Using Sensitivity Analysis for Selective Parameter Update in Bayesian Network Learning

Haiqin Wang*
Mathematics and Computing Technology
the Boeing Company
P.O. Box 3707, MC 7L-44
Seattle, WA 98124
haiqin.wang@boeing.com

Irina Rish, Sheng Ma
IBM T.J. Watson Research Center
30 Saw Mill River Rd
Hawthorn, NY 10532
{rish,shengma}@us.ibm.com

Abstract

The process of building a Bayesian network model is often a bottleneck in applying the Bayesian network approach to real-world problems. One of the daunting tasks is the quantification of the Bayesian network that often requires specifying a huge number of conditional probabilities. On the other hand, the sensitivity of the network's performance to variations in different probability parameters may be quite different; thus, certain parameters should be specified with a higher precision than the others. We present a method for a selective update of the probabilities based on the results of sensitivity analysis performed during learning a Bayesian network from data. We first perform the sensitivity analysis on a Bayesian network in order to identify the most important (most critical) probability parameters, and then further update those probabilities to more accurate values. The process is repeated until refining the probabilities any further does not improve the performance of the network. Our method can also be used in active learning of the Bayesian networks, in which case the sensitivity can be used as a criterion guiding active data selection.

Introduction

Bayesian networks provide a graphical framework for compact representation of multivariate probabilistic distributions and efficient reasoning under uncertainty. Graphical probabilistic models are widely used in various applications, including medical diagnosis, computer troubleshooting, traffic control, airplane failure isolation, speech recognition, and error-correcting codes, to name a few. However, Bayesian network construction is often considered a major difficulty when applying this framework to real-world problems. One of the daunting tasks is the quantification of the Bayesian network that often requires specification of thousands of conditional probabilities, as the probability matrix for each node is exponential in the number of its parents.

*Ph.D candidate in Intelligent Systems Program at University of Pittsburgh. The work was done at IBM T.J. Watson Research Center.

One way to acquire the probability distributions is to elicit the probability parameters by interviewing domain experts. However, such knowledge-engineering approach can be quite expensive and time-consuming. Another way is to estimate the parameters from the available data. Unfortunately, the typically huge number of probability parameters in a Bayesian network may require quite large data sets in order to learn accurate parameter estimates, especially for probability distributions describing rare events. In real-life applications, the data base is often too scarce and results in erroneous values for the rare-event probabilities. The third way utilizes the domain knowledge as well as the data, and becomes the standard method for estimating probability distributions in Bayesian learning. It views the prior knowledge of a domain expert as an equivalent of a pseudo (or imaginary) data set which observes Dirichlet distributions (Geiger & Heckerman 1995). The Dirichlet exponent parameters (also called hyperparameters) are used to represent the equivalent sample size of the experts' prior knowledge (Cooper & Herskovits 1992). However, the number of the hyperparameters is as large as the number of the probability parameters in a Bayesian network.

On the other hand, not all parameters are equally important since they all have different effects on the network's performance. Sensitivity analysis can identify the most important parameters. In the past few years, significant progress has been made in developing sensitivity analysis techniques for Bayesian networks. Efficient sensitivity analysis methods based on inference algorithms have been provided in (Coupé *et al.* 2000; Darwiche 2000; Kjærulff & van der Gaag 2000). Those techniques have been used for efficient quantification of large-scale Bayesian networks (Coupé *et al.* 1999).

In this paper, we present a method that uses sensitivity analysis for a selective update of the probabilities when learning a Bayesian network. We first run sensitivity analysis on a Bayesian network learned with uniform hyperparameters to identify the most important probability parameters. Then we update this set of probabilities to their accurate values by acquiring their informative hyperparameters. The process is repeated until further elaboration of probabilities does

not improve the performance of the network. Our method can also be applied to active learning of Bayesian networks, where the sensitivity analysis can suggest which data should be collected or selected for further learning.

Bayesian Network

Bayesian networks (also called *belief networks*) provide a increasingly popular graphical framework for Bayesian reasoning, a probabilistic approach to inference based on combining prior knowledge with observed data using the *Bayes' rule*:

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)}, \qquad (1)$$

where $P(H)$ is the *prior* probability of hypothesis H, $P(D)$ is the prior probability of observing data D, $P(D|H)$ (called *likelihood*) is the probability of observing D if hypothesis H holds, and $P(H|D)$ is the *posterior* probability of H after observing data D.

