Learning Bayesian Networks from Incomplete Data

Moninder Singh
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389
msingh@gradient.cis.upenn.edu

Abstract

Much of the current research in learning Bayesian Networks fails to effectively deal with missing data. Most of the methods assume that the data is complete, or make the data complete using fairly ad-hoc methods; other methods do deal with missing data but learn only the conditional probabilities, assuming that the structure is known. We present a principled approach to learn *both* the Bayesian network structure as well as the conditional probabilities from incomplete data. The proposed algorithm is an iterative method that uses a combination of Expectation-Maximization (EM) and Imputation techniques. Results are presented on synthetic data sets which show that the performance of the new algorithm is much better than ad-hoc methods for handling missing data.

Introduction

Many real-life domains are replete with missing values e.g. unobserved symptoms in a medical domain, nonresponse in a survey or missing readings from nonfunctioning sensors. For learning techniques to develop accurate models for such domains, it is essential that the missing values be properly accounted for, since failure to do so may result in severe biases and, thus, very poor models.

Research on handling data with missing values has been going on within the Statistics community for over three decades. Several techniques have been developed to handle such data, the principal ones being the Expectation-Maximization or EM algorithm (Dempster, Laird, & Rubin 1977), Gibbs sampling (Geman & Geman 1984) and Multiple Imputation (Rubin 1987).

However, within the Machine Learning community, relatively little attention has been paid to the development of effective techniques for handling incomplete data. This paper deals with the task of learning Bayesian networks from data with missing values. Although several methods have been developed for inducing Bayesian networks from data, e.g. (Cooper & Herskovits 1992; Singh & Valtorta 1995; Heckerman, Geiger, & Chickering 1995). most of these techniques assume that the data from which the network is to be learned is complete. In situations where this is not the case, as in most real-world problems, the data is made complete by "filling-in" values using a variety of, often ad-hoc, methods. Some work has been done in adapting methods such as EM and Gibbs sampling to the task of learning Bayesian network parameters (conditional probability tables) from incomplete data under the assumption that the structure is known; however, the general, and more practical, task of learning both network structure and parameters from incomplete data has not been fully explored.

In this paper, we present a principled approach to learn *both* the Bayesian network structure as well as the conditional probabilities from data with missing values. The proposed algorithm is an iterative method that successively refines the learned Bayesian network through a combination of Expectation-Maximization (EM) and Imputation techniques. Our experimental results show that the quality of the learned distribution (represented by the induced network) is much closer to the "true" distribution than the distribution learned by commonly used ad-hoc methods of handling missing data. This shows that it is important to properly account for missing values while inducing Bayesian networks from incomplete data, failing which very poor models may be learned.

The rest of the paper is organized as follows. We first discuss the important mechanisms that lead to missing data and some techniques that are commonly used to handle missing values, including some relevant work in the context of learning Bayesian network parameters, given the structure, from incomplete data. We then describe an algorithm for learning both the structure and parameters of Bayesian networks from incomplete data. We then present and discuss some experimental results, and close the paper by summarizing our contributions, and commenting on directions for future research.

Background

Knowledge about the mechanism that led to the missing data is often very important in choosing the appropriate method for analyzing the data. We first discuss these mechanisms, following the statistical framework adopted by Little and Rubin (1987), and then describe

some commonly used methods to handle missing data,

Mechanisms Leading to Missing Data

Based on different statistical assumptions, the mechanisms leading to missing data can be broadly categorized into three classes: Missing Completely at Random (MCAR), Missing at Random (MAR) and Not Missing at Random (NMAR). Let $D = \langle D^{obs}, D^{mis} \rangle = (d_{li})_{m \times n}$ where D^{obs} and D^{mis} represent the observed and missing component of the data D, m is the number of cases, d_{li} denotes the value of the i^{th} attribute in the l^{th} case, and n is the number of attributes. Let θ be the parameters governing the generation of the data, and let $P(D|\theta) \equiv P(D^{obs}, D^{mis}|\theta)$ denote the density of the joint distribution of D^{obs} and D^{mis}.

