A Theoretical Framework for Learning Bayesian Networks with Parameter Inequality Constraints

Radu Stefan Niculescu
Siemens Medical Solutions
Malvern PA, USA
stefan.niculescu@siemens.com

Tom M. Mitchell
Carnegie Mellon University
Pittsburgh PA, USA
tom.mitchell@cs.cmu.edu

R. Bharat Rao
Siemens Medical Solutions
Malvern PA, USA
bharat.rao@siemens.com

Abstract

The task of learning models for many real-world problems requires incorporating domain knowledge into learning algorithms, to enable accurate learning from a realistic volume of training data. Domain knowledge can come in many forms. For example, expert knowledge about the relevance of variables relative to a certain problem can help perform better feature selection. Domain knowledge about the conditional independence relationships among variables can help learning of the Bayesian Network structure.

This paper considers a different type of domain knowledge for constraining parameter estimates when learning Bayesian Networks. In particular, we consider domain knowledge that comes in the form of inequality constraints among subsets of parameters in a Bayesian Network with known structure. These parameter constraints are incorporated into learning procedures for Bayesian Networks, by formulating this task as a constrained optimization problem. The main contribution of this paper is the derivation of closed form Maximum Likelihood parameter estimators in the above setting.

1 Introduction

Probabilistic models have become increasingly popular in the last decade because of their ability to capture non-deterministic relationships among variables describing many real world domains. Among these models, Bayesian Networks [Heckerman, 1999] have received significant attention because of their ability to compactly encode conditional independence assumptions over random variables and because of the development of effective algorithms for inference and learning based on these representations. A Bayesian Network consists of two components: a structure, which encodes the assumption that a variable is conditionally independent of its non-descendants in the network, given the value of its parents, and a set of parameters, which describe how each variable relates probabilistically to its parents. A Bayesian Network encodes a unique joint probability distribution, which can be easily computed using the chain rule.

When learning Bayesian Networks, the correctness of the learned network of course depends on the amount of training data available. When training data is scarce, it is useful to employ various forms of prior knowledge about the domain to improve the accuracy of learned models. For example, a domain expert might provide prior knowledge specifying conditional independencies among variables, constraining or even fully specifying the network structure of the Bayesian Network. In addition to helping specify the network structure, the domain expert might also provide prior knowledge about the values of certain parameters in the conditional probability tables (CPTs) of the network, or knowledge in the form of prior distributions over these parameters. While previous research has examined a number of approaches to representing and utilizing prior knowledge about Bayesian Network parameters, the type of prior knowledge that can be utilized by current learning methods remains limited, and is insufficient to capture many types of knowledge that may be readily available from experts.

The main contribution of our paper consists of deriving closed form Maximum Likelihood estimators for the parameters in a Bayesian Network, in the setting that expert domain knowledge is available in the form of parameter inequality constraints, which current methods can not accommodate. With our estimators comes a theorem that describes the performance in the case when the domain knowledge represented by the inequality constraints might not be entirely accurate.

The next section of the paper describes related research on constraining parameter estimates for Bayesian Networks. Section 3 presents the task of parameter estimation in the presence of general parameter constraints and formulates it as a constrained optimization problem. Section 4 details the main contribution of this paper: closed form solutions for parameter estimates for several classes of parameter inequality constraints. Some formal guarantees about our estimators are discussed in section 5. We conclude with a brief summary of this research along with several directions for future work.

2 Related Work

The main methods to represent relationships among parameters of a Bayesian Network fall into two main categories: Dirichlet Priors and their variants (including smoothing techniques) and Parameter Sharing of several kinds.

In [Geiger and Heckerman, 1997], it is shown that Dirichlet Priors are the only possible priors for discrete Bayesian Networks, provided certain assumptions hold. One can think of a Dirichlet Prior as an expert's guess for the parameters in a discrete Bayesian Network, allowing room for some variance around the guess. One of the main problems with Dirichlet Priors and related models is that it is impossible to represent even simple equality constraints between parameters (for example the constraint: $\theta_{111} = \theta_{121}$ where $\theta_{ijk} = P(X_i = x_{ij}|Parents(X_i) = pa_{ik})$) without using priors on the hyperparameters of the Dirichelet Prior, in which case the marginal likelihood can no longer be computed in closed form, and expensive approximate methods are required to perform parameter estimation. A second problem is that it is often beyond the expert's ability to specify a full Dirichlet Prior over the parameters of a Bayesian Network.

