AN ABSTRACT OF THE THESIS OF

Eric E. Altendorf for the degree of Masters of Science in Computer Science
presented on May 6, 2005.
Title: Constraining Bayesian Network Learning with Qualitative Models.

Abstract approved: Redacted for privacy

Thomas G. Dietterich

Machine learning encompasses probabilistic and statistical techniques that can build models from large quantities of *extensional* information (examples) with minimal dependence on *intensional* information (domain knowledge). This focus of machine learning is reflected in the never-ending quest for "off-the-shelf" classifiers.

To generalize to unseen data, however, we *must* make use of more information than is contained in the training data. Successful learning from data, especially from sparse data, relies on effective incorporation of domain knowledge into the learning algorithm. Unfortunately, there are very few existing techniques or technologies for doing this. Feature engineering, feature selection, model structure selection, parameterization, and algorithm selection are common techniques, but all of these are difficult, time-consuming, and limited in the type of knowledge they can express.

In this paper, we show how to interpret qualitative knowledge about probabilistic influences, in particular, knowledge about monotonicity, synergy, and strength of influence, as constraints on conditional probability distributions. We then describe a method for incorporating this knowledge into a parameter estimation algorithm for Bayesian networks. We provide results demonstrating improved accuracy for monotonicity-constrained networks, especially with very small training sets, compared to unconstrained networks and other learning algorithms.

Constraining Bayesian Network Learning with Qualitative Models

by
Eric E. Altendorf

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Masters of Science

Presented May 6, 2005
Commencement June 2006

Masters of Science thesis of Eric E. Altendorf presented on May 6, 2005

APPROVED:

Redacted for privacy

Major Professor, representing Computer Science

Redacted for privacy

Associate Director of the School of Electrical Engineering and Computer Science

Redacted for privacy

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Eric E. Altendorf, Author

ACKNOWLEDGMENTS

I thank all of those at Oregon State University who have contributed ideas and guidance in this research, especially (in alphabetical order): Bruce D'Ambrosio, Yaroslav Bulatov, Thomas Dietterich, Matthieu Labbé, Sriraam Natarajan, Angelo Restificar, and Prasad Tadepalli.

Angelo Restificar spent a significant amount of time researching our data sets in order to become our "domain expert". Based on his research, he prepared the data sets used in our experiments and designed the Bayesian network structure and qualitative monotonicity relationships we used. He also contributed for this thesis text and figures describing the data sets. Thomas Dietterich contributed code used for computing McNemar's test, as well as specific suggestions for revisions of various sentences and paragraphs in this paper.

Many fellow students have also assisted with my questions about LaTeX (thanks especially to Rogan Creswick) and Gnuplot (Wade Holmes). I also thank Aron W. Culotta, Charles Sutton, and Jerod Weinman from the Mallet developers maillist for their advice on the use of Mallet's L-BFGS optimizer.

Also deserving of acknowledgment are the countless individuals who have contributed to the Free software I use daily and which was instrumental in carrying out this research and writing this thesis, especially the Linux kernel, the GNU system utilities, LaTeX, Gnuplot, and Weka.

I thank the unknown author whose aside in a preface informed me that placing sentence punctuation inside quotation marks is not a universally expected convention. I personally find it awkward and problematic, and so in this work, quotes contain only the quoted material, and the punctuation, which rightly belongs to the enclosing sentence, is placed outside.

I thank the staff and owners of Interzone, Sunnyside Up, The Beanery (north, south, and campus), and Piazza, for their generally exquisite coffee and pleasant cubicle-free office space.

I thank my mother, Jodi Altendorf, to whom I owe whatever good English writing skills I might have, my father, John Altendorf, who first taught me to write code, and my girlfriend, Celina Thornton, for her continuing patience and support.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

Figure Page

LIST OF FIGURES (Continued)

LIST OF TABLES

Table Page

X, Y, Z, X_i	Random variables, or, equivalently, nodes in a Bayesian network
x, y, z	Values of random variables
$x_0, x_1, \ldots$	(Generally) values 0, 1, ...of random variable X
x_i	(Generally) a particular value of a random variable X_i
$p_0, p_1, \ldots$	Cells in a conditional probability table (like θ_{ijk} below, but subscripted in a context-specific way)
r_i	The number of states a discrete node X_i may take
G	The graphical structure of a Bayesian network
Q	The qualitative model (i.e., annotations on G)
π_i	The set of parents of X_i
$\mathbf{Pa}_i$	The set of parent configurations of X_i
q_i	The number of parent configurations of X_i, that is, $\lvert\mathbf{Pa}_i\rvert$
$\mathbf{pa}_i^j$	The jth configuration of the parents of X_i
r_{XY}	The "strength of relation" between X and Y
s	The "strength of influence"
θ_{ijk}	The probability of value k given parent configuration $\mathbf{pa}_i^j$
θ_{ij}	The parameter vector for parent configuration $\mathbf{pa}_i^j$
θ_i	The matrix of parameters for X_i
θ	The complete collection of matrices θ_i
μ_{ijk}	A reparameterization of θ s.t. $\theta_{ijk} = \exp(\mu_{ijk})/\sum_{k'=1}^{r_i} \exp(\mu_{ijk'})$
D	The observed data
N_{ijk}	The observed counts of $X_i = k$ given $\mathbf{pa}_i^j$, plus $\alpha_{ijk} - 1$
ξ	A prior for Bayesian learning
ξ^G	The conditional independence assumptions of the structure of G
ξ^Q	The constraints implied by our qualitative model
ξ^P	The prior over parameter values (e.g., a Dirichlet distribution)
α_{ijk}	The specific hyperparameters (or psudocounts) for ξ^P
$Z^i_{jk_c}$	Shorthand for $\sum_{k=1}^{k_c} \exp(\mu_{ijk})$
$J_L(\theta_i)$	The unconstrained log-likelihood of the parameters
$C^{i,k_c}_{j_1,j_2}$	The inequality constraint corresponding to node i, between parent configurations j_1, j_2 and the local state index k_c
δ	The amount by which a constraint is violated
$P^{i,k_c}_{j_1,j_2}$	The penalty function of the corresponding constraint
w	The weight by which penalty functions are multiplied
$J(\theta_i)$	The complete penalized objective function

Notation used in this thesis

Constraining Bayesian Network Learning with Qualitative Models

Chapter 1: Introduction

We begin this thesis with some history of the field of artificial intelligence, and the context which gave rise to the study of machine learning. This leads us to discuss outstanding problems in machine learning in general, which in turn naturally leads to the context and motivation for our current research projects, including "knowledge intensive machine learning". After covering our general research goals, we will finally focus on the subject of this thesis.

1.1 History

"Traditional" artificial intelligence systems (for example, early expert systems, and, in general, many AI systems through about 1980) were largely composed of facts, rules, and inference engines, generally hand-engineered from general logical rules and expert domain knowledge. This process of knowledge engineering was exceedingly difficult and time consuming for both the domain expert and the system engineer. Furthermore, in many cases (for example, perception tasks such as computer vision, speech recognition, and so on), knowledge engineering approaches are probably completely infeasible. While some researchers have continued on the knowledge engineering path (one of the best known and most persevering being Douglas Lenat and the Cyc project [Lenat and Guha, 1989]), for the past few decades, artificial intelligence researchers have begun to explore alternatives. (See [Russell and Norvig, 1995, pp. 16-26] for a brief discussion.)

Machine learning was born out of a hope that computers could induce rules from data more quickly, more accurately, and with less time from domain experts and knowledge engineers. In many cases these hopes have been realized. Machine learning has become a hugely successful field, and arguably has long surpassed pure knowledge-engineering approaches to building many types of intelligent systems (for example, speech recognition, optical character recognition, bio-informatics, biometrics, anomaly detection, and information extraction). However, it has of course not addressed all the problems of engineering such systems. Constructing machine learning models is still time-consuming and requires significant machine learning expertise for tasks such as model structure selection, learning algorithm selection, propositionalizing real-world data, and feature construction in general. Moreover, incorporating domain knowledge that could give us hope in data-sparse problems is even more difficult.

1.2 "Knowledge Intensive Machine Learning"

Our research group's long-term goal is the development of a computer-assisted modeling environment which speeds up and partially automates the construction of machine learning models. This project encompasses many sub-tasks and goals, including:

- Modeling of higher-level, relational domains
- Automatic feature selection, model structuring
- Graphical editors for rapid model engineering / re-engineering
- Learning algorithms which make use of higher-level qualitative domain knowledge

In one sense, all model construction for machine learning involves some amount of domain knowledge. Even the use of an off-the-shelf classifier on data from an unknown source makes an assumption about the domain: that at least some of the attributes of the data are related to other attributes.[1]

[1] This is indeed a non-trivial assumption: if we believed there were no relation between the attributes of the training data, applying a learning algorithm would be inappropriate since any "learning" would necessarily be overfitting. Thus, the mere fact that we apply a learning algorithm to data implies we have a prior belief about the data.

Another way to look at this is to realize that in order to generalize, a machine learning algorithm must make use of information other than that contained in the training data—namely, domain knowledge. In theory, this is obvious (learning is not possible without a defined model space and/or a prior), but it is also empirically clear that good knowledge engineering makes a huge difference in performance. Very few successful real-world applications of machine learning make use of off-the-shelf learning algorithms without the incorporation of significant amounts of domain knowledge.

However, the incorporation of domain knowledge is a task which an engineer can carry on at many levels of sophistication. This level of sophistication, or in other words, the *effective* incorporation of *more* and *more informative* domain knowledge, is arguably the most important and most time-consuming aspect of designing a successful machine learning model.

Current use of domain knowledge is generally restricted to feature engineering, feature selection, model structure selection, parameterization, and algorithm selection. These approaches require significant knowledge and coordination from both domain experts and machine learning experts, and a significant overall amount of engineering time. This presents an opportunity: by formalizing domain knowledge in a principled way, machine learning models could make use of this knowledge for feature selection, engineering, and parameter estimation, improving performance while reducing the workload on domain experts and machine learning experts.

Developing algorithms and tools which support the incorporation of domain knowledge into machine learning systems is what we mean by the term "knowledge intensive machine learning".

1.3 Knowledge-Constrained Learning

The first task we consider in our research program is a system to make automatic use of domain knowledge for constraining the model space. Constraining the model space can be viewed as a type of regularization, or simply as an *a-priori* model selection step.

Bayesian networks form a natural and well-known example of the application of constraints to the model space, since model builders use domain knowledge to specify the causal structure or conditional independencies of a domain in order to reduce the number of parameters which must be learned from data. It is, however, difficult to find other examples of this concept, and developing such examples is the goal behind our research in knowledge-constrained learning.

The specific domain knowledge which we will address in this thesis involves influence and relation (the conditional independencies of Bayesian networks), strength of influence and relation, monotonicity constraints, and synergistic interactions. We build upon Bayesian networks because they already allow us to easily express domain knowledge of these conditional independencies.

After providing an overview of prior work, we will begin by describing the qualitative concepts which we (or the domain expert) would like to be able to express. Although in this thesis we do not formulate a language *per se* for expressing these concepts,[2] we formalize them with precise definitions in terms of what priors or constraints the various qualitative statements impose on a statistical model, in particular, Bayesian networks. We will also discuss how these constraints can be implemented in a feasible way, to do better parameter estimation from less data. Finally, we provide experimental results showing the value of incorporating even simple domain knowledge into a machine learning algorithm, and discuss outstanding issues and avenues for further research. Following the conclusion, Chapter 8 delves a bit deeper into some related statistics literature, which is postponed to the end to avoid distracting the reader with lengthy tangential discussions.

[2] We are working on such a language, including both qualitative relations and relational probabilistic modeling concepts. This language will be specified in a separate document, to be published later as a technical report.

Chapter 2:
Prior Work

This thesis lies at a juncture between symbolic AI, qualitative reasoning, statistics, and machine learning. As such, there is much relevant prior work in many diverse areas. There is also some limited work specifically on the application of monotonicity constraints to machine learning algorithms. In this chapter we briefly review the existing relevant literature.[1]

2.1 Qualitative Reasoning

Some of our terminology comes from the extensive prior work in qualitative reasoning (perhaps most popular in the 1980's; see for example [Bobrow, 1985,Kuipers, 1994]). Qualitative reasoning (or the closely associated field of qualitative physics) is a field which addresses methods for modeling systems qualitatively, in terms of what things influence what other things, in what ways, and under what conditions. Frequently these are physical systems, such as a bathtub, a pressure regulator, or a chemical processing plant. The goal in most of this work has been qualitative simulation—the ability to answer questions such as: "what happens to the level of the water in the bathtub if I pull the plug?" or "how will the chemical plant operate if I increase the temperature of holding vat X?". Such a system could deduce from a physical description of a pressure regulator that the pressure output is in fact more or less limited, but also that it is susceptible to oscillations. For the most part, however, qualitative reasoning deals with reasoning, not learning.

[1] In addition, some of the key ideas and experimental results presented here also appear in [Altendorf et al., 2005].

2.2 Knowledge-Based Model Construction

Another relevant field, popular in the early 1990's, is knowledge-based model construction (KBMC). KBMC involves using domain knowledge to construct a model "on-the-fly" to answer a specific question, such as a query on the truth or probability of a proposition or formula, or a decision query [Wellman et al., 1992, pg. 26]. The model construction phase determines relevant and irrelevant variables, and constraints that hold on relationships between variables. The models are usually some form of a graphical model such as a qualitative probabilistic network [Wellman et al., 1992], a dynamic belief network [Glesner and Koller, 1995,Ngo and Haddawy, 1996], or a neural network [Towell and Shavlik, 1994].

Knowledge-based model construction of graphical models, however, generally assumes either knowledge-engineered probabilities, or qualitative non-Bayesian-network relationships with no parameters to learn, rather than fully parameterized graphical models which can be learned from data [Wellman et al., 1992, pg. 31]. Others forms of knowledge-based model construction, such as [Towell and Shavlik, 1994], have limitations on the knowledge forms (e.g., only propositional logic on variables).

2.3 Extensions of Logic Programming

Inductive logic programming (ILP, see for instance [Bergadano and Gunetti, 1996]) combines domain knowledge with data, performing learning from data given knowledge. The "data", however, generally takes the form of knowledge or facts expressed in a first-order logic language, which are incorporated with the prior knowledge. In addition, there is generally no probabilistic component, so modeling uncertainty is difficult. In general, ILP is less commonly used for the types of classification tasks we address in this paper, though this is not to say that they could not be used or that there does not exist an analog of this work (e.g., monotonicities), applicable to ILP systems.

Bayesian Logic Programs and Stochastic Logic Programs (see for instance [De Raedt and Kersting, 2003]) extend first-order logic programming with probabilities and allow probabilistic inferences to be calculated. In addition, their probabilities can be learned from data. The fact that Bayesian and Stochastic Logic Programs can model higher-level relational domains in a more human-understandable way makes them appealing for knowledge-intensive machine learning, but in a way orthogonal to the goals of this particular thesis.

2.4 Machine Learning

Clark and Matwin [Clark and Matwin, 1993] share our motivations for knowledge-constrained learning, but perform learning only of parameters of a qualitative process model (that is, for temporal simulation).

Several researchers have incorporated prior knowledge into artificial neural networks (ANN's). Knowledge-based ANN's [Towell and Shavlik, 1994] have structures and parameters initialized from knowledge bases of Horn clauses. ANN's also lend themselves to connection weight constraints that enforce monotonicity; such models are developed and applied in [Archer and Wang, 1993, Daniels and Kamp, 1999]. Abu-Mostafa, in [Abu-Mostafa, 1995], discusses *hints*, or "auxiliary information" not contained in the training data (i.e., domain knowledge), including statements of invariance, monotonicity, and mock training examples. One notable contribution he provides is a discussion of VC-dimensions for such "hinted" models. Kay and Ungar also discuss monotonic neural networks with confidence intervals in [Kay and Ungar, 1993].

In [Daniels et al., 2002], Daniels, Feelders, and Velikova improve on prior work in monotonic regression techniques by employing both constrained neural networks and decision trees. Monotonicity in trees in further discussed in [Potharst and Feelders, 2002, Feelders, 2000, Ben-David, 1995]. Monotonicity in trees is important because the CART and C4.5 learning algorithms tend to build non-monotonic trees even given monotonic data [Potharst and Feelders, 2002]. Interestingly, some of the decision tree work (e.g., [Feelders, 2000]) reports poorer classification performance under monotonicity constraints, the advantage being found in simpler, more understandable decision trees. Much of this work was also motivated by the need to make justifiable decisions (e.g., in school admissions or job or loan applications, see for instance [Ben-David, 1995]).

Fung, Mangasarian, and Shavlik [Fung et al., 2001] have developed knowledge-constrained support vector machines, although the "prior knowledge" is not qualitative, but rather mock training data which must be expressed as polyhedral sets in the input space.

2.5 Statistics

The literature on machine learning with monotonicities appears to be restricted to classifiers and regression models, despite the fact that the statistics and financial literature is rich with discussions of stochastic ordering of probability distributions. A seminal work is [Lehmann, 1955]; a comprehensive recent treatment can be found in [Szekli, 1995]. The specific problem of estimating a pair of distributions $P_1(X)$ and $P_2(X)$ under a first order stochastic dominance constraint (see Definition 3.1) is discussed in [Dykstra, 1982], though we found no work on estimating partially ordered sets of distributions. Along different lines, Hellerstein discusses nonparametric estimation of confidence intervals for prediction of monotonic functions in [Hellerstein, 1990].

The statistical literature covers a variety of formulations of monotonicity for random variables. However, this work generally approaches the problem from a perspective of data analysis, not parameter estimation. To our knowledge, there is only one paper [Agresti and Chuang, 1985] which discusses learning parameters under monotonicity priors, though [Dykstra, 1982] is also relevant. The former paper, by Agresti and Chuang, uses a stricter form of monotonicity defined on joint distributions by the local odds-ratios (with both a hard sign constraint and a Gaussian prior) and presents techniques for both constrained MLE and MAP inference under such priors [Agresti and Chuang, 1985]. This work is discussed in greater detail in Chapter 8.

Chapter 3:
Qualitative Model Concepts and Semantics

In this chapter we develop the "language" of qualitative domain statements. The language is described only at a conceptual, abstract level; we omit discussion of the grammar and syntax as it is tangential and mostly irrelevant for this work (as mentioned, one possible language specification will be published separately as a technical report). Moreover, different grammars or incarnations of the concepts described herein may be more or less appropriate for integration with different learning systems or in different contexts.

We begin with a very brief overview of the types of qualitative statements we support and the reasons for supporting them. We then discuss how to describe the qualities of the data over which the model is built. Next, we move to the main topic of this chapter and discuss in turn the most general form of qualitative statement (the influence), and specializations and annotations thereof for monotonicities, synergies, and strength of relation.

For readers interested primarily in the experimental results (presented in Chapters 5 and 6), it suffices to read up to Section 3.4, since the remainder of the chapter deals with other qualitative concepts which we do not experimentally evaluate in our current work.

3.1 Overview

Given some set of random variables, we, as domain experts, wish to encode some qualities of their probabilistic relation to each other. This could include the existence of a probabilistic dependence, the directionality of the relation, or a set of constraints on the conditional distribution of one given the other. Such statements constrain the model space or parameter space in some way.

We have several different types of qualitative statements which accomplish this in different ways. Basic influences and relations[1] set up the conditional independence structure (the same way defining arcs in a Bayesian network do). Monotonic relations constrain the form of the probability distributions. Strength of monotonicity statements adjust the strength of such constraints. Synergies constrain forms for monotonic CPDs with multiple parents. Strength of influence statements constrain the mutual information of variables, or in other words, how peaked or diffuse the CPD is for each value of the conditioning variables. These constraints in theory can be applied to either conditional probability tables (in tabular or tree-based form) or to continuous functional forms for the probability density. For simplicity, we consider in this work only discrete variables and probability tables.

In this section we discuss such qualitative statements abstractly (referring to the related variables only as X or Y). These variables (or *features*) in the propositional case (that is, the standard machine learning situation) correspond simply to attributes in the training data. In a relational model, they could be specified in some form of a path expression or first order logic conditional expression in respect to a defined relational schema.

3.2 Schema and Attribute Definitions

In propositional modeling, there is not much to be said about the schema to which the data conforms. One important point, however, is the measurement type of random variables. The measurement type determines the general form of the CPD (discrete or continuous), the kinds of constraints that may be applied, the semantics of measurement, and so on. We consider the following types (for more detail, see [Krantz et al., 1971]):

[1] Sometimes we will use the term *relation* rather than *influence*; this reflects the fact that many of these ideas are generalizable to undirected relations, as in a Markov network, though we do not provide semantics for such relations here. In Chapter 8, we discuss concepts from statistical literature including notions of monotonicity for joint distributions, which may be of use for extending this work to Markov networks.

Nominal: Unordered, discrete set of values. E.g.: {female, male}, or {blue, green, red}.

Ordinal: Ordered set of values, where differences between values may not be comparable (not "constant scale"). E.g.: {low,med,high}.

Interval: Ordered set of values, with constant scale, but with no meaningful zero point (implies that ratios are undefined). E.g.: date, time, or temperature (measured on a Fahrenheit or Celsius scale).[2]

Ratio: Ordered (possibly dense/continuous) set of values with constant scale and defined zero point. E.g.: any real-valued physical measurement: mass, length, etc.

The primary distinction we will make is between nominal-valued variables and non-nominal variables, that is, variables whose values can be ordered. This is because our most useful qualitative constraint will be one of monotonic effect of one ordered variable on another. However, we do care about distinctions between the three ordered types for certain cases of interpreting synergistic and strength of monotonicity constraints.