The missing data mechanism can be represented in the form of a missing data indicator matrix, $R = (r_{li})_{m \times n}$ such that r_{li} is 1 if d_{li} is observed, and 0 otherwise. If ψ is the set of parameters governing the distribution for the missing data mechanism, we can write (treating R as a random variable) the joint distribution of R and D as

$$P(D^{obs}, D^{mis}, R \mid \theta, \psi) = P(D^{obs}, D^{mis} \mid \theta) P(R|D^{obs}, D^{mis}, \psi). \quad (1)$$

If the probability that the data is missing is independent of both the observed, D^{obs}, as well as the missing, D^{mis}, data, i.e. $P(R|D^{obs}, D^{mis}, \psi) = P(R| \psi)$, then the missing data is said to be MCAR. However, if the probability that the data is missing for a particular variable may depend on the values of the observed variables in that case, but is independent of the values of the missing component itself (given the values of the observed component), i.e. $P(R|D^{obs}, D^{mis}, \psi) = P(R|D^{obs}, \psi)$, the missing data is said to be MAR. Finally, if the probability that the data is missing depends on the value of the missing component itself, and possibly on the value of the observed component as well, then the missing data is said to be NMAR.

The type of the missing data mechanism plays a critical role in determining the types of problems that can be addressed by various learning algorithms. The likelihood of θ based on the observed component of the data D^{obs}, and taking the missing data mechanism into account, is given by

$$L(\theta, \psi | D^{obs}, R) \propto P(D^{obs}, R|\theta, \psi).$$

From equation 1, we get

$$P(D^{obs}, R|\theta, \psi) = \int P(D|\theta) P(R|D^{obs}, D^{mis}, \psi) \mathrm{dD}^{mis}.$$

Now, if the data is missing at random (MAR), then

$$P(D^{obs}, R|\theta, \psi) = P(D^{obs}|\theta) P(R|D^{obs}, \psi).$$

Thus, for maximum likelihood methods, maximizing $L(\theta|D^{obs})$ as a function of θ is equivalent to maximizing $L(\theta, \psi|D^{obs}, R)$. Thus, the parameters of the missing data mechanism can be ignored for the purposes of estimating θ (Little & Rubin 1987). It is precisely these kinds of problems (where the missing data is either MAR or MCAR) that we are interested in. The case where the missing data is NMAR cannot be handled without first formulating a model for the missing data mechanism, and is beyond the scope of this paper.[1]

We now discuss some of the methods commonly used to handle missing data, especially those that are used to learn Bayesian network probabilities, given the structure, from incomplete data.

Methods for Handling Missing Data

The simplest way for handling missing data is to use only fully-observed cases. However, this approach may lead to serious biases in the presence of large amounts of missing data, and also, may not be practical when the amount of available data is fairly small. A more common approach within the Machine learning community is to assign to each missing value a 'new value' corresponding to an 'unknown category' in the value set. Although this method is straightforward, it suffers from the drawback that it adds an additional parameter to be estimated. Moreover, the extra value does not reflect the fact that the missing value actually came from the original multinomial value set; thus, for a dataset where the value of a particular attribute is often missing, an algorithm may form a class based on that attribute value being unknown, which may not be desirable in a classifier (Ghahramani & Jordan 1994).

Other commonly used techniques to handle incomplete data involve replacing each missing value by a single value (Single Imputation) computed by some method. Some of the frequently used procedures involve replacing a missing value by the mean of the observed values for that attribute (mean imputation) or, more generally, by sampling from the unconditional distribution of that attribute estimated from the observed values (hot-deck imputation) (Little & Rubin 1987). These methods, though simple, can lead to serious biases since a single value can in no way convey the uncertainty about the true value; rather, it depicts (incorrectly) that the imputed value is in fact the 'actual' missing value. A much better technique is to use Multiple imputation methods (Rubin 1987) — that impute more than one value for the missing data points. Each missing value is replaced by an ordered vector of $M \geq 2$ imputed values thus giving M completed data sets, each of which can be analyzed using standard complete-data methods. As discussed by Rubin (1987), multiple imputation has several advantages over single imputation. First, the uncertainty about the missing data is easily represented. Second,

