# Bayesian Network Structure Learning

Md. Faizul Bari
David R. Cheriton School of Computer Science
University of Waterloo
Email: mfbari@uwaterloo.ca

***Abstract*—Learning the structure of Bayesian network is useful for a variety of tasks, ranging from density estimation to scientific discovery. Unfortunately, learning the structure from data considering all possible structures exhaustively is an NP-hard problem. Hence, structure learning require either sub-optimal heuristic search algorithms or algorithms that are optimal under certain assumptions. In light of these requirements, this paper distills a set of criteria for comparison of structure learning algorithms: time and space complexity, completeness of search space, search optimality, structural correctness and classification accuracy. Based on these criteria, a representative set of existing algorithms are summarized and compared.**

## I. Introduction

A Bayesian network (BN) is a directed graph that represents the joint probability distribution among a large number of variables and allows for performing probabilistic inference with these variables. It has been applied to a wide range of tasks such as natural spoken dialog systems, vision recognition, expert systems, medical diagnosis, and genetic regulatory network inference to name a few. A BN consists of two important components: a directed acyclic graph (DAG) representing the dependency structure among the variables in the network and a conditional probability table (CPT) for each variable in the network given its parent set. Learning the structure of these networks from data is one of the most challenging problems. For complete data, best exact known methods take exponential time on the number of variables and are applicable to small settings (around 20 to 30 variables). Approximate procedures can handle larger networks, but usually they get stuck in local maxima.

The importance of correctly reconstructing the network structure depends on our learning goal. One is for *knowledge discovery*: by examining the dependencies in the learned network, we can learn both the dependency structure relating variables in our domain. A Bayesian network structure reveals much finer structure than statistical independence tests. For instance, it can potentially distinguish between direct and indirect dependencies. The second and more common reason is to perform *density estimation*: that is to estimate a statistical model of the underlying distribution. This model can then be used reasoning about instances that were not in the training data. In other words, we want our network model to generalize to new instances.

Constructing a network structure for a given application by domain experts is a time-consuming task. It is extremely difficult to build a network of moderate size by hand. In some domains, the amount of knowledge required is just too large or the expert's time is too valuable. In others, there are simply no experts who have sufficient understanding of the domain. In many domains, the properties of the distribution change from one application site to another or over time, and we cannot expect an expert to sit and redesign the network every few weeks. Alternatively, one may learn the dependency structure automatically from data, via computational methods. In recent years, there has been a growing interest in learning the structures of BNs from data. Consequently, many structure-learning methods have been proposed in the literature, including methods based on conditional independence tests and methods based on a scoring metric and a search algorithm.

The rest of the paper is organized as follows: Section II briefly categorizes the main approaches for learning Bayesian network from data. Section III introduces the criteria for comparing the existing algorithms. A brief overview of the structure learning algorithms is given in Section IV and the comparison between them is presented in Section V. Section VI outlines some guidelines for selecting a learning algorithm and the current state-of-the-art in various domains. Some open problems are discussed in Section VII. Finally, Section VIII concludes the paper and provides some future directions.

## II. Primary Approaches for Structure Learning

Structure learning of Bayesian network is an active field of research and has drawn a lot of attention from various areas of science in recent years. Though there are numerous algorithms in the literature for structure learning, all of them fall under the three principle approaches discussed below:

### A. Search and Score (S&S)

One approach to structure learning known as Search and Score (S&S), which combines a strategy for searching through the space of possible structures with a scoring function measuring the fitness of each structure to the data. The structure achieving the highest score is then selected. But the space of Bayesian networks is a combinatorial space, consisting of a super–exponential number of structures — $O(n!2^{\binom{n}{2}})$, where $n$ is number of nodes in the network. For this reason, S&S algorithms are mostly heuristic and usually have no proof of correctness. These algorithms may also require node ordering, in which a parent node precedes a child node so as to narrow the search space. Search-and-score algorithms allow the incorporation of user knowledge through the use of prior probabilities over the structures and parameters. By

considering several models altogether, the S&S approach may enhance inference and account better for model uncertainty. Recently, it was shown that when applied to classification, a structure having a higher score does not necessarily provide a higher classification accuracy.

There are also exact search algorithms, which are typically based on dynamic programming. These algorithms search a restricted version of the search space (super-structures of a bounded tree width and depth), but the search is completely. Even with the restricted search space, all currently available exact algorithms have an exponential time and memory complexity. Such complexity makes them unsuitable for large networks mostly due to memory consumption.

Score base methods consider the whole structure at once; they are therefore less sensitive to individual failures and better at making compromises between the extend to which variables are dependent in the data and the "cost" of adding the edge. The disadvantage of these methods is that they pose a search problem that may not have an elegant and efficient solution.

