Error-Bounded Probabilistic Graphical Model Simplification

by

Scott A. Wahl

A proposal submitted in partial fulfillment
of the requirements of the comprehensive

for

Doctor of Philosophy

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

April 2011

Contents

ABSTRACT ii

1. PROBLEM STATEMENT 1

2. BACKGROUND WORK 3

Bayesian Networks 3
Diagnostic Bayesian Networks 5
Variable Representation 7
Continuous Variables 7
Noisy OR-gate 7
Dynamic Bayesian Networks 8
Inference and Simplification 10

3. RELATED WORK 12

Model Approximation 12
Hidden Variable Discovery 13
Diffusion of Context 14

4. EXPERIMENTAL DESIGN 17

Model Simplification 17
Logical Closure 17
Transitive Reduction 18
Model Correction 18
Data Sets 19
Schedule 19

5. PRELIMINARY RESULTS 24

Transitive Reduction Accuracy 24
Decision Tree Extraction 26

REFERENCES CITED 31

ABSTRACT

Due to the complexity of performing inference in probabilistic graphical models, both diagnostics and prognostics are computationally expensive tasks which are vital to a number of industries. To combat this issue, approximate methods have been proposed for estimating posterior distributions during the inference procedure. The majority of these approaches directly approximate the calculation of the distribution. For large networks this is still a daunting task, even if only due to the storage requirements necessary in working on a large distribution. Instead, a method for approximating and simplifying the representation of the network is proposed.

A framework for performing efficient inference through model approximation in Bayesian and dynamic Bayesian networks is proposed with applications for diagnosis and prognosis in real systems. Model approximation is performed by the removal of arcs within the probabilistic graphical models. Due to the changes in the independence relationships and possible diffusion of context, the model must be updated with the introduction of hidden variables, dynamic parameter adjustment, and reinsertion of arcs. These controlled modifications will ensure both a simplification of the network and a bound on the classification error and divergence of the resulting network with the original while performing inference.

PROBLEM STATEMENT

Beyond standard classification tasks, probabilistic graphical models are frequently used in performing diagnostics and prognostics for complex systems. Especially in the fields of aviation and many others, proper diagnosis and prognosis of equipment is vital for both safety and financial concerns. Effective diagnosis is also a considerable concern in medical fields. Unfortunately, probabilistic graphical models used to perform these tasks are frequently extremely large and complex. Given the difficulty of performing exact inference in such a model, it is often necessary to use approximate techniques to perform queries. Additionally, depending on resources, it may not be feasible to store an exceptionally large network with large conditional probability tables representing distributions.

Further problems are encountered specifically in relation to performing prognostics. Compared to diagnostics, the field of prognostics is quite young, in part due to the difficulty of accurately predicting future failures and the remaining useful life of parts, especially in the presence of uncertainty [1, 2]. Recent publications in this area are still establishing appropriate metrics and tools to compare the performance of differing algorithms [3, 4, 5]. The usual alternative to data-based approaches to prognostics are mathematical models which attempt to accurately describe the physics of the system. While accurate, these systems can be exceptionally complex [6].

The primary motivation for this proposal is based on work performed by Cooper [7]. By using a reduction from 3SAT, it was shown that performing inference in a Bayesian network is an NP-hard problem. Especially important in the realm of diagnostics and prognostics,

it was also shown that in a constrained network structure (specifically a bipartite network) inference is still an NP-hard problem. Furthermore, the inference task is NP-hard even with a bipartite network where the conditional probability distributions are represented by a noisy OR-gate. Therefore, performing diagnostics and prognostics in a large system requires the use of specialized or approximation algorithms.

BACKGROUND WORK

Bayesian Networks

Bayesian networks are an implementation of a probabilistic graphical model which takes the form of a directed, acyclic graph [8, 9, 10, 11, 12]. This graph is used to represent a joint probability distribution over a set of variables. To represent this distribution compactly, first consider a set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ and a joint probability distribution given by $\mathbf{P}(\mathbf{X}) = \mathbf{P}(X_1 \ \ldots, X_n)$.

Definition 1 *Given a joint probability distribution over a set of variables* $\{X_1, \ldots, X_n\}$*, the product rule provides a factoring of the joint probability distribution by the following*

$$\mathbf{P}(X_1, \ldots, X_n) = \mathbf{P}(X_1) \prod_{i=2}^{n} \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1}).$$

While this adequately describes the joint probability distribution, there is an issue in that each "factor" of the network is represented by a conditional probability table whose size is exponential in the number of variables. However, exploiting conditional independence properties allows the Bayesian network formulation to reduce the complexity of the representation.

Definition 2 *A variable* X_i *is conditionally independent of variable* X_j *given* X_k *if*

$$\mathbf{P}(X_i, X_j \mid X_k) = \mathbf{P}(X_i \mid X_k)\, \mathbf{P}(X_j \mid X_k).$$

Using these definitions, the joint probability can be represented more compactly by calculating the distribution by the set of conditional independence relations. In particular,

the Bayesian network is able to encapsulate this representation by creating a node in the graph for every variable in $\mathbf{X}$. For any two nodes, or variables, in the network given by X_i and X_j, X_j is referred to as a parent of X_i if there is an arc from X_j to X_i. Based on this, the function $Parent\left(X_i\right)$ is defined to be equal to the set of nodes with arcs leading into X_i. Associated with every variable X_i is a conditional probability distribution which can be denoted by $\mathbf{P}\left(X_i \mid Parents\left(X_i\right)\right)$. From this, the parameters of the network Θ define $\mathbf{P}\left(X_i \mid Parents\left(X_i\right)\right)$. Based upon this representation, a simplification of the joint probability distribution can be performed as

$$\mathbf{P}\left(X_1, \ldots, X_n\right) = \prod_{X_i \in \mathbf{X}} \mathbf{P}\left(X_i \mid Parents\left(X_i\right)\right).$$

As a small example, consider a joint distribution given by $\mathbf{P}\left(X_1, X_2, X_3, X_4, X_5, X_6\right)$ with conditional independence relations which allow it to be factored as

$$\mathbf{P}\left(\mathbf{X}\right) = \mathbf{P}\left(X_1\right)\mathbf{P}\left(X_2\right)\mathbf{P}\left(X_3\right)\mathbf{P}\left(X_4 \mid X_1\right)\mathbf{P}\left(X_5 \mid X_1, X_2\right)\mathbf{P}\left(X_6 \mid X_1, X_2, X_3\right).$$

Figure 1 shows the directed graph which represents this distribution. Assuming binary variables in the distribution above, the full joint probability in table form would require $2^6 = 64$ entries. However, the Bayesian network representation from the given factorization only requires $2^0 + 2^0 + 2^0 + 2^1 + 2^2 + 2^3 = 17$, a rather substantial reduction.

A variety of different approaches have been made for learning Bayesian networks from data or for using Bayesian techniques in learning models from data [13, 14, 15]. One of the classical approaches is based upon a structural expectation-maximization algorithm [16] which attempts to balance the likelihood of the resulting network with the complexity. This basic idea is prevalent in most structure learning methods with the use of scoring metrics,

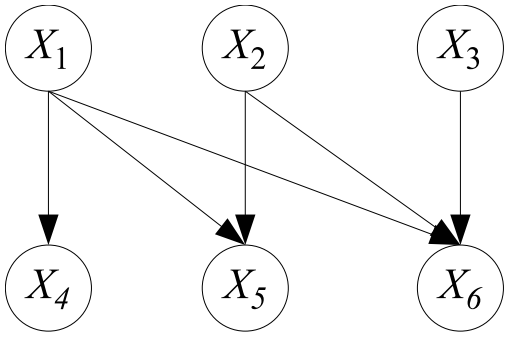
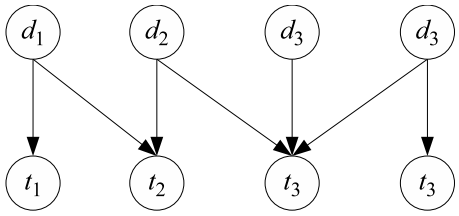


Figure 1. Example Bayesian Network.

of which there are many and most described in the structure learning papers. However, there are more recent results which use spectral decomposition approximate the model score [17]. Various methods of implementing the search have been attempted, such as genetic algorithms [18, 19], exact and approximate search over ordered variables [20, 21, 22], online search [23], and local structure search instead of global search [24, 25]. Additional work has been performed in analyzing the sample complexity of learning networks [26].