Formally, a Bayesian network B is a pair (G, Θ), where G is a directed acyclic graph in which nodes represent random variables of interest (e.g., the temperature of a device, the gender of a patient, a feature of an object, an occurrence of an event) and the edges denote probabilistic dependencies. Since the directed edges are often interpreted as direct causal influences between the variables, Bayesian networks are also called *causal networks*. Let $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$ be a set of random variables, and let $\Theta = \{\theta_{x_i,\mathbf{pa_i}}\}$ be the set of parameters that represent conditional probabilities for each node X_i given its parents $\mathbf{Pa_i}$ (the nodes pointing to X_i in the graph) in G, i.e. $\theta_{x_i,\mathbf{pa_i}} = P(X_i = x_i|\mathbf{Pa_i} = \mathbf{pa_i})$ (or, using a shorter notation, $P(x_i|\mathbf{pa_i})$). The distributions $P(X_i|\mathbf{Pa_i})$, associated with each node X_i, are called *local probability distributions* (Heckerman 1998). Typically, Bayesian networks are defined for discrete variables with finite number of states. Thus, the local probability distributions are represented by $(m+1)$-dimensional *conditional probability tables (CPTs)*, where m is the number of parents, and each entry $\theta_{x_i,\mathbf{pa_i}}$ corresponds to a particular value assignment to X_i and its parents. A Bayesian network represents a joint probability distribution over $\mathbf{X}$ as a product of local distributions:

$$P(x_1, ..., x_n) = \Pi_{i=1}^{n} P(x_i|\mathbf{pa_i}) \; . \qquad (2)$$

Exploiting conditional independence assumptions allows a compact representation of multivariate probabilistic distributions and allows for efficient reasoning techniques.

Figure 1 shows an example of Bayesian network. It is a small fragment of *HEPAR* (Oniśko, Druzdzel, & Wasyluk 2000) network built for medical diagnosis for liver diseases. The causal relationship between liver disorder to possible causes (e.g., gallstone, alcoholism) and to symptoms (e.g., fatigue, jaundice) can be read directly from the links in the graph. In this network, node *Disorder* has three binary parents: *Alcoholism*, *Hepatotoxic medications*, and *Gallstones*, each of which is a causal factor contributing to each of six possible liver disorders. There are totally 48 probability parameters to define node *Disorder* conditioned on its parent configurations. For a root node (i.e., a node having no parents), the prior probability distribution is defined over the node's outcomes. *HEPAR* includes 94 variables and requires over 3,700 numerical parameters for its full quantification.

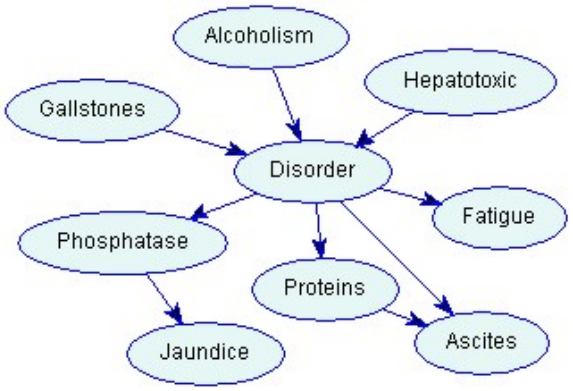

Figure 1: An example of a Bayesian network.

Traditionally, Bayesian networks have been used as a knowledge-engineering tool for representing uncertain expert knowledge and for subsequent reasoning under uncertainty. However, the process of building and debugging a Bayesian network is recognized as a major difficulty in applying this approach to real-world problems. Over the last decade, the research focus is shifting more towards learning Bayesian networks from data, especially with increasing volumes of data available in biomedical, internet, and e-business applications. In the past few years, significant progress has been made in developing techniques for learning Bayesian networks (Heckerman 1998). Most recently, there is a growing interest to the reliability (sensitivity) of Bayesian networks in the presence of noise in its parameters, and to the validation of Bayesian network models (Pradhan *et al.* 1996; Kipersztok & Wang 2001) by using sensitivity analysis techniques. Based on reasoning algorithms for probabilistic inference, efficient computation methods have been developed for sensitivity analysis (Coupé *et al.* 2000; Darwiche 2000; Kjærulff & van der Gaag 2000) and made the technique applicable to quantifying large-scale real-world Bayesian networks (Coupé *et al.* 1999). For a brief introductory survey on the advances in Bayesian network learning, see (Rish 2000); for a comprehensive one, refer to (Heckerman 1998).