A widely used form of parameter constraints employed by Bayesian Networks is *Parameter Sharing*. Models using different types of Parameter Sharing include: Dynamic Bayesian Networks [Murphy, 2002] and their special case Hidden Markov Models [Rabiner, 1989], Module Networks [Segal *et al.*, 2003], Context Specific Independence models [Boutilier *et al.*, 1996] such as Bayesian Multinetworks, Recursive Multinetworks and Dynamic Multinetworks [Geiger and Heckerman, 1996; Pena *et al.*, 2002; Bilmes, 2000], Probabilistic Relational Models [Friedman *et al.*, 1999], Object Oriented Bayes Nets [Koller and Pfeffer, 1997], Kalman Filters [Welch and Bishop, 1995] and Bilinear Models [Tenenbaum and Freeman, 2000]. Parameter Sharing methods constrain parameters to share the same value, but do not capture more complicated constraints among parameters such as inequality constraints or constraints on sums of parameter values. The above methods are restricted to sharing parameters at either the level of sharing a conditional probability table (CPT) (Module Networks, HMMs), at the level of sharing a conditional probability distribution within a single CPT (Context Specific Independence), at the level of sharing a state-to-state transition matrix (Kalman Filters) or at the level of sharing a style matrix (Bilinear Models). None of the above models allow sharing at the level of granularity of individual parameters. [Niculescu *et al.*, 2005] introduces a parameter equality constraint framework that describes many models that use parameter sharing: Module Networks, Context Specific Independence models, HMMs and Dynamic Bayesian Networks. Within this framework, it is showed how efficient parameter learning is performed from both a frequentist and bayesian point of view, from both observable and partially observable data.

All the models described so far can only take advantage of certain types of parameter equality constraints. Only recently, [Altendorf *et al.*, 2005; Feelders and van der Gaag, 2006] study the feasibility of incorporating simple inequality constraints in the learning of parameters of Bayesian Networks. The constraints analyzed in these papers are somehow restrictive, in the sense that each constraint must involve all parameters in a conditional probability table, whereas our framework allows for constraints at individual distribution level of granularity. Additionally, in [Altendorf *et al.*, 2005], the authors employ an approximation algorithm, whereas we derive exact estimators. Further, [Altendorf *et al.*, 2005] assumes the values of the variables can be totally ordered and [Feelders and van der Gaag, 2006] assumes all variables are binary. Our framework makes none of these assumptions. In the next section we will place parameter learning with inequality constraints in a general constrained optimization framework and we will show how closed form learning of the parameters of Bayesian Networks can be performed when expert knowledge is available in the form of either of two types of parameter inequality constraints.

3 Problem Definition and Approach

Here we define the problem and suggest a general optimization based approach to solve it. This approach has serious limitations when the constraints are arbitrary. However, it constitutes the basis for computing the Maximum Likelihood estimators for the classes of inequality constraints described in section 4. We begin by describing the problem along with several assumptions.

3.1 The Problem

Our task here is to perform parameter estimation in a Bayesian Network where the structure is known in advance. To accomplish this task, we assume a dataset of examples is available. In addition, a set of parameter equality and/or inequality constraints is provided by a domain expert. The equality constraints are of the form $g_i(\theta) = 0$ for $1 \leq i \leq m$ and the inequality constraints are of the form $h_j(\theta) \leq 0$ for $1 \leq j \leq k$, where θ represents the set of parameters of the Bayesian Network.

Initially we will assume the domain knowledge provided by the expert is correct. Later, we investigate what happens if this knowledge is not completely accurate. Next we enumerate several assumptions that must be satisfied for our methods to work. These are similar to common assumptions made when learning parameters in standard Bayesian Networks.

First, we assume that the examples in the training dataset are drawn independently from the underlying distribution. In other words, examples are conditionally independent given the parameters of the Bayesian Network. Second, we assume that all the variables in the Bayesian Network can take on at least two different values. This is a safe assumption since there is no uncertainty in a random variable with only one possible value. Any such variables in our Bayesian Network can be deleted, along with all arcs into and out of the nodes corresponding to those variables. Third, when computing parameter estimators, we additionally assume that all observed counts corresponding to parameters in the Bayesian Network are strictly positive. We enforce this condition in order to avoid potential divisions by zero, which may impact inference negatively. In the real world it is expected there will be observed counts which are zero. This problem can be solved by using priors on parameters, that essentially have the effect of adding a positive quantity to the observed counts and essentially create strictly positive virtual counts.