Diagnostic Bayesian Networks

One specialized form of Bayesian networks used as a classifier for performing fault diagnosis is the diagnostic network [27, 28, 29, 30, 31, 32]. The diagnostic network consists of two different types of nodes: class (i.e., diagnosis) and attribute (i.e., test) nodes. During the diagnosis procedure, every test performed is an indicator for a possible set of faults which are indicated as parents of the test. For a small example, consider a test of the state of a light bulb with the results of on or off. With no other information, this potentially indicates a variety of potential problems such as a broken filament or damaged wiring. Based on this, every test node in the network will have a set of diagnosis nodes as parents. Like the



(a) Bipartite Diagnostic Network

	t_1	t_2	t_3	t_4
d_1	1	1	0	0
d_2	0	1	1	0
d_3	0	0	1	0
d_4	0	0	1	1

(b) D-Matrix

Figure 2. Example diagnostic Bayesian network.

Bayesian network it is based upon, every node in the network has an associated conditional probability distribution. More specifically, the distribution of a diagnosis node is representative of some probability of failure. The distribution of a test node is representative of the outcomes of the test given the parent failures.

Based on the previous formulation, creating a network from diagnosis and test nodes results in a bipartite Bayesian network. Due to this structure, it is possible to represent the structure of a network with a specialized adjacency matrix referred to as a D-Matrix. Consider a diagnostic network with the set of diagnoses $\mathbf{D} = \{d_1, \ldots, d_n\}$ and the set of tests $\mathbf{T} = \{t_1, \ldots, t_m\}$. Using the simplest interpretation, every row of the D-Matrix corresponds to a diagnosis from $\mathbf{D}$ whereas each column corresponds to a test from $\mathbf{T}$. From this construction, the following definition is derived.

Definition 3 *A D-Matrix* $\mathbf{M}$ *is an* $n \times m$ *matrix where every entry* $m_{i,j}$ *contains a binary value* 0 *or* 1*. Each entry of* 1 *indicates that* d_i *is a parent of* t_j *while each entry of* 0 *indicates that* d_i *is not a parent of* t_j*.*

Figure 2 provides an example diagnostic Bayesian network with four classes and four attributes labeled d_1, d_2, d_3, d_4 and t_1, t_2, t_3, t_4 respectively. The figure also contains the corresponding D-Matrix for this network.

Variable Representation

Beyond the standard representation of discrete states and conditional probability tables, many other formulations of both variables and distributions have been used in Bayesian networks.

Continuous Variables

One classical approach to handling continuous variables within a Bayesian network relies on transforming the variables into discrete variables [33]. The methods used for this vary dramatically in complexity. A very simple method creates bins where the values are sorted into k bins where the number of values in each bin is equal, or as close to equal as possible. More complex techniques use top-down and bottom-up decision tree techniques based on information gain and merging based upon the χ^2 test. Another technique was created to work in concert with structure learning to minimize a description length which includes the discretization bins [34]. Working with continuous variables directly frequently involves using a Gaussian distribution, or a mixture of Gaussians, in order to represent the variable [35, 36, 37].

Noisy OR-gate

Briefly mentioned previously, the noisy OR-gate is a common representation for the conditional probability distribution within diagnostic networks [8, 38, 39, 40, 41]. One reason for its popularity is that it is simple and effective in those settings due to the disjunctive interaction. Consider multiple potential faults in a system, each of which are likely to be indicated by some test. Assume one of those faults is currently active and the test is indicating there is a problem. If another fault occurs during this time, the test should not be then less likely to indicate a problem is occurring. With a well-built system, this should be a valid assumption for most cases. Two assumptions can be used to approximate this interaction which forms the basis of the noisy OR-gate. The first, accountability, states that any test will be false if none of its parent faults are active. This is a desirable trait in a diagnostic setting since that would indicate a false alarm. The second is that anything which prevents a test from detecting a specific fault does not also prevent that test from detecting a separate fault. Again, this is desirable, but it is easy to see that events such as power failures violate this constraint.

Dynamic Bayesian Networks

Dynamic Bayesian networks (DBNs) are an extension of Bayesian networks used for temporal analysis [42, 43, 44, 45]. Like Bayesian networks, DBNs represent a joint distribution over a set of variables. The set of variables $\mathbf{X}$ defined above for Bayesian networks are now extended to template variables $\mathbf{X}[t]$ representing the set of variables $\mathbf{X}$ as captured through time. Due to differing dependency relationships, two separate types of arcs need to

be defined. Inter-temporal dependency arcs are those which define conditional dependences among variables at different time intervals which can be defined by the arcs between variables $\mathbf{X}[i]$ and $\mathbf{X}[j]$ where $i \neq j$. In general, it is assumed that arcs are unidirectional in moving forward through time indicating that for any arc from a variable in $\mathbf{X}[i]$ to a variable in $\mathbf{X}[i]$ implies that $i \leq j$. In addition to the inter-temporal arcs, there are intra-dependency arcs which define conditional dependencies among variables $\mathbf{X}[i]$ and $\mathbf{X}[j]$ where $i = j$. For DBNs, the function $Parents(X_i[t])$ incorporates both inter- and intra- temporal arcs.

Due to these temporal dependencies, the parameters Θ must be expanded such that $\Theta[t] = \mathbf{P}(\mathbf{X}[t] \mid \mathbf{X}[t-1], \ldots, \mathbf{X}[0])$. Generally, the first-order Markov assumption is applied resulting in $\Theta[t] = \mathbf{P}(\mathbf{X}[t] \mid \mathbf{X}[t-1])$. Another assumption generally applied is that the distribution is stationary. In other words, the variables in the set $Parents(X_i[t])$ are independent of the specific value t. Also, the parameters $\Theta[t] = \mathbf{P}(X_i[t] \mid Parents(X_i[t]))$ are also independent of the specific value t. With the assumption that the time series begins with $t = 0$, there is a caveat to the previous rule since marginal and conditional probabilities for the variables are necessary to define the initial distribution at $t = 0$ where no temporal arcs to a previous time-step exist.

Using these assumptions, a DBN can be defined by providing the node, arc, and parameter definitions defining the distributions at $t = 0$ and $t \neq 0$. Using a DBN then requires "unrolling" the network to the highest time slice required. This "unrolling" creates individual nodes for each variable at every time slice and adds in the appropriate arcs defined previously. In effect, this creates a flat Bayesian network over which standard querying and inference techniques can be applied.

Inference and Simplification

The general problem of inference concerns a given query $P(\mathbf{Y} \mid \mathbf{Z})$ wherein a posterior distribution must be calculated over the variables in $\mathbf{Y}$ with respect to evidence given for the variables in $\mathbf{Z}$. Approaches to solving this problem can attempt to use either exact [46, 47, 48, 49] or approximate inference [8, 50, 51, 52, 53, 54]. As mentioned before, however, performing exact inference is an NP-hard problem. Thus, especially for large diagnostic networks such as the QMR database, exact inference is intractable.

