Robustness Analysis of Bayesian Networks with Finitely Generated Convex Sets of Distributions

Fabio Cozman

CMU-RI-TR 96-41

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

January 28, 1997

Abstract

This paper presents exact solutions and convergent approximations for inferences in Bayesian networks associated with finitely generated convex sets of distributions. Robust Bayesian inference is the calculation of bounds on posterior values given perturbations in a probabilistic model. The paper presents exact inference algorithms and analyzes the circumstances where exact inference becomes intractable. Two classes of algorithms for numeric approximations are developed through transformations on the original model. The first transformation reduces the robust inference problem to the estimation of probabilistic parameters in a Bayesian network. The second transformation uses Lavine's bracketing algorithm to generate a sequence of maximization problems in a Bayesian network. The analysis is extended to the ϵ-contaminated, the lower density bounded, the belief function, the sub-sigma, the density bounded, the total variation and the density ratio classes of distributions.

This research is supported in part by NASA under Grant NAGW-1175. Fabio Cozman was supported under a scholarship from CNPq, Brazil.

1 Introduction

Robustness analysis employs sets of distributions to model perturbations in the parameters of a probability distribution [Berger, 1985; 1990; Huber, 1980; Kadane, 1984; Wasserman, 1992b]. Robust Bayesian inference is the calculation of bounds on posterior values given such perturbations. This paper focuses on perturbations that can be modeled by finitely convex sets of distributions and presents the first convergent approximations for calculation of posterior bounds in Bayesian networks.

Models that attempt to combine Bayesian networks with probability intervals have faced serious difficulties. There has been no general model of probability intervals with the same style of efficient propagation used in Bayesian networks. The inefficiency comes from the fact that successive application of Bayes rule can create exponentially large number of vertices in a polytope representation. Two possibilities arise. One is to focus on particular classes of intervals that admit efficient algorithms [Chrisman, 1996b]. Another is to settle for approximations of inference bounds [Breese & Fertig, 1991; Tessem, 1992]. So far, no approximation with convergence guarantees or provably small error bounds has been found.

This paper derives exact solutions and convergent approximations for robustness analysis of Bayesian networks associated with finitely generated sets of distributions. Exact algorithms which generalize previous ideas [Cano, Delgado, & Moral, 1993] are presented and their complexity is characterized. We describe the situations that lead to high complexity in exact algorithms. Two classes of algorithms for numeric approximations are developed. The first class of algorithms reduces the robust inference problem to the estimation of probabilistic parameters in a Bayesian network. The second class of algorithms uses Lavine's bracketing method [Lavine, 1991] to simplify the optimization procedures required for robust inferences.

We use the theory of convex sets of probability distributions, referred to as Quasi-Bayesian theory [Giron & Rios, 1980]. Other names for the mathematical theory are theory of lower expectations [Huber, 1980] and theory of lower previsions [Walley, 1991]. In this theory, the imprecision in probability assessments can be due either to difficulties in eliciting information from experts, or to difficulties in processing or combining data.

Section 2 discusses the importance of robustness analysis in Bayesian networks. We then introduce some basic results and notation in Section 3, and briefly introduce Quasi-Bayesian theory in Section 4. Section 5 defines the concept of finitely generated convex sets of distributions and network representations. Section 6 presents exact algorithms for robust inferences and the trade-offs between them. Section 6 also presents a transformation that reduces a Quasi-Bayesian network to a Bayesian network, which is central to this paper. Section 7 also discusses the complexity of such inferences and some generic strategies to circumvent it, suggesting a simulated scheme that yields approximate bounds. Section 8 shows how to use the transformed network to reduce robust inference to a problem of parameter estimation, and how to use gradient descent techniques to solve them. Section 9 presents algorithms for calculation of posterior bounds which are based on Lavine's bracketing algorithm. A series of classes of distributions is considered in section 10: ϵ-contaminated class (Section 10.1), lower density bounded class (Section 10.2), belief function class (Section 10.3), the sub-sigma class (Section 10.4, the density bounded class (Section 10.5), the total variation class (Section 10.6) and the density ratio (Section 10.7). Inference algorithms are derived by reducing each one of these classes to the class of finitely generated sets of distributions. Section 11 discusses the generalization of the results to expected utility and variance problems.

2 Motivation

In the real world we can rarely meet all the assumptions of a Bayesian model. First, we have to face imperfections in an agent's beliefs, either because the agent had no time, resources, patience, or confidence to provide exact probability values. Second, we may deal with a group of disagreeing experts, each specifying a particular distribution [Levi, 1980; Seidenfeld & Schervish, 1990]. Third, we may be interested in abstracting away parts of a model and assessing the effects of this abstraction [Chrisman, 1996a; Ha & Haddawy, 1996].

To emphasize the relationship between abstraction

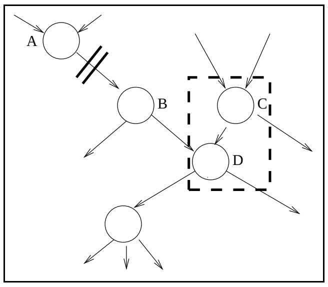

Figure 1: Abstraction in Bayesian networks: elimination of weak links and grouping of variables.

and robustness analysis, suppose a decision-maker is contemplating a situation with several hundred variables. For a particular inference, some variables may minimally affect the probabilities of interest. In Figure 1, node A and its ancestors weakly affect node B, and node C and D can be collapsed into a single node. The original model is abstracted by breaking links and grouping nodes. Now the decision-maker can use robustness analysis to determine bounds on the posterior quantities in the abstracted model. Large bounds indicate poor performance and suggest reevaluation of the abstraction procedure.

3 Background

We consider a set $\tilde{x}$ of discrete variables. Each variable x_i has a finite set of values $\hat{x}_i$. A Bayesian network defines a probability distribution through the expression [Pearl, 1988]:

$$p(\tilde{x}) = \prod_i p(x_i|\text{pa}(x_i)), \tag{1}$$

where $\text{pa}(x_i)$ is a set of variables, the *parents* of variable x_i. $p(\cdot)$ refers to a single probability distribution; $q(\cdot)$ refers to an arbitrary distribution and $r(\cdot)$ refers to an arbitrary member of a set of distributions.

For any function $f(\cdot)$, we use the abbreviation f_i for $f(x_i|\text{pa}(x_i))$ and $f_i(a)$ for $f(x_i = a|\text{pa}(x_i))$. In this notation, expression (1) can be rewritten as $p(\tilde{x}) = \prod_i p_i$.

Suppose a set of variables is associated with evidence $e = \{x_1 = a, x_3 = b, \ldots, x_n = k\}$. Write $f^e(\cdot)$ for a function where the values e are fixed. For example, $p^e(\tilde{x}) = p(x_1 = a, x_2, x_3 = b, x_4, \ldots, x_n = k)$ for the evidence above. Notice $p^e(x)$ is not the conditional $p(x|e)$, but the function $p(x)$ with some variables fixed to e.

Our algorithms assume efficient solutions for two standard problems in usual Bayesian networks: computation of posterior marginals and maximum a posteriori hypothesis. The calculation of posterior marginals for a queried variable can be done by a technique like variable elimination [Zhang & Poole, 1996] or bucket elimination [Dechter, 1996], or a more sophisticated clustering method such as peeling [Cannings & Thompson, 1981] or graph triangulation [Jensen, 1996].

