i\theta$, $\theta \in \Gamma_f$, where r_i is not observed and $\mathbb{D} \not\models l_i\theta$ (namely q defines a distribution over missing cases), a *soft characteristic function* for the case

$l_i\theta = (r_i(x) = v)$ is given by

$$\hat{\mathbb{I}}(r_i(x) = v, Q) = \begin{cases} 1 & ; r_i \textit{ is observed} \wedge \mathbb{D} \models (r_i(x) = v) \\ 0 & ; r_i \textit{ is observed} \wedge \mathbb{D} \models (r_i(x) = v') \wedge v \neq v' \\ Q(r_i(x) = v) & ; \textit{otherwise} \end{cases} \quad (2.3)$$

Given a logical formula $f = f_1 \wedge \ldots \wedge f_n$, a substitution $\theta \in \Gamma_f$ and some probability function Q defined for all $f\theta'$ where $\theta' \in \Gamma_f$, then a soft characteristic function for f is given by

$$\tilde{\mathbb{I}}(f\theta, Q) = \hat{\mathbb{I}}(f_1\theta, Q) \cdot \hat{\mathbb{I}}(f_2\theta, Q) \cdots \hat{\mathbb{I}}(f_n\theta, Q)$$

Then, a soft count is thus

$$\tilde{\#}_{\mathbb{D}}(f, Q) = \sum_{\theta \in \Gamma_f} \tilde{\mathbb{I}}(f\theta, Q)$$

If all conjuncts of f are observed, then soft count (Eq. 2.1.2) is equivalent to the normal count (Eq. 2.2).

2.1.3 Loss Functions

As alluded to in the introduction (Ch. 1), we are interested in models that will yield the *correct* probability of random events. In statistical terminology, this generally refers to probability estimators whose error is zero in the limit of infinite data. A key measure of accuracy in this work is *empirical loss*, also called empirical risk, which is commonly used for evaluation of statistical estimators.

In general, the true probability of random quantities cannot be observed, but empirical loss can be used to quantify the discrepancy between the estimated probability and the true probability, where true probability is defined as the expected value of data labels in the limit of infinite data. We assume that binary-valued random quantities, as definitions of loss for multi-valued variables remain an open problem.

Rather than specifying the exact loss functions we will use, we describe the general form of empirical loss for binary probability estimation, as required for

our results in Ch. 6. Our definitions are sourced from Shen (2005).

Given some observed label $y \in \{0, 1\}$ (a Bernoulli observation), we are interested in estimating the probability $\gamma = P(y = 1)$. Also, let $\hat{y} \in [0, 1]$ be our estimate for γ.

Let the discrepancy for estimator $\hat{y}$ for cases $y = 1$ and $y = 0$ be non-negative (e.g. the 1-norm $|y - \hat{y}|$), and respectively define *partial losses* to be $l_{1,\hat{y}}$ and $l_{0,\hat{y}}$ which are increasing functions of the discrepancy. The point-wise empirical loss incurred by $\hat{y}$ is

$$\mathbb{L}(y|\hat{y}) \equiv y l_{1,\hat{y}} + (1 - y) l_{0,\hat{y}} \tag{2.4}$$

For a set of labels $\mathbf{y} = \{y_1, \ldots, y_n\}$, the point-wise expected loss or *empirical loss* of $\hat{y}$ is then

$$\mathbf{L}(\hat{y}, \gamma) = \mathbb{E}_{\mathbf{y}} \left[\mathbb{L}(\mathbf{y}|\hat{y})\right] = \gamma l_{1,\hat{y}} + (1 - \gamma) l_{0,\hat{y}} \tag{2.5}$$

Choosing $l_{1,\hat{y}}$ and $l_{0,\hat{y}}$ to be convex functions renders Eq. 2.5 convex. It further obeys Fischer-consistency, namely that $\hat{y} = \gamma$ yields the unique minimum of Eq. 2.5.

A common choice is $l_{1,\hat{y}} = -\log(\hat{y})$ and $l_{0,\hat{y}} = -\log(1 - \hat{y})$, which leads to the commonly-used *log-loss* function. Log-loss corresponds to negative log-likelihood, which we use for evaluating predictive accuracy of our models. Our definitions here are general as required by theoretical results in Ch. 6.

The loss functions we consider are computed assuming that each test case in $\mathbf{y}$ are independent samples. Relational learning has emphasised that the independence assumptions are not appropriate for relational data due to intrinsic correlations amongst data points (Getoor and Taskar, 2007). However, we note that with a relational predictor which accounts for correlations amongst data, we may proceed to make predictions for each test case independent of other test data, thus the use of loss functions of the form of Eq. 2.5 is valid.

2.2 Relational Models

The following covers some well-known languages for expressing relational models, as well as some choices one faces in relational modelling.

2.2.1 Languages

First-order Logic

First-order logic (FOL) has been the standard basis for representing relational models. FOL syntax contain symbols that naturally represent primitive elements of relational domains; constants denote individuals, variables can denote sets of individuals, functional and relational predicates can represent mappings and relational ties between individuals. In addition, complex theories can be formed using formulae (in the form of Eq. **??**) constructed from the available symbols. For example, rules in kinship domains may be captured, e.g.

$$mother(X, Y) \leftarrow female(X) \wedge parent(X, Y).$$
$$grandmother(X, Z) \leftarrow mother(X, Y) \wedge parent(Y, Z).$$

Moreover, FOL can even represent executable programs, e.g. a *member* function that tests whether object X is a member of $List$ by recursing through the list:

$$member(X, List) \leftarrow equal(X, headOf(List)).$$
$$member(X, List) \leftarrow remove\_head(List, newList) \wedge member(X, newList).$$

The flexibility of FOL for expressing a wide range of knowledge is important for modelling relational domains, where complex knowledge structure often arise, e.g. in the biology domain (Srinivasan et al., 1996). A major shortcoming of FOL, however, is that it does not contain the syntax and semantics necessary to capture *quantitative uncertainty* that may be as a result of domain noise.

Bayesian Networks

An important proposal, due to Pearl (1988), is the language of belief networks or *Bayesian networks*. Bayesian networks integrate logical abduction with probability theory, and specifies a compact representation of the joint distribution over a set of propositions (random variables). For a set of random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$, a Bayesian network is a tuple $\langle \mathbf{X}, G, \Theta \rangle$ where G is a directed acyclic graph over $\mathbf{X}$ and Θ are parameters of the network. The network specifies

a joint distribution

$$p(\mathbf{X}) = \prod_i p\left(X_i \mid par(X_i)\right) \tag{2.6}$$

where $par(X)$ represent the set of *parents* (direct ancestors) of X in graph G, and the parameters of each product term (a conditional distribution) is included in Θ. It can be seen that Bayesian networks are compact representations of joint probability distributions, by representing as a product of simpler distributions. Figure 2.1 provides a simple illustration

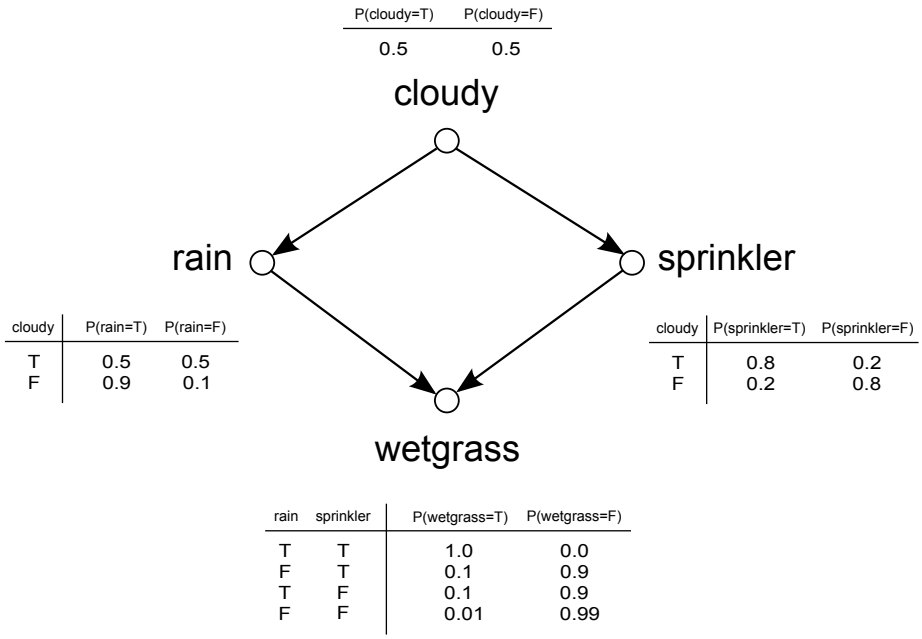


Figure 2.1: A simple Bayesian network for the 'sprinkler' domain. (From *http://bnt.googlecode.com/svn/trunk/docs/usage.html*)

Each conditional probability table (CPT) defines a probabilistic dependency for a node in the network, containing the probabilities of the node conditioned on the state of its parent variables. Each table entry can in turn be expressed by a propositional rule with a probability annotation. For example, the 'wetgrass' table

has the equivalent rule-based representation

$$\begin{array}{lllll} 0.0: & wetgrass & \leftarrow & \neg sprinkler & \wedge \ \neg rain \\ 0.9: & wetgrass & \leftarrow & sprinkler & \wedge \ \neg rain \\ 0.9: & wetgrass & \leftarrow & \neg sprinkler & \wedge \ rain \\ 0.99: & wetgrass & \leftarrow & sprinkler & \wedge \ rain \end{array} \tag{2.7}$$

and similarly for $\neg wetgrass$.

Despite its successes for modelling many real-world problems, the propositional nature of Bayesian networks has drawbacks from the standpoint of relational modelling. Primarily, representing each property and relational tie as a proposition leads to a representation size that is polynomial in the number of individuals in the domain, and the network quickly becomes unmanageable for subsequent learning and reasoning tasks. Consider a relational domain with the relations $likes(X, Y)$ and $funny(X)$ where X and Y denote persons, a corresponding Bayesian network may look like

Although Bayesian networks are not practical for relational domains in this manner, it is crucial as the underlying formalism for relational probabilistic languages developed later. The following section outlines the main contributions.

Probabilistic Relational Languages

Many languages have been proposed which integrate FOL and probability, allowing one to state first-order rules and dependencies as well as probabilities (Jaeger, 1997; Kersting and De Raedt, 2000; Milch et al., 2005; Muggleton, 1995b; Poole, 1993b, 1997, 2000; Richardson and Domingos, 2006; Sato and Kameya, 1997). These languages are collectively referred to as FOPLs.

A notable early contribution was due to Halpern (1990), which allows statistical statements such as

$$\forall X, Y \quad 0.76 : \textit{friends}(X, Y) \leftarrow \textit{smokes}(X) \wedge \textit{smokes}(Y). \tag{2.8}$$

where 0.76 denotes a conditional probability.

Subsequent proposals exploit the conditional independence assumptions of Bayesian

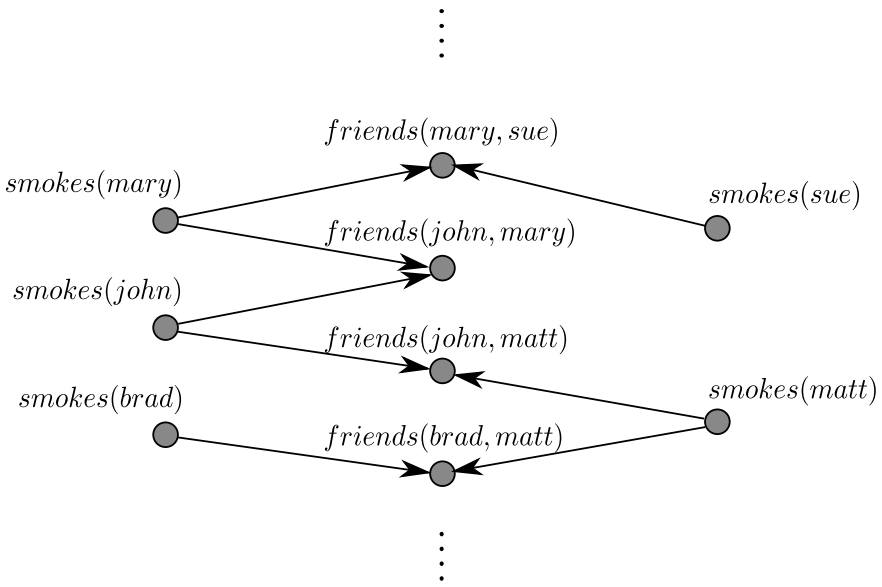


Figure 2.2: A Bayesian network for a simple relational domain modelling relations $likes(X, Y)$ and $funny(X)$, where X and Y denote persons.

networks, leading to FOPLs that are compact representations of Bayesian networks which succinctly represents joint distributions over all propositions entailed by the model. The first to demonstrate this combination was probabilistic Horn abduction (PHA) (Poole, 1993b) (which evolved to ICL (Poole, 1997, 2000)), which has a logic program component containing Horn clauses – where each clause specifies a set of logical *choices* – and a probabilistic component that specifies the probabilistic consequences of these choices. The probabilistic component can be understood as a collection of dependencies that cover possible combinations of choices. For instance, a PHA theory can be used to express the propositional model corresponding to Eq. 2.7, containing dependencies for $wetgrass$ conditioned on choices for $sprinkler$ and $rain$[1]. PRISM (Sato and Kameya, 1997) and is a closely-related language with similar semantics. Other languages that also directly model de-

[1]Note that whilst Eq. 2.7 is defined for propositions, choices can be made in the same way for first-order atoms.

pendencies amongst (first-order) atoms are Bayesian logic programs (BLP) (Kersting and De Raedt, 2000), relational Bayesian networks (RBNs) (Jaeger, 1997) and probabilistic relational models (PRMs) (Friedman et al., 1999). In contrast to PHA/ICL and **PRISM!** (**PRISM!**), these languages do not represent an explicit logic program component.

Whilst the aforementioned languages have probabilistic semantics of Bayesian networks, a well-known exception to this class is the Markov logic network (MLN), which compactly represents Markov networks (undirected probabilistic models) (Richardson and Domingos, 2006). MLNs can be seen as a network of relations in FOL, where dependency links are have no directionality. Associated with each link is a *weight* parameter that indicate the strength of the dependency. For instance, a MLN for the smoking example is given by

$$\forall X \forall Y \quad 1.12 : \mathit{friends}(X, Y) \Leftrightarrow \mathit{smokes}(X) \wedge \mathit{smokes}(Y). \tag{2.9}$$

where 1.12 is real number not restricted to being probabilistic as is the case for parameters in Bayesian networks.

A general point of difference between directed models and undirected models is *local interpretability*. For example, from Eq. 2.8, the proposition $\mathit{friends}(x, y)$ for any pair (x, y) only depends on propositions $\mathit{smokes}(x)$ and $\mathit{smokes}(y)$. In the undirected case (Eq. 2.9), the proposition $\mathit{friends}(x, y)$ necessarily implicates every other proposition in the ground network. This is due to the fact that for directed models, normalization for conditional probabilities is available locally, whilst normalisation is global for undirected networks (see Koller and Friedman (2009)). Although local interpretability is a convenient property, there are examples (e.g. computer vision), where the direction of statistical influence amongst propositions is unclear and are reasonable to model as undirected networks.

Another interesting exception is BLOG (Milch et al., 2005), which combine FOL and probability that covers uncertainty about the number of individuals in the domain. In this work we focus on the case where the number of domain individuals are fixed and known. Interested readers are referred to (Milch et al., 2005) for details. For a good recent survey of FOPLs see Milch and Russell (2007).

2.2.2 Dimensions of in Relational Models

Having described languages that represent relational models, some relevant modelling choices faced in this works are discussed.

Types

Already built into our notation is the notion of *types*. We have assumed a simple type system, where domain individuals are partitioned into disjoint sets, where each set has a designated type, e.g. *animal* or *book*.

A type can be regarded as a property of individuals, and can in principle be expressed as a unary relation in the predicate language, e.g. the fact that x is a bird can be expressed in the language by $is\_bird(x)$. However, in domains where there is clear typing in the domain, a built-in type system help avoid overhead from type-checking when dealing with domain individuals and thus expedite learning and reasoning. A notable example of *typed languages* is PRMs (Friedman et al., 1999), whose primitive components include the notion of *class*, which is essentially a data table recording properties of individuals of a single type.

The simple type system may be inappropriate for many natural domains, e.g. when types may be organised in a hierarchy. For example, *bird* is a sub-type of *animal*. This is particularly true in systems which organise entities into classes and sub-classes. Probabilistic reasoning which account for hierarchical type structures have been explored in (desJardins et al., 2007; Sharma and Poole, 2003).

In addition to hierarchical domains, it can happen that domain individuals can belong to different types. For example, a *graphic novel* is in general regarded as both a *comic* and a *novel*. Being a member of different type domains, it violates the disjointness assumptions of our simple type structure.

To circumvent this problem, a natural solution in an evolving knowledge system is to update the ontology by introducing non-empty type intersections as new types, thereby recovering disjointness. For the current example, *graphic_novel* can be introduced as a new domain type.

In this work we typically consider domains where a simple disjoint type system would suffice. However, our models and learning setting do not preclude the introduction of more complex types in the language.

Uncertainty

There are several types of uncertainty that one encounters in real-world domains. The following discusses key types of uncertainty, and how they may be represented in languages listed above.

In the smoking example (Eq. 2.8), the rule itself, without the probability annotation, is bound to fail on subsets of the population. This is due to the fact that there is noise in the domain, and/or that our knowledge about how friendships are formed is incomplete. Thus, to obtain a more accurate representation of the domain, we also aim to represent the noise. FOPLs described in Sec. 2.2.1 provide a coherent way to do so.

There are more exotic forms of uncertainty, such as *identity uncertainty* (Pasula et al., 2002) and *existence uncertainty* (Milch et al., 2005) that arise frequently. *Identity uncertainty* arises when different individuals have the same identifier, e.g. two people in the phone book having the same name. In this work, however, we make the common assumption that domain individuals are uniquely identified – i.e. the *unique names assumption* (Russell and Norvig, 2009) – in order to avoid identity uncertainty.

Existence uncertainty (Getoor et al., 2002; Milch et al., 2005) occurs when the number of domain individuals is not known in advance, which leads to questions about whether observations are attributes of known or previously unknown individuals. In this work, we make the standard assumption that the domain is known and fixed.

These kinds of uncertainty provide interesting challenges that are beyond the scope of this thesis, but are more appropriate for future investigations.

Dependencies and Rules

Covered in our discussion about Bayesian networks in Sec. 2.2.1 is the notion of rules (a specification of logical choices) versus dependencies which encompass all possible choices. The main purpose of distinguishing dependencies and rules is that the choice impact how learning may be done. For example, rules such as Eq. 2.8 – that friendships are due to smoking – can be learned using data corresponding pairs of smokers. ILP algorithms such as FOIL (Quinlan, 1990) directly induce

rules that cover as many positive data cases and as few negative cases as possible. ILP algorithms can be used to learn dependencies if they all rules entailed by the dependency are supported by examples. If there is no data corresponding to non-smokers, for example, it may be argued that there is insufficient reason to learn rules for non-smokers. By extension, it brings into question the role of dependencies, where some of the rules it encompasses have no empirical support.

The lack of data for non-smokers, however, does not imply that there are no smokers in the world, since our observations are generally incomplete. On these grounds, we consider models that represent dependencies, like BLP (Kersting and De Raedt, 2000) and PRMs (Friedman et al., 1999), rather ones based on rules.

Directed and Undirected Models

FOPLs can represent directed or undirected models (see Sec. 2.2.1), and the difference between directed and undirected models were illustrated. In this work, however, we choose to model directed models, as there are often interesting causal influences that are best captured with directed models. With the help of intervention data (Pearl, 2009), one can test directed hypotheses and discover directions of influence.

Undirected models, however, abstract over the direction of influence and are best for modelling correlations. That said, whether directed or undirected models are favourable depends on the problem at hand, e.g. it may not obvious that smoking is causal to friendships or vice versa (Singla and Domingos, 2008).

Directed models yield simple probabilistic explanations for propositions. Of particular interest to this work is the use of latent unary relations to explain observed relations in a directed way.

Latent Properties and Clusters

Many relational models are known as *clustering* models, where hidden cluster structure of individuals is represented instead of a dependency structure over the observed relations. As mentioned in Ch. 1, proposals from network analysis and collaborative filtering as well as recent works in SRL, e.g. (Airoldi et al., 2008; Hofmann and Puzicha, 1999; Kemp et al., 2006; Ungar and Foster, 1998; Xu et al.,

2006), have investigated the clustering approach. A clustering model for a single relation, such as $rating(User, Movie)$ consists of clusters of users and clusters of movies, and how the value of *rating(x,y)* probabilistically depend on which clusters x and y are assigned. The number of clusters and their membership are probabilistically inferred from observed the *rating* ties. A clustering model for multiple relations applies the same principle.

In this work we represent clusters of individuals via *latent properties*. Namely, each domain individual x has some latent property $\alpha(x)$, where α is a latent *unary* relation (see Sec. 2.1). The value of $\alpha(x)$ equivalent to a cluster assignment, e.g. $\alpha(x) = 1$ means that x is a member of cluster 1. By representing clusters in predicate language, we can thus define a clustering model in predicate language, and directly augment the model with probabilistic dependencies amongst relations to achieve our unifying representation for relational data (see Ch. 1, Sec. 1.3.). Such a model is formally given in Sec. 2.3.

2.3 A Unifying Representation

Here we define latent relational models (LRMs); our language for representing relational models. The language of LRMs is a simplification of the relational probabilistic language independent choice logic (ICL) proposed by Poole (1997)[2], where the logic program component of ICL is omitted. As such, LRMs is strictly a modelling language and not a programming language like ICL.

Definition 1. *Given an input database $\mathbb{D}$, a vocabulary $\Sigma = \langle \mathbf{R}_o \cup \mathbf{R}_l, \mathbf{F}, \mathbf{V} \rangle$ where $\mathbf{R}_o$ and $\mathbf{R}_l$ are sets of observed and latent relations with respect to $\mathbb{D}$ respectively, $\mathbf{F}$ is a set of functions, and $\mathbf{V}$ is a set of logical variables, a LRM is a triple $\mathcal{L} = \langle \mathbf{A}, G, \Theta \rangle$, where*

- $\mathbf{A}$ *is a set of atoms formed using Σ, e.g. $r(X, Y)$ where $r \in \mathbf{R}_o$ and $X, Y \in \mathbf{V}$. We assume that variables appearing in each atom are local to those atoms.*

- *G is a directed-acyclic graph over $\mathbf{A}$, specified as a set of arcs of the form*

[2]Languages such as BLP (Kersting and De Raedt, 2000) and PRISM (Sato and Kameya, 1997) bear strong similarities to ICL.

$c \rightarrowtail d$, *where* $c, d \in \mathbf{A}$.

– Θ *contain conditional probability parameters for each atom in* $\mathbf{A}$. *Each* $\theta \in \Theta$ *is a set of parameters.*

The parents of an atom $a \in \mathbf{A}$ *according to* G *is defined by*

$$Par_G(a) = \{\pi : \pi \in \mathbf{A}, (\pi \rightarrowtail a) \in G\}$$

Each parameter $\theta \in \Theta$ *characterises a conditional probability distribution. For example, the parameter* $\theta_{r(X,Y)} \in \Theta$ *describes the conditional distribution* $p(r(X,Y) \mid Par_G(r))$.

A LRM is illustrated below in Fig. 2.3 for a friendship domain. The LRM contains a relation $friends(X,Y)$ that indicates whether X and Y are friends, and $smoking(X)$ for whether individual X smokes, and there is a parameter for each relation. The LRM is illustrated in the plate notation (Buntine, 1994), in which each rectangle – or *plate* – represent a collection of plate instances. Each plate contains an index variable, and there is a plate instance for each value of the index variable. Indexed symbols in each plate represent a set of random variables, one for each plate instance. For example, in Fig. 2.3, there is a random variable $smoking(x)$ for each individual x, and similarly for $smoking(y)$. Index symbols appearing in intersections of plates yield one random variable per element of cross-product space of the intersecting plates. Such symbols can be regarded as relations, e.g. there is a $friends(x,y)$ variable for every (x,y) pair in the domain. A LRMs can be seen as a generalisation of plate models.

In the grounding, a LRM defines a joint distribution over all ground atoms of each relation in $\mathbf{R}$. First we define the parents for a ground atom.

Definition 2. *Let* $vars(\{a, Par_G(a)\}) = \{X_1, \ldots, X_n\}$ *be logical variables appearing in* a *and its parents, and* $\sigma = \{X_1 \backslash x_1, \ldots, X_n \backslash x_n\}$ *a corresponding substitution, then we define the parents of a ground atom* g_a *of* a *as* $par(g_a) = \sigma Par_G(a)$.

To define the joint distribution represented by a LRM, where $\mathbf{A}$ are relations,

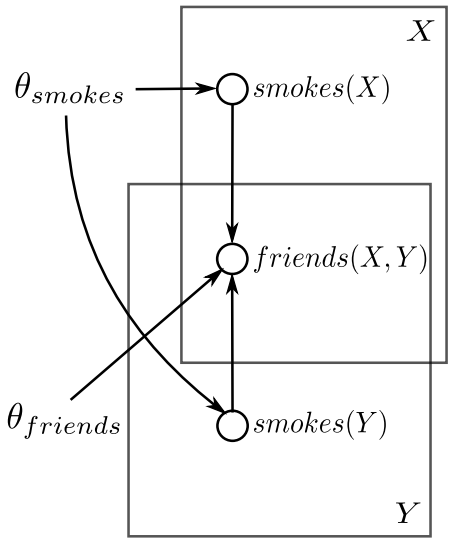


Figure 2.3: A Bayesian network (in plate notation) representing a ground LRM in the smoking-friends domain. Shared parameters for $smokes$ and $friends$ are given by θ_{smokes} and $\theta_{friends}$ respectively.

the set of all ground atoms obtained by grounding relations in $\mathbf{A}$ is given by

$$S_{\mathbf{A}} = \bigcup_{A \in \mathbf{A}} \bigcup_{\sigma \in \Gamma_A} \sigma A$$

where Γ_A is the substitution space of relation A. The joint distribution over all ground atoms of $\mathbf{A}$ is

$$p\left(S_{\mathbf{A}}\right) = \prod_{a \in S_{\mathbf{A}}} p\left(a \mid par(a)\right) \tag{2.10}$$

A key property of FOPLs is that there is parameter sharing amongst ground atoms of the same relation/function, and this property is inherited by the LRM. Suppose the atom $r(X,Y)$ appears in a LRM, and that $p(r(X,Y) \mid Par_G(r(X,Y)))$ is parametrised by θ_r, then all ground instances $r(x,y)$ of $r(X,Y)$ will have condi-

tional probability distributions $p(r(x, y) \mid par(r(x, y)))$ parametrised by θ_r.

The LRM addresses our representational needs, as it directly allows one to model observed and latent relations, and complex structure over the relations.

2.4 Learning Problems

This section describes different relational learning settings as a function of the input to the learning process. Given different settings, existing proposals define various learning problems which we classify using three main characteristics: (i) the number of observed relations given, (ii) the presence of hidden relations, and (iii) a graph structure over relations. These dimensions allow us to summarise key types of relational learning problems addressed in literature. A generic description of learning inputs is described as follows.

Definition 3. ***Inputs***

* A *database* $\mathbb{D}$ over *datasets* $\mathbb{D}_{r_1}, \ldots, \mathbb{D}_{r_n}$, where each relation r_i has a domain composed of elements of $\mathcal{D} = \{\mathcal{D}(\tau_1), \ldots, \mathcal{D}(\tau_p)\}$.
* A *vocabulary* containing:

  A set of logical variables $X_1, \ldots, X_l$, where each variable has type $\tau \in \{\tau_1, \ldots, \tau_p\}$.

  A set of *relational symbols* $R_{obsv} = \{r_1, \ldots, r_n\}$ corresponding to observed relations according to $\mathbb{D}$. Each $r \in R_{obsv}$ has domain $\Omega_r = \mathcal{D}(\tau_1) \times \ldots \times \mathcal{D}(\tau_{n_r})$, where $\mathcal{D}(\tau_i) \in \mathcal{D}$.

  A set of *relational symbols* $R_{latn} = \{\alpha_1, \ldots, \alpha_m\}$ representing latent relations. Each $\alpha \in R_{latn}$ has domain $\Omega_\alpha = \mathcal{D}(\tau_1) \times \ldots \times \mathcal{D}(\tau_{n_\alpha})$, where $\mathcal{D}(\tau_i) \in \mathcal{D}$.

  A set of *function symbols* $F = \{f_1, \ldots, f_k\}$. Each $f \in F$ has domain $\Omega_\alpha = \mathcal{D}(\tau_1) \times \ldots \times \mathcal{D}(\tau_{n_f})$, where $\mathcal{D}(\tau_i) \in \mathcal{D}$.

The generic setting defined here is intended to be general enough to cover those tackled in literature, with the following caveats.

Firstly, for reasons covered in Sec. 2.2.2, we aim to learning models containing only directed dependencies. Secondly, the range of latent relations are assumed to

be known and fixed, thereby fixing the number of parameters of the model. It is generally the case, however, that the best number of parameters defined by the model is not known *a priori*. Allowing the number of parameters to be uncertain in an unbounded range leads to a non-parametric model e.g. (Kemp et al., 2006; Xu et al., 2006). For the purpose of explaining learning problems, we focus on parametric models.

2.4.1 Single Observed Relation

Proposed models for analysing networks and user preferences in collaborative filtering often consider domains where only a single relation is observed. This setting elicits trivial dependency structures amongst relations, whereas learning clusters of individuals to explain the observed relation has been a more common approach (Airoldi et al., 2008; Hofmann and Puzicha, 1999; Kok and Domingos, 2007; Neville and Jensen, 2005; Taskar et al., 2001; Ungar and Foster, 1998; Xu et al., 2009). Clustering models for this setting adhere to schemas 1 and 2 in Fig. 1.2 in Ch. 1.

As explained in Sec. 2.2.2, clustering can be achieved by introducing latent properties (latent unary relations) to represent assignments of individuals to clusters. With input data for the sole observed relation $r(X,Y)$ denoted by $\mathbb{D}_r$ – assuming for now that X, Y are of the same type τ – and let $\alpha : \mathcal{D}(\tau) \to V_\alpha$ be a latent unary relation where $\mathcal{D}(\tau)$ and V_α are known and fixed, we can represent a clustering model as the LRM (Sec. 2.3) $\mathcal{L} = \langle \mathbf{A}, G, \Theta \rangle$, where

$$\begin{aligned}
\mathbf{A} &= \{r(X,Y), \alpha(X), \alpha(Y)\} \\
\Theta &= \{\theta_r, \theta_\alpha\} \\
G &= \{\alpha(X) \rightarrowtail r(X,Y), \alpha(Y) \rightarrowtail r(X,Y)\}
\end{aligned}$$

For the case where X is of type τ_1 and Y is of type τ_2 (e.g. following schema 2 in Fig. 1.2), we introduce a separate latent property for each type, i.e. $\alpha_1 : \mathcal{D}(\tau_1) \to V_{\alpha_1}$ and $\alpha_2 : \mathcal{D}(\tau_2) \to V_{\alpha_2}$. A LRM for this case is $\mathcal{L} = \langle \mathbf{A}, G, \Theta \rangle$

where

$$\mathbf{A} = \{r(X,Y), \alpha_1(X), \alpha_2(Y)\}$$
$$\Theta = \{\theta_r, \theta_{\alpha_1}, \theta_{\alpha_2}\}$$
$$G = \{\alpha_1(X) \rightarrowtail r(X,Y), \alpha_2(Y) \rightarrowtail r(X,Y)\}$$

Figure 2.4 provides an illustration of this LRM. (Note that for the setting where

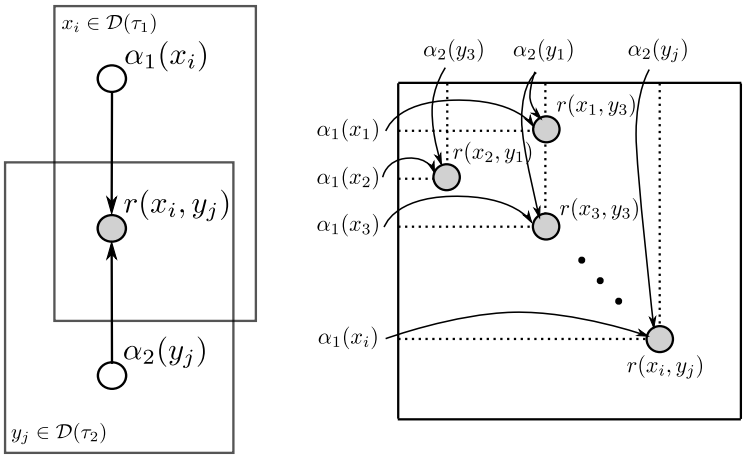


Figure 2.4: A plate representation (left) of a Bayesian network modelling one observed relations $r(X,Y)$, and two latent unary relation $\alpha_1(X)$ and $\alpha_2(Y)$, where $r(X,Y)$ depends on both $\alpha_1(X)$ and $\alpha_2(Y)$. An unrolled (ground) network is shown on the right, with parameters omitted. Shaded nodes are observations.

X, Y are of the same type, a similar illustration can be obtained by appropriately replacing the domains and latent properties.)

An instructive way to view the effects of clustering is that it induces partitions in the space of relational ties. Figure 2.5 below illustrates how observed ties for $r(X,Y)$ can be sorted into "blocks" by clustering domain individuals. The presence of latent properties introduces sets of latent random variables in the ground model, at least one per individual (see Fig. 2.4), and learning necessitates probabilistic inference to calculate the marginal distribution over the latent variables.

The inference problem has a complexity that is exponential in the number of latent random variables, thus exact inference task becomes intractable for all but the smallest domains. Approximate inference, e.g. variational Bayes or Monte Carlo Markov chain (MCMC) inference (see Koller and Friedman (2009)), is then necessary. In Ch. 4 we provide one such method for inference that caters to our general desideratum of learning rich theories. This scenario is true for existing clustering models; the classification/clustering framework of (Taskar et al., 2001) based on PRM language, collaborative filtering models (Hofmann and Puzicha, 1999; Ungar and Foster, 1998), as well as network analysis models (Airoldi et al., 2008).

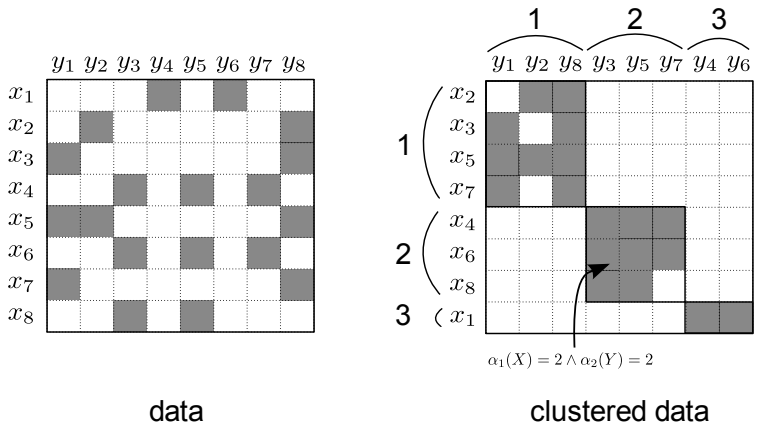


Figure 2.5: An illustration of the partitioning of relational data induced by clustering domain individuals (indices of the along the dimensions of the relation). Data for relation $r(X, Y)$ is partitioned by clusterings represented by unary relations $\alpha_1(X)$ and $\alpha_2(Y)$.