3.3 Influences and Relations

A directed qualitative relation, or a qualitative influence, is a statement of the form $X \overset{Qinf}{\succ} Y$ about a probabilistic relation between two random variables, called the *influent* (loosely, the cause, in this case X), and the *resultant* (loosely, the effect, in this case Y). Their probabilistic semantics correspond to arcs in a Bayesian network, but they were originally inspired by similar ideas ("determinations") in logical systems, due to [Russell, 1989] (also see [Davies, 1988, Mahadevan and Tadepalli,

[2] Matthieu Labbé points out (private communication) that the choice between interval and ratio is somewhat subjective, since it is based on the "meaningfulness" of the zero-point, which depends on many things, including the context and interpretation of the measurement.

1994]). We were also inspired by work in qualitative modeling due to [Wellman, 1990].

The most basic and general form of a directed qualitative influence is $X \overset{Qinf}{\succ} Y$, or X influences Y. This is applicable when we know two things are related but have no prior knowledge of how their joint probability distribution should be constrained.

We will introduce constructs for specifying higher-order interactions (synergies and relative orderings of strength of monotonicity or relation), but in the absence of explicit statements to the contrary, we define the language with a *ceteris paribus* assumption and require that each statement operates independently of other statements, that is, that each statement holds for "all other things being equal".

3.4 Monotonicity

The first specialized type of influence we will address is monotonic influence. A monotonic influence is denoted $X \overset{Q+}{\succ} Y$ (or $X \overset{Q-}{\succ} Y$). Intuitively, the idea is that stochastically, higher values of X result in higher (or lower, for $\overset{Q-}{\succ}$) values of Y. For example, we might expect a higher risk for diabetes in persons with a higher body mass index.

In the qualitative reasoning literature (see for instance [Kuipers, 1994]), there is a difference between $\overset{Q+}{\succ}$ (which has a positive effect which however is subject to overriding negative effects from other variables) and $\overset{M+}{\succ}$ (which has an unconditional positive effect). The latter is not *ceteris paribus*, and does not seem to be very useful in the real-world modeling applications we have considered. We do not discuss it further. In addition, different authors have used different notations; for instance, in [Wellman, 1990] the preferred symbol is $S+$, and in [Forbus, 1985] we find $\propto_{Q+}$.

Monotonic relations may further be annotated by strength of monotonicity statements, which may be absolute and expressed in terms of odds ratios, or relative to other influents on the same resultant (e.g., "Y and Z both monotonically influence X, but the monotonicity of Y is stronger"). This will be discussed in Section 3.7. It is conceivable that we would also wish to constrain for "weakness of monotonicity" for weak monotonic effects; however it seems that such a constraint would have very little effect on parameter estimation and so we do not consider it further.

3.4.1 Semantics for Conditional Probabilities

Our basic question is: how does the statement $X \overset{Q+}{\succ} Y$ constrain a probability distribution $P(Y \mid X)$? Intuitively, we want increasing values of X to shift the probability mass on Y upwards. This idea can be seen in Figure 3.1.

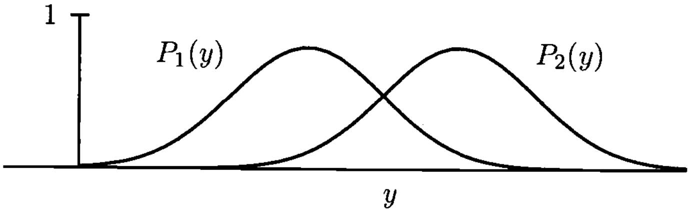


FIGURE 3.1: Example of "shifting a probability mass upwards", comparing P_2 to P_1

To formalize this notion, we consider cumulative distribution functions instead of the probability density functions. This gives us the concept of *first-order stochastic dominance* (FSD), whose definition we now provide.

Definition 3.1 (First Order Stochastic Dominance) *Given two probability distributions P_1 and P_2, and their respective cumulative distribution functions F_1 and F_2, then $P_2 \succeq_{(1)} P_1$ if and only if for all y we have $F_2(y) \leq F_1(y)$.*

This concept dates back at least to [Lehmann, 1955]; for a recent and readable coverage in a context similar to that of this thesis, see [Wellman, 1990, pg. 10]. Note that we are aware of no standard or accepted way to read the first order stochastic dominance symbol. We suggest "P_2 stochastically dominates P_1 to the first order" or "P_2 dominates P_1 in the first order stochastic sense".

The intuition of this definition is presented visually in Figure 3.2. The definition applies to both continuous distributions (as pictured in the figures) as well as discrete (which are of course the type we consider in this thesis).

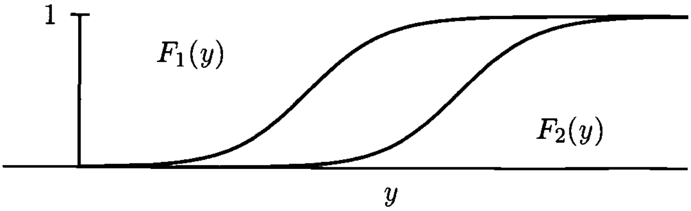


FIGURE 3.2: First-order stochastic dominance, visualized with cumulative distribution functions

For an example over discrete conditional probability distributions, consider a random variable X with boolean ordinal domain D. If we state that $P(Y \mid X = t) \succeq_{(1)} P(Y \mid X = f)$, then we know that for all $y \in D$, we have

$$P(Y \leq y \mid X = t) \quad \leq \quad P(Y \leq y \mid X = f). \tag{3.1}$$

From this, we take our basic interpretation for monotonicity.

Definition 3.2 (FSD Monotonicity) *For X, Y ordinal, we say Y is FSD isotonic (antitonic) in X in a context C if for all x_1, x_2 such that $x_1 \geq x_2$ (respectively, $x_1 \leq x_2$), we have*

$$P(Y \mid X = x_1, C) \succeq_{(1)} P(Y \mid X = x_2, C). \tag{3.2}$$

There are alternative definitions for probabilistic monotonicity. One example is *modal monotonicity* [van der Gaag et al., 2004]. The definition is the same as above except we replace $P(Y \mid X = x_1) \succeq_{(1)} P(Y \mid X = x_2)$ with $mode(Y \mid X = x_1) \geq mode(Y \mid X = x_2)$, where where $mode(Y \mid X)$ refers to the most likely value of Y given X. These two constraints are not equivalent, nor does one subsume the other, and they have different applications. Modal monotonicity in $P(Y \mid X)$ is useful when we are interested in the most likely value of Y, and FSD monotonicity is useful when we need to use $P(Y)$ in further calculations, such as Bayesian network inference. As we have already mentioned, two relevant resources for stochastic dominance are [Lehmann, 1955, Szekli, 1995]. See also the discussion in Chapter 8.

Although it is shown in [van der Gaag et al., 2004] that the problem of determining whether or not two variables X and Y in an arbitrary Bayesian network satisfy either the modal monotonicity or the FSD monotonicity constraints is coNP$^{\mathrm{PP}}$-complete, the same question given a (small) conditional probability table is clearly straightforward. We now provide a definition to formalize our $\overset{Q+}{\succ}$ and $\overset{Q-}{\succ}$ statements, including their *ceteris paribus* nature.

Definition 3.3 ($\overset{Q+}{\succ}$, $\overset{Q-}{\succ}$ **Statements)** *Suppose that* Y *has multiple parents* $\pi_Y = X_1, X_2 \ldots X_q$. *The statement* $X_i \overset{Q+}{\succ} Y$ *means for all contexts (configurations of other parents)* $C \in \times_{j \neq i} X_j$, *that* Y *is FSD isotonic in* X_i *in context* C. *Likewise,* $\overset{Q-}{\succ}$ *denotes antitonicity.*

3.4.2 Examples

We provide two examples of monotonic CPT constraints. The first example is the simplest possible monotonic relation: $X \overset{Q+}{\succ} Y$ for X, Y binary. The CPT and single constraint can be found in Table 3.1.

	$P(Y \mid X)$	
x	$y = 0$	$y = 1$
0	p_0	$1 - p_0$
1	p_1	$1 - p_1$

Constraints
$p_0 \geq p_1$

TABLE 3.1: Example of CPT for a binary variable Y given X (also binary) with qualitative statement $X \overset{Q+}{\succ} Y$.

The second example is more complex and involves two parents: $X \overset{Q+}{\succ} Z$, and $Y \overset{Q-}{\succ} Z$, with Y, Z ternary and X binary. The CPT can be found in Table 3.2 and the constraints in Table 3.3.

		$P(Z \mid X,Y)$		
x	y	$z=0$	$z=1$	$z=2$
0	0	p_{000}	p_{001}	$(1-p_{000}-p_{001})$
0	1	p_{010}	p_{011}	$(1-p_{010}-p_{011})$
0	2	p_{020}	p_{021}	$(1-p_{020}-p_{021})$
1	0	p_{100}	p_{101}	$(1-p_{100}-p_{101})$
1	1	p_{110}	p_{111}	$(1-p_{110}-p_{111})$
1	2	p_{120}	p_{121}	$(1-p_{120}-p_{121})$

TABLE 3.2: Example of CPT for a ternary variable Z given X (binary) and Y (ternary). The three subscripts on p refer to the value of x, y and z, respectively.

$p_{000} \geq p_{100}$	$p_{000}+p_{001} \geq p_{100}+p_{101}$
$p_{010} \geq p_{110}$	$p_{010}+p_{011} \geq p_{110}+p_{111}$
$p_{020} \geq p_{120}$	$p_{020}+p_{021} \geq p_{120}+p_{121}$
$p_{000} \leq p_{010}$	$p_{000}+p_{001} \leq p_{010}+p_{011}$
$p_{010} \leq p_{020}$	$p_{010}+p_{011} \leq p_{020}+p_{021}$
$p_{100} \leq p_{110}$	$p_{100}+p_{101} \leq p_{110}+p_{111}$
$p_{110} \leq p_{120}$	$p_{110}+p_{111} \leq p_{120}+p_{121}$

TABLE 3.3: Example of CPT constraints for $X \overset{Q+}{\succ} Z$ (above), and $Y \overset{Q-}{\succ} Z$ (below). The three subscripts on p refer to the value of x, y and z, respectively.

3.4.3 Number of Induced Constraints

The number of constraints induced by a monotonicity statement is straightforward to compute. Suppose a node Y has r states, parents π, and constrained parents $\pi^Q \subset \pi$. For each constrained parent $X_i \in \pi^Q$, having r_i states, we get $r_i - 1$ sets of constraints for each configuration of other parents $\pi \setminus X_i$, each set having $(r_j - 1)$ constraints (corresponding to parent X_j). Thus the total number of constraints is

$$(r-1)\sum_{X_i \in \pi^Q}(r_i - 1)\prod_{X_j \in \pi \setminus X_i} r_j. \tag{3.3}$$

This is exponential in the number of total parents, linear in the number of constrained parents, linear in the number of states of the child, and superlinear in the number of states of each parent. Clearly, having many parents, and especially many parents with many states, results in a very large number of constraints.

3.5 Synergistic Interactions

When more than one influent monotonically influences a resultant, it is possible to constrain pairs of influences with synergistic statements. Such statements are qualitative descriptions of the interaction between the influences.

3.5.1 Types of Synergistic Interactions

In this work, we consider three basic types of synergistic interaction:

Independent: If $X \overset{Q+}{\succ} Z$ and $Y \overset{Q+}{\succ} Z$ independently then increasing X has the same effect for high values of Y as for low values of Y (and likewise for increasing Y with fixed X). For interval-measured variables, this can be referred to as an "additive" synergy, and implies $(X + kY) \overset{Q+}{\succ} Z$ for some k.

Synergistic: If $X \overset{Q+}{\succ} Z$ and $Y \overset{Q+}{\succ} Z$ synergistically, then increasing X has a greater effect for high values of Y than low values of Y (and likewise for increasing Y with fixed X). For interval-measured variables, we call this "superadditive".

Antisynergistic: If $X \overset{Q+}{\succ} Z$ and $Y \overset{Q+}{\succ} Z$ antisynergistically, then increasing X has a lesser effect for high values of Y than low values of Y (and likewise for increasing Y with fixed X). For interval-measured variables, we call this "subadditive".

It is important to recognize that "independence" as a synergy implies a specific constraint on the combined distribution; it is not a statement of "no knowledge" about the combined effect. As discussed before, the definitions of monotonicity make no assumption of synergy.[3] Since synergy does not make sense unless both combined effects are monotonic, we should leave distributions unconstrained by synergistic relations when no synergy is specified.

3.5.2 Semantics for Qualitative Synergies

We now formalize these ideas in terms of constraints on probability distributions. As discussed earlier, we take "$X \overset{Q+}{\succ} Z$ and $Y \overset{Q+}{\succ} Z$ independently" to mean that X and Y have independent isotonic influences on Z. That is, the magnitude of effect of changing X is the same regardless of the value of Y, and vice versa. Likewise, synergies and antisynergies are defined in terms of whether the effect of changing one influent is greater or lesser depending on the value of the other influent. As discussed in [Wellman, 1990, pg. 27], these ideas can be formalized using the notion of modularity. We begin with a definition:

Definition 3.4 (Modularity) *A function $f(x,y)$ with ordinal domains and interval range is modular iff*

$$x \geq x' \wedge y \geq y' \quad \Longrightarrow \quad f(x,y) + f(x',y') = f(x,y') + f(x',y). \qquad (3.4)$$

If the equality in the consequent is changed to $\leq$, then $f()$ is submodular, *and if it is changed to $\geq$, then $f()$ is* supermodular. *Additionally, a function $f(x,y)$ is*

[3] It is not entirely obvious that this is the correct assumption. It may make sense to specify additive synergy as the default for monotonic influences of interval-measured influents. This question essentially depends on how rich an interaction the learning algorithm should expect by default. If additive were made the default for interval-measured influents, we would need an explicit "unknown" synergy annotation to remove the constraint.

supermodular if and only if $\frac{\partial^2}{\partial x \partial y} f(x,y) \geq 0$ *(see [Ross, 1983, pg. 6] for a proof). Furthermore, if* $f(x,y)$ *is supermodular then* $-f(x,y)$ *is submodular, and vice versa.*

Given the notion of modularity, we may now define synergistic, independent, and antisynergistic interaction of two qualitative influences, following Wellman.[4]

Definition 3.5 (Semantics of Synergies and Independence) *Suppose that* $f_z(x,y)$ *is the cumulative distribution function for the conditional probability* $P(Z \mid X,Y)$*. Given two qualitative influence statements* $X \overset{Q+}{\succ} Z$*,* $Y \overset{Q+}{\succ} Z$*, we say they are "independent" (respectively, "synergistic", "antisynergistic") to mean that* $f_z(x,y)$ *is modular (supermodular, submodular).*

A similar definition may be provided for $\overset{Q-}{\succ}$.

This definition satisfies our informal explanation of independence and synergy provided above, because

$$\begin{aligned} f(x,y) + f(x',y') &= f(x,y') + f(x',y) \\ \Rightarrow f(x,y) - f(x',y) &= f(x,y') - f(x',y'), \end{aligned} \tag{3.5}$$

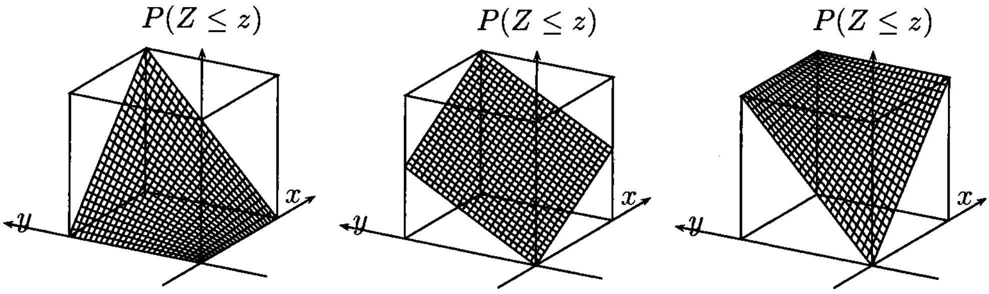
which precisely encodes the notion that changing X from x to x' has the same effect regardless of whether Y is y or y'.

Figure 3.3 displays examples of modular, supermodular, and submodular functions, as values of a cumulative distribution function over Z for some particular value of z.

3.5.3 Example

Finally, we consider a simple example of synergistic CPT constraints for binary variables. The primary part of Table 3.4 shows the CPT and the monotonicity constraints. The second part shows the constraints resulting from each of the three possible synergy statements. Note that some of the constraints are redundant. This

[4] Wellman's terms are "synergy", "zero synergy", and "sub-synergy", respectively.



Supermodular (synergistic)

Modular (independent)

Submodular (antisynergistic)

FIGURE 3.3: Graphs visualizing function modularity, interpreted as synergies as follows: $P(Z \mid X, Y)$ has a cdf which is a function of 3 parameters, visualized here by fixing one ($Z = z$) to obtain a surface showing $P(Z \leq z \mid X, Y)$. (If Z increased, the surface would rise, analogous to the way a one-dimensional cdf increases as its argument increases.) The synergy holds iff the associated modularity holds for every value z of Z.

is particularly true in the boolean case, where, for instance, the two constraints resulting from a synergy are in fact the same:

$$p_3 - p_1 \geq p_2 - p_0 \quad \Longleftrightarrow \quad p_3 - p_2 \geq p_1 - p_0.$$

3.6 Saturation

Saturation is an additional restriction on a monotonic influence. We here consider two different types, hard and soft saturation.

3.6.1 Hard Saturation

The first notion of saturation is that of a strict threshold value at which point an effect saturates (has no further effect). More specifically, we say $X \overset{Q+}{\succ} Y$ hard-saturates high (low) at a threshold point x_{sat} to mean that Y varies monotonically

<table>
<tr><th>x</th><th>y</th><th>$P(z = 1 \mid x, y)$</th><th>Constraints for $\overset{Q+}{\succ}$</th></tr>
<tr><td>0</td><td>0</td><td>p_0</td><td>(none)</td></tr>
<tr><td>0</td><td>1</td><td>p_1</td><td>$p_1 \geq p_0$ (monotonicity)</td></tr>
<tr><td>1</td><td>0</td><td>p_2</td><td>$p_2 \geq p_0$ (monotonicity)</td></tr>
<tr><td>1</td><td>1</td><td>p_3</td><td>$p_3 \geq p_1, p_3 \geq p_2$ (monotonicity)</td></tr>
<tr><td colspan="4">$$\text{and} \begin{cases} p_3 - p_1 = p_2 - p_0, & \\ \quad p_3 - p_2 = p_1 - p_0 & \text{if independent} \\ p_3 - p_1 \geq p_2 - p_0, & \\ \quad p_3 - p_2 \geq p_1 - p_0 & \text{if synergistic} \\ p_3 - p_1 \leq p_2 - p_0, & \\ \quad p_3 - p_2 \leq p_1 - p_0 & \text{if antisynergistic} \end{cases}$$</td></tr>
</table>

TABLE 3.4: Example of synergistic CPD constraints for boolean conditioning variables and a boolean resultant. The upper part shows the CPT and the constraints resulting from monotonicity; the lower shows the constraints which could result from different choices of synergistic interaction. In the boolean case, each pair of constraints is redundant; in general, we can expect many redundant constraints.

with X, but when X increases (decreases) past x_{sat}, it has no further effect on Y. The saturation point x_{sat} may be specified, or it may be unknown (simply known to exist). Our definition is as follows:

Definition 3.6 (Semantics of Hard Saturation) *Suppose that $X \overset{Q+}{\succ} Y$ hard-saturates high at x_{sat} (for $x_{sat} \in Dom(X)$). Then, taking $x_1 \leq x_2 \in Dom(X)$, and defining $\mathbf{Pa}_Y^{X=x}$ to be the set of configurations of parents of Y in which X equals x, we have three cases:*

1. *$x_1, x_2 \leq x_{sat}$: Standard monotonicity constraints apply*

2. *$x_1 \leq x_{sat} \leq x_2$: No constraints apply*

3. *$x_{sat} \leq x_1, x_2$: The saturation is achieved and for all $p_1 \in \mathbf{Pa}_Y^{X=x_1}, p_2 \in \mathbf{Pa}_Y^{X=x_2}$, we have $P(Y \mid p_1) = P(Y \mid p_2)$.*

A similar definition applies to hard-saturation low.

This definition is "local" in that it does not affect or contradict other qualitative statements about other influences of the same resultant.

3.6.2 Soft Saturation

The second notion of saturation (soft saturation) is that of a limiting value to which one variable can cause another to approach, but never exceed. Many observed phenomena show a sigmoid response curve which suggests soft saturation high and low. While intuitive, this soft saturation is significantly more difficult to formalize. We will discuss the issues rather than attempt a definition.

The first issue is that we cannot specify the saturation point as a point in the domain of the influent (since it essentially occurs at infinity); it must be a point in (or on the boundary of!) the domain of the resultant. Second, this in turn means that (unlike hard saturation) soft saturating influences are not local; it is not clear how to combine arbitrary statements. Suppose someone claims $X \overset{Q+}{\succ} Z$ soft-saturating at $Z = 10$ and $Y \overset{Q+}{\succ} Z$ soft-saturating at $Z = 12$. It is not clear that there is any sensible interpretation of the conjunction of these two statements. If one assumes the soft saturation points are the same, then one is apparently making the claim that this saturation point is a maximum value for the resultant. This casts doubt on the necessity of marking particular influences as soft-saturating on this resultant.

Another approach which might be more useful is to specify, instead of a maximum value for the resultant, a point of "diminishing returns" for the influent. The intuition is that when the influent increases past this point, it has a markedly lower effect. This could be formalized in terms of second derivatives or differences.

3.6.3 Discussion

In the qualitative physics literature, the notion of qualitatively different aspects of influence are usually modeled with "regimes". For example, instead of a monotonic influence saturating at a certain point, that certain point marks a regime transition into a new state in which the original influent no longer influences the resultant. We have adopted a different approach in this work because we feel that, in general, most such dynamics are local and are more simply modeled by local conditionals on the influence statements rather than global regime transitions which require an entirely new set of influence statements.