Another standard problem is the determination of maximum a posteriori hyposthesis (called MAP) [Dechter, 1996]. Suppose we select a set of variables d as *decision variables*, and the objective is to find values for the decision variables such that we obtain the maximum value of marginal probability $p(d)$. The notation p_i^d indicates that the function p_i has the decision variables fixed. With this notation, the objective of the MAP algorithm is to obtain:

$$\max_d \left[\sum_{x \notin d} \left(\prod_i p_i^d \right) \right]. \tag{2}$$

There are algorithms which attack this problem in the most general form [Dechter, 1996]. A class of algorithms attack this problem when decision problem can be represented as an influence diagram [Jensen & Jensen, 1994].

4 Quasi-Bayesian theory

Quasi-Bayesian theory uses convex sets of distributions to represent beliefs and to evaluate decisions [Giron & Rios, 1980]. Several other theories use probability intervals: inner/outer measures [Good, 1983; Halpern & Fagin, 1992; Ruspini, 1987; Suppes, 1974], lower probability theory [Breese & Fertig, 1991; Chrisman, 1996b; Fine, 1988; Smith, 1961]), convex Bayesianism [Kyburg Jr., 1987], Dempster-Shafer theory [Ruspini, 1987; Shafer, 1987], probability/utility sets [Seidenfeld, 1993].

Given a set of functions f_i, their convex combination is determined by $\sum_i \alpha_i f_i$, where all α_i are positive and $\sum \alpha_i = 1$. The convex combination of the elements of a set generates a convex set, which can be expressed as the convex hull of its vertices. We denote the convex hull of a finite set of functions $f_i(\cdot)$ by $\bigoplus_i f_i$.

The set of distributions maintained by an agent is called the *credal* set, and its existence is postulated on the grounds of axioms about preferences [Giron & Rios, 1980]. To simplify terminology, we use the term credal set only when it refers to a set of distributions containing more than one element. Convex sets of conditional distributions are used to represent conditional beliefs. Inference is performed by applying Bayes rule to each distribution in a prior credal set; the posterior credal set is the union of all posterior distributions. An introduction to technical aspects of Quasi-Bayesian theory, with a larger list of references, can be found at http://www.cs.cmu.edu/~fgcozman/qBayes.html.

We use two well-known results about posterior credal sets in this paper. First, to obtain a posterior credal set, one has to apply Bayes rule only to the vertices of a prior credal set and take the convex hull of the resulting distributions [Giron & Rios, 1980]. Second, to obtain maximum and minimum values of posterior probabilities, we must look only at the vertices of the posterior credal sets [Walley, 1991].

Given a convex set $\mathcal{K}$ of probability distributions, a probability interval can be created for every event A by defining lower and upper bounds:

$$\underline{p}(A) = \inf_{p \in \mathcal{K}} p(A) \qquad \overline{p}(A) = \sup_{p \in \mathcal{K}} p(A).$$

In the remainder of this paper, we will refer to maximization procedures for the determination of upper bounds, and leave implicit that similar minimization procedures lead to lower bounds. It is also possible to obtain lower bounds from upper bounds using the general identity $\underline{p}(A) = 1 - \overline{p}(A^c)$ for any event A.

Notice that a convex set of distributions always creates lower and upper bounds of probability, but it is *not* true that a set of lower and upper bounds of probability always defines a unique convex set of distributions. Much of the previous work on interval analysis of probability has foundered due to this fact. Application of Bayes rule essentially demands the construction of a convex set that obeys original prior intervals, but this construction is mathematically ill-defined. The Quasi-Bayesian approach sidesteps this difficulty by taking convex sets as basic entities.

5 Finitely generated convex sets of distributions

A finitely generated credal set is the convex hull of a finite number of probability distributions.

Consider the situation where a single variable x_1 is associated with a credal set, and the credal set for x_1 is the convex hull of m different distributions: $\left(\bigoplus_{j=1}^{m} p_{1,j}\right)$. In this case the set of distributions modeled by the network is:

$$\left(\bigoplus_{j=1}^{m} p_{1,j}\right) \prod_{i>1} p_i.$$

This credal set is convex, with m vertices. The vertices are:

$$r_j(\tilde{x}) = p_{1,j} \prod_{i>1} p_i.$$

Consider a Quasi-Bayesian network with two credal sets associated with variables x_1 and x_2. To produce a convex set of joint distributions, we use the convex hull:

$$\bigoplus_{j,k} \left(p_{1,j} p_{2,k} \prod_{i>2} p_i \right). \tag{3}$$

The "convexification" of the joint set of distributions does not affect the ordering of decisions obtained by expected utility. Any decision or inference in a Quasi-Bayesian procedure is obtained through expectation, and expectation is maximized/minimized only at the vertices of the credal set. Since the vertices of this convex hull are the same distributions generated by cross-multiplication of terms $p_{1,j}$ and $p_{2,k}$, we do not lose information by taking the convex hull operation.

This "convexification" technique is assumed in the remainder of the paper[1]. When a network with several credal sets is used, we take the convex hull of the combined credal sets.

[1]Note that there may be other ways of extending local credal sets to form a Quasi-Bayesian network with the same marginals.

6 Exact algorithms and the basic transformation

We start by considering exact algorithms. Cano et all. considered the same type of structures analyzed here, focusing more on the axiomatic aspects of inference, and producing explicit algorithms for networks with polytree topology [Cano, Delgado, & Moral, 1993]. Here we discuss how to generate exact algorithms for arbitrary networks associated with finitely generated credal sets.

The technique we use to generate exact algorithms is based on a transformation that operates on the Quasi-Bayesian network and produces a Bayesian network. This transformation is central to our exact and approximate algorithms.

6.1 The basic transformation

Start from a network where some variables are associated with credal sets. Call these variables z_i. For each variable z_i, suppose its credal set has m_i vertices $p_{i,j}$.

The basic transformation starts by modifying the original Bayesian network:

- For each variable z_i associated with a credal set:
  - Add a new variable z_i' with no parents to the network. The variable z_i' has z_i as its only child. The credal set for variable z_i has m_i vertices, then z_i' has m_i integer values:

$$\hat{z}_i' = \{1, \ldots, m_i\}.$$

  - Replace the variable z_i by a new variable z_i'' with the same values of z_i, all the parents of z_i plus z_i', and all the children of z_i. Define the distribution of z_i'' to be:

$$p(z_i''|\mathrm{pa}(z_i'')) = (p_{i,j}(z_i|\mathrm{pa}(z_i)) \text{ when } z_i' = j) \,.$$

Figure 2 illustrates the basic transformation. Call the variables z_i' *artificial* variables. The original network represents a finitely generated credal set, and each vertex in this credal set can be obtained by setting the artificial variables to arbitrary values. This is the crucial property of the basic transformation.

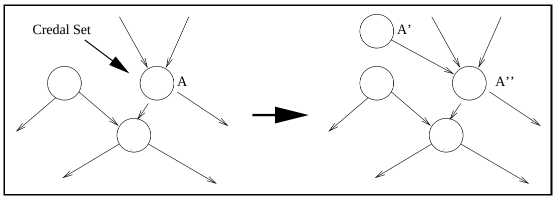

Figure 2: The basic transformation for the credal set associated with variable A

6.2 Exact algorithms

Consider a network with s credal sets, each credal set represented by m_i vertices. There are $M = \prod_{i=1}^{s} m_i$ possible vertices in the joint credal set. Based on the previous discussion, we have an exact algorithm for computation of posterior bounds as follows.