Learning Probability Parameters

Assume that we have a *complete* (no missing values) data set $D = \{\mathbf{d^1}, \ldots, \mathbf{d^N}\}$ over a set of discrete, multinomial variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, where each variable X_i has r_i possible values $x_i^1, \ldots, x_i^{r_i}$, $i = 1, \ldots, n$. We denote by θ_{ijk} the probability $P(X_i = x_i^k | \mathbf{Pa_i} = \mathbf{pa_i}^j)$, where $\mathbf{pa_i}^j$ is the j-th possible configuration of X_i's parents. Similarly, we use $\theta_{ij} = \{\theta_{ijk} | 1 \leq k \leq r_i\}$ to denote the set of parameters describing the local distribution $P(X_i | \mathbf{pa_i}^j)$. Also, we assume *parameter independence*, which says that θ_{ij} is independent of $\theta_{ij'}$ for all $j \neq j'$. For convenience, we also use θ to denote any distribution θ_{ij} in this section.

In classical statistical approach, the probability parameters are viewed as *physical property*, though unknown, of the world. They are assumed *objective* constants that can be estimated purely from a data set of training examples using *maximum-likelihood (ML)* estimates. The log-likelihood $logP(D|\Theta)$ can be decomposed according to the graph structure G using the chain-rule representation of joint probability in the equation 2 and expressed as below:

$$logP(D|\Theta) = \sum_{i,j,k} N_{ijk} log\theta_{ijk}, \tag{3}$$

where N_{ijk} are *sufficient statistics* representing the number of data instances matching the instantiations $X_i = x_i^k$ and $\mathbf{Pa_i} = \mathbf{pa_i}^j$. It is easy to show that this expression is maximized by the frequencies (maximum-likelihood estimates) $\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}}$, where N_{ij} is the number of samples matching the assignment $\mathbf{Pa_i} = \mathbf{pa_i}^j$, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Bayesian approach takes a different view at the probability parameters. In Bayesian statistics, the probabilities represent degree of *subjective* belief. The parameters are unknown variables governed by probability distributions. We assume some prior belief (e.g., based on background knowledge or historical information) in θ that is represented by the *prior distribution* $P(\theta)$. When a new data set D becomes available, this belief is updated according to Bayes' rule $P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$. Thus, the Bayesian approach takes advantage of prior knowledge about the parameters, which is especially useful when data are scarce. Imagine all possible values of θ from which this data set could have been generated. The *maximum a posteriori (MAP)* estimate of θ is the expectation of θ with respect to our posterior beliefs about its value:

$$E_{p(\theta|D)}(\theta) = \int \theta \, p(\theta|D) \, d\theta \, .$$

A common approach to modeling the prior belief over *multinomial* variables X with the parameters θ uses *Dirichlet distribution*, a *conjugate* distribution to multinomial, which has a nice property that the posterior $P(\theta|D)$ belongs to the same *conjugate family* as the prior $P(\theta)$ (Geiger & Heckerman 1995). The *Dirichlet distribution* is defined as follows:

$$Dir(\theta|\alpha_{ij1}, \ldots, \alpha_{ijr_i}) \equiv \frac{\Gamma(\alpha_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} \, ,$$

where $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$ and $\Gamma(\cdot)$ is the *Gamma-function* which satisfies $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(1) = 1$. The exponent parameters α_{ijk} are often called *hyperparameters*, in order to distinguish them from the θ_{ijk} parameters of the corresponding multinomial distribution. A common interpretation for α_{ijk} parameter is the number of times that an expert has previously observed the instantiation of $X_i = x_i^k$ and $\mathbf{Pa_i} = \mathbf{pa_i}^j$. For that, the α-parameters are also called *equivalent sample size* (i.e. the size of a data set that is an equivalent of the expert's knowledge). Thus, larger parameters reflect higher confidence in our prior. Given a set of observations D on multinomial variables X with the parameters $\theta = \{\theta_{ijk} | 1 \leq k \leq r_i\}$, it is easy to see that the posterior $P(\theta|D)$ is also Dirichlet:

$$\begin{aligned} P(\theta|D) &\propto P(D|\theta)P(\theta) \\ &\propto \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}} \cdot \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} \\ &\propto \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}+\alpha_{ijk}-1} \, . \end{aligned}$$