An alternative approach for modelling properties of individuals is based on matrix factorization (Salakhutdinov and Mnih, 2008). The general idea is the observed relation $r(X, Y)$, with domain $\mathcal{D}(\tau_1) \times \mathcal{D}(\tau_2)$ is represented as a $N \times M$ matrix, and expressed as a product of two feature matrices A^T and B. A^T is a $N \times D$ matrix where each row is a coefficient vector for some individual in $\mathcal{D}(\tau_1)$ and B

is a $D \times M$ matrix where each column is coefficient vector for some individual in $\mathcal{D}(\tau_2)$. The learning goal is to find the best D-rank approximation of $r(X, Y)$. The matrix factorization formulation have led to state-of-the-art predictive performance, as demonstrated in the Netflix collaborative filtering competition[3]. Other proposals based on matrix factorization have also demonstrated good predictive accuracy (Marlin and Zemel, 2004; Rennie and Srebro, 2005).

So far we have assumed that the range of values of a latent property is known and fixed *a priori*, i.e. a fixed number of clusters of individuals. Proposals such as the IRM (Kemp et al., 2006), IHRM (Xu et al., 2006), and Bayesian clustered tensor factorisation (BCTF) (Sutskever et al., 2010) (based on the IRM) have been developed such that a distribution is placed over the number of clusters. Specifically, these models embed the Chinese restaurant process (CRP) (Aldous, 1983) – a *nonparametric* prior distribution – as the prior distribution over an unbounded number of clusters. Using MCMC inference, the number of clusters and their membership are generated stochastically.

With the exception of IRM, IHRM, and BCTF, models from the literature discussed in this section do not naturally handle multiple observed relations. We discuss this scenario next.

2.4.2 Multiple Observed Relations

When there are multiple observed relations, three possibilities arise. First, one can learn a dependency-based model without latent properties (see schemas 7 and 8 in Fig. 1.2 in Ch. 1). Second, one can draw on all relations to form a latent-property model (schemas 3 and 4), extending the single-relation problem described in Sec. 2.4.1). Third, combine the two methods to generate a model that represents both clusters and dependencies (schemas 5 and 6). The following sections discuss works related to each schema that involve multiple relations.

Modelling Dependencies Amongst Observed Relations

With more than one observed relation, the traditional approach studied in ILP and SRL can be describes as searching for the best dependency structure over the ob-

[3] www.netflixprize.com

served relations (see schemas 7 and 8 in Fig. 1.2 of Ch. 1). Latent properties or relations are generally not considered except in the context of logical *predicate invention* where new predicates are induced by inverting logical entailment or logical implication (Kramer, 1995; Muggleton, 1994). Statistical counterparts of predicate invention simply refer to clustering (Craven and Slattery, 2001; Kok and Domingos, 2007), which is described in the next section.

The structure of dependencies can be learned by standard ILP/SRL methods (see De Raedt (2008); Friedman et al. (1999); Getoor et al. (2002); Muggleton (2003); Muggleton and De Raedt (1994)). Figure 2.6 shows an example model containing two relations, $s(X,Y)$ and $r(X,Y)$, and a directed dependency from s to r. Note that the plate model in Fig. 2.6 illustrates a LRM $\mathcal{L} = \langle \mathbf{A}, \Theta, G \rangle$ where

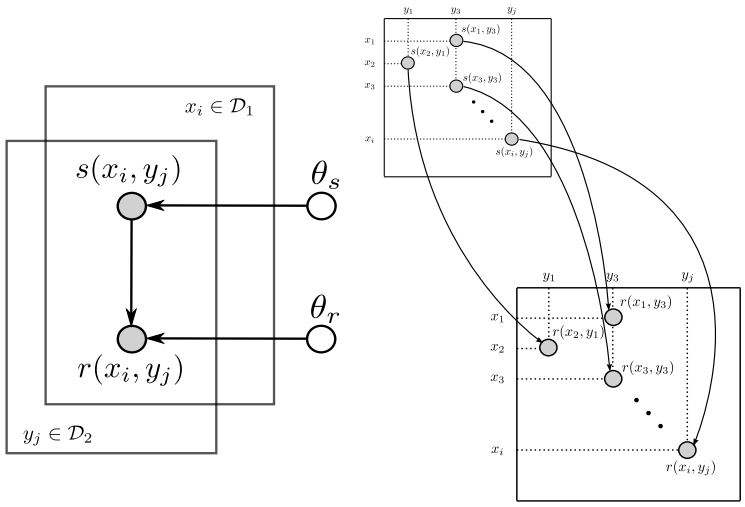


Figure 2.6: A plate representation (left) of a Bayesian network modelling two relations $r(X,Y)$ and $s(X,Y)$, where $r(X,Y)$ depends probabilistically on $s(X,Y)$. An unrolled network, with parameters omitted, is shown on the right. Shaded nodes are observations.

$$\mathbf{A} = \{r(X,Y), s(X,Y)\}$$
$$\Theta = \{\theta_r, \theta_s\}$$
$$G = \{s(X,Y) \rightarrowtail r(X,Y)\}$$

The parameters of the model can be learned through counting. For instance, $\theta_r[u,v]$ is an entry in θ_r that models the conditional probability $P(r(X,Y) = u \mid s(X,Y) = v)$, where $u \in V_r$, $v \in V_s$. Assuming we have some marginal probability distribution Q defined all unobserved ground atoms of $r(X,Y)$ and $s(X,Y)$, a maximum likelihood (ML) value of $\theta_r[u,v]$ is given by

$$\theta_r[u,v] = \frac{\tilde{\#}_{\mathbb{D}}(s(X,Y) = v \wedge r(X,Y) = u, Q)}{\sum_{u' \in V_r} \tilde{\#}_{\mathbb{D}}(s(X,Y) = v \wedge r(X,Y) = u', Q)} \tag{2.11}$$

$\tilde{\#}_{\mathbb{D}}(f,Q)$ denotes the soft count of instances when formula f holds relative to a database $\mathbb{D}$ (see Sec. 2.1.2). The marginal distribution Q is typically learned through EM (Dempster et al., 1977), with modifications for relational models (Getoor and Taskar, 2007; Kameya and Sato, 2000).

A special problem arises for directed relational models when a direct ancestor (or parent) relation involves logical variables not found in the child. The presence of the additional variable(s) in the parents induces a complex dependency in ground model, leading to an unmanageable representation of conditional probability distributions. Figure 2.7 illustrates this scenario where relation $r(X)$ depends on relation $s(X,Y)$.

Figure 2.7 shows that each ground instance $r(x)$ of $r(X)$ depends on the set $S = \{s(x,y) : y \in \mathcal{D}(\tau_2)\}$. The conditional distribution (a table) $p(r(X) \mid s(X,Y))$ thus has a representation size exponential in $|S|$. Compact representation of the conditional probability table can be recovered via aggregation. Illustrating with an example, consider the rule

$$\theta: \quad \textit{likes}(U,M) \leftarrow \textit{won}(M, Award)$$

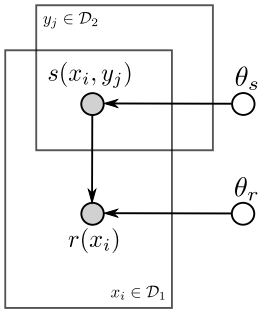


Figure 2.7: A plate representation of a Bayesian network modelling two relations $r(X)$ and $s(X, Y)$, where $r(X)$ depends probabilistically on $s(X, Y)$.

which amounts to

$$\begin{aligned}
\theta^{(1)}: &\quad \textit{likes}(U, M) \leftarrow \textit{won}(M, award_1) \\
\theta^{(2)}: &\quad \textit{likes}(U, M) \leftarrow \textit{won}(M, award_2) \\
\theta^{(3)}: &\quad \textit{likes}(U, M) \leftarrow \textit{won}(M, award_3) \\
&\quad \vdots \\
\theta^{(n)}: &\quad \textit{likes}(U, M) \leftarrow \textit{won}(M, award_n)
\end{aligned}$$

To compute parameter θ, we combine $\theta^{(1)}, \ldots, \theta^{(n)}$ using a combination function (discussed in (Jaeger, 1997; Kersting and De Raedt, 2000)), i.e.

$$\theta = \textit{comb}(\{\theta^{(i)} : i = 1, \ldots, n\})$$

where $comb$ may be the mean function, max, min, noisy-or, or noisy-max and so on. The meaning of the combined parameter depends on the choice of combination function. Suppose the function $\textit{mean}(\cdot)$ is used, then θ reflects the average of probabilities for users liking movies according to different awards. Decomposable aggregation functions have also been studied for probabilistic and relational models (Natarajan et al., 2005; Zhang and Poole, 1996).

Modelling Latent Properties

To yield clusters from multiple observed relations, without defining dependency links between the relations, we place latent unary relations as parents of each relation. Such models correspond to schemas 3 and 4 in Fig. 1.2 (Ch. 1). For a simple case with two observed relations $s(X, Y)$ and $r(X, Y)$ where X, Y are of types τ_1 and τ_2 respectively, Fig. 2.8 illustrates such a clustering model.

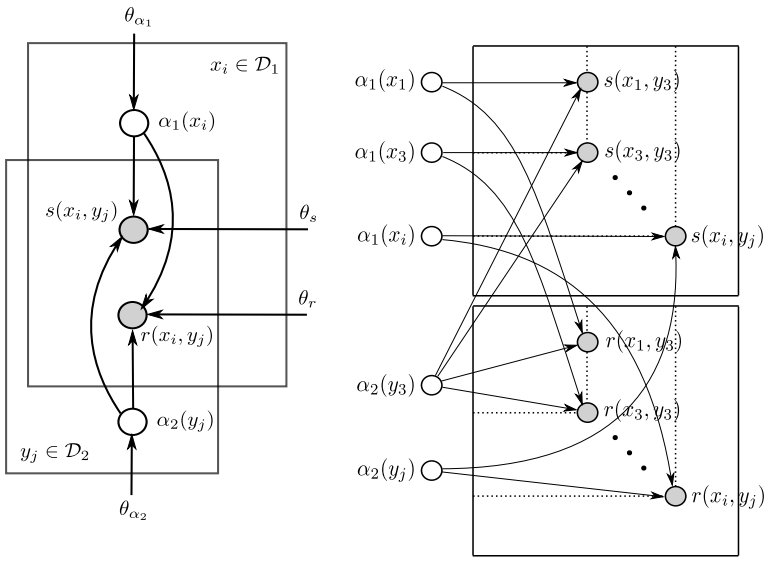


Figure 2.8: A plate representation (left) of a Bayesian network modelling two observed relations $r(X, Y)$ and $s(X, Y)$, and two latent unary relation $\alpha_1(X)$ and $\alpha_2(Y)$. Both $r(X, Y)$ and $s(X, Y)$ depend on both $\alpha_1(X)$ and $\alpha_2(Y)$. An unrolled network, with parameters omitted, is shown on the right. Shaded nodes are observations.

Despite being referred to as a cluster-based representation, dependency links exist from latent relations to the observed relations. The main idea is that whilst no direct links are defined between observed relations $s(X, Y)$ and $r(X, Y)$, statistical signals between them are propagated via latent properties $\alpha_1(X)$ and $\alpha_2(Y)$. It is argued in (Xu et al., 2006) that the latent properties are sufficient to explain the observed relations as well as interdependence between them, without needing explicit dependency structure. The IRM (Kemp et al., 2006) and IHRM (Xu

et al., 2006) are well-known cluster-based representations for multiple observed relations. For the model shown in Fig. 2.8, a corresponding LRM can be written as $\mathcal{L} = \langle \mathbf{A}, G, \Theta \rangle$ where

$$\begin{aligned}
\mathbf{A} &= \{r(X,Y), s(X,Y), \alpha_1(X), \alpha_2(Y)\} \\
G &= \{\alpha_1(X) \rightarrowtail r(X,Y), \alpha_2(Y) \rightarrowtail r(X,Y), \\
&\quad \alpha_1(X) \rightarrowtail s(X,Y), \alpha_2(Y) \rightarrowtail s(X,Y)\} \\
\Theta &= \{\theta_r, \theta_s, \theta_{\alpha_1}, \theta_{\alpha_2}\}
\end{aligned}$$

One interesting property demonstrated by the IRM (Kemp et al., 2006) is that observed relations can also be clustered. The general idea is that observed relations (represented as a matrix) with the same domains viewed as slices of a hypercube. Entries in the hypercube represent data, and the IRM generates a linear set of clusters of these entries via the CRP. Projecting the clusters onto each dimension of the hypercube yields clusters of individuals as well as clusters of relations. Figure 2.9 illustrates this idea

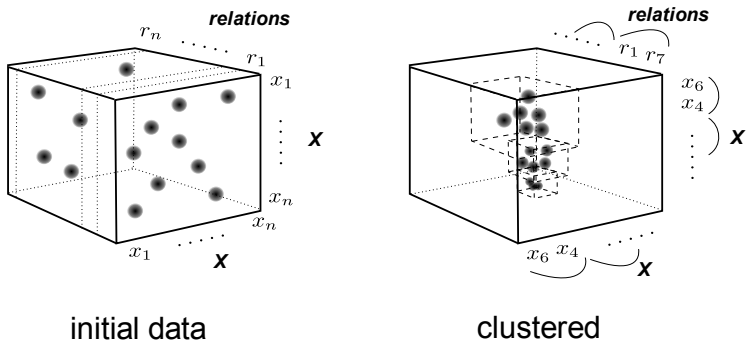


Figure 2.9: Multiple relations, each represented as a two-dimensional slice, are concatenated to form a hypercube. Clustering of individuals and relations can be done by clustering elements of the hypercube (Kemp et al., 2006).

This property of the IRM is exploited in combination with a tensor factorisation approach to yield the Bayesian clustered tensor factorisation (BCTF) (Sutskever et al., 2010). The clustering of relations is intended to achieve an interpretable model about relations, whilst accurate predictions are gained through the tensor factorisation representation – an extension of the matrix factorisation approach.

Whilst representations like the IRM facilitate discovery of interesting clusters of relations – allowing one to learn similarity between relations – explicit dependencies amongst relations can yield information about how relations interoperate to explain the data, e.g. causal links between relations. Representation of dependencies between relations is listed as a desired extension for IRMs (Kemp et al., 2006).

Modelling Dependencies Amongst Observed Relations and Latent Properties

This thesis combines dependency structures over relations with clustering, implementing schemas 5 and 6 depicted in Fig. 1.2. Learning clusters of relations not considered in this work. A schematic of our representation is shown in Fig. 2.10, which differs from Fig. 2.8 by including a dependency link between $s(X, Y)$ and $r(X, Y)$.

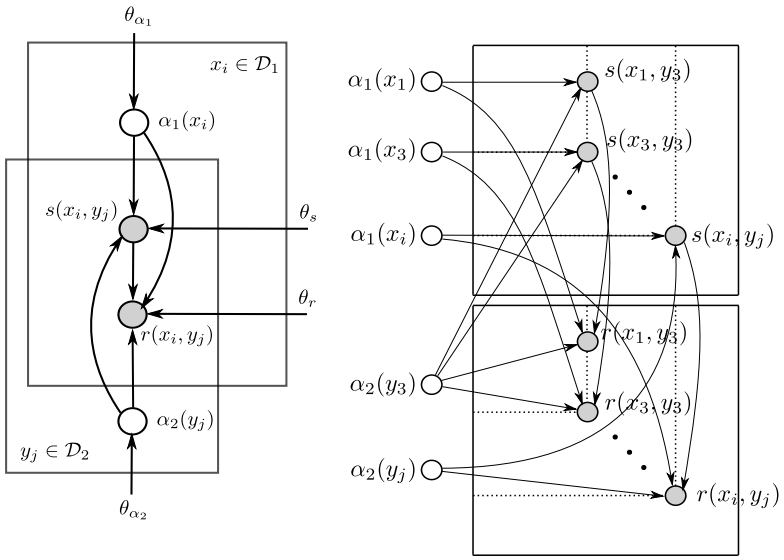


Figure 2.10: A plate representation (left) of a Bayesian network modelling two observed relations $r(X, Y)$ and $s(X, Y)$, and two latent unary relation $\alpha_1(X)$ and $\alpha_2(Y)$. Both $r(X, Y)$ and $s(X, Y)$ depend on both $\alpha_1(X)$ and $\alpha_2(Y)$. Additionally, $r(X, Y)$ also depends on $s(X, Y)$. An unrolled network with parameter omitted is shown on the right. Shaded nodes are observations.

Accurate predictions using such models can be achieved through optimising latent properties of individuals, as demonstrated in Ch. 4. The corresponding

LRM is $\mathcal{L} = \langle \mathbf{A}, G, \Theta \rangle$ where

$$\begin{aligned}
\mathbf{A} &= \{r(X,Y), s(X,Y), \alpha_1(X), \alpha_2(Y)\} \\
G &= \{\alpha_1(X) \rightarrowtail r(X,Y), \alpha_2(Y) \rightarrowtail r(X,Y), \\
&\quad \alpha_1(X) \rightarrowtail s(X,Y), \alpha_2(Y) \rightarrowtail s(X,Y), \\
&\quad s(X,Y) \rightarrowtail r(X,Y)\} \\
\Theta &= \{\theta_r, \theta_s, \theta_{\alpha_1}, \theta_{\alpha_2}\}
\end{aligned}$$

This section has discussed models corresponding to schemas shown in Fig. 1.2, relative to models proposed in literature. We showed that LRMs is expressive enough to implement all of the schemas covered, and thus extends existing proposals.

The next challenge is to develop learning algorithms that can fit LRMs to data. Our proposals for learning are described in Ch. 4, 5 and 6.

Chapter 3

An Argument for Latent Relational Models

This chapter compares relational models composed of observed relations with those that also contain latent properties of individuals. Specifically, the aim is to assess whether the models considered can produce the *correct* probability underlying given queries.

In our analysis we show that both classes of relational models can be interpreted in terms of *reference classes*, and explain the accuracy based on reference classes. Under explicit assumptions about how relations are formed, we show analytically that models with only observed relations incur residual errors in the limit of infinite data, whilst those with latent properties of individuals can represent the correct probability.

Experiments in synthetic domains show that when data is generated according to our assumption, latent-property models dominate. In real-world domains, we show that this is also the case.

3.1 Modelling with Observed Properties and Relations

In this section, we show that relational learning models built from observed features directly construct reference classes. We first describe reference classes.

3.1.1 Reference Classes

A *reference class* is a set of individual or tuples of individuals. It is common to adopt the constraint that a reference class is defined via logical formulae (Bacchus et al., 1996; Kyburg, 1983, 1974; Levi, 1981), e.g. the set of individuals who are *tall* and *athletic*. We also define reference classes in this way, given as follows.

Starting with domains $D = \{\mathcal{D}(\tau_1), \ldots, \mathcal{D}(\tau_n)\}$, let $A = \{a_1, \ldots, a_m\}$ be atoms defined over domains in D; that each atom represents a relation whose domain is composed of elements in D. Let $\mathcal{A} = V_1 \times \ldots \times V_m$ denotes the space of value assignments for the m atoms, where each V_i is the range of the relation for atom a_i. Then, the *basic formula space* is a set of formulae defined over all possible value assignments of the m relations, given by $\mathcal{F} = \{\bigwedge_{i=1}^{m} a_i = u_i : (u_1, \ldots, u_m) \in \mathcal{A}\}$. For any $f \in \mathcal{F}$, the set of tuples that satisfy f is given by

$$S(f) = \{\textit{subs}(\theta) \in \Omega : \theta \in \Gamma_f, \mathbb{D} \models f\theta\} \tag{3.1}$$

which represents all tuples that satisfy f which are entailed by database $\mathbb{D}$. $\textit{subs}(\theta)$ are the constants specified in substitution θ, and Γ_f is the substitution space of f.

Now we define reference classes for the case where only a subset of atoms in A have given values. Let indices $I \subseteq \{1, \ldots, k\}$ where $k \leq m$, and for each a_j where $j \in I$, a value u'_j is given. We can equivalently write this as a formula $f' = \bigwedge_{j \in I} a_j = u'_j$, where $a_j \in A$. Thus, a *restricted formula space* is defined by $\mathcal{F}'(f') = \{(\bigwedge_{i=1}^{m} a_i = u_i) \in \mathcal{F} : \forall j \in I, u_j = u'_j\}$. If $I = \emptyset$, i.e. no set values for any atoms, then $\mathcal{F}' = \mathcal{F}$. A reference class is henceforth given by

$$\mathbb{C}\left(f'\right) = \bigcup_{f \in \mathcal{F}'(f')} S\left(f\right) \tag{3.2}$$

Given a reference class $\mathbb{C}(f')$, the proportion of the reference class such that the formula $h = v$ holds is given by

$$\mathbb{P}(h = v, f') = \frac{|\mathbb{C}(f' \wedge h = v)|}{\sum_u |\mathbb{C}(f' \wedge h = u)|} = \frac{\#_{\mathbb{D}}\left(f' \wedge h = v\right)}{\sum_u \#_{\mathbb{D}}\left(f' \wedge h = u\right)} \tag{3.3}$$

where h is a relation whose domain is composed of elements of D, and v is a value

of the relation. The second equality follows by noting that $|\mathbb{C}(f' \wedge h = v)| \equiv \#_{\mathbb{D}}(f' \wedge h = v)$ by comparing Eq. 3.1 and Eq. 2.2.

We illustrate reference classes with examples. Suppose our domain of discourse is $D = \mathcal{D}(person)$, and we have two Boolean-valued atoms *tall*(X) and *athletic*(X), then the basic formula space is given by

$$\begin{aligned}\mathcal{F} = \{ & \\ & \textit{tall}(X) \wedge \textit{athletic}(X) \\ & \textit{tall}(X) \wedge \neg\textit{athletic}(X) \\ & \neg\textit{tall}(X) \wedge \textit{athletic}(X) \\ & \neg\textit{tall}(X) \wedge \neg\textit{athletic}(X) \\ & \}\end{aligned}$$

Now suppose we have formula $f' = \textit{tall}(X)$, then the restricted formula space is

$$\begin{aligned}\mathcal{F}' = \{ & \\ & \textit{tall}(X) \wedge \textit{athletic}(X) \\ & \textit{tall}(X) \wedge \neg\textit{athletic}(X) \\ & \}\end{aligned}$$

Similarly we can define other restricted formula spaces, e.g. using $f' = \textit{athletic}(X)$ or $f' = \textit{tall}(X) \wedge \textit{athletic}(X)$. The reference class $\mathbb{C}(\textit{tall}(X))$ is then

$$\mathbb{C}(\textit{tall}(X)) = S(\textit{tall}(X) \wedge \textit{athletic}(X)) \cup S(\textit{tall}(X) \wedge \neg\textit{athletic}(X))$$

To consider a second example, let our domains of discourse be $\mathcal{D}(student)$ and $\mathcal{D}(course)$, and atoms *technical*(S) and *require_math*(C) to represent whether student S is technically-minded, and whether course C involves mathematics. The

basic formula space is

$$\begin{aligned}\mathcal{F} = \{ & \\ & \textit{technical}(S) \wedge \textit{require\_math}(C) \\ & \textit{technical}(S) \wedge \neg\textit{require\_math}(C) \\ & \neg\textit{technical}(S) \wedge \textit{require\_math}(C) \\ & \neg\textit{technical}(S) \wedge \neg\textit{require\_math}(C) \\ & \}\end{aligned}$$

Some reference classes may then be defined as follows:

(i) $\mathbb{C}(\mathsf{T})$ – the set of all student-course tuples.

(ii) $\mathbb{C}(\textit{technical}(S))$ – the set of all student-course tuples with only technically-minded students.

(iii) $\mathbb{C}(\textit{require\_math}(C))$ – the set of all student-course tuples with only courses that involve mathematics.

(iv) $\mathbb{C}(\textit{technical}(S) \wedge \textit{require\_math}(C))$ – the set of all student-course tuples with technically-minded students and courses that involve mathematics.

3.1.2 Relational Probabilistic Models

There have been significant contributions to the representation of probabilistic knowledge in relational domains. The contributions are in the form of languages that combine first-order logic structures and probabilities, so that one can express first-order knowledge as well as uncertainty in relational domains (Halpern, 1990; Kersting and De Raedt, 2000; Milch et al., 2005; Muggleton, 1995b; Poole, 1993b, 1997; Richardson and Domingos, 2006; Sato and Kameya, 1997). Whilst all of these are highly expressive languages, accompanying learning methods have been more restricted, and generally cannot cover the entire space of models entailed by the language. In particular, we focus on the fact that most learning methods use only *observed relations*, derived from the input data.

In well-known languages such as the independent choice logic (Poole, 1997), PRISM (Sato and Kameya, 1997), Bayesian logic programs (Kersting and De Raedt,

2000), or Markov Logic Networks (Richardson and Domingos, 2006), models represent probabilistic dependencies or probabilistic rules over relations. Since probabilistic dependencies can be expressed by a set of probabilistic rules, our discussion will be based on such rules. A probabilistic rule is a rule in logic that is annotated by a probability parameter, i.e.

$$\theta : \quad h \leftarrow b_1 \wedge \ldots \wedge b_n \tag{3.4}$$

where $h, b_1, \ldots, b_n$ are literal corresponding to observed relations. h is known as the *head* of the rule, whilst $b_1 \wedge \ldots \wedge b_n$ is the *body*. Parameter θ characterises a some potential $\phi(h, b_1, \ldots, b_n)$ that denotes the uncertainty associated with the rule. For languages that combine first-order logic and probabilistic semantics of Bayesian networks, e.g. (Kersting and De Raedt, 2000; Muggleton, 1995b; Poole, 1993b, 1997; Sato and Kameya, 1997), $\phi(h, b_1, \ldots, b_n)$ represents the conditional probability $P(h \mid b_1, \ldots, b_n)$. Since the languages use discrete symbols, $P(h \mid b_1, \ldots, b_n) = \theta$. We focus on relational models that model such conditional probabilities, and explain how they relate to reference classes.

Depending on the placement of logic variables in a rule, different types of rules can be derived. Whilst there are many sophisticated forms of rules, we focus on the two most common forms: the *constrained* rule and the *range-restricted* rule (Nienhuys-Cheng and de Wolf, 1997).

A constrained rule is one where all logical variables appearing in the body must appear in the head. For example, the rule $\mathit{likes}(U, M) \leftarrow \mathit{drama}(M)$ constrained (U denotes a user, and M denotes a movie). The domain of discourse for a constrained rule is defined by the domain of the head literal, e.g. $\Omega = \mathcal{D}(user) \times \mathcal{D}(movie)$. The body imposes an additional constraint on Ω such that the rule only holds for those $(u, m) \in \Omega$ for which $\mathit{drama}(m)$ is true. The set of all such tuples is denoted here by $\Omega' \subseteq \Omega$. Keeping only tuples in Ω' entailed by the database $\mathbb{D}$ leads to $\Omega'_{\mathbb{D}}$, which is called the *coverset* in the ILP literature. We can immediate see that $\mathbb{C}(\mathit{drama}(M)) \equiv \Omega'_{\mathbb{D}}$, i.e. the coverset of a constrained probabilistic rule *is* a reference class. Further, the maximum likelihood value for θ

is the proportion of $\Omega'_{\mathbb{D}}$ such that *likes*(U, M) is true, namely

$$\theta = \frac{\#_{\mathbb{D}}\left(\textit{likes}(U,M) \wedge \textit{drama}(M)\right)}{\#_{\mathbb{D}}\left(\textit{likes}(U,M) \wedge \textit{drama}(M)\right) + \#_{\mathbb{D}}\left(\neg\textit{likes}(U,M) \wedge \textit{drama}(M)\right)}$$
$$\equiv \mathbb{P}\left(\textit{likes}(U,M), \textit{drama}(M)\right)$$

Henceforth, for any query proposition *likes*(u', m'), if *drama*(m') is true, the probability of *likes*(u', m') being true is ascribed the value θ. This demonstration shows that constrained probabilistic rules directly implement reference classes and that inference also yields the same answer. An illustration of the example used is given in Fig. 3.1 below.

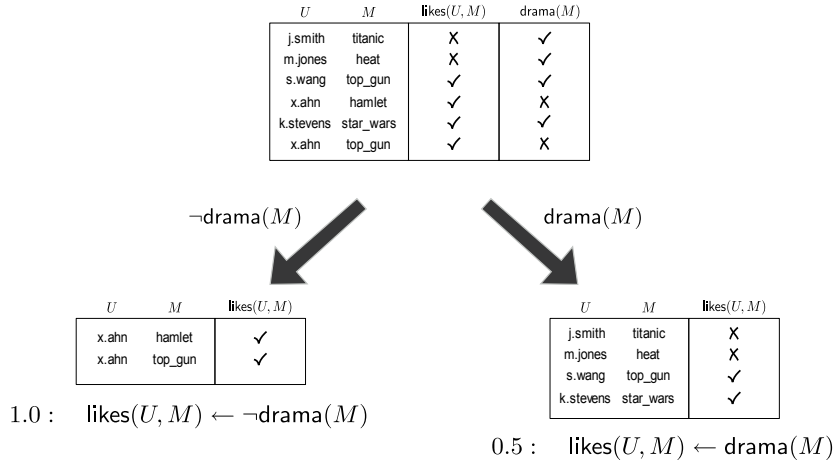


Figure 3.1: A simplified user-movie rating domain, where *likes*(U, M) is a Boolean relation denoting that user U likes movie M. *drama*(M) is an observed Boolean property of movies. The illustration splits observed cases for *likes*(U, M) by values of *drama*(M).

Range-restricted probabilistic rules leads to a more complicated connection with reference classes. A range-restricted rule is one where all variables appearing in the head also appears in the body. In the deterministic setting, logical variables appearing in the body but not in the head are existentially quantified, and the value of the head of the rule is determined by the OR aggregator. For example, the rule $\mathit{likes}(U, M) \leftarrow \mathit{won}(M, Award)$ states that any given user likes movies that has won some award. In the probabilistic setting, however, the goal is to quantify the degree to which the head is true as a probabilistic function of awards the movie has won. This means that the accompanying parameter is an aggregated value. Specifically, the range-restricted rule

$$\theta: \quad \mathit{likes}(U, M) \leftarrow \mathit{won}(M, Award)$$

amounts to

$$\begin{aligned}
\theta^{(1)}: &\quad \mathit{likes}(U, M) \leftarrow \mathit{won}(M, award_1) \\
\theta^{(2)}: &\quad \mathit{likes}(U, M) \leftarrow \mathit{won}(M, award_2) \\
\theta^{(3)}: &\quad \mathit{likes}(U, M) \leftarrow \mathit{won}(M, award_3) \\
&\quad \vdots \\
\theta^{(n)}: &\quad \mathit{likes}(U, M) \leftarrow \mathit{won}(M, award_n)
\end{aligned}$$

Then, parameter θ can be computed from $\theta^{(1)}, \ldots, \theta^{(n)}$ using a combination function (discussed in (Jaeger, 1997; Kersting and De Raedt, 2000)), i.e.

$$\theta = \mathit{comb}(\{\theta^{(i)} : i = 1, \ldots, n\})$$

The meaning of the combined parameter depends on the choice of combination function. Suppose the function $\mathit{mean}(\cdot)$ is used, then θ reflects the average of probabilities for users liking movies according to different awards.

3.2 Modelling Latent Properties of Individuals

Numerous proposals have appeared in different fields that represent and learn unobserved properties to explain observed relations (Airoldi et al., 2008; Handcock et al., 2007; Hofman and Wiggins, 2008; Hofmann and Puzicha, 1999; Koren,

2008). The typical problem setting involves one observed relation, and one unary latent relation to represent unobserved properties for individuals of each domain type (see Ch. 2 Sec. 2.4.1). In the movie-rating problem, for example, the lone observed relation is *likes*(U, M), and latent properties $\alpha(U)$ and $\beta(M)$ model unobserved properties of users and movies respectively.

In general, we can represent express probabilistic rules that involve multiple observed relations and latent properties (denoted by latent unary relations), i.e.

$$\theta : h \leftarrow \underbrace{b_1 \wedge \ldots \wedge b_n}_{\text{observed}} \wedge \underbrace{l_1 \wedge \ldots \wedge l_m}_{\text{latent}}. \tag{3.5}$$

where l_i are literals corresponding to latent relations, which we will assume to be unary relations for now. For the movie-rating domain, we have $n = 1$ (rating relation) and $m = 2$ types (users and movies). Recall that rules such as Eq. 3.5 are encapsulated by probabilistic dependencies of LRMs.

Whilst not immediately clear, LRMs can be interpreted using reference classes. Consider the following illustration of a simple movie-rating domain and the partitioning of observed data for the relation *likes*(U, M) induced by latent properties $\alpha(U)$ and $\beta(M)$.

In Fig. 3.2, the partitioning of *likes*(U, M) is induced by treating $\alpha(U), \beta(M)$ as if they are observed features. As $\alpha(U)$ and $\beta(M)$ are not actually observed, their respective columns in the data table must be inferred. As such, it is clear that every full specification of latent feature values induces a different partitioning of data at the leaves. Each leaf partition is a reference class.

Given a partitioning, inference subsequently follows direct inference, e.g. the probability that *likes*$(j.smith, titanic)$ given that $\alpha(j.smith) = \mathsf{T}$ and $\beta(titanic) = \mathsf{F}$ is 1. In drawing connection to reference classes, we can see that latent-feature models effectively allow one to optimise over possible reference classes by estimating the best latent feature values.

It is worthwhile noting that latent features can be inferred as probabilities, in which case inference yields a marginal probability. For instance, suppose that $\hat{p}(\alpha(j.smith)) = (0.3, 0.7)$ is the inferred distribution for $\alpha(j.smith)$ and $\hat{p}(\beta(titanic)) =$

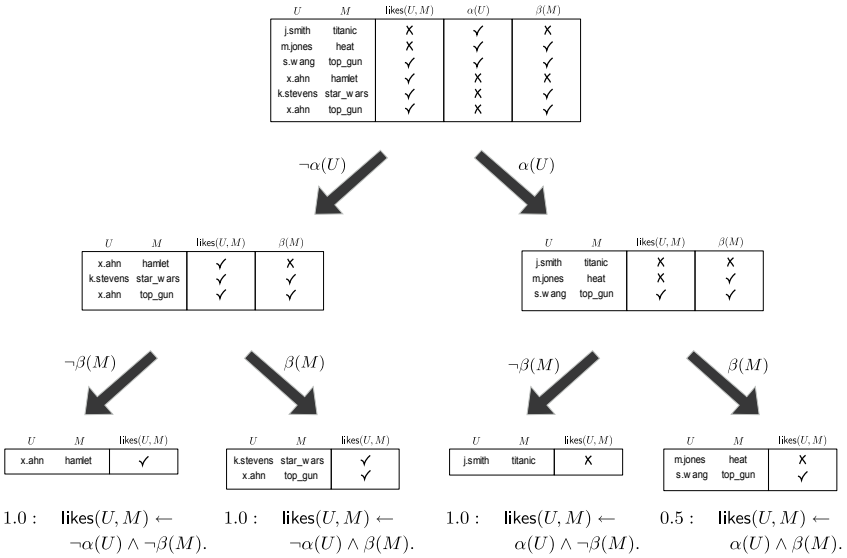


Figure 3.2: A simplified user-movie rating domain, where *likes*(U, M) is a Boolean relation denoting that user U likes movie M. $\alpha(U)$ is a Boolean unary relation representing some (hidden) property of user U, and $\beta(M)$ is similarly defined for movies. The illustration splits observed cases for *likes*(U, M) as if α and β are observed. Probabilistic rules for each partition are also show, with probability parameters indicating the proportion where *likes*$(U, M) = \mathsf{T}$.

$(0.15, 0.85)$ for $\beta(titanic)$, then

$$\begin{aligned}
&\hat{P}\left(\textit{likes}(j.smith, titanic)\right) \\
&= \hat{P}\left(\textit{likes}(j.smith, titanic) \mid \alpha(j.smith), \beta(titanic)\right) \times 0.7 \times 0.85 + \\
&= \hat{P}\left(\textit{likes}(j.smith, titanic) \mid \alpha(j.smith), \neg\beta(titanic)\right) \times 0.7 \times 0.15 + \\
&= \hat{P}\left(\textit{likes}(j.smith, titanic) \mid \neg\alpha(j.smith), \beta(titanic)\right) \times 0.3 \times 0.85 + \\
&= \hat{P}\left(\textit{likes}(j.smith, titanic) \mid \neg\alpha(j.smith), \neg\beta(titanic)\right) \times 0.3 \times 0.15
\end{aligned}$$

where $\hat{P}(\textit{likes}(j.smith, titanic) \mid \alpha(j.smith), \beta(titanic))$ corresponds to the pa-

rameter value θ for the rule

$$\theta : \quad \mathit{likes}(U, M) \leftarrow \alpha(U) \wedge \beta(M)$$

and similarly for other values of α and β. In this setting, inference is simply a weighted sum of reference class estimates.