Algorithm

1. Initialize the lower and upper bounds for the posterior marginals with ones and zeros respectively.
2. Run any Bayesian inference algorithm on the transformed network, for each one of the M vertices of the original Quasi-Bayesian network. Each vertex is generated by fixing the values of the artificial variables. For each vertex, store the posterior marginal values that are smaller than the lower bounds, and the posterior marginal values that are larger than the upper bounds.

The algorithm has essentially the same memory requirements as usual Bayesian inferences. The drawback is that the algorithm does not use computations efficiently, wasting calculations by repeating the same inferences again and again. To speed up the algorithm, an implementation must cache parts of the network that have no credal sets, for example by forming cliques of those parts and caching intermediate results for such cliques. Based on this idea, a new algorithm can be devised which which trades memory requirements for speed. Instead of repeating computations, the following algorithm performs a single Bayesian inference in the transformed network.

Algorithm

1. Run any Bayesian inference algorithm on the transformed network, leaving the values of the artificial variables unspecified, such that they form a single cluster. The result is the conditional $p(x_q|\{z'_i\})$.

2. Initialize the lower and upper bounds for the posterior marginals with ones and zeros respectively.

3. For all combinations of values for the artificial variables, store the posterior marginal values that are smaller than the lower bounds, and the posterior marginal values that are larger than the upper bounds.

---

7 Complexity and generic approximations

As more and more variables are associated with credal sets, algorithmic complexity increases, since the number of vertices M increases exponentially with the number of credal sets. For exact algorithms, this fact either leads to exponential time complexity (first algorithm above) or exponential memory requirements (second algorithm above). To tackle the most complicated problems, we must use approximations.

One general strategy to produce approximations is use optimization algorithms that maximize a generic non-linear function defined over a discrete set. There are heuristic algorithms developed in the learning community and a variety of genetically-inspired algorithms that can be used as optimization tools for this purpose. One class of algorithms that can be useful for discrete optimization is the class of simulated annealing algorithms. The idea is to visit a set of configurations for the artificial variables and select only the configurations that are either better than the current configuration, or that are accepted based on a Gibbs distribution [Geman & Geman, 84; Press *et al.*, 1992]. The Gibbs distribution is regulated through a "temperature schedule", which is set heuristically. Casting the problem of robust inference as a general optimization problem is an interesting perspective for future research, but this paper concentrates on more specific approximations which use the structure of the networks for efficiency.

8 Approximate inferences through parameter estimation

In this section we recast the robust inference problem as a parameter estimation problem. Consider a Quasi-Bayesian network and the transformed Bayesian network with artificial variables $\{z'_i\}$. Each artificial variable has values $\{1, 2, \ldots, |\hat{z'_i}|\}$. Assume z'_i is a random variable with distribution $\theta_{ij} = p(z'_i = j)$. Call Θ the vector of all θ_{ij}.

Suppose x_q is a queried variable. The objective is to find:

$$\overline{p}(x_q = a|e) = \max_{\Theta} \left(\frac{\sum_{x \notin \{e, x_q = a\}} \prod_i p_i^{\{e, x_q = a\}}}{\sum_{x \notin e} \prod_i p_i^e} \right).$$

This problem is very similar to the problem of estimating the vector of parameters Θ given data e by maximum likelihood. Note that the optimizing vector must have, for each i, $\theta_{ij} = 1$ for some j. The difference between this and maximum likelihood parameter estimation is that the distribution to be optimized involves the posterior distribution instead of the prior. We can use learning techniques for this problem [Buntine, 1994]. Notice that the optimization procedure has to be repeated for each of the values of the queried variable x_q.

8.1 Gradient descent techniques

Optimization through gradient descent techniques, such as the conjugate gradient algorithm, have been used for learning in Bayesian networks. These algorithms benefit from the fact that the necessary gradient calculations can be computed with standard Bayesian network algorithms [Russell *et al.*, 1995].

To solve the robust inference problem, we must maximize the posterior log-likelihood for Θ (minimization is accomplished by maximizing the negative log-likelihood):

$$L(\Theta) = \log p(x_q = a|e) = \log \frac{p(x_q = a, e)}{p(e)},$$

which is:

$$L(\Theta) = \log p(x_q = a, e) - \log p(e).$$

The gradient of $L(\Theta)$ is obtained by computing, for each θ_{ij}:

$$\frac{\partial L(\Theta)}{\partial \theta_{ij}} = \frac{\partial \log p(x_q = a, e)}{\partial \theta_{ij}} - \frac{\partial \log p(e)}{\partial \theta_{ij}}.$$

Each term in the right-hand side of this expression can be written as $\frac{\partial \log p(e')}{\partial \theta_{ij}}$ where e' is some subset of the variables $\tilde{x}$. We have:

$$\frac{\partial \log p(e')}{\partial \theta_{ij}} = \frac{\partial \sum_j p(e'|z'_i = j) p(z'_i = j)}{\partial \theta_{ij}} \frac{1}{p(e')}.$$

Since $p(z'_i = j) = \theta_{ij}$ and this term appears only once in the summation, we have:

$$\frac{\partial \log p(e')}{\partial \theta_{ij}} = \frac{p(e'|z'_i = j)}{p(e')}.$$

and using Bayes rule:

$$\frac{\partial \log p(e')}{\partial \theta_{ij}} = \frac{p(z'_i = j|e') p(e')}{p(z'_i = j)} \frac{1}{p(e')} = \frac{p(z'_i = j|e')}{p(z'_i = j)}.$$

The final expression for the gradient of $L(\Theta)$ with respect to θ_{ij} is:

$$\frac{\partial L(\Theta)}{\partial \theta_{ij}} = \frac{p(z'_i = j|x_q = a, e)}{\theta_{ij}} - \frac{p(z'_i = j|e)}{\theta_{ij}}, \qquad (4)$$

which can be obtained through standard Bayesian network algorithms using local computations. A conjugate gradient descent can be constructed by selecting an initial value for Θ and, at each step, normalizing the values of Θ to ensure they represent proper distributions [Russell *et al.*, 1995].

8.2 Expectation Maximization (EM) algorithm

The EM algorithm produces a maximum likelihood estimate by maximizing the log-likelihood expression [Dempster, Laird, & Rubin, 1977]. To solve the robust inference problem, we must maximize the posterior log-likelihood $L(\Theta)$. We show how the original EM algorithm can be extended to a Quasi-Bayesian Expectation Maximization (QEM) algorithm and prove convergence properties.

The algorithm begins by assuming that the artificial variables are actual random quantities with distributions specified by Θ, and that we could even observe those variables as evidence. An initial estimate Θ^0 is assumed for Θ.

Suppose we had l sets of complete data for the transformed network, i.e., we had observed l trials for all variables in the network, including the artificial variables. The log-likelihood for this complete data would be:

$$L(\Theta) = \sum_{ijk} l_i(j,k) \log \theta_{ijk},$$

where $l_i(j,k)$ indicates the number of data points when the variable x_i is instantiated in its j value with its parents instantiated in their k value.