While attempting to perform model simplification, a primary goal is to reduce the effective computation requirement of performing exact inference so that an exact method can be run over an approximate model. Consider the two exact inference algorithms Lauritzen-Spiegelhalter and Hugin [46, 48]. Both of these methods perform exact inference by creating a junction tree where each node contains a subset of the variables within the original network. A requirement is imposed such that if a variable is located within two different nodes, then it must be in every other node on the path between the nodes. This tree is created by first moralizing graph by adding edges connecting the parents of any node $t_i \in \mathbf{T}$. Afterwards, the resulting graph is triangulated by adding edges between a pair of variables where those variables define a chord, or minimal loop, of four or more variables, which is a computationally difficult problem to perform optimally. With a triangulated graph, it then becomes possible to easily determine the cliques and decompose the result into a tree where the nodes represent cliques within the graph where computational complexity is based upon the size of the largest cliques (or tree-width). Therefore, in order to reduce the complexity of the

simplified system, it must be shown that the resulting network will have a small tree-width than the original network.

RELATED WORK

Model Approximation

In contrast to performing approximate inference over a Bayesian network, varying methods for approximating the network while allowing for exact inference have been attempted. One of the earlier attempts at performing model approximation by edge, or arc, deletion was proposed by van Engelen [55]. In this work, arcs were selected for deletion by balancing the divergence of the networks caused by the arcs with the number of loops the arc is involved in within the graph generated for inference. Additionally, a requirement was imposed that limited the set of arcs chosen for deletion such that no two arcs deleted had the same parent. Specifically, this is a requirement in providing a bound on the difference in posterior distributions. However, the method is not sensitive to the current set of evidence. More importantly, it is noted by the author that in cases where the probability of the evidence is low and the arcs in the network represent strong dependencies, the resulting method can fail to approximate the network adequately. For the purposes of prognostics and diagnostics, this is a sizable problem since diagnostic networks frequently have just such cases. Similarly, Kjærulff proposed a method of removing weak dependencies [56, 57]. Unlike the previous method, the weak dependencies selected were those contained in the moralized graph.

A more recent approach attempts to remove arcs based upon the junction tree created by moralizing and triangulating a Bayesian network prior to performing exact inference [58]. In doing so, the algorithm is given as input a user-specified upper bound on the

clique size desired in the triangulated graph. Based on this requirement, induced edges from moralization as well as the original edges are selected for deletion based upon moral edges which are unique to a single variable's parents and which original edge removals allow for the most removals of moral edges. Since the treewidth of the network is then bounded, inference can be performed in effectively polynomial time. However, although there is reference to determining the root mean squared error of the posterior distributions, no mention is made to setting evidence in the network.

A considerable amount of research in this area has been performed by Choi and Darwiche [59, 60, 61, 62, 63, 64]. The basis of their work relies on the intuition that, in the presence of evidence, near-deterministic distributions on nodes provide likely candidates for the removal of arcs. To do so, however, requires updating the parameters of the nodes such that the near-deterministic distribution is preserved. Various experiments have been performed in differing ways to update the parameters, such as with variational methods, as well as splitting nodes in addition to edge deletions to break loops within the network.

Hidden Variable Discovery

Another method of performing simplification which has not been explored in the previously mentioned edge deletion literature is that of determining hidden variables within the network. As an introductory example, a sample was given by Russell, Binder, Koller, and Kanazawa [65] showing that a network with hidden variables can be more compact than one without the hidden variables. Unfortunately, in most cases, determining the likelihood of

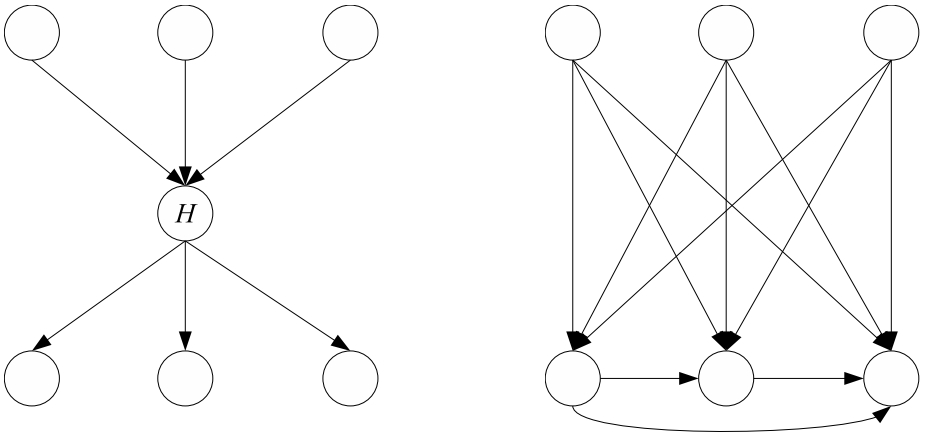
network structures in the presence of hidden variables is difficult, resulting in approximations being developed to improve the rate of learning [66, 67].

Early approaches to learning these hidden variables used a gradient descent approach over the conditional probability tables [65] or incorporating hidden variables into the structural EM procedure [68]. A more sophisticated technique is available when performing structure learning in a Bayesian network with continuous variables [69]. By calculating an ideal parent for a variable based upon the linear Gaussians, if no suitable potential parent already exists, a new variable is created which will match that ideal value and be inserted as the parent of that node.

However, the most relevant method for finding hidden variables, with regards to the logical closure and transitive reduction process, was developed by Elidan as a structure guided search [70, 71, 72]. The structurally guided search locates subsets of variables within the graph which nearly makeup a clique, using the intuition that those areas are likely candidates for the insertion of a hidden variable, as seen in Figure 3. Once inserted, the cardinality of the hidden variable is separately calculated by successive combining of states based upon a likelihood statistic.

Diffusion of Context

One difficulty in performing prognostics is related to the problem of Markovian models suffering from diffusion of context [73]. This problem is especially relevant to dynamic



(a) Network with Hidden Variable H

(b) Simplest Network Representing the Same Dependencies

Figure 3. Model Simplification by Hidden Variables.

Bayesian networks since they are a specialized form of a Markov model. Attempts at mitigating this problem have been applied to both Markovian models [74, 75] and recurrent neural networks [76, 77].

Consider a general first-order Markovian model which is defined by a set of states $\{s_1, \ldots, s_m\}$, a transition function, and a probability output function. Additionally, consider a matrix A defining the transition probabilities where $A_{i,j}$ is the probability of transitioning from state i to state j over a single time-step. In terms of the prior variables, let $\mathbf{X}[t]$ be defined as before as the set of all variables $\mathbf{X}$ at time t. Then, let $\mathbf{X}[t]^s$ define the specific setting of values to all variables. From this, the transition probability of state i to state j is listed in the matrix as $A_{i,j} = P\left(\mathbf{X}[t]^j \mid \mathbf{X}[t-1]^i\right)$.

By using this matrix, Bengio and Frasconi were able to use a form of the Perron-Frobenius theorem to prove the susceptibility of Markov models to diffusion of context. If A is a primitive stochastic matrix, then as t approaches ∞, A^t approaches the unique stationary

distribution of the Markov model. In addition, the rate of approach is geometric. To show that this holds for the Markov model, matrix A must be a primitive stochastic matrix. For the matrix to be primitive, the matrix must be non-negative and there must exist a positive integer k such that $A^k \geq 0$, or $A^k_{i,j} \geq 0$ for all i and j. First, since all entries in A represent transition probabilities, by definition the values are required to be in the range $[0, 1]$. The proof of the second portion is more complex, but it relies on the fact that the matrix A will be irreducible, and therefore, for every pair of indexes, there exists some k such that $A^k_{i,j} \geq 0$. Finally, a matrix is considered to be row stochastic if

$$\sum_{j=1}^{m} A_{i,j} = 1 \quad \forall i \in [1, m]$$

Again, by the definition of probabilities, this is required to be true as the sum of probabilities from one state to all other states must be equal to one.