One contribution of latent features is that it is equivalent to introducing *disjunctions of inequalities* in the language for defining reference classes. Specifically, it allows statements such as

$$\begin{aligned} 1.0 : \quad \mathit{likes}(U, M) \leftarrow & ((U = k.stevens) \vee (U = x.ahn)) \\ & ((M = star\_wars) \vee (M = top\_gun)) \end{aligned}$$

which corresponds to a leaf in Fig. 3.2. Constituents of each disjunction are determined by values α and β for each individual, as shown in the partition tree in Fig. 3.2. An advantage arises as values of latent features α and β can be optimised, thus allowing optimisation of disjunctions to best explain the data.

So far, we have assumed that inferred values of latent properties are deterministic (i.e. a *hard-clustering model*), where it is also possible that they take probability values (a *soft-clustering model*). Namely, for some x and y, instead of inferring the most likely value of $\alpha(x)$ and $\beta(y)$, one can infer a distribution over possible values of $\alpha(x)$ and $\beta(y)$. Computing soft-clustering models can be seen as computing a distribution over possible hard-clustering models. To consider the simplest case where we cluster a single individual x, ascribing the probability of 0.7 to $\alpha(x) = \mathsf{T}$ is equivalent to ascribing a probability of 0.7 to the hard-clustering model where $\alpha(x) = \mathsf{T}$, and 0.3 to the model where $\alpha(x) = \mathsf{F}$.

In the next section, we compare inferences made by observed-feature and latent-feature relational models, in the context of recovering the true underlying probabilities of propositions.

3.3 Understanding Relational Inference

The connections between relational models and reference classes means that relational models are confronted with the reference class problem (Reichenbach,

1949). Although much focus has been given to the justifiability of reference classes and direct inference, as well as the rationality behind the identification of reference classes, we seek a more direct evaluation of whether a given relational model can reproduce the right probability for given propositions. We take a pragmatic approach by assuming that the true probabilities of propositions are known. That is, we explicitly assume a particular generative process of relational data which harbours the true probabilities.

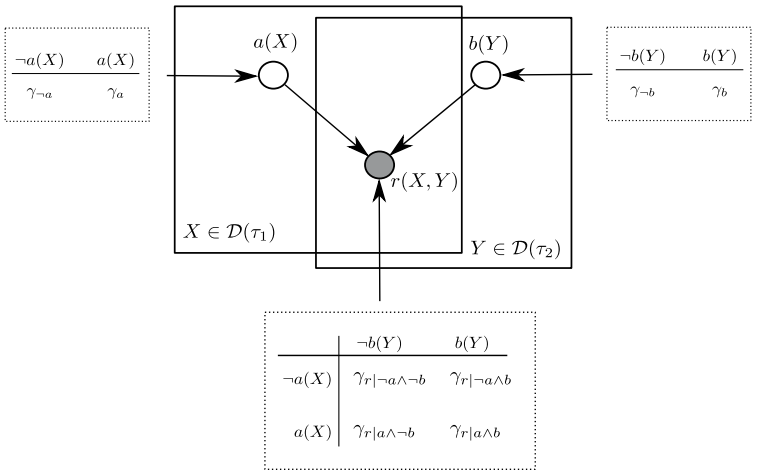
3.3.1 Generative Model

Our generative process reflects the common intuition that relations amongst individuals are attributed to (hidden) properties of participating individuals. We focus on the simplest possible example of such a generative process, and generate data that is used for learning. We assess observed-feature and latent-feature relational models learned from this data for recovering the correct probabilities, i.e. those defined in the generative process.

Definition 4. *We assume a Boolean relation* $r : \mathcal{D}(\tau_1) \times \mathcal{D}(\tau_2) \rightarrow \{\mathsf{F}, \mathsf{T}\}$*, where domains* $\mathcal{D}(\tau_1), \mathcal{D}(\tau_2)$ *are known, and two unary Boolean relations:* $a : \mathcal{D}(\tau_1) \rightarrow \{\mathsf{F}, \mathsf{T}\}$ *and* $b : \mathcal{D}(\tau_2) \rightarrow \{\mathsf{F}, \mathsf{T}\}$ *(the range of* a *and* b *will be called* V_a *and* V_b *respectively in the rest of this chapter). Each ground atom* $a(x)$ *of* $a(X)$ *is a Bernoulli variable with parameter* γ_a *corresponding to the probability of* $a(x) = \mathsf{T}$*. Similarly, each ground atom* $b(x)$ *of* $b(X)$ *has Bernoulli parameter* γ_b*. Each ground atom* $r(x, y)$ *of* $r(X, Y)$ *is also a Bernoulli variable, with conditional probability parameter* $\gamma_{r|a=u,b=v}$ *for the probability of* $r(x, y) = \mathsf{T}$ *given* $a(x) = u$ *and* $b(y) = v$*. We call the model* $\mathcal{G}$ *in the remainder of this chapter.* $\mathcal{G}$ *represents a Bayesian network which defines the joint distribution*

$$J = \prod_{x \in \mathcal{D}(\tau_1)} p_{\mathcal{G}}\left(a(x)\right) \prod_{y \in \mathcal{D}(\tau_2)} p_{\mathcal{G}}\left(b(y)\right) p_{\mathcal{G}}\left(r(x, y) \mid a(x), b(y)\right) \tag{3.6}$$

The network and its parameters are illustrated in Fig. 3.3. We call this model $\mathcal{G}$ *in the remainder.*



$$
\begin{aligned}
P_{\mathcal{G}}\left(a(X)\right) &= \gamma_a \\
P_{\mathcal{G}}\left(b(Y)\right) &= \gamma_b \\
P_{\mathcal{G}}\left(r(X,Y)|a(X) \wedge b(Y)\right) &= \gamma_{r|a\wedge b} \\
P_{\mathcal{G}}\left(r(X,Y)|a(X) \wedge \neg b(Y)\right) &= \gamma_{r|a\wedge\neg b} \\
P_{\mathcal{G}}\left(r(X,Y)|\neg a(X) \wedge b(Y)\right) &= \gamma_{r|\neg a\wedge b} \\
P_{\mathcal{G}}\left(r(X,Y)|\neg a(X) \wedge \neg b(Y)\right) &= \gamma_{r|\neg a\wedge\neg b}
\end{aligned}
$$

Figure 3.3: A generative model for a hypothetical relational domain shown as a parametrised Bayesian network. The structure is shown on the left, and parameters of the model are shown on the right. Logical variables X, Y have different types.

3.3.2 Sampling from $\mathcal{G}$

We assume that $r(X,Y)$ is an observed relation whilst $a(X), b(Y)$ are latent relations. Thus, our dataset will consist of only $\mathbb{D}_r$, which is generated using our generative model (Def. 4) as follows. First assume we have two domains $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$. Then for every $x \in \mathcal{D}(\tau_1)$, randomly generate the value for $a(x)$ by sampling the Bernoulli distribution with parameter γ_a. Similarly, for every $y \in \mathcal{D}(\tau_2)$, randomly generate the value for $b(y)$ by sampling the Bernoulli distribution with parameter γ_b. Then, for every pair $(x,y) \in \mathcal{D}(\tau_1) \times \mathcal{D}(\tau_2)$, where $a(x) = u$ and

$b(y) = v$ are previously sampled values, randomly generate the value for $r(x, y)$ according to a Bernoulli distribution with parameter $\gamma_{r|a=u,b=v}$. We can simulate missing-at-random data by randomly removing generated cases of $r(X, Y)$ *post hoc*.

3.3.3 Inference with Reference Classes

Under the simple problem setting where data is generated according to $\mathcal{G}$ (Def. 4), this section examines inference with observed-feature models. Since we have only one observed relation $r(X, Y)$, relational probabilistic models described in Sec. 3.1.2 for this problem are described by probabilistic rules with empty bodies, i.e.

$$\theta : \quad r(X, Y) \leftarrow .$$

The equivalent reference class is $\mathbb{C}(\mathsf{T})$ which consists of all tuples $(x, y) \in \Omega_r$ that is entailed by the database. The reference class statistic $\mathbb{P}(r, \mathsf{T})$ is simply the proportion where r is true for tuples in $\mathbb{C}(\mathsf{T})$.

For given queries, specific reference classes can be obtained by specialising on the identity of individuals in the queries. For instance, for the query $r(x', y')$, reference classes $\mathbb{C}(X = x')$ and $\mathbb{C}(X = y')$ may be used. For our analysis of observed-feature models, we consider the reference classes $\mathbb{C}(\mathsf{T})$, $\mathbb{C}(X = x')$, and $\mathbb{C}(X = y')$.

The main result we show is that corresponding reference class statistics $\mathbb{P}(r, \mathsf{T})$, $\mathbb{P}(r, X = x')$, and $\mathbb{P}(r, Y = y')$ are approximations of marginal probabilities of $\mathcal{G}$. We first list the marginal distributions of $\mathcal{G}$ below.

Marginalise out both $a(X)$ and $b(Y)$ from $\mathcal{G}$ leads to the first marginal

$$\begin{aligned}
&P_{\mathcal{G}}(r(X, Y)) \\
&= \sum_u \sum_v P_{\mathcal{G}}(r(X, Y)|a(X) = u, b(Y) = v) P_{\mathcal{G}}(a(X) = u) P_{\mathcal{G}}(b(Y) = v) \\
&= \begin{array}{llll}
\gamma_{r|a\wedge b} \times \gamma_a \gamma_b & + & \gamma_{r|\neg a\wedge b} \times (1 - \gamma_a)\gamma_b & + \\
\gamma_{r|a\wedge\neg b} \times \gamma_a(1 - \gamma_b) & + & \gamma_{r|\neg a\wedge\neg b} \times (1 - \gamma_a)(1 - \gamma_b) &
\end{array}
\end{aligned} \tag{3.7}$$

where $u, v \in \{\mathsf{F}, \mathsf{T}\}$ are Boolean values. Marginalising over $b(Y)$ only yields

$$P_{\mathcal{G}}(r(X,Y)|a(X)) = \gamma_{r|a\wedge b}\gamma_b + \gamma_{r|a\wedge\neg b}(1-\gamma_b) \tag{3.8}$$

$$P_{\mathcal{G}}(r(X,Y)|\neg a(X)) = \gamma_{r|\neg a\wedge b}\gamma_b + \gamma_{r|\neg a\wedge\neg b}(1-\gamma_b) \tag{3.9}$$

Finally, marginalising $a(X)$ gives

$$P_{\mathcal{G}}(r(X,Y)|b(Y)) = \gamma_{r|a\wedge b}\gamma_a + \gamma_{r|\neg a\wedge b}(1-\gamma_a) \tag{3.10}$$

$$P_{\mathcal{G}}(r(X,Y)|\neg b(Y)) = \gamma_{r|a\wedge\neg b}\gamma_a + \gamma_{r|\neg a\wedge\neg b}(1-\gamma_a) \tag{3.11}$$

To show that $\mathbb{P}(r, \mathsf{T})$, $\mathbb{P}(r, X = x')$, and $\mathbb{P}(r, Y = y')$ approximate Eq. 3.8 to Eq. 3.11, observe that each datum in the dataset $\mathbb{D}_r$ is sampled from $\mathcal{G}$ with four different conditional distributions: $p_{\mathcal{G}}(r(X,Y) \mid \neg a(X), \neg b(Y))$, $p_{\mathcal{G}}(r(X,Y) \mid \neg a(X), b(Y))$, $p_{\mathcal{G}}(r(X,Y) \mid a(X), \neg b(Y))$ and $p_{\mathcal{G}}(r(X,Y) \mid a(X), b(Y))$. Grouping the data according to their generative distributions we may rewrite $\mathbb{D}_r$ as a union of the partitions, namely $\mathbb{D}_r = D_{\neg a,\neg b} \cup D_{\neg a,b} \cup D_{a,\neg b} \cup D_{a,b}$.

For partition $D_{a,b}$, there are $n^+_{a,b}$ cases where $r(X,Y) = \mathsf{T}$, and $n^-_{a,b}$ cases for $r(X,Y) = \mathsf{F}$. The total number of cases is $n_{a,b} = |D_{a,b}| = n^+_{a,b} + n^-_{a,b}$. Similar counts are defined for other partitions of $\mathbb{D}_r$. Given these counts, we can then express reference class estimate $\mathbb{P}(r = w, \mathsf{T})$, which implicates all of $\mathbb{D}_r$, as follows

$$\begin{aligned}\mathbb{P}(r,\mathsf{T}) &= \frac{n^+_{\neg a,\neg b} + n^+_{a,\neg b} + n^+_{\neg a,b} + n^+_{a,b}}{N} \\ &= \frac{n^+_{\neg a,\neg b}}{n_{\neg a,\neg b}}\frac{n_{\neg a,\neg b}}{N} + \frac{n^+_{a,\neg b}}{n_{a,\neg b}}\frac{n_{a,\neg b}}{N} + \frac{n^+_{\neg a,b}}{n_{\neg a,b}}\frac{n_{\neg a,b}}{N} + \frac{n^+_{a,b}}{n_{a,b}}\frac{n_{a,b}}{N}\end{aligned} \tag{3.12}$$

By matching terms in Eq. 3.12 with those in Eq. 3.7, one can see that reference class proportion $\mathbb{P}(r, \mathsf{T})$ is simply an empirical approximation of Eq. 3.7.

For the reference class $\mathbb{C}(X = x')$ consists of $r(X,Y)$ such that $X = x'$. Each case $r(x', y)$ has value T with probability $\gamma_{a=u,b=v}$, where $a(x') = u$ and $b(y) = v$ happens to be true as a result of the generative process. Let $n(x') = |\mathbb{C}(X = x')|$ be the number of cases in $\mathbb{C}(X = x')$, $n^+(x')$ the number of cases with value T, and $n^-(x')$ the number of cases with value F. We further split $\mathbb{C}(X = x')$ into

two set, corresponding to $b(Y) = \mathsf{T}$ and $b(Y) = \mathsf{F}$. We then have counts $n_b(x')$, $n_b^+(x')$, $n_b^-(x')$, and $n_{\neg b}(x')$, $n_{\neg b}(x')$, and $n_{\neg b}(x')$. The reference class statistic $\mathbb{P}(r, X = x')$ can be expressed as

$$\begin{aligned}
\mathbb{P}\left(r, X = x'\right) &= \frac{n_b^+(x') + n_{\neg b}^+(x')}{n(x')} \\
&= \frac{n_b^+(x')}{n_b(x')}\frac{n_b(x')}{n(x')} + \frac{n_{\neg b}^+(x')}{n_{\neg b}(x')}\frac{n_{\neg b}(x')}{n(x')} \\
&= \frac{\frac{n_b^+(x')}{n_b(x')}\frac{n_b(x')}{N}}{\frac{n(x')}{N}} + \frac{\frac{n_{\neg b}^+(x')}{n_{\neg b}(x')}\frac{n_{\neg b}(x')}{N}}{\frac{n(x')}{N}} \\
&= \frac{\frac{n_b^+(x')}{n_b(x')}\frac{n(x')}{N}\frac{n_b}{N}}{\frac{n(x')}{N}} + \frac{\frac{n_{\neg b}^+(x')}{n_{\neg b}(x')}\frac{n_{\neg b}(x')}{N}\frac{n_{\neg b}}{N}}{\frac{n(x')}{N}} \\
&= \frac{n_b^+(x')}{n_b(x')}\frac{n_b}{N} + \frac{n_{\neg b}^+(x')}{n_{\neg b}(x')}\frac{n_{\neg b}}{N}
\end{aligned} \tag{3.13}$$

where n_b is the count of cases where $b(Y)$ is true. Following the same derivation we obtain $\mathbb{P}(r, Y = y')$

$$\mathbb{P}\left(r, Y = y'\right) = \frac{n_a^+(y')}{n_a(y')}\frac{n_a}{N} + \frac{n_{\neg a}^+(y')}{n_{\neg a}(y')}\frac{n_{\neg a}}{N} \tag{3.14}$$

Using Eq. 3.13 and 3.14, the following theorem gives conditions when reference class statistics can model the correct probability.

Theorem 1. Let $r(X, Y)$ be a Boolean relation where X, Y are of types τ_1 and τ_2 respectively. Suppose $r(X, Y)$ is the sole observed relation with respect to a database $\mathbb{D}$, with dataset $\mathbb{D}_r$ generated using the generative model $\mathcal{G}$ (see Def. 4). Assume that $\gamma_a \in (0, 1)$ and $\gamma_b \in (0, 1)$ (i.e., they are non-extreme probability values).

For any $(x', y') \in \mathcal{D}(\tau_1) \times \mathcal{D}(\tau_2)$, the true probability of $r(x', y') = \mathsf{T}$ is $\mu = \gamma_{r|a=u \wedge b=v}$, where the generative process gives $a(x') = u$ and $b(y') = v$.

Consider the limit as $N = |\mathbb{D}_r|$ approaches ∞. It follows that

$$\begin{array}{lll} (i) & \lim_{N\to\infty} \mathbb{P}(r, \mathsf{T}) = \mu & \textit{iff} \quad \gamma_{r|a\wedge b} = \gamma_{r|a\wedge\neg b} = \gamma_{r|\neg a\wedge b} = \gamma_{r|\neg a\wedge\neg b} \\ (ii) & \lim_{N\to\infty} \mathbb{P}(r, X = x') = \mu & \textit{iff} \quad \big(\gamma_{r|a\wedge b} = \gamma_{r|a\wedge\neg b}\big) \wedge \big(\gamma_{r|\neg a\wedge b} = \gamma_{r|\neg a\wedge\neg b}\big) \\ (iii) & \lim_{N\to\infty} \mathbb{P}(r, Y = y') = \mu & \textit{iff} \quad \big(\gamma_{r|a\wedge b} = \gamma_{r|\neg a\wedge b}\big) \wedge \big(\gamma_{r|a\wedge\neg b} = \gamma_{r|\neg a\wedge\neg b}\big) \end{array} \tag{3.15}$$

where $\gamma_{\cdot|\cdot}$ are parameters of the generative model $\mathcal{G}$.

Proof. To begin with, $\mu = \gamma_{r|a=u\wedge b=v}$ is the true probability of the query $r(x', y') = \mathsf{T}$, where $a(x') = u$ and $b(y') = v$ is given by the generative process $\mathcal{G}$. The observed cases of $r(X, Y)$ is given by $\mathbb{D}_r$, and let $N = |\mathbb{D}_r|$. To evaluate whether reference class statistics $\mathbb{P}(r, \mathsf{T})$, $\mathbb{P}(r, X = x')$ and $\mathbb{P}(r, Y = y')$ can yield μ, we consider the limit as N approaches ∞.

The reference class statistic $\mathbb{P}(r, \mathsf{T})$ has the limiting value

$$\lim_{N\to\infty} \mathbb{P}(r, \mathsf{T}) = \gamma_{r|a\wedge b}\gamma_a\gamma_b + \gamma_{r|a\wedge\neg b}\gamma_a\gamma_{\neg b} + \gamma_{r|\neg a\wedge b}\gamma_{\neg a}\gamma_b + \gamma_{r|\neg a\wedge\neg b}\gamma_{\neg a}\gamma_{\neg b}$$

and to determine when $\lim_{N\to\infty} \mathbb{P}(r, \mathsf{T}) = \mu$, directly solve

$$\gamma_{r|a\wedge b}\gamma_a\gamma_b + \gamma_{r|a\wedge\neg b}\gamma_a\gamma_{\neg b} + \gamma_{r|\neg a\wedge b}\gamma_{\neg a}\gamma_b + \gamma_{r|\neg a\wedge\neg b}\gamma_{\neg a}\gamma_{\neg b} = \mu \tag{3.16}$$

Since μ has one of $\gamma_{r|a\wedge b}$, $\gamma_{r|a\wedge\neg b}$, $\gamma_{r|\neg a\wedge b}$, or $\gamma_{r|\neg a\wedge\neg b}$, it must be the case that either γ_a and γ_b are extreme, or that $\gamma_{r|a\wedge b} = \gamma_{r|a\wedge\neg b} = \gamma_{r|\neg a\wedge b} = \gamma_{r|\neg a\wedge\neg b} = \mu$. Since we assume $\gamma_a, \gamma_b \in (0, 1)$, then it must be the case that $\gamma_{r|a\wedge b} = \gamma_{r|a\wedge\neg b} = \gamma_{r|\neg a\wedge b} = \gamma_{r|\neg a\wedge\neg b} = \mu$. Applying this condition to Eq. 3.16 yields

$$\mu\underbrace{(\gamma_a\gamma_b + \gamma_a\gamma_{\neg b} + \gamma_{\neg a}\gamma_b + \gamma_{\neg a}\gamma_{\neg b})}_{=1} = \mu$$

Thus proving statement (i) of Eq. 3.15. (Note the parenthesised sum is 1 because $\gamma_a + \gamma_{\neg a} = 1$ and $\gamma_b + \gamma_{\neg b} = 1$.)

The value of $\mathbb{P}(r, X = x')$ (Eq. 3.13) in the limit as $N \to \infty$ is

$$\lim_{N\to\infty} \mathbb{P}(r, X = x') = \begin{cases} \gamma_{r|a\wedge b}\gamma_b + \gamma_{r|a\wedge\neg b}\gamma_{\neg b} & \text{, if } a\,(x') = \mathsf{T} \\ \gamma_{r|\neg a\wedge b}\gamma_b + \gamma_{r|\neg a\wedge\neg b}\gamma_{\neg b} & \text{, if } a\,(x') = \mathsf{F} \end{cases} \tag{3.17}$$

To establish conditions when $\mathbb{P}(r, X = x')$ can model the correct probability in the limit, we solve $\lim_{N\to\infty} \mathbb{P}(r, X = x') = \mu$. Suppose $\mu = \gamma_{r|a\wedge b}$, then

$$\begin{aligned} \lim_{N\to\infty} \mathbb{P}(r, X = x') &= \gamma_{r|a\wedge b} \\ \gamma_{r|a\wedge b}\gamma_b + \gamma_{r|a\wedge\neg b}\gamma_{\neg b} &= \gamma_{r|a\wedge b} \\ (\gamma_b - 1)\gamma_{r|a\wedge b} &= (\gamma_b - 1)\gamma_{r|a\wedge\neg b} \\ \therefore \gamma_{r|a\wedge b} &= \gamma_{r|a\wedge\neg b} \end{aligned}$$

This shows that since γ_b is a non-extreme probability, it follows that $\lim_{N\to\infty} \mathbb{P}(r, X = x') = \mu$ if and only if $\gamma_{r|a\wedge b} = \gamma_{r|a\wedge\neg b}$. Similarly, for the case when $\mu = \gamma_{r|a\wedge\neg b}$, solve

$$\begin{aligned} \lim_{N\to\infty} \mathbb{P}(r, X = x') &= \gamma_{r|a\wedge\neg b} \\ \gamma_{r|\neg a\wedge b}\gamma_b + \gamma_{r|\neg a\wedge\neg b}\gamma_{\neg b} &= \gamma_{r|\neg a\wedge b} \\ (\gamma_b - 1)\gamma_{r|\neg a\wedge b} &= (\gamma_b - 1)\gamma_{r|\neg a\wedge\neg b} \\ \gamma_{r|\neg a\wedge b} &= \gamma_{r|\neg a\wedge\neg b} \end{aligned}$$

which implies that $\lim_{N\to\infty} \mathbb{P}(r, X = x') = \mu$ if and only if $\gamma_{r|\neg a\wedge b} = \gamma_{r|\neg a\wedge\neg b}$. Since it is not observed whether $\mu = \gamma_{r|a\wedge b}$ or $\mu = \gamma_{r|a\wedge\neg b}$, $\lim_{N\to\infty} \mathbb{P}(r, X = x') = \mu$ if and only if $\gamma_{r|a\wedge b} = \gamma_{r|a\wedge\neg b} \wedge \gamma_{r|\neg a\wedge b} = \gamma_{r|\neg a\wedge\neg b}$, thus proving condition *(ii)* in Eq. 3.15.

For $\mathbb{P}(r, Y = y')$, the limiting value is

$$\lim_{N\to\infty} \mathbb{P}(r, Y = y') = \begin{cases} \gamma_{r|a\wedge b}\gamma_a + \gamma_{r|\neg a\wedge b}\gamma_{\neg a} & \text{, if } b\,(y') = \mathsf{T} \\ \gamma_{r|a\wedge\neg b}\gamma_a + \gamma_{r|\neg a\wedge\neg b}\gamma_{\neg a} & \text{, if } b\,(y') = \mathsf{F} \end{cases} \tag{3.18}$$

and since $\mathbb{P}(r, Y = y')$ is a symmetric case of $\mathbb{P}(r, X = x')$, the same arguments used to derive condition *(ii)* of Eq. 3.15 to show that $\lim_{N\to\infty} \mathbb{P}(r, Y = y') = \mu$ if and only if $\gamma_{r|a\wedge b} = \gamma_{r|\neg a\wedge b} \wedge \gamma_{r|a\wedge\neg b} = \gamma_{r|\neg a\wedge\neg b}$, and thus proving condition *(iii)*

of Eq. 3.15.

□

Theorem 1 shows that assuming $\mathcal{G}$ is the true process that underlying our data, and under the stated conditions about $\mathcal{G}$, reference class statistics $\mathbb{P}(r, \mathsf{T})$, $\mathbb{P}(r, X = x')$ and $\mathbb{P}(r, Y = y')$ represent the true probability of $r(x', y') = \mathsf{T}$ in the limit. Figure 3.4 provides an instructive view of Thm. 1, showing several configurations of parameters of $\mathcal{G}$ ($\gamma_a = \gamma_b = 0.5$ is assumed for illustration purposes).

Firstly, Fig. 3.4(c) satisfies the condition for case (ii) in Thm. 1, in which case the true probability (those inside the grid) matches marginal probabilities (shown outside the grid), and that reference class statistic $\mathbb{P}(r, X = x')$ corresponds to these marginals in the limit. Thus, $\mathbb{P}(r, X = x')$ will yield the correct probability for $r(x', y')$. Figure 3.4(c) is also illustrative of (iii) in Thm. 1, due to the symmetry between (ii) and (iii).

Figures 3.4(a), 3.4(b), and 3.4(d) correspond to the case where the stated conditions in Thm. 1 do not hold. Hence, the marginal probabilities do correspond to the true probability of $r(x', y')$, thus $\mathbb{P}(r, \mathsf{T})$, $\mathbb{P}(r, X = x')$ and $\mathbb{P}(r, Y = y')$ cannot represent the true probability of $r(x', y')$.

It can be argued that $\mathcal{G}$ is an over-simplification of generative processes that actually underlie relational domains; that realistic generative processes are much more complex. However, our analysis shows that even in the case of such a simple process, reference classes can fail to entail the true probability in inference. The analysis is sufficient, in that the existence of a scenario where reference classes are suboptimal. In addition, the design of $\mathcal{G}$ is based on common intuitions about how data is generated in widely-studied relations domains, and not contrived to directly expose reference classes.

3.3.4 Inference with Latent Relational Models

Section 3.2 introduced relational models designed for clustering individuals, e.g. (Airoldi et al., 2008; Hofmann and Puzicha, 1999; Kemp et al., 2006; Xu et al., 2006), where clusters can be defined via latent properties. In this section we discuss inference for such models and the potential for obtaining the correct answer for probabilistic queries.

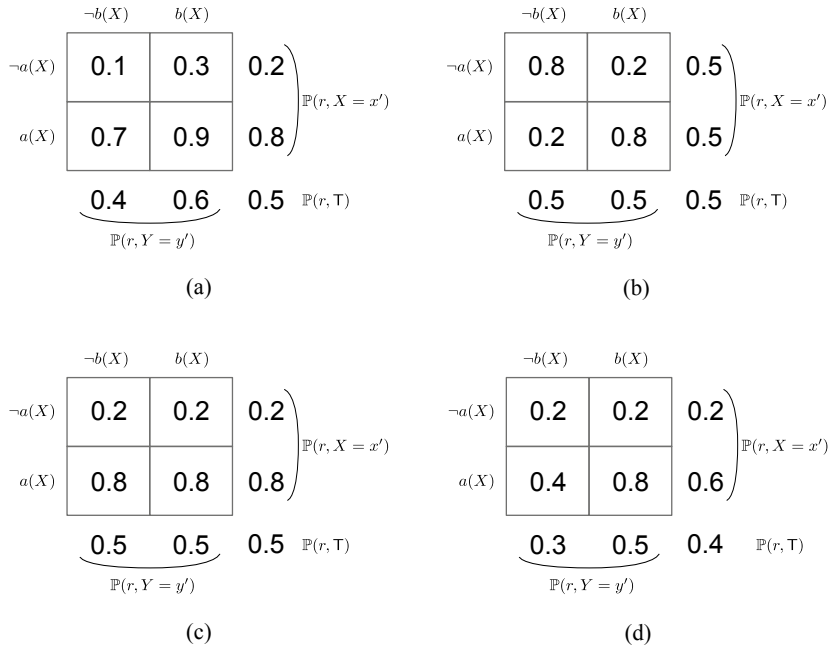


Figure 3.4: Four example configurations of θ_r in $\mathcal{G}$ (numbers inside the box). For each example, marginal probabilities are displayed outside of the box, computed assuming $\gamma_a = P_{\mathcal{G}}(a(X)) = 0.5$ and $\gamma_b = P_{\mathcal{G}}(b(Y)) = 0.5$. The corresponding reference classes are shown adjacently.

We use our language of LRMs to represent a clustering model. The LRM we consider for the problem at hand (see beginning of Sec. 3.3) is given by $\mathcal{L} = \langle \mathbf{A}, G, \Theta \rangle$ where

$$\mathbf{A} = \{r(X,Y), \alpha_1(X), \alpha_2(Y)\}$$
$$G = \{\alpha_1 \rightarrowtail r, \alpha_2 \rightarrowtail r\}$$
$$\Theta = \{\theta_r, \theta_{\alpha_1}, \theta_{\alpha_2}\}$$

Relation $r(X,Y)$ is observed, whilst $\alpha_1(X)$ and $\alpha_2(Y)$ are latent unary relations.

Whilst $\mathcal{L}$ is has the same dependency structure as $\mathcal{G}$, $\mathcal{L}$ may not produce the correct probability to answer queries. To see this we inspect the prediction of $\mathcal{L}$

given data $\mathbb{D}_r$, i.e.

$$
\begin{aligned}
& P_{\mathcal{L}}\left(r(x', y') = \mathsf{T} \mid \mathbb{D}_r\right) \\
& = \sum_{u \in V_{\alpha_1}} \sum_{v \in V_{\alpha_2}} P\left(r(x', y') \mid \alpha_1(x') = u, \alpha_2(y') = v, \mathbb{D}_r\right) \\
& \qquad\qquad P\left(\alpha_1(x') = u \mid \mathbb{D}_r\right) P\left(\alpha_2(y') = v \mid \mathbb{D}_r\right)
\end{aligned} \tag{3.19}
$$

Assume for now that the latent properties of $\mathcal{L}$ have the same range of values as those in $\mathcal{G}$, i.e. $V_{\alpha_1} = V_a$ and $V_{\alpha_2} = V_b$. The key is that if $\mathcal{L}$ represents $P(r(x', y') \mid \alpha_1(x') = u, \alpha_2(y') = v, \mathbb{D}_r)$, $P(\alpha_1(x') = u \mid \mathbb{D}_r)$, $P(\alpha_2(y') = v \mid \mathbb{D}_r)$ that matches $P_{\mathcal{G}}(r(x', y') \mid a(x') = u, b(y') = v)$, $P_{\mathcal{G}}(a(x') = u)$ and $P_{\mathcal{G}}(b(y') = v)$ respectively, then $P_{\mathcal{L}}(r(x', y') = \mathsf{T} \mid \mathbb{D}_r) = P_{\mathcal{G}}(r(x', y'))$; that $\mathcal{L}$ represents the correct probability.

The ability of $\mathcal{L}$ to achieve the correct probability in inference pertains directly to the learning of $\mathcal{L}$ from data. The basic requirement is to correctly estimate the values of $\alpha_1(x)$ and $\alpha_2(y)$, for all x and y, and to partition $\mathbb{D}_r$ in the same way as $\mathcal{G}$. Given the correct partitioning and that $r(X, Y)$ is observed, counting proportions in each partition will yield $P(r(x', y') \mid \alpha_1(x') = u, \alpha_2(y') = v, \mathbb{D}_r) = P_{\mathcal{G}}(r(x', y') \mid a(x') = u, b(y') = v)$ (in the limit of infinite data).

Computing $P(\alpha_1(x') = u \mid \mathbb{D}_r)$ and $P(\alpha_2(y') = v \mid \mathbb{D}_r)$ exactly will render $P_{\mathcal{L}}(r(x', y') = \mathsf{T} \mid \mathbb{D}_r)$ a *Bayes optimal* predictor, and $P(\alpha_1(x') = u \mid \mathbb{D}_r)$ approaches $P_{\mathcal{G}}(a(x') = u)$ in the limit of infinite data (similarly for $P(\alpha_1(x') = u \mid \mathbb{D}_r)$). For all but the simplest domains, computing the posterior quantities $P(\alpha_1(x') = u \mid \mathbb{D}_r)$ and $P(\alpha_2(y') = v \mid \mathbb{D}_r)$ exactly is confronted by a computational bottleneck. For example, computing $P(\alpha_1(x') = u \mid \mathbb{D}_r)$ requires marginalising over all other latent random variables in the model, i.e. the set of $\alpha_1(x)$ where $x \neq x'$ and all $\alpha_2(y)$ (including $\alpha_2(y')$). The number of sum terms is exponential in the number of variables in the marginalisation. The marginalisation problem cannot be easily simplified due to the dense correlations that exist amongst latent variables of the model. One can alleviate the computational complexity by seeking an approximation of $P(\alpha_1(x') = u \mid \mathbb{D}_r)$ that is efficient to compute as done in many variables Bayes approaches, or by sampling via MCMC methods (see (Friedman and Koller, 2009) for a general coverage).

It can be concluded that with finite amount of data the best result available

using the LRM is a Bayes optimal predictor, where optimality relies on an ability to infer the exact posterior latent property values for each individual. Learning can yield LRMs that represent the correct probabilities with increasing amount of data, and thus one concludes that, in theory, the correct probability is represented by LRMs. In practise, however, the intractability of the inferring the exact latent property values necessitates approximate solutions. Predictions using $\mathcal{L}$ thus cannot claim Bayes optimality. Depending on the accuracy of the approximate inference algorithm for determining latent property values, various degrees of error may result in the limit.

Whether LRMs are better predictors than reference classes – i.e. that they produce better approximations of the correct probabilities – is an empirical question which we address in Sec. 3.4. The answer depends on the inference algorithm for latent properties as well as the nature of the underlying domain.

At this point we note since a and b are not observed, we do not know the size of V_a and V_b. Thus, our LRM contains latent properties α and β such that V_α and V_β do not have *a priori* defined size. The arguments presented in this section have assumed that V_α and V_β match V_a and V_b respectively.

3.4 Experiments

In our experiments we compare reference classes with two forms of latent-property models: the LRMs and the IRM (Kemp et al., 2006). We evaluate predictive accuracy using simple domains containing only one observed relation, and report *log-loss* (or negative log-likelihood) (see Ch. 2, Sec. 2.1.3). Lower values of loss indicates higher accuracy.

Our first experiment uses data simulated by sampling $\mathcal{G}$ (Def. 4). Although Sec. 3.3 shows that LRMs can achieve the correct probabilities with less restrictive conditions than reference classes, our experiments aim to quantify the advantage of LRMs over reference classes. We validate by generating many examples of $\mathcal{G}$ with randomly generated latent properties and parameters.