Finally, the functions $g_1, \ldots, g_m$ and $h_1, \ldots, h_k$ must be twice differentiable, with continuous second derivatives. This assumption justifies the formulation of our problem as a con-

strained maximization problem that can be solved using standard optimization methods.

3.2 A General Approach

In order to solve the problem described above, here we briefly suggest an approach based on already existing optimization techniques. The idea is to formulate our problem as a constrained maximization problem where the objective function is the data log-likelihood $\log P(D|\theta)$ and the constraints are given by $g_i(\theta) = 0$ for $1 \leq i \leq m$ and $h_j(\theta) \leq 0$ for $1 \leq j \leq k$. It is easy to see that, applying the Karush-Kuhn-Tucker conditions theorem [Kuhn and Tucker, 1951], the maximum must satisfy a system with the same number of equations as variables. To solve this system, one can use any of several already existing methods (for example the Newton-Raphson method [Press *et al.*, 1993]).

It is well known that finding a solution for the system given by the KKT conditions is not enough to determine the optimum point, but fortunately the objective function is concave and most of the constraints we have encountered in real life are linear equalities or inequalities. Therefore several well known sufficiency criteria can guarantee optimality.

Unfortunately, the above methods have serious shortcomings in the general case. With a large number of parameters in the Bayesian Network, they can be extremely expensive because they involve potentially multiple runs of the Newton-Raphson method and each such run requires several expensive matrix inversions. Other methods for finding the solutions of a system of equations can be employed, but, as noted in [Press *et al.*, 1993], all these methods have limitations in the case when the constraints are arbitrary, non-linear functions. The worst case happens when there exists a constraint that explicitly uses all parameters in the Bayesian Network.

The above method should be regarded as a mere suggestion and, because of its shortcomings, we believe it is not computationally feasible with arbitrary constraints. We mentioned it here only to show how learning in the presence of parameter constraints can be formulated as a general constrained maximization problem. This general approach also provides the starting point for finding closed form Maximum Likelihood estimators given the particular classes of parameter inequality constraints presented in the next section.

4 Learning with Inequality Constraints

In this section we derive closed form Maximum Likelihood estimators for the parameters in a discrete Bayesian Network with known structure when domain knowledge is provided as either of the two types of inequality constraints.

4.1 Inequalities between Sums of Parameters

Briefly, this type of Parameter Domain Knowledge states that the sum of several parameters within one conditional probability distribution is bounded by the sum of other parameters in the same distribution of the Bayesian Network. Intuitively, one can think of this constraint in terms of the parts of speech of a language. Usually, an adverb comes along with a verb and therefore it is reasonable to assume that a language expert can specify that the aggregate probability mass of adverbs is no greater than the aggregate probability mass of the verbs in a given language. Formally, in this type of domain knowledge, the parameters of a conditional probability distribution, denoted by $\theta_1, \ldots, \theta_n$, can be partitioned into $\theta = \cup_{k=1}^{s} A_k \cup_{k=1}^{s} B_k \cup C$ such that $\sum_{\theta_i \in A_k} \theta_i \leq \sum_{\theta_i \in B_k} \theta_i$ for all $1 \leq k \leq s$. Let us denote by N_{A_k} the sum of the observed counts corresponding to parameters in A_k. Similar definitions hold for N_{B_k} and N_C. Let N be the sum of all observed counts N_i corresponding to parameters θ.

An expert can potentially specify different such constraints for several conditional probability distributions in the Bayesian Network. Because of the decomposability of log-likelihood, the problem of computing the Maximum Likelihood parameter estimators can be decomposed in a set of independent optimization subproblems, one for each conditional probability distribution in the network. We have the following theorem:

Theorem 4.1. *If all N_i are strictly positive, the Maximum Likelihood Estimators of parameters θ are given by:*

a) $\hat{\theta}_i = \frac{N_i}{N} \cdot \frac{N_{A_k}+N_{B_k}}{2 \cdot N_{A_k}}$ *if* $\theta_i \in A_k$ *and* $N_{A_k} \geq N_{B_k}$

b) $\hat{\theta}_i = \frac{N_i}{N} \cdot \frac{N_{A_k}+N_{B_k}}{2 \cdot N_{B_k}}$ *if* $\theta_i \in B_k$ *and* $N_{A_k} \geq N_{B_k}$

c) $\hat{\theta}_i = \frac{N_i}{N}$ *if* $\theta_i \in A_k \cup B_k$ *and* $N_{A_k} < N_{B_k}$

d) $\hat{\theta}_i = \frac{N_i}{N}$ *if* $\theta_i \in C$

Proof. Finding Maximum Likelihood estimators is equivalent to maximizing $l(\theta) = \sum_i N_i \cdot \log \theta_i$ subject to the domain knowledge constraints, including the constraint that $g(\theta) = \sum_i \theta_i - 1 = 0$. Since this problem contains inequality constraints, we can attempt to solve it using Karush-Kuhn-Tucker theorem. We introduce the Lagrange Multiplier λ for g and μ_k for inequality constraint $h_k(\theta) = \sum_{\theta_i \in A_k} \theta_i - \sum_{\theta_i \in B_k} \theta_i \leq 0$. The optimum $\hat{\theta}$ can then be found among the solutions of the system:

$$\begin{cases} \nabla_\theta l(\hat{\theta}) - \lambda \cdot \nabla_\theta g(\hat{\theta}) - \sum_k \mu_k \cdot \nabla_\theta h_k(\hat{\theta}) = 0 \\ g(\hat{\theta}) = 0 \\ \mu_k \cdot h_k(\hat{\theta}) = 0 \\ h_k(\hat{\theta}) \leq 0 \\ \mu_k \geq 0 \end{cases}$$

From the first equation we obtain:

$$\hat{\theta}_i = \begin{cases} \frac{N_i}{\lambda + \mu_k} \text{ if } \theta_i \in A_k \\ \frac{N_i}{\lambda - \mu_k} \text{ if } \theta_i \in B_k \\ \frac{N_i}{\lambda} \text{ if } \theta_i \in C \end{cases}$$

Therefore, $\sum_{\theta_i \in A_k} \hat{\theta}_i = \frac{N_{A_k}}{\lambda+\mu_k}$ and $\sum_{\theta_i \in B_k} \hat{\theta}_i = \frac{N_{B_k}}{\lambda-\mu_k}$. Based on whether constraint k is tight or not we have:

- If $h_k(\hat{\theta}) = 0$, then $\frac{N_{A_k}}{\lambda+\mu_k} = \frac{N_{B_k}}{\lambda-\mu_k}$. This implies $\frac{N_{A_k}}{\lambda+\mu_k} = \frac{N_{B_k}}{\lambda-\mu_k} = \frac{N_{A_k}+N_{B_k}}{2\lambda}$ and therefore $\sum_{\theta_i \in A_k \cup B_k} \hat{\theta}_i = \frac{N_{A_k}+N_{B_k}}{\lambda}$. In this case, we also have $\lambda \cdot (N_{A_k} - N_{B_k}) = \mu_k \cdot (N_{A_k} + N_{B_k})$. Since $\mu_k \geq 0$, we also must have $N_{A_k} \geq N_{B_k}$ in order for constraint k to be tight.

- If $h_k(\hat{\theta}) < 0$, then $\mu_k = 0$ and therefore we again have $\sum_{\theta_i \in A_k \cup B_k} \hat{\theta}_i = \frac{N_{A_k}+N_{B_k}}{\lambda}$. In this case we also have $h_k(\hat{\theta}) = \frac{N_{A_k}-N_{B_k}}{\lambda}$ and since $h_k(\hat{\theta}) < 0$, we must also have $N_{A_k} < N_{B_k}$.

The above observations allow us to conclude that a constraint is tight if and only if $N_{A_k} \geq N_{B_k}$. Now, summing up over all parameters in the conditional probability distribution we get:

$$1 = \sum_i \hat{\theta}_i = \frac{N_C + \sum_k (N_{A_k} + N_{B_k})}{\lambda} = \frac{N}{\lambda}$$

This gives us: $\lambda = N$ and therefore:

$$\hat{\theta}_i = \begin{cases} \frac{N_i}{N+\mu_k} \text{ if } \theta_i \in A_k \\ \frac{N_i}{N-\mu_k} \text{ if } \theta_i \in B_k \\ \frac{N_i}{N} \text{ if } \theta_i \in C \end{cases}$$