Now consider a current state vector and the transition matrix. The resulting state distribution is dependent on successive multiplications of the transition matrix, given by A^k. Given the prior theorem, the state distribution moves toward the stationary distribution at a geometric rate, losing the information of the current state vector. This is a significant problem when attempting to perform prognostics in the presence of evidence and prior history. As can be seen in the preliminary results, it is also possible that this phenomenon can occur with a specialized form of diagnostic networks. It remains to be determined if the same result will apply to generalized networks simplified by the proposed method.

EXPERIMENTAL DESIGN

Model Simplification

For diagnostic networks, one preliminary method explored for performing model simplification involves the procedure of logical closure and transitive reduction. As is shown, however, this method by itself is insufficient for providing bounded error when performing simplification on general diagnostic Bayesian networks. Due to this issue, after performing the simplification by logical closure and transitive reduction, additional steps must be taken to correct the model, especially in the presence of evidence.

Logical Closure

Consider a deterministic diagnostic network with a set of tests $\mathbf{T}$ and a set of diagnoses $\mathbf{D}$. With a deterministic network wherein a fault has occurred, the probability distribution on the test nodes which are children of that fault, or diagnosis, require that the test must be active. Let t_i be a test with parent diagnoses $\mathbf{d}_i$ and let a_j be a test with parent diagnoses $\mathbf{d}_j$. Furthermore, assume $\mathbf{d}_i \subseteq \mathbf{d}_j$ and a current fault $d_x \in \mathbf{d}_i$. From the previous definition, d_x is a parent of a_i, and therefore, a_i must be active. Additionally, since $d_x \in \mathbf{d}_i$ and $\mathbf{d}_i \subseteq \mathbf{d}_j$, d_x is also a parent of a_j. Thus, it must also be true that a_j must be active. Alternatively, consider the process of diagnosis with the test a_i having been performed, resulting in an positive indication of a fault. This indicates a diagnosis d_x in its parent set which is a subset of the parents of a_j. Therefore, a_j must also indicate a fault. Based on this insight, an arc can be added from a_i to a_j as a potential fault result on a_i informs a_j.

Transitive Reduction

Following the logical closure procedure, transitive reduction simplifies the network by removing the transitive links from the network [78]. Specifically, consider any three nodes in the network X_i, X_j, X_k drawn from $\mathbf{X}$ which includes both the diagnosis and test nodes. Transitive reduction proceeds by deleting any directed arc $X_i \rightarrow X_k$ where there exists arcs $X_i \rightarrow X_j \rightarrow X_k$. In a deterministic network, the resulting simplified network will provide the same classifications as the original network in any valid setting of evidence. This directly follows from the logical closure formulation from above.

However, for non-deterministic networks, without some correction factor, the simplified network will not have the same posterior distribution as the original network under evidence. This is due to the change in the independence relationships within the graph. Correcting for this problem will come in three separate methods: hidden variable introduction, dynamic parameter adjustment, and arc reinsertion.

Model Correction

The two primary methods intended to be used for correcting the posterior distribution of the network are hidden variable introduction and dynamic parameter adjustment. Consider two variables t_i and t_j where, prior to simplification, the parents of t_i were a subset of t_j. In this case, during the normal procedure of moralizing and triangulating the graph for performing exact inference, the parents of t_i as well as the nodes t_i and t_j form a near-clique. Based on principles from Elidan's prior work, this becomes a likely candidate for the insertion of a hidden variable. However, an additional ordering will be imposed such that variables

which are likely to suffer from diffusion of context, as seen in the preliminary results, will have priority for the insertion of a hidden variable in order to mitigate the geometric rate of loss of information.

Following this procedure, techniques developed in calculating error-bars during inference will be applied in attempt to isolate queries which have a high probability of large error [79]. Based upon these values, the parameters of the variables within the query will be updated to improve the final results of the query. In the event that this procedure fails to provide the desired level of accuracy, arcs deleted from the network previously will be reintroduced to the network which is followed by additional parameter adjustments.

Data Sets

In testing the complete system, a combination of real world and synthetic data sets will be used. For the real world data, a selection of problems will be drawn, such as the standard ALARM and INSURANCE networks, in order to facilitate comparison with other edge deletion and approximation algorithm techniques. The synthetic datasets will draw on prior work in the analysis of creating random networks which have controllable complexity when performing inference [80, 81, 82, 83]. Prior work performed by Elidan also provides a method for learning Bayesian networks from data with a bounded treewidth [84]. Such a technique will allow for similar testing on the real world data as that which will be performed with the synthetic data.

Schedule

The following is a work schedule for progressing on the thesis. All models and proofs will be verified over multiple variable and parameter representations: multinomial, discrete, continuous, and noisy OR-gates as some examples.

1. Computational Complexity Reduction Proof

   First, a formal proof must be developed which shows that performing logical closure and transitive reduction will result in a network that, assuming optimal triangulation, will have a smaller tree-width. It can easily be shown that the resulting parent set of the test variables in the simplified network is less than the parent set in the original network. It remains to be proven that a triangulation of the moralized graph will guarantee a lower tree-width.

2. Generalized Logical Closure and Generalized Transitive Reduction

   After developing the formal proof, a generalized implementation of the base simplification procedure is necessary which will be able to handle diagnostic and standard Bayesian networks. Doing so will require a deeper analysis of the network and parameters than what was performed in the preliminary experiments. Since not all networks follow the same rigid formulation which was used in the initial experiments, performing the simplification must ensure that correct state interpretations are used for determining test implications. To do so requires an analysis of the conditional probability distribution to identify the states which most strongly indicate parent states.

3. Develop Generalized Bayesian Network Inference Engine

   Due to the nature of the research, a functional inference engine must be developed incorporating both exact inference and approximate inference algorithms in order to test the efficacy of the simplification procedure.

4. Transitive Reduction and Diffusion of Context in Subgraphs Proof

   Briefly mentioned earlier, and elaborated upon slightly in the preliminary results, given specific forms of a diagnostic Bayesian network, it is possible to show that the resulting simplified network suffers from the issue of diffusion of context when performing classification. Based upon the subset relationships of the test nodes, a proof of the conditions under which subgraphs of the network undergo diffusion must be analyzed.

5. Structure Guided Hidden Variable Discovery

   Based upon the prior stages, the next task is to introduce hidden variables into the simplified network. The placement of the variables will be initially selected based upon near-clique sets of nodes in the triangulated network as well as identified areas subject to diffusion of context as determined above. After initial placement, structural refinement will occur to improve the resulting accuracy.

6. Parameter Modification Formulation

   Following hidden variable introduction, tests will be performed by querying the network and determining the error of the query response based upon the methods used in [79].

7. Grouping Analysis of Test Subsets

The final modification procedure during the diagnostic phase first requires analyzing groupings of tests to determine subsets of tests that are necessary to obtain higher classification accuracy. Initial attempts at determining these subsets were performed in the preliminary results with extracting decision trees from the network for test selection.

8. Error-Bounding Proof of Posterior Distribution

   Finally, a formal error-bounding proof of the posterior distribution must be shown in the presence of the above modifications.

9. Empirical Analysis of Simplification Procedure

   The empirical analysis of the diagnostic system will require testing over a variety of networks with varying tree-width, both based on real world networks and synthetic networks. A comparison will be performed with approximate inference methods as well as other arc deletion model approximations.