In the second experiment, we use real-world relational data from the WebKB project[1] which describes hyperlinks between web pages from five academic world

[1] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/index.html

wide web domains.

Thirdly, we use a movie-rating dataset from the EachMovie project[2], which provides ratings by a approximately 60,000 users for 1,600 movies.

3.4.1 Models

Reference Class Predictors

In the kind of single-relation domains considered here, the three reference class predictors $\mathbb{P}(r, \mathsf{T})$, $\mathbb{P}(r, X = x')$ and $\mathbb{P}(r, Y = y')$ are again used, where $r(X, Y)$ represents the sole relation in the domain of interest, e.g. $hyperlink$ in the WebKB domain. To answer the query $r(x', y')$ we construct two predictions using the given reference classes. The first of which, called REF, simply chooses the best prediction from the three reference classes. REF is therefore an inadmissible predictor as we use test data to determine the best predictive model. However, it represents a gold standard for single reference class predictions that we use to compare with other predictors; any method that can outperform REF can outperform all other reference class predictors in the set.

A second reference class predictor, called POOL, combines all three reference classes by linear interpolation (e.g. weighted combination with weights summing to 1). However, we again construct a gold standard for this type of reference class predictors by adopting the following scheme. Suppose for each test case $r(x, y)$ that the true probability $P(r(x, y)) = \eta$ is known (this is true for synthetic data). Let δ_{min} be the minimum of $\mathbb{P}(r, \mathsf{T})$, $\mathbb{P}(r, X = x')$ and $\mathbb{P}(r, Y = y')$, and δ_{max} the maximum. Then, if $\eta \in [\delta_{min}, \delta_{max}]$, we assume a perfect interpolator POOL outputs η. If $\eta \notin [\delta_{min}, \delta_{max}]$, then REF is used. In the case that the true probability is unknown, $P(r(x, y)) \in \{0, 1\}$ (e.g. in real-world experiments), POOL again defaults to REF.

Latent-property Models

We use two latent-property models: the IRM (Kemp et al., 2006) and a LRM.

The IRM models latent properties to explain relational data (see Ch. 2, Sec.

[2]http://www.grouplens.org/node/76

2.4.1), and is a *non-parametric* model that stochastically generates the number of values for latent properties, as well as the value of latent properties for each individual.

Given a query $r(x', y') = w$, prediction is based on the latent property values of x' and y' respectively. Suppose that $\alpha_1(x') = u$ and $\alpha_2(y') = v$ in the IRM (or, x' belongs to cluster u and y' belongs to cluster v in the IRM nomenclature), the probability ascribed to $r(x', y') = w$ is the empirical proportion

$$\frac{|\{r\,(x,y) = w : \alpha_1(x) = u, \alpha_2(y) = v, \mathbb{D} \models r(x,y)\}|}{\sum_{w'} |\{r\,(x,y) = w' : \alpha_1(x) = u, \alpha_2(y) = v, \mathbb{D} \models r(x,y)\}|}$$

where $\mathbb{D}$ is the database function mapping $\mathbb{D}_r$ to a value.

The LRM used here is described in Sec. 3.3.4, where the latent properties are assumed Boolean. Learning is done via an approximate EM algorithm described in Ch. 4.

3.4.2 Protocol

Synthetic Relations

We simulate 2000 datasets, each generated by sampling a generative model in the form of $\mathcal{G}$ (see Sec. 3.3). Parameters of the generative model are generated randomly. There are two types, with domains $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$, where both $|\mathcal{D}(\tau_1)|$ and $|\mathcal{D}(\tau_2)|$ are restricted to be between 50 to 150. The two latent properties $a(X)$ and $b(Y)$ can have between 2 and 10 values. Each generated dataset contain observed cases for the observed relation $r(X, Y)$ only. A supervised learning framework is assumed, where 90% of the observed cases are used for training, whilst 10% are reserved for testing.

Training of the IRM uses MCMC with default parameters specified in the accompanying software[3], whilst the LRM training is done over 30 restarts. Each restart is run for a maximum of 100 iterations. The LRM which attains the best training set likelihood over restarts is returned.

[3] http://www.psy.cmu.edu/ ckemp/code/irm.html

WebKB

The WebKB contains web-page hyperlinks in the domain of five universities. The data consists of instances of the relation *link*(URL_1, URL_2) as well as features based on the words appearing in each web page. In these experiments, word features are omitted and we use only the hyperlink data so as to focus on the value of using latent properties to explain the relation.

Models REF, IRM and LRM are used for this experiment, where POOL was omitted as it defaults to REF when the ground truth probabilities are unknown (as is the case for real-world data). The IRM and LRM were trained using the same settings as described in the previous experiment. 2000 randomly sampled groups of 200 web pages are generated from the original data, each forming a standalone dataset. In each sample, 90% of data are used for training, and 10% for prediction.

Movie Ratings

We repeat our previous WebKB experiment with the EachMovie dataset[4]. The rating values are from 1 to 5, but we threshold these labels to be Boolean-valued by the global mean of all ratings. We take 500 independent sub-samples of the rating data and run independent experiments on each subsample. 90% of ratings in each sample are used for training, whilst 10% are used for testing.

3.4.3 Results

Synthetic Relations

The results from our 2000 experiments are binned according to the percentage of observed tuples to the maximum number of possible tuples for $r(X, Y)$. That is, the results are binned according to the amount of data available for learning. Figure 3.5 shows a plot of log-losses of REF, POOL and IRM and LRM, where each point in the graph represents the average log-loss in one bin of experiments using one predictor. Standard error is also shown. Each bin contains 250 to 300 data sets.

The outcome indicate a significant advantage towards the LRM and IRM (i.e. lower log-loss). Desirably, the log-loss of both and the LRM and IRM improves

[4] http://www.grouplens.org/node/76

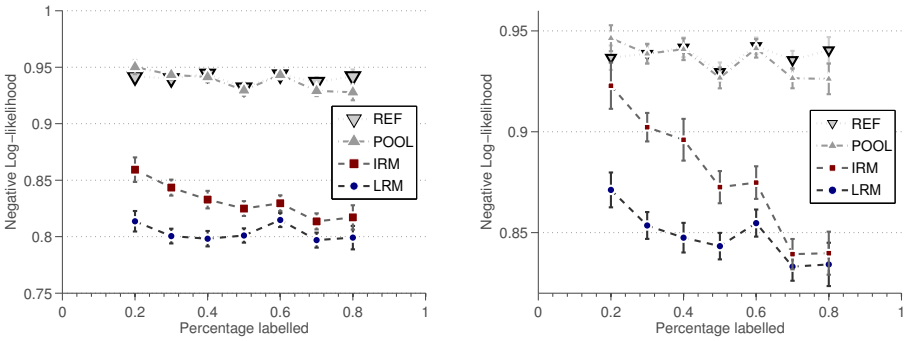


Figure 3.5: Log-loss for the IRM, LRM, and reference class predictions **REF** and **POOL**. Losses measure on training data (left) and test data (right) are shown for 2000 sets of simulated data. Each point in the figure corresponds to an average loss for bins of ≈ 200 datasets (with standard error shown). The bins are sorted in increasing order of percentage of observed data.

with the number of data points, indicating an ability to exploit information as they becomes available. The reference class approaches REF and POOL, on the other hand, appears to be insensitive to the amount of the observed data. The ability of LRMs to minimise loss when more information becomes available suggests that sufficient statistics encoded in the LRMs are better approximations of those in the underlying model than reference classes. The advantage of the LRM over the IRM is likely due to the soft-clustering nature of LRMs compared to the hard-clustering nature of IRM.

WebKB

We take the average log-loss over the 2000 experiments with REF, and IRM, shown in Table 3.1

The first observation is that each method performs well overall, scoring low log-losses. (Note random guessing of probabilities will yield log-loss of 1, using log of base 2, whilst the best possible score is 0). This indicates that the sampled datasets present easy prediction problems, due to the sparse linkage patterns in these sets. The fact that IRM and LRM again achieves significantly (in the statistical sense) better log-loss than REF emphasises the value of modelling latent

ALG.	$\mathbf{L}_{train}$	$\mathbf{L}_{test}$
REF	0.0216 ± 0.000402	0.0210 ± 0.000549
IRM	0.0179 ± 0.000268	0.0190 ± 0.000475
LRM	0.0181 ± 0.000260	0.0185 ± 0.000431

Table 3.1: Average log-loss over 2000 sampled datasets from the WebKB domain for REF, IRM and LRM on both the training and test sets.

properties, particular in these simple data samples where relative limited information is available. In turn, this suggests that the IRM and LRM are effective in exploiting information that is available. The separation between LRM and IRM are not significant in this case.

Movie Ratings

Tables 3.2 lists the training and test performance of each method measured in log-loss.

ALG.	$\mathbf{L}_{train}$	$\mathbf{L}_{test}$
REF	0.7373 ± 0.00844	0.7340 ± 0.00852
IRM	0.0173 ± 0.00819	0.4359 ± 0.01261
LRM	0.4728 ± 0.00818	0.5372 ± 0.00888

Table 3.2: Average log-loss over 500 sampled datasets from the EachMovie domain for REF, IRM and LRM on both the training and test sets.

The EachMovie dataset contains a more complex relational structure than the WebKB dataset, and the linkage density (e.g. ratings per user) is greater than that of WebKB (links per web page). As such, it represents a more difficult relational prediction problem, which is reflected in the overall increase of losses. The IRM's ability to exploit latent clusters of individuals likely contributes to its superior score, as on average it returned between 3 and 8 clusters of users (and movies), compared to all other methods tested. The LRM learned using LRM is restricted to modelling two clusters for users and movies and yields higher loss,

but still maintains its advantage relative to REF.

It is possible that the generative assumptions used in our analyses (which mirrors those behind LRMs) may hold in the world, thus contributing to LRMs' higher accuracy relative to reference classes. However, it is unlikely that the generative assumptions embodied in LRMs fully reflect the complexities of the true generative model in the world, thus highlighting the potential of modelling with latent properties.

3.5 Remarks

This chapter analysed two main classes of relational models in the context of prediction. We showed that relational models representing only observed relations can only entail the correct probability of queries (in the limit of infinite data) under rather restrictive conditions about the underlying generative process. On the other hand, relational models that also model latent properties of individuals, i.e. latent relational models (LRMs), can represent the correct probability in the limit. Whilst this is true in theory, obtaining such latent-property models requires exact inference for the latent properties, which is infeasible in general domains.

The question of whether LRMs yield better predictions in practice is tested empirically. In a synthetic domain where the generative process of data is that used in our theoretical analysis, we showed that LRMs dominate models with only observed relations. In real-world domains – a website domain and a movie-rating domain – we also observed that LRMs predict significantly better, supporting the notion that latent properties of individuals in fact play a meaningful role in the formation of relations. These observations are corroborated by using the IRM Kemp et al. (2006), which is a well-known latent-property model for relational data.

An additional result relates to the acquisition of reference classes. Whilst we have shown that relational models with only observed relations are implmenetations of reference classes, we also show that relational models with latent-properties of individuals are alternative representations of reference classes that correspond to using disjunctions of equalities in the representation. Further, demonstrating that latent-property models are significantly more accurate in prediction presents an attractive alternative to standard methods for obtaining reference classes.

Chapter 4

Learning Latent Relational Models

The goal of this chapter is to develop a method for learning latent relational models (LRMs). The goal of learning is to estimate parameters governing probabilistic dependencies in LRMs, and at the same time learn latent properties about each individual. With the ability to estimate LRMs that model dependencies over observed relations as well as latent properties, we can evaluate the question of whether latent properties of individuals are sufficient explanations for the observed relations – supported by Xu et al. (2006) – or that additionally modelling dependencies amongst the observed relations can yield a better model.

The general framework of EM (Dempster et al., 1977) can, in theory, address our learning goal. In practice, however, EM incurs prohibitive computational costs for all but the simplest relational domains, as LRMs for these domains represent large probabilistic models with many densely correlated latent random variables. We make a basic approximation that yields a computationally tractable approximate EM framework.

4.1 Estimating LRMs

Grounding a LRM with individuals of the domain yields a Bayesian network. Every ground atom of each latent property present in the LRM corresponds to a latent

random variable in the network. Learning parameters of a LRM that best fits the data requires estimating the marginal posterior distribution over all latent random variables. EM (Dempster et al., 1977) may be applied to this problem, and its limitations.

Algorithms in this chapter will be described at the ground level, i.e. Bayesian networks, where the set of observed random variables are denoted by $\mathbf{Y}$ corresponding observed values by $\mathbf{y}$. Latent random variables are denoted by $\mathbf{Z}$, with instantiations given by $\mathbf{z}$.

4.1.1 Expectation Maximisation

EM (Dempster et al., 1977) is a framework for learning maximum likelihood parameters of probabilistic models with latent random variables. Starting with some initial guess $\Theta^{(0)}$, EM iteratively constructs the sequence of parameters $\Theta^{(0)}, \Theta^{(1)}, \ldots$, until a local maximum of the *expected log-likelihood* is reached.

The expected log-likelihood is the expectation of the log-likelihood $\log P(\mathbf{y} \mid \Theta)$, with respect to a marginal function Q over latent variables. Namely,

$$\begin{aligned} L(\Theta; \mathbf{y}) &= \mathbb{E}_{\mathbf{z}} \left[\log P(\mathbf{y} \mid \Theta) \right] \\ &= \log \sum_{\mathbf{z}} P(\mathbf{y} \mid \mathbf{z}, \Theta) \, P(\mathbf{z} \mid \Theta) \end{aligned}$$

Maximising $L(\Theta; \mathbf{y})$ is an under-determined problem as both Θ and the latent marginal distribution $P(\mathbf{z} \mid \Theta)$ are unknown. (We denote $P(\mathbf{z} \mid \Theta)$ by $Q(\mathbf{z} \mid \Theta)$ in the rest of this chapter.) To find Θ and Q that jointly maximises $L(\Theta; \mathbf{y})$, EM begins with an initial guess $\Theta^{(0)}$, and for each iteration t, a new $Q^{(t)}(\mathbf{Z} \mid \Theta^{(t)})$ is computed (the E step), and using $Q^{(t)}$ a new $\Theta^{(t+1)}$ is computed (the M step) which maximises the log-likelihood $L(\Theta^{(t+1)}; \mathbf{y})$.

Neal and Hinton (1998) showed that $L(\Theta; \mathbf{y})$ has a lower-bound called the *variational free energy*, $\mathcal{F}(Q, \Theta)$, with which they explain the behaviour of EM.

This lower-bound can be derived using Jensen's inequality.

$$
\begin{aligned}
L(\Theta; \mathbf{y}) &= \log \sum_{\mathbf{z}} P(\mathbf{y}, \mathbf{z} \mid \Theta) \\
&= \log \sum_{\mathbf{z}} Q(\mathbf{z} \mid \Theta) \frac{P(\mathbf{y}, \mathbf{z} \mid \Theta)}{Q(\mathbf{z} \mid \Theta)} \\
&\geq \sum_{\mathbf{z}} Q(\mathbf{z} \mid \Theta) \log P(\mathbf{y}, \mathbf{z} \mid \Theta) - Q(\mathbf{z} \mid \Theta) \log Q(\mathbf{z} \mid \Theta) \\
&= \mathcal{F}(Q, \Theta)
\end{aligned}
\tag{4.1}
$$

Under the variational free energy interpretation, at each iteration t, the E step chooses a Q that makes the $\mathcal{F}(Q, \Theta)$ a tight likelihood lower-bound. In fact, it is shown that setting $Q^{(t)}$ to the marginal posterior distribution of $\mathbf{Z}$, i.e. $Q^{(t)}(\mathbf{Z} \mid \Theta^{(t)}) = p(\mathbf{Z} \mid \mathbf{y}, \Theta^{(t)})$, leads to the equality $L(\Theta^{(t)}; \mathbf{y}) \equiv \mathcal{F}(Q, \Theta^{(t)})$ (Neal and Hinton, 1998).

Next, the M step computes a new $\Theta^{(t+1)}$ given $Q = Q^{(t)}$. $\mathcal{F}(Q, \Theta)$ is convex in Θ given a fixed Q, thus there is a unique maximiser $\Theta^{(t+1)}$ for any fixed Q. Repeating the E and M steps is guaranteed to successively increase the expected log-likelihood until a local maximum.

In the context of learning LRMs, EM can be applied such that maximum likelihood parameters for dependencies for the LRM are computed in the M step, whilst posterior estimates of latent random variables for latent properties of individuals are computed in the E step. We discuss the suitability of EM for learning LRMs next.

4.1.2 EM for LRMs

EM suffers from practical limitations when there are many latent variables in the model. Specifically, storing the posterior distribution $Q(\mathbf{Z} \mid \Theta)$ requires space that is exponential in $|\mathbf{Z}|$, and inferring Q in the E step therefore requires computing the same order of probabilities. The same also applies to evaluating $L(\Theta^{(t)}; \mathbf{y})$.

One may exploit conditional independence in graphical models and represent Q as a product of less complex distributions. Such a representation can help reduce storage and computational requirements. However, when latent random variables are densely correlated, inferring their marginal distributions remain costly.

For relational domains, modelling latent properties of individuals can lead to many latent random variables in the ground network, and they are densely correlated. Consider a simple LRM shown in Fig. 4.1(a) (in plate notation), whose ground network has a bipartite structure (Fig. 4.1(b)). Each observed node in the ground network correlates its unobserved parents (i.e. the *explaining away* phenomenon in Bayesian networks). The fact that different observed nodes can have common (latent) parents induces a complex network of correlations across all latent parents in the network. For example, latent variables $\alpha_1(i_1)$, $\alpha_1(i_3)$, and $\alpha_2(j_3)$

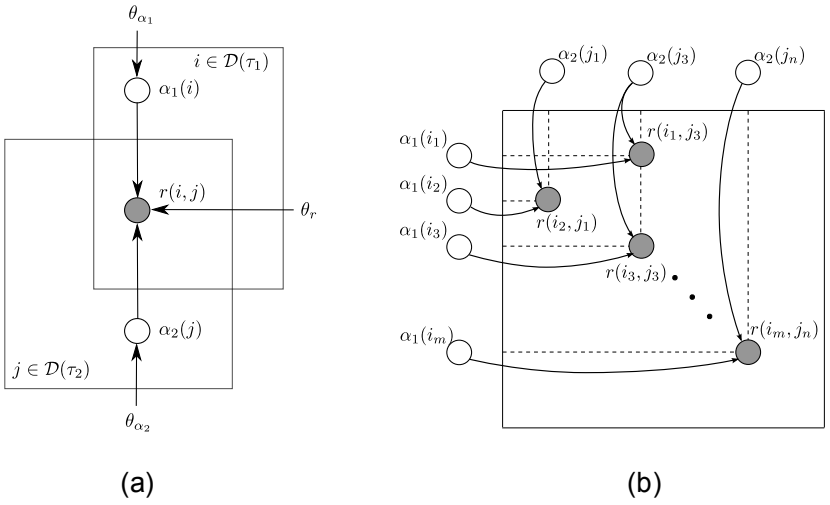


Figure 4.1: (a) a LRM in plate notation, and (b) an illustration of the corresponding ground LRM (Bayesian network) exhibiting a bipartite structure. Observed variables are shaded.

are intercorrelated by the observed nodes $r(i_1, j_3)$ and $r(i_2, j_3)$.

Assuming that all latent variables have k values, computing Q in the E step amounts to computing in the order of $\mathcal{O}(k^{|\mathbf{Z}|})$ probabilities; one per state of $\mathbf{Z}$. The exponential complexity in storage and computation means that EM algorithm quickly becomes infeasible, which is often the case in relational domains. An approximation is presented in the next section that alleviates this cost.

4.2 An Approximate EM Method for LRMs

Our first step to approximating the EM procedure is to represent the marginal distribution $Q(\mathbf{Z} \mid \Theta)$ in factor form $\prod_i Q(Z_i \mid \Theta)$. The number of marginal probabilities to compute is now in the order of $\mathcal{O}(k \cdot |\mathbf{Z}|)$, compared to $\mathcal{O}(k^{|\mathbf{Z}|})$ (where k is the number of values of each variable in $\mathbf{Z}$). Computing each $Q(Z_i \mid \Theta)$ in the E step still requires expensive marginalisation, however. The following describes approximations to obtain a tractable E step.

4.2.1 Expectation Step

We computing the distribution $\prod_i Q(Z_i \mid \Theta)$ in the E step by cycling through each $Z_i \in \mathbf{Z}$ and compute $Q(Z_i \mid \Theta)$ independently. Our main assumption for computing $Q(Z_i \mid \Theta)$ tractably is given in the following.

Localised Inference

Given parameters Θ and data $\mathbf{y}$, the problem of computing $Q(Z_i \mid \Theta)$, where $i \in \{1, \ldots, |\mathbf{Z}|\}$, is cast as computing the posterior distribution

$$\begin{aligned} P(z_i \mid \mathbf{y}, \Theta) &= \sum_{\mathbf{z}_{-i}} P(z_i, \mathbf{z}_{-i} \mid \mathbf{y}, \Theta) \\ &= \sum_{\mathbf{z}_{-i}} P(z_i \mid \mathbf{z}_{-i}, \mathbf{y}, \Theta)\, P(\mathbf{z}_{-i} \mid \mathbf{y}, \Theta) \end{aligned}$$

where $\mathbf{z}_{-i} = \mathbf{z} - z_i$. Since we assume a factorised representation, then

$$P(z_i \mid \mathbf{y}, \Theta) = \sum_{\mathbf{z}_{-i}} P(z_i \mid \mathbf{z}_{-i}, \mathbf{y}, \Theta) \prod_j P(z_j \mid \mathbf{y}, \Theta) \tag{4.2}$$

Note that $|\mathbf{z}_{-i}|$ can still be large, and Eq. 4.2 remains a large nested summation.

Now we apply our main approximation which restricts the number of nested summations. We assume that $P(z_i \mid \mathbf{y}, \Theta)$ is computed using only the portion of the ground network representing the *Markov blanket* of latent variable Z_i. The portion of the network outside of the Markov blanket are artificially pruned, reducing the number of latent variables in the marginalisation, and hence reduces the number of nested summations.

The Markov blanket of any variable X in a graphical model is defined as the set of neighbours of X. In Bayesian networks (directed graphical models), the Markov blanket of X is defined as the union of the graph parents of X, the graph children of X and the parents of each child of X (excluding X). A ground LRM is a Bayesian network, and for any node in the network its Markov blanket can be defined as follows:

Definition 5. *Let ground atom a_r be some ground instance of relation r, $A = par(a_r)$ the parents of a_r, $S = ch(a_r)$ the children of a_r, and $C = \bigcup_{c \in S} par(c) \backslash a_r$ the parents of each child. The Markov blanket of a_r is then $\mathcal{M}(a_r) = A \cup S \cup C$. The $par(\cdot)$ relation is defined in Ch. 2, Def. 2, and $ch(x) = \{y : x \in par(y)\}$.*

Figure 4.2 illustrates the use of a Markov blanket for computing the marginal of $\alpha(cs101)$ in a simple student-course domain. The rest of the network is effective pruned (greyed out).

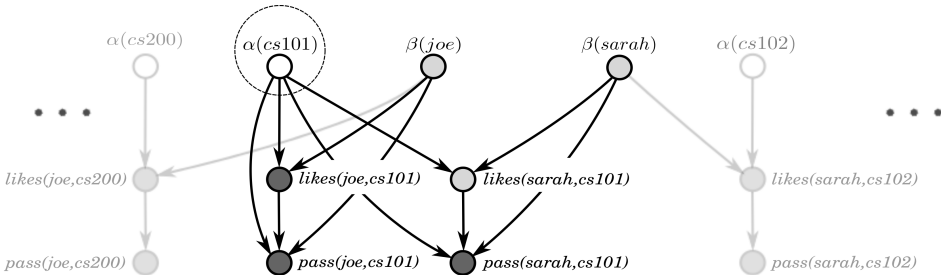


Figure 4.2: A Bayesian network localised to $\alpha(cs101)$, showing nodes $\alpha(cs101) \cup M(\alpha(cs101))$, where $\mathcal{M}(\alpha(cs101))$ is the Markov blanket of $\alpha(cs101)$. Dark-shaded nodes are observed, whilst light-shaded nodes are fixed to some marginal posterior distribution. The new marginal posterior estimate of $\alpha(cs101)$ is by probabilistic inference in this network.

Aside from computational tractability, the Markov blanket assumption has some intuitive properties for relational modelling. In Fig. 4.2, the Markov blanket $\alpha(cs101)$ contains ground atoms most relevant to the target individual $cs101$, i.e. grades involving $cs101$ as well as latent properties of students who took the $cs101$. Also, the Markov blanket directly encapsulates *dependencies* amongst relations, e.g. the directed arc between *likes* and *pass* in fig. 4.2. Thus, we can compute the

latent property of $cs101$ in a way that accounts for explicit dependencies amongst observed relations, whilst models proposed by (Kemp et al., 2006; Xu et al., 2006) do not allow for such dependencies.

By iteratively computing marginal posterior distributions for each latent variable, statistical correlations are propagated throughout the network. For example, in Fig. 4.2, latent properties of courses outside of the Markov blanket of $\alpha(cs101)$ indirectly correlate with the value of $\alpha(cs101)$. The correlation is made through other latent variables in the Markov blanket, such as those representing latent properties of students who took $cs101$.

E-step Update

To compute the marginal posterior value of some latent variable Z_i, let $\mathbf{Y}^{[i]}$ and $\mathbf{Z}^{[i]}$ denote the observed variables and latent variables that appear in the Markov blanket $\mathcal{M}(Z_i)$. For instance, for $Z_i = \alpha(cs101)$, we have from Fig. 4.2

$$\mathbf{Y}^{[i]} = (\textit{likes}(joe, cs101), \textit{pass}(joe, cs101), \textit{pass}(sarah, cs101))$$
$$\mathbf{Z}^{[i]} = (\beta(joe), \beta(sarah), \textit{likes}(sarah, cs101))$$

Given parameters $\Theta^{(t)}$, we update of marginal posterior for $Q(z_i \mid \Theta^{(t)})$ by incorporating the Markov blanket assumption into Eq. 4.2. The final update is

$$Q\left(z_i \mid \Theta^{(t)}\right) = \sum_{\mathbf{z}^{[i]}} P\left(z_i \mid \mathbf{y}^{[i]}, \mathbf{z}^{[i]}, \Theta^{(t)}\right) \prod_{z_j \in \mathbf{z}^{[i]}} Q\left(z_j \mid \Theta^{(t-1)}\right) \tag{4.3}$$

where t is an iteration number. Exact methods for probabilistic inference such asvariable elimination (Zhang and Poole, 1996) can be used to compute Eq. 4.3.

An interesting note about Eq. 4.3 is that it directly applies to ground instances of observed relations whose value is not observed. Thus, it conveniently provides a way to handle missing data (e.g. $\textit{likes}(sarah, cs101)$ in Fig. 4.2) on top of clustering when applied to latent properties.

4.2.2 Maximisation Step

In the M step, LRM parameters Θ are re-calculated using the marginal posterior distributions computed in the E step.

At iteration t, each $\theta_r \in \Theta$ is updated by with $\theta_r^{(t+1)}$ given by Eq. 4.4 as follows[1].

$$\theta_r^{(t+1)}[\mathbf{u}, v] = \frac{\tilde{\#}_{\mathbb{D}}\left(Par_G(r) = \mathbf{u} \wedge r = v, Q^{(t)}\right)}{\sum_{v'} \tilde{\#}_{\mathbb{D}}\left(Par_G(r) = \mathbf{u} \wedge r = v', Q^{(t)}\right)} \tag{4.4}$$

where $\tilde{\#}_{\mathbb{D}}(\cdot)$ is the *expected count* of the condition (a logical formula) in the parentheses, and is given by

$$\begin{aligned} &\tilde{\#}_{\mathbb{D}}\left(Par_G(r) = \mathbf{u} \wedge r = v, Q^{(t)}\right) = \\ &\sum_{\phi \in \Gamma_g} \hat{\mathbb{I}}\left(r\phi = v, Q^{(t)}\right) \hat{\mathbb{I}}\left(\pi_1\phi = u_1, Q^{(t)}\right) \ldots \hat{\mathbb{I}}\left(\pi_k\phi = u_k, Q^{(t)}\right) \end{aligned} \tag{4.5}$$

Here, $Par_G(r) = \{\pi_1, \ldots, \pi_k\}$ and $\mathbf{u} = (u_1, \ldots, u_k)$. Each π_i is a relation, and u_i is a value of the relation. Let g denote the formula $Par_G(r) = \mathbf{u} \wedge r = v$, Γ_g is the space of substitutions for the formula, and $\hat{\mathbb{I}}(\cdot)$ is the soft characteristic function (Eq. 2.3). (General definitions of $\tilde{\#}_{\mathbb{D}}(\cdot)$, Γ_g and $\hat{\mathbb{I}}$ can be found in Sec. 2.1.2 of Ch. 2.)

Using the example LRM relating to Fig. 4.2, we see that the parameter value $\theta_{pass}^{(t+1)}\,[(\mathsf{T},\mathsf{T},\mathsf{F}), \mathsf{T}]$ is given by

$$\theta_{pass}^{(t+1)}[(\mathsf{T},\mathsf{T},\mathsf{F}), \mathsf{T}] = \frac{\tilde{\#}_{\mathbb{D}}^{(t)}\left(\beta(S) = \mathsf{T} \wedge \alpha(C) = \mathsf{T} \wedge \mathit{likes}(S,C) = \mathsf{F} \wedge \mathit{pass}(S,C) = \mathsf{T}, Q^{(t)}\right)}{\sum_{v \in \{\mathsf{T},\mathsf{F}\}} \tilde{\#}_{\mathbb{D}}^{(t)}\left(\beta(S) = \mathsf{T} \wedge \alpha(C) = \mathsf{T} \wedge \mathit{likes}(S,C) = \mathsf{F} \wedge \mathit{pass}(S,C) = v, Q^{(t)}\right)}$$

[1] Equation 4.4 gives a maximum likelihood estimate, but maximum *a posteriori* is possible if pseudo-counts are available.

where T,F are Boolean values for true and false, and the expected counts are

$$\tilde{\#}_{\mathbb{D}}\left(\beta(S)=\mathsf{T}\wedge\alpha(C)=\mathsf{T}\wedge \textit{likes}(S,C)=\mathsf{F}\wedge \textit{pass}(S,C)=v, Q^{(t)}\right)$$
$$=\sum_{(s,c)}\hat{\mathbb{I}}\left(\textit{pass}(s,c)=v, Q^{(t)}\right)$$
$$\hat{\mathbb{I}}\left(\beta(s)=\mathsf{T}, Q^{(t)}\right)\hat{\mathbb{I}}\left(\alpha(c)=\mathsf{T}, Q^{(t)}\right)\hat{\mathbb{I}}\left(\textit{likes}(s,c)=\mathsf{F}, Q^{(t)}\right)$$

4.2.3 Likelihood

For the same reason that the E step of EM is computationally intractable, the exact $L(\Theta;\mathbf{y})$ is also intractable. We thus require a likelihood approximation.

A simple approximation is the *pseudo-likelihood* approximation (Besag, 1975). The basic idea underlying pseudo-likelihood is that the marginal likelihood for each observed node in a graph can be computed from its graph neighbours, and that the overall log-likelihood of data is a summation of individual marginal log-likelihoods.

We can easily construct a pseudo-likelihood by reusing the Markov blankets we have already defined in the E step of our approximate EM method. Namely, we compute the likelihood of *groups* of random variables, where each group corresponds to observed variables in each Markov blankets defined. In computing the marginal posterior distribution of a latent random variable using its Markov blanket, the likelihood of observed nodes in the blanket is also computed in the process. Our *pseudo-log-likelihood* is obtained by storing and summing log-likelihoods from all Markov blankets.

To illustrate, a marginal posterior probability for any $Z_i \in \mathbf{Z}$, given by Eq. 4.3, can also be written as

$$Q(z_i \mid \mathbf{y}, \Theta^{(t)}) = \frac{\sum_{\mathbf{z}^{[i]}} P(\mathbf{y}^{[i]} \mid \mathbf{z}^{[i]}, z_i, \Theta^{(t)}) Q(\mathbf{z}^{[i]} \mid \Theta^{(t)}) Q(z_i \mid \Theta^{(t)})}{\sum_{z_i'}\sum_{\mathbf{z}^{[i]}} P(\mathbf{y}^{[i]} \mid \mathbf{z}^{[i]}, z_i', \Theta^{(t)}) Q(\mathbf{z}^{[i]} \mid \Theta^{(t)}) Q(z_i' \mid \Theta^{(t)})}$$

where the denominator is expected likelihood of observed nodes in the Markov

blanket $\mathcal{M}(Z_i)$, which we denote as $\tilde{P}(\mathbf{y}^{[i]} \mid \Theta^{(t)})$. Finally, our pseudo-log-likelihood is simply the sum of log-likelihoods from all Markov blankets:

$$\tilde{L}(\Theta^{(t)} \mid \mathbf{y}) = \sum_{i=1}^{|\mathbf{Z}|} \log \tilde{P}\left(\mathbf{y}^{[i]} \mid \Theta^{(t)}\right) \tag{4.6}$$

An immediate problem with Eq. 4.6, however, is that it may be a poor estimate of the true log-likelihood. A key reason for this is that random variables appearing in the intersection of overlapping Markov blankets are *over-counted*. To alleviate this, one can adopt approximations that explicitly account for overlaps, e.g. the *region-based free energy approximations* of Yedidia et al. (2004) which generalises many well-known approximations such as the mean-field, Bethe and Kikuchi approximations. We empirically test the merits of using Eq. 4.6 in Sec. 4.4, and discuss alternative representations in Sec. 4.5.

Finally, having defined the main updates (Equations 4.3 and 4.4) of our approximation of EM method for learning LRMs, along with an expression for approximating the likelihood, our approximate EM algorithm is listed in Alg. 1 below.

4.3 Properties

4.3.1 Convergence

We can show that Alg. 1 produces parameters that monotonically improve the pseudo-log-likelihood. The analysis is similar to that for standard EM (see Sec. 4.1.1). The main step is to obtain the variational lower-bound for the pseudo-log-

Algorithm 1: An approximate EM algorithm for learning LRMs

Input: Database $\mathbb{D}$
Initialise:
  $\Theta^{(0)} \leftarrow \Theta_0$ and
  $\forall r \in \mathbf{A}, \forall a_r \in r; Q(a_r \mid \Theta^{(0)}) \leftarrow Q_0(a_r)$, where $\mathbb{D} \not\models a_r$
for $t = 1, 2, \ldots$ **do**
  ($\mathbf{E}$ step)
  for all $r \in \mathbf{A}$ **do**
    for all $a_r \in r$ where $\mathbb{D} \not\models a_r$ **do**
      Compute marginal posterior distribution $Q^{(t)}(a_r \mid \Theta^{(t)})$
        ... (Eq. 4.3)
    end for
  end for

  ($\mathbf{M}$ step)
  Update $\Theta^{(t+1)}$ given $Q^{(t)}$ ... (Eq. 4.4)

  if $\mid \tilde{L}(\Theta^{(t+1)}; \mathbf{y}) - \tilde{L}(\Theta^{(t)}; \mathbf{y}) \mid = 0$ **then**
    return $\Theta^{(t+1)}$ and $Q^{(t)}$
  end if
end for

likelihood (Eq. 4.6), shown as follows via Jensen's inequality.

$$
\begin{aligned}
\tilde{L}(\Theta; \mathbf{y}) &= \sum_{i=1}^{|\mathbf{Z}|} \log \sum_{\mathbf{z}^{[i]}, z_i} P\left(\mathbf{y}^{[i]}, \mathbf{z}^{[i]}, z_i \mid \Theta\right) \\
&= \sum_{i=1}^{|\mathbf{Z}|} \log \sum_{\mathbf{z}^{[i]}, z_i} Q\left(\mathbf{z}^{[i]}, z_i \mid \Theta\right) \frac{P\left(\mathbf{y}^{[i]}, \mathbf{z}^{[i]}, z_i \mid \Theta\right)}{Q\left(\mathbf{z}^{[i]}, z_i \mid \Theta\right)} \\
&\geq \sum_{i=1}^{|\mathbf{Z}|} \sum_{\mathbf{z}^{[i]}, z_i} Q\left(\mathbf{z}^{[i]}, z_i \mid \Theta\right) \log P\left(\mathbf{y}^{[i]}, \mathbf{z}^{[i]}, z_i \mid \Theta\right) \\
&\qquad\qquad + Q\left(\mathbf{z}^{[i]}, z_i \mid \Theta\right) \log Q\left(\mathbf{z}^{[i]}, z_i \mid \Theta\right) \\
&= \sum_{i=1}^{|\mathbf{Z}|} \mathcal{F}_i\left(Q, \Theta\right)
\end{aligned}
\tag{4.7}
$$

The above expression shows that there is a separate variational free energy for each Markov blanket. That is, the marginal likelihood from each Markov blanket

has its own variational lower-bound. Updating each marginal posterior estimate $Q(Z_i \mid \Theta)$ in the E step then tightens each bound $\mathcal{F}_i(Q, \Theta)$ individually, and the bound corresponding has its own maximiser. Maximising $\tilde{L}(\Theta; \mathbf{y})$ in the M step requires computing maximisers Θ_i for each $\mathcal{F}_i(Q, \Theta)$. However, since Θ defines shared LRM parameters, maximisation via Eq. 4.4 in fact chooses new parameters that represents the *average* of the individual maximising parameters. The averaged parameter choice will still improve each $\mathcal{F}_i$ and leads to monotonic improvement of $\tilde{L}(\Theta; \mathbf{y})$. Alg. 1 will thus monotonically improve $\tilde{L}(\Theta; \mathbf{y})$ until a local maximum.