Therefore (taking into account normalization constant),

$$P(\theta|D) = Dir(\theta|\alpha_{ij1} + N_{ij1}, ..., \alpha_{ij1} + N_{ijri}) \, .$$

Given a network structure G, a complete data set D, a set of Dirichlet prior parameters α_{ijk}, and the assumption of parameter independence, it can be shown that the expected value of the parameters of the network with respect to the posterior distribution $p(\theta|D, G, \alpha_{ij})$ can be estimated by Equation 4:

$$\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}} \, , \tag{4}$$

where $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

As is apparent from Equation 4, the Dirichlet exponents α_{ijk} completely specify a user's current knowledge about the domain for purposes of learning probability parameters of the Bayesian network. Unfortunately, the specification of α_{ijk} for all possible $(x_i^k, \mathbf{pa_i^j})$ configurations corresponding to all values of i, j, and k is formidable. Thus, most learning algorithms simply adopt an uninformative assignment. For example, Cooper and Herskovits (Cooper & Herskovits 1992) suggest a uniform distribution with $\alpha_{ijk} = 1$ for all values of i, j, and k; Buntine (Buntine 1991) suggests an uninformative assignment $\alpha_{ijk} = \alpha/(r_i \cdot q_i)$, where r_i is the number of X_i's possible values and q_i is the total number of

$\mathbf{Pa_i}$'s configuration. With additional assumption of *likelihood equivalence* [1] and introducing *complete network* structures [2], Heckerman *et al.*(Heckerman, Geiger, & Chickering 1995) derived an exponent constraint on the α_{ijk} parameters. As a consequence, informative prior for the α_{ijk} parameters can be constructed by building a complete network S_c and assessing an equivalent sample size α for S_c.

$$\alpha_{ijk} = \alpha \cdot p(X_i = x_i^k, \mathbf{Pa_i} = \mathbf{pa_i^j}) \ ,$$

where $p(X_i = x_i^k, \mathbf{Pa_i} = \mathbf{pa_i^j})$ is the joint probability in the complete network S_c. Whereas assessing an equivalent sample size α is quite an easy task, building a complete network is demanding, to say the least. Most of the current learning algorithms simply ignore the variety of background knowledge of the domain experts by using uninformative prior for the α_{ijk} parameters. How to efficiently combine domain knowledge with data remains an unsolved problem.

Sensitivity of Bayesian networks

As we discussed above, learning probability parameters from data uses either *maximum-likelihood (ML)* estimate or *maximum a posteriori (MAP)* estimate for probability distributions to quantify a Bayesian network. However, a typically huge number of probability parameters in a Bayesian network may require very large datasets in order to learn accurate estimates, especially for probability distributions involving rare events. However, in many real-world applications data are relatively scarce and result into extreme (close to 0 or 1) values for the probability parameters, especially in *ML* estimates. The *MAP* estimate can avoid this problem by choosing appropriate hyperparameters that represent domain background knowledge. However, the uniform distributions used for the hyperparameters in the current learning algorithms ignore the variety of prior probability distributions. Therefore, the *MAP* estimates also can deviate from the true probability values. When there is no data available for learning, the domain experts' estimates are the only possible resource for Bayesian network quantification. Unfortunately, the subjective estimates are notoriously inconsistent and biased (Morgan & Henrion 1990; Cooke 1991). Even for a consistent belief, the numerical estimation can vary within a not small range. For example, a person believes a certain event happens *very likely* may assign the probability value as 0.8, or 0.9. The inherent inaccuracy of the probability estimation raises some interesting questions to Bayesian networks: when the probability parameters vary in a reasonable range, how the performance of the Bayesian network changes? Which parameters are the most important with regard to their effects on the sensitivity of the network?