The primary goal of the research is covered by the above items. However, future efforts will expand the results to incorporate prognostics with dynamic Bayesian networks. If the work is complete within the appropriate time constraints, the following work will be pursued and incorporated into the dissertation.

10. Dynamic Bayesian Network Structural Simplification

    After completion of the analysis for diagnostics, a similar simplification procedure must be developed for dynamic Bayesian networks. Given the potential size of a fully

"unrolled" network, the goal of simplification will be to create a smaller flat network by simplifying the DBN specification instead of attempting to simplify the full network and learning hidden variables within a network with potentially thousands of nodes and arcs.

11. Hidden Variable Discovery in Dynamic Bayesian Networks for Long-Term Memory

    Given the especially difficult task of performing prognostics, structure and data based hidden variable discovery will be performed to reduce the impact of diffusion of context.

12. Parameter Modification for Dynamic Bayesian Networks

    Similar to the procedure for modifying parameters in the Bayesian network, dynamic updating of the resulting network will be performed to improve the accuracy of prognostics. In addition to improving accuracy, parameter modification for DBNs can also work to mitigate diffusion.

13. Error-Bounding Proof and Empirical Analysis of Prognostics Simplification

    The final stage of the thesis will be to provide error bounds in the dynamic Bayesian network and empirically verify the results with a range of test networks.

PRELIMINARY RESULTS

In preparation for this proposal, some preliminary experiments have been performed in diagnostic network simplification. For the both of these experiments, random bipartite networks were made which represented diagnostic Bayesian networks. In their creation, parents of the test nodes were selected randomly. Additionally, parameters were also selected randomly where the parameter value given is representative of the false positive and false negative rate of the given test.

Transitive Reduction Accuracy

In examining the results of performing diagnostic network simplification by logical closure and transitive reduction, multiple different tests were performed based on connectivity and parameter range. Another test was performed over networks which give the largest simplification: serial networks. These serial networks are defined where the parents of some test t_i are the set of diagnoses $\{d_j : \forall j \leq i\}$.

In the tables which follow, the columns for the mean, variance, and standard deviation are calculated by the average number of parents for each test node. It is provided as an indication of the general complexity of the network. The accuracy columns provide the network's accuracy in correctly classifying a single fault provided evidence of its indicator tests.

As can be seen empirically from the results of these experiments, the degradation in approximation ability increases with the removal of additional arcs, which is to be expected.

Connectivity	Original Network				Simplified Network			
	Mean	Variance	St. Dev.	Acc.	Mean	Variance	St. Dev.	Acc.
0.10	1.740	0.786	0.879	0.911	1.700	0.705	0.825	0.900
0.30	3.450	4.129	2.010	0.969	3.250	3.464	1.839	0.922
0.500	5.650	8.257	2.861	1.000	5.280	6.685	2.578	0.871
0.700	7.860	17.021	4.120	1.000	6.710	9.566	3.088	0.915
0.900	10.470	25.016	4.994	1.000	7.820	11.934	3.443	0.846

Table 1. Results of performing simplification with varying connectivity.

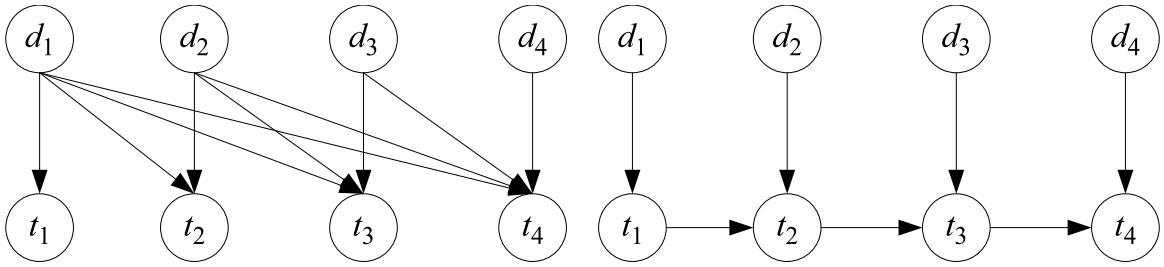
Parameters	Original Network				Simplified Network			
	Mean	Variance	St. Dev.	Acc.	Mean	Variance	St. Dev.	Acc.
0.00	3.840	4.283	2.058	1.000	3.600	3.235	1.788	0.960
0.20	3.680	4.740	2.162	0.988	3.530	4.197	2.021	0.960
0.40	3.430	3.527	1.865	0.958	3.300	3.063	1.730	0.876
0.60	3.500	3.833	1.930	0.841	3.370	3.222	1.760	0.761
0.80	3.350	4.007	1.989	0.362	3.090	2.737	1.649	0.253

Table 2. Results of performing simplification with varying parameter range.

Size	Original Network				Simplified Network			
	Mean	Variance	St. Dev.	Acc.	Mean	Variance	St. Dev.	Acc.
5×5	3.000	2.000	1.414	0.890	1.800	0.160	0.400	0.598
10×10	5.500	8.250	2.872	0.938	1.900	0.090	0.300	0.202
15×15	8.000	18.667	4.320	0.909	1.933	0.062	0.249	0.159
20×20	10.500	33.250	5.766	0.917	1.950	0.048	0.218	0.114

Table 3. Results of performing simplification with serial networks.

Except in the cases of significant reduction in the model complexity, the performance of the simplified network was comparable, though significantly different than the original network. Another interesting result is in the serial networks. Examining the accuracy values, it appears the accuracy decreases at a geometric rate. Analyzing the resulting networks, the cause of this dramatic reduction in accuracy is evident. Consider Figure 4 which shows a simplified serial network as an example. Reworking the semantics of the network slightly,



(a) Serial Network (b) Unclosed Serial Network

Figure 4. Result of Logical Closure and Transitive Reduction on a Serial Network.

this simplification can be made to represent a Markovian model as described by Bengio and Frasconi. Therefore, the issue of diffusion of context will cause a geometric rate of information loss, as is shown empirically in the results. This result emphasizes the importance in mitigating this issue for both the diagnostic and prognostic problems.

Decision Tree Extraction

In an attempt to determine the likelihood of reinserting arcs into the network to mitigate the issues shown in the previous experiment, a follow-up experiment was performed in an attempt to isolate subsets of variables which together obtain a high accuracy in performing diagnostics. Such groups of variables can then be used to test the resulting networks by allowing for swifter testing by setting evidence in the subset of variables instead of many combinations of the entire set. The results of the research was published in the Prognostics and Health Management conference [85]. The basis of the following experiment was to extract decision trees from the network by using varying methods of calculating information gain for selecting the partitions and creating the tree. Three different methods were originally selected based on their prior use in past experiments and analysis: forward sampling, maximum

expected utility, and KL-divergence. Additionally, three new methods were proposed based upon the structure of the D-Matrix representation of the diagnostic network.