[1] The NMAR case can conceivably be handled by augmenting the original dataset D with R such the new dataset consists of $2n$ attributes, the first n attributes corresponding to D and the next n attributes corresponding to R. The methods described in this paper can then be used to learn Bayesian networks from the augmented dataset even if the missing data is NMAR.

if the mechanism is unknown, then the imputed values represent the uncertainty about the different reasons (models) for the missing values. Third, by using complete data methods to analyze each of the M complete data sets, the results can be combined to yield a single, more representative model.

Another method of handling missing data is to sum over all possible values for each missing data point while calculating the required parameters. Cooper and Herskovits (1992) discuss a theoretical approach for learning both structure and parameters for Bayesian networks when the missing data is MCAR using this method. Although, Cooper (1995) extends this approach further to yield a relatively efficient algorithm, the method may still be computationally expensive for many real world problems.

A number of methods have also been developed that define a model for the partially missing data and base inferences on the likelihood under that model. Two commonly used techniques are the EM algorithm (Dempster, Laird, & Rubin 1977) and Gibbs sampling (Geman & Geman 1984). Of these, the EM algorithm is especially appealing having the advantages of being conceptually easy to design, and more importantly, having the property of converging relatively quickly; however, it suffers from the disadvantage of often finding only local maxima. These methods are reviewed in detail in (Heckerman 1995). Heckerman (1995) also discusses an approximation method for computing the likelihood of a Bayesian network structure given the data; however, using this technique to compare various structures and determine the most likely network can be computationally inefficient.

Here, we briefly describe the application of the EM algorithm to the task of learning Bayesian network probabilities as proposed by Lauritzen (1995). The likelihood function is given by

$$L_{B_S}(\theta|D) \propto P_{B_S}(D|\theta) = \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}\theta_{ijk}^{N_{ijk}}$$

where B_S is the Bayesian network structure, n is the number of attributes, q_i is the number of possible instantiations of the parents, $pa(i)$, of attribute i, r_i is the number of values of attribute i, N_{ijk} is the number of cases in which attribute i is instantiated to k while $pa(i)$ is instantiated to j, and $\theta = (\theta_{ijk})_{n\times q_i \times r_i}$ are the conditional probabilities. The **E** step finds the conditional expectation of the complete-data loglikelihood, given the observed component of the data and the current values of the parameters. Thus, it requires determining the value of

$$\mathrm{E}(N_{ijk}|D^{obs},\ \theta) \quad = \quad \sum_{l=1}^{m}\mathrm{E}_\theta(\chi_{ijk}^l|d_l^{obs}) \qquad (2)$$

where d_l^{obs} is the observed component of the l^{th} case, and χ_{ijk}^l is defined as

$$E_\theta(\chi_{ijk}^l|d_l^{obs}) = \begin{cases} 1, & i,\ pa(i) \text{ obs.; } i = k \ \&\ pa(i) = j \\ 0, & i,\ pa(i) \text{ obs.; } i \neq k \text{ or } pa(i) \neq j \\ P_{B_S}(i = k, pa(i) = j|d_l^{obs},\ \theta), & \text{oth.} \end{cases}$$

To determine $P_{B_S}(i = k, pa(i) = j|d_l^{obs}, \theta)$, the Bayesian network $\langle B_S, \theta\rangle$ is instantiated with d_l^{obs}, and inference is carried out. The **M** step then consists of using these estimated values to compute θ_{ijk}. Thus,

$$\theta_{ijk} \quad = \quad \frac{\mathrm{E}_\theta(N_{ijk}|D^{obs},\ \theta)}{\mathrm{E}_\theta(N_{ij}|D^{obs},\ \theta)}; \qquad N_{ij} = \sum_k N_{ijk}.$$