Assume now that $N_{A_k} \geq N_{B_k}$. According to the observations above, it means constraint k is tight and we have: $\frac{N_{A_k}}{N+\mu_k} = \frac{N_{B_k}}{N-\mu_k} = \frac{N_{A_k}+N_{B_k}}{2 \cdot N}$. From this we immediately derive: $\hat{\theta}_i = \frac{N_i}{N} \cdot \frac{N_{A_k}+N_{B_k}}{2 \cdot N_{A_k}}$ if $\theta_i \in A_k$ and $N_{A_k} \geq N_{B_k}$ and $\hat{\theta}_i = \frac{N_i}{N} \cdot \frac{N_{A_k}+N_{B_k}}{2 \cdot N_{B_k}}$ if $\theta_i \in B_k$ and $N_{A_k} \geq N_{B_k}$.

If $N_{A_k} < N_{B_k}$, then, as discussed above, μ_k must be 0 and therefore $\hat{\theta}_i = \frac{N_i}{N}$ if $\theta_i \in A_k \cup B_k$ and $N_{A_k} < N_{B_k}$. Because the log-likelihood objective function is concave and because the constraints are linear inequalities, it follows that $\hat{\theta}$ is the set of Maximum Likelihood estimators. This concludes the proof of our theorem. □

4.2 Upper Bounds on Sums of Parameters

Here the domain expert provides upper bounds on the sum of several parameters within one conditional probability distribution in the Bayesian Network. Consider the same language example described in the introduction of the previous subsection. Here the expert may state that the aggregate probability of nouns is no greater than 0.4, the aggregate probability of verbs is no greater than 0.4 and the aggregate probability of adjectives is no greater than 0.3. Even though the combined probability mass of all words equals one, the sum of the upper bounds provided by the expert can be greater than one. Formally, in this type of domain knowledge, the parameters of a conditional probability distribution, denoted by $\theta_1, \ldots, \theta_n$, can be partitioned in $\theta = \cup_{k=1}^{s} A_k$ such that $\sum_{\theta_i \in A_k} \theta_i \leq \alpha_k$ for all $1 \leq k \leq s$, where α_k is a given positive constant. Again, denote by N_{A_k} the sum of the observed counts corresponding to parameters in A_k and by N be the sum of all observed counts N_i corresponding to parameters θ. If there are parameters not involved in any of these constraints, then we can consider they belong to their own set A_k with $\alpha_k = 1$.

In the previous subsection we found an easy way to decide whether a constraint is tight at the optimum point. For the type of constraints we deal with here, we are not able to derive such a simple criterion. However, we show a simple, linear algorithm that computes the set of tight constraints at the optimum point. This algorithm starts with an empty set and at each step adds one of the final tight constraints.

Again, an expert can specify different sets of such constraints for different conditional probability distributions in the network and, because of the decomposability of log-likelihood, we can decompose the task of finding the Maximum Likelihood estimators into a set of independent optimization subproblems:

Theorem 4.2. *Assume all observed counts N_i are strictly positive and also assume we know the set $K = \{k_1, \ldots, k_l\}$ of constraints that are tight at the point given by the Maximum Likelihood estimators $\hat{\theta}$. Then, we have:*

a) $\hat{\theta}_i = \alpha_k \cdot \frac{N_i}{N_{A_k}}$ if $\theta_i \in A_k$ and $k \in K$

b) $\hat{\theta}_i = (1 - \sum_{j \in K} \alpha_j) \cdot \frac{N_i}{\sum_{m \notin K} N_{A_m}}$ if $\theta_i \in A_k$ and $k \notin K$

Proof. We can approach the problem of finding the Maximum Likelihood estimators in a similar fashion as in Theorem 4.1. The data log-likelihood is given by $l(\theta) = \sum_i N_i \cdot \log \theta_i$ which we have to maximize with respect to the domain knowledge constraints, including the constraint that $g(\theta) = \sum_i \theta_i - 1 = 0$. Again, we use Karush-Kuhn-Tucker theorem. We introduce the Lagrange Multiplier λ for g and μ_k for inequality constraint $h_k(\theta) = \sum_{\theta_i \in A_k} \theta_i - \alpha_k \leq 0$. The optimum $\hat{\theta}$ can then be found among the solutions of the system:

$$\begin{cases} \nabla_\theta l(\hat{\theta}) - \lambda \cdot \nabla_\theta g(\hat{\theta}) - \sum_k \mu_k \cdot \nabla_\theta h_k(\hat{\theta}) = 0 \\ g(\hat{\theta}) = 0 \\ \mu_k \cdot h_k(\hat{\theta}) = 0 \\ h_k(\hat{\theta}) \leq 0 \\ \mu_k \geq 0 \end{cases}$$