### B. Constraint Based (CB)

In a second approach known as constraint-based (CB), which view a Bayesian network as a representation of independence and each structure edge is learned if meeting a constraint usually derived from comparing the value of a statistical or information-theory-based test of conditional independence (CI) to a threshold. Meeting such constraints enables the formation of an undirected graph, which is then further directed based on orientation rules. CB algorithms are quite intuitive: the decouple the problem of finding structure from the notion of independence, and they follow more closely the definition of Bayesian networks. Given a distribution that satisfies a set of independence, they attempt to find an I-map for this distribution.

Algorithms of the CB approach are generally asymptotically correct. They are relatively quick and have a well-defined stopping criterion. However, they depend on the threshold selected for CI testing and may be unreliable in performing CI tests using large condition sets and a limited data size. They can also be unstable in the sense that a CI test error may lead to a sequence of errors resulting in an erroneous graph.

### C. Bayesian Model Averaging (BMA)

This Approach does not attempt to learn a single structure; instead it generates an ensemble of possible structures. These methods use the Bayes theorem to compute a posterior over all the possible structures. If we are interested in the probability of some structural feature $f$ (e.g., $f(G) = 1$ if there is an edge from node $i$ to $j$ and $f(G) = 0$ otherwise), we can compute posterior mean estimate $E(f|D) = \sum_G f(G)p(G|D)$. Similarly, to predict future data, we can compute the posterior predictive distribution $p(xjD) = \sum_G p(x|G)p(G|D)$. Since the number of possible structures is super-exponential these may seem impossible. For some class of methods this can be done efficiently, and for others approximation methods are our best choice.

Additional information on the above three approaches, their advantages and disadvantages, may be found in [1]–[4]. In addition, hybrid algorithms have been suggested in which a CB algorithm is employed to create an initial ordering, to obtain a starting graph or to narrow the search space for an S&S algorithm.

## III. Description of Criteria

To provide a comprehensive and in-depth analysis of the various approaches to structure learning, we hereby define a set of evaluation criteria. These criteria are generated from issues considered in a various structure learning algorithms, and from the discussion presented earlier. This section provides a short description of each criterion and the rationale for its inclusion.

### A. Time Complexity

This measure expresses the amount of computational time required to run the algorithms. In most cases this is expressed using the number of nodes $n$, in the learned structure, number of data samples $m$ and maximum parent set size $k$.

### B. Space Complexity

Space complexity is the amount of memory required my the algorithm. Almost all algorithms for learning BNs require exponential amount of memory. This is mostly for computing different scores e.g. Mutual Information, Conditional Independence, Bayesian Dirichlet score etc.

### C. Search Space Completeness

This criteria evaluates whether the search space considered by the algorithm is complete or not.

### D. Search Optimality

An algorithm has search optimality if it can guarantee that, it will find the optimal structure if it exists in the search space (may be restricted).

### E. Structural Correctness

Structural correctness can be measured using different scores. However, some of the scores suggested in the literature are for example, BDeu score, KL divergence, SHD metric etc. If the true structure is known then SHD metric can be used. It measures the structural errors by measuring number like missing edges, miss–directed edges, extra edges etc. On the other hand if the true structure is no know then measures like BDeu and KL divergence are mostly used. Roughly speaking, these metrics measure the difference between the data generated by the learned network and the true underlying distribution.

### F. Classification Accuracy

This measures the performance of a learned bayesian network when applied to the task of classification in a supervised or unsupervised learning model. Most of the algorithms considered in this survey have provided experiments on the UCI Repository [5] databases.

## IV. Overview of Structure Learning Algorithms

In this section we give a brief overview of several existing approaches to structure learning.

### A. Search and Score (S&S) Algorithms

*1) Ancestrally Constrained Optimal Search (ACOS):* The ACOS algorithm [6] divides the super–structure into several clusters and performs an optimal search on each of them. During the search within a cluster an Ancestral Constraint (AC) is maintained at all times to avoid cycles. An optimal BN and its score on a cluster obtained by merging two clusters are computed for every cluster maintaining AC. After the repeated computation of optimal BNs and scores on merged clusters, an optimal BN and its score on a single cluster covering the entire super–structure are finally computed.

*2) Bayesian Dirichlet Scores (BDS):* The Bayesian Dirichlet score is one of the most common criteria for evaluating Bayesian networks. Its a decomposable score and can be written with respect to local nodes of the graph, $s(G) = \sum_{i=1}^{n} s_i(\Pi_i)$, with

$$s_i(\Pi_i) = \sum_{j=1}^{r_{\Pi_i}} \Big( \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + n_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})} \Big)$$

$s_i$ is the contribution of node $X_i$ with parent set $\Pi_i$ to the global score. Properties of this score have been extensively studied in [7] and the authors have provided a caching mechanism for the local scores of nodes and their parents with reduced memory consumption and these scores can be later used by the search algorithm. Central to this algorithm is a non-increasing monotonicity of the scoring function [7] with increasing parent set size that makes it possible to discard parent sets without even inspecting them. As all the previously computed scores are cashed in memory substantial computational improvement is achieved.