The first step of the QEM algorithm is to obtain the expected value of the log-likelihood given the evidence and assuming Θ^0 is correct:

$$Q(\Theta|\Theta^k) = E\left[\log\left(p(x_q = a, e)\right) - \log\left(p(e)\right)\right].$$

Suppose we wanted the expected value of $L(\Theta)$ for given evidence e. We have:

$$\begin{aligned} E\left[L(\Theta)\right] &= E\left[\sum_{ijk} l_i(j,k) \log \theta_{ij}\right] \\ &= \sum_{ijk} \left(\sum_{l} p(x_i, \mathrm{pa}(x_i)|e)\right) \log \theta_{ijk}. \end{aligned}$$

Based on this expression, we have:

$$\begin{aligned} Q(\Theta|\Theta^k) &= \sum_{ijk} p(x_i, \mathrm{pa}(x_i)|x_q = a, e) \log \theta_{ijk} \\ &- \sum_{ijk} p(x_i, \mathrm{pa}(x_i)|e) \log \theta_{ijk}. \end{aligned}$$

The second step of the QEM algorithm is to maximize $Q(\Theta|\Theta^k)$ for Θ. Only a few terms in the expression for $Q(\Theta|\Theta^k)$ will be free, since only the θ_{ij} for z'_i are estimated. Collecting these terms we obtain:

$$\sum_{ij} p(z'_i = j|x_q = a, e) \log \theta_{ij} - \sum_{ij} p(z'_i = j|e) \log \theta_{ij}, \qquad (5)$$

To perform maximization, use gradient descent with Θ^k as a starting point and ensure that at the end of the process we have $Q(\Theta^{k+1}|\Theta^k) > Q(\Theta^k|\Theta^k)$. The gradient is:

$$\frac{p(z'_i = j|x_q = a, e)}{\theta_{ij}} - \frac{p(z'_i = j|e)}{\theta_{ij}},$$

which can be obtained through standard Bayesian network algorithms. Note this is the expression used in the previous section to perform gradient descent optimization; in the QEM algorithm, we must ensure that $Q(\Theta^{k+1}|\Theta^k) > Q(\Theta^k|\Theta^k)$ to obtain convergence. Now set Θ^{k+1} to these values go to the next iteration.

Theorem 1 *The QEM algorithm produces a sequence that converges to a local maximum of $L(\Theta)$.*

Proof. We have:

$$H(\Theta'|\Theta) = -\sum_{x \notin \{x_q, e\}} \log\left(\frac{p(x, x_q|e, \Theta')}{p(x_q|e, \Theta')}\right) p(x|x_q, e, \Theta)$$

$$Q(\Theta'|\Theta) = \sum_{x \notin \{x_q, e\}} \log p(x, x_q|e, \Theta') p(x|x_q, e, \Theta).$$

Consequently:

$$L(\Theta') = Q(\Theta'|\Theta) + H(\Theta'|\Theta)$$

and (by [Dempster, Laird, & Rubin, 1977, Lemma 1]):

$$H(\Theta^{k+1}|\Theta^k) \geq H(\Theta^k|\Theta^k).$$

Since by construction we have $Q(\Theta^{k+1}|\Theta^k) > Q(\Theta^k|\Theta^k)$, we obtain:

$$\begin{aligned} L(\Theta^{k+1}) &= Q(\Theta^{k+1}|\Theta^k) + H(\Theta^{k+1}|\Theta^k) \\ &> Q(\Theta^k|\Theta^k) + H(\Theta^k|\Theta^k) \\ &= L(\Theta^k). \end{aligned}$$

The algorithm produces a monotonically increasing and bounded sequence, so the sequence converges to a local maximum of $L(\Theta)$.

9 Approximate inferences through Lavine's bracketing algorithm

Here we borrow a technique from robust Bayesian Statistics, called Lavine's algorithm [Lavine, 1991; Wasserman, 1992b]. We now adapt Lavine's technique to Bayesian networks with finitely generated credal sets.

9.1 The bracketing algorithm

In the most general case, we seek a posterior bound for the expected value of a function $u(\tilde{x})$, conditional on evidence e. For now we place no restriction to $u(\tilde{x})$; later we specialize it to the case of posterior probabilities. The quantity we seek is the upper bound for the expected value of $u(\tilde{x})$:

$$\overline{E}[u] = \max\left[\frac{\sum_{x \notin e} \left(u(\tilde{x}) \prod_i p_i^e\right)}{\sum_{x \notin e} \prod_i p_i^e}\right],$$

where p_i^e indicates that the evidence has been fixed in the distribution p_i. The maximization is with respect to all the distributions in the convex hull of the credal sets p_i.

Instead of attacking the maximization of $E[u]$ directly, Lavine's algorithm settles for deciding whether or not $\overline{E}[u]$ is larger than a given value k. When we obtain this result, we can construct a bracketing algorithm for $\overline{E}[u]$. The bracketing algorithm is the following. Pick a real number k and check whether $\overline{E}[u]$ is larger, smaller or equal to k, and respectively increase k, decrease k or stop. Repeat this until the solution was found or the bracketing interval is small enough. This algorithm is convergent and improves monotonically.

To decide whether or not $\overline{E}[u]$ is larger than k, some transformations are necessary. First, notice that $\overline{E}[u] > k$ if and only if:

$$\max\left[\frac{\sum_{x \notin e} \left((u(\tilde{x}) - k) \prod_i p_i^e\right)}{\sum_{x \notin e} \prod_i p_i^e}\right] > 0.$$

Second, notice that if this expression is larger than zero, then the maximum value of the numerator must be larger than zero; conversely, if the maximum value of the numerator is larger than zero, the expression is larger than zero. Hence Lavine's algorithm depends on the solution of a maximization problem for a given k:

$$\overline{E}_k[u] = \max\left[\sum_{x \notin e} \left((u(\tilde{x}) - k) \prod_i p_i^e\right)\right]. \qquad (6)$$

9.2 Probability bounds

To obtain the posterior probability for an event $\{x_q = a\}$ (x_q is the queried variable), we take:

- $u(\tilde{x}) = \delta_a(x_q)$, where $\delta_a(x_q)$ is 1 if $x_q = a$ and 0 otherwise. We have $E[\delta_a] = p(x_q = a|e)$, so Lavine's algorithm generates $\overline{p}(x_q = a|e)$.

- $u(\tilde{x}) = -\delta_a(x_q)$. We have $E[-\delta_a] = -p(x_q = a|e)$, so Lavine's algorithm generates $-\underline{p}(x_q = a|e)$.

To obtain the bounds on the posterior distribution of x_q, we have must Lavine's algorithm twice for each value of x_q. For each value of x_q, we must rerun Lavine's bracketing algorithm for $\delta_a(x_q)$ and $-\delta_a(x_q)$.

Lavine's algorithm is reduced to iterations which calculate:

$$\max \left[\sum_{x \notin e} \left((\delta_a(x_q) - k) \prod_i p_i^e \right) \right]. \qquad (7)$$

The maximization is with respect to all distributions in the joint credal set, but only the vertices of the joint credal set must be examined.

9.3 Efficient iterations for Lavine's algorithm

This section presents an algorithm for the calculation of expression (7). Start from a network where some variables are associated with credal sets and perform a basic transformation. For each variable z_i, suppose its credal set has vertices $p_{i,j}$. We describe a single iteration of Lavine's algorithm, for which the values of a and k have been fixed. Suppose x_q is associated with a single distribution p_q. Then multiply the distribution p_q by $(\delta_a(x_q) - k)$. Suppose x_q is associated with a credal set with vertices $p_{q,j}$. Then multiply the vertices $p_{q,j}$ by $(\delta_a(x_q) - k)$.