From the first equation we obtain:

$$\hat{\theta}_i = \frac{N_i}{\lambda + \mu_k} \text{ if } \theta_i \in A_k$$

Therefore, $\sum_{\theta_i \in A_k} \hat{\theta}_i = \frac{N_{A_k}}{\lambda+\mu_k}$. Based on whether constraint k is tight or not we have:

- If $h_k(\hat{\theta}) = 0$ i.e. $k \in K$, then $\frac{N_{A_k}}{\lambda+\mu_k} = \alpha_k$. This implies $\hat{\theta}_i = \frac{N_i}{\lambda+\mu_k} = \alpha_k \cdot \frac{N_i}{N_{A_k}}$.
- If $h_k(\hat{\theta}) < 0$ i.e. $k \notin K$, then $\mu_k = 0$ and therefore we have $\sum_{\theta_i \in A_k} \hat{\theta}_i = \frac{N_{A_k}}{\lambda}$.

Summing up over all parameters not involved in the tight constraints, we get:

$$(1 - \sum_{j \in K} \alpha_j) = \sum_{\theta_i \in A_k, k \notin K} \theta_i = \frac{\sum_{j \notin K} N_{A_j}}{\lambda}$$

We obtain $\lambda = \frac{\sum_{m \notin K} N_{A_m}}{1 - \sum_{j \in K} \alpha_j}$ and further: $\hat{\theta}_i = (1 - \sum_{j \in K} \alpha_j) \cdot \frac{N_i}{\sum_{m \notin K} N_{A_m}}$ if $\theta_i \in A_k$ and $k \notin K$. Because

the log-likelihood objective function is concave and because the constraints are linear inequalities, it follows that $\hat{\theta}$ is the set of Maximum Likelihood estimators. This concludes our derivation of the Maximum Likelihood estimators when we know in advance which constraints are satisfied by our estimators. □

Next we describe the algorithm that finds the set K of tight constraints:

Algorithm 4.1. (Finding the set of tight constraints if $\sum_j \alpha_j \neq 1$)

STEP 1. *Start with $K = \emptyset$ and at each step add a constraint to K.*

STEP 2. *If $K = \{k_1, \ldots, k_l\}$, let $\lambda_l = \frac{\sum_{m \notin K} N_{A_m}}{1 - \sum_{j \in K} \alpha_j}$ as in the above theorem.*

STEP 3. *If there exists $k_{l+1} \notin K$ such that $\frac{N_{A_{k_{l+1}}}}{\alpha_{k_{l+1}}} \geq \lambda_l$, let $K = K \cup \{k_{l+1}\}$ and GO TO Step 2. Otherwise STOP and declare K the set of tight constraints.*

Proof. **(Correctness of Algorithm)** We start by making the following observation, based on the proof of Theorem 4.2:

- If h_k tight, then $\frac{N_{A_k}}{\lambda + \mu_k} = \alpha_k$. Because $\mu_k \geq 0$, we must have $\frac{N_{A_k}}{\alpha_k} \geq \lambda$.
- If h_k not tight, then $\mu_k = 0$ and therefore we have $0 > h_k(\hat{\theta}) = \frac{N_{A_k}}{\lambda} - \alpha_k$ and therefore we must have $\frac{N_{A_k}}{\alpha_k} < \lambda$. It is obvious that $\lambda \geq 0$ must hold, otherwise we would have negative parameters.

We have just developed a criterion to test if a set K of constraints is the set of tight constraints:

Lemma 4.1. *Given λ (which depends on K) computed as in Theorem 4.2, K is the set of tight constraints if and only if $\frac{N_{A_k}}{\alpha_k} \geq \lambda$ for all $k \in K$ and $\frac{N_{A_k}}{\alpha_k} < \lambda$ for all $k \notin K$.*

Before proving that our algorithm produces the set of tight constraints, let us prove another useful result:

Lemma 4.2. *If $\sum_j \alpha_j \neq 1$ then $N = \lambda_0 \geq \lambda_1 \geq \ldots$, and the quantity $1 - \sum_{j \in K} \alpha_j$ is always strictly positive.*

Proof. **(of lemma)** Since initially $K = \emptyset$, it is obvious that $1 - \sum_{j \in K} \alpha_j \geq 0$. It is also obvious $\lambda_0 = N$. Let us verify the induction step.