*3) Akaike Information Criteria with Branch and Bound (AIC-B&B):* Another well known score function is the AIC score defined as, $s(G) = max_\theta(L_D(\theta) - t)$, where $\theta$ represent all parameters of the model and $t = \sum_{i=1}^{n}(q_i(r_i - 1))$ is the number of free variables and $L_D$ is the log–likelihood function:

$$L_D(\theta) = \log \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{n_{ijk}},$$

where $n_{ijk}$ indicate how many elements of $D$ contains both $x_i^k$ and $\Pi_i^j$. As the BD score AIC is also fully decomposable and the same approach of caching as the BDS algorithm is applied in [8]. The Branch and Bound algorithm is used for decomposing the structure into smaller subgraphs and computing local structure scores. Then the overall score is measured by merging the subgraphs into a single structure.

*4) Order based on Mutual Information – Classification Rate for parent selection (OMI-CR):* Pernkopf et. al. [9] introduced a simple order–based greedy heuristic algorithm that establishes an order of $n$ random variable based on conditional mutual information between nodes and then selects eligible parents from the ordering by employing Classification Rate (i.e. Risk) testing. OMI-CR builds upon the naive Bayesian model and each iteration of the greedy algorithm, the node yielding the highest information about the root node $C$ in the naive Bayesian model conditioned on previously chosen nodes is selected. More specifically, the algorithm forms an ordered sequence of nodes $X_\prec^{1:n} = \{X_\prec^1, X_\prec^2, \ldots, X_\prec^n\}$ according to

$$X_\prec^j \leftarrow \underset{X \in X_{1:n} \setminus X_\prec^{1:j-1}}{\operatorname{argmax}} \big[ I\big(C; X | X_\prec^{1:j-1}\big) \big], \quad where\ j \in \{1, \ldots, n\}$$

Here $I$ is the Conditional Mutual Information (CMI) measure and $I(C; X_A|X_B) = -H(C, X_A, X_B) + H(X_A, X_B) + H(C, X_B) - H(X_B)$, where entropy of $X$ is defined by $H(X) = -\sum_x p(x) \log p(x)$.

In the second step, parent set for each node is selected. The maximum number of parent is bounded by $k$ and for selecting the $k$ best parents, all $O(\binom{n}{k})$ possibilities are evaluated using CR. The CR test can be defined as

$$CR(B_s|S) = \frac{1}{M} \sum_{m=1}^{M} \delta(B_s(x_{1:n}^m), c^m)$$

where the expression $\delta(B_s(x_{1:n}^m), c^m) = 1$ if the Bayesian network classifies $B_s(x_{1:n}^m)$ trained with samples in $S$ assigns the correct class label $c^m$ to the attribute values $x_{1:n}^m$, and is equal to 0 otherwise. There will be some parent less nodes after the above procedure and according to the naive Bayesian assumption $C$ is considered the only parents of these nodes.

*5) Bayesian Network Structure Simplification (BSIM):* In [10] the authors have proposes two generic algorithm to improve the structure learned by different structure learning algorithms. One of them is the BSIM algorithm, which is a score based algorithm that removes extra parents of the overly complex nodes. BSIM takes as inputs a pre-learned DAG $G$, the data set $D$ and the threshold $k$ representing the upper bound of the CPT size. It goes through each node $X_i$ in $G$, if the CPT size of $X_i$ is larger than $k$, then it calculates an array containing the scores of the node $X_i$ for each of its current parent. BSIM then deletes one parent with the lowest score at a time until the CPT size of $X_i$ is smaller than the threshold $k$. Empirical study shows that BSIM could correctly control and effectively limits the CPT size of a node without deleting correct arcs in the learned structure.

*6) Score–based Partial Order Refinement (SPORT):* The second generic algorithm proposed in [10] is the SPORT algorithm, which uses a score based heuristic to re-select parents based on the original parent set of each node. It also makes good use of a specially designed procedure to determine the

direction of arcs, preventing correct arcs from being deleted. The SPORT algorithm focuses on reducing added arcs. The improvement is more significant on the learning algorithms that learn structures with more false positive. Overall, after applying the SPORT algorithm, the BN structures are becoming less complex, more correct and closer to the probability distribution of training data set.

### B. Constraint Based (CB) Algorithms

*1) Recursive Autonomy Identification (RAI):* The Recursive Autonomy Identification (RAI) algorithm is proposed in [11], which is a CB model that learns the structure of a BN by sequential application of CI tests, edge direction and structure decomposition into autonomous sub-structures that comply with the Markov Blanket property. This sequence of operations is performed recursively for each autonomous sub-structure. In each recursive call of the algorithm, the order of the CI test is increased. By performing CI tests of low order (i.e., tests employing small conditions sets) before those of high order, the RAI algorithm performs more reliable tests first, and thereby obviates the need to perform less reliable tests later.