In any case, both types of saturation points are generally only useful for finely discretized or continuous variables. With few states (say, binary), a saturation makes little sense. With ordinal, non-interval, variables, one must be careful to ensure the discretization matches the notion of saturation. In particular, saturations may already be encoded into the discretization of the variable.

A saturation point introduces an additional parameter into the model (x_{sat}). It would be very useful to be able to learn such parameters, as their values are often not precisely known by domain experts. However, this task is beyond the scope of this thesis.

3.7 Strength of Relation

We now discuss strength of relation statements. We begin with two notions of strength of monotonicity, as it is an extension of monotonic constraints. We next discuss strength of relation for non-ordinal variables. For all, we discuss both absolute and relative statements.

3.7.1 Log-Odds Strength of Monotonicity

We first consider a quantitative notion of strength of monotonicity which applies to binary variables. We define the log odds ratio strength of monotonicity as follows:

Definition 3.7 (Log Odds-Ratios Monotonicity) *We define the log odds ratio of a boolean variable Y conditioned on a boolean variable X to be*

$$r_{XY} = \ln\left(\frac{p/(1-p)}{q/(1-q)}\right), \tag{3.6}$$

where $p = P(Y = f \mid X = f)$, $q = P(Y = f \mid X = t)$.

Specifying a value of r_{XY} imposes an equality constraint on cells of the CPT. Specifying a limit on the value of r_{XY} imposes inequality constraints. Another approach related to monotonicity and strength of monotonicity is discussed in [Agresti and Chuang, 1985], where the authors operationalize monotonicity in terms of a Gaussian prior or sign constraint on local log odds ratios. This is discussed further in Section 8.3.1.

For a $\overset{Q+}{\succ}$ relation, is it easy to show $r_{XY} > 0$. Thus, larger lower bounds on r_{XY} imply stronger monotonicity.

For ordinal multinomial influents and resultants, we have a variety of options for defining log odds ratio strength of monotonicity. One possibility is to constrain each local log odds ratio (i.e., that formed by each set of four adjacent cells) in the same way. Another is to define a threshold value which effectively discretizes the resultant into a binary variable, and then to constrain the log odds for each successive pair of values of the resultant variable. Finally, one could also define a threshold for the influent variable and constrain the log odds resulting from crossing this particular threshold. In the first case, by the assumption that the effect is the same at all levels of the influent and all levels of the resultant, it probably makes sense to assume both must be interval-measured. In the second case, it is likewise reasonable to assume the influent is interval-measured. These options are displayed visually in Figure 3.4.

Log odds ratio statements are standard and common in many fields, and may be available as domain knowledge in some cases.[5] However, they are still quantitative, coming from data, rather than a qualitative understanding of the domain. Thus, we would like a qualitative version as well.

A natural form of qualitative domain knowledge that a domain expert may have is a relative ordering on the strengths of monotonicity for sets of influents of a resultant. Even when the domain expert does not know the odds ratio of X on Z, he or she may know it is greater than the odds ratio of Y on Z.

[5] This brings up the interesting question of when domain knowledge is really prior domain knowledge. If a doctor specifies a log odds ratio which she has taken from an empirical study, is it valid to use as prior domain knowledge? We do not discuss this issue in depth here, but in applications it should not be ignored.

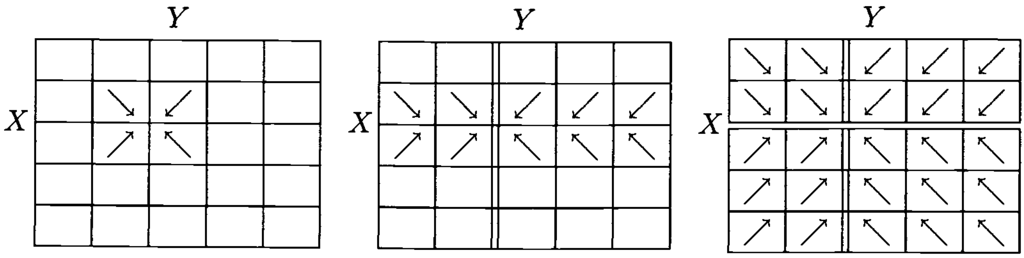


FIGURE 3.4: Three options for multinomial strength of monotonicity. The double lines indicate specified thresholds. Unthresholded dimensions are constrained at all levels: the left-hand figure applies the constraint to all adjacent 4-tuples of cells (one which one is marked to illustrate the idea), and the middle applies the constraint to all pairs of adjacent rows (one of which is marked). Each of the four arguments of the odds ratio (p, $1-p$, q, $1-q$) can be computed by summing the cells with the respective mark ($\searrow$, $\swarrow$, $\nearrow$, and $\nwarrow$).

A simple way to turn quantitative statements into qualitative ones is to define statements of relative specification of strength of monotonicity. For example, we could say that Y has a stronger influence than X on Z:

$$r_{XZ} < r_{YZ}.$$

This imposes inequality constraints on cells of the CPT. In this case, the resultants of both referenced influence statements must be the same. If the resultant is multinomial, then the boolean-discretization threshold must be the same for both influence statements.

3.7.2 Margin-Based Strength of Monotonicity

Strength of monotonicity turns out to have an important application as a practical consideration for learning the parameters of conditional probabilities. This is because, when observed data violates a monotonicity constraint, the maximum likelihood parameters are the same for each parent configuration. That is, if the data violates our monotonicity domain knowledge for a particular influence, then we essentially sever that Bayesian network arc and remove the influence.

It is debatable whether or not this is the intended or desired behavior. On the one hand, one could argue that if the data violates our prior belief, then perhaps we should not trust either, and the best option would be to ignore the influence. On the other hand, one could say that the point of expressing domain knowledge is to overcome insufficient and/or noisy data, and the monotonicity should prevail and somehow be enforced even in the face of observed data violating it.[6]

If we pursue the second point of view expressed above, a natural way to "enforce" the monotonicity is to use a strength of monotonicity statement. In the absence of particular domain knowledge about odds ratios, however, we cannot pick a justified strength value, and we are reduced to making an ad-hoc choice. As such, we simplify and make use of margins rather than odds ratios. Specifically, we enforce the strength of monotonicity by adding a margin to each inequality, replacing Equation 3.1 by

$$P_2 \succeq_{(1)} P_1 \quad \text{iff} \quad F_1(y) \geq \epsilon \wedge 1 - F_2(y) \geq \epsilon \implies F_2(y) + \epsilon \leq F_1(y)$$

As an ad-hoc monotonicity "enforcer", we consider this sufficient. Note that even for boolean variables, margin-based strength of monotonicity does not correspond directly to log odds ratio based strength of monotonicity (in the sense of one being equivalent to the other for an appropriate choice of ϵ given r_{XY}, or vice versa), due to the nonlinearity of the latter. We here consider a simple example (for $X \overset{Q+}{\succ} Y$ with both variables boolean). In this case the ϵ-margin constraint is simply $p_1 \leq p_0 - \epsilon$,

[6] The third option, obviously, is to use a soft prior, so that observed data contrary to the domain knowledge can override it. Though this is probably ultimately the best option, we leave it for future research.

while the log odds ratio constraint is given by

$$
\begin{aligned}
\ln \frac{p_0/(1-p_0)}{p_1/(1-p_1)} &\geq r \implies \\
\frac{(1-p_1)/p_1}{(1-p_0)/p_0} &\geq e^r \implies \\
\frac{1-p_1}{p_1} &\geq e^r \frac{1-p_0}{p_0} \implies \\
\frac{1}{p_1} &\geq 1 + e^r \frac{1-p_0}{p_0} \implies \\
p_1 &\leq \frac{1}{1+e^r \frac{1-p_0}{p_0}}.
\end{aligned}
\tag{3.7}
$$

These two constraints are plotted as constraint boundaries (for different values of their respective parameters) in Figure 3.5.

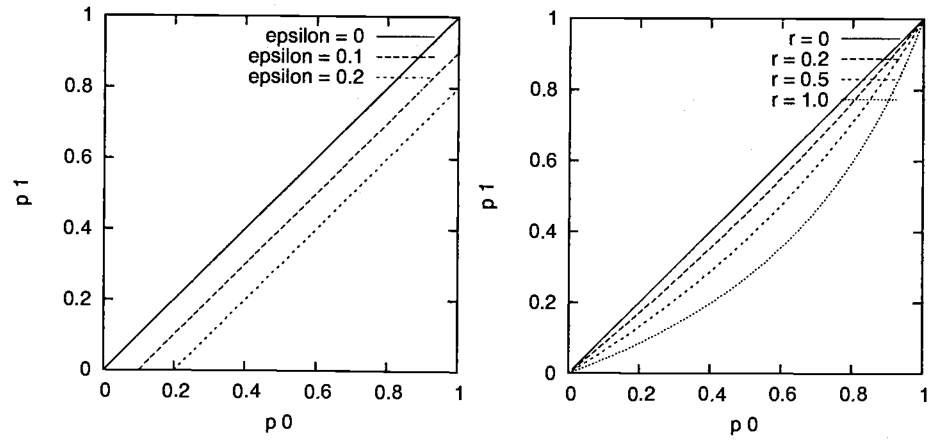


FIGURE 3.5: Constraint boundaries for two-parameter $\overset{Q+}{\succ}$ CPTs (binary influent, binary resultant) under strength of monotonicity constraints (left: ϵ-margin, right: log odds ratio)

This figure makes the intuitive difference clear: the ϵ-margin is a much stronger constraint at extreme probabilities (close to 0 or 1). In particular, it completely eliminates the possibility of one parameter approaching 0 or 1, no matter how extreme the other parameter. If we knew this were the correct prior, we should get more effective learning, but in general it is a riskier assumption. In our experiments, however, it is probably of little consequence because with small samples and a Laplace correction we will not see extreme probabilities.

We must be careful not to make ϵ too large, or it will strengthen the constraints to the point where they have no solution (this is because inequalities are transitive, e.g.: $F_1(y) + \epsilon \leq F_2(y)$, $F_2(y) + \epsilon \leq F_3(y), \ldots$). The greatest possible length of such a "chain" of inequalities bounds the maximum size of ϵ, since every distribution function F_i is bounded by 1.

Consider the simplest case first: that of a boolean resultant. If the influents are boolean and all monotonically constraining, the number of inequalities equals the number of parents, q_i. If the influents are not boolean, it will be greater. One way to visualize this is to think of the CPT, which is essentially q_i-dimensional[7], and has its monotonically influenced extrema in opposite corners. Thus, we can find the length of this chain of inequalities by computing the Manhattan distance between the minimum-influence and maximum-influence corners of the CPT, given by

$$d_1^i = \left|\mathbf{pa}_i^{max} - \mathbf{pa}_i^{min}\right|_1 = \prod_{X_j \in \pi_i} (r_j - 1), \tag{3.8}$$

where π_i is the set of parents of X_i. Therefore, we define a global margin parameter ε and let each node X_i have its own ϵ_i margin, where

$$\epsilon_i = \frac{\varepsilon}{d_1^i}. \tag{3.9}$$

Theoretically, ε could range up to 1.0, but we find that our current gradient search algorithms have difficulty finding the feasible region for ε greater than 0.2.

In general, this analysis remains true for non-boolean resultants, because the additional constraints imposed (i.e., ones involving inequalities on cumulative distributions made up of more than one term) operate "in parallel" with the constraints from the boolean-resultant case (i.e., the ones involving inequalities on single terms). More formally, given that the set of first terms of all cdfs (that is, the first column

[7] Or $q_i + 1$-dimensional, if you consider the dimension of the binary resultant as well.

of the CPT) satisfies the constraints, the rest of the constraints can be satisfied by setting all CPT entries except those in the last column to zero and setting the last column to satisfy the simplex constraint, i.e., one minus the value in the first column. Thus if the boolean-resultant case has a non-zero feasible region, so will the non-boolean case.

Returning to the boolean-resultant case, we consider an example. Suppose $q_i = 2$, and each parent has three states. In this case, we get $d_1^i = 4$ inequalities, and our maximum allowable value for ϵ is 0.25. This is illustrated in Figure 3.6.

		X_2		
		1	2	3
X_1	1	p^a	p_2^b	p_3^c
	2	p_1^b	p_2^c	p_2^d
	3	p_1^c	p_1^d	p^e
$P(Y = f \mid X_1, X_2)$				

$$\begin{aligned} p^a &\geq p_*^b + \epsilon \\ p_*^b &\geq p_*^c + \epsilon \\ p_*^c &\geq p_*^d + \epsilon \\ p_*^d &\geq p^e + \epsilon \end{aligned}$$

FIGURE 3.6: Counting d_1^i, the transitivity of inequalities, for two-parent, 3-bin case (p_1^b, p_2^b, and p_1^c, p_2^c, p_3^c, and p_1^d, p_2^d are not ordered). The displayed table is a slice of the CPT for $Y = \mathit{true}$.

This is not the only difficulty. In the case where the number of states of the node in question is greater than two, although the maximum technically allowable value for ϵ is not reduced, the redundancy of constraints and increased dimensionality of the problem means that gradient search techniques will have a very hard time finding the feasible region, or traversing through it. One can think of a very narrow feasible channel through which the parameters must jointly pass during likelihood maximization, even when their individual partials may have them moving towards the constraint boundaries and infeasible region.

As a final side note, we observe the following possible point of contention. As the number of bins is increased, the amount of data available per bin is decreased, meaning that each CPT cell's parameter will be estimated from less data. All else being equal, when we estimate from less data we want stronger priors or constraints. This suggests that there should be another factor in calculating ϵ_i, one which strengthens the margin when less data is available per bin (i.e., for finer discretizations).

3.7.3 Absolute Strength of Relation

Strength of monotonicity may be seen as a special case of a more general type of domain knowledge: strength of relations in general. Non-ordinal data cannot be restricted by monotonicity or strength of monotonicity statements, and if we wish to constrain the strength of influence, we must take a different approach. A natural approach is to use ideas from information theory: mutual information, entropy, and conditional entropy.

We will consider two definitions of absolute strength of relation that provide different semantics at different computational costs. The first is a measure which we call "observational" strength of influence because it is based on values measurable in a system with purely observed data (no interventions). The second we call "interventional" strength of influence because it is based on values measurable in a system in which we can intervene. The discussion below will clarify.

Definition 3.8 (Observational Strength of Relation) *We define the observational strength of influence for a qualitative relation* $X \overset{Q^{inf}}{\succ} Y$ *as* $s \in [0,1]$, *where*

$$\begin{aligned} s &= \frac{I(Y;X)}{H(Y)} \\ &= 1 - \frac{H(Y \mid X)}{H(Y)} \\ &= 1 - \frac{\sum_{x \in X} P(x) \sum_{y \in Y} P(y \mid x) \ln P(y \mid x)}{\sum_{y \in Y} P(y) \ln P(y)}, \end{aligned} \qquad (3.10)$$

where $H(Y \mid X)$ *is the conditional entropy of* Y *given* X, *and* $H(Y)$ *is the entropy of* Y, *and we take* $s = 0$ *if* $H(Y)$ *is zero. Note that* $H(Y) \geq H(Y \mid X) \geq 0$, *so the equation is well defined for* $s \in [0,1]$.

Intuitively, the problem with observational strength of influence is that it assigns information on Y unevenly between values in $dom(X)$ depending on their respective probabilities. This can be seen by the presence of the marginal $P(X)$ in the expression above. Since we are constraining not only the local conditional probability distribution $P(Y \mid X)$, or even the marginal $P(Y)$, but simultaneously constraining the marginal of the parent X, solving the constrained optimization problem becomes much more difficult, because it is not local to the parameters of the CPT $P(Y \mid X)$. It also involves terms from the marginal $P(X)$, which may be simultaneously constrained by further strength of relation statements. Thus, any subgraph of nodes connected by strength of relation statements must have all parameters for all nodes optimized jointly, a generally infeasible task.

In order to make the computation local, we wish to remove the marginal of the parent from the formula. Therefore we form a new definition which assumes a uniform distribution on the parent X, and replace $P(x)$ with $\frac{1}{|Dom(X)|}$. This can be viewed as either an approximation to the real-world case, or as a counterfactual view in that we are measuring the information provided about the child supposing that we could intervene and control the parent.

Definition 3.9 (Interventional Strength of Influence) *Taking $P(y \mid X_{unif})$ to be the probability of y given a uniform distribution over the parent X, we define the interventional strength of influence as*

$$\begin{aligned} s &= 1 - \frac{\sum_{x \in X} \frac{1}{|Dom(X)|} \sum_{y \in Y} P(y \mid x) \ln P(y \mid x)}{\sum_{y \in Y} P(y \mid X_{unif}) \ln P(y \mid X_{unif})} \\ &= 1 - \frac{\sum_{x \in X, y \in Y} P(y \mid x) \ln P(y \mid x)}{\sum_{y \in Y} P(y \mid X_{unif}) \ln P(y \mid X_{unif})}. \end{aligned} \tag{3.11}$$

3.7.4 Relative Strength of Relation

Absolute strength of relation is of very limited use. The problem is that it is quantitative rather than qualitative. Domain experts will not in general have prior knowledge about the mutual information between two random variables in their field of study (even if occasionally they may have prior knowledge of odds ratios for monotonic influences). As we mentioned before, such knowledge would (generally) come from data, not from a qualitative understanding of the domain.

Again, it is reasonable to expect that domain experts may have knowledge of relative orderings on the strengths of influence for sets of influents of a resultant. The domain expert may not know the precise mutual information of X and Y, but he or she may know it is greater than the mutual information of X and Z.

Thus, we now consider relative strength of relation statements. These could be based either on observational or interventional statements; for the computational reasons discussed above, we will assume we are restricted to interventional statements.

Definition 3.10 (Relative Strength of Influence, Interventional) *Suppose X influences Z with strength s_X and Y influences Z with strength s_Y. Stating that X has a greater influence than Y implies the constraint*

$$s_X > s_Y.$$

Deriving from the above definition, we have

$$\begin{aligned} s_X &> s_Y \quad \Rightarrow \\ H(Z \mid X) &< H(Z \mid Y) \\ \sum_{x \in X, z \in Z} P(z \mid x) \ln P(z \mid x) &< \sum_{y \in Y, z \in Z} P(z \mid y) \ln P(z \mid y). \end{aligned} \tag{3.12}$$

One difficulty still present with relative strength of relation statements is that, unlike monotonicities and synergies, the constraints imposed are complex and nonlinear. This may make it difficult to use these statements in a machine learning algorithm.

Chapter 4:

Knowledge-Constrained Statistical Learning

Our general strategy is to use statements of monotonicities and other qualitative knowledge to impose priors on the parameters of our models. For example, we could view our learning goal as the computation of the parameter values which best fit the data while conforming to our qualitative constraints. This would be a form of constrained Maximum Likelihood Estimate (MLE). More generally, however, we are able to perform Maximum a Posteriori (MAP) estimation when the prior consists of a Dirichlet distribution along with the qualitative constraints.

In this section we will derive the formulas and explain the techniques required to learn parameters under the constraints imposed by some of the qualitative statements we have discussed. In particular, we will discuss monotonicity statements, and margin-based strength of monotonicity. Although we do not provide the details, synergistic statements impose inequality constraints on cumulative distributions from the CPT which are very similar to the constraints imposed by monotonicities, and so the analysis provided for monotonicities extends easily to handle synergies.

4.1 Learning Qualitatively-Constrained Conditional Distributions

We now investigate the derivation of the parameter estimates, using notation following [Heckerman, 1999], some of which has already been introduced.

Our parameters are jointly referred to as θ, which is the set of matrices (θ_i for $i \leq n$) representing the conditional probability distributions for the n nodes in the graph G. We denote the number of states of a discrete variable (node) $X_i \in G$ by r_i. We let $\mathbf{Pa}_i$ represent the set of possible configurations of the parent nodes of node X_i and let j index into the set to represent some particular configuration $\mathbf{pa}_i^j$. When the context is clear, we refer to the parent configuration merely by its index j. The number of such configurations is $q_i = |\mathbf{Pa}_i|$. Thus, θ_{ij} represents the parameter vector for parent configuration $\mathbf{pa}_i^j$ and θ_{ijk} the parameter for (probability of) value k given parent configuration $\mathbf{pa}_i^j$. Finally, our fully observed data is denoted D, and our prior over the parameters is denoted ξ, consisting of ξ^G, the conditional independencies corresponding to the structure of G; ξ^Q, the monotonicity, synergistic, and strength of influence constraints implied by our qualitative model Q; and ξ^P, the prior over parameter values (e.g., a Dirichlet distribution).

The full Bayesian approach to learning involves computing the posterior distribution ($P(\theta \mid D, \xi)$) over θ given the data and our prior. This allows us to estimate the probability of events which depend on θ, for example, some new data D':

$$P(D') = \int_{\theta} P(D' \mid \theta) dP(\theta).$$

Full Bayesian model averaging, however, is generally computationally intractable, and for our purposes, we will assume a maximum a posteriori parameter estimate will be sufficient. In this case we simply find the most probable value for θ, which is

$$\begin{aligned}
\hat{\theta} &= \text{argmax}_{\theta} P(\theta \mid D, \xi) \\
&= \text{argmax}_{\theta} \frac{P(D \mid \theta, \xi) P(\theta \mid \xi)}{P(D \mid \xi)} \\
&= \text{argmax}_{\theta} P(D \mid \theta) P(\theta \mid \xi),
\end{aligned}$$

since the probability of the observed data $P(D \mid \xi)$ is a constant that does not depend on the parameters θ, and because D depends on ξ only through θ. Furthermore, we have

$$\hat{\theta} = \text{argmax}_{\theta} \underbrace{P(D \mid \theta)}_{1} \underbrace{P(\theta \mid \xi_i^Q, \xi^P)}_{2} \underbrace{P(\xi_i^Q, \xi^P \mid \xi^G)}_{3} \tag{4.1}$$

because ξ^G (our graph structure assumption) conditions the other priors.