From $\frac{N_{A_{k_l}}}{\alpha_{k_l}} \geq \lambda_l$ and because $1 - \sum_{j \in K} \alpha_j > 0$ we get:

$$N_{A_{k_l}} \cdot (1 - \alpha_{k_l} - \sum_{j \in K} \alpha_j) \geq \alpha_{k_l} \cdot \sum_{m \notin K \cup \{k_l\}} N_{A_m} \qquad (1)$$

It follows $(1 - \sum_{j \in K \cup k_l} \alpha_j) \geq 0$ with equality if and only if we processed all constraints, in which case we have $1 = \sum_j \alpha_j$ and it is obvious that all constraints must be tight. However, since we assumed $\sum_j \alpha_j \neq 1$, we must have $(1 - \sum_{j \in K \cup k_l} \alpha_j) > 0$ and the first part of the induction step is proved.

If in both sides of inequality 1 we add the quantity $(1 - \alpha_{k_l} - \sum_{j \in K} \alpha_j) \cdot \sum_{m \notin K \cup \{k_l\}} N_{A_m}$, we obtain:

$$(1 - \alpha_{k_l} - \sum_{j \in K} \alpha_j) \cdot \sum_{m \notin K} N_{A_m} \geq (1 - \sum_{j \in K} \alpha_j) \cdot \sum_{m \notin K \cup \{k_l\}} N_{A_m}$$

which, given that $1 - \alpha_{k_l} - \sum_{j \in K} \alpha_j > 0$, is equivalent to $\lambda_l \geq \lambda_{l+1}$. This concludes the proof of our lemma. □

Applying Lemma 4.2, it follows that, in the case when $\sum_j \alpha_j \neq 1$, the Algorithm 4.1 ends at a step l such that $\frac{N_{A_{k_j}}}{\alpha_{k_j}} \geq \lambda_j \geq \lambda_l$ for all $k_j \in K$ and $\frac{N_{A_k}}{\alpha_k} < \lambda_l$ for all $k \notin K$. From Lemma 4.1 it follows that K is the set of tight constraints in the case when $\sum_j \alpha_j \neq 1$ and therefore Algorithm 4.1 is correct. Another case is when all constraints are processed and we are not left with a λ_l to compare with. This situation can not happen, because, at the last step, we would have:

$$\frac{N_{A_{k_s}}}{1 - \sum_{j \neq k_s} \alpha_j} \leq \frac{N_{A_{k_s}}}{\alpha_{k_s}}$$

and therefore either $\sum_j \alpha_j = 1$ or $\sum_j \alpha_j < 1$. In the second case, the constraints are contradictory, which can not happen because we assume the domain expert provides accurate domain knowledge. If $\sum_j \alpha_j = 1$ (case which is not covered by Algorithm 4.1), it is obvious that the all constraints must be tight not only for the Maximum Likelihood estimators, but for every feasible value of θ. □

5 Formal Guarantees

Sometimes it may happen that the constraints provided by an expert are not completely accurate. In all our methods so far, we assumed that the constraints are correct and therefore errors in domain knowledge can prove detrimental to the performance of our learned models. In this section we investigate the relationship between the true, underlying distribution of the observed data and the distribution estimated using our methods based on inequality constraints. Because of space reasons, we omit the proofs for the theorem and the corollary presented in this section.

Suppose P is the true distribution from which data is sampled. Let P^* be the the closest distribution to P(in terms of $KL(P, \cdot)$) that factorizes according to the given structure and obeys the expert's inequality constraints. We have:

Theorem 5.1. *With an infinite amount of data, the distribution $\hat{P}$ given by the Maximum Likelihood estimators in Theorem 4.1 converges to P^* with probability 1.*

Corollary 5.1. *If the true distribution P factorizes according to the given structure and if the parameter inequality constraints provided by the expert are completely accurate, then the distribution $\hat{P}$ given by the estimators computed in Theorem 4.1 converges to P with probability 1.*

Similar results hold for the inequality constraints described in subsection 4.2.