By directing edges while testing conditional independence, the RAI algorithm can consider parent–child relations so as to rule out nodes from condition sets and thereby to avoid unnecessary CI tests and to perform tests using smaller condition sets. CI tests using small condition sets are faster to implement and more accurate than those using large sets.

By decomposing the graph into autonomous sub-structures, further elimination of both the number of CI tests and size of condition sets is obtained. Graph decomposition also aids in subsequent iterations to direct additional edges. By recursively repeating both mechanisms for autonomies decomposed from the graph, further reduction of computational complexity, database queries and structural errors in subsequent iterations is achieved.

*2) Mutual Information based Ordering in K2 (MIO-K2):* MIO-K2 can be divided in two steps. In the first step an ordering of the $n$ variables is achieved by using Mutual Information (MI) and Conditional Independence (CI) tests. This step can be further subdivided in three steps. In the first step, MI and CI is used to obtain a Undirected Network (UDN). MI between two random variables $X$ and $Y$ is, $I(X;Y) = H(X) - H(X|Y)$, where $H(X)$ is the entropy of random variable $X$ and $H(X|Y)$ is the conditional entropy of $X$ given $Y$. Next, the graph structure is refined by d–separation rule and CI tests. All triangular loops are eliminated by means of CI tests. Finally, the edges are assigned direction again by application of CI tests. All subgraphs with four and five nodes loops are identified and exhaustive search in this subgraph are used to assign edge directions.

After the node ordering is obtained the K2 algorithm [12] is used for learning the BN from data. K2 is a greedy search algorithm that imposes a restriction on the maximum number of parents a node can have and keeps adding parents to a nodes parent set until there are no more legal parents or the addition do not improve the score. It attempts to select the network structure that maximizes the network's posterior probability given the data.

### C. Bayesian Model Averaging (BMA) Algorithms

*1) Active Learning of Bayesian Networks (ALBN):* The ALBN algorithm introduced by Murphy [13], uses experimental interventions to distinguish between Bayesian models that are Markov equivalent. All of the above mentioned algorithms are able to learn up to a Markov equivalent class. Though this is adequate for density estimation, but its not enough if the goal is causal discovery. Two BNs may be Markov equivalent e.g. $X \rightarrow Y$ and $Y \rightarrow X$, but make very different assertions about the effect on $Y$ of changing $X$.

The ALBN algorithm performs interventional experiments by clamping a subset of the variables to fixed values. An existing Bayesian scoring method can be adapted for this operation by simply refraining from updating the parameters of the nodes that are clamped. This algorithm provides an elegant way to evaluate which kind of interventions to perform in an efficient manner.

The basic idea is to compute a posterior probability distribution over graph structures given the data, $P(G|D)$, and then for each possible experimental action compute the expected utility of the action with respect to the current belief about the model. Then the current belief is updated given the outcome of the experiment and repeat. Since the number of DAGs grows super–exponentially with the number of nodes, a form of online MCMC is used to approximate the belief state. In addition, computing the expected utility of an action requires enumerating all possible observations, which takes $O(2^n)$ time. This step is approximated by using importance sampling.

*2) Dynamic Programming with MCMC (DP-MCMC):* This algorithm introduced in [14], [15] is an extension of the ALBN algorithm and it uses the Dynamic Programming procedure outlined in [16] to marginalize over orders analytically. The DP algorithm in [16] has three major problem: modular prior, posterior of modular features and difficulty to compute a predictive density. These problems are avoided in DP-MCMC algorithm by combining DP with the Metropolis Hastings (MH) algorithm. The basic idea is simply to use the DP algorithm as an informative (data driven) proposal distribution for moving through DAG space, thereby getting the best of both worlds: a fast deterministic approximation, plus unbiased samples from the correct posterior. These samples can then be used to compute the posterior mean of arbitrary features or the posterior predictive distribution. The only limitation of this method is its current limit of 22 nodes, imposed by the exponential time and space complexity of the underlying DP algorithm.

### D. A Theoretical Complexity Bound

In [17] the authors have studied the worst-case time complexity of exact Bayesian structure learning under graph-theoretic restrictions on the super-structure. In particular, they have considered bounds on the tree–width and on the

maximum–degree of super-structures. Their results can be summarized as follows:

1) Exact Bayesian structure learning is feasible in non-uniform polynomial time if the tree–width of the super-structure is bounded by an arbitrary constant.
2) Exact Bayesian structure learning is feasible in linear time if both tree–width and maximum degree of the super-structure are bounded by arbitrary constants.