Now run a MAP algorithm as defined by expression (2), to determine the values of the decision variables z_i'. For the transformed network, the MAP solution will give:

$$\max_{\{z_i'\}} \left[\sum_{x \notin \{z_i', e\}} \left((\delta_a(x_q) - k) \prod_i p_i^{\{z_i', e\}} \right) \right], \qquad (8)$$

which is identical to expression (7). We now have all the elements to apply Lavine's algorithm.

9.4 Summary of the algorithm

Repeat the following for upper and lower bounds.

Algorithm

Transform the original network as explained in the previous section. For each value of the queried variable x_q, initialize k. To bracket k, iterate the following step.

Run MAP in the enlarged network; if the result is zero, stop. If not, bracket the interval of k values or stop if this interval is small enough.

9.5 When to use Lavine's bracketing algorithm

If we can solve the MAP problem in expression (8) efficiently, Lavine's algorithm becomes an efficient and provably convergent way of generating robust inferences. In general, solution of the MAP problem involves construction of a cluster of variables with all artificial variables, because in the worst case we cannot interchange any of the maximizations and summations in expression (8). This fact apparently limits the appeal of Lavine's bracketing algorithm, since it seems that in the worst case every one of its iterations may be as expensive as applying the exact algorithms in Section 6. However, there are three general situations where Lavine's algorithm can provide substantial savings:

- The network is partitioned by evidence.
- The transformed network can be cast as an influence diagram.
- The credal sets can be expressed as a set of linear inequalities.

First, suppose the evidence in a network is such that the joint distribution $p(\tilde{x})$ can be written as the product of two blocks, $p(\tilde{x}) = p_1(\tilde{x}_1)p_2(\tilde{x}_2)$. In this case the artificial variables in one of the blocks can be maximized independently of the variables in the other block. Note that, even though it is always true that such decompositions will reduce the complexity of usual Bayesian inference, the exact algorithms in Section 6 cannot benefit from it, because the denominator in Bayes rule mixes all distributions together.

Second, suppose the transformed network can be cast as an influence diagram; this requires a specific topology for the network. We must be able to divide the variables into m groups I_k, each group with the artificial variables Z_k associated with it, such that

$$p(I_k|I_0 \ldots I_{k-1}, D_1 \ldots D_m) = p(I_k|I_0 \ldots I_{k-1}, D_1 \ldots D_k);$$

if this condition is true, then efficient algorithms can be found to solve the MAP problem [Jensen & Jensen, 1994].

Finally, suppose the joint distribution can be written as:

$$p(\tilde{x}) = \left(\prod_{i>s} p(x_i|\mathrm{pa}(x_i)) \right) p(x_1, \ldots, x_s),$$

where $p(x_1, \ldots, x_s)$ is a convex set of multivariate distributions defined by a set of linear inequalities (the distribution $p(x_1, \ldots, x_s)$ can even be a conditional distribution). In this case we can drop the artificial variables and consider the direct maximization of expression (7) for each one of the iterations in Lavine's algorithm. Expression (8) then becomes a linear programming problem which can be solved by standard algorithms. The advantadge of this approach is that the number of vertices of the credal set may be much larger than the set of its defining inequalities, for example if the credal set is a density bounded class (analyzed in Section 10.5) or a density ratio class (analyzed in Section 10.7). Again, the linear structure of this problem can only be explored using Lavine's algorithm, as the direct application of Bayes rule produces a non-linear problem.

10 Classes of finitely generated convex sets of distributions

This sections demonstrates the generality of the previous algorithms by reducing a large number of classes of distributions to finitely generated credal sets. Inferences with each class can be performed using the algorithms in this paper as long as the classes are formulated as finitely generated convex sets. We first present the ϵ-contaminated and lower density bounded classes due to their simplicity. Then the belief function class is analyzed. All other classes admit simple reductions to the belief function class, except the density ratio classes which requires special treatment.

Classes of distributions are used to define the prior distribution $p(x)$ for a variable x. These results can always be extended to conditional distributions of the form $p(x|\mathrm{pa}(x))$. In this case, the vertices of the joint distribution are defined in the following way. For $\mathrm{pa}(x) = c$, define the vertices of the distribution $r(x|\mathrm{pa}(x) = c)$; the vertices of the joint distribution are defined by all combinations of $r_j(x|\mathrm{pa}(x))$. Since there are $|\mathrm{pa}(x)|$ possible values for c, there are $|\hat{x}|^{|\mathrm{pa}(x)|}$ vertices for the joint distribution.

10.1 The ϵ-contaminated Class

An ϵ-contaminated class is characterized by a distribution $p(\cdot)$ and a real number $\epsilon \in (0, 1)$ [Berger, 1985]:

$$r(x) = (1 - \epsilon)p(x) + \epsilon q(x). \tag{9}$$

This model is appropriate when we expect $p(x)$ to be correct a fraction $(1 - \epsilon)$ of the time, but otherwise any distribution is possible.

An ϵ-contaminated class for a variable without parents is the convex hull of the functions:

$$r_{a_k}(x) = (1 - \epsilon)p(x) + \epsilon\delta_{a_k}(x), \quad \text{for all } a_k \in \hat{x},$$

where $\hat{x}$ is the set of values of x and $\delta_a(x)$ is 1 if $x = a$ and 0 otherwise.

The exact and iterative methods derived before apply, but the fact that the vertices of an ϵ-contaminated class are exclusively composed of zeros and ones leads to computational savings, because many multiplications of real numbers can be avoided.

The simple structure of the ϵ-contaminated class makes it possible to obtain closed-form expressions for the expected bounds on a function $u(x)$ in the univariate case. The expected value with respect to distribution $r_{a_k}(x)$ is $(1 - \epsilon)E[u] + \epsilon a_k$, so:

$$\underline{E}[u] = (1 - \epsilon)E[u] + \epsilon\underline{a}_k$$
$$\overline{E}[u] = (1 - \epsilon)E[u] + \epsilon\overline{a}_k,$$

where $\underline{a}_k$ and $\overline{a}_k$ are the minimum and maximum values of x respectively.

10.2 The lower density bounded class

Consider the set of all distributions $p(x)$ such that

$$p(x) \geq l(x),$$

where $l(\cdot)$ is an arbitrary non-negative function whose sum is smaller than one [Breese & Fertig, 1991].

Call this class the *lower density bounded class.* There are approximations (without error bounds) for inferences with this class [Breese & Fertig, 1991]. The results in this paper improve this to exact and provably convergent inference algorithms. The key fact is that this class is identical to the ϵ-contaminated class since it can be specified as the set of distributions:

$$r(x) = (1 - (1 - L))\frac{l(x)}{L} + (1 - L)q(x),$$

where $L = \sum l(x)$ and $q(x)$ is an arbitrary distribution. A lower density bounded class is the convex hull of:

$$r_{a_k}(x) = l(x) + (1 - L)\delta_{a_k}(x), \quad \text{for all } a_k \in \hat{x}.$$

10.3 The belief function class

Consider a discrete variable x with a finite set of values $\hat{x}$. To define a probability distribution we must specify $|\hat{x}|$ numbers, one for each element of $\hat{x}$. Suppose we cannot specify precise numbers for those events, so we distribute some probability mass into subsets of $\hat{x}$. For example, we place 0.1 mass in the set $\{x_1, x_2\}$ without specifying whether the mass goes to x_1 or x_2. Call the mass associated to subset A a *basic mass assignment.* Since the mass that is distributed in $\hat{x}$ must add to one, the basic mass assignments must add to one.