Learning Bayesian Network Structure and Parameters from Incomplete Data

In this section, we propose an iterative algorithm for learning not only conditional probabilities from data with missing values but also to learn the network structure as well. This algorithm uses a combination of EM and Imputation techniques to iteratively refine the network structure and the parameters as follows: it uses the current estimate of the structure and the incomplete data to refine the conditional probabilities, then imputes new values for missing data points by sampling from the new estimate of the conditional probabilities, and then refines the network structure from the new estimate of the data using standard algorithms for learning Bayesian networks from complete data. Since the EM algorithm does not "fill-in" values but rather, estimates the conditional expectation of the complete-data loglikelihood, it cannot be directly used for this purpose. However, as explained in the previous section, it can be used to estimate the conditional probabilities from the missing data assuming that the structure is known. Thus, the current estimate of the structure can be used along with the original missing data to refine the probabilities, which in turn can be used as described above. In order to impute values for a missing data point, the current estimate of the Bayesian network is used to determine the conditional distribution of the attribute in question, given the available evidence in the corresponding case, and the imputed values are randomly drawn from this distribution.

Thus, the algorithm may be described as follows:

1. Create M complete-datasets, $\hat{D}_s^{(0)}$, $1 \leq s \leq M$, by sampling M values for each missing value from the prior distribution of each attribute.
2. For $s := 1$ to M do

   2a. From the complete-dataset $\hat{D}_s^{(t)}$, induce the Bayesian network structure, $B_s^{(t)}$, that has the maximum posterior probability given the data, i.e. maximizes $P(B_s|\hat{D}_s^{(t)})$.

   2b. Use the EM algorithm to learn the conditional probabilities $\theta_s^{(t)}$ using the original, incomplete data D and the network structure $B_s^{(t)}$.
3. Fuse the networks to create a single Bayesian network $< B^{(t)}, \theta^{(t)} >$ as follows. Construct the network structure $B^{(t)}$ by taking the arc-union of the individual, network-structures, i.e. $B^{(t)} = \bigcup_{s=1\ldots M} B_s^{(t)}$. If the orderings imposed on the attributes by the various network structures are not

consistent, then it is possible to construct $B^{(t)}$ by choosing one of the orderings (e.g. a total ordering consistent with the network structure with the maximum posterior probability), making all the other network structures consistent with this ordering by performing necessary arc-reversals, and then taking the graph union of all the resultant structures. Matzkevich and Abramson (1993) describe an efficient algorithm to do so. Then, create $\theta^{(t)}$ by taking a weighted-average of the individual distributions.

4. If the convergence criteria is achieved, stop. Else, go to step 5.
5. Create M new, complete-datasets $\hat{D}_s^{(t+1)}$ by sampling M values for each missing value $d_{li} \in d_i^{mis}$ by sampling from the distribution $P(d_{li}|d_l^{obs}, < B^{(t)}, \theta^{(t)} >)$. Go to step 2.

Experimental Evaluation

In order to evaluate the performance of our learning algorithm, we used the following method. We start with a given Bayesian network, the gold-standard network and generate a database of cases from it using repeated sampling. Then, we systematically remove data from it under various assumptions (MCAR or MAR), and then use our algorithm to learn a Bayesian network from the incomplete data. Finally, we measure the performance of the algorithm by comparing the true and the learned joint probability distributions.