By non-uniform it means that the order of the polynomial depends on the tree–width. From this result it is clear that without bounding the super–structure (i.e. restricting the search space) there is no way to escape exponential complexity.

## V. Comparison of Structure Learning Algorithms

Based on the criteria defined earlier we now compare and contrast naming approaches introduced in the previous section.

### A. Time Complexity

*1) Ancestrally Constrained Optimal Search (ACOS):* If the maximum degree of the super–structure is $d$, then the time complexity is $O(n.2^d)$. If we also restrict the maximum number of parents per node to $k$, then the time complexity becomes $O(n.d^k)$.

*2) Bayesian Dirichlet Scores (BDS):* The worst case time complexity is $O(n.2^n)$, but in many practice networks the time complexity is much smaller.

*3) Akaike Information Criteria with Branch and Bound (AIC-B&B):* The worst case time complexity is $O(n.2^n)$, but in many practice networks the time complexity is much smaller.

*4) Order based on Mutual Information – Classification Rate for parent selection (OMI-CR):* For a given ordering this algorithm requires $O(n^k)$ operations, where $n$ is the number of nodes and $k$ is the maximum three width (maximum number of parents) of the sub–graph considered over the variables.

*5) Recursive Autonomy Identification (RAI):* In the worst case it will neither decompose the structure nor direct any edge, as a result identifying the entire structure as a single autonomous system. Given the maximum number of possible parents $k$ and the number of nodes $n$, the worst case time complexity will be bounded by $O(n^k)$.

*6) Mutual Information based Ordering in K2 (MIO-K2):* Time complexity of this algorithm is $O(n^4)+O(m.n^2)$, where $m$ is the number of samples. Though the time complexity is polynomial in $n$ it has dependency on the data size.

*7) Active Learning of Bayesian Networks (ALBN):* Time complexity of the algorithm is mostly dominated by the clamping operation and if the maximum number of nodes clammed simultaneously is $c$, then the worst case complexity is $O(n^c)$.

*8) Dynamic Programming with MCMC (DP-MCMC):* The DP algorithm has exponential time complexity.

### B. Space Complexity

*1) Ancestrally Constrained Optimal Search (ACOS):* For storing information about the different clusters and maintaining a acyclic graph the worst case memory requirement can be as bad as $O(m.d^n)$, where $m$ is the number if clusters and $d$ is the maximum degree.

*2) Bayesian Dirichlet Scores (BDS):* The worst case space complexity is also $O(n.2^n)$, but in many practice networks its much smaller. Due to caching memory requirement can be bounded by $O(n.2^{n/l})$, where $l$ is called the reduction factor and it has value between 2 to 4.

*3) Akaike Information Criteria with Branch and Bound (AIC-B&B):* The worst case space complexity is also $O(n.2^n)$, but in many practical networks its much smaller. Due to caching memory requirement can be bounded by $O(n.2^{n/l})$, where $l$ is called the reduction factor and it has value between 2 to 4.

*4) Order based on Mutual Information – Classification Rate for parent selection (OMI-CR):* For performing CMI and CR testing an exponential amount of memory is required to store information about all parent child relations.

*5) Recursive Autonomy Identification (RAI):* The BDeu scoring function (part of the BNT toolbox) is used for selecting a threshold for the CI tests and this causes the required memory space to grow exponentially $O(n.2^n)$ with the number of nodes $n$, even if only two parents are allowed per node.

*6) Mutual Information based Ordering in K2 (MIO-K2):* As most structure learning algorithms, this algorithm also requires exponential amount of memory.

*7) Active Learning of Bayesian Networks (ALBN):* As with most other algorithms, space complexity is $O(n.2^n)$,

*8) Dynamic Programming with MCMC (DP-MCMC):* The DP algorithm has exponential space complexity.

### C. Search Space Completeness

*1) Ancestrally Constrained Optimal Search (ACOS):* Search space is not complete but it can learn Bayesian network considering super–structures of high degree (up to 4) and hundreds of nodes.

*2) Bayesian Dirichlet Scores (BDS):* Using the BD score metric the search space is reduced to a restricted version, so the search space is not complete.

*3) Akaike Information Criteria with Branch and Bound (AIC-B&B):* Using the AIC score metric the search space is reduced to a restricted version, so the search space is not complete.

*4) Order based on Mutual Information – Classification Rate for parent selection (OMI-CR):* This algorithm only considers one ordering of the nodes so the search space is restricted to only those structures representable by the chosen ordering.

*5) Recursive Autonomy Identification (RAI):* RAI always starts with a completely connected super–structure and as a result the search space is complete.

*6) Mutual Information based Ordering in K2 (MIO-K2):* The MI and CI based node ordering reduces the search space. But as there is no guarantee that the ordering will be optimal, the search space is not complete.

*7) Active Learning of Bayesian Networks (ALBN):* Search space is not complete, as importance sampling is used instead of exhaustively enumerating all possible actions.