First, the forward sampling method created a dataset from the network based upon sampling single faults. Based on the marginal probabilities of the diagnostic nodes in **D**, a single fault was selected and set to TRUE while the remainder of the nodes were set to FALSE. Given the bipartite structure of the network, all of the test nodes in **T** could be sampled based strictly on their conditional probability tables without performing additional inference or belief propagation. With a complete sample created over all individuals, the result was added to the database. Given the database, the standard multi-class version of ID3 generates a decision tree by calculating the information gain given by

$$I\left(d_1, \ldots, d_n\right) = -\sum_{i=1}^{n} \frac{d_i}{d_1 + \cdots + d_n} \lg \frac{d_i}{d_1 + \cdots + d_n}$$

and

$$E\left(t\right) = \sum_{i=1}^{arity(t)} \frac{d_{i,1} + \cdots + d_{i,n}}{d_1 + \cdots + d_n} I\left(d_{i,1}, \ldots, d_{i,n}\right)$$

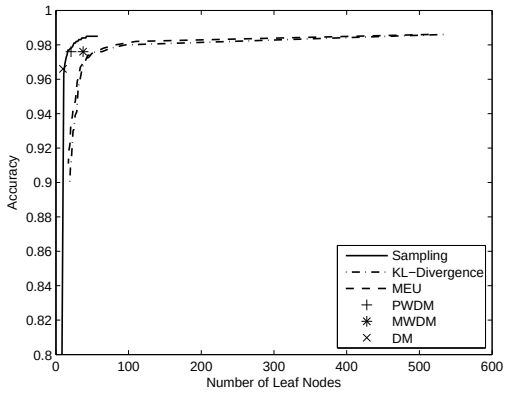
where d_i represents the count of that diagnosis class in the partition and t is the test node being evaluated.

The maximum expected utility and KL-Divergence approaches use a different approach wherein inference is performed on the network in order to determine the utility and divergence gained by selecting tests, respectively. The D-Matrix based approaches are more similar to the forward sampling approach wherein a small dataset is created by treating every row in the D-Matrix as an individual in the data. The simplest of the approaches (DM) uses solely this information and performs the standard ID3 calculations, as in forward sampling,

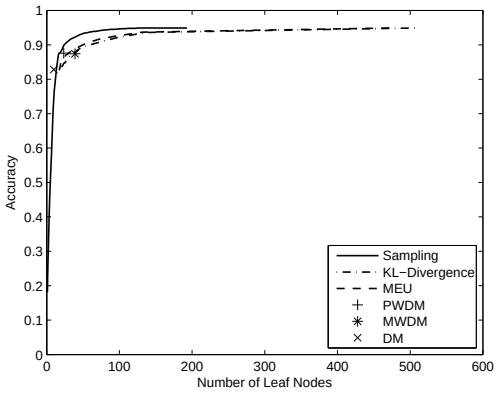
partitioning the D-Matrix at every level. Another method, PWDM, weights the information gain by determining the probability of the individuals, or diagnoses, in the partition which must be determined by performing inference. The MWDM approach instead simply weights the individual based upon their marginal distributions, resulting in a very fast decision tree creation.

Given the potential size of the trees, especially for the MEU and KL-Divergence approaches, successively aggressive pruning was applied to the trees to analyze the effect of the tree on classification accuracy. The results of the experiment are shown in the figures 5 and 6. Both sets of graphs represents the accuracy of performing inference in the Bayesian network after setting evidence recommended to it by the resulting decision tree. The first set compares this accuracy to the overall size of the network, while the second set compares the accuracy to the number of recommended tests. Like in the previous experiment, networks with varying parameter ranges were tested to empirically test the effect of the level of near-determinism in the network.

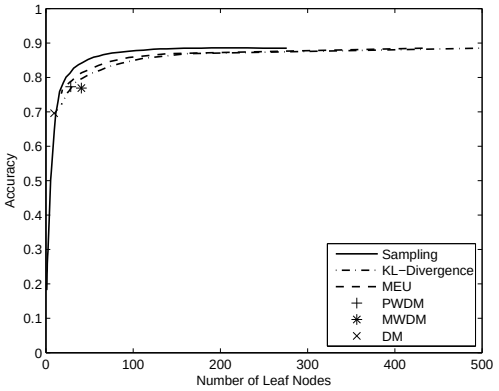
As can be seen in the first set of graphs, forward sampling in general outperforms the other methods, although many of the differences in performance are negligible. The second set of graphs shows a larger gap between the performance of the other algorithms when compared with the forward sampling approach. Thus, with an appropriate heuristic, it was shown that reasonable accuracy could be obtained while selecting a small number of tests for use in performing test selection.



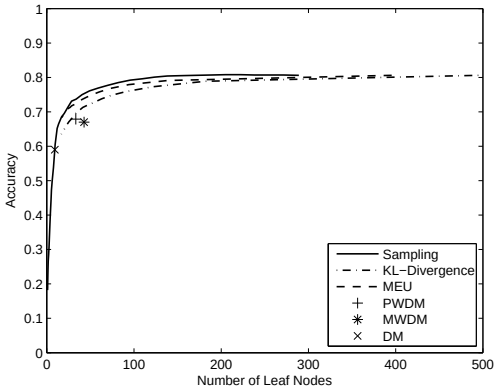
(a) Average accuracy for networks with parameters in the range [0.001,0.01]



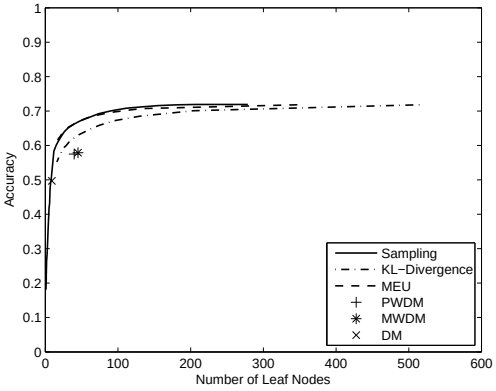
(b) Average accuracy for networks with parameters in the range [0.001,0.10]



(c) Average accuracy for networks with parameters in the range [0.001,0.20]

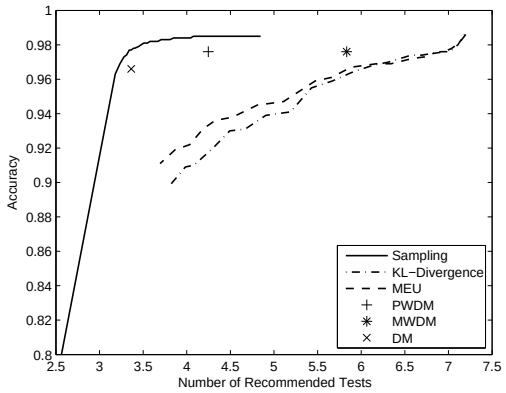


(d) Average accuracy for networks with parameters in the range [0.001,0.30]

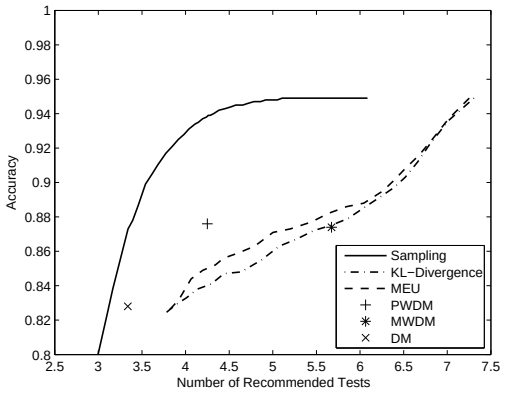


(e) Average accuracy for networks with parameters in the range [0.001,0.40]

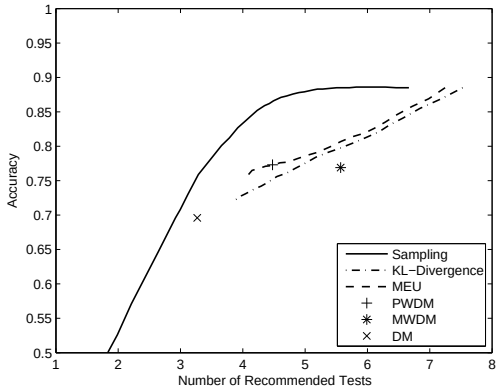
Figure 5. Accuracy of decision trees created from networks with varying parameters with respect to tree size.