4.3.2 Complexity

Suppose there are n latent variables in the model, and all are k-valued. The E step of Alg. 1 then computes n k-ary marginal posterior distributions. The E step complexity is thus $\mathcal{O}(n)$, compared to $\mathcal{O}(k^n)$ in standard EM, in terms of the number of probabilities computed. Note, however, each univariate marginal posterior computed in our approximate E step may have exponential complexity in marginalisation, depending on the size of Markov blankets and the inference algorithm used. Assuming exact inference algorithm such as variable elimination (Zhang and Poole, 1996), let m be the maximum number of latent variables in any Markov blanket used, and d the maximum dimensionality of conditional probability tables in the LRM. Each marginal posterior inference then has a worst-case complexity of $\mathcal{O}(m \cdot k^d)$. The E step therefore has a worst-case complexity of $\mathcal{O}(n \cdot m \cdot k^d)$ in terms of summation operations.

4.4 Experiments

4.4.1 Likelihood Optimisation

The goal of this experiment is to assess the effectiveness of Alg. 1 to identify LRM parameters that yields good likelihood scores. We synthesise small LRMs for generating data, for two reasons: (i) so that learned parameters can be compared to the (known) true generating parameters, and (ii) the pseudo-likelihood score (Eq. 4.6) can be compared to the exact expected log-likelihood (computable due to small model size).

Data

We create 1000 small LRMs, where each LRM $\mathcal{L} = \langle \mathbf{A}, G, \Theta \rangle$ has relations $\mathbf{A} = \{r(X,Y), \alpha_1(X), \alpha_2(Y)\}$, where $r(X,Y)$ is a Boolean relation with domain $\Omega_r = \mathcal{D}(\tau_1) \times \mathcal{D}(\tau_2)$ whilst $\alpha_1(X)$ and $\alpha_2(Y)$ are unary relations with domains $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$ respectively. The graph structure over relations is given by $G = \{\alpha_1 \rightarrowtail r, \alpha_2 \rightarrowtail r\}$. For each LRM, the number of values of α_1 and α_2 (i.e. V_{α_1} and V_{α_2}) are randomly chosen between 2 and 10. Domain sizes $|\mathcal{D}(\tau_1)|$ and $|\mathcal{D}(\tau_1)|$ are randomly chosen between 5 and 20. Conditional probability parameters $\Theta = \{\theta_{\alpha_1}, \theta_{\alpha_2}, \theta_r\}$ are also generated randomly, and are the *ground truth* parameters underlying the data.

Data is sampled from each synthetic LRM as follows. We first sample ground atoms for the unary relations. For each $x \in \mathcal{D}(\tau_1)$, where the range of α_1 is $V_{\alpha_1} = \{a_1, \ldots, a_{N_1}\}$, we flip a N_1-sided coin with bias θ_{α_1}. Similar procedure is carried out for all $y \in \mathcal{D}(\tau_2)$ using α_2. Then, for every pair $(x,y) \in \mathcal{D}(\tau_1) \times \mathcal{D}(\tau_2)$, where $\alpha_1(x) = u$ and $\alpha_2(y) = v$ are previously sampled, we first flip a fair coin to determine whether $r(x,y)$ will be observed (i.e. we simulate missingness). If observed, then another coin flip generates the observed value for $r(x,y)$ according to the bias $\theta_r[u,v]$, where $\theta_r[u,v]$ is the conditional probability parameter over values of r given $\alpha_1 = u \wedge \alpha_2 = v$. Samples for $r(X,Y)$ represent the data used for learning, which we denote as the dataset $\mathbb{D}_r$. Samples for α_1 and α_2 are kept hidden. For each of the 1000 sets of data, Alg. 1 run over 100 restarts to estimate Θ for the respective LRM.

Results

We compare the exact and approximate (pseudo) log-likelihoods for both true and learned parameters. Exact log-likelihoods are denotes by $L(\cdot)$ whilst approximate log-likelihoods are denoted by $\tilde{L}(\cdot)$. True parameters for model i are denoted by Θ_i^* and learned parameters are $\hat{\Theta}_i$. Recorded log-likelihoods for each model are shown as a function of increasing value of $L(\mathbf{y}_i; \Theta^*)$, where $\mathbf{y}_i$ is the data generated from model i, for $i = 1, \ldots, 1000$.

The first observation from Fig. 4.3 is that the exact likelihood of learned parameters correspond well with that of the true parameters in spite of the fact that Alg.

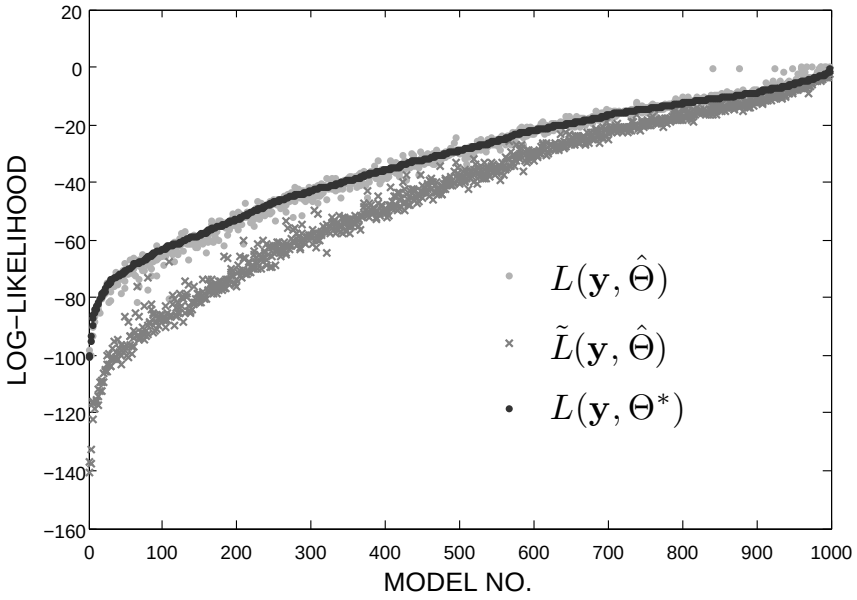


Figure 4.3: Results on simulated datasets from 1000 synthetic LRM. In all figures shown, $L(\cdot)$ denotes exact log-likelihood, and $\tilde{L}(\cdot)$ is the pseudo-log-likelihood (Eq. 4.6). Θ^* denote the true parameters used to generate the data, whilst Θ^* are those learned from the data. The exact likelihoods for the true and learned parameters for each of the 1000 models are plotted in ascending order of $L(\mathbf{y}_1; \Theta_1^*) \ldots L(\mathbf{y}_{1000}; \Theta_{1000}^*)$.

1 is formulated to optimise the pseudo-log-likelihood $\tilde{L}(\cdot)$. For the small domain considered here, this result can be expected as they represent simpler optimisation problems. In larger, more complex domains, a degradation in performance can be expected.

We can also see that the pseudo-likelihood $\tilde{L}(\mathbf{y}; \hat{\Theta})$ converges with the exact likelihood of the true parameters. One possible explanation is that some of our generative models have simpler structures than others. Simple patterns may induce lower over-counting errors in the pseudo-log-likelihood (Eq. 4.6). In such cases, pseudo-log-likelihoods can achieve good approximation of the exact log-likelihood. We attempt to quantify these effects by characterising the the complexity of relational patterns using two features: *link density* and the *number of*

overlaps.

Link density of a relation $r(X, Y)$ is the proportion of ground instances of $r(X, Y)$ that are observed, defined as $|\mathbb{D}_r|/|\Omega_r|$ where $\mathbb{D}_r$ is the dataset for $r(X, Y)$ and whilst Ω_r is the domain of $r(X, Y)$.

The number of overlaps measure, for some ground atom for a latent property, is defined as the number of Markov blankets in which the variable appears. For example, suppose $\alpha_1(x)$, $\alpha_1(x)$ may be correlated to $\alpha_2(y_1), \ldots, \alpha_2(y_n)$ in the ground network by observed nodes $r(x, y_1), \ldots, r(x, y_n)$. $\alpha_1(x)$. $\alpha_1(x)$ will therefore appear in the Markov blankets of $\alpha_2(y_1), \ldots, \alpha_2(y_n)$, resulting in an overlap count of n. The overlap count represents the number of times that $\alpha_1(x)$ will be over-counted. We define the *overlaps* measure as the average number of overlaps over all ground atoms for latent properties in the model.

In Fig. 4.4 we show that the overlap feature has a correlation coefficient of almost -1 with the exact log-likelihood; that models with lower overlap counts leads to higher log-likelihoods, and vice versa. Link density on the other hand appears a weaker indicator.

Given the strong correlation between the exact log-likelihood and overlap numbers, it is expected that over-counting errors in the pseudo-log-likelihood will be diminished against diminishing number of overlaps. The results of Fig. 4.3 are redrawn below as a function of *decreasing* values of link density and overlap numbers separately. These results show that the overlap count feature confirm and they confirm our expectations.

4.4.2 Systems of Relations

In this experiment we compare relational models that model (i) latent properties only, (ii) dependencies only, and (iii) both latent properties and dependencies. The main question is whether latent properties alone are sufficient to explain relational data (as argued for by (Xu et al., 2006)), or that the inclusion of dependencies amongst relations can further improve accuracy.

We use a simple experimental set-up consisting of two observed relations; the *target relation* and *co-relation*. We evaluate prediction on the target relation, whilst assessing the value conditioning on the co-relation in addition to latent properties.

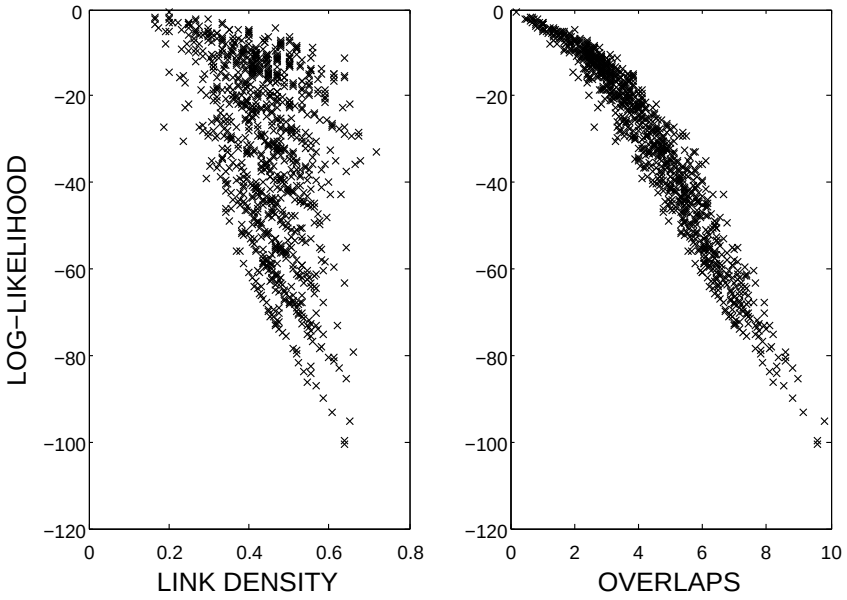


Figure 4.4: Exact log-likelihood of true generating parameters as a function of link density (left) and average overlap count (right).

We examine co-relation that are weakly, moderately, and strongly informative of the target relation (measured by *mutual information*). This will allow us to quantify the value of modelling dependencies in addition to latent properties. We describe our data, models and experimental protocol below.

Data

We use a reduced version of the *dimensionality of nations* dataset (Rummel, 1999) used by Kemp et al. (2006), where the domain consists of 14 countries countries, and 56 different Boolean two-placed relations are observed amongst them. Each relation has the form $r(X, Y)$, where logical variables X, Y denote nations.

We choose one target relation from the 56 available relations, and three co-relations that are weakly, moderately, and strongly informative of the target relation. The target relation is chosen for it highest entropy value[2]. In this case, the

[2]A maximum entropy value of 1 indicates an even distribution of positive labels (trues) and negative labels (falses). An entropy value of 0 indicates that all observed labels have the same value.

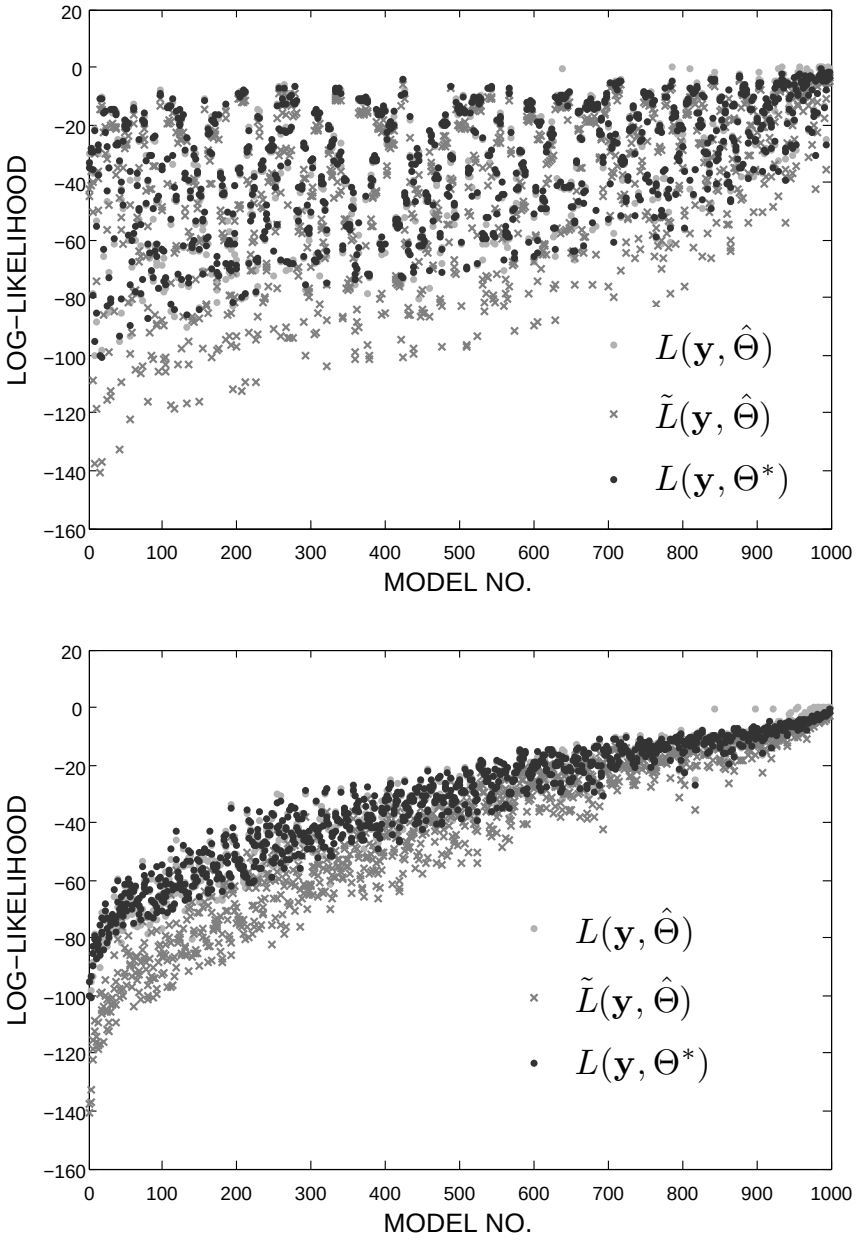


Figure 4.5: Results on 1000 simulated models, as a function of link density (top) and average overlap count (bottom).

best target relation is "intergovernmental organizations" (*intergov*), which represents the establishment of organizations between governments. Co-relations are chosen using the mutual information measure with the target relation

$$I(r_t; r_c) = \sum_{u,v \in \{\mathsf{F},\mathsf{T}\}} \hat{P}(r_t = u \wedge r_c = v) \log \frac{\hat{P}(r_t = u \wedge r_c = v)}{\hat{P}(r_t = u) P(r_c = v)}$$

where r_t and r_c denote the target relation and co-relation respectively, and

$$\hat{P}(r_t = u \wedge r_c = v) = \frac{\tilde{\#}_{\mathbb{D}}(r_t = u \wedge r_c = v)}{\sum_{u',v' \in \{\mathsf{F},\mathsf{T}\}} \tilde{\#}_{\mathbb{D}}(r_t = u' \wedge r_c = v')}$$

Three co-relations are chosen: "sever diplomatic relations" (*sever*), "non-government organisations" (*ngo1*), and a second non-government organisations relation (*ngo2*). Their mutual information measures with *intergov* are These values show that *sever*

$\mathbf{r}_1$	$\mathbf{r}_2$	$I(\mathbf{r}_1; \mathbf{r}_2)$
sever	*intergov*	5×10^{-5}
ngo1	*intergov*	0.63
ngo2	*intergov*	0.17

Table 4.1: Mutual information scores for each co-relation (*ngo1,ngo2,sever*) taken with the target relation *intergov*.

is the least informative of *intergov*, whilst *ngo1* is the most informative.

Models

We learn different LRMs using the given data, as depicted in Fig. 4.6, where *target*(X, Y) denotes the target relation, and *co-reln*(X, Y) denotes the co-relation.

The first LRM (Fig. 4.6 top-left) uses only the target relation and a latent property α. We denote this LRM by $\mathcal{L}_c$. The second LRM (Fig. 4.6 top-right) uses both the target and co-relation with α, and is denoted by $\mathcal{L}_{c+}$. Finally, a dependency between the target and the co-relation is modelled, which we call $\mathcal{L}_{cd}$ (Fig. 4.6 bottom). (Here, subscript c indicates the presence of a latent property

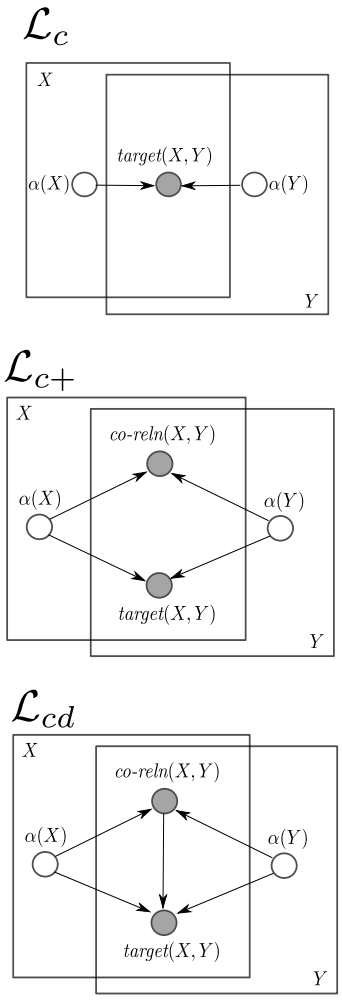


Figure 4.6: Three LRMs in plates notations: a single-relation latent-property model (top-left), a multiple-relation latent-property model (top-right), and multiple-relation latent-property model with a dependency link between the co-relation and the target relation (bottom).

for *clustering*, $'+'$ indicate more than one observed relation, and d indicates the dependency between the observed relations).

The latent property α is assumed to have a fixed number of values, i.e. $|V_\alpha|$ is fixed. A special case occurs when $|V_\alpha| = 1$, where α can effectively be removed

from the model, resulting in LRMs that contain only the two observed relations. In our experiments, we obtain LRMs for different number of values of $|V_\alpha|$.

We also use the infinite relational model (IRM)[3] due to Kemp et al. (2006), which is a nonparametric Bayesian hierarchical model for automatically generating an *a priori* unknown number of object clusters from multiple observed relations. It is one of the few existing proposals that allow clustering over multiple observed relations. Another well-known example is the IHRM (Xu et al., 2006), which defines a similar model to the IRM. We use the IRM as a representative model of this class. (Note that of the three LRMs described, $\mathcal{L}_c$ has the closest description the IRM.)

Protocol

Since we have three co-relations to consider, the experimental procedure described below applies for each co-relation.

Alg. 1 over 50 random restarts is used to train our LRMs, whilst 50 IRMs are trained using the MCMC with default parameters of software accompanying (Kemp et al., 2006). Five-fold cross-validation is run; where 20% observed cases for the target relation *intergov* are held out for prediction, whilst each model is trained on the remaining data. All data for the co-relation are used during training.

Whilst the IRM automatically determines the number of clusters, for LRMs we search over the number (values of $|V_\alpha|$) from 1 to 10. The cross-validation procedure described previously is repeated for each number.

Accuracy is measured in log-loss, and we record the average log-loss over folds of the cross-validation for both the training and test sets.

Results

The first experiment uses *sever* as the co-relation, and our experimental procedure described in the previous section produced the results shown in Fig. 4.7 for the LRMs tested.

The low mutual information value between *sever* and *intergov* (5×10^{-5}) means that conditioning on *sever* is unlikely to benefit predictions about *intergov*. This is

[3] Software from http://www.psy.cmu.edu/ ckemp/code/irm.html

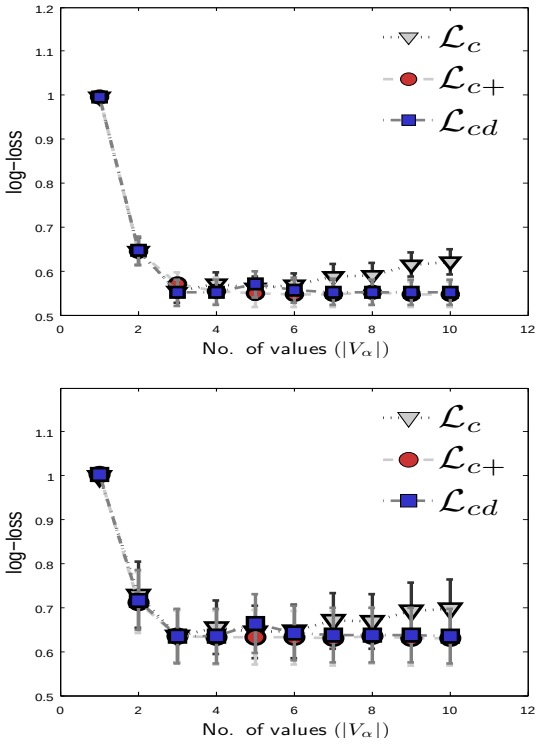


Figure 4.7: 5-fold cross-validated log-loss in predicting the *intergov* relation using LRMs shown in Fig. 4.6. Relation *sever* is used as the co-relation in this experiment. Results on training data (top) and test data (bottom) are shown. The horizontal axis correspond to different numbers of values of the latent relation α, and error bars indicate standard error. Lower log-loss mean greater accuracy.

seen at $|V_\alpha| = 1$, where latent property α are essentially absent, all models score poorly (log-loss ≈ 1)[4]. The results indicate that this is the case in both training and test data.

When latent property α is utilised (where $|V_\alpha| > 1$), significant reductions in log-loss is achieved which indicates that latent properties is providing benefits to explaining *intergov* data better than *sever*. The differences between $\mathcal{L}_c$, $\mathcal{L}_{c+}$, and $\mathcal{L}_{cd}$ for $|V_\alpha| > 1$ appear insignificant, however, suggesting that informative latent

[4]The log-loss value being close to 1 is due to the high self-entropy of the target relation *intergov*. A log-loss of 1 corresponds to random guessing.

properties were fitted in all three models. A noticeable trend is that $\mathcal{L}_c$ degrades as $|V_\alpha|$ increases, due likely to the corresponding increase in difficulty of optimising latent properties.

Now using *ngo2* as the co-relation, where the mutual information value between *ngo2* and *intergov* is higher at 0.17, we repeat the experiment and obtained results shown in Fig. 4.8, which shows similar trends to Fig. 4.7, except that *ngo2* directly yields improvements in accuracy, shown at $|V_\alpha| = 1$, where $\mathcal{L}_{cd}$ achieves a lower log-loss.

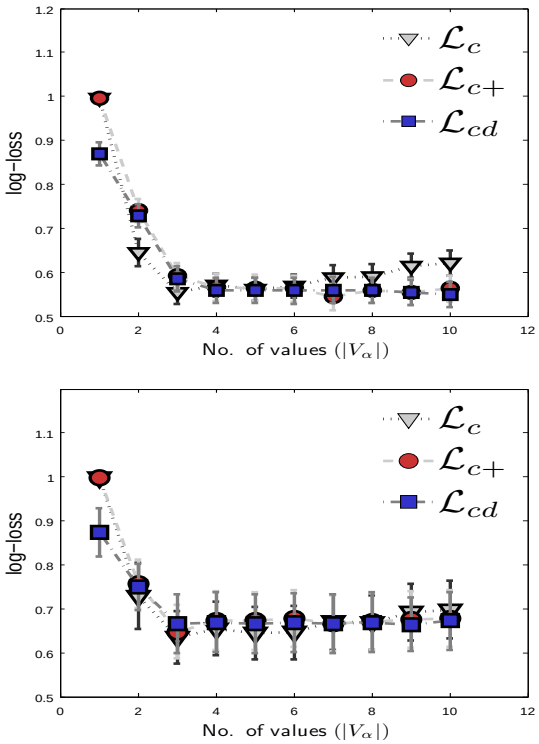


Figure 4.8: 5-fold cross-validated log-loss in predicting the *intergov* relation using LRMs shown in Fig. 4.6. Relation *ngo2* is used as the co-relation in this experiment. Results on training data (top) and test data (bottom) are shown. The horizontal axis correspond to different numbers of values of the latent relation α. Error bars indicate standard error.

Finally, using co-relation *ngo1* (with mutual information of 0.63), we observe

significant changes in behaviour. Consider the results shown in Fig. 4.9.

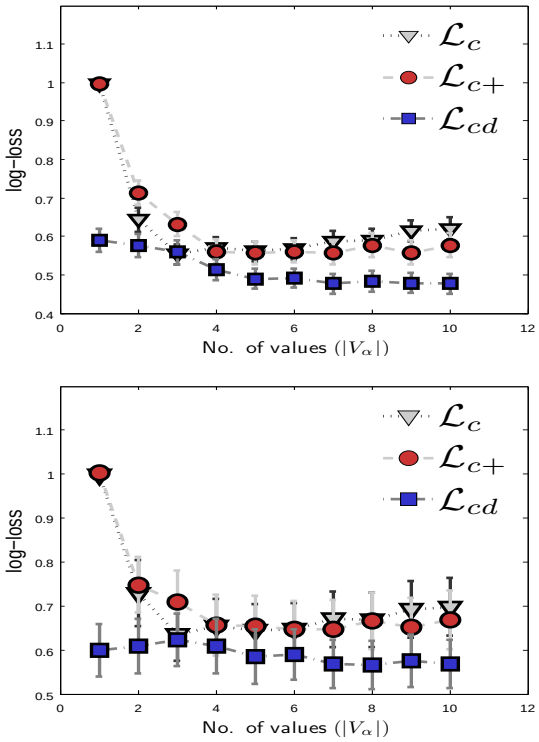


Figure 4.9: 5-fold cross-validated log-loss in predicting the *intergov* relation using LRMs shown in Fig. 4.6. Relation *ngo1* is used as the co-relation in this experiment. Results on training data (top) and test data (bottom) are shown. The horizontal axis correspond to different numbers of values of the latent relation α. Error bars indicate standard error.

Figure 4.9 shows that explicitly modelling the dependency between the target relation and the co-relation yields noticeable improvements, due to the informativeness of *ngo1* (seen at $|V_\alpha| = 1$). Furthermore, increasing $|V_\alpha|$ continues to improve log-loss on both training and test data, evidence of the increasing value of latent properties. Models with no explicit dependency between *intergov* and *ngo1* (i.e. $\mathcal{L}_c$ and $\mathcal{L}_{c+}$) achieves their best log-loss of ≈ 0.6 on training data, $\approx 0.65 \sim 0.7$ on test data (this is also true when using other co-relations). This suggest that latent properties alone may not be sufficient to model the information provided by

the co-relation.

Finally, we compare log-loss of the best IRM on datasets corresponding to different co-relations, with the log-loss of the best LRMs in these experiments. The results are shown in Table 4.2.

TRAIN

	co-relations			
	none	*sever*	*ngo2*	*ngo1*
IRM	$0.796 \pm 0.056(4)$	$0.819 \pm 0.010(3)$	$0.962 \pm 0.024(4)$	$0.836 \pm 0.016(5)$
$\mathcal{L}_c$	$0.553 \pm 0.027(3)$	–	–	–
$\mathcal{L}_{c+}$	–	$0.546 \pm 0.029(10)$	$0.544 \pm 0.029(7)$	$0.556 \pm 0.030(5)$
$\mathcal{L}_{cd}$	–	$0.550 \pm 0.029(3)$	$0.549 \pm 0.029(10)$	$\mathbf{0.477 \pm 0.025}(10)$

TEST

	co-relations			
	none	*sever*	*ngo2*	*ngo1*
IRM	$0.858 \pm 0.116(4)$	$0.854 \pm 0.017(3)$	$1.001 \pm 0.041(3)$	$0.834 \pm 0.031(5)$
$\mathcal{L}_c$	$0.636 \pm 0.059(3)$	–	–	–
$\mathcal{L}_{c+}$	–	$0.630 \pm 0.062(10)$	$0.648 \pm 0.0615(3)$	$0.646 \pm 0.065(6)$
$\mathcal{L}_{cd}$	–	$0.635 \pm 0.062(3)$	$0.664 \pm 0.061(10)$	$\mathbf{0.568 \pm 0.055}(7)$

Table 4.2: Five-fold cross-validated log-loss (with standard errors) from prediction on training and test of the target relation *intergov*. Models tested are: the IRM, and LRMs $\mathcal{L}_c$, $\mathcal{L}_{c+}$, and $\mathcal{L}_{cd}$. For each loss reported, the corresponding value for $|V_\alpha|$ (e.g. the number of clusters) is also shown. All losses for the LRMs are the best results achieved over different number of clusters, obtained from data underlying Figs. 4.7, 4.8 and 4.9. Lower values indicate better performance.

4.5 Remarks

In this chapter we have presented a learning method for learning relational models that represent dependencies amongst observed and latent properties. The algorithm is an approximation of EM, and avoids the main computational issues that EM incurs. Convergence and complexity of the proposed algorithm is also discussed. Simulations show that the algorithm is effective in learning models that can achieve good likelihood scores.

The algorithm was then used to answer one of the main questions of this thesis, regarding whether modelling dependencies as well as latent properties can yield

more accurate models than modelling either alone.

From experiments on real-world data, we showed that (i) latent properties can help improve model accuracy, particularly when observed relations are not mutually informative; (ii) when there exist relations that are mutually informative, it is advantageous to explicitly model dependencies amongst them; (iii) the added presence of latent properties can help further improve accuracy, and does not lead to over-fitting.

We have confirmed that modelling with latent properties yielded better predictive accuracy (on training as well as test data) than models based on observed properties alone. More importantly, we demonstrated the combination of latent properties and dependencies further improves model accuracy.

Chapter 5

Learning with Uncertainty about Size of Latent Properties

In Ch. 4 we assumed that latent properties have *a priori* known number of values. In this chapter, we relax this assumption and allow the number of values for latent properties to be *a priori* unknown and unbounded.

To express our uncertainty over the number of values (which we refer to as the *size*) of latent properties, we consider a distribution over the unbounded set of possible sizes. For instance, a latent property whose size may be any positive integer, we may use a Poisson distribution as the prior distribution over sizes.

There are proposals which address the problem of unbounded sizes of latent properties in relational models, exemplified by the IHRM (Xu et al., 2006) and IRM (Kemp et al., 2006). These models draw from results in *Bayesian nonparametrics* (Ghosh and Ramamoorthi, 2003) and use stochastic processes as prior distributions over sizes. Stochastic processes such as the Dirichlet process (DP) (Ferguson, 1973) or CRP (Aldous, 1983) stochastically generate Dirichlet distributions whose number of classes are *a priori* undetermined. Running the process infinitely many times produces all Dirichlet distributions. The bias of DPs and CRPs is towards smaller number of classes. Joint probability models involving stochastic process priors typically proceeds by Monte Carlo sampling.

In this chapter we propose an extension to LRMs, called infinite-size latent relational model (ILRM), which explicitly represents the distribution over size of

latent properties. We develop a learning procedure for ILRMs that searches search over the sizes of latent properties, and at each search step learn parameters for a standard LRM. A main feature of the search approach is that error bounds on the likelihood of data for each step can be calculated (derived in Sec. 5.3.1). In other words, where the exact likelihood of data is obtained by averaging over the (infinite) space of models, our search-based approach computes error bounds on the likelihood associated with having only searched a finite number of models. The error diminishes as we evaluate more of the model space. The proposed procedure is an alternative to stochastic-process based representations estimated via Monte Carlo sampling.

The likelihood bounds also directly leads to bounds on the posterior of latent property sizes, and in turn yields an approximate Bayesian averaging scheme for prediction. Bayesian averaging involves averaging over posterior-weighted predictions of all models, leading to robust predictions that are sometimes more accurate than single-model predictors (Hoeting et al., 1999).

The ability to perform averaging and have bounds indicating the quality of the averaged prediction extends existing work. In addition, our formulation permits a flexible choice of prior distributions over latent property sizes, where previous proposals are fixed to the CRP (Kemp et al., 2006; Xu et al., 2006).

Experiments in Sec. 5.5 demonstrates the error bounds obtained, and applies Bayesian averaging for prediction. On a real-world dataset used in Ch. 4, we show that our approximate Bayesian predictions of ILRMs perform competitively with the best LRMs found in Ch. 4. A key observation is that a prior distribution not restricted to favouring smaller sized latent properties led to the best averaged predictions.

5.1 Nonparametric Relational Models

Existing relational models that represent latent properties and uncertainty about the size of latent properties generally rely on stochastic processes such as the CRP. Well-known examples include the infinite relational model (IRM) (Kemp et al., 2006) and infinite hidden relational model (IHRM) (Xu et al., 2006). The underlying semantics of these models is that latent properties of individuals generate rela-

tions amongst individuals. Latent properties often mean clusters, and a stochastic process is used to stochastically generate clusters of domain individuals and prior probabilities of the clusters. Since relations are not use to generate other relations in the IRM/IHRM, these models are hierarchical Bayesian models.