[1] Likelihood equivalence says that, for any database D, the probability of D is the same given hypotheses corresponding to any two equivalent network structures. Two network structures are said to be equivalent when they encode the same independence relationships between nodes, only that the directions of edges can be different.

[2] A complete network is a network that has no missing edges. It encodes no assertions of conditional independence. In a domain with n variables, there are $n!$ complete network structures.

Sensitivity Analysis

Briefly, sensitivity analysis studies how sensitive a model's performance is to minor changes in the model. In a Bayesian network, the conclusion is drawn based on posterior probabilities of user queries, sensitivity analysis often investigates the effect of the changes in probability parameters on the posterior probabilities corresponding to the queries posed to the network. However, depending on the query (e.g., decision-making versus belief updating), other criteria may be more indicative measures of its performance. For example, in multi-failure diagnosis problem, a change in the order (rank) of most likely failures (Kipersztok & Wang 2001) is more appropriate than the value change of their posterior probabilities; in a network which involves recommending actions and decision making, a decision change is more indicative measure than the probability change (van der Gaag & Coupé 2000). In this paper, however, we will focus on the change in posterior probability of the target query as the sensitivity measure for a Bayesian network.

A *sensitivity function* can be used to express the change in posterior probability of the target query due to the variation in probability parameters of a Bayesian network. Laskey first proposed to use partial derivative of the sensitivity function and used an iterative approximation methods to determine the necessary derivatives (Laskey 1995). Castillo et al later proved that any posterior probability of a query is a fraction of two linear functions of a parameter (Castillo, Gutiérrez, & Hadi 1997). Since then, more efficient algorithms are proposed for sensitivity analysis based on message passing and joint tree propagation (Coupé *et al.* 2000; Darwiche 2000; Kjærulff & van der Gaag 2000). These analytical algorithms compute the posterior probability of the query as a function of the parameters according to Theorem 1 using various inference technique.

Theorem 1 *Let B be a Bayesian network, x be a probability parameter, y be a query, and e be evidence entered into B. The posterior probability $p(y|e)(x)$ is a fraction of two linear functions of x.*

$$p(y|e)(x) = \frac{\alpha_1 x + \beta_1}{\gamma_1 x + \delta} \ .$$

For simplicity, the function can be normalized as following:

$$p(y|e)(x) = \frac{\alpha x + \beta}{\gamma x + 1} \ .$$

Then the partial derivative of $p(y|e)(x)$ on x can be ex-

pressed as

$$\frac{\partial p(y|e)}{\partial x} = \frac{\alpha - \beta\gamma}{(\gamma x + 1)^2} \ . \quad (5)$$

Take the value of x in the Equation 5, we can get the *sensitivity value* of query y at x given e.

To determine the value of α, β, and γ, there are only three message propagations necessary for each given evidence. For example, if we use message passing scheme in junction tree inference with x's value set as $0, 0.5$, and 1, the values of the coefficients are completely determined as the following:

$$\begin{cases} \beta = p^0 \\ \gamma = \frac{\beta - p^{0.5}}{p^{0.5} - p^1} - 1 \\ \alpha = p^1(\gamma + 1) - \beta \end{cases} .$$

where $p^0, p^{0.5}$, and p^1 denote the corresponding posterior probabilities of $p(y|e)$ respectively. Note that when x's value was manipulated, the items in the same conditional probability table (CPT) column as x should be changed accordingly, since the summation of a single column parameters is equal to 1. *Proportional scaling* is often used to change the related parameters in such a way that they keep the original proportion. We use *meta parameter* to denote the parameter being manipulated to distinguish it from other dependent parameters in the same probability distribution.

As we see, the partial derivative $\frac{\partial p(y|e)}{\partial x}$ reflects the changing behavior of the posterior probability of y due to the changes in x under the evidence scenario e. It defines how sensitive the single query variable y to the probability parameter variable x given evidence e. We call this simplest sensitivity in Bayesian network *Parameter Sensitivity*.

Definition 2 (Parameter Sensitivity) *Let B be a Bayesian network, x be a probability parameter, y be a query, and e be evidence entered into B. Parameter sensitivity of x on y given e is a partial derivative:*

$$S(x|y,e) = \frac{\partial p(y|e)}{\partial x} \ .$$

Obviously, one way to measure the sensitivity of the whole Bayesian network is to take the average over all of the parameter sensitivities under the possible combinations of the queries, the evidence and the parameters. Note that different evidence can occur with different marginal likelihood, so the average over the weighted parameter sensitivities $p(e) \cdot \frac{\partial p(y|e)}{\partial x}$ may be more indicative as a measure of network sensitivity.