The basic mass assignments define a convex set of distributions. Since the value of $m(A)$ corresponds to a probability mass that can be arbitrarily spread among the subsets of A, we can generate vertices of the convex set by concentrating the non-zero basic mass assignments into each one of their subsets, one at a time. There are finitely many ways to distribute the mass among subsets, because x has finitely many values, so the convex set defined by the basic mass assignments is finitely generated. When the non-zero basic mass assignments carry mass to single elements of $\hat{y}$, the convex set collapses to a standard Bayesian model.

A belief function can be defined from the basic mass assignment [Shafer, 1976]:

$$\text{Bel}(A) = \sum_{B \subset A} m(B)$$

A belief function contains the same information carried by a basic mass assignment, which is called the Möbius transform of the belief function. Mathematically, the Möbius transform of a belief function $\text{Bel}(\cdot)$ is:

$$m(A) = \sum_{B \subset A} (-1)^{|A-B|}\text{Bel}(B),$$

and there are algorithms to translate between belief functions and basic mass assignments [Kennes, 1992]. The value of $m(A)$ corresponds to a probability mass that can be arbitrarily spread among the subsets of A. To generate the vertices of the credal set, we must concentrate the non-zero values of the Möbius transform into each one of their subsets, one at a time.

Usually a belief function is associated with a plausibility function:

$$\text{Pl}(A) = 1 - \text{Bel}(A^c).$$

The convex set of distributions defined by the associated basic mass assignment can be expressed in terms of the belief and plausibility functions, as the unique convex set of distributions $p(\cdot)$ such that [Wasserman, 1990]:

$$\text{Bel}(A) \leq p(A) \leq \text{Pl}(A).$$

Application of Bayes rule to belief function classes is a difficult problem given the non-linear character of Bayes rule. There are expressions that generate exact bounds for posterior quantities in univariate distributions [Halpern & Fagin, 1992], but the same does not apply to marginalization of multivariate belief function classes[2]. Our solution is to reduce belief function classes to finitely generated convex sets of distributions and apply the methods derived before.

A different way to represent a belief function class is through a set of inequalities. For each subset A with a non-zero basic mass assingment, a inequality can be written:

$$p(A) \geq \text{Bel}(A).$$

The set of all such inequalities defines the belief function class.

[2]Belief and plausibility functions have been efficiently used in Dempster-Shafer theory [Shenoy & Shafer, 1990], where inference is *not* performed through Bayes rule.

10.4 The sub-sigma class

Consider a variable x with values $\hat{x}$. Sometimes it is hard to specify all the numbers $p(x)$ for all elements of $\hat{x}$; a more straightforward approach is to specify probability masses for non-overlapping subsets of $\hat{x}$. This characterizes a convex set of distributions for x called a sub-sigma class [Berger, 1990; Lambert & Duncan, 1986; Manski, 1981].

For example, suppose $\hat{x}$ has 20 values between zero and one, and one wants to specify a distribution for x that is precise for the values around 0.5 but rather imprecise in the tails. One might consider specifying a single probability mass for the lower 5 values, a single probability mass for the upper 5 values, and probability masses for each of the 10 intermediate values.

Every sub-sigma class is a belief function class, since the Möbius transform of a sub-sigma class can be directly read from the the specification of the class. Results of the previous section can be applied to this class.

10.5 The density bounded class

A density bounded class [Lavine, 1991] is the set of all distributions $p(x)$ such that

$$l(x) \leq p(x) \leq u(x),$$

where $l(\cdot)$ and $u(\cdot)$ are arbitrary non-negative measures such that $\sum_x l(x) \leq 1$ and $\sum_x u(x) \geq 1$. The particular case of $u(x) \equiv 1$ leads to the lower density bounded class, but the method used before in the lower density bounded class does not apply to the more general setting.

10.6 The total variation class

The total variation class is the set of distributions $p(x)$ such that [Wasserman, 1992b]:

$$|p(A) - r(A)| \leq \epsilon \quad \text{for any event } A,$$

where $r(x)$ is a given probability distribution. This class forms the neighborhood of all distributions close to $r(x)$ up to ϵ. The definition above defines a set of inequalities that characterizes a class, so application of Lavine's algorithm with linear programming is possible. This class presents efficiency problems since even this straighforward translation into inequalities requires a substantial amount of space (there are $2^{|\hat{x}|}$ inequalities, one for each subset of $\hat{x}$).

10.7 The density ratio class

A density ratio class consists of all probability densities $p(A)$ so that for any event A [DeRobertis & Hartigan, 1981]:

$$l(A) \leq \alpha p(A) \leq u(A)$$

where $l(A)$ and $u(A)$ are arbitrary positive measures such that $l(\cdot) \leq u(\cdot)$ and α is some positive real number. Notice that the distributions defined by a density ratio class form a convex set.

If we draw two graphs, one for $l(\cdot)$ and one for $u(\cdot)$, then all functions that we can draw between them will be in this family. The definition fixes the "shape" boundaries of the class, but leaves unspecified the particular values that can be taken by distributions; to obtain these values we have to normalize the elements of the class.

The density ratio class is probably the easiest to elicit in a Bayesian network setup. Consider the following equivalent definition [Berger, 1990]: a density ratio class consists of all probability densities $p(A)$ so that, for any events A and B:

$$\frac{l(A)}{u(B)} \leq \frac{p(A)}{p(B)} \leq \frac{u(A)}{l(B)}. \tag{10}$$

The class is defined by intervals of *probability odds*: the ratio between the probability of two events. Odds have been advocated for the elicitation of Bayesian networks, as the most natural method to obtain probability judgements from experts [Heckerman, 1990]. It is a short step to accept that intervals of odds can be used to model imprecise judgements of probability.

Expression (10) also indicates how the specification of a density ratio class can be turned into a set of linear inequalities of the form:

$$p(A) \leq \frac{u(A)}{l(B)} p(B), \tag{11}$$

for arbitrary A and B.

The density ratio class is attractive in univariate models because posterior densities are obtained through operations on the measures $l(A)$ and $u(A)$ [DeRobertis & Hartigan, 1981]. This computational advantage breaks in multivariate settings as in general we cannot take marginals only on the lower and upper measures [Wasserman, 1992a], since for every variable x with $|\hat{x}|$ values, there are $|\hat{x}||\hat{x}-1|$ probability inequalities represented by expression (11).

To produce inferences with this class, we use the following result [Wasserman & Kadane, 1992]. A density ratio class defined by $l(\cdot)$ and $u(\cdot)$ can be reduced to the union of all density bounded classes defined by $\beta l(\cdot)$ and $\beta u(\cdot)$ as $\beta \in [1/(\sum_x u(x)), 1/(\sum_x l(x))]$. Suppose we obtain the bounds for all these density bounded classes; in practice we must discretize the interval of β values. The upper bound for the density ratio class will be the maximum of the upper bounds for the density bounded classes, and the lower bound for the density ratio class will be the minimum of the lower bounds for the density bounded classes.

11 Expected utility and variance

For simplicity, this paper concentrated on inferences. Decision making involves the calculation of expected utilities rather than inferences, given a *utility* function $u(\cdot)$. A set of *decision* variables d is selected and the objective is to obtain bounds on expected utility for fixed values of d. If $u(x_q) = x_q$ with no decision variables, the result of expected utility calculations is the expected value of x_q.