For a simple domain consisting of only one relation $r(A, B)$ where A and B are of the same type, an IRM corresponds to the probabilistic model.

$$\begin{aligned} \forall X, \quad & \alpha(X), S \mid \gamma && \sim \mathsf{CRP}\left(\gamma\right) \\ \forall A \forall B, \quad & r\left(A, B\right) \mid \alpha(A), \alpha(B) && \sim p\left(r(A, B) \mid \alpha(A), \alpha(B)\right) \end{aligned} \tag{5.1}$$

where $\alpha(X)$ is a latent property, and S represents the size (number of values) of α, which can be unbounded. This means that for any pair of individuals (a, b), the value of $r(a, b)$ is dependent on the value of $\alpha(a)$ and $\alpha(b)$. The size and prior distribution (a Dirichlet distribution) over the values of α are drawn from the CRP[1] with *concentration* parameter γ. The CRP generates Dirichlet prior distributions biased towards smaller number of classes.

The alternative model we consider in this thesis is one that generates the value of S, followed by the value of α. Then, for all pairs (a, b) the value of relation $r(a, b)$ is generated from $\alpha(a)$ and $\alpha(b)$. This generative model is described as follows

$$\begin{aligned} & S \mid \gamma && \sim \phi(\gamma) \\ \forall X, \quad & \alpha(X) \mid S && \sim Q\left(\alpha(X) \mid S\right) \\ \forall A \forall B, \quad & r\left(A, B\right) \mid \alpha(A), \alpha(B) && \sim p\left(r(A, B) \mid \alpha(A), \alpha(B)\right) \end{aligned} \tag{5.2}$$

Here, ϕ represents our prior distribution over S, where prior distributions such as the geometric or Poisson distribution may now be considered instead of the CRP. For some $S = s$, α has s values, and distribution Q is defined over the values of α. Note that a CRP generates Dirichlet classes sequentially, and weigh earlier-generated classes more heavily, our approach now allows one to prefer each class equally, e.g. a symmetric Dirichlet distribution over the values of α can be used.

[1] In their original paper, Kemp et al. (2006) specify the exact form of the conditional distribution as well as a prior distribution for $r(X, Y)$. We describe it in general form and omit the prior an focus on the role of the CRP.

For example, suppose a Dirichlet distribution $p(\alpha)$ with classes $1 \ldots k$ is generated by a CRP, for any pair $i \leq j$ it follows that $P(\alpha) = i \geq P(\alpha) = j$.

When S is fixed, the model can be represented as a LRM where all relations have knwon sizes, and can be learned by our approximate EM algorithm presented in Ch. 4.

With respect to the generative model in Eq. 5.2, the IRM couples the process of generating the size of α with that of generating the distribution Q over values of α for each individual. By decoupling the two steps – generating S and generating α given S – Eq. 5.2 allows different choices for ϕ and Q. In the next section we define an extension of LRMs that represents uncertainty over configurations.

5.2 Infinite-size LRMs

The general learning approach presented in this chapter involves searching over the size of latent properties of a LRM, and for each step of the search learn the parameters of the corresponding fixed-size LRM. For a LRM whose latent properties have sizes represented in vector $\mathbf{C}$, an assignment of values to $\mathbf{C}$ is a *configuration*. Definitions relating to configurations are as follows.

Definition 6. ***Configuration ordering***
Given two configurations $\mathbf{a} = \{a_1, \ldots, a_m\}$ *and* $\mathbf{b} = \{b_1, \ldots, b_m\}$, $a \preceq b$ *if* $\forall i \in \{1 \ldots m\}, a_i \leq b_i$.

Definition 7. ***Configuration space***
Given a configuration $\mathbf{u}$*, a configuration space induced by* $\mathbf{u}$ *is* $S^{\mathbf{u}} = \{\mathbf{c} : \mathbf{c} \preceq \mathbf{u}\}$.

We augment the LRM with a random vector representing the size of latent properties, where instantiations of the vector yields configurations. The augmented representation is called an ILRM, defined as follows:

Definition 8. ***Infinite-size latent relational models***
An infinite-size latent relational model (ILRM) is a tuple $\langle \mathbf{V}, G, \Theta \rangle$*, where* $\mathbf{V} = (\mathbf{R}, \mathbf{R}_I, \mathbf{S})$. $\mathbf{R}$ *contains relations whose sizes are known a priori,* $\mathbf{R}_I$ *whose sizes are not known a priori, and random variables* $\mathbf{S}$ *represent sizes of relations in* $\mathbf{R}_I$*. G is a DAG over* $\mathbf{V}$*. The parents of any* $S \in \mathbf{S}$ *can only consist of other variables in*

$\mathbf{S}$. *The parents of any member in* $\mathbf{R}_I$ *must include exactly one size variable from* $\mathbf{S}$. *If* $R \in \mathbf{R}_I$ *is a parent of some relation* T*, then the size parent of* R *is also a parent of* T. Θ *is a set of conditional probability parameters*[2] *for each member of* $\mathbf{V}$. *We make* Θ *explicit by adding it to the right hand side of conditional probabilities.*

For each $V \in \mathbf{V}$ *there is a CPD,* $p(V \mid Par_G(V), \Theta)$*, of* V *given its parents* $Par_G(X)$*, parametrised in* Θ. *An ILRM represents the joint distribution over all groundings of* $\mathbf{V}$. *Let* $\mathbf{A} = \mathbf{R} \cup \mathbf{R}_I$*, and*

$$W_{\mathbf{A}} = \bigcup_{A \in \mathbf{A}} \bigcup_{\sigma \in \Gamma_A} \sigma A$$

be the set of all ground atoms of relations in $\mathbf{A}$*, where* Γ_A *is the substitution space of relation* A. *Then, let* $W_{\mathbf{V}} = W_{\mathbf{A}} \cup \mathbf{S}$ *represent all random variables in a ground ILRM, the joint distribution represented by the ground ILRM is*

$$p\left(W_{\mathbf{V}} \mid \Theta\right) = \prod_{S \in \mathbf{S}} p\left(S \mid Par_G(S), \Theta\right) \prod_{A \in W_{\mathbf{A}}} p\left(A \mid Par_G(A), \Theta\right) \tag{5.3}$$

5.3 Learning

We describe how search is done in the (unbounded) space of configurations for learning ILRMs, and how to derive bounds about the likelihood at each search step, discussed first as follows.

5.3.1 Likelihood Bounds

For purposes of exposition, we use the following notation for the rest of this chapter. Let $\mathbf{Z}$ be the set of all latent random variables in a ground LRM (i.e. ground atoms corresponding to latent properties and unobserved cases of observed relations), and $\mathbf{Y}$ the set of all observed random variables (observed ground instances of relations). Observations are given by $\mathbf{y}$.

The likelihood of data, $P(\mathbf{y} \mid \Theta)$ is obtained by marginalising over the size variables. The marginal is an infinite sum where each summand corresponds to

[2]Note that the size of Θ is *a priori* unbounded. Thus this forms a nonparametric model.

a size configuration. Our approach is to enumerate only a finite number of summands in the marginal, and quantify the approximation error with bounds on the likelihood.

Our first step towards a likelihood bound is to show that the likelihood given a Bayesian network with configuration $\mathbf{s}'$, written as $P(\mathbf{y} \mid \theta^*_{\mathbf{s}'})$, is lower-bounded by that of one with a lesser configuration $\mathbf{s} \preceq \mathbf{s}'$. We prove this lower-bound for the case that optimal parameters $\theta^*_{\mathbf{s}} \in \Theta$ are available given any configuration $\mathbf{c}$. $\theta^*_{\mathbf{c}}$ is given by

$$\theta^*_{\mathbf{c}} = \arg\max_{\theta \in \Theta} P(\mathbf{y} \mid \theta, \mathbf{c}) \tag{5.4}$$

Proposition 1. Given LRMs whose latent properties have sizes given by configurations $\mathbf{s}$ and $\mathbf{s}'$ respectively. If $\mathbf{s} \preceq \mathbf{s}'$, and $\mathbf{y}$ represent the observations, then

$$P(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}) \leq P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}'}, \mathbf{s}'\right) \leq 1 \tag{5.5}$$

where $\theta^*_{\mathbf{s}}$ and $\theta^*_{\mathbf{s}'}$ are given by Equation 5.4.

Proof. Firstly, for any configuration $\mathbf{s}$ and parameter $\theta_{\mathbf{s}}$,

$$P(\mathbf{y} \mid \theta_{\mathbf{s}}, \mathbf{s}) \leq P(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}) \tag{5.6}$$

where $\theta^*_{\mathbf{s}}$ is the likelihood-maximising parameter (Equation 5.4). For any two configurations $\mathbf{s}$ and $\mathbf{s}'$ such that $\mathbf{s}' \succeq \mathbf{s}$, the space of LRMs with configuration $\mathbf{s}$ is a subset of that with configuration $\mathbf{s}'$. Thus, it is always possible to choose parameters for LRM $\mathcal{L}'$ with configuration $\mathbf{s}'$ that yields a likelihood at least that of LRM $\mathcal{L}$ with configuration $\mathbf{s}$. Namely

$$P(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}) \leq P\left(\mathbf{y} \mid \theta_{\mathbf{s}'}, \mathbf{s}'\right)$$

Applying Equation 5.6 to $P(\mathbf{y} \mid \theta_{\mathbf{s}'}, \mathbf{s}')$ then yields the lower-bound of Equation 5.5

$$P(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}) \leq P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}'}, \mathbf{s}'\right)$$

The upper-bound of Equation 5.5 is true by definition of probability. □

A clustering analogy is illustrative here. Proposition 1 states that the best n-cluster model is always in the space of $(n+1)$-cluster models. Thus, the best $(n+1)$-cluster model must score at least as well as the best n-cluster model.

Using Proposition 1, we next derive upper and lower bounds for the likelihood $P(\mathbf{y} \mid \Theta)$, which is given exactly by the infinite sum

$$P\left(\mathbf{y} \mid \Theta^*\right) = \sum_{\mathbf{s} \in dom(\mathbf{S})} P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}\right) \tag{5.7}$$

with $\theta^*_{\mathbf{s}} \in \Theta^*$ given by Equation 5.4. The bounds of interest are given by Lemma 1 below.

Lemma 1. Given a fixed configuration $\bar{\mathbf{s}}$ where $\forall c \in \bar{\mathbf{s}}, c < \infty$. $P(\mathbf{y})$ satisfies

$$f + P\left(\mathbf{y} \mid \theta^*_{\bar{\mathbf{s}}}, \bar{\mathbf{s}}\right) g \leq P\left(\mathbf{y} \mid \Theta^*\right) \leq f + g \tag{5.8}$$

where

$$f = \sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}\right)$$

$$g = \sum_{\mathbf{s}' \in dom(\mathbf{S}) \backslash \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{s}' \mid \theta^*_{\mathbf{s}'}\right)$$

Proof. We first rewrite Equation 5.7 as

$$\begin{aligned} P\left(\mathbf{y} \mid \Theta^*\right) = & \sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}\right) \\ & + \underbrace{\sum_{\mathbf{s}' \in dom(\mathbf{S}) \backslash \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}'}, \mathbf{s}'\right) P\left(\mathbf{s}' \mid \theta^*_{\mathbf{s}'}\right)}_{W} \end{aligned} \tag{5.9}$$

Here, the first summation involves configurations in $\mathbb{C}^{\bar{\mathbf{s}}}$ which is the configuration space induced by $\bar{\mathbf{s}}$. The second summation W involves the (unbounded) set of configurations not in $\mathbb{C}^{\bar{\mathbf{s}}}$. Since $\bar{\mathbf{s}} \in \mathbb{C}^{\bar{\mathbf{s}}}$, it follows that $\forall \mathbf{s}' \in dom(\mathbf{S}) \backslash \mathbb{C}^{\bar{\mathbf{s}}}, \bar{\mathbf{s}} \preceq \mathbf{s}'$. Thus, using Proposition 1, it follows that for all $\mathbf{s}' \in dom(\mathbf{S}) \backslash \mathbb{C}^{\bar{\mathbf{s}}}$

$$P\left(\mathbf{y} \mid \theta^*_{\bar{\mathbf{s}}}, \bar{\mathbf{s}}\right) \leq P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}'}, \mathbf{s}'\right) \leq 1$$

Substituting these bounds in the infinite sum W yields

$$P\left(\mathbf{y} \mid \theta^*_{\bar{\mathbf{s}}}, \bar{\mathbf{s}}\right) \sum_{\mathbf{s}' \in dom(\mathbf{S}) \setminus \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{s}' \mid \theta^*_{\mathbf{s}'}\right) \leq W \leq \sum_{\mathbf{s}' \in \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{s}' \mid \theta^*_{\mathbf{s}'}\right) \tag{5.10}$$

Applying Equation 5.10 directly to Equation 5.9 finally yields the bounds expressed by Equation 5.8.

□

Equation 5.8 has the desirable property that the upper and lower bounds converge on $P(\mathbf{y} \mid \Theta^*)$ as $\bar{\mathbf{s}}$ expands. Namely, as $\mathbb{C}^{\bar{\mathbf{s}}}$ grows with increasing $\bar{\mathbf{s}}$, f in Equation 5.8 approaches $P(\mathbf{y})$ and g approaches 0[3]. Bounds on $P(\mathbf{y} \mid \theta^*_{\mathbf{s}})$ can then be used to bound the posterior of size variables:

Lemma 2. Let $\bar{\mathbf{s}}$ be some fixed configuration where for all $k \in \bar{\mathbf{s}}, k < \infty$. Then for any configuration $\mathbf{s}$,

$$\begin{aligned} &\frac{P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}\right)}{f + g} \leq P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}, \mathbf{y}\right) \leq \\ &\frac{P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}\right)}{f + P\left(\mathbf{y} \mid \theta^*_{\bar{\mathbf{s}}}, \bar{\mathbf{s}}\right) g} \end{aligned} \tag{5.11}$$

where terms f and g are stated in Lemma 1.

Proof. Using Bayes' rule,

$$P(\mathbf{s} \mid \theta^*_{\mathbf{s}}, \mathbf{y}) = \frac{P(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}) P(\mathbf{s} \mid \theta^*_{\mathbf{s}})}{P(\mathbf{y} \mid \Theta^*)} \tag{5.12}$$

and applying Lemma 1 on denominator $P(\mathbf{y} \mid \Theta)$ directly yields Equation 5.11.

□

5.3.2 Non-optimal Parameters and Asymptotic Errors

So far, Lemmas 1 and 2 assumes that optimal parameters $\theta^*_{\mathbf{s}}$ are available for any size configuration $\mathbf{s}$. Instead of using the optimal parameters, here we discuss the case when we use non-optimal parameters.

[3] g approaches 0 because of the diminish tail mass of the prior distribution as $\bar{\mathbf{S}}$ expands.

Often it is the case that $\theta^*_{\mathbf{s}}$ is expensive to calculate. The main bottleneck lies in computing the posterior distribution $p(\mathbf{Z} \mid \mathbf{Y}, \mathbf{S})$ where $\mathbf{Z}$ represents many densely correlated latent variables. Approximate inference procedures such as loopy belief propagation (Kschischang et al., 2001; Yedidia et al., 2003) can be applied in these situations to efficiently acquire an approximation of $\theta^*_{\mathbf{s}}$.

Approximating the latent posterior distribution leads to discrepancies in the likelihood bounds in Lemma 1. Specifically, any choice of $\theta_{\mathbf{s}}$ satisfies $P(\mathbf{y} \mid \theta_{\mathbf{s}}, \mathbf{s}) \leq P(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s})$. By rewriting $P(\mathbf{y} \mid \theta_{\mathbf{s}}, \mathbf{s})$ as $P(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}) - \Delta_{\mathbf{s}}$, where $\Delta_{\mathbf{s}} \geq 0$ is some unknown error, we obtain an approximation $\tilde{f}$ of the finite sum f in Lemma 1:

$$\begin{aligned} \tilde{f} &= \sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{y} \mid \theta_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}\right) \\ &= \sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} \left(P\left(\mathbf{y} \mid \theta^*_{\mathbf{s}}, \mathbf{s}\right) - \Delta_{\mathbf{s}}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}\right) \\ &= f - \sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} \Delta_{\mathbf{s}} P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}\right) \end{aligned} \tag{5.13}$$

As $\mathbb{C}^{\bar{\mathbf{s}}}$ expands, f converges to $P(\mathbf{y} \mid \Theta^*)$ whilst $\tilde{f}$ converges to $P(\mathbf{y} \mid \Theta) - \sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} \Delta_{\mathbf{s}} P(\mathbf{s} \mid \theta^*_{\mathbf{s}})$. Thus, using $\tilde{f}$ in the upper and lower bounds of $P(\mathbf{y} \mid \Theta)$ converge to $P(\mathbf{y} \mid \Theta^*) - \sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} \Delta_{\mathbf{s}} P(\mathbf{s} \mid \theta^*_{\mathbf{s}})$ in the limit. The error term $\sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} \Delta_{\mathbf{s}} P(\mathbf{s} \mid \theta^*_{\mathbf{s}})$ can be seen as an average of likelihood errors for over configuration enumerated.

Devoting more time for parameter learning, e.g. with EM (expectation maximisation (Dempster et al., 1977)) or approximate EM, at each step of the configuration search will reduce $\Delta_{\mathbf{s}} P(\mathbf{s} \mid \theta^*_{\mathbf{s}})$ and hence the error of our bounds in the limit. To quantify the error in the limit is as challenging a problem as quantifying errors of approximate inference algorithms, and thus remains an open question in this work.

5.3.3 Proposed Algorithm

Given data, we learn ILRM parameters by search, and assume that DAG is given as input. Algorithm 2 lists our search-based procedure: The algorithm can be terminated at any time and return error bounds, e.g. if some run-time limit is exceeded or some prescribed error bound width is reached. Continuing the search

Algorithm 2: ILRM-Search

1: Input:
2: $\mathbf{V} = (\mathbf{R}, \mathbf{R}_I, S)$ – relations
3: G – a DAG over $\mathbf{V}$
4: $\mathbf{y}$ – observations
5: Initialise:
6: $\mathcal{Q} = \{(1, 1, \ldots, 1)\}$; // Configuration queue
7: $\Theta = \emptyset$; // Parameters
8: $\mathcal{H} = \emptyset$; // Likelihoods
9: **repeat**
10: Select $\mathbf{s} \in \mathcal{Q}$;
11: Learn parameters $\theta_{\mathbf{s}}$ of ILRM with configuration $\mathbf{s}$;
12: Evaluate likelihood $P(\mathbf{y} \mid \theta_{\mathbf{s}}, \mathbf{s})$, add to $\mathcal{H}$;
13: Compute error bounds $b = \langle l, u \rangle$ using $\mathcal{H}$ (Lemma 1);
14: Add $\theta_{\mathbf{s}}$ to Θ;
15: Add *successors*$(\mathbf{s})$ to $\mathcal{Q}$;
16: **until** termination
17: **return** $\langle \Theta, b \rangle$

will reduce further reduce the error.

For the simple case where all there is only one size variable S in the IBN, Algorithm 2 can be described as a line search depicted by Figure 5.1, where search proceeds from $S = 1$ to $S = k$. Each shaded node abstracts steps 11 to 15 of Algorithm 2.

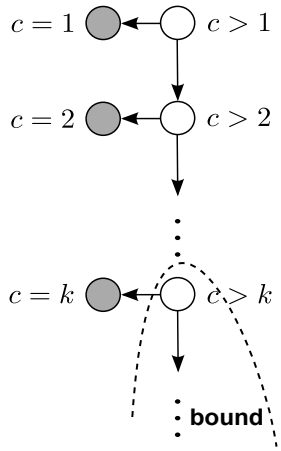


Figure 5.1: single size

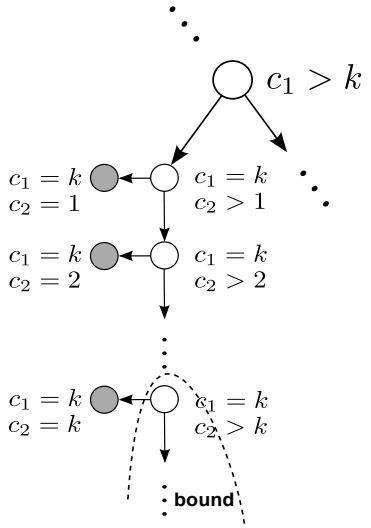


Figure 5.2: Two sizes

Figure 5.1 shows a search-tree over configurations containing only one unknown size variable S. The error from not expanding the infinite sub-tree for $S > k$ is quantified by error bounds (Section 5.3.1). Figure 5.2 depicts a search over a bivariate configuration (S_1, S_2), where S_2 is assumed dependent on S_1. Shaded nodes indicate specific configurations enumerated, i.e. those for whom a Bayesian network is learned. The set of Bayesian networks learned are part of an IBN.

The general idea is that for each size configuration enumerated, a Bayesian network is learned. In Figure 5.1 k networks are learned, and likelihood contributions from networks for the non-enumerated configurations are bound.

For the case when an IBN has more than one size variable, search proceeds in a higher dimensional space. Figure 5.2 depicts the case where there are two size variables, and search proceeds in a two-dimensional space of values of $\mathbf{S} = (S_1, S_2)$. Suppose that IBN graph structure is such that $pa(S_2) = \{S_1\}$, we search by expanding S_1 before S_2 (note that if $pa(S_1) = pa(S_2) = \emptyset$, any order of enumerating is allowed).

We generally want to fully evaluate bounds given by Lemma 1, we start the search at the identity configuration $\mathbf{s} = (1, 1, \ldots, 1)$ and expand the configuration in each dimension such that, at any step, all lesser configurations have already

been enumerated. This will result in a tighter lower-bound of the likelihood than not enumerating all lesser configurations. To do so, we maintain a configuration queue. At each step, a configuration is removed from the queue, and all immediate successors of that configuration not already in the queue are inserted. An immediate successor is obtained by incrementing only one component of the configuration by 1, e.g. immediate successors for $(S_1 = i_1, S_2 = i_2)$ are $(S_1 = i_1 + 1, S_2 = i_2)$ and $(S_1 = i_1, S_2 = i_2 + 1)$.

5.4 Prediction

Here we show how ILRMs support Bayesian predictions. Given data $\mathbf{y}$ and some unobserved query proposition q (e.g. $likes(joe, sue)$), the desired prediction $P(q \mid \mathbf{y})$ is computed by Bayesian model averaging (BMA) (Hoeting et al., 1999)

$$P_{bma}\left(q \mid \mathbf{y}\right) = \sum_{\mathcal{M}} P\left(q \mid \mathcal{M}, \mathbf{y}\right) P\left(\mathcal{M} \mid \mathbf{y}\right) \tag{5.14}$$

Whilst the true BMA solution also requires marginalising over parameters, we describe our averaged prediction in terms of maximum likelihood (ML) parameters[4]. In particular we describe the averaged prediction for the case where optimal ML parameters given any configuration are available:

$$\begin{aligned} P\left(q \mid \mathbf{y}\right) &= \sum_{\mathbf{s} \in dom(\mathbf{S})} P\left(q \mid \theta^*_{\mathbf{s}}, \mathbf{s}, \mathbf{y}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}, \mathbf{y}\right) \\ &= \sum_{\mathbf{s} \in dom(\mathbf{S})} P\left(q \mid \theta^*_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^*_{\mathbf{s}}, \mathbf{y}\right) \end{aligned} \tag{5.15}$$

where we assume $P\left(q \mid \theta^*_{\mathbf{s}}, \mathbf{s}, \mathbf{y}\right) = P\left(q \mid \theta^*_{\mathbf{s}}, \mathbf{s}\right)$ to mean that predictions are drawn from the model only. (Note that a full Bayesian treatment will also require marginalising out ILRM parameters. We assume point-estimate parameters – thus our approach is "semi" Bayesian – but note that we can approximate the full Bayesian solution by assuming Dirichlet priors over parameters the Bayesian treatment is within reach.)

Algorithm 2 learns ILRMs that contain LRMs with all configurations that are

[4] A Bayesian treatment of parameters are not precluded in the following, however.

lesser than the upper-bound $\bar{\mathbf{s}}$; that the configuration space $\mathbb{C}^{\bar{\mathbf{s}}}$ is enumerated. Equation 5.15 can then be written in terms of a finite and infinite sum:

$$P\left(q \mid \mathbf{y}\right) = \underbrace{\sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} P\left(q \mid \theta^{*}_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^{*}_{\mathbf{s}}, \mathbf{y}\right)}_{W} + \underbrace{\sum_{\mathbf{s}' \in dom(\mathbf{S}) \setminus \mathbb{C}^{\bar{\mathbf{s}}}} P\left(q \mid \theta^{*}_{\mathbf{s}'}, \mathbf{s}'\right) P\left(\mathbf{s}' \mid \theta^{*}_{\mathbf{s}'}, \mathbf{y}\right)}_{W'}$$

Using Lemma 2, we bound each $P(\mathbf{s} \mid \theta^{*}_{\mathbf{s}}, \mathbf{y})$ in the finite sum W to obtain

$$\begin{aligned} &\sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} P\left(q \mid \theta^{*}_{\mathbf{s}}, \mathbf{s}\right) \frac{P\left(\mathbf{y} \mid \theta^{*}_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^{*}_{\mathbf{s}}\right)}{f + g} \leq W \leq \\ &\sum_{\mathbf{s} \in \mathbb{C}^{\bar{\mathbf{s}}}} P\left(q \mid \theta^{*}_{\mathbf{s}}, \mathbf{s}\right) \frac{P\left(\mathbf{y} \mid \theta^{*}_{\mathbf{s}}, \mathbf{s}\right) P\left(\mathbf{s} \mid \theta^{*}_{\mathbf{s}}\right)}{f + P\left(\mathbf{y} \mid \theta^{*}_{\bar{\mathbf{s}}}, \bar{\mathbf{s}}\right) g} \end{aligned} \tag{5.16}$$

where f and g are given in Lemma 1. Similarly, we can bound the infinite sum W' by using Lemma 2. However, each summand in W' involves $P(q \mid \theta^{*}_{\mathbf{s}'}, \mathbf{s}')$ whose value is unknown since $\mathbf{s}'$ is not in the enumerated set $\mathbb{C}^{\bar{\mathbf{s}}}$. We replace it with a prior $P(q)$, giving

$$\begin{aligned} &\frac{P\left(q\right)}{f + g} \sum_{\mathbf{s}' \in dom(\mathbf{S}) \setminus \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{y} \mid \theta^{*}_{\mathbf{s}'}, \mathbf{s}'\right) P\left(\mathbf{s}' \mid \theta^{*}_{\mathbf{s}'}\right) \leq W' \leq \\ &\frac{P\left(q\right)}{f + P\left(\mathbf{y} \mid \theta^{*}_{\bar{\mathbf{s}}}, \bar{\mathbf{s}}\right) g} \sum_{\mathbf{s}' \notin S^{\bar{\mathbf{s}}}} P\left(\mathbf{y} \mid \theta^{*}_{\mathbf{s}'}, \mathbf{s}'\right) P\left(\mathbf{s}' \mid \theta^{*}_{\mathbf{s}'}\right) \end{aligned}$$

The likelihood $P\left(\mathbf{y} \mid \theta^{*}_{\mathbf{s}'}, \mathbf{s}'\right)$ is also unknown, but can be bound using Proposition 1 using $\bar{\mathbf{s}}$, resulting in

$$\begin{aligned} &\frac{P\left(\mathbf{y} \mid \theta^{*}_{\bar{\mathbf{s}}}, \bar{\mathbf{s}}\right) P\left(q\right)}{f + g} \sum_{\mathbf{s}' \in dom(\mathbf{S}) \setminus \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{s}' \mid \theta^{*}_{\mathbf{s}'}\right) \leq W' \leq \\ &\frac{P\left(q\right)}{f + P\left(\mathbf{y} \mid \theta^{*}_{\bar{\mathbf{s}}}, \bar{\mathbf{s}}\right) g} \sum_{\mathbf{s}' \in dom(\mathbf{S}) \setminus \mathbb{C}^{\bar{\mathbf{s}}}} P\left(\mathbf{s}' \mid \theta^{*}_{\mathbf{s}'}\right) \end{aligned} \tag{5.17}$$

Then, $P(q \mid \mathbf{y})$ is bounded as follows:

$$L_W + L_{W'} \leq P\left(q \mid \mathbf{y}\right) \leq U_W + U_{W'} \tag{5.18}$$

where $L_W, L_{W'}$ are lower-bounds in Eqs. 5.16 and 5.17 respectively, and $U_W, U_{W'}$ are corresponding upper-bounds.

5.5 Experiments

In our experiments, we again use the *dimensionality of nations* dataset (Rummel, 1999) analysed in (Kemp et al., 2006), which was also used in Ch. 4. The domain consists of 14 countries, and 56 different two-place Boolean relations are observed amongst them. Each relation has the form $r(X, Y)$, where X, Y are logical variables of type *nation*. One latent property α is included.

One of the 56 relations provided, one is chosen as the *target relation* for prediction, and two chosen as *co-relations* on which the target relation can condition on in our LRMs. The target relation is *intergov*, whilst the chosen co-relations are *ngo* and *sever*. The target relation was chosen for its high degree of entropy, whilst the co-relations are chosen based on their mutual information measure with the target relation; the degree to which the co-relation is informative of the target relation. The co-relation *sever* has the lowest mutual information score at 5×10^{-5}, whilst *ngo* has a score of 0.63 (see Table 4.1). High-scoring co-relations are more informative.

In this experiment we first demonstrate characteristics of the bounds derived in Sec. 5.3.1 as we search over the size of α. We also show the effects of different choices of priors.

In the second experiment we learn ILRMs and evaluate its predictions using Eq. 5.18. Like the experiments in Ch. 4 Sec. 4.4, we assess models with different co-relations. We learn our ILRMs by searching over different sizes of α. For each co-relation, we compare the prediction of the ILRM with its best corresponding LRM found in Ch. 4.

5.5.1 Error Bounds

For this experiment we consider an ILRM that models two observed relations *ngo1* and *intergov*, with a latent property α (of countries). The graph structure over relations is given by $G = \{\alpha \rightarrowtail \textit{intergov}, \alpha \rightarrowtail \textit{ngo1}, \textit{ngo1} \rightarrowtail \textit{intergov}\}$. ILRM parameters Θ are learned using Alg. 1.

We search over configurations of the ILRM, from $S = 1$ to $S = 10$. We record the prior distribution, the likelihood bounds, and posterior bounds computed at each step of the search. We use two different prior distributions over configurations: (i) the geometric distribution $\texttt{geom}(\lambda)$ for $\lambda = 0.1, 0.5, 0.9$ and (ii) a Poisson distribution $\texttt{poiss}(\lambda)$ for $\lambda = 1, 5, 9$. The results are shown in Fig. 5.3.

The figure illustrates the behaviour of bounds derived in Sec. 5.3.1, where our uncertainty about the likelihood $P(\mathbf{y})$ leads to uncertainty about the posterior over the size of latent property α. An observation is that using the geometric prior over the size of latent property α, smaller sizes of α favoured in the posterior, and with higher values of the geometric parameter, the bias becomes stronger. Using the Poisson prior (right-hand column of Fig. 5.3), on the other hand, can utilise prior knowledge that different sizes of α may be favourable. The mode of the Poisson can then be centred according to the prior knowledge.

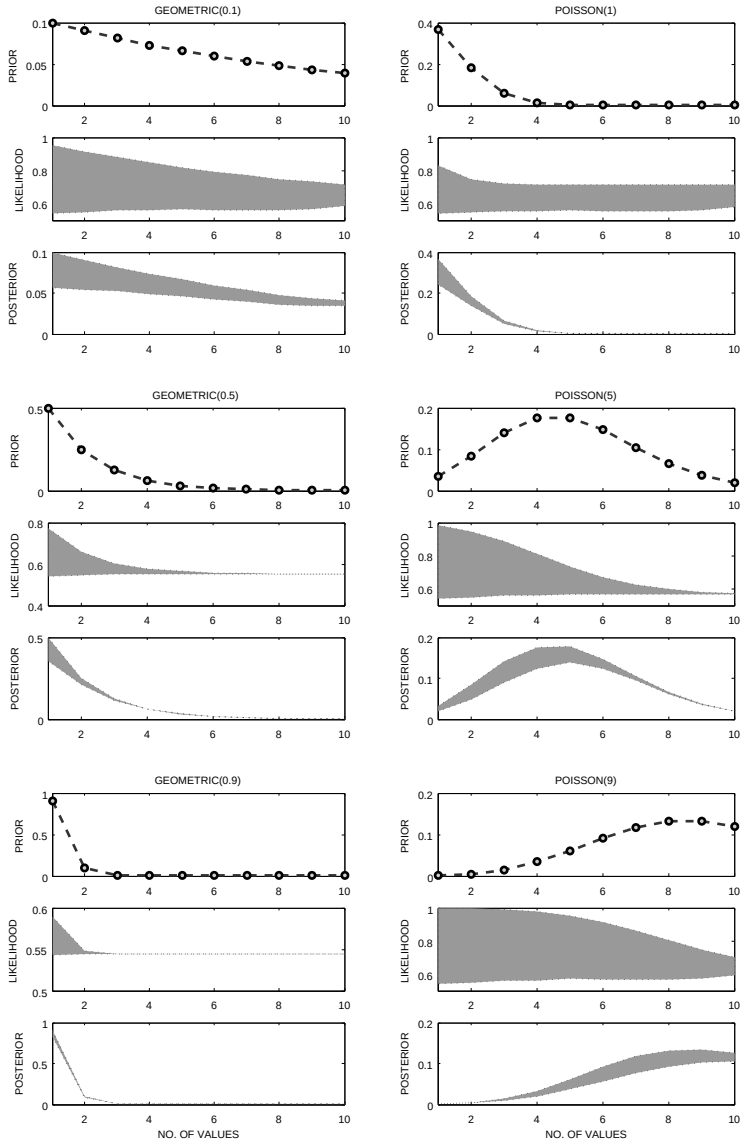


Figure 5.3: Search over configuration of α, from 1 to 10. Left column corresponds to using a geometric distribution as the prior over the size α, with parameter setting 0.1, 0.5 and 0.9. The right column shows uses the Poisson distribution with parameter 1, 5, and 9. Bounds for the likelihood and posterior of the size of α are shown by the shaded regions.

5.5.2 Bayesian Prediction

Next we examine the accuracy of ILRMs (Eq. 5.18) in prediction. For different co-relations, we compare the log-loss of the averaged prediction with the corresponding best single predictor found in Ch. 4 (Table 4.2).

No co-relations

The first test involves only the target relation is *intergov*, with no co-relations. The ILRM for this test – denoted by $\mathcal{L}_c$ – contain only *intergov*, and latent property α. The structure is given by $G = \{\alpha \rightarrowtail \textit{intergov}\}$. Θ is learned using Alg. 1 introduced in Ch. 4. The sizes of α considered are $S = 1, 2, \ldots, 10$. $\mathcal{L}_c$ corresponds to ten fixed-size LRMs which are averaged during prediction. The geometric and Poisson distributions are used as prior distributions over the space of configurations. We search over parameters of the geometric distribution, that is we use $p(S) \sim \texttt{geom}(\lambda)$ for λ, from $0.1, 0.2, \ldots, 1$. For the Poisson distribution, $p(S) \sim \texttt{poiss}(\lambda)$, we search from $\lambda = 1, \ldots, 10$.

For both choices of priors and each parameter value we perform 5-fold cross-validation, where 20% of observed cases for the target relation are used for prediction, and the rest for training. For each fold, 50 ILRMs are learned, each of which learned with size of α up to (and including) $S = 10$. The ILRMs with the best test-set log-loss in each fold are used for computing the cross-validated log-loss. Figure 5.4 shows the log-loss for different choices of prior distributions for $S = 1 \ldots 10$. In order to interpret these results, recall from Ch. 4 that models corresponded $S > 3$ generally yielded lower losses (see Figs. 4.7 to 4.9). As such, in averaging, it is reasonable to expect that weighting those models more heavily will also yield the best results.

From Fig. 5.4, we see that using geometric prior generally leads to higher losses (worse predictive performance) than the Poisson prior. For $\texttt{geom}(\lambda)$, increasing values of λ correspondingly favours lower values of S. Thus, as expected, high losses are incurred with increasing values of λ. It is noteworthy that the CRP prior also defines priors that monotonically weights simpler models (lower values of S) more heavily.