Based on parameter sensitivity, we can identify which parameters in a Bayesian network are the most important. Intuitively, when the different parameters undergo the same amount of variation, those with higher parameter sensitivity causes bigger changes in the query, and thus, affect the network's performance stronger. With consideration of multiple queries and evidence scenarios, we can define the importance of a parameter in a Bayesian network as below:

Definition 3 (Parameter Importance) *Let B be a Bayesian network, x be a probability parameter, y be a query, and e be evidence entered into B. Parameter importance of x in B refers to the average of x's sensitivity over possible y and e:*

$$I(x) = \frac{1}{mn} \sum_{y,e} S(x|y,e) = \frac{1}{mn} \sum_{y,e} \frac{\partial p(y|e)}{\partial x} \ ,$$

where m is the number of queries and n is the number of evidence scenarios.

Selective Parameter Update in Bayesian Network Learning

Definition of parameter importance provides a measure to the importance of the probability parameters in a Bayesian network with respect to the sensitivity of its queries under the possible evidence scenarios. Efficient algorithms for sensitivity analysis in Bayesian networks made it possible to recognize the important probability parameters using Definition 3. As a result, we can apply sensitivity analysis to identify those important parameters and focus our effort to refine their corresponding prior hyperparameters in Bayesian learning. Similar efforts have been made to improve the efficiency of quantifying a medical application of Bayesian networks using sensitivity analysis technique (Coupé *et al.* 1999; Coupé, van der Gaag, & Habbema 2000).

As summarized in Algorithm 1, the parameters in a Bayesian network can be selectively refined as follows. It begins with a Bayesian network learned from data with uniform distributions for the prior hyperparameters. Then importance of each meta parameter in the network is computed using sensitivity analysis algorithm. Given a threshold of the importance value, those parameters with a higher value than the threshold are identified and put into the *important parameter set*. For the important parameters not refined before, informative prior parameters are collected from experts or learned from new data. With the updated informative hyperparameters for the important probability parameters, *MAP* estimates are recomputed for these probabilities and the network is repopulated with the newly learned probability parameters. Iteratively perform sensitivity analysis and reparameterize the network with informative priors until no more parameters seem important and unrefined, or until the cost of further elicitation outweigh the benefits of higher accuracy. Practically, the stopping rules should include: a) satisfactory behavior of the network is achieved, and b) higher accuracy can no longer be attained due to lack of knowledge. In this

iteratively repeated procedure, the domain experts can focus their attention on the probabilities to which the network's behavior shows high sensitivity. Those uninfluential parameters can be left with crude estimates.

Algorithm 1 Selective Parameter Refinement

Input: a Bayesian network $B = (G, \Theta)$, threshold δ
Output: a Bayesian network $B = (G, \Theta')$
$cose\_effective = true$;
while $cost\_effective$ **do**
(1) Calculate importance $I(\theta)$ for all meta parameters θ in B;
(2) Identify a set of the most sensitive parameters $paraIndex = \{\theta : I(\theta) > \delta\}$;
if all $paraIndex$ have been refined before, or cost of knowledge extraction is unaffordable **then**
$cost\_effective = false$;
else
(3) Extract priors α for all $paraIndex$ from experts or new data;
(4) Recompute parameters for all $paraIndex$ with α and data counts;
(5) Update B with the new parameters Θ', s.t. $B = (G, \Theta')$;
end if
end while
Output B

Implementation

We implemented the algorithm to utilize sensitivity analysis in learning probability parameters of Bayesian networks. BNT (Murphy 2001), a Matlab Bayesian Network Toolbox developed at UC Bekerley, was used for the implementation. Equation 5 was used to calculate parameter sensitivity for parameter importance.

Experiment and Results

We used the Sprinkler network adapted from Russell and Norvig (Russell & Norvig 1995) in our experiment. The Sprinkler network is a very simple network which has four nodes: *Cloudy, Sprinkle, Rain* and *Wetgrass*. There are four edges totally in the network: *Cloudy→Sprinkle, Cloudy→Rain, Sprinkle→Wetgrass,* and *Rain→Wetgrass*. The network describes a simple mechanism about the probabilistic dependence relationships among the wet grass and its possible causes: whether it rains or the sprinkler is turned on. And the later two events depend on whether it is cloudy or not.