We used the ALARM network (Beinlich *et al.* 1989) as the gold-standard for our experiments. We generated a database of 10000 complete cases from this network using the case-generation facility of HUGIN (Anderson *et al.* 1989). We then, systematically, removed data from the database in one of two ways:

1. MCAR: To remove data under the MCAR assumption, we did the following:
   1. Let δ be the required percentage of missing data.
   2. For each case, l in the dataset, and each attribute x_i, remove the observed value of x_i in case l with probability δ. This ensured that, at any stage, the probability that a value was missing did not depend on either the observed or the missing component of the dataset.
2. MAR: To generate missing data under the MAR assumption, we followed the method described below
   1. Let δ be the required percentage of missing data
   2. If the current percentage of missing data equals δ, then stop. Else, go to Step 3.
   3. Generate a MAR rule as follows:
      a. Randomly select an attribute x_i.
      b. Randomly select a set of attributes Y, as well as an instantiation y of Y.
      c. Randomly select a probability p
   4. For each case k, apply the MAR rule as follows: if in case k, the set of attributes Y is observed and is instantiated to y, then remove the observed value of attribute x_i with probability p. This ensures that the probability that the value of x_i is missing depends on the observed component of the data in that case i.e. MAR
   5. Go to Step 2.

We removed various amount of data under the two assumptions, and used our algorithm to learn both the Bayesian network structure as well as the conditional probability tables from the incomplete data. We used the log-likelihood of the data to assess the progress of the algorithm, stopping when the change in the log-likelihood, relative to the previous iteration, was less that 10^{-4}. Note that the log-likelihood function can be evaluated without any extra work by simply summing over all cases the log of the normalization constant obtained after propagating the evidence in that case.

For inducing Bayesian networks from complete-data (step 2a), we used the K2 algorithm (Cooper & Herskovits 1992), with a total ordering on the attributes, consistent with the gold-standard network, as input. Furthermore, in the experiments involving multiple imputation, we used the posterior probability that a node has a given set of parents (as computed by the K2 algorithm) as the weights for taking the mixture of distributions in Step 3 of the algorithm.

To measure the quality of the learned network, we computed the *cross-entropy* (Kullback & Leibler 1951) from the actual distribution (represented by the gold-standard) to the distribution represented by the learned network. The cross-entropy measure is defined as follows. Let P denote the joint distribution represented by the gold-standard network and Q the joint distribution represented by the learned network. Then, the cross-entropy $H(P,Q)$ is given by

$$H(P,Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)} \qquad (3)$$

Low values of cross-entropy correspond to a learned distribution that is close to the gold standard. We used the method suggested by Heckerman *et al.* (1995) to evaluate Equation 3 efficiently without summing over an exponential number of terms.

Results and Discussion

We carried out a series of experiments by varying three parameters: a) the mechanism leading to the missing values, b) the amount of missing data, and c) the number of imputations used. Due to space limitations, we present and discuss the results of only some of these experiments.

Figure 1 shows the results of evaluating the algorithm on data containing about 20% missing values, removed completely at random (MCAR). The algorithm used only single imputation to fill-in the missing values at every stage. The figures plot the improvement in the cross-entropy and the log-likelihood function against the progress of the algorithm. Similarly, Figure 2 depicts the results for data containing about 40%

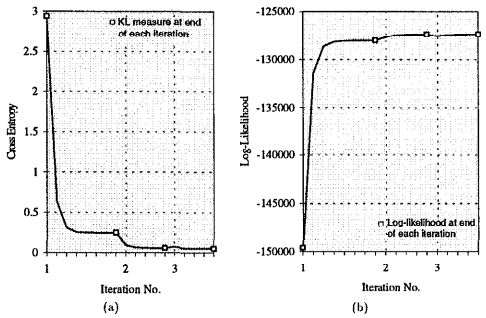

Figure 1: Learning with 20% MCAR data using 1-Imputation: (a) Cross-entropy (b) Log-likelihood

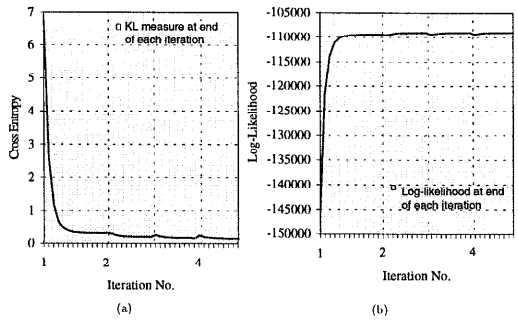

Figure 2: 40% MCAR data; algorithm used 2-Imputations (a) Cross-entropy (b) Log-likelihood

missing values removed completely-at-random using 2-imputations, Figure 3 shows the results for data containing about 20% values missing at random (MAR) using 1-imputation.