Using $\texttt{poiss}(\lambda)$ on the other hand, the mode can be centred flexibly. Figure

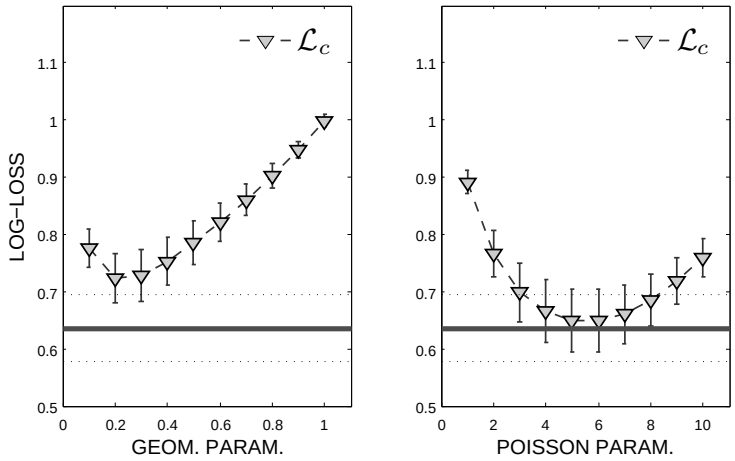


Figure 5.4: Five-fold cross-validated log-loss for ILRM predictions on the *intergov* relation, no co-relations are present. ILRMs with different prior distributions over the size of latent property α are evaluated: the geometric distribution (left) and the Poisson distribution (right) with different parameter values are considered. Horizontal lines in the plots are the score and standard error for the single best LRM found in Ch. 4 for the same problem.

5.4 shows that for $\lambda \sim 5$, where models of configurations around $S = 5$ are weighted more heavily, whilst those farther away have less weight. With $\lambda \sim 5$ we observe the lowest log-losses, almost matching that of the best single model found in Ch. 4 (Fig. 4.7). This is expected as the best single models found in Ch. 4 correspond to α with sizes between 3 and 10, thus weight these models more heavily in the averaged prediction better exploits their accuracy.

With co-relation

The same experiment is repeated with the addition of a co-relation r, and we test two ILRMs. The first, denoted by $\mathcal{L}_{c+}$, contain a target relation, one co-relation, and one latent property α The structure of $\mathcal{L}_{c+}$ is given by $G = \{\alpha \rightarrowtail \textit{intergov}, \alpha \rightarrowtail r\}$, where r denote a co-relation. The second ILRM, denoted by $\mathcal{L}_{cd}$, differs from $\mathcal{L}_{c+}$ by its graph structure, namely that now $r \rightarrowtail \textit{intergov}$ appears in G. Parameter learning given configurations is again done using Alg. 1

from Ch. 4. Results for different choices of co-relation r (i.e. *sever*,*ngo2*, and *ngo*) are shown in Figs. 5.5, 5.6, and 5.7 respectively.

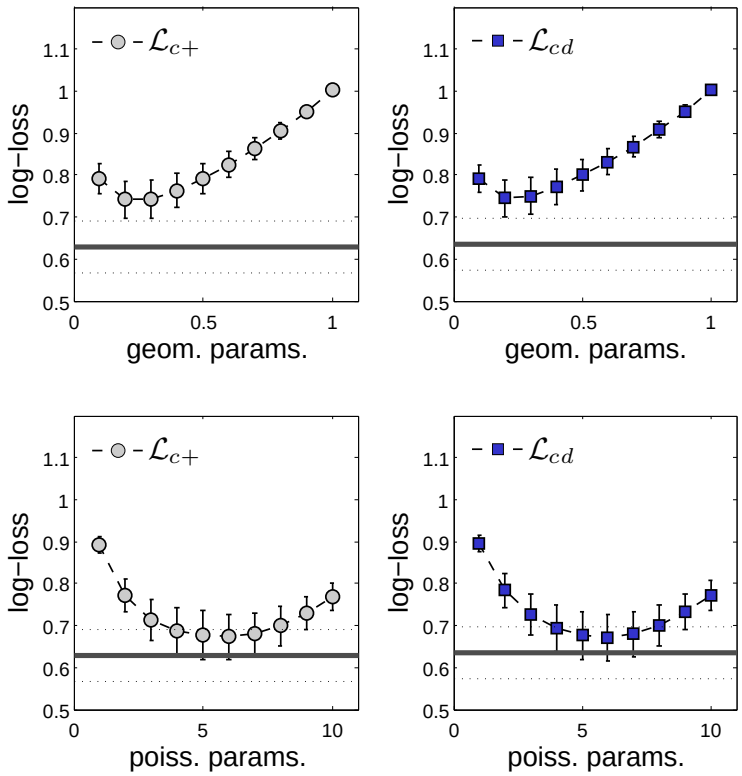


Figure 5.5: Five-fold cross-validated log-loss for ILRM predictions on the *intergov* relation, with co-relation *sever*. ILRMs $\mathcal{L}_{c+}$ (left column, no dependency link between target and co-relation) and $\mathcal{L}_{cd}$ (right column, with a dependency link from co-relation to target relation) are evaluated. Different prior distributions over the size of latent property α are evaluated: the geometric distribution (top row) and the Poisson distribution (bottom row) with different parameter values are considered. Horizontal lines in the plots are the score and standard error for the single best LRM found in Ch. 4 for the same problem.

Using co-relation *sever*, we see that averaged predictions of $\mathcal{L}_c$ and $\mathcal{L}_{cd}$ have similar performance. This is expected, since mutual information between *sever*

and *intergov* is low ($\sim 5 \times 10^{-5}$), and modelling the dependency between *sever* and *intergov* yields little advantage. The important observation is that the choice of the Poisson prior again leads to log-loss that is closer to the best single model.

Using co-relations *ngo2* and *ngo1* generates the following results in Figs. 5.6 and 5.7.

In this case, we observe similar characteristics to the case where *sever* is used as the co-relation. The higher mutual information between *ngo2* and *intergov* brings about a slight advantage for the $\mathcal{L}_{cd}$ model, whilst *ngo1* provides the greatest advantage. The use of $\texttt{poiss}(\lambda)$ for $\lambda \sim 5$ again yields the best log-loss scores, which correspond to models where α has approximately $S \sim 5$ values, and represents the region where the best single models are found.

The general finding of this experiment is that the choice of prior distribution over configurations has an important effect on the accuracy of averaged predictions. In particular, for this dataset, the best single models were found to be those where latent property α has least 3 values, and allowing greater weighting for these models leads to better averaged predictions. The use of a Poisson distribution appears more appropriate for this task, as it is not restricted to favouring small number of values. The results of averaged predictions are shown to be close to the best single models found in Ch. 4 using the Poisson prior.

5.6 Remarks

In this chapter we have demonstrated an alternative to the standard sampling-based method for handling uncertainty over the size of latent properties in relational models. Our proposed method searches over possible sizes of latent properties and bounds the error of the likelihood computed from models found during search. The bounds lead to posterior bounds which then leads to an approximate prediction. The proposed method makes fewer assumptions than existing proposals (e.g. Kemp et al. (2006); Xu et al. (2006)) about the prior distribution over sizes of latent properties, and allows us to examine different forms of this prior.

Our experiments show the behaviour of the bounds, and accuracy of averaged predictions (using only models found in the search) that achieve accuracy close to the single best model found in Ch. 4 for the same dataset, providing that we use an

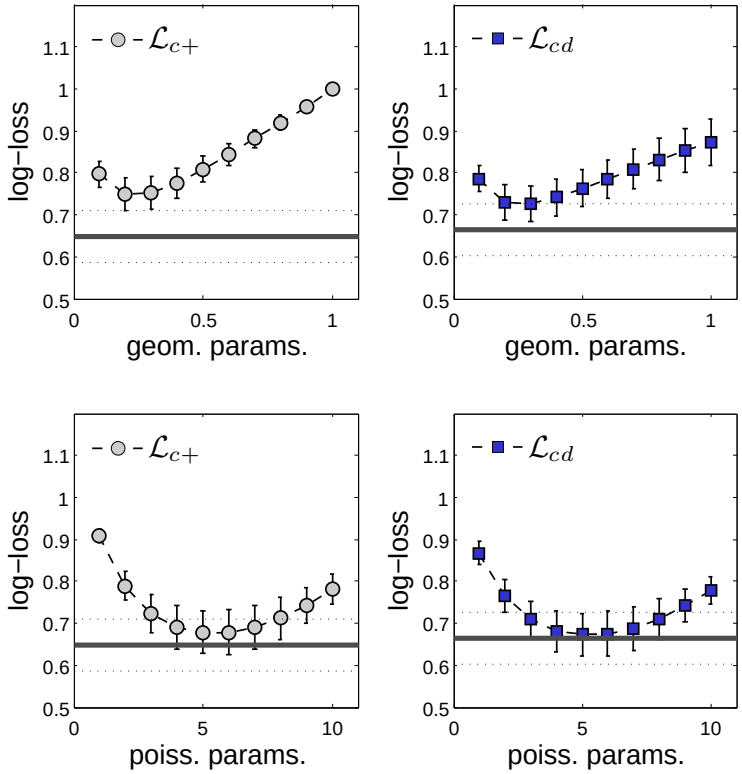


Figure 5.6: Five-fold cross-validated log-loss for ILRM predictions on the *intergov* relation, with co-relation *ngo2*. ILRMs $\mathcal{L}_{c+}$ (left column, no dependency link between target and co-relation) and $\mathcal{L}_{cd}$ (right column, with a dependency link from co-relation to target relation) are evaluated. Different prior distributions over the size of latent property α are evaluated: the geometric distribution (top row) and the Poisson distribution (bottom row) with different parameter values are considered. Horizontal lines in the plots are the score and standard error for the single best LRM found in Ch. 4 for the same problem.

appropriate prior distribution over the size of latent properties.

On a bibliographic note, the search-based framework for bounding the likelihood is based on the search-based method for evaluating probabilities proposed by (Poole, 1993a). Poole's approach performs probabilistic inference by only eval-

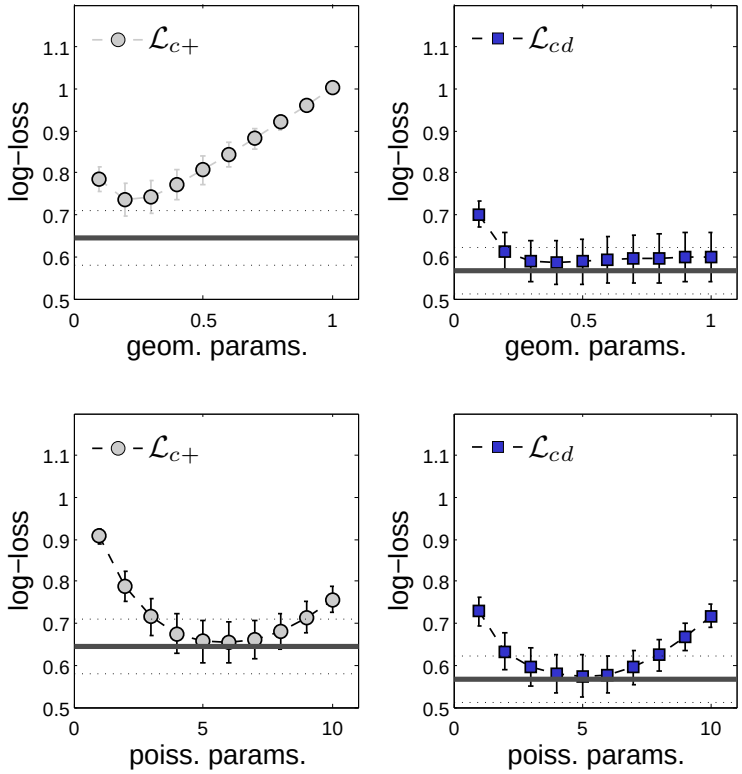


Figure 5.7: Five-fold cross-validated log-loss for ILRM predictions on the *intergov* relation, with co-relation *ngo1*. ILRMs $\mathcal{L}_{c+}$ (left column, no dependency link between target and co-relation) and $\mathcal{L}_{cd}$ (right column, with a dependency link from co-relation to target relation) are evaluated. Different prior distributions over the size of latent property α are evaluated: the geometric distribution (top row) and the Poisson distribution (bottom row) with different parameter values are considered. Horizontal lines in the plots are the score and standard error for the single best LRM found in Ch. 4 for the same problem.

uating part of the (finite) state space and bounding the remaining mass. We use this approach to evaluate the likelihood bounds the where state space is infinite. Also, the general scheme of successively increasing the number of parameters of statistical model, where the parameter space is infinite, can be related to the *method*

of sieves for non-parametric ML estimation Geman and Hwang (1982). More in depth connections of our approach to the method of sieves would be interesting for future work.

Chapter 6

Learning with Functional Relations

In previous chapters we have only considered rules and dependencies that contain relations. In this chapter the focus is on rules where a *functional relation* appears in the body, e.g.

$$\begin{array}{llll} \forall X_1, \ldots, \forall X_n & \theta_1: & r(X_1, \ldots, X_n) = v \leftarrow f(X_1, \ldots, X_n) = y_1. \\ \forall X_1, \ldots, \forall X_n & \theta_2: & r(X_1, \ldots, X_n) = v \leftarrow f(X_1, \ldots, X_n) = y_2. \\ & & \vdots \\ \forall X_1, \ldots, \forall X_n & \theta_m: & r(X_1, \ldots, X_n) = v \leftarrow f(X_1, \ldots, X_n) = y_m. \end{array} \tag{6.1}$$

where r is a relation, and f is a function with range $V_f = \{y_1, \ldots, y_m\}$. Each θ_i is a probability estimate learned from data, described by Eq. 2.11.

To illustrate, consider a simple example drawn from the 1990 U.S. Census (U.S. Census Bureau, 1990), which records preferred modes of transportation taken to work across major U.S. cities. Using the Boolean predicate *drive_alone* to represent driving alone to work, and function *lives_in* to represent the city of residence,

the example can be represented by the following set of rules, one rule per city.

$$\begin{aligned}
\forall Person, \quad \theta_1 : \quad & \textit{drive_alone}(Person) \leftarrow \textit{lives_in}(Person) = \textit{new_york_city} \\
\forall Person, \quad \theta_2 : \quad & \textit{drive_alone}(Person) \leftarrow \textit{lives_in}(Person) = \textit{los_angeles} \\
& \vdots \\
\forall Person, \quad \theta_m : \quad & \textit{drive_alone}(Person) \leftarrow \textit{lives_in}(Person) = \textit{willmington}
\end{aligned} \tag{6.2}$$

where each θ_i is a proportion of people who drive alone to work in a corresponding city. Thus, if x lives in New York city, the probability that x drives to work is θ_1, if x lives in Los Angeles then θ_2, and so on.

The number of rules required to model drive-alone probabilities is proportional to the the number of known cities. Ideally, we would like the representation size of our model to be independent of the number of individuals (e.g. the number of cities in the example). In addition, we wish to avoid over-fitting, which can occur when statistics accompanying a rule is drawn from an unreliable samples, e.g. the drive-alone statistics of small towns.

A standard solution to obtain a compact representation is *aggregation*, where the probabilities over all cities are combined, resulting in a general statement

$$\forall Person, \quad \bar{\theta} : \textit{drive\_alone}(Person). \tag{6.3}$$

where $\bar{\theta}$ represents some aggregate probability, i.e. the drive-alone probability averaged over all cities.

From the standpoint of compactness, Eq. 6.3 is the obvious choice as it is a single rule. However, in prediction, Eq. 6.2 is more precise. Note that we are choosing between two reference classes here, one general and one specific.

To highlight the implications on prediction, we make the example more precise. In the 1990 U.S. census (U.S. Census Bureau, 1990), approximately 18 million people were surveyed across 50 major U.S. cities about their drive-to-work habits. It was determined that approximately 72% of the sample drive to work. The data also shows that about 3 million samples were taken from New York city, which has an average drive-to-work rate of 24%. Thus, applying Eq. 6.3 and predicting that future subjects drive to work with probability 0.72 (without considering the city of

residence), is likely to incur high estimation errors for subjects coming from New York city. Since New York city has a large population (approximately 8.4 million), a high accumulated error can be expected by using Eq. 6.3.

Our goal in this chapter is an algorithm for finding a fixed-size representation that leverages the tension between compactness of representation and predictive accuracy. The task is to find optimal clusters of values of the given function, where optimality is achieved by minimising empirical loss. For the travel to work example, we seek a fixed number of rules that summarise Eq. 6.2, e.g.

$$\begin{aligned}
&\forall Person, \bar{\theta}^1 : \textit{drive_alone}(Person) \leftarrow \textit{member}(\textit{lives_in}(Person), L_1).\\
&\forall Person, \bar{\theta}^2 : \textit{drive_alone}(Person) \leftarrow \textit{member}(\textit{lives_in}(Person), L_2).\\
&\qquad\qquad\vdots\\
&\forall Person, \bar{\theta}^k : \textit{drive_alone}(Person) \leftarrow \textit{member}(\textit{lives_in}(Person), L_k).
\end{aligned}$$

where $k \leq m$, L_i is a set of cities, and *member*(Y, L_i) is a invented membership relation that is true if city Y belong to group L_i, and false otherwise.

6.1 Problem Statement

We assume a database $\mathbb{D}$ that entails a relation $r(X_1, \ldots, X_n)$ and a function $f(X_1, \ldots, X_n)$. From $\mathbb{D}$ the following table of statistics can be formed

$f(X_1, \ldots, X_n)$	count	total
y_1	c_1^+	c_1
y_2	c_2^+	c_2
$\vdots$	$\vdots$	$\vdots$
y_m	c_m^+	c_m

where c_i^+ and c_i correspond to counts of observed cases where relation $r(X_1, \ldots, X_n)$ has value v where $X_1, \ldots, X_n$ satisfies $f(X_1, \ldots, X_n) = y_i$, given by

$$\begin{aligned}
c_i^+ &= \#_{\mathbb{D}}(r(X_1, \ldots, X_n) = v \wedge f(X_1, \ldots, X_n) = y_i)\\
c_i &= \sum_{v' \in V_r} \#_{\mathbb{D}}(r(X_1, \ldots, X_n) = v' \wedge f(X_1, \ldots, X_n) = y_i)
\end{aligned} \tag{6.4}$$

We also take an integer $k \leq m$ as input. From the table, the following rule set can be formed:

$$\begin{aligned}
&\forall X_1, \ldots, \forall X_n, \quad \theta_1 : r(X_1, \ldots, X_n) \leftarrow f(X_1, \ldots, X_n) = y_1 \\
&\forall X_1, \ldots, \forall X_n, \quad \theta_2 : r(X_1, \ldots, X_n) \leftarrow f(X_1, \ldots, X_n) = y_2 \\
&\qquad\qquad\qquad \vdots \\
&\forall X_1, \ldots, \forall X_n, \quad \theta_m : r(X_1, \ldots, X_n) \leftarrow f(X_1, \ldots, X_n) = y_m
\end{aligned} \tag{6.5}$$

where each $\theta_i = c_i^+ / c_i$.

The learning problem of interest here is to generate the best set of k rules that summarise the verbose rule set in Eq. 6.5, and our strategy is to partition V_f.

Given a set of partitions $S = \{S_1, \ldots, S_k\}$ such that $\bigcup_{i=1}^{k} S_i = V_f$ and $S_i \cap S_j = \emptyset$ if $i \neq j$, the summary rule set we seek has the form

$$\begin{aligned}
&\forall X_1, \ldots, \forall X_n, \quad \bar{\theta}_1 : r(X_1, \ldots, X_n) \leftarrow \textit{member}\,(f(X_1, \ldots, X_n), S_1) \\
&\forall X_1, \ldots, \forall X_n, \quad \bar{\theta}_2 : r(X_1, \ldots, X_n) \leftarrow \textit{member}\,(f(X_1, \ldots, X_n), S_2) \\
&\qquad\qquad\qquad \vdots \\
&\forall X_1, \ldots, \forall X_n, \quad \bar{\theta}_k : r(X_1, \ldots, X_n) \leftarrow \textit{member}\,(f(X_1, \ldots, X_n), S_k)
\end{aligned} \tag{6.6}$$

where $\textit{member}(A, B)$ is true if $A \in B$, and false otherwise. We consider the best clustering S^* as the one that yields the lowest empirical loss. First let $T = \{T_1, \ldots, T_k\}$ be the corresponding clustering of probabilities induced by S, where each T_i contains the set of probabilities from the verbose rule set (Eq. 6.5) for members of S_i. For instance, if $S_i = \{y_1, y_2\}$, the $T_i = \{\theta_1, \theta_2\}$. Each T_i is said to contain the partition parameters for partition S_i. The optimisation problem can be formally stated as follows.

$$S^* = \arg\min_{S} \sum_{i=1}^{k} \mathbb{E}\left[\mathbf{L}(\bar{\theta}_i, T_i)\right] \tag{6.7}$$

where $\mathbb{E}[\mathbf{L}(\bar{\theta}_i, T_i)]$ is short for the expected empirical loss of applying $\bar{\theta}_i$ over all $\theta \in T_i$.

Given a partitioning S, the probability value $\bar{\theta}_i$ can also be optimised. Namely, $\mathbb{E}\left[\mathbf{L}(\bar{\theta}_i, T_i)\right]$ is the empirical loss incurred by $\bar{\theta}_i$ in for partition parameters T_i, the $\bar{\theta}_i$ that minimises $\mathbb{E}\left[\mathbf{L}(\bar{\theta}_i, T_i)\right]$ is given by

$$\bar{\theta}_i^* = \arg\min_{\bar{\theta}_i} \mathbb{E}\left[\mathbf{L}(\bar{\theta}_i, T_i)\right] = \arg\min_{\bar{\theta}_i} \mathbb{E}\left[\sum_{\theta \in T_i} \mathbf{L}(\bar{\theta}_i, \theta)\right] \tag{6.8}$$

We assume log-loss in this work, i.e. $l_{1,\alpha} = -\log \alpha$ and $l_{0,\alpha} = -\log(1-\alpha)$, and since $\mathbf{L}(\cdot)$ is a convex function, taking the derivative and equating to zero yields σ_i^* as the average of T_i:

$$\bar{\theta}_i^* = \frac{\sum_{\theta \in T_i} \theta}{|T_i|} \tag{6.9}$$

The remaining problem is to solve Eq. 6.7 to find the best clustering clustering S^*, which we discuss next.

6.2 Optimal Partitioning

The space of possible partitionings consist of all possible assignments of elements of V_f to k partitions. Assigning m elements to $k < m$ bins yields k^m possible assignments. We show in the following, that to assign m probabilities in Θ to $k < m$ partitions, we only need to consider partitionings such that partitions are ordered. Specifically, we show that the optimal partitioning is $S^* = \{S_1^*, \ldots, S_k^*\}$ with corresponding partition parameters $T^* = \{T_1^*, \ldots, T_k^*\}$ such that for all $\theta \in T_i^*$ and $\theta' \in T_{i+1}^*$, $\theta \geq \theta'$. The following lemma provide the theoretical justification for this result. Notation for empirical loss is provided in Ch. 2 Sec. 2.1.3.

Lemma 3. Let p_1 and p_2 be probabilities, such that $p_1 > p_2$. Also, let t_1 and $\hat{p}_2$ be probabilistic estimates of p_1 and p_2 respectively, such that $\hat{p}_1 > \hat{p}_2$. Applying each estimator to p_1 and p_2,

$$\mathbf{L}(\hat{p}_2, p_1) \leq \mathbf{L}(\hat{p}_1, p_1) \rightarrow \mathbf{L}(\hat{p}_2, p_2) < \mathbf{L}(\hat{p}_1, p_2) \tag{6.10}$$

Proof. The premise $\mathbf{L}(\hat{p}_2, p_1) \leq \mathbf{L}(\hat{p}_1, p_1)$ in Eq. 6.10 has the expanded form

$$p_1 l_{1,\hat{p}_2} + (1 - p_1)\, l_{0,\hat{p}_2} \leq p_1 l_{1,\hat{p}_1} + (1 - p_1)\, l_{0,\hat{p}_1} \tag{6.11}$$

Rewriting p_1 as $p_1 = p_2 + \delta$ where $\delta > 0$, Eq. 6.11 becomes

$$(p_2 + \delta)\, l_{1,\hat{p}_2} + (1 - p_2 - \delta)\, l_{0,\hat{p}_2} \leq (p_2 + \delta)\, l_{1,\hat{p}_1} + (1 - p_2 - \delta)\, l_{0,\hat{p}_1}$$

Some rearrangement yields

$$\underbrace{p_2 l_{1,\hat{p}_2} + (1 - p_2)\, l_{0,\hat{p}_2}}_{\mathbf{L}(\hat{p}_2, p_2)} + \delta\,(l_{1,\hat{p}_2} - l_{0,\hat{p}_2}) \leq \underbrace{p_2 l_{1,\hat{p}_1} + (1 - p_2)\, l_{0,\hat{p}_1}}_{\mathbf{L}(\hat{p}_1, p_2)} + \delta\,(l_{1,\hat{p}_1} - l_{0,\hat{p}_1})$$

It follows that

$$\mathbf{L}\,(\hat{p}_2, p_2) + \delta\,(l_{1,\hat{p}_2} - l_{0,\hat{p}_2}) \leq \mathbf{L}\,(\hat{p}_1, p_2) + \delta\,(l_{1,\hat{p}_1} - l_{0,\hat{p}_1}) \tag{6.12}$$

Since $l_{1,\cdot}$ and $l_{0,\cdot}$ are increasing functions and that $\hat{p}_1 > \hat{p}_2$, it follows that $l_{1,\hat{p}_2} > l_{1,\hat{p}_1}$ and $l_{0,\hat{p}_2} < l_{0,\hat{p}_1}$, and therefore

$$l_{1,\hat{p}_2} - l_{0,\hat{p}_2} > l_{1,\hat{p}_1} - l_{0,\hat{p}_1} \tag{6.13}$$

Given inequality 6.13, for Eq. 6.12 to be true, it then must be the case that

$$\mathbf{L}\,(\hat{p}_2, p_2) < \mathbf{L}\,(\hat{p}_1, p_2) \tag{6.14}$$

which proves Eq. 6.10. Note that an equivalent statement of Eq. 6.10 is that

$$\mathbf{L}\,(\hat{p}_1, p_2) \leq \mathbf{L}\,(\hat{p}_2, p_2) \rightarrow \mathbf{L}\,(\hat{p}_1, p_1) < \mathbf{L}\,(\hat{p}_2, p_1) \tag{6.15}$$

□

Figure 6.1 illustrates Lem. 3, for the case where the premise $\mathbf{L}\,(\hat{p}_2, p_1) \leq \mathbf{L}\,(\hat{p}_1, p_1)$ holds and when it does not. When the premise holds, the optimal assignment of estimators is that both $\hat{p}_1$ and $\hat{p}_2$ be assigned for p_1. An equivalent statement is that when $\mathbf{L}\,(\hat{p}_1, p_2) \leq \mathbf{L}\,(\hat{p}_2, p_2)$ holds, assigning both $\hat{p}_1$ and $\hat{p}_2$ to

p_2 is optimal.

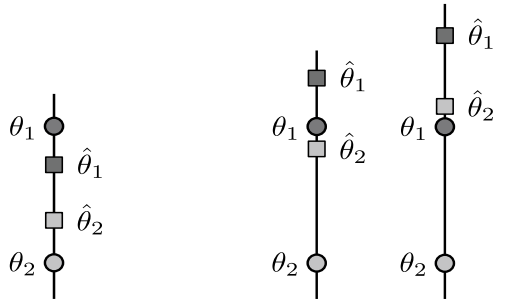


Figure 6.1: An illustration of Lem. 3. The first figure (left) shows when $\mathbf{L}(\hat{p_2}, p_1) \not\leq \mathbf{L}(\hat{p_1}, p_1)$, i.e. when the premise of Eq. 6.10 in Lem. 3 does not hold. When the premise holds, we see that the distance between $\hat{p}_2$ and p_1 is less than that between $\hat{p}_1$ and p_1.

Importantly, Lem. 3 is logically equivalent to the statement that either $\mathbf{L}(\hat{p}_2, p_1) \leq \mathbf{L}(\hat{p}_1, p_1)$ is true, or that $\mathbf{L}(\hat{p}_1, p_2) \leq \mathbf{L}(\hat{p}_2, p_2)$ is true, but not both. In other words, assigning $\hat{p}_1$ to estimate p_2 and $\hat{p}_2$ to p_1 cannot simultaneously achieve the lowest loss for both p_1 and p_2. Thus, to estimate the value of two probabilities $p_1 > p_2$, the optimal choice is to assign estimates in a sorted order.

For our partitioning problem, suppose we sort the set of probabilities $\Theta = \{\theta_1, \ldots, \theta_m\}$ in descending order, then split into $k < m$ partitions and generate a $\bar{\theta}_i$ for each partition i according to Eq. 6.8. It will also be the case that $\bar{\theta}_1, \ldots, \bar{\theta}_k$ will be in descending order. According to Lem. 3, swapping any pair θ_i, θ_j from different partitions will necessarily increase the loss incurred. As such, the optimal partitioning lies in the space of partitionings on a sorted set Θ. The optimal partitioning S^* thus has accompanying partition parameters T^* such that

$$\forall \theta_i \in T_i^* \forall \theta_j \in T_j^*, \quad \theta_i > \theta_j \quad \textit{where} \quad i < j \tag{6.16}$$

Figure 6.2 illustrates the partition process on a sorted set of probabilities.

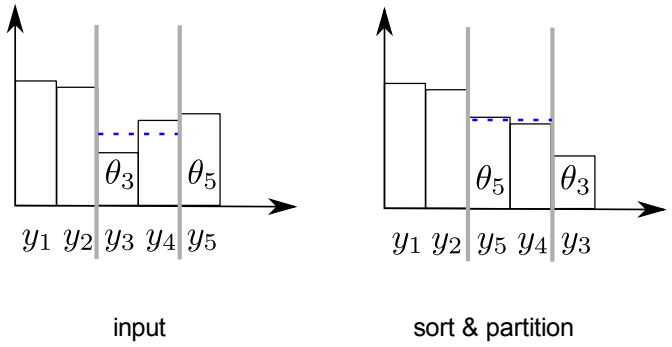


Figure 6.2: An illustration partitioning on an initial set $y_1, \ldots, y_5$ (left), and partitioning on a sorted set (right). The dashed line in the middle partition is a probability estimate for the partition, averaged over probabilities assigned to the partition. The illustration shows that swapping y_3 and y_5 to yield the sorted set reduces the error incurred by the estimate.

Having shown that the optimal partitioning lies in the space of partitionings with ordered sets of probabilities, the remaining task is to find optimal *cut-points* in this set. The following section describes a dynamic programming algorithm for this task.

6.3 Optimal Partitioning by Dynamic Programming

The previous section showed that the optimal partitioning lies in the space of partitionings where probabilities are sorted. As such, the objective function (Eq. 6.7) can be solved recursively, i.e.

$$\min_{S} \sum_{i=1}^{k} \mathbb{E}\left[\mathbf{L}\left(\bar{\theta}_i^*, T_i\right)\right] = \min_{S_k} \mathbb{E}\left[\mathbf{L}\left(\bar{\theta}_k^*, T_k\right)\right] + \min_{S-S_k} \sum_{i=1}^{k-1} \mathbb{E}\left[\mathbf{L}\left(\bar{\theta}_i^*, S_i\right)\right] \quad (6.17)$$

With the objective function in recursive form, dynamic programming can be applied.

Let $\Theta = \{\theta_1, \ldots, \theta_m\}$ be the accompanying probabilities to the rule set in Eq. 6.5. Let π be an ordering that sorts Θ, resulting in Θ_π. To find the best $k-1$ cut-points $c^* = \{c_1^*, \ldots, c_{k-1}^*\}$ (for k partitions of Θ_π), the dynamic programming procedure is a recursion such that for each candidate cut-point c_i in the search for c_i^*, a recursive call is done to find c_{i-1}^*.

We implement the procedure in by constructing a table with m rows and k columns. To define table entries, let Θ_π^i be first i elements of Θ_π, and S^i and T^i be corresponding partitioning and partition parameters on Θ_π^i. Then, $d_{i,j}$ is the empirical loss of the best $(j+1)$-partition model on Θ_π^i

$$d_{i,j} = \min_{\{S_1^i, \ldots, S_{j+1}^i\}} \sum_{n=1}^{j+1} \mathbb{E}\left[\mathbf{L}\left(\bar{\theta}_n^*, T_n^i\right)\right] \tag{6.18}$$

where each $T_{n \leq j}$ consists of elements of Θ_π^i. $d_{i,j}$ also has a recursive definition

$$d_{i,j} = \begin{cases} d_{i-1,j-1} + \min\limits_{c \leq i} d_{c,j} \; ; & i > 0 \textit{ and } j > 0 \\ 0 \; ; & \textit{otherwise} \end{cases} \tag{6.19}$$

The main procedure of our partitioning algorithm is to populate this table column by column, starting from $d_{1,1}$. Namely, we start from a 1-partition model and build incrementally to the k-partition model. The entry $d_{m,k}$ represents the empirical loss incurred. After completing the table, the cut-points that yields the optimal partitioning S^* is read from the table, i.e.

$$c_j^* = \arg\min_c d_{c,j}; \quad j = 1, \ldots, k-1 \tag{6.20}$$

The algorithm for completing the table for partitioning is given as follows. Note that the term $d_{i-1,j-1}$ is directly found in the table, and $d_{c,j}$ can be computed using Eq. 6.18.

The process of partitioning by completing the table is illustrated in Fig. 6.3 below.

Algorithm 3: Completing the table of partitioning losses for dynamic programming

for $j = 1 \ldots k$ **do**
  for $i = 1 \ldots m$ **do**
    $H = \emptyset$
    for $c = 1 \ldots i$ **do**
      Add $(d_{i-1,j-1} + d_{c,j})$ to H
    end for
    $d_{i,j} \leftarrow \min H$
  end for
end for

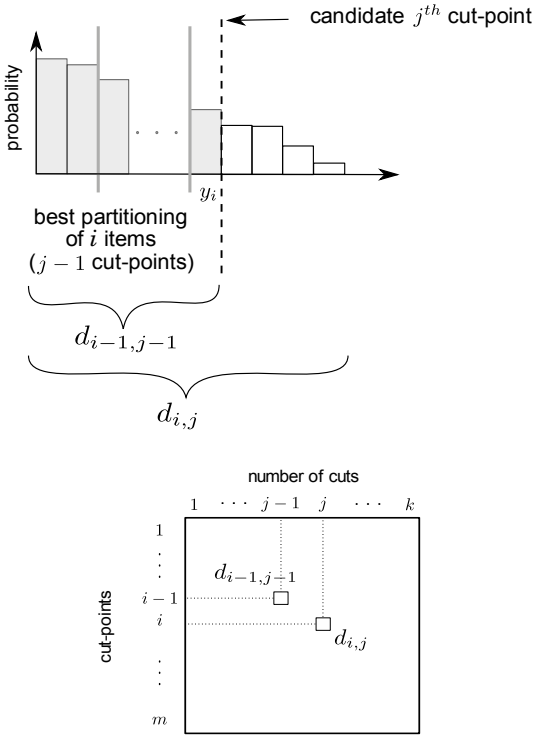


Figure 6.3: An illustration of optimal partitioning by dynamic programming.