*8) Dynamic Programming with MCMC (DP-MCMC):* The search space is complete and exponential as the DP algorithm considers all possible ordering sequences.

### D. Search Optimality

*1) Ancestrally Constrained Optimal Search (ACOS):* The search strategy is optimal and a detailed proof of its optimality is provided in [6].

*2) Bayesian Dirichlet Scores (BDS):* The reduction in the search space with the BD score guarantees that the optimal network according to the BD score will remain in the restricted space. So the search is optimal in the restricted search space.

*3) Akaike Information Criteria with Branch and Bound (AIC-B&B):* The reduction in the search space with the AIC score guarantees that the optimal network according to the AIC score will remain in the restricted space. The B&B search is an any–time procedure and because, if stopped, it provides the best current solution and an estimation about how far it is from the global solution.

*4) Order based on Mutual Information – Classification Rate for parent selection (OMI-CR):* As Classification Rate (CR) is not decomposable and OMI-CR only considers $k$–trees as underlying structure, so it can't guarantee that it will find the best structure in the restricted search space.

*5) Recursive Autonomy Identification (RAI):* RAI introduces a tradeoff between run–time and search optimality by application of edge direction and structural decomposition in the early stage of the algorithm. It reduces time complexity but looses search optimality guarantee, as early edge direction and structural decomposition may leave out some potential networks from the search space.

*6) Mutual Information based Ordering in K2 (MIO-K2):* The K2 algorithm performs a greedy search based on the posterior probability of the structure given the data. So the search is not guaranteed to be optimal.

*7) Active Learning of Bayesian Networks (ALBN):* Due the use of MCMC for approximating the posterior probability over the BNs, search optimality can't be claimed.

*8) Dynamic Programming with MCMC (DP-MCMC):* It applies a greedy search method based on the Metropolis Hastings algorithm with a proposal distribution that is a mixture of the standard local proposal, that adds, deletes or reverses an edge at random. So the search is not guaranteed to return the optimal result.

### E. Structural Correctness

*1) Ancestrally Constrained Optimal Search (ACOS):* The authors didn't provide any structural correctness proofs or experimental evaluation.

*2) Bayesian Dirichlet Scores (BDS):* The target of the BDS algorithm is to approximate the underlying distribution, so it can't provide any kind of guarantees about structural correctness.

TABLE I
A COMPARISON OF CLASSIFICATION ACCURACY

| Database | RAI | OMI-CR |
|---|---|---|
| australian | 85.5 (0.5) | 82.0 |
| breast | 96.5 (1.6) | 97.4 |
| chess | 93.5 | 94.9 |
| cleve | 81.4 (5.4) | 81.7 |
| corral | 100 (0) | 99.2 |
| crx | 86.4 (2.6) | 84.1 |
| flare C | 84.3 (2.5) | 82.74 |
| iris | 93.3 (2.4) | 93.3 |
| mofn 3-7-10 | 93.2 | 91.41 |
| shuttle (s) | 99.2 | 99.2 |
| vehicle | 70.2 (2.8) | 67.4 |

*3) Akaike Information Criteria with Branch and Bound (AIC-B&B):* The target of the AIC-B&B algorithm is to approximate the underlying distribution, so it can't provide any kind of guarantees about structural correctness.

*4) Order based on Mutual Information – Classification Rate for parent selection (OMI-CR):* OMI-CR can't provide any structural guarantees as the search is based on a single ordering and only $k$–trees are considered as potential structures.

*5) Recursive Autonomy Identification (RAI):* It can guarantee that if there is a dependency between two nodes in the data then there will be a path between them in the learned network. But it can't guarantee that direct and indirect dependencies will be captured correctly.

*6) Mutual Information based Ordering in K2 (MIO-K2):* Though there is no theoretical correctness guarantee but experimental results show that MIO-K2 can learn complex and large networks with very little errors.

*7) Active Learning of Bayesian Networks (ALBN):* Instead of learning a single structure, ALBN learns a posterior distribution over all possible structures given data for approximating an underlying distribution, so structural correctness is not an issue here.

*8) Dynamic Programming with MCMC (DP-MCMC):* Like ALBN, DP-MCMC also learns a posterior distribution over all possible structures given data for approximating an underlying distribution, so structural correctness is not an issue.

### F. Classification Accuracy

For testing classification accuracy the databases of the UCI Repository [5] are a very popular choice. Two of the algorithms considered in this survey have provided experiments on these databases. So we have summarized the classification accuracy and standard deviation from different papers in Table I. The comparison may not be accurate due to nonuniform testing methodology applied by different papers.