An important point to note in these figures is that the first plotted-value in each graph (shown on the Y-axis) represents the Bayesian network learned from the data which had been made complete by filling-in missing values using the prior distributions (unconditional imputation). Clearly, the learned Bayesian network in this case is far off from the true model which strengthens our claim that ad-hoc methods used to handle missing values often result in very poor models. Similar results would be expected from another often used approach of replacing each missing value by its (sample) mean, since that is a special case of the above method.

For the case where only 20% data is missing under the MCAR assumption, the algorithm induces a Bayesian network with a joint distribution that is very close to the true representation (as represented by the gold-standard network) with a cross-entropy of only 0.05. As a reference point, the cross-entropy value for the Bayesian network learned by the K2 algorithm from the *complete* dataset was 0.04.

As the amount of the missing data increases to 40%, the deviation of the learned distribution from the true distribution also increases (entropy is 0.14). However, this is to be expected since the quality of the ML estimate is directly related to the amount of missing information. Nevertheless, the learned Bayesian network is still much better than the one learned by replacing missing values by sampling from unconditional distributions.

Another important issue is the rate of convergence of the algorithm under various assumptions. Whereas the labels along the X-axis show the number of iterations of the main (outer) loop of the algorithm, the marks on this axis show the number of iterations of the EM algorithm while learning the conditional probabilities from the missing data given the current estimate of the network structure.

A number of interesting observations about the rate of convergence can be derived from the figures. Firstly, the maximum gain is generally achieved during the first iteration of the algorithm, especially when data is missing under the MCAR assumption (Figures 1–2). Thus, if computational effort is an issue, then fairly good Bayesian networks can be learned by stopping after the first iteration. When the missing data is MAR (Figure 3), then the improvement during subsequent iterations is substantial, though much smaller than the improvement in the first iteration. In this case, it is advantageous to run the algorithm a little longer. Also, the number of iterations needed for convergence increase with the amount of the missing data, both in terms of the number of times the network structure is revised (main loop) as well as the number of times the conditional probabilities have to be updated (inner, EM loop). Again this is to be expected since the rate of convergence should be proportional to the fraction of information in the observed data, D^{obs}.

Table 1: Number of missing/extra arcs in learned network structures

Cases	10000	1000	100
K2	2e,2m	2e,2m	13e,6m
MCAR 20%	4e,2m	5e,2m	18e,7m

We also measured the number of extra/missing arcs in the induced networks (Table 1) for the case where 20% data was missing completely-at-random. For datasets of sizes 10000 and 1000 cases, the induced network just had 2 and 3 more extra edges, respectively, than the network induced by K2 from the complete-dataset. As the sample size was reduced even further (to 100 cases), the quality of the induced network deteriorated slightly, recovering 5 more arcs and 1 less arc than that recovered by K2.

Finally, in our experiments, using multiple imputation did not have a very large impact. The distributions learned by using 2-imputation were only marginally better than those learned using single imputation. However, in the MAR case, multiple imputation resulted in faster convergence of the algorithm. This improvement, both in terms of the quality of the network learned as well as in the convergence

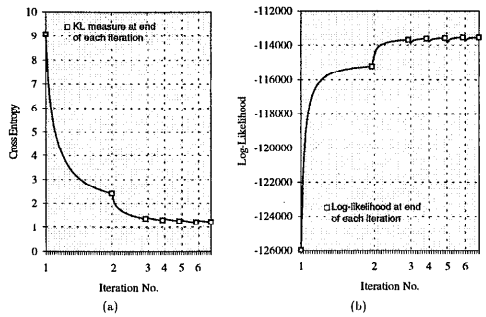

Figure 3: 20% MAR Data; algorithm used 1-Imputation (a) Cross-entropy (b) Log-likelihood

rate should become more pronounced as the amount of missing data increase and/or the size of the available dataset decreases. In either case, there will be much more uncertainty about the missing values than in the current situation. However, additional experiments are needed to justify this claim.