We now examine factor 2 in Equation 4.1, which further factors into two parts, $P(\theta \mid \xi_i^Q)$ and $P(\theta \mid \xi_i^P)$. The former part is zero when θ_i violates the constraints, and a constant value (say k_i; the actual value will not matter during maximization) when it satisfies the constraints. We may thus write it as the constant times an indicator function: $P(\theta \mid \xi_i^Q) = k_i I_{(\theta_i \in \Theta_i^Q)}$, where we take Θ_i^Q to be the set of θ_i satisfying the constraints of Q. The latter part, $P(\theta \mid \xi_i^P)$, is our prior over the parameter values, ignoring the monotonicity constraints. Factoring this requires us to make use of the graph structure assumption, encoded in factor 3 in 4.1. Under this assumption, our prior over parameter values can be decomposed per node and parent configuration. This assumption also allows us to factor the likelihood (factor 1 in 4.1) the same way. Hence, we have

$$\begin{aligned}
\hat{\theta} &= \text{argmax}_{\theta} \Big(P(D \mid \theta)\Big) \Big(P(\theta \mid \xi_i^Q, \xi^P)\Big) P(\xi_i^Q, \xi^P \mid \xi^G) \\
&= \text{argmax}_{\theta} \left(\prod_{i=1}^{n} \prod_{j=1}^{q_i} P(D_{ij} \mid \theta_{ij}) \right) \left(k_i I_{(\theta_i \in \Theta_i^Q)} \prod_{i=1}^{n} \prod_{j=1}^{q_i} P(\theta_{ij} \mid \xi_{ij}^P) \right)
\end{aligned}$$

Our data is multinomial, and thus our likelihood is a Dirichlet distribution [Heckerman, 1999]. We denote the counts of observed data by N^O_{ijk}. We assume that we express our prior ξ^P as a Dirichlet distribution as well (with pseudocounts or hyperparameters denoted by α_{ijk}), which allows us to expand and combine as follows (letting $N_{ijk} = N^O_{ijk} + \alpha_{ijk} - 1$):

$$\begin{aligned}
\hat{\theta} &= \operatorname{argmax}_\theta \left(\prod_{i=1}^{n} \prod_{j=1}^{q_i} Dir(\theta_{ij1}, \ldots \theta_{ijr_i} \mid N^O_{ij1}, \ldots N^O_{ijr_i}) \right) \\
&\quad \left(k_i I_{(\theta_i \in \Theta_i^Q)} \prod_{i=1}^{n} \prod_{j=1}^{q_i} Dir(\theta_{ij1}, \ldots \theta_{ijr_i} \mid \alpha_{ij1}, \ldots \alpha_{ijr_i}) \right) \\
&= \operatorname{argmax}_\theta \prod_{i=1}^{n} k_i I_{(\theta_i \in \Theta_i^Q)} \prod_{j=1}^{q_i} Dir(\theta_{ij1}, \ldots \theta_{ijr_i} \mid N_{ij1} + 1, \ldots N_{ijr_i} + 1) \\
&= \operatorname{argmax}_{\theta \in \Theta^Q} \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\sum_k^{r_i} N_{ijk})}{\prod_k^{r_i} (\Gamma(N_{ijk}))} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}} \\
&= \operatorname{argmax}_{\theta \in \Theta^Q} \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}.
\end{aligned}$$

We thus have a large number of maximization tasks which are independent, aside from the constraint $\theta \in \Theta^Q$.

4.2 Constrained MAP Optimization

There is a well-established field of techniques for optimization (see for instance [Bazaraa et al., 1993, Pierre, 1986, Fletcher, 1987]). Many techniques are available for different types of functions (linear, quadratic, differentiable, and so on) under different types of constraints.

Lagrange multipliers are useful for exact solutions to problems with equality constraints and can be extended to handle problems with inequality constraints. The problem with Lagrange multipliers for our constrained optimization problem is that in general, for each constraint we must consider the problem in the case in which the constraint is active, as well as the case in which the constraint is inactive.

With potentially hundreds or thousands of constraints, this approach does not scale well [Pierre, 1986, p.45].

As an aside, it has recently come to our attention that it may be possible to analytically determine which constraints are active at the solution point. For example, in two and three dimensions, it is fairly obvious that the constraints which are violated at the MLE point will be the constraints which are active at the solution point. We do not have a proof that this is a general rule, however, so for the time being, we do not take this approach. Such a result seems plausible, however, since in θ-space, we have a convex (or concave) optimization problem. This can be easily shown by first noticing that the constraints are affine and, when non-contradictory, form a convex feasible region. Second we show the Hessian of our log likelihood $J_L(\theta_i) = \ln \prod_{jk} \theta_{ijk}^{N_{ijk}}$ is negative semi-definite. In particular,

$$\frac{\partial}{\partial\theta_{ij_2k_2}}\frac{\partial}{\partial\theta_{ij_1k_1}}J_L(\theta_i) \quad = \quad \frac{\partial}{\partial\theta_{ij_2k_2}}\frac{\partial}{\partial\theta_{ij_1k_1}}\sum_{jk} N_{ijk}\ln\theta_{ijk} \quad = \quad \frac{\partial}{\partial\theta_{ij_2k_2}}N_{ij_1k_1}\frac{1}{\theta_{ij_1k_1}},$$

which is zero if $j_1 \neq j_2$ or $k_1 \neq k_2$, and otherwise, letting $j = j_1 = j_2$ and $k = k_1 = k_2$, is $-N_{ijk}\theta_{ijk}^{-2}$. Thus the Hessian is diagonal with all entries less than or equal to zero.

We instead adopt an exterior penalty method, subtracting from the objective function penalizers which take on large values when the constraints are violated. This approach, while perhaps more subject to failure (for example, the algorithm may not find the feasible region at all!), is flexible and scales linearly with the number of constraints.

4.2.1 Reparameterization and Notation

To simplify the problem, however, we first eliminate the need for the simplex constraints (i.e., that $\forall ij\ \sum_k \theta_{ijk} = 1$) with a reparameterization. We define μ_{ijk} such that

$$\theta_{ijk} \equiv \frac{\exp(\mu_{ijk})}{\sum_{k'=1}^{r_i}\exp(\mu_{ijk'})}. \tag{4.2}$$

Of course, this adds a redundant parameter (one of the μ's, which could be arbitrarily set to zero or some other constant value). Although not fixing this parameter leaves an extra degree of freedom which could lead to runaway optimization, we have seen no such troubles in practice. Additionally, we avoid asymmetry problems: if we fix a parameter and the learning algorithm suggests that this parameter should be increased, we can only obtain this effect by decreasing all other parameters.

To avoid overly verbose formulae in the following discussion, we introduce an abbreviated notation, defining

$$Z^i_{jk_c} \equiv \sum_{k=1}^{k_c} \exp(\mu_{ijk}). \tag{4.3}$$

If the second subscript is omitted, we have the usual normalizing constant (specifically, we take the sum over the full range of values for k, that is, $Z^i_j \equiv Z^i_{jr_i}$).

We also make frequent use of indicator functions (i.e., $I_{(exp)}$, which takes the value 1 when the expression is true and 0 when it is false). For example, we will often use the following fact:

$$\frac{\partial}{\partial \mu_{ijk}} Z^i_{j_1 k_c} \;=\; I_{(j=j_1)}\; I_{(k \le k_c)}\; \exp(\mu_{ijk}). \tag{4.4}$$

Finally, from this point forward, we will occasionally omit the upper bound in summations and products over parent configurations and local states (normally denoted j and k respectively). In this case, the bounds are taken to be complete, that is, $1 \ldots q_i$ for j, and $1 \ldots r_i$ for k.

4.2.2 Likelihood Function

Our goal is thus to maximize, subject to the qualitative constraints (our simplex constraints being satisfied by the reparameterization), the expression for the likelihood:

$$\prod_{jk} \left(\frac{\exp(\mu_{ijk})}{\sum_{k'} \exp(\mu_{ijk'})} \right)^{N_{ijk}}. \tag{4.5}$$

It is equivalent to maximize the natural logarithm of this expression, which gives us our unpenalized objective function for the unconstrained log-likelihood of the parameters:

$$\begin{aligned} J_L(\theta_i) &= \sum_{jk} N_{ijk} \left(\mu_{ijk} - \ln \left(\sum_{k'} \exp(\mu_{ijk'}) \right) \right) \\ &= \sum_{jk} N_{ijk} \left(\mu_{ijk} - \ln Z^i_j \right) \end{aligned}$$

4.2.3 The Gradient of the Log-Likelihood

We now compute the gradient of the log-likelihood function. We consider a particular partial (note that we change the summation indices in the objective function to keep them distinct from the indices of the μ to which we are taking the partial with respect). This is given by

$$\begin{aligned} \frac{\partial}{\partial \mu_{ijk}} J_L(\theta_i) &= \frac{\partial}{\partial \mu_{ijk}} \sum_{j'k'} N_{ij'k'} \left(\mu_{ij'k'} - \ln Z^i_{j'} \right) \\ &= \left(\sum_{j'k'} N_{ij'k'} \frac{\partial}{\partial \mu_{ijk}} \mu_{ij'k'} \right) - \left(\sum_{j'k'} N_{ij'k'} \frac{\partial}{\partial \mu_{ijk}} \ln Z^i_{j'} \right) \\ &= \left(\sum_{j'k'} N_{ij'k'} I_{(j=j' \wedge k=k')} \right) - \left(\sum_{j'k'} N_{ij'k'} \frac{\frac{\partial}{\partial \mu_{ijk}} Z^i_{j'}}{Z^i_{j'}} \right) \\ &= N_{ijk} - \sum_{j'k'} N_{ij'k'} \frac{I_{(j=j')} \exp(\mu_{ijk})}{Z^i_{j'}} \qquad \text{(By Eq. 4.4)} \\ &= N_{ijk} - \sum_{k'} N_{ijk'} \frac{\exp(\mu_{ijk})}{Z^i_j} \\ &= N_{ijk} - \frac{\exp(\mu_{ijk})}{Z^i_j} \sum_{k'} N_{ijk'}. \end{aligned} \tag{4.6}$$

Note that this can also be written as $N_{ijk} - \theta_{ijk} \sum_{k'} N_{ijk'}$. Since the sum is just the total number of counts for the given parent configuration, the formula represents the difference between the observed counts and the expected counts predicted by the parameters (θ). It will be zero precisely when the two agree.

4.2.4 Constraints and Penalty Functions

We now consider our exterior penalty functions. Individual inequality constraints resulting from monotonicity statements are indexed by four variables: the node i to which the constraint applies, the two parent configurations j_1, j_2 being compared, and the state index k_c for which the cumulative distribution function is evaluated (c is not a variable, but only stands for "cumulative distribution function"). Without loss of generality, we consider monotonically increasing constraints ($\overset{Q+}{\succ}$), such that the cdf corresponding to parent configuration j_1 is everywhere greater than or equal to the cdf for j_2 (perhaps by an ϵ-margin), when j_1 and j_2 are the same for all parents except one, for which the value specified by j_2 is greater than that specified by j_1. We denote a particular constraint as $C^{i,k_c}_{j_1,j_2}$. In a model with only monotonicity statements, and with only one parent (so j_1, j_2, q_i are scalar and thus can be naturally ordered), the complete set of qualitative constraints is

$$\xi^Q_i \quad = \quad \{C^{i,k_c}_{j_1,j_2} \mid j_1 < j_2 \leq q_i \wedge k_c < r_i\}.$$

The case with multiple parents follows analogously, though the notation becomes verbose.

The constraints are formally defined in terms of our parameters as follows (note the use of ϵ-margins as discussed in Section 3.7.2):

$$\begin{aligned} C^{i,k_c}_{j_1,j_2} &\Leftrightarrow \\ 0 &\geq P(X_i \leq k_c \mid \mathbf{pa}_i^{j_2}) - P(X_i \leq k_c \mid \mathbf{pa}_i^{j_1}) + \epsilon \\ &= \sum_{k'=1}^{k_c} \theta_{ij_2k'} - \sum_{k'=1}^{k_c} \theta_{ij_1k'} + \epsilon \\ &= \sum_{k'=1}^{k_c} \frac{\exp(\mu_{ij_2k'})}{Z^i_{j_2}} - \sum_{k'=1}^{k_c} \frac{\exp(\mu_{ij_1k'})}{Z^i_{j_1}} + \epsilon \\ &= \frac{Z^i_{j_2k_c}}{Z^i_{j_2}} - \frac{Z^i_{j_1k_c}}{Z^i_{j_1}} + \epsilon \\ &\equiv \delta. \end{aligned} \tag{4.7}$$

This term, denoted as δ, will be positive when the constraint is violated. Thus, we define the natural penalty function $P^{i,k_c}_{j_1,j_2}$ for each constraint $C^{i,k_c}_{j_1,j_2}$ as follows:

$$P^{i,k_c}_{j_1,j_2} = I_{(\delta>0)}\ \delta^2. \tag{4.8}$$

4.2.5 Gradients of Penalty Functions

The gradient of this penalty function can be found with the derivation

$$\begin{aligned}
&\frac{\partial}{\partial \mu_{ijk}} P^{i,k_c}_{j_1,j_2} \\
=\ & \frac{\partial}{\partial \mu_{ijk}} I_{(\delta\geq 0)}\ \delta^2 \\
=\ & 2I_{(\delta\geq 0)}\ \delta \left(\frac{\partial}{\partial \mu_{ijk}} \frac{Z^i_{j_2 k_c}}{Z^i_{j_2}} - \frac{\partial}{\partial \mu_{ijk}} \frac{Z^i_{j_1 k_c}}{Z^i_{j_1}} + \frac{\partial}{\partial \mu_{ijk}} \epsilon \right) \\
=\ & 2I_{(\delta\geq 0)}\ \delta \left(\frac{Z^i_{j_2} \frac{\partial}{\partial \mu_{ijk}} Z^i_{j_2 k_c} - Z^i_{j_2 k_c} \frac{\partial}{\partial \mu_{ijk}} Z^i_{j_2}}{\left(Z^i_{j_2}\right)^2} - \frac{Z^i_{j_1} \frac{\partial}{\partial \mu_{ijk}} Z^i_{j_1 k_c} - Z^i_{j_1 k_c} \frac{\partial}{\partial \mu_{ijk}} Z^i_{j_1}}{\left(Z^i_{j_1}\right)^2} \right) \\
=\ & 2I_{(\delta\geq 0)}\ \delta \left(\frac{Z^i_{j_2} I_{(j=j_2 \wedge k\leq k_c)}\ \exp(\mu_{ijk}) - Z^i_{j_2 k_c} I_{(j=j_2)}\ \exp(\mu_{ijk})}{\left(Z^i_{j_2}\right)^2} \right. \\
& \qquad \left. - \frac{Z^i_{j_1} I_{(j=j_1 \wedge k\leq k_c)}\ \exp(\mu_{ijk}) - Z^i_{j_1 k_c} I_{(j=j_1)}\ \exp(\mu_{ijk})}{\left(Z^i_{j_1}\right)^2} \right) \\
=\ & 2I_{(\delta\geq 0)}\ \delta \exp(\mu_{ijk}) \Big(I_{(j=j_2)}\ \left(I_{(k\leq k_c)}\ Z^i_{j_2} - Z^i_{j_2 k_c}\right) / \left(Z^i_{j_2}\right)^2 \\
& \qquad -I_{(j=j_1)}\ \left(I_{(k\leq k_c)}\ Z^i_{j_1} - Z^i_{j_1 k_c}\right) / \left(Z^i_{j_1}\right)^2 \Big) \\
=\ & 2I_{(\delta\geq 0)}\ \delta \exp(\mu_{ijk}) \left(I_{(j=j_2)}\ - I_{(j=j_1)}\ \right) \left(\left(I_{(k\leq k_c)}\ Z^i_j - Z^i_{jk_c}\right) / \left(Z^i_j\right)^2 \right).
\end{aligned} \tag{4.9}$$

Due to the squared δ, the gradient is everywhere continuous, though the Hessian is not. (We suspected that the discontinuous Hessian could cause problems for the L-BFGS optimizer—see Section 5.3—but using cubes instead of squares did not seem to make any appreciable difference.)

4.2.6 Total Objective Function

Using exterior penalty methods, the final function to optimize will be the log-likelihood minus the sum of the penalty functions times a penalty weight w:

$$J(\theta_i) = J_L(\theta_i) - w \sum_{C^{i,k_c}_{j_1,j_2} \in \xi^Q_i} P^{i,k_c}_{j_1,j_2}. \tag{4.10}$$

We provide four examples of this total objective function for a simple single parent ($X \overset{Q+}{\succ} Y$) boolean-boolean CPT. Figures 4.1, 4.2, 4.3, 4.4 display the exponential of the objective function (essentially, the penalized likelihood) in θ-space, for a penalty weight of 200. Figures 4.5, 4.6, 4.7, 4.8 display the same functions but in the reparameterized μ-space. The plots show the surface of the objective function (higher values corresponding to better parameter settings), along with contours of that surface and the gradient vector field, which shows the direction gradient ascent will take in finding the maximum.

The graphs of the gradients use a penalty weight of 10, and the vectors have magnitude equal to the logarithm of one plus the actual vector magnitude, divided by 10. These choices were made for better visualization. In μ-space, it was not necessary to divide the gradient vector magnitudes by 10.

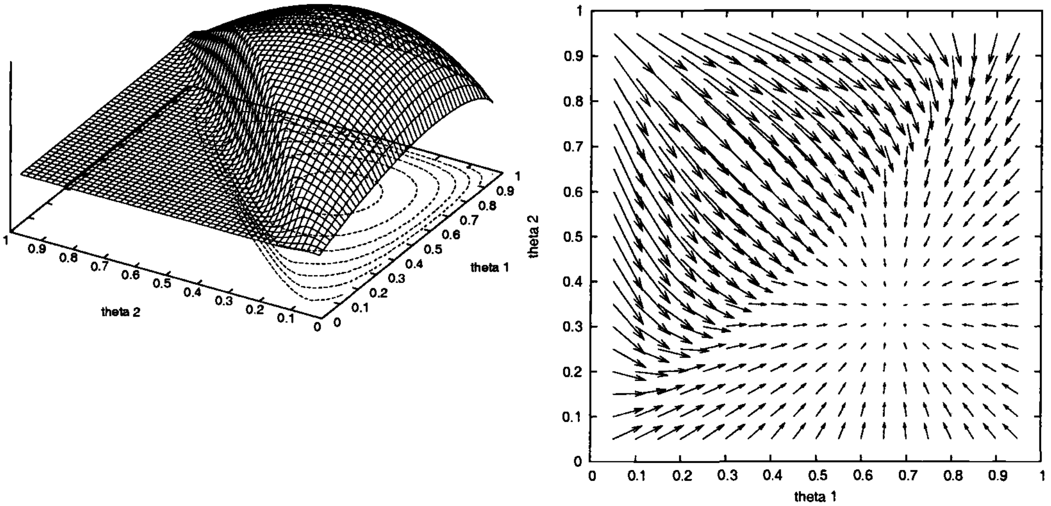


FIGURE 4.1: Objective function for symmetric counts and inactive constraints

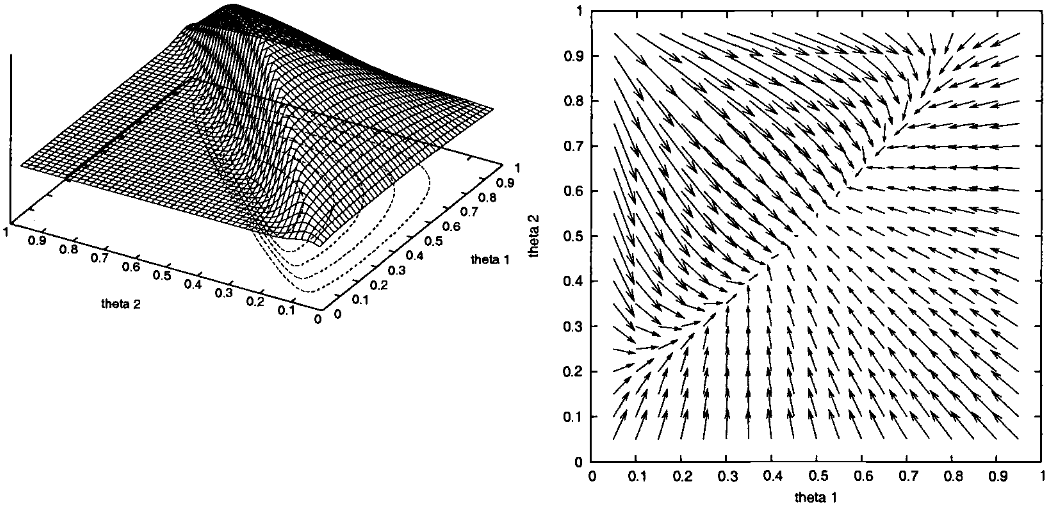


FIGURE 4.2: Objective function for symmetric counts and active constraints

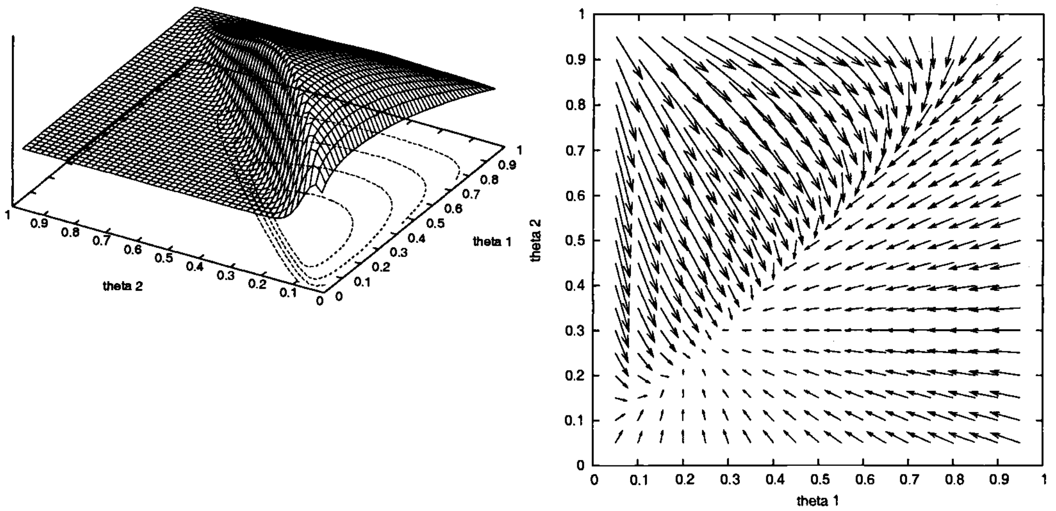


FIGURE 4.3: Objective function for asymmetric counts (θ_1 and θ_2 not near 0.5) and active constraints