We chose the Sprinkler network for our tests because this simple network allows us to do exact inference for complete set of meta parameters in a very short time. For large networks to which exact inference is too expensive to apply, approximate inference can be used for sensitivity analysis. In our experiment, a query variable is randomly generated. Since our network is very small, we only assign one query node and two evidence nodes. In addition, the query node and evidence nodes are exclusive.

Our training data were generated by probabilistic logic sampling (Henrion 1988) from the true model. The data set contains 100 sample cases. 5 evidence scenarios were used for sensitivity analysis. Each of the evidence scenario consists of 2 observations. The observed nodes and their states are generated randomly. We assume the first state of the query node is of interest, and calculate its posterior probability for sensitivity analysis.

We tested our algorithm for selective parameter refinement with Sprinkler network. The goal is to illustrate the convergence of probability distributions with selective refinement to the true probability distributions. We used Hellinger distance between the two distributions to measure their difference. Hellinger distance between the two probability distributions p and q is defined as

$$H = \sum (\sqrt{p} - \sqrt{q})^2 .$$

To simulate an expert's prior knowledge, we generated a second data set using probabilistic logic sampling. This data set is used to estimate the informative hyperparameters α in step (3) of the Algorithm 1. When setting the parameter importance threshold $\delta = 0.2$, we found three parameters are important and recompute their probability values using informative α parameters. As expected, the Hellinger distance decreased with the selective refinement and thus, the probability distributions converged to the true distributions. Therefore, the algorithm for selective parameter refinement is demonstrated to be effective.

Figure 2 shows the parameter sensitivity values in Sprinkler network (totally 8 independent probability parameters). The figure also displays the Hellinger distances of the probability distributions before and after refining the most important 3 probabilities to the true distributions, which is 0.0307 and 0.0277 respectively. The decreased distance indicates the refined probabilities are closer to the true probabilities.

Discussion and Future Work

The algorithm for refining the probabilities can also be used for active information gathering. The active learning enables selecting new data for those important probabilities for more accurate estimate. When collecting data is very expensive, or when data volume is extremely large and requires too long

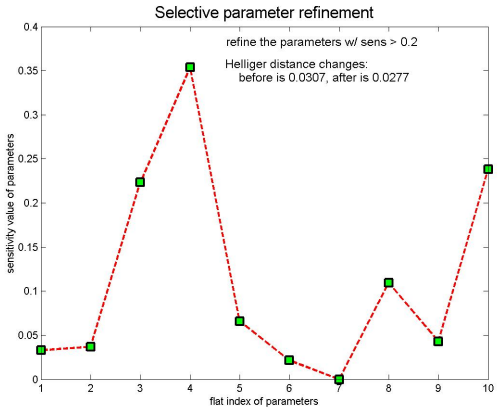

Figure 2: Parameter sensitivities of Sprinkler network and Hellinger distances of the probability distributions before and after refining 3 important probabilities to the true distributions

time to process, it may be unaffordable to get all of the probability parameters at the same accuracy level. Under these situation, our algorithm adaptable to active learning is very useful.

One factor missed in the current criteria for sensitivity is *sample risk*, which describes the absolute change in any probability parameter x measured by the variance of the parameter distribution. When we computed sensitivity value by the partial derivative of the sensitivity function, $\frac{\partial p(y|e)}{\partial x}$, we used x's value encoded in the network without considering x's distribution and its probability of taking this value. As discussed in section of learning parameters, the probability value learned from data is a *MAP estimate*. The parameter x is unknown variable governed by a probability distribution. Taking into account the variance of x's distribution can model its noise level. This way, the sensitivity value gives a more indicative measure of the change of the network's performance as result of x's uncertainty.

One part of our future work is to incorporate sample risk into sensitivity measure and improve sensitivity analysis technique for Bayesian networks. In addition, we plan to do more experiments on data for large-scale networks and enhance our learning algorithms to be more efficient and applicable to real-world problems. We will also extend our algorithm for active learning and selective data gathering.