Conclusions and Future Work

A systematic, principled approach to learning Bayesian networks, both structure and conditional probabilities, from incomplete data has been presented. Previous methods for learning Bayesian networks from data assume that the data is complete; in cases where the data is incomplete, ad-hoc methods are often used to make it complete. Existing methods that do account for missing values assume that the network structure is known, and only learn the network parameters from the incomplete data.

Our experimental results show that the quality of Bayesian networks learned by the new algorithm from incomplete data is much better compared to commonly used, ad-hoc methods. Moreover, the learned distribution (as encoded by the induced Bayesian network) is fairly close to the true distribution, especially when amount of missing data is not very large.

However, there are at least two important issues that must be explored further. First, further experimentation should be done using other evaluation functions such as classification accuracy, or ideally, utility functions. This is especially important since, for problems in the real domain, the true model is unknown and hence the Kullback-Leibler distance cannot be measured. Second, further experiments are needed on real-world datasets. For such domains, it may not be possible to get a good ordering on the domain attributes. In such cases, it is possible to incorporate algorithms for learning Bayesian networks that do not require a node ordering e.g. (Singh & Valtorta 1995) with the proposed algorithm to learn Bayesian networks from incomplete data. These are both important issues which we hope to address in the future.

Acknowledgments. This research is funded by an IBM Cooperative Fellowship. The author is thankful to Dr. David Heckerman for several useful discussions, and to Dr. Gregory Provan and the anonymous reviewers for their comments and suggestions. The author is also grateful to Hugin Expert A/S for providing the HUGIN software to him for his Ph.D. research.

References

Anderson, S.; Olesen, K.; Jensen, F.; and Jensen, F. 1989. HUGIN - A Shell for building Bayesian Belief Universes for Expert Systems. In *Procs. IJCAI'89*, 1080–1085.

Beinlich, I.; Suermondt, H.; Chavez, R.; and Cooper, G. 1989. The ALARM monitoring system: A Case Study with Two probabilistic inference techniques for belief networks. In *Procs. Second European Conference on AI in Medicine*, 247–256.

Cooper, G. 1995. A Bayesian method for learning Belief networks that contain hidden variables. *Journal of Intelligent Systems* 4:71–88.

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

Dempster, A.; Laird, N.; and Rubin, D. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society (Series B)* 39:1–38.

Geman, S., and Geman, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on PAMI* 6:721–741.

Ghahramani, Z., and Jordan, M. 1994. Learning from incomplete data. A.I. Memo 1509, MIT AI Lab.

Heckerman, D. 1995 (revised Nov. 1996). A tutorial on learning with Bayesian networks. Technical report MSR–TR–95–06, Microsoft Research, Redmond, WA.

Heckerman, D.; Geiger, D.; and Chickering, M. 1995. Learning Bayesian networks–the combination of knowledge and statistical data. *Machine Learning* 20(3):197–243.

Kullback, S., and Leibler, R. 1951. Information and sufficiency. *Ann. Math. Statistics* 22.

Lauritzen, S. 1995. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis* 19:191–201.

Little, R. J. A., and Rubin, D. B. 1987. *Statistical Analysis with Missing Data.* John Wiley & Sons.

Matzkevich, I., and Abramson, B. 1993. Deriving a minimal i-map of a belief network relative to a target ordering of its nodes. In *Procs. UAI'93*, 159–165.

Rubin, D. 1987. *Multiple Imputation for Nonresponse in Surveys.* New York: Wiley.

Singh, M., and Valtorta, M. 1995. Construction of Bayesian network structures from data: a brief survey and an efficient algorithm. *International Journal of Approximate Reasoning* 12:111–131.