6.4 Experiments

6.4.1 Journey-to-work data

In this experiments we use statistics collected about people's travel-to-work habits, collected in the 1990 United States Census (U.S. Census Bureau, 1990). The aim is to demonstrate partitionings generated by Alg. 3. The input is a table percentages corresponding to different ways people travel to work in 50 major U.S. cities. A sample is shown below.

city	pop.	drove alone	carpool	public transit	other
New York, NY	3,183,088	24.0 %	8.5 %	53.4 %	14.0 %
Los Angeles, CA	1,629,096	65.2 %	15.4 %	10.5 %	8.9 %
⋮	⋮	⋮	⋮	⋮	⋮
Omaha, NE	166,449	78.0 %	12.2 %	3.2 %	6.8 %
Toledo, OH	137,772	81.5 %	10.5 %	3.0 %	5.2 %
Buffalo, NY	127,790	61.6 %	13.7 %	13.4 %	11.3 %

The table can be represented as a set of rules of the form of Eq. 6.5. For example, the 'drove alone' class can be described by the following rules

$$\begin{aligned}
&\forall X \quad 0.24 : \textit{drove_alone}(X) \leftarrow \textit{lives_in}(X) = \textit{"New York, NY"} \\
&\forall X \quad 0.65 : \textit{drove_alone}(X) \leftarrow \textit{lives_in}(X) = \textit{"Los Angeles, CA"} \\
&\qquad \vdots \\
&\forall X \quad 0.78 : \textit{drove_alone}(X) \leftarrow \textit{lives_in}(X) = \textit{"Omaha, NE"} \\
&\forall X \quad 0.81 : \textit{drove_alone}(X) \leftarrow \textit{lives_in}(X) = \textit{"Toledo, OH"} \\
&\forall X \quad 0.62 : \textit{drove_alone}(X) \leftarrow \textit{lives_in}(X) = \textit{"Buffalo, NY"}
\end{aligned} \tag{6.21}$$

One can use this rule set as a predictive model for where people drive to work, based on their city of residence. The aim of Alg. 3 is to yield simpler models by partitioning cities. Table 6.1 shows the results for $k = 2, 5$ and 10.

The partitions correspond to simpler rule-sets than Eq. 6.21. For instance, the

Toledo, OH	0.813
Oklahoma City, OK	0.808
Tulsa, OK	0.807
Virginia Beach, VA	0.783
Indianapolis, IN 2/	0.781
Nashville-Davidson, TN 2/	0.780
Albuquerque, NM	0.780
Omaha, NE	0.778
Fresno, CA	0.778
Charlotte, NC	0.772
San Jose, CA	0.768
Fort Worth, TX	0.767
Columbus, OH	0.765
Jacksonville, FL 2/	0.755
Memphis, TN	0.754
Kansas City, MO	0.747
El Paso, TX	0.740
Phoenix, AZ	0.737
Austin, TX	0.737
San Antonio, TX	0.734
Dallas, TX	0.724
Houston, TX	0.717
Sacramento, CA	0.717
San Diego, CA	0.708
Long Beach, CA	0.699
Tucson, AZ	0.698
Denver, CO	0.685
Detroit, MI	0.679
Cincinnati, OH	0.673
Milwaukee, WI	0.672
St. Louis,	0.665
Los Angeles, CA	0.653
Portland, OR	0.650
Cleveland, OH	0.647
Buffalo, NY	0.616
Atlanta, GA	0.612
Miami, FL	0.609
Minneapolis, MN	0.603
Seattle, WA	0.587
New Orleans, LA	0.586
Oakland, CA	0.569
Honolulu CDP, HI	0.550
Baltimore, MD	0.510
Pittsburgh, PA	0.489
Chicago, IL	0.463
Philadelphia, PA	0.447
Boston, MA	0.401
San Francisco, CA	0.385
Washington, DC	0.350
New York, NY	0.240
log-loss (train) = 0.8811	

Toledo, OH	0.813
Oklahoma City, OK	0.808
Tulsa, OK	0.807
Virginia Beach, VA	0.783
Indianapolis, IN 2/	0.781
Nashville-Davidson, TN 2/	0.780
Albuquerque, NM	0.780
Omaha, NE	0.778
Fresno, CA	0.778
Charlotte, NC	0.772
San Jose, CA	0.768
Fort Worth, TX	0.767
Columbus, OH	0.765
Jacksonville, FL 2/	0.755
Memphis, TN	0.754
Kansas City, MO	0.747
El Paso, TX	0.740
Phoenix, AZ	0.737
Austin, TX	0.737
San Antonio, TX	0.734
Dallas, TX	0.724
Houston, TX	0.717
Sacramento, CA	0.717
San Diego, CA	0.708
Long Beach, CA	0.699
Tucson, AZ	0.698
Denver, CO	0.685
Detroit, MI	0.679
Cincinnati, OH	0.673
Milwaukee, WI	0.672
St. Louis,	0.665
Los Angeles, CA	0.653
Portland, OR	0.650
Cleveland, OH	0.647
Buffalo, NY	0.616
Atlanta, GA	0.612
Miami, FL	0.609
Minneapolis, MN	0.603
Seattle, WA	0.587
New Orleans, LA	0.586
Oakland, CA	0.569
Honolulu CDP, HI	0.550
Baltimore, MD	0.510
Pittsburgh, PA	0.489
Chicago, IL	0.463
Philadelphia, PA	0.447
Boston, MA	0.401
San Francisco, CA	0.385
Washington, DC	0.350
New York, NY	0.240
log-loss (train) = 0.8716	

Toledo, OH	0.813
Oklahoma City, OK	0.808
Tulsa, OK	0.807
Virginia Beach, VA	0.783
Indianapolis, IN 2/	0.781
Nashville-Davidson, TN 2/	0.780
Albuquerque, NM	0.780
Omaha, NE	0.778
Fresno, CA	0.778
Charlotte, NC	0.772
San Jose, CA	0.768
Fort Worth, TX	0.767
Columbus, OH	0.765
Jacksonville, FL 2/	0.755
Memphis, TN	0.754
Kansas City, MO	0.747
El Paso, TX	0.740
Phoenix, AZ	0.737
Austin, TX	0.737
San Antonio, TX	0.734
Dallas, TX	0.724
Houston, TX	0.717
Sacramento, CA	0.717
San Diego, CA	0.708
Long Beach, CA	0.699
Tucson, AZ	0.698
Denver, CO	0.685
Detroit, MI	0.679
Cincinnati, OH	0.673
Milwaukee, WI	0.672
St. Louis,	0.665
Los Angeles, CA	0.653
Portland, OR	0.650
Cleveland, OH	0.647
Buffalo, NY	0.616
Atlanta, GA	0.612
Miami, FL	0.609
Minneapolis, MN	0.603
Seattle, WA	0.587
New Orleans, LA	0.586
Oakland, CA	0.569
Honolulu CDP, HI	0.550
Baltimore, MD	0.510
Pittsburgh, PA	0.489
Chicago, IL	0.463
Philadelphia, PA	0.447
Boston, MA	0.401
San Francisco, CA	0.385
Washington, DC	0.350
New York, NY	0.240
log-loss (train) = 0.8702	

Table 6.1: Tables showing 50 surveyed U.S. cities with sampled percentages of people who drive to work in each city. Each table shows a partitioning generated by Alg. 3, for $k = 2$ (left), $k = 5$ (centre), and $k = 10$ (right). Log-loss on training data for each partition model is shown at the bottom.

$k = 2$ partitioning can be represented by

$$\forall X \quad \bar{\theta}_1 : \mathit{drove\_alone}(X) \leftarrow \mathit{member}(\mathit{lives\_in}(X), L_1)$$
$$\forall X \quad \bar{\theta}_2 : \mathit{drove\_alone}(X) \leftarrow \mathit{member}(\mathit{lives\_in}(X), L_2)$$

where

$$L_1 = \{\text{"Toledo, OH", "Oklahoma City, OK"}, \ldots\}$$
$$L_2 = \{\text{"Buffalo, NY", "Atlanta, GA"}, \ldots\}$$

.

The benefit of Alg. 3 for achieving a more compact model than that obtained by directly splitting on all values of the function is demonstrated here. In each partition the probability of drive-to-work appear similar.

6.4.2 E. coli Gene Data

In this experiment we evaluate the predictive accuracy of partition models generated by Alg. 3. We use the 2001 KDD E. coli dataset[1], which contains data for 4289 genes of the E. coli bacterium genome. For each gene, a function classification is provided (there are 144 possible gene functions[2]), along with numerous biological properties. In addition, the interaction of each gene with other genes in the E. coli genome are given. We consider is task of predicting interactions or *link* between genes, i.e. $\textit{link}(Gene, Gene')$, from their respective function classifications, i.e. $f(Gene)$ and $f(Gene')$. We do not compare with other models tested in KDD Cup 2001, due to the contrasting nature of our model.

We test three models, the first of which is a simple model

$$\forall G \forall G' \quad \theta : \textit{link}(G, G') \tag{6.22}$$

where θ represents the global average probability of linkage between any pair of genes.

The second model considers all possible function classifications in determining

[1] KDD Cup 2001 (http://www.cs.wisc.edu/ dpage/kddcup2001).

[2] The functions have hierarchical structure, but for the purposes of this work we flatten the hierarchy to assume independent functions.

linkage:

$$\begin{aligned}
\forall G \forall G' \quad & \theta_{1,1} : \mathit{link}(G, G') \leftarrow f(G) = f_1 \wedge f(G') = f_1. \\
\forall G \forall G' \quad & \theta_{1,2} : \mathit{link}(G, G') \leftarrow f(G) = f_1 \wedge f(G') = f_2. \\
& \vdots \\
\forall G \forall G' \quad & \theta_{m \times m} : \mathit{link}(G, G') \leftarrow f(G) = f_m \wedge f(G') = f_m.
\end{aligned}$$

where m is the number of gene functions. Model parameters $\theta_{i,j}, i, j \in \{1, \ldots, m\}$ represent linkage proportion amongst pairs genes that have function classifications f_i and f_j respectively. For this experiment, there are $m = 144$ gene functions.

The third model is a partition model generated by using Alg. 3:

$$\begin{aligned}
\forall G \forall G' \quad & \bar{\theta}_{1,1} : \mathit{link}(G, G') \leftarrow \wedge \mathit{member}(f(G), L_1) \wedge \mathit{member}(f(G'), L'_1). \\
\forall G \forall G' \quad & \bar{\theta}_{1,2} : \mathit{link}(G, G') \leftarrow \wedge \mathit{member}(f(G), L_1) \wedge \mathit{member}(f(G'), L'_2). \\
& \vdots \\
\forall G \forall G' \quad & \bar{\theta}_{k \times k} : \mathit{link}(G, G') \leftarrow \wedge \mathit{member}(f(G), L_{k_1}) \wedge \mathit{member}(f(G'), L'_{k_2}).
\end{aligned} \tag{6.23}$$

where k_1 and k_2 are the number of partitions of gene functions. We use Alg. 3 to separately learn k_1 partitions of function classes for G, and k_2 partitions for G'. The clusters are denoted by $L_1, \ldots, L_{k_1}$ and $L'_1, \ldots, L'_{k_2}$ respectively. Each parameter $\bar{\theta}_{i,j}$ represent the proportion of linkage amongst gene pairs (G, G') such that $\mathit{member}(f(G), L_i)$ and $\mathit{member}(f(G), L'_j)$ hold.

We generate the partitions independently for values of $f(G)$ and $f(G')$, thus making the assumption that clusterings for $f(G)$ and $f(G')$ are independent. To generate a partitioning of values of $f(G)$, we aggregate over values of $f(G')$, and to generate a clustering for $f(G')$ we aggregate over $f(G)$. Figure 6.4 illustrates this process (Note that by partitioning the function values for G and G' we are partitioning in two dimensions. Algorithm 3 is optimal for the one-dimension case. Using Alg. 3 in this way – by partitioning each dimension independently – cannot guarantee an optimal partitioning.)

Learning our partition model is done with Alg. 3, where k_1 and k_2 range from 1 to 20. We also directly compute the simple model (Eq. 6.22) and the verbose

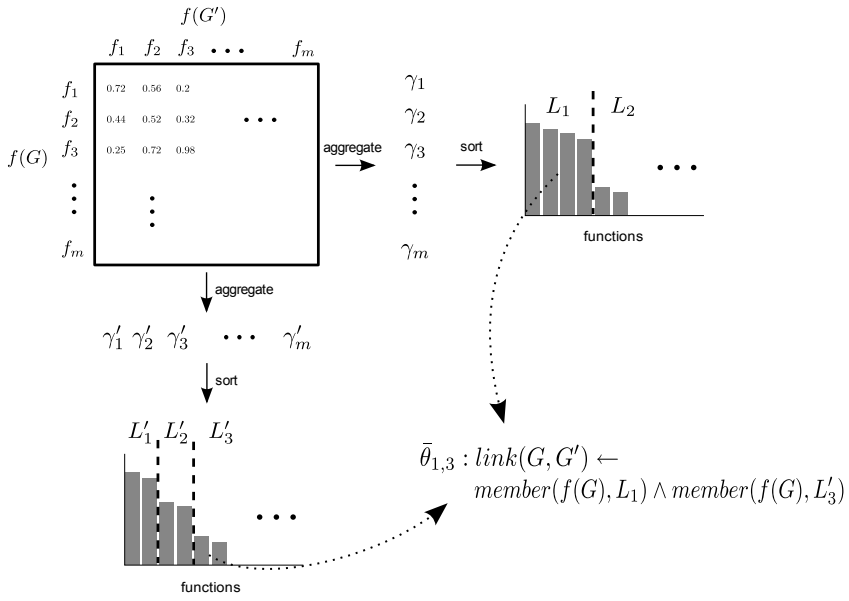


Figure 6.4: An illustration of the independent application of Alg. 3 on the functions of both genes G and G' in the relation *link*(G, G'). An example rule resulting from the partitioning is shown.

model (Eq. 6.4.2) by counting links.

We perform 10-fold cross-validation: 90% of cases for $link(G, G')$ are used to compute our models, and 10% used for prediction. We measure the log-loss in prediction for the simple (Eq. 6.22), verbose (Eq. 6.4.2), and partition models (Eq. 6.23) averaged all folds. The results are shown in Table 6.2 below, which firstly shows that the prediction problem is generally *easy*, in the sense that the number of links between genes is sparse. However, the figures show that partitioning still allows one to achieve better predictions, evident by the verbose model achieving statistically significant improvements. It is also shown that with $k_1 = k_2 = 19$, the partition model achieves competitive scores to the verbose model.

model	$\mathbf{L}_{test}$
simple	0.0437 ± 0.000374
verbose	0.0338 ± 0.000314
partition (19,19)	0.0337 ± 0.000316

Table 6.2: Log-losses and standard error for the simple (Eq. 6.22), verbose (Eq. 6.4.2), and partition models (Eq. 6.23) in predicting gene linkage based on respective gene functions. The numbers adjacent to the partition model indicate the number of partitions in each dimension (i.e. the best partition model found 19 partitions in each dimension).

We also display the relative log-loss between models[3]. The relative log-loss between our partition models (Eq. 6.23) and the verbose model (Eq. 6.4.2) is shown in Fig. 6.5. The relative log-loss between the simple model (Eq. 6.22) and the verbose model is equivalent to the relative log-loss between the partition model where $k_1 = k_2 = 1$ and the verbose model.

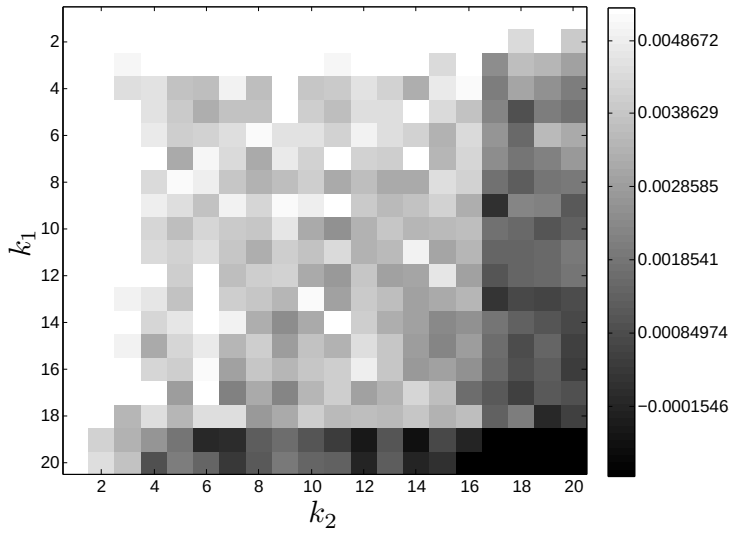


Figure 6.5: Relative log-loss of predictions over all functional classes for the gene classification dataset.

[3]Relative log-loss is simply the difference in log-loss between two models.

Figure 6.5 shows that log-loss approaches the performance of the verbose model as more partitions are introduced. In the region where $k_1, k_2 \sim 20$ shows negative relative log-loss, where the partition model achieves lower log-loss than the verbose model.

6.5 Summary

The presence of functional relations induce verbose rule sets whose size is governed by the number of values of the function. The standard approach of aggregation (Jaeger, 1997; Kersting and De Raedt, 2000; Natarajan et al., 2005) fully compacts this rule set into a single rule, but is less accurate than the verbose model. We proposed an algorithm that optimally creates a partitioning values of the function to yield a rule set whose size can be fixed *a priori*. The learned models are shown to achieve good predictive accuracy compared to the aggregated model, and can outperform the verbose model with much fewer parameters.

The theorem and algorithm provided in this chapter find an optimal partitioning of the range of a function. A limitation to the proposed method is that optimality is no longer guaranteed when functions in the model map to a multidimensional object domain. For example, the rule

$$\begin{aligned} \textit{drove\_alone}(Person) \leftarrow \quad & \textit{lives\_in}(Person) = \textit{new\_york\_city} \wedge \\ & \textit{occupation}(Person) = \textit{carpenter} \end{aligned}$$

maps people to the joint domain of cities and jobs. There may be meaningful structure between cities and jobs that prevent one from achieving a monotonically decreasing map of probabilities required to ensure optimal clustering. Finding good partitionings in such scenarios is considered in on-going work.

Chapter 7

Conclusions

7.1 Contributions

A Unified Framework for Relational Learning

This thesis presents a general framework that allows a unified approach to representing and learning dependency-based models that include observed relations as well as latent properties of individuals. Past works have either modelled latent properties of individuals to explain the observed relations, or dependency structures amongst observed relations, but not both (see Ch. 1 and 2 for a discussion of these works).

In order to achieve our goal, we require a language that is sufficiently expressive; one that can express rich dependency structure over latent and observed relations. We describe the LRM in Ch. 3, which is a lifted relational probabilistic model that represents a directed acyclic dependency network over an arbitrary set of relations[1], which can directly include latent properties. Algorithms provided in Ch. 4 to 6 handle learning of the LRM's dependency parameters and values of latent properties.

By unifying the representation and learning of dependencies and latent properties, we can consider models that go beyond state-of-the-art models such as the

[1]The formalism essentially can be regarded as a simplified version of FOPLs such as ICL (Poole, 1997, 2000).

IRM (Kemp et al., 2006). The IRM, and similar models such as the IHRM (Xu et al., 2006), generally assume a hierarchical structure, where relations are assumed to be explained by latent properties alone. Dependencies amongst observed relations are not considered, let alone modelling of rich dependency structures.

An Argument for Latent Properties

An key motivation for this work grew out of the *reference class problem* (Reichenbach, 1949), which poses the question of which sample statistic one should bring to bear in answering queries about an individual. In Ch. 3 we show that, in the simplest setting, dependency-based learning with only observed relations yield reference classes and therefore suffers from philosophical and technical difficulties that accompany reference classes.

We further examine the predictive accuracy of reference class-based methods, and present a theoretical argument that they lead to residual errors in prediction. Specifically, we assumed that values of relations are directly attributed to latent properties of individuals in the true generative process of data, and showed that reference class estimates only capture marginals of the process and results in residual *bias* in the statistical sense. By introducing latent properties of individuals, unbiased predictions can be represented. Empirical analysis using real-world data shows that models containing latent properties generally dominate reference classes in accuracy, measured in log-loss – an empirical measure of bias of an estimator.

Learning LRMs

In Ch. 4 we provide our main algorithm for learning LRMs, which comprises of learning parameters governing probabilistic dependencies as well as values of latent properties.

A standard solution to the learning problem is EM, which is particularly suitable for this task because it tightly integrate the problems of learning dependency parameters with that of learning latent properties. However, posterior inference in the E step of EM is computationally prohibitive due to the large number densely correlated latent random variables in LRMs. We therefore require an approximate

solution, and proposed a simple approximation of EM that avoids the computational bottleneck.

Empirical analyses show that the proposed approximate EM algorithm is effective in learning LRMs with good predictive performance. The results reaffirm that modelling latent properties lead to better predictive performance than those which model only observed features. Furthermore, the ability of the algorithm to achieve accurate LRMs is tested against the IRM (Kemp et al., 2006) with positive outcomes.

A key result is that modelling dependencies as well as latent properties can lead to significantly gains in accuracy. For instance, when two observed relations are mutually informative – quantified by measuring their *mutual information* – representing and learning a dependency between them can lead to significant gains in both on training and test data. Additionally learning latent properties can achieve further improvements, and does not degrade accuracy. When the relations are not mutually informative, there is little difference between learning or omitting the dependency.

Dependency-based theories are not only interpretable, the addition of latent properties yield gains in predictive accuracy. Our results further indicate that modelling latent properties alone as explanations of observed relations can benefit from modelling dependencies amongst observed relations. The findings in general suggest that learning dependency-based theories with latent relations is favourable from the standpoint of predictive accuracy, in addition to ease of interpretation.

Learning with Uncertainty about Size of Latent Properties

A hallmark of the IRM (Kemp et al., 2006) and IHRM (Xu et al., 2006) is that they are non-parametric representations, where the number of values of latent properties have no *a priori* upper-bound. Our approximate EM algorithm assumes *a priori* fixed number of values for latent properties.

In Ch. 5, we relax this assumption and place a distribution over the number of values latent properties can represent. We proposed a search-based method, where the global loop searches over the number of values of latent properties in a LRM, and the inner-loop calls the approximate EM procedure of Ch. 4.

Existing proposals use non-parametric distributions such as the Dirichlet process (DP) or Chinese restaurant process (CRP) as stochastic generators of latent properties. Learning relies on posterior sampling using MCMC to simulate the generative process (Kemp et al., 2006; Sutskever et al., 2010; **?**).

Our approach is an alternative to the sampling-based approach, with some distinguishing features. Firstly, we allow arbitrary priors can be placed over possible numbers of values for latent properties, where DPs and CRPs represent a specific class of prior. Secondly, our search process is accompanied by error bounds on the likelihood. Where the exact likelihood can be computed with an infinite number of models (one per configuration of the number of values for latent properties) our bounds reflect the error incurred from marginalising over only models visited during the search process. As error bounds accompany each step of the search process, the proposed algorithm also has an *any time* property. Thirdly, bounding the likelihood leads to bounds on the posterior over number of values of latent properties, and subsequently bounds for Bayesian averaging based prediction using models visited in the search process.

Empirical results show that allowing a flexible choice of priors, i.e. the Poisson distribution in our experiments, can lead to better weighting of models corresponding to the best number of latent property values. Given an appropriate choice of priors, averaged predictions can achieve predictive performance close to the single best model for the given domain.

Optimal Partitioning for Functions

An interesting case arises when functional relations are used. As explained in Ch. 6, the presence of a functional relation in the body of any dependency creates a verbose model whose number of parameters is proportional to the number of values of the function. The standard solution is to create a single summary dependency by aggregation (Jaeger, 1997; Kersting and De Raedt, 2000; Natarajan et al., 2005). However, the summary model generally leads to poor predictions, despite its compact size. The desired goal is then to find a model of some fixed intermediate size which also predicts accurately.

In Ch. 6 we achieve this goal by solving the problem of finding good partitions

of the verbose parameter set. We seek a fixed number of partitions where the membership of each partition is optimised for minimising empirical loss on the training data. We show that the optimisation problem can be solved optimally and efficiently via dynamic programming.

The representation size of the partitioned model no longer depend on domain size, and exhibits predictions that dominate the fully aggregated model, and can surpass that of the original verbose model.

7.2 Extensions

There are several directions for further work that extends this thesis. Some of these extensions are listed below.

Structure Learning

Chapters 4 and 5 of this thesis focused on learning parameters of LRMs, where the dependency structures of LRMs are assumed to be part of the input. One can extend the learning framework to also incorporate search over the space of structures, in much the same way as done for BLPs (Kersting and De Raedt, 2002) or PRMs (Friedman et al., 1999; Getoor et al., 2002).

The simplest way is to embed our approximate EM algorithm from Ch. 4 as a subroutine of a global search procedure over the space of dependency structures. More sophisticated forms of structure learning (for Bayesian networks) may be considered, e.g. extending the method of posterior sampling of Bayesian network structures proposed by Friedman and Koller (2003) to learn structure of LRMs.

Since we are also interested in handling uncertainty about the number values for latent properties, combining our search method from Ch. 5 with a structure search scheme is an interesting future direction for research.

A Convergent Message Passing Algorithm

The E step of the approximate EM algorithm proposed in Ch. 4 computes the marginal posterior distribution of each latent random variable in a ground LRM. Repeating the E step alone (without the M step) is observed to be a convergent process resulting in steady-state marginal posterior distributions. A suitable direc-

tion for on-going research is to formally characterise the convergence behaviour of the iteration. Establishing convergence would provide grounds for using the E step iteration as a viable approximate inference scheme in large graphical models in its own right. It can serve as an alternative to belief propagation (BP) (Pearl, 1988) – which is not guaranteed to converge – and contribute to the library of convergent approximate inference methods.

Likelihood Approximations

Another facet of our approximate EM algorithm that warrants further investigation is its approximation of the likelihood function. Currently, a pseudo-likelihood is used to evaluate the quality of parameters learned. The pseudo-likelihood is a sum of likelihoods computed from fragments of the ground LRM, and latent variables appearing in overlaps of network fragments contribute to an under-estimation of the likelihood. To enable better estimates of model parameters, it is desirable to improve one's approximation of the log-likelihood.

One way forward is to employ the region-based energy approximations devised by (Yedidia et al., 2004). The result is a free energy lower-bound of the likelihood that is a sum of contributions from *regions* of the ground network (more precisely the factor graph of the ground network), with over-counting explicitly accounted for. A closer approximation to the log-likelihood is therefore achieved.

The region-based approximation is appropriate for our work since we have implicitly defined regions in the E step of our approximate EM algorithm, and it remains to perform the appropriate subtractions. However, eliminating over-counting terms leads to a non-convex energy function. The loss of convexity can adversely affect the monotonic convergence property of EM scheme, as the M step is no longer guaranteed to optimise the likelihood bound computed in the E step. For this reason, we did not include region-based energy functions in our evaluations.

A Tool for Analysing Relational Domains

This thesis has the paradigm of learning rich dependency-based relational probabilistic models that also models latent properties of individuals. The proposed

learning framework is, however, not limited to latent properties. In fact, arbitrary latent relations can be introduced in the model, without modification of the learning methods proposed.

An example application of latent relations is the non-random missing data problem. Rather than assume that data values are missing at random, one can explicitly model the cause of missingness. For instance, for every case of the relation *likes*$(User, Movie)$, one may model the cause of each case being observed or missing with a latent random variable. Doing so for all user-movie pairs amounts to introduce the latent relation *will_rate*(X, Y), for example.

The ability to additionally postulate and learn about latent relations goes beyond discovering clusters of individuals with latent properties. Applying such models to analysing relational domains is an interesting direction for future work.

Bibliography

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008. ISSN 1533-7928.

D. Aldous. Exchangeability and related topics. In *l'cole d't de probabilits de Saint-Flour, XIII*. Springer, 1983.

F. Bacchus, A. J. Grove, J. Y. Halpern, and D. Koller. From statistical knowledge bases to degrees of belief. *Artificial Intelligence*, 87(1-2):75–143, 1996.

J. E. Besag. Statistical analysis of non–lattice data. *Statistician*, 24:179–195,, 1975.

I. Bratko and S. Muggleton. Applications of inductive logic programming. *Communications of the ACM*, 38(11):65–70, 1995.

R. Breiger, S. Boorman, and P. Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of Mathematical Psychology*, 12:328–383, 1975.

W. L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225, 1994.

J. Chang and D. M. Blei. Hierarchical relational models for document networks. *Annals of Applied Statistics*, 4(1):124–150, 2010.

M. Craven and S. Slattery. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, 43(1/2):97–119, 2001.

L. De Raedt. *Logical and Relational Learning*. Springer, 2008.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 34:1–38, 1977.

M. desJardins, P. Rathod, and L. Getoor. Learning structured Bayesian networks: Combining abstraction hierarchies and tree-structured conditional probability tables. *Computational Intelligence*, 24(1):1–22, 2007.

T. Ferguson. Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50: 95 – 125, 2003.

N. Friedman and D. Koller. *Probabilistic Graphical Models*. MIT Press, 2009.

N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 1300–1309, 1999.

S. Geman and C.-R. Hwang. Nonparametric maximum likelihood estimation by the method of sieves. *Annals of Statistics*, 10 (2):401–414, 1982.

L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. 2007.

L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 2002.

J. Ghosh and R. Ramamoorthi. *Bayesian Nonparametrics*. Springer-Verlag, 2003.

J. Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.

M. S. Handcock, E. Raftery, Adrian, and J. M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society*, 170(2):301–354, March 2007.

J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: a tutorial (with discussion). *Statistical Science*, 14:4:382417, 1999.

J. M. Hofman and C. H. Wiggins. Bayesian approach to network modularity. *Physical Review Letters*, 100(25), 2008.

T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *Proceeding of the 16th International Joint Conference on Artificial Intelligence*, pages 688–693, 1999.

D. Hume. *An Enquiry Concerning Human Understanding.* P.F. Collier & Son., 1748.

M. Jaeger. Relational Bayesian networks. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, pages 266–273. Morgan Kaufmann, 1997.

D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Proceedings of the 10th ACM SIGKDD conference Knowledge Discovery and Data Mining*, pages 593–598, 2004.

Y. Kameya and T. Sato. Efficient EM learning with tabulation for parameterized logic programs. In *Computational Logic*, 2000.

C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

K. Kersting and L. De Raedt. Bayesian logic programs. In J. Cussens and A. Frisch, editors, *Proceedings of the 10th International Conference on Inductive Llogic Programming (Work-in-progress track)*, pages 138–155, 2000.

K. Kersting and L. De Raedt. Basic principles of learning Bayesian logic programs. Technical report, University of Freiburg, 2002.

S. Kok and P. Domingos. Learning the structure of Markov logic networks. In *Proceedings of the 22th International Conference on Machine Learning*, pages 441–448, 2005.

S. Kok and P. Domingos. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.

S. Kramer. Predicate invention: A comprehensive view. Technical Report OFAI-TR-95-32, 1995.

F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2): 498–519, 2001.

H. Kyburg. The reference class. *Philosophy of Science*, 50:374–397, 1983.

H. E. Kyburg. *Logical Foundations of Statistical Inference*. D. Reidel Publishing Co., 1974.

I. Levi. Direct inference and confirmational conditionalization. *Philosophy of Science*, 48(4):532–552, 1981.

B. Marlin and R. S. Zemel. The multiple multiplicative factor model for collaborative filtering. In *Proceedings of the 21st international conference on Machine learning*, ICML '04, pages 73–, New York, NY, USA, 2004. ACM.

A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000. www.research.whizbang.com/data.

T. McGrew. Direct inference and the problem of induction. *The Monist*, 84, 2001.

B. Milch and S. Russell. First-order probabilistic languages: Into the unknown. In *Proceedings of the 17th International Conference on Inductive Logic Programming*, 2007.

B. Milch, B. Marthi, S. Russell, D. Sontag, D. L. Ong, and A. Kolobov. BLOG: Probabilistic models with unknown objects. In *19th International Joint Conference on Artificial Intelligence*, pages 1352–1359, 2005.

S. Muggleton. Inverting implication. In *Proceedings of the Second Inductive Logic Programming Workshop*, pages 19–39, Tokyo, 1992. ICOT (Technical report TM-1182).

S. Muggleton. Predicate invention and utilisation. *Journal of Experimental and Theoretical Artificial Intelligence*, 6(1):127–130, 1994.

S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995a.

S. Muggleton. Stochastic logic programs. In L. De Raedt, editor, *Proceedings of the 5th International Workshop on Inductive Logic Programming*, page 29. Department of Computer Science, Katholieke Universiteit Leuven, 1995b.

S. Muggleton. Learning structure and parameters of stochastic logic programs. In S. Matwin and C. Sammut, editors, *Proceedings of the 12th International Conference on Inductive Logic Programming*, volume 2583 of *LNAI*, pages 198–206. SV, 2003.

S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994.

S. Muggleton, M. Bain, J. Hayes-Michie, and D. Michie. An experimental comparison of human and machine learning formalisms. In *Proceedings of the Sixth International Workshop on Machine Learning*, 1989.

S. Natarajan, P. Tadepalli, E. Altendorf, T. G. Dietterich, A. Fern, and A. Restificar. Learning first-order probabilistic models with combining rules. In *Proceedings of the 22nd International Conference on Machine learning*, pages 609–616, 2005.

R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1998.

J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *IEEE International Conference on Data Mining*, pages 322–329, 2005.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

S. Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*. Lecture notes in Artificial Intelligence. Springer-Verlag, Germany, 1997.

H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpister. Identity uncertainty and citation matching. In *Proceedings of the 15th Conference on Advances in Neural Information Processing Systems*, 2002.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Reasoning*. Morgan Kaufmann, San Mateo, 1988.

J. Pearl. *Causality*. Cambridge University Press, 2009.

D. Poole. Average-case analysis of a search algorithm for estimating prior and posterior probabilities in bayesian networks with extreme probabilities. In *Proceedings of the 13th international joint conference on Artifical intelligence - Volume 1*, pages 606–612, San Francisco, CA, USA, 1993a. Morgan Kaufmann Publishers Inc.

D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993b.

D. Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 95(1-2):7–56, 1997.

D. Poole. Abducing through negation as failure: stable models within the independent choice logic. *Journal of Logic Programming*, 44(1-3):5–35, 2000.

J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5: 239–266, 1990.

H. Reichenbach. *The Theory of Probability*. University of California Press, 1949.

J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62 (1-2):107–136, 2006.

R. J. Rummel. Dimensionality of nations project: attributes of nations and behavior of nation dyads, 1950–1965. ICPSR data file., 1999.

S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition, 2009.

R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

T. Sato and Y. Kameya. PRISM: a symbolic-statistical modeling language. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pages 1330–1335, 1997.

R. Sharma and D. Poole. Efficient inference in large discrete domains. In *Proceeding of 19th Conference on Uncertainity in Artificial Intelligence*, 2003.

Y. Shen. *Loss functions for binary classification and class probability estimation*. PhD thesis, University of Pennsylvania, 2005.

P. Singla and P. Domingos. Lifted first-order belief propagation. In *Proceedings of Association for the Advancement of Artificial Intelligence (AAAI)*, 2008.

A. Srinivasan, S. Muggleton, R. King, and M. Sternberg. Theories for mutagenicity: a study of first-order and feature based induction. *Artificial Intelligence*, 85(1,2):277–299, 1996.

I. Sutskever, R. Salakhutdinov, and J. Tenenbaum. Modelling relational data using Bayesian clustered tensor factorization. In *Advances in Neural Information Processing Systems*, 2010.

B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 870–878, 2001.

L. H. Ungar and D. P. Foster. Clustering methods for collaborative filtering. In *Americal Association for Artificial Intelligence Workshop on Recommendation Systems*. AAAI Press, 1998.

U.S. Census Bureau. Travel to work characteristics for the 50 largest cities by population in the united states: 1990 census, 1990.

J. Vickers. The problem of induction. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2009. URL http://plato.stanford.edu/archives/spr2009/entries/induction-problem/.

Z. Xu, V. Tresp, K. Yu, and H. Kriegel. Infinite hidden relational models. In *Proceedings of the 22nd Conference on Uncertainty in Aritificial Intelligence*, 2006.

Z. Xu, V. Tresp, A. Rettinger, and K. Kersting. Social network mining with nonparametric relational models. In H. Zhang, M. Smith, L. Giles, and J. Yen, editors, *Advances in Social Network Mining and Analysis*, LNCS. Springer, 2009.

J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical report, Mitsubishi Electric Research Laboratories, 2003.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, 2004.

N. L. Zhang and D. Poole. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328, 1996.