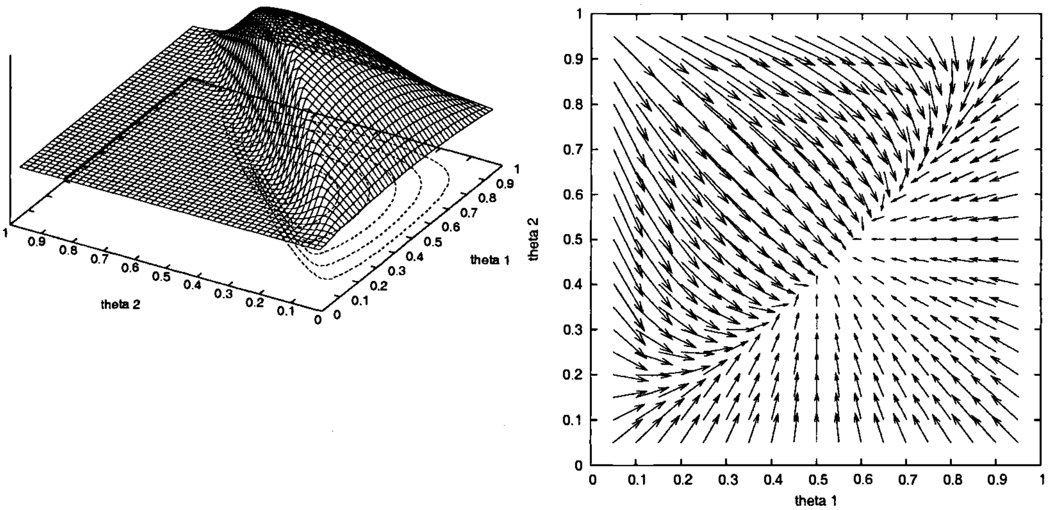


FIGURE 4.4: Objective function for symmetric counts, active constraints and a 0.1 ϵ-margin

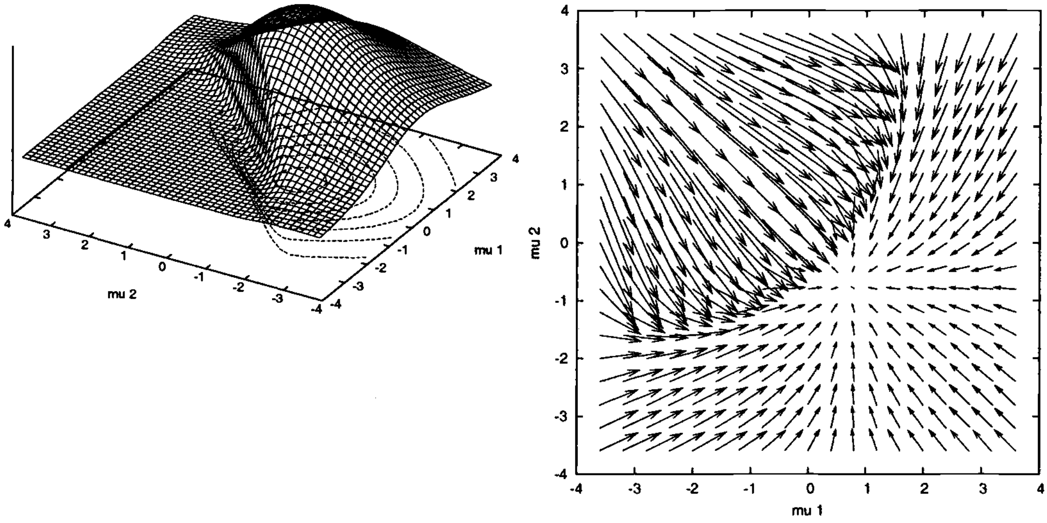


FIGURE 4.5: Objective function for symmetric counts and inactive constraints

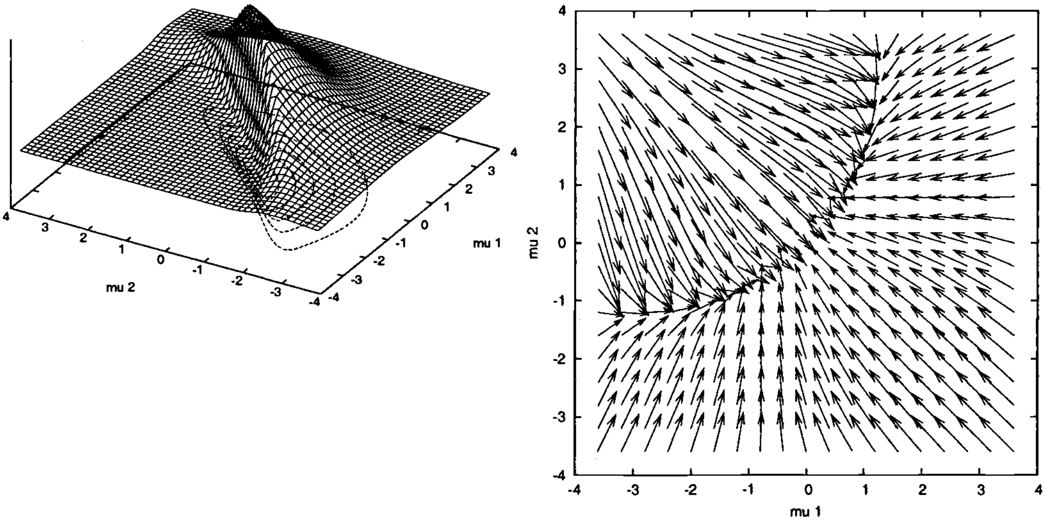


FIGURE 4.6: Objective function for symmetric counts and active constraints

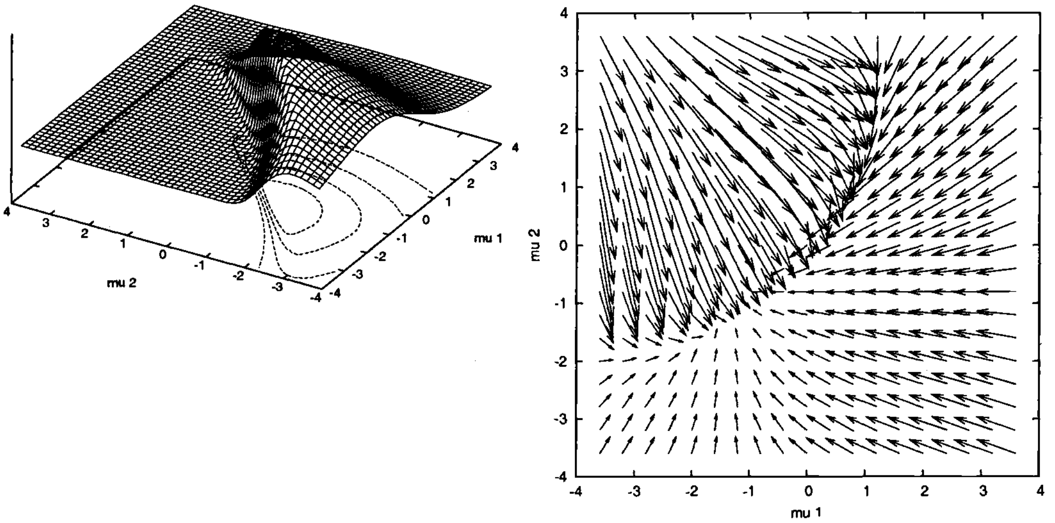


FIGURE 4.7: Objective function for asymmetric counts (θ_1 and θ_2 not near 0.5) and active constraints

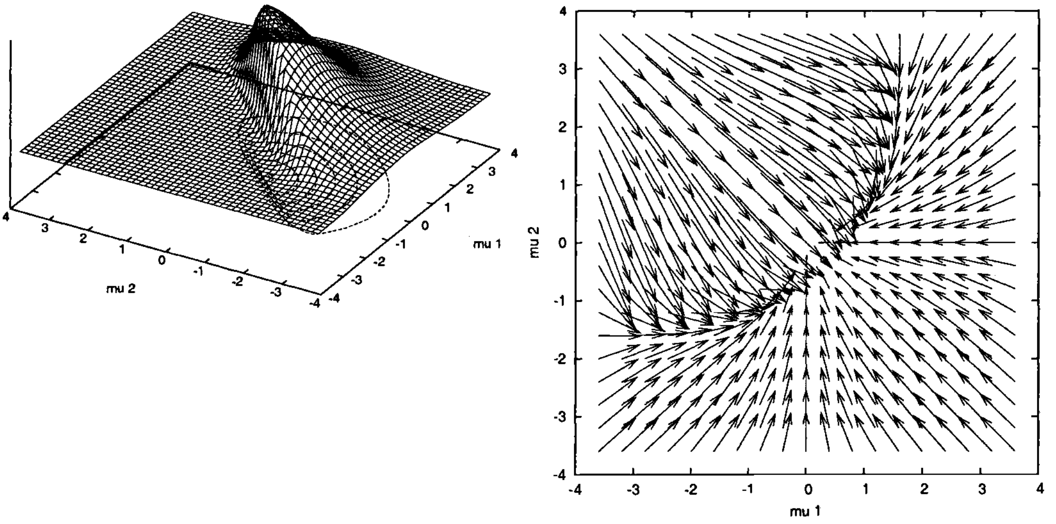


FIGURE 4.8: Objective function for symmetric counts, active constraints and a 0.1 ϵ-margin

Chapter 5:
Experiments and Evaluation

To determine how well qualitative monotonicities assist Bayesian network classifiers in learning from data and generalizing to new data, we designed and ran a set of experiments comparing constrained Bayesian networks against unconstrained networks, as well as other classifiers.

We also ran experiments on synthetic data which allowed us to estimate the frequency with which we can expect the constraints to have an effect. We consider these results first.

5.1 Synthetic Data Evaluations

We here consider data drawn from a known probability distribution. In particular, we consider the CPTs shown in Table 5.1. In the first case, we have $p_0 = 0.6 \geq p_1 = 0.4$, and in the second, $p_0 = 0.7 \geq p_1 = 0.3$. Both satisfy the monotonicity constraint $X \overset{Q+}{\succ} Y$ for $\epsilon \leq 0.2$.

	$P(Y \mid X)$	
x	$y = 0$	$y = 1$
0	$p_0 = 0.6$	0.4
1	$p_1 = 0.4$	0.6

	$P(Y \mid X)$	
x	$y = 0$	$y = 1$
0	$p_0 = 0.7$	0.3
1	$p_1 = 0.3$	0.7

TABLE 5.1: CPTs used to generate synthetic data.

We may generate n data points according to the probability distribution given by the CPT (sampling $n/2$ from the first row and $n/2$ from the second row, effectively assuming a uniform distribution over the parent). From these data points, we may compute the maximum likelihood estimates (with a Laplace correction) for p_0 and p_1, which we will call $\hat{p}_0$ and $\hat{p}_1$. We plot 500 such estimates in each subfigure in Figure 5.1.

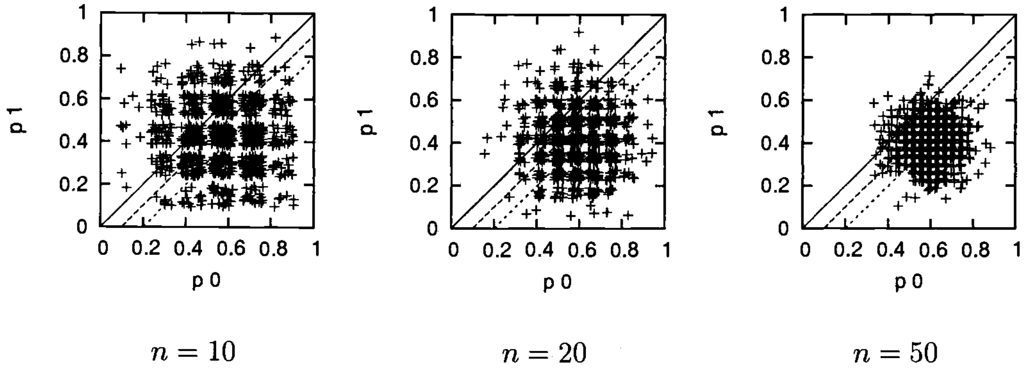
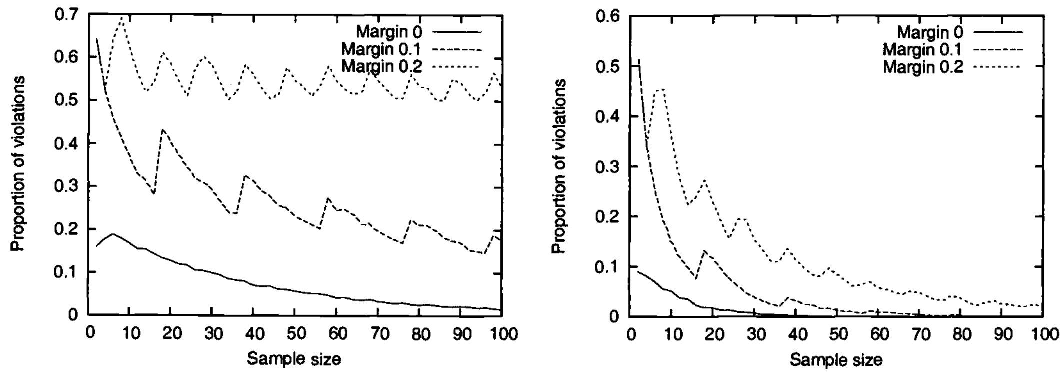


FIGURE 5.1: Illustration of violation of monotonicity constraints: three scatterplots, each of 500 independent estimates of $\hat{p}_0$ and $\hat{p}_1$ (jittered), given $n = 10, 20$, or 50 samples from the monotonic CPT with $p_0 = 0.6 \geq p_1 = 0.4$. Also plotted are monotonicity constraint boundaries including ϵ-margins 0.1 and 0.2 (constraints $p_0 \geq p_1 + \epsilon$).

The feasible region is the lower portion of the plotted parameter space, so a constraint is violated when an estimate lies above the respective boundary. As can be seen intuitively in the sequence of plots, with larger n we expect the constraints to be violated a smaller proportion of the time.

In Figure 5.2 we demonstrate this relationship by plotting the proportion of violations (y-axis) against n (x-axis, only even values of n), for margins $\epsilon = 0.0, 0.1, 0.2$. These plots were generated from simulated data, with 10,000 repeats at each n.



$p_0 = 0.6 \geq p_1 = 0.4$ $p_0 = 0.7 \geq p_1 = 0.3$

FIGURE 5.2: Proportion of violations for a given n at ϵ-margins 0, 0.1, and 0.2, with data sampled from the displayed true distribution.

The first thing to note is that, overall, the frequency of violation decreases with sample size n, as we expect.

There is another obvious pattern that requires explanation: the non-zero margin curves show regular periodic increases in frequency of violation as n increases. This is because of the discrete clustering of estimates (as was seen in Figure 5.1). Since we estimate $\hat{p_0}, \hat{p_1}$ from only $n/2$ data points each, with a Laplace correction, our estimates will be chosen from $\left\{\frac{1}{n/2+1}, \frac{2}{n/2+1} \cdots \frac{n/2}{n/2+1}\right\}$. (Recall that we pick only n even for this experiment.) As n increases, the discretization becomes finer. At certain values of n, a diagonal row of estimates will lie just barely on the "wrong" side of the constraint boundary, leading to a disproportionately large number of estimates violating the constraints. This effect diminishes with larger values of n.

The plots show approximate ranges of expected effectiveness. For the strongly monotonic data (the right-hand plot, with $p_0 = 0.7 \geq p_1 = 0.3$), the zero-margin constraint appears effective only through about 10 or 20 samples, while the margin-constrained estimates remain effective for much longer. For the more weakly monotonic data (the left-hand plot, with $p_0 = 0.6 \geq p_1 = 0.4$), the no-margin constraint appears significantly effective up to $n = 30$, 40 or 50, while the medium constraint ($\epsilon = 0.1$) remains effective throughout all plotted n. The strongest constraint ($\epsilon = 0.2$), of course, remains effective approximately 50% of the time for all n, since the true parameter values lie directly on the constraint boundary.

5.2 Benchmark Application Data Sets

We have chosen five data sets from the University of California at Irvine Machine Learning repository: auto-mpg [Quinlan, 1993], haberman [Haberman, 1976], pima-indian-diabetes [Smith et al., 1988], breast-cancer-wisconsin [Bennett and Mangasarian, 1992], and car [Bohanec and Rajkovic, 1988, Zupan et al., 1997]. For each of these data sets we constructed the structure of the network (KB structure) using domain knowledge, and inserted monotonicity annotations ($\overset{Q+}{\succ}$ or $\overset{Q-}{\succ}$) on each of the network links according to our domain knowledge. In some cases, this domain knowledge was based on "common" knowledge (e.g., desirability of various features of an automobile in a purchasing decision). In many cases, however, determining the networks structure and monotonicity annotations required significant time for researching the domain and reading previous publications concerning the data sets. It is important to note that all the information came from true "knowledge"; in particular we did not examine the data itself.

We hypothesized that monotonicity constraints would prove more helpful at finer discretizations. To test this, for each data set, attributes with numeric values were discretized using Weka's (the Waikato Environment for Knowledge Analysis [Witten and Frank, 2000], version 3.4) equal-frequency discretization tool to generate data sets with numbers of bins 2, 3, and 5, yielding a total of 15 data sets for our experiments. All class variables have two classes. Moreover, all incomplete rows in any of the data sets have been removed.

We had originally chosen ten datasets, but of these, only five had a tractable knowledge-engineered Bayesian network structure. The others had nodes with 8–11 incoming arcs, making the optimization task very difficult, and yielding low performance on all Bayesian network classifiers. We felt it was important for these experiments to fully use our domain knowledge, so we chose to omit datasets whose "true" knowledge engineered structure was intractable rather than to adjust the structure (say, by reversing arcs) for computational reasons.

Note that because our data is fully observed, only nodes in the Markov blanket have an effect on predicting the class. This means that for some of our experiments, the knowledge engineered structures ignore certain features.

5.2.1 Automobile MPG Data Set

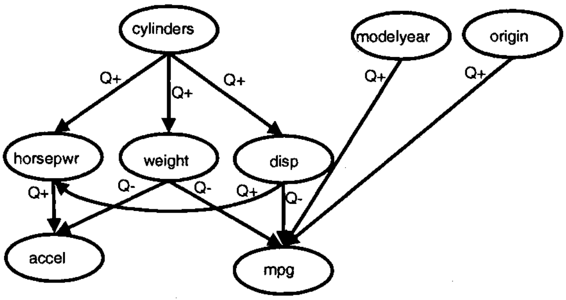


FIGURE 5.3: Annotated network structure for auto-mpg data set

Figure 5.3 shows the KB structure and monotonicity constraints for data set auto-mpg. In this data set, the classification problem is to predict whether a car has low ($\leq$ 28) or high ($>$ 28) mileage per gallon (mpg). The auto-mpg data set has 392 instances of which 106 are labeled positive examples. Domain knowledge suggests that an increase in the number of cylinders (cylinders) usually leads to an increase in horsepower (horsepwr), displacement (disp), and vehicle weight (weight). An increase in weight leads to a decrease in mpg. The heavier the vehicle, the slower it accelerates (accel). The larger the displacement, the greater the horsepower. However, large displacement also means low mileage per gallon. Finally, newer models (modelyear) tend to be more fuel-efficient, as do vehicles imported for sale in the United States from (origin) Japan (encoded as 1), as opposed to vehicles imported from Europe or vehicles manufactured and sold in the United States (encoded as 0). These monotonicity relations are encoded as constraints in the network as shown in Figure 5.3.

5.2.2 Pima Indian Diabetes Data Set

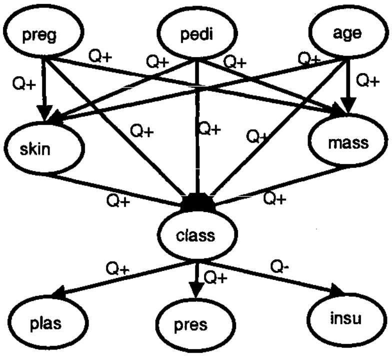


FIGURE 5.4: Annotated network structure for pima-indian-diabetes data set

Figure 5.4 shows the KB structure and monotonicity constraints for pima-indian-diabetes. The problem for this data set is to classify individuals that tested positive for diabetes. This data set has 768 instances of which 268 are labeled positive. Domain knowledge suggests that an increase in each of the triceps' skin fold thickness (skin) is expected with an increase in the number of experienced pregnancies (preg), an increase in age (age), and perceived risk due to pedigree (pedi). The same monotonic relations are also suggested in body mass index (mass). An increase in preg, pedi, age, skin, or mass increases the risk of diabetes (class). Most diabetics have high levels of plasma glucose concentration (plas) and most suffer from high blood pressure (pres) while having low levels of insulin (insu).