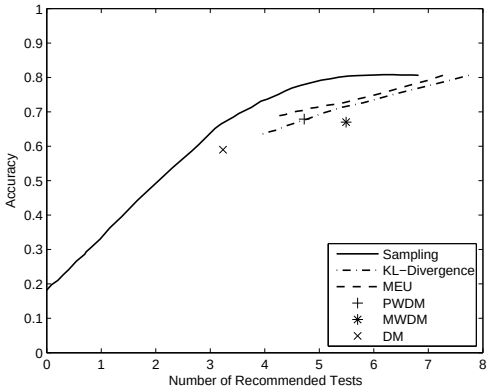
(a) Average accuracy for networks with parameters in the range [0.001,0.01]



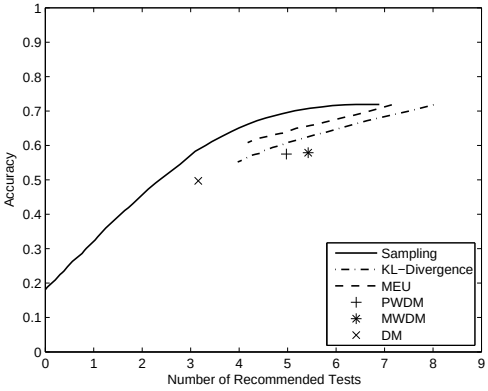
(b) Average accuracy for networks with parameters in the range [0.001,0.10]



(c) Average accuracy for networks with parameters in the range [0.001,0.20]



(d) Average accuracy for networks with parameters in the range [0.001,0.30]



(e) Average accuracy for networks with parameters in the range [0.001,0.40]

Figure 6. Accuracy of decision trees created from networks with varying parameters with respect to the number of recommended tests.

REFERENCES CITED

[1] B. Saha and K. Goebel, "Uncertainty management for diagnostics and prognostics of batteries using Bayesian techniques," in *Proceedings of the 2008 IEEE Aerospace Conference*, pp. 1–8, 2008.

[2] K. Goebel, B. Saha, and A. Saxena, "A comparison of three data-driven techniques for prognostics," in *Proceedings of the 62nd Meeting of the Society for Machinery Failure Prevention Technology*, pp. 119–131, April 2008.

[3] B. P. L. ao, T. Yoneyama, G. C. Rocha, and K. T. Fitzgibbon, "Prognostic performance metrics and their relation to requirements, design, verification and cost-benefit," in *Proceedings of the 2008 International Conference on Prognostics and Health Management*, 2008.

[4] J. W. Sheppard, M. A. Kaufman, and T. J. Wilmering, "IEEE standards for prognostics and health management," in *AUTOTESTCON*, pp. 97–103, 2008.

[5] S. Uckun, K. Goebel, and P. J. F. Lucas, "Standardizing research methods for prognostics," in *Proceedings of the 2008 International Conference on Prognostics and Health Management*, 2008.

[6] J. Luo, M. Namburu, K. Pattipati, L. Qiao, M. Kawamoto, and S. Chigusa, "Model-based prognostic techniques," in *Proceedings of the 2003 IEEE Systems Readiness Technology Conference AUTOTESTCON*, pp. 330–340, September 2003.

[7] G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial Intelligence*, vol. 42, pp. 393–405, March 1990.

[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[9] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, 1992.

[10] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, pp. 197–243, 1995.

[11] N. Friedman and M. Goldszmidt, "Building classifiers using Bayesian networks," in *Proceedings of the 13th National Conference on Artificial Intelligence*, 1996.

[12] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131–163, 1997.

[13] N. Friedman, I. Nachman, and D. Peér, "Learning Bayesian network structure from massive datasets: The 'Sparse Candidate' algorithm," in *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, 1999.

[14] D. A. McAllester, "PAC-Bayesian stochastic model selection," *Machine Learning*, vol. 51, pp. 5–21, 2003.

[15] R. S. Niculescu, *Exploiting Parameter Domain Knowledge for Learning in Bayesian Networks*. PhD thesis, Carnegie Mellon University, 2005.

[16] N. Friedman, "The Bayesian structural em algorithm," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.

[17] L. R. Peraza and D. M. Halliday, "Fourier Bayesian information criterion for network structure and causality estimation," in *International Conference on Signals and Electronic Systems*, pp. 33–36, IEEE, September 2010.

[18] A. Delaplace, T. Brouard, and H. Cardot, "Two evolutionary methods for learning Bayesian network structures," in *2006 International Conference on Computational Intelligence and Security*, vol. 1, pp. 137–142, November 2006.

[19] T. Grégory, B. Stéphane, and A. Alexandre, "Learning Bayesian network structures by estimation of distribution algorithms: An experimental analysis," in *2nd International Conference on Digital Information Management*, vol. 1, pp. 127–132, October 2007.

[20] N. Friedman and D. Koller, "Being Bayesian about network structure," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 2000.

[21] N. Friedman and D. Koller, "Being Bayesian about Bayesian network structure: A Bayesian approach to structure discovery in Bayesian networks," *Machine Learning*, vol. 50, no. 1–2, pp. 95–125, 2003.

[22] M. Koivisto and K. Sood, "Exact Bayesian structure discovery in Bayesian networks," *Journal of Machine Learning Research*, vol. 5, pp. 549–573, December 2004.

[23] N. Friedman and M. Goldszmidt, "Sequential update of Bayesian network structure," in *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, 1997.

[24] N. Friedman and M. Goldszmidt, "Learning Bayesian networks with local structure," in *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, 1996.

[25] N. Friedman and M. Goldszmidt, *Learning in Graphical Models*, ch. Learning Bayesian Networks with Local Structure, pp. 421–459. Cambridge, MA, USA: MIT Press, 1999.

[26] N. Friedman and Z. Yakhini, "On the sample complexity of learning Bayesian networks," in *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, 1996.

[27] M. A. Shwe, B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz, H. P. Lehmann, and G. F. Cooper, "Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base: I. The probabilistic model and inference algorithms," *Methods of Information in Medicine*, vol. 30, pp. 256–267, 1991.

[28] W. R. Simpson and J. W. Sheppard, *System Test and Diagnosis*. Norwell, MA: Kluwer Academic Publishers, 1994.

[29] D. Heckerman, J. S. Breese, and K. Rommelse, "Decision-theoretic troubleshooting," *Communications of ACM*, vol. 38, pp. 49–57, March 1995.

[30] C. Skaanning, F. V. Jensen, and U. Kjærulff, "Printer troubleshooting using Bayesian networks," in *Proceedings of the 13th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp. 367–379, 2000.

[31] F. V. Jensen, U. Kjærulff, B. Kristiansen, H. Langseth, C. Skaanning, J. Vomlel, and M. Vomlelová, "The SACSO methodology for troubleshooting complex systems," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 15, pp. 321–333, 2001.

[32] J. Vomlel, "Two applications of Bayesian networks," in *Proceedings of Conference Znalosti*, pp. 73–82, February 2003.

[33] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proceedings of the 12th International Conference on Machine Learning*, pp. 194–202, 1995.

[34] N. Friedman and M. Goldszmidt, "Discretizing continuous attributes while learning Bayesian networks," in *Proceedings of the 13th International Conference on Machine Learning*, 1996.

[35] N. Friedman, M. Goldszmidt, and T. J. Lee, "Bayesian network classification with continuous attributes: Getting the best of both discretization and parametric fitting," in *Proceedings of the 15th International Conference on Machine Learning*, 1998.

[36] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, 1995.

[37] S. Davies and A. Moore, "Mix-nets: Factored mixtures of Gaussians in Bayesian networks with mixed continuous and discrete variables," in *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pp. 168–175, 2000.