We can extend the framework presented in this paper to utility functions. The exact algorithms extend readily: instead of calculating posterior marginals, calculate the solution of the MEU problem. Gradient-based search can also be effortlessly adapted to utilities; the QEM convergence proof may not apply however if $u(\cdot)$ is negative. Lavine's algorithm, given as expression (6), is valid for any utility function $u(\tilde{x})$.

Another useful measure for probabilistic inference is the variance of a variable x_q, defined as $V_p[x_q] = E[x_q^2] - (E[x_q])^2$ for a given probability distribution $p(x_q)$. Define the lower and upper variance respectively as:

$$\underline{V}[x_q] = \min_p V_p[x_q] \qquad \overline{V}[x_q] = \max_p V_p[x_q]$$

Calculation of bounds for variances in a Quasi-Bayesian network is a great challenge because the expression for $V_p[x_q]$ is quadratic on $p(x_q)$. Approximations for the lower and upper variances can be made through generic optimization algorithms, such as gradient descent or simulated annealing, but convergence properties are lost.

To produce a convergent algorithm for calculation of lower and upper variances, we can use Walley's variance envelope theorem [Walley, 1991, Theorem G2], which demonstrates that $\underline{V}[x_q] = \min_\mu(\underline{E}[(x_q - \mu)^2])$ and $\overline{V}[x_q] = \min_\mu(\overline{E}[(x_q - \mu)^2])$. The calculation of lower and upper variances becomes a unidimensional optimization problem, which can be solved by discretizing μ (note that μ must be larger than zero and smaller than the square of the largest value of x_q). The computational burden of this procedure is very intense since for each value of μ it is necessary to obtain the bounds for expected value of $u(x_q) = (x_q - \mu)^2$.

12 Conclusion

We presented exact solutions and iteratively convergent approximations for inferences in Bayesian models associated with finitely generated convex sets of probability distributions. Almost all classes of distributions used in robust Statistics were reduced to the class of finitely generated convex sets.

Robust Bayesian analysis is a field with clear practical relevance. Any application of Bayesian networks must determine the relationship between the accuracy of probability values and the accuracy of inferences. Yet research on Bayesian networks has not fully explored this aspect of inference, due to the lack of algorithms for inferences with sets of distributions. This paper offers a solution to this problem, opening the field of robust Bayesian Statistics to graphical approaches.

There are several trade-offs that can be exercised when deciding which algorithm to use. Typically, a network is built or estimated, and then some critical variables and links are analyzed with respect to robustness. This process will rarely involve the consideration of too many credal sets at a time; more often, different portions of the network and structural assumptions will be analyzed carefully and separately. Exact algorithms may be feasible in most such cases where a few credal sets are under consideration.

Even when a small number of credal sets are present, there may be difficulties in running exact algorithms if each one of the credal sets is represented by a large number of vertices. In some cases, a representation in terms of linear inequalities may be much more compact than the enumeration of vertices. These cases are best handled with Lavine's algorithm and linear programming.

Finally, for the situations where many large credal sets are present, the QEM algorithm is indicated as it is the first algorithm for numeric inferences in Quasi-Bayesian networks with provably convergent behavior.

Acknowledgements

We thank Eric Krotkov for substantial support during the research that led to this work, and Lonnie Chrisman for reading an earlier draft and suggesting substantial improvements.

A Exact solutions for the ϵ-contaminated class

In the particular case of a single ϵ-contaminated variable x_1 without parents, queries about x_1 can be generated in closed-form. To generate upper bounds in this case we solve:

$$\overline{r}(x_1 = a|e) = \max \left(\frac{m(a,e)\left((1-\epsilon)p_1(a) + \epsilon q_1(a)\right)}{\sum_{x_1} m(x_1,e)\left((1-\epsilon)p_1 + \epsilon q_1\right)} \right),$$

where:

$$m(a,e) = \sum_{x \notin \{x_1, e\}} \left(\prod_{i>1} p_i^{\{x_1=a,e\}} \right).$$

The maximizing distribution puts all mass in $x_1 = a$:

$$\overline{r}(x_1 = a|e) = \left(\frac{(1-\epsilon)m(a,e)p_1(a) + \epsilon m(a,e)}{\sum_{x_1}\left((1-\epsilon)m(x_1,e)p_1\right) + \epsilon m(a,e)} \right). \tag{12}$$

Calculation of expression (12) is immediate given a variable elimination solution for a network using p_1.

Suppose now we have a Bayesian network with a single contamination at variable x_1 without parents. To query another variable x_2, we must calculate:

$$\overline{r}(x_2 = b|e) = \max_{x_1} \left[\frac{(1-\epsilon)m(b,e) + \epsilon n(x_1,b,e)}{\sum_{x_2}\left((1-\epsilon)m(b,e) + \epsilon n(x_1,b,e)\right)} \right],$$

where

$$n(a,b,e) = \sum_{x \notin \{x_1,x_2,e\}} \left(\prod_{i>1} p_i^{\{x_1=a,x_2=b,e\}} \right),$$

$$m(b,e) = \sum_{x_1} \left(n(x_1,b,e)p_1 \right).$$

This expression can be computed as soon as we have $p(x_1, x_2)$ in a variable elimination scheme. Each element of $p(x_1, x_2)$ is equal to $n(a,b,e)$ and the summation with respect to x_2 is $m(b,e)$.

If x_1 does have parents, the situation gets more complex as we have to calculate:

$$\overline{r}(x_2 = b|e) = \max_{q_1} \left[\frac{l(b,e)}{\sum_{x_2} l(x_2,e)} \right],$$

where

$$l(b,e) = (1-\epsilon)m(b,e) + \epsilon \sum_{x \in M, x \notin N} q_1 n(x_1, \mathrm{pa}(x_1), b, e)$$

$$n(a,c,b,e) = \sum_{x \notin \{M,N\}} \left(\prod_{i>1} p_i^{\{x_1=a,x_2=b,\mathrm{pa}(x_1)=c,e\}} \right),$$

$$m(b,e) = \sum_{M} \left(n(x_1, \mathrm{pa}(x_1), b, e)p_1 \right).$$

$$M = \{x_1, \mathrm{pa}(x_1)\}$$

$$N = \{x_2, e\}$$

Here the calculations will involve generation of all values of $n(\cdot)$ and $m(\cdot)$, a potential source of complexity. In large networks, approximation algorithms will be the most effective solution.

References

[Berger, 1985] Berger, J. O. 1985. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag.

[Berger, 1990] Berger, J. O. 1990. Robust bayesian analysis: Sensitivity to the prior. *Journal of Statistical Planning and Inference* 25:303–328.

[Breese & Fertig, 1991] Breese, J. S., and Fertig, K. W. 1991. Decision making with interval influence diagrams. In Bonissone, P. P.; Henrion, M.; Kanal, L. N.; and Lemmer, J. F., eds., *Uncertainty in Artificial Intelligence 6*. North-Holland: Elsevier Science. 467–478.

[Buntine, 1994] Buntine, W. L. 1994. Operations for learning with graphical models. *Journal of Artificial Intelligence Research* 2:159–225.

[Cannings & Thompson, 1981] Cannings, C., and Thompson, E. A. 1981. *Genealogical and Genetic Structure.* Cambridge: Cambridge University Press.