Acknowledgement

The first author is grateful to Dr. Greg Cooper for his inspiration of the idea and early discussion of its possibility to implement. Special thanks go to Dr. Marek Druzdzel for his support to fulfil the work at IBM T.J. Watson Research Center.

References

Buntine, W. 1991. Theory refinement on Bayesian networks. In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–91)*, 52–60. San Mateo, California: Morgan Kaufmann Publishers.

Castillo, E. F.; Gutiérrez, J. M.; and Hadi, A. S. 1997. Sensitivity analysis in discrete Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 27(4):412–423.

Cooke, R. 1991. *Experts in Uncertainty: Opinion and Subjective Probability in Science*. Oxford University Press.

Cooper, G. F., and Herskovits, E. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9(4):309–347.

Coupé, V. M.; Peek, N.; Ottenkamp, J.; and Habbema, J. D. F. 1999. Using sensitivity analysis for efficient quantification of a belief network. *Artificial Intelligence in Medicine* 17:223–247.

Coupé, V. M.; Jensen, F. V.; Kjaerulff, U.; and van der Gaag, L. C. 2000. A computational architecture for n-way sensitivity analysis of bayesian networks. Technical report, Aalborg University.

Coupé, V. M.; van der Gaag, L. C.; and Habbema, J. D. F. 2000. Sensitivity analysis: an aid for belief-network quantification. *The Knowledge Engineering Review* 15(3):215–232.

Darwiche, A. 2000. A differential approach to inference in Bayesian networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, 123–132. San Francisco, CA: Morgan Kaufmann Publishers.

Geiger, D., and Heckerman, D. 1995. A characterization of the Dirichlet distribution with application to learning Bayesian networks. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, 196–207. San Francisco, CA: Morgan Kaufmann Publishers.

Heckerman, D.; Geiger, D.; and Chickering, D. M. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20:197–243.

Heckerman, D. 1998. A tutorial on learning with Bayesian networks. In Jordan, M. I., ed., *Learning in Graphical Models*. Cambridge, Massachusetts: The MIT Press. Available on web at ftp://ftp.research.microsoft.com/pub/tr/TR-95-06.ps.

Henrion, M. 1988. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In Kanal, L.; Levitt, T.; and Lemmer, J., eds., *Uncertainty in Artificial Intellgience 2*. New York, N. Y.: Elsevier Science Publishing Company, Inc. 149–163.

Kipersztok, O., and Wang, H. 2001. Another look at sensitivity of Bayesian networks to imprecise probabilities. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics (AISTAT-2001)*, 226–232. San Francisco, CA: Morgan Kaufmann Publishers.

Kjærulff, U., and van der Gaag, L. C. 2000. Making sensitivity analysis computationally efficient. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, 317–325. San Francisco, CA: Morgan Kaufmann Publishers.

Laskey, K. B. 1995. Sensitivity analysis for probability assessments in Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics* 25(6):901–909.

Morgan, M. G., and Henrion, M. 1990. *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge: Cambridge University Press.

Murphy, K. P. 2001. The Bayesian Net Toolbox for Matlab. *Computing Science and Statistics* 31. Available at http://www.cs.berkeley.edu/∼murphy/Bayes/usage.html.

Oniśko, A.; Druzdzel, M. J.; and Wasyluk, H. 2000. Extension of the Hepar II model to multiple-disorder diagnosis. In M.K̃lopotek, M.M̃ichalewicz, S. W., ed., *Intelligent Information Systems*, Advances in Soft Computing *Series*, 303–313. Heidelberg: Physica-Verlag (A Springer-Verlag Company).

Pradhan, M.; Henrion, M.; Provan, G.; del Favero, B.; and Huang, K. 1996. The sensitivity of belief networks to imprecise probabilities: An experimental investigation. *Artificial Intelligence* 85(1–2):363–397.

Rish, I. 2000. Advances in Bayesian learning. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, 95–102. Las Vegas, Nevada: CSREA Press.

Russell, S. J., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall.

van der Gaag, L. C., and Coupé, V. M. 2000. Sensitivity analysis for threshold decision making with Bayesian belief networks. In Lamma, E., and Mello, P., eds., *AI*IA 99: Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence*. Berlin: Springer-Verlag. 37 – 48.