[38] F. J. Díez, "Parameter adjustment in Bayes networks. The generalized noisy OR-gate," in *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pp. 99–105, 1993.

[39] S. Srinivas, "A generalization of the noisy-or model," in *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pp. 208–218, 1993.

[40] F. J. Díez and S. F. Galán, "Efficient computation for the noisy max," *International Journal of Intelligent Systems*, vol. 18, pp. 165–177, 2003.

[41] J. Vomlel, "Noisy-or classifier," *International Journal of Intelligent Systems*, vol. 21, pp. 381–398, 2006.

[42] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," tech. rep., Brown University, 1989.

[43] Z. Ghahramani, "Learning dynamic Bayesian networks," in *Adaptive Processing of Temporal Information*, 1997.

[44] N. Friedman, K. Murphy, and S. Russell, "Learning the structure of dynamic probabilistic networks," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.

[45] C. P. de Campos, Z. Zeng, and Q. Ji, "An improved structural em to learn dynamic Bayesian nets," in *Proceedings of the 20th International Conference on Patter Recognition*, pp. 601–604, August 2010.

[46] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society*, pp. 157–224, 1988.

[47] C. Huang and A. Darwiche, "Inference in belief networks: A procedural guide," *International Journal of Approximate Reasoning*, vol. 11, pp. 1–45, 1994.

[48] V. Lepar and P. P. Shenoy, "A comparison of Lauritzen-Spiegelhalter, Hugin, and Shenoy-Shafer architectures for computing marginals of probability distributions,"

in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pp. 328–337, Morgan Kaufmann, 1998.

[49] W. Liao, W. Zhang, and Q. Ji, "A factor tree inference algorithm for Bayesian networks and its applications," in *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, 2004.

[50] A. F. M. Smith and G. O. Roberts, "Bayesian computation via the gibbs sampler and related markov chain monte carlo methods," *Journal of the Royal Statistical Society*, vol. 55, no. 1, pp. 3–23, 1993.

[51] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, pp. 183–233, 1999.

[52] T. S. Jaakkola and M. I. Jordan, "Variational probabilistic inference and the QMR-DT network," *Journal of Artificial Intelligence Research*, vol. 10, pp. 75–87, 1999.

[53] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 689–695, 2000.

[54] I. Rish, M. Brodie, and S. Ma, "Accuracy vs. efficiency trade-offs in probabilistic diagnosis," in *Proceedings of the 18th National Conference on Artificial Intelligence*, pp. 560–566, 2002.

[55] R. A. van Engelen, "Approximating Bayesian belief networks by arc removal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 916–920, August 1997.

[56] U. Kjærulff, "Approximation of Bayesian networks through edge removals," tech. rep., Department of Mathematics and Computer Science, Aalborg University, 1993.

[57] U. Kjærulff, "Reduction of computational complexity in Bayesian networks through removal of weak dependencies," in *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pp. 374–382, 1994.

[58] J. A. Thornton, "Approximate inference of Bayesian networks through edge deletion," Master's thesis, Kansas State University, 2005.

[59] A. Choi, H. Chan, and A. Darwiche, "On Bayesian network approximation by edge deletion," in *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pp. 128–135, 2005.

[60] A. Choi and A. Darwiche, "A variational approach for approximating Bayesian networks by edge deletion," in *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pp. 80–89, 2006.

[61] A. Choi and A. Darwiche, "An edge deletion semantics for belief propagation and its practical impact on approximation quality," in *Proceedings of the 21st National Conference on Artificial Intelligence*, pp. 1107–1114, 2006.

[62] A. Choi, M. Chavira, and A. Darwiche, "Node splitting: A scheme for generating upper bounds in Bayesian networks," in *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pp. 57–66, 2007.

[63] A. Choi and A. Darwiche, "Approximating the partition function by deleting and then correcting for model edges," in *Proceedings of othe 24th Conference on Uncertainty in Artificial Intelligence*, pp. 79–87, 2008.

[64] A. Choi and A. Darwiche, "Relax then compensate: On max-product belief propagation and more," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pp. 351–359, 2009.

[65] S. Russell, J. Binder, D. Koller, and K. Kanazawa, "Local learning in probabilistic networks with hidden variables," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 1146–1152, 1995.

[66] D. M. Chickering and D. Heckerman, "Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables," *Machine Learning*, vol. 29, pp. 181–212, 1997.

[67] M. J. Beal and Z. Ghahramani, "Variational Bayesian learning of directed graphical models with hidden variables," *Bayesian Analysis*, vol. 1, no. 4, pp. 793–832, 2006.

[68] N. Friedman, "Learning belief networks in the presence of missing values and hidden variables," in *Proceedings of the 14th International Conference on Machine Learning*, 1997.

[69] G. Elidan, I. Nachman, and N. Friedman, "'Ideal parent' structure learning for continuous variable Bayesian networks," *Journal of Machine Learning Research*, vol. 8, pp. 1799–1833, 2007.

[70] G. Elidan, N. Lotner, N. Friedman, and D. Koller, "Discovering hidden variables: A structure-based approach," *Neural Information Processing Systems*, pp. 479–485, 2000.

[71] G. Elidan and N. Friedman, "Learning the dimensionality of hidden variables," in *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, 2001.

[72] G. Elidan, *Learning Hidden Variables in Probabilistic Graphical Models*. PhD thesis, Hebrew University, 2004.

[73] Y. Bengio and P. Frasconi, "Diffusion of context and credit information in Markovian models," *Journal of Artificial Intelligence Research*, vol. 3, pp. 249–270, 1995.

[74] Y. Bengio and P. Frasconi, "An input output HMM architecture," in *Proceedings of the Conference on Neural Information Processing Systems*, pp. 427–434, 1994.

[75] J. Callut and P. Dupont, "Inducing hidden Markov models to model long-term dependencies," in *European Conference on Machine Learning*, pp. 513–521, 2005.

[76] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[77] J. Chen and N. S. Chaudhari, "Segmented-memory recurrent neural networks," in *IEEE Transactions on Neural Networks*, vol. 20, pp. 1267–1280, 2009.

[78] A. V. Aho, M. R. Garey, and J. D. Ullman, "The transitive reduction of a directed graph," *SIAM*, pp. 131–137, 1972.

[79] T. V. Allen, A. Singh, R. Greiner, and P. Hooper, "Quantifying the uncertainty of a belief net response: Bayesian error-bars for belief net inference," *Artificial Intelligence*, vol. 172, pp. 483–513, 2008.

[80] J. S. Ide and F. G. Cozman, "Random generation of bayesian networks," in *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence*, pp. 366–375, 2002.

[81] J. S. Ide, F. G. Cozman, and F. T. Ramos, "Generation of random Bayesian networks with constraints on induced width with application to the average analysis of d-connectivity, quasi-random sampling, and loopy propagation," tech. rep., University of São Paulo, 2003.

[82] J. S. Ide, F. G. Cozman, and F. T. Ramos, "Generating random Bayesian networks with constraints on induced width," in *Proceedings of the 16th European Conference on Artificial Intelligence*, pp. 118–127, 2004.

[83] O. J. Mengshoel and D. C. Wilkins, "Controlled generation of hard and easy Bayesian networks: Impact on maximal clique size in tree clustering," *Artificial Intelligence*, vol. 170, pp. 1137–1174, November 2006.

[84] G. Elidan and S. Gould, "Learning bounded treewidth Bayesian networks," *Journal of Machine Learning Research*, vol. 9, pp. 2699–2731, December 2008.

[85] S. Wahl and J. W. Sheppard, "Extracting decision trees from diagnostic Bayesian networks to guide test selection," in *Annual Conference of the Prognostics and Health Management Society*, 2010.