5.2.3 Breast Cancer Wisconsin Data Set

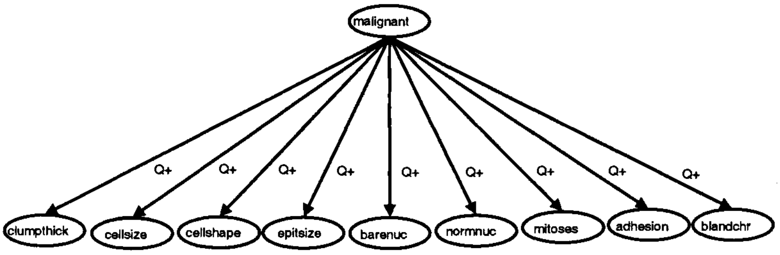


FIGURE 5.5: Annotated network structure for breast-cancer-wisconsin data set

The KB structure and monotonicity constraints for breast-cancer-wisconsin are shown in Figure 5.5. The classification problem is to predict whether a given example is malignant (malignant) or benign. The data set breast-cancer-wisconsin has 683 examples, of which 239 are positive. The attributes in the database have been assigned values that range from 1 (normal state) to 10 (most abnormal state). The attributes are: clump thickness (clumpthick), uniformity of cell size (cellsize), uniformity of cell shape (cellshape), single epithelial cell size (epitsize), bare nuclei (barenuc), normal nucleoli (normnuc), mitoses, marginal adhesion (adhesion), and bland chromatin (blandchr). These attributes have been visually assessed using fine needle aspirates taken from patients' breasts. Malignant samples have observed abnormal states, i.e., the more the malignant a sample the higher the state of abnormality. Hence, all network links from malignant to other attributes have $\overset{Q+}{\succ}$ monotonicity constraints.

5.2.4 Haberman Survival Data Set

Figure 5.6 shows the KB structure and monotonicity constraints for haberman. The problem in this data set is to predict the survival status of a patient who has undergone breast cancer surgery. haberman has 306 instances of which 225 are positive

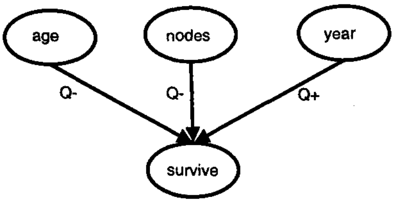


FIGURE 5.6: Annotated network structure for haberman data set

examples. The data set has three attributes: the age of patient at time of operation (age), the patient's year of operation (year), and the number of positive axillary lymph nodes detected (nodes). We expect the survivability of the patient to decrease as the patient gets older, to decrease as the number of positive nodes detected increases, and increase with the operation year, i.e., more recent implying better survival.

5.2.5 Car Acceptability Data Set

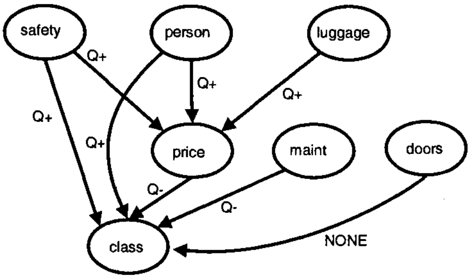


FIGURE 5.7: Annotated network structure for car data set

Figure 5.7 shows the relevant information for the car data set. The prediction problem in this data set is to determine whether a given instance is acceptable (class) given the following attributes: (price), cost of maintenance (maint), capacity in number of persons (person), size of luggage space (luggage), estimated safety rating (safety), and number of doors (doors). car has a total of 1728 examples of which 30% are positive. This data set has non-numeric attributes and for uniformity and clarity we re-encoded their respective values as numbers, in their respective ordinal order. Common knowledge about car buying preferences suggests that as price and maintenance costs increase, acceptability should decrease. Increases in the safety rating and passenger capacity would generally increase acceptability. It is not clear whether the number of doors plays a significant role in a car's acceptability, but here we assume that it does not. Also, an increase in the cost of making safe cars, an increase in passenger capacity, and an increase in luggage space would normally lead to an increase in price. These monotonicity relations are encoded in the network shown in Figure 5.7.

5.3 Experimental Setup / Details of Implementation

Since the focus of this research was on whether or not knowledge based constraints on a machine learning algorithm can improve its performance, not on efficient techniques for numerical optimization, we first tried simple optimization algorithms. Our original implementation used a simple steepest-direction gradient ascent with decreasing step sizes. This proved too slow, so we tried implementing a golden-ratio line search for ascending to the peak in the steepest direction. Unfortunately this was not sufficiently stable or efficient enough to be generally applicable either. This should perhaps not be surprising; the dimensionality of the parameter space for the largest models in our experiments was several thousand.

We therefore selected an implementation of the L-BFGS[1] algorithm from the conditional random fields package Mallet [McCallum, 2002] to optimize our penalized objective function. To ensure convergence in the feasible region, the L-BFGS maximization was wrapped in the outer-level algorithm given in Figure 5.8.

[1] Limited-memory BFGS, a variation of the *Broyden-Fletcher-Goldfarb-Shanno* algorithm. See for example [Press et al., 1988, pg. 324].

1. Initialize the μ_{ijk} parameters at the unconstrained MAP point (found simply by counting the observations, with a Laplace correction)
2. If this point satisfies the constraints, return it
3. Otherwise, initialize a weight w for the penalty functions
4. Take steps in the steepest direction of the penalized likelihood until convergence
5. If we converged outside the feasible region, increase the penalty weight and repeat the previous step.

FIGURE 5.8: Constrained optimization algorithm

In addition, L-BFGS would sometimes fail to converge, or simply fail to continue from certain points in the parameter space. We suspect that these problems were due to the relatively sharp edges at the constraint boundary. To work around this, upon failure we slightly increased the penalty weight w (just to change the shape of the function) and re-ran the maximizer routine (perhaps many times). This proved sufficient for many of our experiments, though we still had to be careful not to set our ε margin too high, since it would make the feasible region so small that it might never be found. We also experimented with variations—for example, using cubed violations for penalty functions rather than squared, or using a very small exponent (such as 1.01). None of the variations resulted in significantly more reliable estimation routines.

For running experiments, we integrated the learning algorithm with the Weka package, which allowed us to easily script learning runs and to run comparisons against other learning algorithms. The algorithms we analyzed were:

Zero-Regression (ZR) Always picks the mode of the observed distribution of the class variable, without regard to the features.

Naïve Bayes (NB) Also known as the simple Bayesian classifier (SBC). Treats the class variable as the parent in a Bayesian network, with all features as children.

Knowledge-based Bayes (KB) Fit the parameters of a Bayesian network whose structure incorporates domain knowledge—specifically, the Bayesian network structures shown in Figures 5.3-5.7. Parameters are fit by maximum likelihood with a Laplace correction.

Constrained Knowledge-based Bayes (CKB) Same as KB, except that the parameters are fit to maximize the posterior probability subject to the inequality constraints induced by the qualitative monotonicity statements. CKB was run with three different margins, $\varepsilon \in \{0.0, 0.1, 0.2\}$. These runs are designated CKB0, CKB0.1, and CKB0.2.

The ZR, NB, and KB learning algorithms from Weka fit parameters to data almost instantaneously. Our learning algorithm for the constrained networks fit most of the data sets and networks in a matter of seconds or minutes (on a typical desktop workstation class machine). A few (particularly, high-dimensional problems with strong constraints) took hours to fit.

5.4 Experimental Methodology

To compare the algorithms on each data set, we first randomly split the data set into a test set (1/3 of the data) and a training set pool (2/3 of the data), stratified by class. The smallest training set pool had approximately 50 instances; most had several hundred.

We then performed 50 replications for each training set size m, for various m, specifically, m from 1 to 10 by 1's, from 10 to 20 by 2's, and from 20 to 50 by 5's. In each replication, we randomly drew m elements without replacement from the training set pool, and trained our algorithms on the set. The resulting fitted networks were then evaluated on the test set.[2]

Chapter 6:

Results and Discussion

The results of evaluating the trained networks on the test set are shown in Figure 6.1. The figure contains 15 charts in three columns for the three different discretizations and five rows for the five different data sets. The learning curves are also re-printed at larger sizes in Figures 6.2-6.6. We also present in Figure 6.7 the results of running McNemar's test on the NB, KB, CKB0 and CKB0.1 classifiers trained on the pima data set.

6.1 Learning from Small Samples

Statistical models fit from small samples tend to suffer from poor performance due to high variance. Simpler or more restricted models tend to be less susceptible to this problem. We hypothesized that by adding monotonicity constraints to the model, we would simplify it and reduce its variance. Assuming the constraints are correct (that is, they are consistent with the data in general, and in particular, the unseen test data), this should improve generalization performance. Furthermore, stronger constraints should yield better performance.

Based on this theory, we hypothesized the following ranking in performance for small sample learning, from worst to best: (1) Zero regression (ZR), (2) Naïve Bayes (NB), (3) Knowledge-based networks (KB), (4) Combined knowledge-based structure and monotonicity constraints (CKB).

[2] With this number of data sets, training set sizes, and repetitions, we had approximately 25,000 learning runs which were submitted to a cluster of about 100 typical desktop workstation class machines, which processed them overnight.

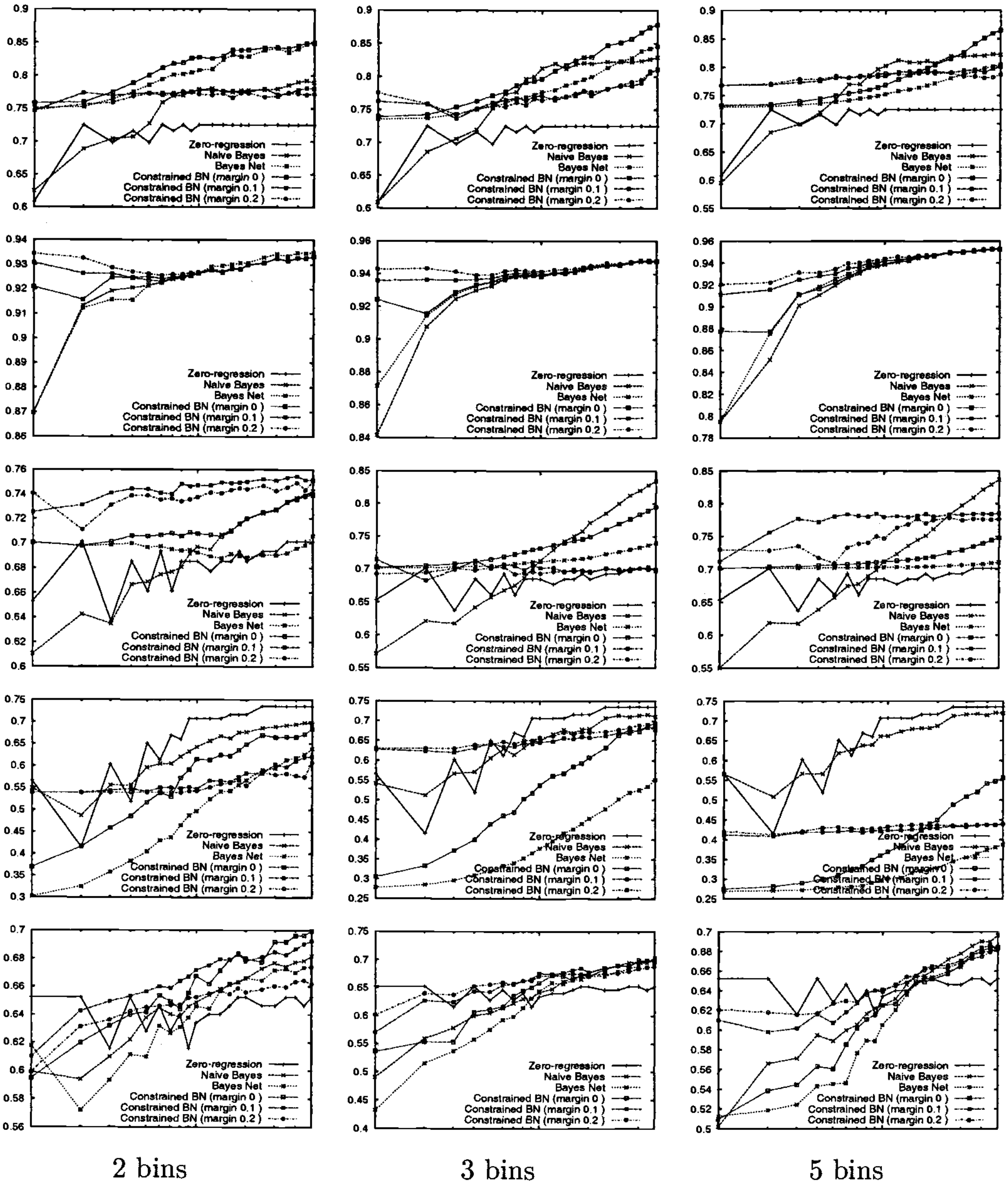


FIGURE 6.1: Learning curves for auto, bcw, car, haberman, and pima domains at 3 discretizations, plotting average accuracy (across 50 runs) against training set size (log scale, 1 through 50). Zero-regression is omitted on the bcw data set as it had performance far below the other algorithms.

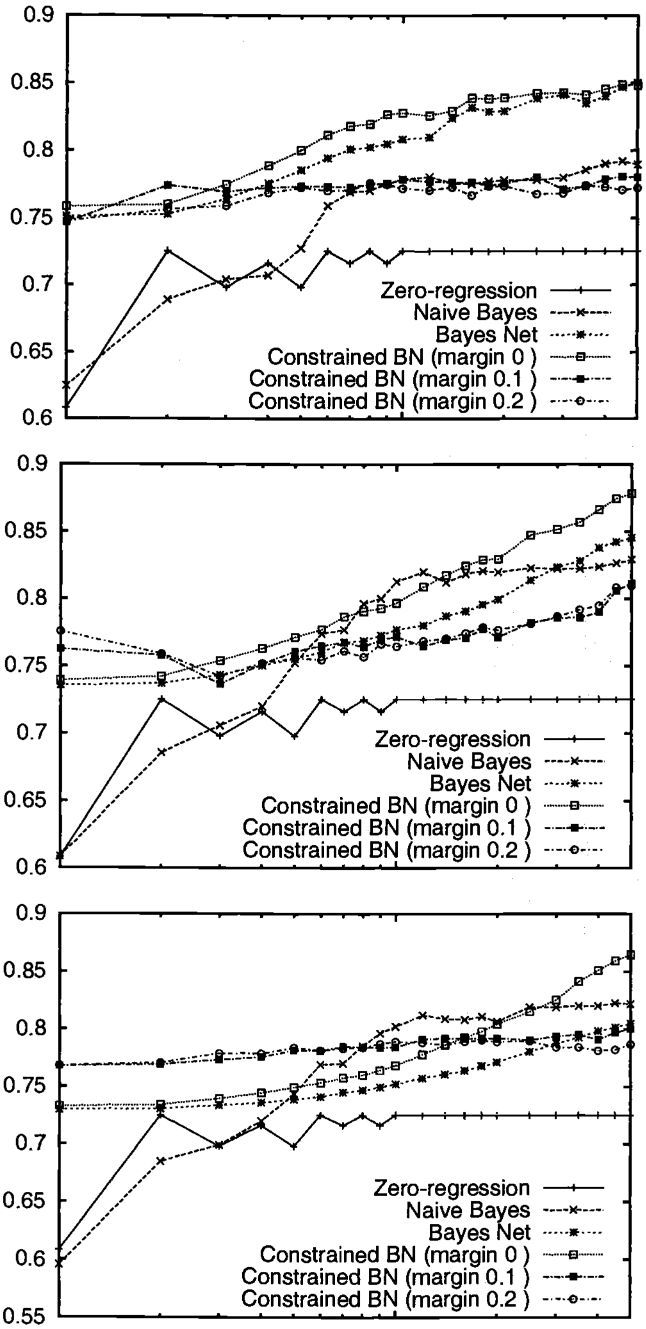


FIGURE 6.2: Learning curves for auto at 3 discretizations, plotting average accuracy (across 50 runs) against training set size (log scale, 1 through 50).

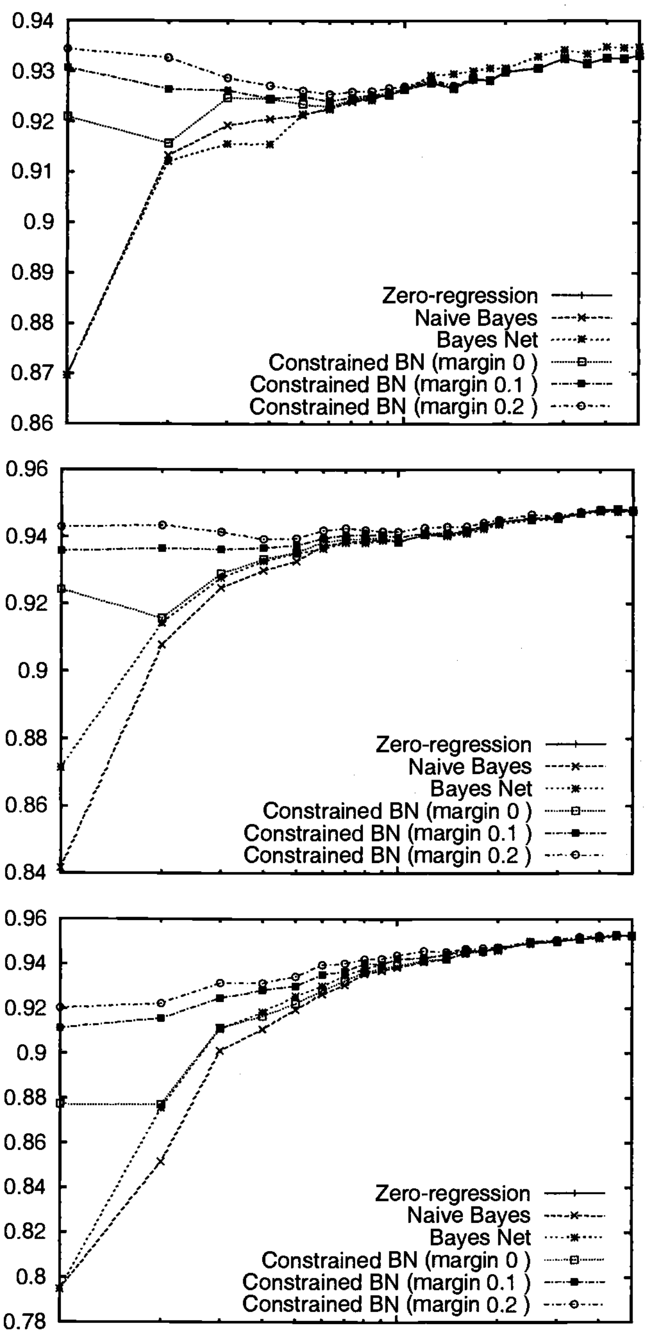


FIGURE 6.3: Learning curves for bcw at 3 discretizations, plotting average accuracy (across 50 runs) against training set size (log scale, 1 through 50). Zero-regression is omitted as it had performance far below the other algorithms.

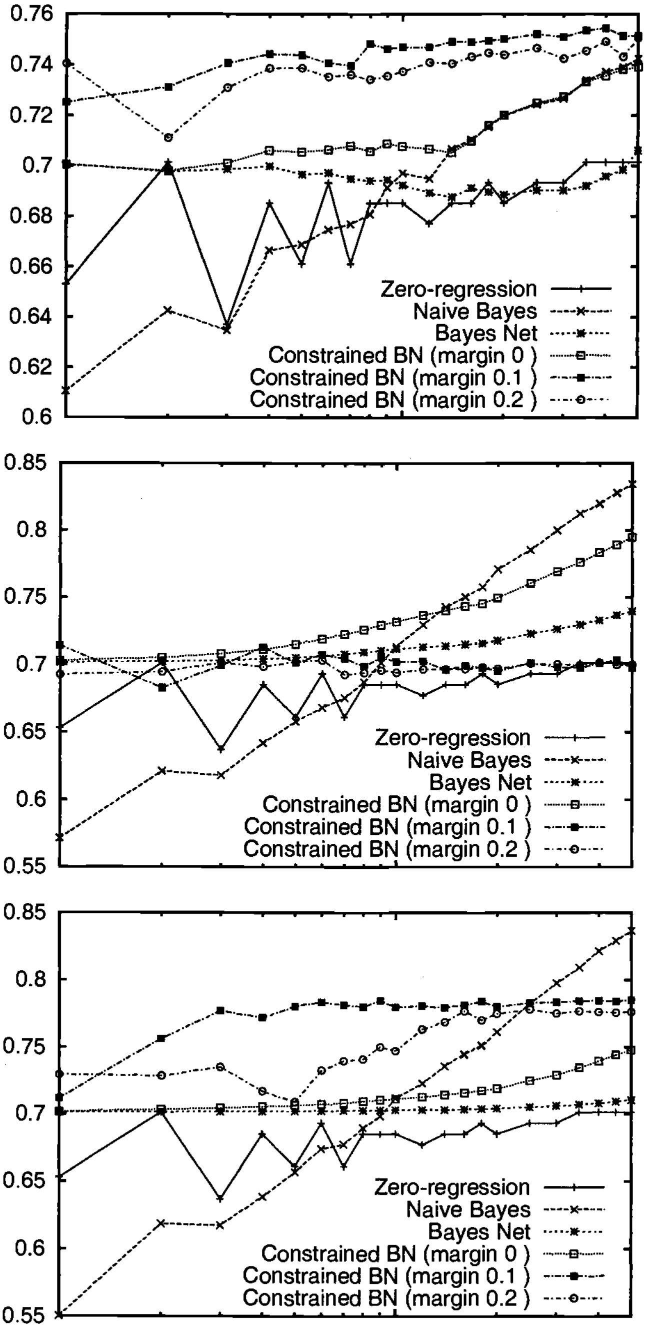


FIGURE 6.4: Learning curves for car at 3 discretizations, plotting average accuracy (across 50 runs) against training set size (log scale, 1 through 50).