[Cano, Delgado, & Moral, 1993] Cano, J.; Delgado, M.; and Moral, S. 1993. An axiomatic framework for propagating uncertainty in directed acyclic networks. *International Journal of Approximate Reasoning* 8:253–280.

[Chrisman, 1996a] Chrisman, L. 1996a. Independence with lower and upper probabilities. *Proc. Twelfth Conference Uncertainty in Artificial Intelligence* 169–177.

[Chrisman, 1996b] Chrisman, L. 1996b. Propagation of 2-monotone lower probabilities on an undirected graph. *Proc. Twelfth Conference Uncertainty in Artificial Intelligence* 178–186.

[Dechter, 1996] Dechter, R. 1996. Bucket elimination: A unifying framework for probabilistic inference. *Proc. of the Twelfth Conference Uncertainty in Artificial Intelligence* 211–219.

[Dempster, Laird, & Rubin, 1977] Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Statistical Society B* 44:1–38.

[DeRobertis & Hartigan, 1981] DeRobertis, L., and Hartigan, J. A. 1981. Bayesian inference using intervals of measures. *The Annals of Statistics* 9(2):235–244.

[Fine, 1988] Fine, T. L. 1988. Lower probability models for uncertainty and nondeterministic processes. *Journal of Statistical Planning and Inference* 20:389–411.

[Geman & Geman, 84] Geman, S., and Geman, D. 84. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Transactios on Patter Analysis and Machine Intelligence* PAMI-6(6):721–741.

[Giron & Rios, 1980] Giron, F. J., and Rios, S. 1980. Quasi-bayesian behaviour: A more realistic approach to decision making? In Bernardo, J. M.; DeGroot, J. H.; Lindley, D. V.; and Smith, A. F. M., eds., *Bayesian Statistics.* Valencia, Spain: University Press. 17–38.

[Good, 1983] Good, I. J. 1983. *Good Thinking: The Foundations of Probability and its Applications.* Minneapolis: University of Minnesota Press.

[Ha & Haddawy, 1996] Ha, V., and Haddawy, P. 1996. Theoretical foundations for abstraction-based probabilistic planning. *Proc. Twelfth Conference Uncertainty in Artificial Intelligence* 291–298.

[Halpern & Fagin, 1992] Halpern, J. Y., and Fagin, R. 1992. Two views of belief: Belief as generalized probability and belief as evidence. *Artificial Intelligence* 54:275–317.

[Heckerman, 1990] Heckerman, D. 1990. An empirical comparison of three inference methods. In Shachter, R. D.; Kanal, L. N.; and Lemmer, J. F., eds., *Uncertainty in Artificial Intelligence 4*. North-Holland: Elsevier Science Publishers. 283–303.

[Huber, 1980] Huber, P. J. 1980. *Robust Statistics.* New York: Wiley.

[Jensen & Jensen, 1994] Jensen, F., and Jensen, F. V. 1994. From influence diagrams to junction trees. Technical Report R94-2013, Aalborg University.

[Jensen, 1996] Jensen, F. V. 1996. *An Introduction to Bayesian Networks.* New York: Springer Verlag.

[Kadane, 1984] Kadane, J. B. 1984. *Robustness of Bayesian Analyses*, volume 4 of *Studies in Bayesian econometrics.* New York: Elsevier Science Pub. Co.

[Kennes, 1992] Kennes, R. 1992. Computational aspects of the mŏbius transformation of graphs. *IEEE Transactions on Systems, Man and Cybernetics* 22(2):201–223.

[Kyburg Jr., 1987] Kyburg Jr., H. E. 1987. Bayesian and non-bayesian evidential updating. *Artificial Intelligence* 31:271–293.

[Lambert & Duncan, 1986] Lambert, D., and Duncan, G. T. 1986. Single-parameter inference based on partial prior information. *The Canadian Journal of Statistics* 14(4):297–305.

[Lavine, 1991] Lavine, M. 1991. Sensitivity in bayesian statistics, the prior and the likelihood. *Journal of the American Statistical Association* 86(414):396–399.

[Levi, 1980] Levi, I. 1980. *The Enterprise of Knowledge*. Cambridge, Massachusetts: The MIT Press.

[Manski, 1981] Manski, C. F. 1981. Learning and decision making when subjective probabilities have subjective domains. *The Annals of Statistics* 9(1):59–65.

[Pearl, 1988] Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kauffman.

[Press *et al.*, 1992] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; and Flannery, B. P. 1992. *Numerical Recipes in C*. Cambridgeshire: Cambridge University Press.

[Ruspini, 1987] Ruspini, E. H. 1987. The logical foundations of evidential reasoning. Technical Report SRIN408, SRI International.

[Russell *et al.*, 1995] Russell, S.; Binder, J.; Koller, D.; and Kanazawa, K. 1995. Local learning in probabilistic networks with hidden variables. *Proc. Fourteenth International Joint Conference on Artificial Intelligence*.

[Seidenfeld & Schervish, 1990] Seidenfeld, T., and Schervish, M. 1990. Two perspectives on consensus for (bayesian) inference and decisions. *IEEE Transactions on Systems, Man and Cybernetics* 20(1).

[Seidenfeld, 1993] Seidenfeld, T. 1993. Outline of a theory of partially ordered preferences. *Philosophical Topics* 21(1):173–188.

[Shafer, 1976] Shafer, G. 1976. *A mathematical theory of evidence*. Princeton University Press.

[Shafer, 1987] Shafer, G. 1987. Probability judgment in artificial intelligence and expert systems. *Statistical Science* 2(1):3–44.

[Shenoy & Shafer, 1990] Shenoy, P. P., and Shafer, G. 1990. Axioms for probability and belief-function propagation. In Shachter, R. D.; Levitt, T. S.; Kanal, L. N.; and Lemmer, J. F., eds., *Uncertainty in Artificial Intelligence 4*. North-Holland: Elsevier Science Publishers. 169–198.

[Smith, 1961] Smith, C. A. B. 1961. Consistency in statistical inference and decision. *Journal Royal Statistical Society B* 23:1–25.

[Suppes, 1974] Suppes, P. 1974. The measurement of belief. *Journal Royal Statistical Society B* 2:160–191.

[Tessem, 1992] Tessem, B. 1992. Interval probability propagation. *International Journal of Approximate Reasoning* 7:95–120.

[Walley, 1991] Walley, P. 1991. *Statistical Reasoning with Imprecise Probabilities*. New York: Chapman and Hall.

[Wasserman & Kadane, 1992] Wasserman, L., and Kadane, J. B. 1992. Computing bounds on expectations. *Journal of the American Statistical Association* 87(418):516–522.

[Wasserman, 1990] Wasserman, L. A. 1990. Prior envelopes based on belief functions. *The Annals of Statistics* 18(1):454–464.

[Wasserman, 1992a] Wasserman, L. 1992a. Invariance properties of density ratio priors. *The Annals of Statistics* 20(4):2177–2182.

[Wasserman, 1992b] Wasserman, L. 1992b. Recent methodological advances in robust bayesian inference. In Bernardo, J. M.; Berger, J. O.; Dawid, A. P.; and Smith, A. F. M., eds., *Bayesian Statistics 4*. Oxford University Press. 483–502.

[Zhang & Poole, 1996] Zhang, N. L., and Poole, D. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research* 301–328.