6 Conclusions and Future Work

Building accurate models from limited training data is possible only by using some form of prior knowledge to augment the data. Prior knowledge in the form of parameter equality constraints has been incorporated in learning of several bayesian network models, including Module Networks, HMMs and Context Specific Independence models.

In this paper we have demonstrated that the standard methods for parameter estimation in Bayesian Networks can be naturally extended to accommodate parameter inequality constraints by formulating this as a constrained maximization problem and deriving closed form Maximum Likelihood parameter estimators. It is also important to note that one can combine the two types of parameter inequality constraints presented in this paper as well as the parameter sharing constraints described in Section 2 when learning the parameters of a Bayesian Network as long as the scopes of these constraints do not overlap.

We have proved that even when the asserted parameter constraints turn out to be incorrect, given an infinite amount of training data, our Maximum Likelihood estimators converge to the best describable distribution; that is, the distribution closest in terms of KL distance from the true distribution, among all distributions that obey the parameter inequality constraints and factor according to the given structure.

We see several useful directions for future work. First, we would like to prove that by incorporating the inequality constraints given by an expert we compute estimators with lower variance than the ones obtained by simply learning the Bayesian Network directly from the data. A second direction to explore is to approach learning from a bayesian (as opposed to a frequentist) point of view. This might be achieved by specifying Constrained Dirichlet Priors which assign zero probability mass over the space where the constraints are not satisfied. However, it is a challenge to compute the normalization constant of such distributions in closed form.

Acknowledgments

We would like to thank John Lafferty, Andrew Moore, Russ Greiner and Zoubin Ghahramani for their useful comments and suggestions. As a student at Carnegie Mellon University, Radu Stefan Niculescu was sponsored by the NSF grants CCR-0085982 and CCR-0122581, by the Darpa PAL program under contract NBCD030010, and by a generous gift from Siemens Medical Solutions.

References

[Altendorf *et al.*, 2005] E. E. Altendorf, A. C. Restificar, and T. G. Dietterich. Learning from sparse data by exploiting monotonicity constraints. In *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 18–26, Arlington, Virginia, 2005. AUAI Press.

[Bilmes, 2000] J. Bilmes. Dynamic bayesian multinets. In *Proceedings of UAI*, pages 38–45, 2000.

[Boutilier *et al.*, 1996] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in bayesian networks. In *Proceedings of 12th UAI*, pages 115–123, 1996.

[Feelders and van der Gaag, 2006] A. J. Feelders and L. C. van der Gaag. Learning bayesian network parameters under order constraints. *International Journal of Approximate Reasoning*, 42:37–53, 2006.

[Friedman *et al.*, 1999] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of 16th IJCAI*, pages 1300–1307, 1999.

[Geiger and Heckerman, 1996] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, 82:45–74, 1996.

[Geiger and Heckerman, 1997] D. Geiger and D. Heckerman. A characterization of the dirichlet distribution through global and local parameter independence. *The Annals of Statistics*, 25:1344–1369, 1997.

[Heckerman, 1999] D. Heckerman. A tutorial on learning with bayesian networks. In M. Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999.

[Koller and Pfeffer, 1997] D. Koller and A. Pfeffer. Object oriented bayesian networks. In *Proceedings of 13th UAI*, pages 302–313, 1997.

[Kuhn and Tucker, 1951] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, 1951.

[Murphy, 2002] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, 2002.

[Niculescu *et al.*, 2005] R. S. Niculescu, T. Mitchell, and R. B. Rao. Parameter related domain knowledge for learning in graphical models. In *Proceedings of SIAM Data Mining conference*, 2005.

[Pena *et al.*, 2002] J. M. Pena, J. A. Lozano, and P. Larranaga. Learning recursive bayesian multinets for data clustering by means of constructive induction. *Machine Learning*, 47(1):63–89, 2002.

[Press *et al.*, 1993] W. H. Press, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1993.

[Rabiner, 1989] R. L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Segal *et al.*, 2003] E. Segal, D. Pe'er, A. Regev, D. Koller, and N. Friedman. Learning module networks. In *Proceedings of 19th UAI*, pages 525–534, 2003.

[Tenenbaum and Freeman, 2000] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, 2000.

[Welch and Bishop, 1995] G. Welch and G. Bishop. An introduction to the kalman filter. Technical Report TR 95-041, University of North Carolina, 1995.