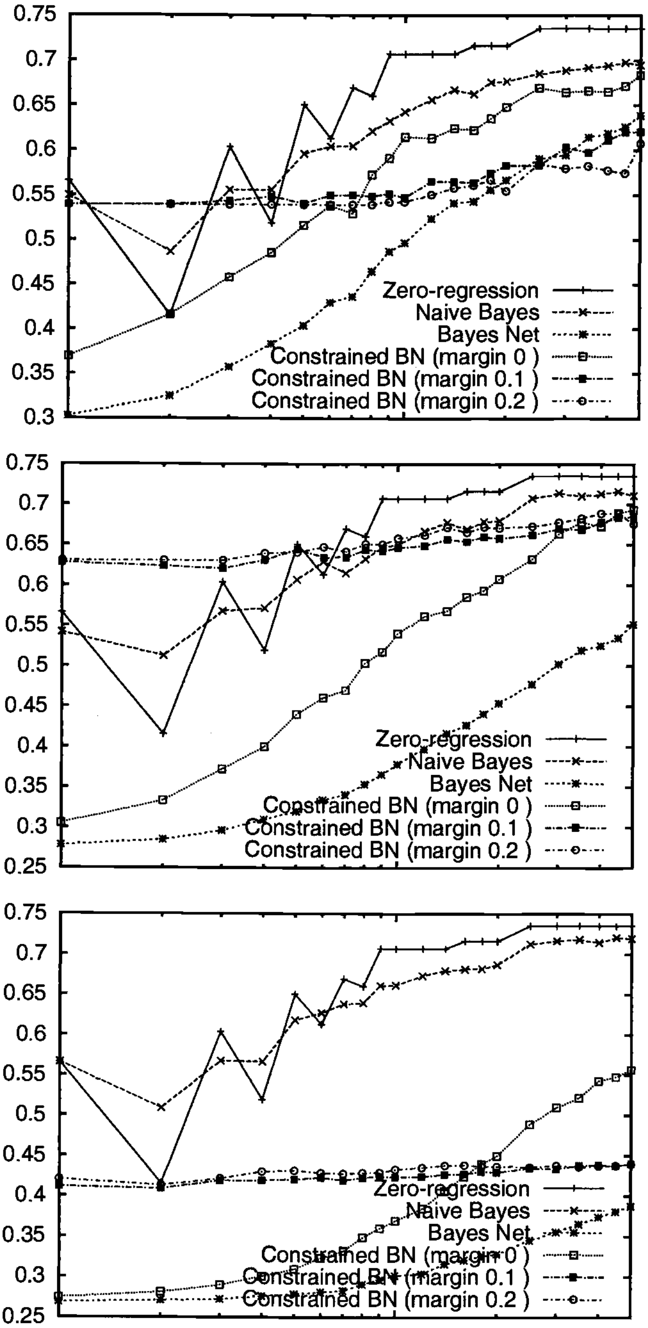


FIGURE 6.5: Learning curves for haberman at 3 discretizations, plotting average accuracy (across 50 runs) against training set size (log scale, 1 through 50).

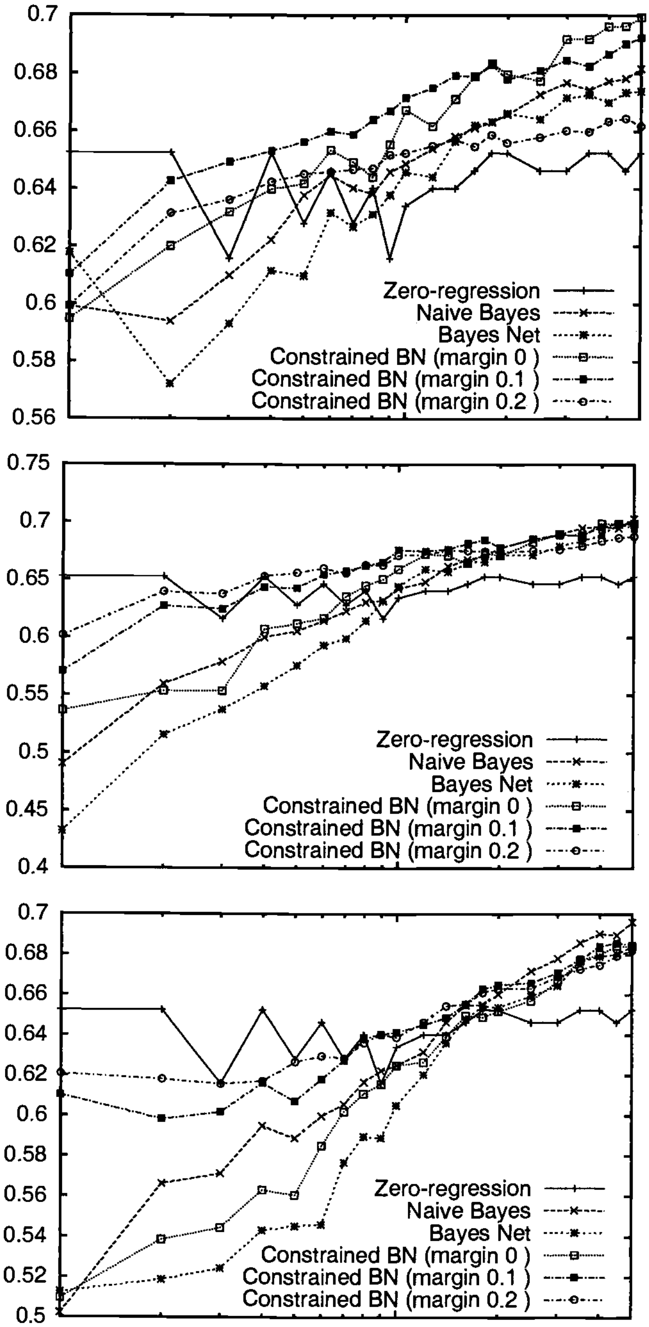


FIGURE 6.6: Learning curves for pima at 3 discretizations, plotting average accuracy (across 50 runs) against training set size (log scale, 1 through 50).

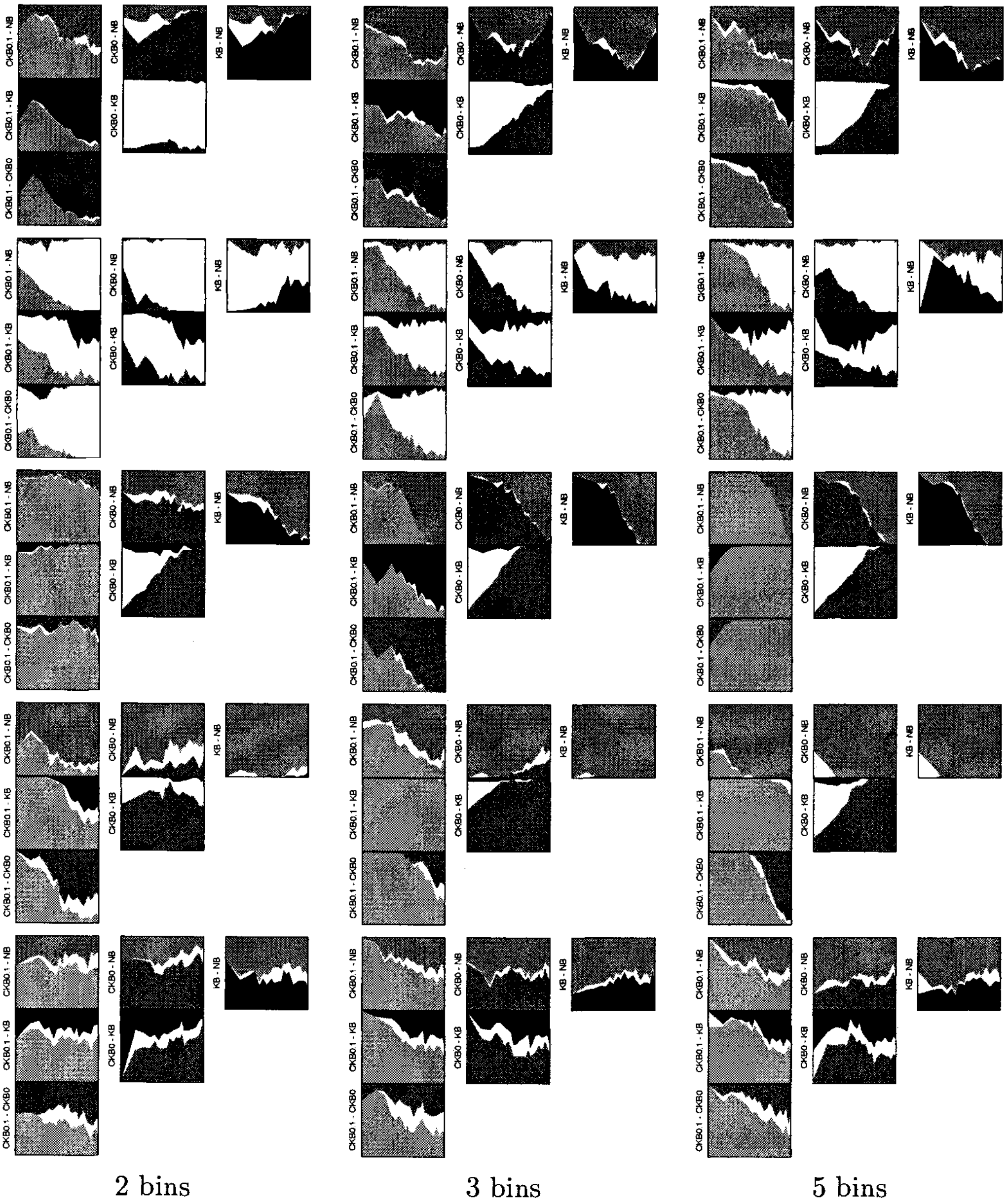


FIGURE 6.7: McNemar's test for auto, bcw, car, haberman, and pima domains (5 rows) at 3 discretizations (columns), comparing pairs of algorithm from CKB0.1, NB, CKB0, and KB (in order from lightest to darkest shade). The 6 pairwise comparisons are run at training set sizes 1 through 50 (x axis, log scale). The lower and upper regions represent the number of statistically significant wins of each algorithm, with the remaining center region indicating ties or statistically insignificant wins.

The plots show that actually, ZR performs surprisingly well: comparable to or better than NB at small sample sizes on all data sets. Furthermore, on haberman, NB and ZR dominate, even on small samples. Otherwise, we do see the expected ranking, though at small sample sizes, ZR and NB frequently tie, as do KB and CKB.

Since our largest training set was of size 50, which we still consider relatively small for models of the complexity used here, we actually expected to see this ranking extend through more of the tested sample sizes. One somewhat surprising result was how well NB performed on car at higher sample sizes with discretizations finer than 2 bins. We were also particularly surprised by the results on the haberman data set, where NB and ZR did very well all the way through $m = 50$. A simple data analysis on the haberman data (2-bin) using correlation and mutual information reveals that the data set exhibits independence between the class variable and any one of the three parent attributes. Moreover, the conditional probability tables reveal that the parameters do not exhibit monotonicity, e.g., the chances of surviving given that the patient is young is high but surprisingly the data also says that the chances of surviving given that the patient is old is also high. Clearly, our assumptions about the structure and monotonicities of this data set were incorrect.

6.2 Learning from Large Samples

We expect our constraints to be useful for learning from small sample sizes, since they reduce the variance of the fitted model. Conversely, when learning statistical models from very large samples, we presumably have sufficient data that we wish to learn what is actually in the data. We might expect that enforcing the prior constraints, especially when the constraints are incorrect or overly strong (e.g., $\varepsilon = 0.2$), would bias the model away from a good fit, and hurt average generalization. On the other hand, if the monotonicities are correct, the learning algorithm should still generalize well even when trained on large samples.

The plots do show flatter learning curves for CKB with $\varepsilon > 0$, compared to CKB with no margin, indicating that strongly enforcing the margins introduces bias—

that is, the true monotonic effects are not as strong as would be suggested by the margin. This difference is also clearly shown in the McNemar's test comparisons between the two.

It is worth noting, however, that without margins, CKB is comparable to or better than KB at nearly all tested sample sizes. This shows that although nonzero margins may be too strong, the base monotonicity constraint is generally in accordance with the data.

There is another way of looking at the zero-margin CKB algorithm: as a form of context-specific per-CPT feature selection. Data violating a zero-margin monotonicity constraint for a particular parent results in CPT rows which do not differ. Thus, we learn to ignore that parent (feature) in contexts in which its effect appeared from the data to violate the monotonicity.

6.3 Learning with Finer Ordinal Discretizations

The fineness of discretization affects the model in a number of ways. Two in which we were particularly interested are: (1) the increase in the number of parameters of the model, and (2) the expected increase in the strength of the monotonic relation.

It is easy to see the number of parameters (CPT cells) increases with finer discretizations. With a greater number of parameters comes higher variance in the fitted model. However, in the monotonically constrained models, we expected this increased model complexity to be somewhat balanced by an increase in the strength of the constraints.

Consider the parameter space for the model $X \overset{Q+}{\succ} Y$, for both variables binary. As shown in Figure 3.1, we have two parameters (p_0 and p_1) and the constraint $p_0 \geq p_1$. This constraint reduces the parameter space to 1/2 its original volume. Now consider discretizing X at three levels. We then obtain three parameters (p_0, p_1 and p_2) and the constraints $p_0 \geq p_1$ and $p_1 \geq p_2$. It is easy to compute that this reduces the parameter space to 1/3 its original volume. In general, constraints reduce a larger proportion of the volume of the parameter space for finer discretizations than for coarser ones. Thus we have a trade-off between number of parameters and the strength of the constraints on those parameters.

Given this, we hypothesized that the monotonicity constraints would help more at finer discretizations. The plots show some support for this: on auto, there is little difference between CKB0 and KB at the 2-bin discretization; at higher discretizations and with more training data, CKB0 dominates KB. CKB0.1 on auto shows very good performance at the 5-bin discretization level. On bcw, all algorithms perform about the same at high sample sizes, though CKB0.1 does well at low sample sizes, and this effect is amplified at finer discretizations. Finally, looking only at lower sample sizes, we observe this effect on the pima data set. The results from the remaining data sets do little to support or disprove this hypothesis.

Chapter 7:
Conclusion and Further Work

In this work we have explored possibilities for formalizing high-level qualitative domain knowledge regarding the nature of interactions between random variables. Our formalization is simple and intuitive for domain experts, and translates easily into constraints on the parameter space of the models. We have in particular focused on monotonicity, and shown experimentally that monotonicity constraints are useful, especially in sparse-data situations.

Still, there is significant work yet to be done.

This document outlines synergies, strength of influence, and saturation ideas with definitions in terms of CPT constraints. However, we have not evaluated these constraints experimentally. It would be good to know whether they are useful for generalization performance, and if so, in what situations.

From a practical standpoint, our gradient descent optimization routine is probably far from ideal in terms of efficiency and stability. One possible research topic would be to investigate a re-parameterization that would allow us reduce the number of parameters and/or simplify the form of the constraints. Even more desirable would be a closed-form algorithm for finding the active constraints, so we could use the method of Lagrange multipliers.

Another practical issue is that with full conditional probability tables, we cannot effectively learn given knowledge-based structures mandating large numbers of parents. The five data sets we used in our experimental work were precisely those which had at most five parents for any given node. We had five other data sets which had more parents—sometimes over ten. We did not want to adopt a structure in conflict with our domain knowledge, and so we were forced to ignore these data sets. In future work we plan to integrate our qualitative constraints with techniques for dealing with this "many-parents problem", in particular, noisy-or, noisy-max, or decision tree representations of the conditional probability distribution. An early exploration in this direction can be found in [Natarajan et al., 2005].

It would be valuable to extend our definitions of qualitative statements to joint probability tables. Some conditional independency structures are more compactly represented in a Bayesian network with conditional probability tables, while some are more compactly represented in a Markov network with undirected relations and potential functions. Perhaps more importantly, when causality is not obvious, it is more natural and comfortable for a domain expert to model an undirected "association" link rather than a Bayesian network directed arc with its connotations of causality. Ideally, our models should support "chain graphs" (mixed graphs of directed and undirected arcs). Note that much of the existing statistics literature on monotonicity (as discussed in Chapter 8) covers joint distributions; this may provide direction for further research.

Another direction for possible extension is towards continuous variables, and the learning of parameters for functional forms of probability distributions. Monotonicities are in general much easier to enforce in continuous models (consider linear regression).

There is currently wide-spread interest in machine learning for relational domains [Getoor et al., 2001, Heckerman, 2004, De Raedt and Kersting, 2003]. Improved performance is one frequently cited reason for adopting relational models, but another, which from our perspective is equally important, is that relational models are closer to the way domain experts think. By expressing features and influences with a relational language (see for instance [Altendorf and D'Ambrosio, 2004]), we narrow the gap from a domain expert's knowledge to the formalized model. This fits precisely within the scope of our goals for knowledge-intensive machine learning. Combining qualitative constraints with relational modeling tools is an essential next research step.

Chapter 8 discusses a small but hopefully representative sample of existing statistics literature about representing monotonicities in statistical models. There are undoubtedly valuable ideas from this body of work which we have not exploited. In particular, it would be valuable to compare various notions of monotonicity. These comparisons could take two different paths: (1) experimental comparisons judging usefulness in terms of generalization performance and (2) philosophical analysis judging concordance with domain expert's intuition (or intuitions) of "monotonicity".

Chapter 8:
Ordinal Data Analysis in Statistics

This chapter serves as an appendix of sorts. We here devote further attention to comparing traditional statistical approaches to estimating model parameters given categorical ordinal data. This is not intended to be a reference work of its own, but merely an introduction and overview with pointers to true reference works.

8.1 General Relation Between Joint and Conditional Constraints

Much of the discussion of monotonicity in the literature on categorical data analysis focuses on discrete probability distributions with joint probability tables. Since our work is primarily concerned with conditional distributions, we begin by discussing the relation between constraints on joint and conditional distributions.

A property or constraint C of a conditional distribution $P(Y \mid X)$ can be transformed to a property or constraint C' of the joint distribution $P(X, Y)$. This is because given a joint distribution, it is simple to calculate the marginal $P(X)$ and thus also the conditional $P(Y \mid X) = P(X,Y)/P(X)$. Thus we may apply our constraints to the conditional, and the joint is appropriately constrained, with remaining additional degrees of freedom corresponding to the unconstrained marginal $P(X)$.

It is not clear under what, if any, conditions it is possible to factor a constraint on a joint distribution to individual constraints on the marginal of one variable and the conditional of the other.

8.2 Models for Categorical Data

There are a wide range of models for categorical and ordinal data other than the traditional joint and conditional probability tables used in Bayesian networks. In this section we review some of the more popular models from the statistics literature.

We begin by introducing some notation and definitions. First, it is common in the statistics literature to write $P(X = x_i, Y = y_j)$ as π_{ij}. The indices i and j are bounded by I (number of rows) and J (number of columns) respectively. Second, many of the models and much of the monotonicity analysis will make use of odds ratios (in various forms). The simplest form of an odds ratio is defined for a 2 × 2 table. Supposing the entries are $\{\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22}\}$, the odds ratio is defined to be $\theta = (\pi_{11}\pi_{22})/(\pi_{12}\pi_{21})$. Other forms of odds ratios are usually formed by slicing or collapsing a larger table to 2 × 2 in various ways (see, for example, the discussion in Section 3.7.1).

8.2.1 Saturated and Independence Models

A *saturated* model is one which has a sufficient number of parameters to explain the data exactly—for example, a table of probabilities with one entry for each possible outcome. Although this is the type of model most frequently used in Bayesian networks, statisticians tend to look for simpler models with fewer parameters.

For a pair of variables X and Y, we have at the other extreme the *independence* model, which assumes no interaction between the two modeled variables. In this model, their joint is the product of their marginals, typically denoted as vectors with components whose logs are λ_i^X and λ_j^Y (and thus, $\sum_i \lambda_i = 0$). The joint can then be expressed as

$$\pi_{ij} = \exp\left(\mu + \lambda_i^X + \lambda_j^Y\right). \tag{8.1}$$

Of course, the independence model cannot capture any correlation or dependence and is generally of limited use. We now investigate other models which lie between these two extremes.

8.2.2 Linear-by-Linear Association

The *linear-by-linear association model* is a joint model which assumes the form [Agresti, 1996, pp. 146, 182]

$$\pi_{ij} = \exp\left(\mu + \lambda_i^X + \lambda_j^Y + \beta\mu_i\nu_j\right), \tag{8.2}$$

where μ_i and ν_i are "scores", and β is a parameter which indicates direction and strength of influence, with $\beta = 0$ corresponding to independence.

The model is called "linear-by-linear" because the deviation from independence is linear in the score parameters for one random variable given a fixed value of the other.

In the case that $\mu_{i+1} - \mu_i$ is a constant for all i and $\nu_{j+1} - \nu_j$ is a constant for all j, the linear-by-linear association model is called a *uniform* association model, and β equals the local log odds ratio everywhere (see Section 8.3.1).

For a significantly more detailed coverage, see [Agresti, 1996, Chapters 6,7] or [Agresti, 1990, Chapter 8.1], which also discusses maximum-likelihood fitting of such models.