## VI. Guidelines for Selection

Depending on the final target, Bayesian structure learning algorithms tend to vary in super–structure properties, search space restrictions, search methodology, score metrics etc. From the above discussion it is clear that Bayesian structure learning is a very active field of research and an explosive amount of methodological variations exists in the literature. Though

TABLE II
COMPARISON OF THE STRUCTURE LEARNING ALGORITHMS

| Algorithm | Time Complexity | Space Complexity | Is the Search Space Complete? | Is the Search Optimal? | Is the Structural Correct? |
|---|---|---|---|---|---|
| ACOS | $O(n.2^d)$ and $O(n.d^k)$ | $O(m.d^n)$ | NO | YES | NO |
| BDS | $O(n.2^n)$ | $O(n.2^{n/l})$ | NO | YES | NO |
| AIC-B&B | $O(n.2^n)$ | $O(n.2^{n/l})$ | NO | YES | NO |
| OMI-CR | $O(n^k)$ | $O(n.2^n)$ | NO | NO | NO |
| RAI | $O(n^k)$ | $O(n.2^n)$ | YES | NO | NO |
| MIO-K2 | $O(n^4) + O(m.n^2)$ | $O(n.2^n)$ | NO | NO | NO |
| ALBN | $O(n^c)$ | $O(n.2^n)$ | NO | NO | NO |
| DP-MCMC | $O(n.2^n)$ | $O(n.2^n)$ | YES | NO | NO |

Here, $n$ = number of nodes in BN, $m$ = number of data samples, $c$ = maximum number of nodes clammed simultaneously, $d$ = maximum three width, $k$ = maximum parent set size, and $l$ = reduction factor (2 to 4).

researchers haven't yet been able to handle the exponential growth of these algorithms, there are algorithms like RAI, BDS, AIC-B&B which can perform quite good in many practical situations. If we need to chose a Bayesian structure learning algorithm then our first step should be to decide the ultimate application of the learned structure. Some of the causes why we should consider learning a Bayesian Network from sample data and the appropriate algorithm for that cause is presented below:

### A. Classification

If our target is to apply the learned structure to classification then algorithms like RAI and OMI-CR are the best choice. From Table I is seems that the RAI algorithm is slightly better than the OMI-CR algorithm in classification accuracy. RAI is also a fast algorithm and can handle large and complex network structures, which makes it even more suitable for classification task.

### B. Causal Reasoning

Causal relationship between random variable can be learned by different statistical methods from data. But a BN goes much father than statistical methods, as a BN can represent both direct and indirect dependencies with a clear and highly reperesentable graph–based model. Among the algorithms discussed here, both of the Bayesian model averaging algorithms namely ALBN and DP-MCMC are the most suitable, as they output an ensemble of learned structures along with a posterior probability over them, which expresses its confidence in a learned structure. If we are interested in the probability of some structural feature $f$ (e.g., $f(G) = 1$ if there is an edge from node $i$ to $j$ and $f(G) = 0$ otherwise), we can compute posterior mean estimate $E(f|D) = \sum_G f(G)p(G|D)$. Similarly, to predict future data, we can compute the posterior predictive distribution $p(x|D) = \sum_G p(x|G)p(G|D)$.

The ACOS and MIO-K2 algorithms also gives emphasis on the causal relations between nodes by application of AC with CI and MI tests respectively. The structure learned by this algorithms can be also used for causal relationship estimation. Among the four algorithms, DP-MCMC has the exponential complexity but it searches a complete search space. So, if running time is of no concern then it can be used for near optimal results. If on the other hand run time is crucial (as it is often is) then MIO-K2 is our best choice. As this algorithm has a polynomial time complexity and produces quite comparable results.

### C. Concerned with Memory

All of the presented algorithms have exponential memory requirements except for BDS and AIC-B&B. These algorithms reduce the memory requirement by applying caching from exponential $O(2^n)$ to $O(2^{n/l})$, where $l$ is between 2 to 4. Many algorithms are restricted to learn networks with 20 to 30 nodes mostly due to memory requirements. The technique applied by BDS and AIC-B&B can be adapted by other algorithms for overcoming this limitation. Among these two algorithms the AIC score penalizes complex structures more than the BD score, so if complex structures are preferred, then AIC-B&B is suitable otherwise BDS is our choice.

### D. Going Further

The other two algorithms namely BSIM and SPORT are generic algorithms that improves upon a BN that has been learned by another BN structure learning algorithm. Application of this algorithms can be considered where accuracy in terms of density estimation is of utmost importance.

A tabular representation of the comparison of the structure learning algorithms based on our chosen criteria is presented in Table II.

## VII. SOME OPEN PROBLEMS

Some open problems that were identified during surveying papers on Bayesian structure learning, are discussed follows:

### A. Learning with the Presence of Corrupted Data

All of the algorithms presented here assume a perfect data, which is quite unrealistic in practice. How the conditional tests and scoring metrics can be adapted to handle corrupted data can be an interesting research topic.