8.2.3 Cumulative Logits and the Proportional Odds Model

The *logit* is a transformation defined as $\text{logit}(\alpha) = \ln[\alpha/(1-\alpha)]$. In ordinal models, we have the *cumulative logits*, defined as the logits of the cumulative probability distributions. Specifically, we have

$$\text{logit}[P(Y \leq y_j)] = \ln\left(\frac{P(Y \leq y_j)}{1 - P(Y \leq y_j)}\right) = \ln\left(\frac{\pi_1 + \ldots \pi_j}{\pi_{j+1} + \ldots \pi_J}\right) \qquad (8.3)$$

for all j. The distributions can be joints or conditionals, and we denote them respectively as

$$L_j = \text{logit}[P(Y \leq y_j)] \qquad \text{and} \qquad L_j(x) = \text{logit}[P(Y \leq y_j \mid X = x)]. \qquad (8.4)$$

This provides additional options for parameterizing the model. Specifically, for a conditional distribution $P(Y \mid X)$, statisticians often define the *proportional odds model*, in which the cumulative logits for each j are assumed to have the form

$$L_j(x) = \alpha_j + \beta x. \qquad (8.5)$$

This formulation is conducive to log odds ratio analysis. If we consider collapsing the discretization of Y to binary at the threshold defined by j, the odds at a particular value x_{i_1} are

$$\frac{P(Y \leq y_j | x_{i_1})}{1 - P(Y \leq y_j | x_{i_2})} = \exp\left(L_j(x_{i_1})\right). \qquad (8.6)$$

Thus the log odds ratio for two values of the conditioning variable X, say x_{i_1} and x_{i_2}, in the collapsed model is given by

$$\begin{aligned} \ln\left(\frac{\exp(L_j(x_{i_2}))}{\exp(L_j(x_{i_1}))}\right) &= L_j(x_{i_2}) - L_j(x_{i_1}) \\ &= (\alpha_j + \beta x_{i_2}) - (\alpha_j + \beta x_{i_1}) \\ &= \beta(x_{i_2} - x_{i_1}). \qquad (8.7) \end{aligned}$$

This value is proportional to the distance between x_{i_1} and x_{i_2} (hence the name *proportional odds model*). This means that it is only sensible for X which are interval-measured. In practice, this type of model is useful for interval discrete as well as continuous conditioning variables. For further discussion and variations on logit models, see [Agresti, 1996, Chapters 8.2, 8.3] or [Agresti, 1990, Chapters 9.2, 9.3].

8.2.4 Concordance, Discordance, Gamma, and Yule's Q

Given a sample of data $D = \{(x_1, y_1), \dots (x_n, y_n)\}$, one way to measure its monotonicity is to count the pairs of examples which agree with the monotonicity (are concordant) and the number which disagree (are discordant). A pair of examples $(x_i, y_j), (x_k, y_l) \in D$ is *concordant* if $(x_i < x_k \wedge y_j < y_l) \vee (x_i > x_k \wedge y_j > y_l)$, and *discordant* if the inequalities are reverse, and *tied* if for any of the conditions strict inequality does not hold. We denote the number of concordances and discordances as C and D. Our *sample-based gamma* statistic is then defined by $\hat{\gamma} = (C-D)/(C+D)$.

For a fitted model or probability distribution, we define the probabilities of concordance and discordance as Π_c and Π_d, respectively, and we have the standard *gamma* statistic is defined to be the difference between the probability of concordance and discordance $\gamma = (\Pi_c - \Pi_d)/(\Pi_c + \Pi_d)$.

If X, Y are binary, our table is 2×2 with 4 entries $\{\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22}\}$, and the gamma statistic reduces to a statistic called *Yule's Q*, defined to be

$$Q = \frac{\pi_{11}\pi_{22} - \pi_{12}\pi_{21}}{\pi_{11}\pi_{22} + \pi_{12}\pi_{21}}. \tag{8.8}$$

This can also be viewed as a transformation of the odds ratio $\theta = (\pi_{11}\pi_{22})/(\pi_{12}\pi_{21})$ into the range $[-1, +1]$, where $Q = (\theta - 1)/(\theta + 1)$. Note that θ here is an odds ratio, as opposed to a CPT entry (as we have used it throughout the other chapters of this thesis).

See [Agresti, 1990, Sec. 2.3] for a more detailed treatment.

8.3 Notions of Multivariate Dependence

As mentioned in this work, first-order stochastic dominance monotonicity is not the only possible definition of monotonicity. A number of authors have discussed various possible operationalizations and proved implications of one to the other. With the background provided by the previous section, we will now discuss these possible definitions. In Section 8.4 we discuss their relationships to each other.

Some definitions due to Alam and Wallenius [Alam and Wallenius, 1976] have been adapted for our purposes. Specifically, we have simplified them to cover only bivariate discrete distributions.

8.3.1 Local Log Odds Ratios (Agresti, etc.)

The local odds ratio at i, j is the odds ratio of the 2×2 table whose least-index corner is at i, j. Specifically, we write

$$\theta_{ij} = \frac{\pi_{i,j}\pi_{i+1,j+1}}{\pi_{i,j+1}\pi_{i+1,j}}. \tag{8.9}$$

Agresti and Chuang propose [Agresti and Chuang, 1985, pg. 8] the uniformity of the sign of the local log odds ratio as a possible operationalization of monotonicity. More formally, the definition for this property states that X and Y vary isotonically if for all i, j we have $\ln \theta_{i,j} \leq 0$ (and antitonically if the inequality is reversed). Following Lehmann and others who have analyzed this property, they call it *positive* (or *negative*) *likelihood-ratio dependence.* This is a stricter notion than our FSD monotonicity, and is defined on a joint distribution rather than a conditional distribution. They also discuss a Gaussian prior on the odds ratios, and present techniques for both MLE and MAP inference under the hard sign constraint as well as the soft Gaussian prior [Agresti and Chuang, 1985].

8.3.2 Monotonicity in a Parameter (Alam and Wallenius)

If the distribution of Y depends on a real parameter θ, then Y is *stochastically increasing in θ* if for all y, $P_\theta(Y > y)$ is nondecreasing in θ.

A distribution has *monotone likelihood ratio* if $y_1 < y_2$ and $\theta_1 < \theta_2$ implies that $P_{\theta_1}(y_1)P_{\theta_2}(y_2) - P_{\theta_1}(y_2)P_{\theta_2}(y_1) \geq 0$. (Interestingly, this is the supermodularity of the function $g(y, \theta) = P_\theta(y)$, suggesting an intuition of synergy between the parameter and the random variable. Actually, all definitions of monotonicity based on likelihood ratios or log odds ratios can be expressed in terms of the modularity of a function.)

8.3.3 Monotonicity in a Random Variable (Alam and Wallenius)

Y is *stochastically increasing in* X if for every y, $P(Y > y \mid X = x)$ is nondecreasing in x. This is denoted $Y \uparrow \mathtt{st}\ X$ [Alam and Wallenius, 1976, Barlow and Proschan, 1975]. This definition is identical to our notion of FSD monotonicity.

Y is *positive likelihood ratio dependent on* X if A distribution has *monotone likelihood ratio* if $x_1 < x_2$ and $y_1 < y_2$ implies that $P(y_1 \mid x_1)P(y_2 \mid x_2) - P(y_1 \mid x_2)P(y_2 \mid x_1) \geq 0$. This is denoted $Y\ \mathtt{plrd}\ X$ [Alam and Wallenius, 1976, Dykstra et al., 1973]. We have already seen this definition, though we repeat it here for completeness.

8.3.4 Monotonicity in a Joint Distribution (Alam and Wallenius)

X and Y are *s^*-positively dependent* if $X \uparrow \mathtt{st}\ Y$ and $Y \uparrow \mathtt{st}\ X$. Furthermore, X and Y are *m^*-positively dependent* if $X\ \mathtt{plrd}\ Y$ and $Y\ \mathtt{plrd}\ X$.

8.4 Relationships and Discussion

8.4.1 m^* and s^*-Positive Dependence

Alam and Wallenius [Alam and Wallenius, 1976] claim it is easy to show that m^*-positive dependence implies s^*-positive dependence, and that the converse does not hold.

8.4.2 Properties and Relationships of the Local Log Odds Ratios

The first property is that uniform sign of all local log odds ratios implies uniform sign of "dispersed" log odds ratios (that is, the log odds ratios for the 2 × 2 tables formed by the cells in the intersection of rows j, l and columns i, k). Suppose two adjacent odds ratios

$$\theta_{i,j} = \frac{\pi_{i,j}\pi_{i+1,j+1}}{\pi_{i,j+1}\pi_{i+1,j}} \qquad \text{and} \qquad \theta_{i,j+1} = \frac{\pi_{i,j+1}\pi_{i+1,j+2}}{\pi_{i,j+2}\pi_{i+1,j+1}}$$

are each greater than 1. Then, we may show that the dispersed log odds ratio, that is, $\pi_{i,j}\pi_{i+1,j+2}/\pi_{i,j+2}\pi_{i+1,j}$, is greater than 1. This is clear because

$$\begin{aligned}\frac{\pi_{i,j}\pi_{i+1,j+2}}{\pi_{i,j+2}\pi_{i+1,j}} &= \frac{\pi_{i,j}}{\pi_{i+1,j}} \cdot \frac{\pi_{i+1,j+2}}{\pi_{i,j+2}} \cdot \frac{\pi_{i+1,j}}{\pi_{i+1,j}} \cdot \frac{\pi_{i+1,j+1}}{\pi_{i+1,j+1}} \\ &= \frac{\pi_{i,j}}{\pi_{i+1,j}} \cdot \frac{\pi_{i+1,j+1}}{\pi_{i+1,j+1}} \cdot \frac{\pi_{i+1,j}}{\pi_{i+1,j}} \cdot \frac{\pi_{i+1,j+2}}{\pi_{i,j+2}} \\ &= \theta_{i,j} \cdot \theta_{i,j+1} \quad > \quad 1\end{aligned}$$

The second property, which we state without proof, is that in a linear-by-linear association model, positive likelihood ratio dependence is equivalent to an ordering of the score parameters [Agresti and Chuang, 1985, eq. 1.2], that is,

$$\forall i, \mu_i < \mu_{i+1} \qquad \forall j, \nu_j < \nu_{j+1}.$$

The final property, discussed in various forms [Alam and Wallenius, 1976,Barlow and Proschan, 1975] is that under the appropriate transformations to account for conditional versus joint distributions, FSD monotonicity is a strictly weaker property than positive likelihood ratio dependence. We here provide just an example which shows that FSD monotonicity (in a conditional distribution $P(Y \mid X)$) does not imply positive likelihood ratio dependence (in the joint distribution $P(Y \mid X)P(X)$). Consider the conditional probability table

$P(Y \mid X) =$

x	$y=0$	$y=1$	$y=2$
0	0.5	0.4	0.1
1	0.1	0.8	0.1

which exhibits FSD monotonicity, since

$$0.5 \geq 0.1 \quad \text{and} \quad 0.5 + 0.4 \geq 0.1 + 0.8.$$

The marginal for X is unknown, so we define $P(X = 0) = \alpha$, for $0 \leq \alpha \leq 1$, and thus we can write the joint for X and Y as

$P(Y \mid X)P(X) =$

x	$y=0$	$y=1$	$y=2$
0	0.5α	0.4α	0.1α
1	$0.1(1-\alpha)$	$0.8(1-\alpha)$	$0.1(1-\alpha)$

which does *not* exhibit local log odds ratio monotonicity, since

$$\begin{aligned}\ln \theta_{00} &= \ln\left(\frac{0.4\ \alpha\ (1-\alpha)}{0.04\ \alpha\ (1-\alpha)}\right) = \ln 10 > 0\\ \ln \theta_{01} &= \ln\left(\frac{0.04\ \alpha\ (1-\alpha)}{0.08\ \alpha\ (1-\alpha)}\right) = \ln 0.5 < 0.\end{aligned}$$

8.5 Further Reading

There are a large number of other notions of multivariate dependence. Barlow provides a useful discussion of these and their relationships amongst each other in [Barlow and Proschan, 1975, pp. 142-155].

Goodman discusses ordinal data analysis and model types (with some relevance to monotonicity) in [Goodman, 1981].

Dykstra et al. summarize prior work (by a number of other researchers) in monotonicity-related multivariate dependence properties [Dykstra et al., 1973, Section 3]. This includes some brief discussion of implications of one property to another.

Bibliography

[Abu-Mostafa, 1995] Abu-Mostafa, Y. S. (1995). Hints. *Neural Computation*, 7(4):639–671.

[Agresti, 1990] Agresti, A. (1990). *Categorical Data Analysis.* Wiley-Interscience.

[Agresti, 1996] Agresti, A. (1996). *An Introduction to Categorical Data Analysis.* Wiley-Interscience.

[Agresti and Chuang, 1985] Agresti, A. and Chuang, C. (1985). Bayesian and maximum likelihood approaches to order-restricted inference for models for ordinal categorical data. In *Proc. Advances in Order Restricted Statistical Inference*, pages 6–27. Springer-Verlag.

[Alam and Wallenius, 1976] Alam, K. and Wallenius, K. (1976). Positive dependence and monotonicity in conditional distributions. *Communications in Statistics: Theory and Methods*, A5(6):525–534.

[Altendorf and D'Ambrosio, 2004] Altendorf, E. E. and D'Ambrosio, B. (2004). Feature definition and discovery in probabilistic relational models. In *Proc. International Conference on Machine Learning Workshop on Learning Statistical Models from Relational Data.*

[Altendorf et al., 2005] Altendorf, E. E., Restificar, A. C., and Dietterich, T. G. (2005). Learning from sparse data by exploiting monotonicity constraints. In *Proc. Uncertainty in Artificial Intelligence (to appear).*

[Archer and Wang, 1993] Archer, N. P. and Wang, S. (1993). Application of the back propagation neural network algorithm with monotonicity constraints for two-group classification problems. *Decision Sciences*, 24(1):60–75.

[Barlow and Proschan, 1975] Barlow, R. E. and Proschan, F. (1975). *Statistical Theory of Reliability and Life Testing: Probability Models.* McArdle Press.

[Bazaraa et al., 1993] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. (1993). *Nonlinear programming: theory and algorithms; (2nd ed.).* John Wiley & Sons.

[Ben-David, 1995] Ben-David, A. (1995). Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning*, 19(1):29–43.

[Bennett and Mangasarian, 1992] Bennett, K. P. and Mangasarian, O. L. (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34.

[Bergadano and Gunetti, 1996] Bergadano, F. and Gunetti, D. (1996). *Inductive Logic Programming: From Machine Learning to Software Engineering.* MIT Press.

[Bobrow, 1985] Bobrow, D. G., editor (1985). *Qualitative Reasoning about Physical Systems.* MIT Press.

[Bohanec and Rajkovic, 1988] Bohanec, M. and Rajkovic, V. (1988). Knowledge acquisition and explanation for multi-attribute decision making. In *Proc. Intl. Workshop on Expert Systems and their Applications*, pages 59–78.

[Clark and Matwin, 1993] Clark, P. and Matwin, S. (1993). Using qualitative models to guide inductive learning. In *Proc. International Conference on Machine Learning*, pages 49–56.

[Daniels et al., 2002] Daniels, H., Feelders, A., and Velikova, M. (2002). Integrating economic knowledge in data mining algorithms. In *Intl. Conference of the Society for Computational Economics.*

[Daniels and Kamp, 1999] Daniels, H. and Kamp, B. (1999). Application of MLP networks to bond rating and house pricing. *Neural Computing & Applications*, 8:226–234.

[Davies, 1988] Davies, T. R. (1988). Determination, uniformity, and relevance: normative criteria for generalization and reasoning by analogy. Technical Report Report No. CSLI-88-126, Center for the Study of Language and Information, Stanford University.

[De Raedt and Kersting, 2003] De Raedt, L. and Kersting, K. (2003). Probabilistic logic learning. *SIGKDD Explor. Newsl.*, 5(1):31–48.

[Dykstra, 1982] Dykstra, R. L. (1982). Maximum likelihood estimation of the survival functions of stochastically ordered random variables. *Journal of the American Statistical Association*, 77(379):621–628.

[Dykstra et al., 1973] Dykstra, R. L., Hewett, J. E., and W. A. Thompson, J. (1973). Events which are almost independent. *The Annals of Statistics*, 1(3):674–681.

[Feelders, 2000] Feelders, A. J. (2000). Prior knowledge in economic applications of data mining. In Zighed, D. A., Komorowski, H. J., and Zytkow, J. M., editors, *Proc. of Fourth European Conf. on Principles and Practice of Knowledge Discovery in Databases*, volume 1910 of *Lecture Notes in Computer Science*, pages 395–400. Springer.

[Fletcher, 1987] Fletcher, R. (1987). *Practical methods of optimization; (2nd ed.)*. Wiley-Interscience.

[Forbus, 1985] Forbus, K. D. (1985). Qualitative process theory. In Bobrow, D. G., editor, *Qualitative Reasoning about Physical Systems*, pages 85–168. MIT Press, Cambridge, MA.

[Fung et al., 2001] Fung, G., Mangasarian, O. L., and Shavlik, J. (2001). Knowledge-based support vector machine classifiers. Technical Report 01-09, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-09.ps.

[Getoor et al., 2001] Getoor, L., Friedman, N., Koller, D., and Taskar, B. (2001). Learning probabilistic models of relational structure. In *Proc. International Conference on Machine Learning*, pages 170–177, San Francisco, California. Morgan Kaufmann.

[Glesner and Koller, 1995] Glesner, S. and Koller, D. (1995). Constructing flexible dynamic belief networks from first-order probabilistic knowledge bases. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 217–226.

[Goodman, 1981] Goodman, L. A. (1981). Association models and canonical correlation in the analysis of cross-classifications having ordered categories. *Journal of the American Statistical Association*, 76(374):320–334.

[Haberman, 1976] Haberman, S. J. (1976). Generalized residuals for log-linear models. In *Proceedings of the 9th International Biometrics Conference*, pages 104–122.

[Heckerman, 1999] Heckerman, D. (1999). A tutorial on learning with bayesian networks. In Jordan, M. I., editor, *Learning in graphical models*, pages 301–354. MIT Press.

[Heckerman, 2004] Heckerman, D. (2004). Probabilistic entity-relationship models, PRMs, and plate models. In *Proc. International Conference on Machine Learning Workshop on Learning Statistical Models from Relational Data.*

[Hellerstein, 1990] Hellerstein, J. (1990). Obtaining quantitative predictions from monotone relationships. In *Proc. National Conference on Artifical Intelligence (AAAI-90)*, pages 388–394.

[Kay and Ungar, 1993] Kay, H. and Ungar, L. (1993). Deriving monotonic function envelopes from observations. In *Proc. Qualitative Reasoning about Physical Systems.*

[Krantz et al., 1971] Krantz, D. H., Luce, R. D., Suppes, P., and Tversky, A. (1971). *Foundations of Measurement.* Academic Press, New York.

[Kuipers, 1994] Kuipers, B. (1994). *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge.* MIT Press.

[Lehmann, 1955] Lehmann, E. L. (1955). Ordered families of distributions. *Annals of Mathematical Statistics*, 26:399–419.

[Lenat and Guha, 1989] Lenat, D. B. and Guha, R. V. (1989). *Building Large Knowledge-Based Systems.* Addison-Wesley.

[Mahadevan and Tadepalli, 1994] Mahadevan, S. and Tadepalli, P. (1994). Quantifying prior determination knowledge using PAC learning model. *Machine Learning*, 17(1):69–105.

[McCallum, 2002] McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

[Natarajan et al., 2005] Natarajan, S., Tadepalli, P., Altendorf, E., Dietterich, T., Fern, A., Herlocker, J., and Restificar, A. (2005). Learning first-order probabilistic models with combining rules. In *Proc. International Conference on Machine Learning (to appear).*

[Ngo and Haddawy, 1996] Ngo, L. and Haddawy, P. (1996). A knowledge-based model construction approach to medical decision making. In Cimino, J. J., editor, *Proc. American Medical Informatics Association Annual Fall Symp.*

[Pierre, 1986] Pierre, D. A. (1986). *Optimization theory with applications.* Dover.

[Potharst and Feelders, 2002] Potharst, R. and Feelders, A. J. (2002). Classification trees for problems with monotonicity constraints. *SIGKDD Explorations*, 4(1):1–10.

[Press et al., 1988] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988). *Numerical recipes in C: the art of scientific computing.* Cambridge University Press.

[Quinlan, 1993] Quinlan, R. (1993). Combining instance-based and model-based learning. In *Proc. International Conference on Machine Learning*, pages 236–243.

[Ross, 1983] Ross, S. M. (1983). *Introduction to Stochastic Dynamic Programming.* Academic Press, Inc.

[Russell, 1989] Russell, S. (1989). *The Use of Knowledge in Analogy and Induction.* Morgan Kaufmann, San Mateo, California.

[Russell and Norvig, 1995] Russell, S. J. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Prentice Hall.

[Smith et al., 1988] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R. S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proc. Symp. on Computer Applications and Medical Care*, pages 261–265.

[Szekli, 1995] Szekli, R. (1995). *Stochastic Ordering and Dependence in Applied Probability.* Springer-Verlag.

[Towell and Shavlik, 1994] Towell, G. G. and Shavlik, J. W. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1–2):119–165.

[van der Gaag et al., 2004] van der Gaag, L. C., Bodlaender, H. L., and Feelders, A. (2004). Monotonicity in Bayesian networks. In *Proc. Uncertainty in Artifical Intelligence*, pages 569–576, Arlington, Virginia. AUAI Press.

[Wellman, 1990] Wellman, M. P. (1990). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303.

[Wellman et al., 1992] Wellman, M. P., Breese, J. S., and Goldman, R. P. (1992). From knowledge bases to decision models. *Knowledge Engineering Review*, 7:35–53.