### B. Learning with Hidden Variables

Another research topic can be the case when some of the attribute values are never observed.

### C. Learning to Introduce Hidden Variables

If can be solved this problem will have a huge effect on the machine learning community. The problem here is to identify the presence of hidden variables from data and introducing them in the correct place.

### D. Discriminative vs Generative

Typically discriminative models are applied to classification tasks, whereas generative models are appropriate (or thought to be) from density estimation. In recent years there have been a couple of approaches with promising performance that discriminately learns a generative model or vice versa. But to what extend these hybrid models will impact structure learning tasks is yet to be discovered.

### E. Online Learning

All of the structure learning algorithms are suitable for off–line learning model, but what about an online structure learning approach! There is yet to be such an approach in the literature.

### F. Continuous Variables

All of the structure learning algorithms assume discrete random variables. Can they be extended to handle continuous random variables?

### G. Score Caching

Can the caching technique of BDS and AIC-B&B algorithms be extended to other algorithms like the RAI or OMI-CR? This will improve the scalability of these algorithms.

## VIII. Discussion and Future Directions

This paper has investigated the characteristics and challenges in designing a BN structure learning algorithm. As a guideline for analysis, and to ease the readers through various aspects of a BN learning algorithm, we first presented an overview of some state-of-the-art algorithms along with a set of evaluation criteria. Next we provided a detailed comparison of the algorithms based on our selected criteria.

There are basically three major approaches to structure learning. First, the search and score methods attempt to find a ordering of nodes that maintains parent–child relationship and then use this ordering to find a high scoring network. By restricting the search space to super–structures of bounded tree–width and limited number of parents per node these methods attempts to narrow the search space. For most of the approaches the search is optimal though the search space is not complete. The second type of methods known as the constraint based approaches also consider super–structures of bounded tree–width, but instead of scoring the whole network together, they test each edge one by one. Structures learned in this way are more accurate but most of the time these algorithms have an exponential running time. Structural decomposition is used for overcoming this problem. Finally, Bayesian model averaging methods do not try to learn a single structure but instead learns a posterior probability distribution over the possible structures, which tells us about the confidence of the learning algorithm over a particular structure.

Recent state-of-the-art algorithms belonging to these primary approaches have been summarized and contrasted in this paper. The algorithms have been described very briefly due to space limitations, but this survey can be the starting point of a extensive survey over the BN structure learning algorithms. For future research direction, one obvious choice is to tackle the open problems mentioned in the previous section. Then, there are also other issues like: inclusion of prior domain knowledge in structure learning, optimal graph partitioning based on conditional tests, formulating better decomposable scores etc. to name a few.

## References


[1] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search, 2nd Edition*, 1st ed. The MIT Press, 2001, vol. 1. [Online]. Available: http://econpapers.repec.org/RePEc:mtp:titles:0262194406

[2] J. Pearl, *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.

[3] D. Heckerman, C. Meek, and G. Cooper, "A bayesian approach to causal discovery," in *Innovations in Machine Learning*, ser. Studies in Fuzziness and Soft Computing, D. Holmes and L. Jain, Eds. Springer Berlin / Heidelberg, 2006, vol. 194, pp. 1–28, 10.1007/3-540-33486-6_1. [Online]. Available: http://dx.doi.org/10.1007/3-540-33486-6_1

[4] D. Dash and M. Druzdzel, "A robust independence test for constraint-based learning of causal structure," in *Proceedings of the Proceedings of the Nineteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*. San Francisco, CA: Morgan Kaufmann, 2003, pp. 167–17.

[5] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[6] K. Kojima, E. Perrier, S. Imoto, and S. Miyano, "Optimal search on clustered structural constraint for learning Bayesian network structure," *The Journal of Machine Learning Research*, vol. 11, pp. 285–310, 2010.

[7] C. de Campos and Q. Ji, "Properties of Bayesian Dirichlet Scores to Learn Bayesian Network Structures," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[8] C. De Campos, Z. Zeng, and Q. Ji, "Structure learning of Bayesian networks using constraints," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 113–120.

[9] F. Pernkopf and J. Bilmes, "Efficient heuristics for discriminative structure learning of Bayesian network classifiers," *Journal of Machine Learning Research*, vol. 11, pp. 2323–2360, 2010.

[10] Y. Tang, K. Cooper, J. Cangussu, K. Tian, and Y. Wu, "Towards effective improvement of the Bayesian Belief Network Structure learning," in *Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on*. IEEE, 2010, p. 174.

[11] R. Yehezkel and B. Lerner, "Bayesian network structure learning by recursive autonomy identification," *The Journal of Machine Learning Research*, vol. 10, pp. 1527–1570, 2009.

[12] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Mach. Learn.*, vol. 9, pp. 309–347, October 1992. [Online]. Available: http://portal.acm.org/citation.cfm?id=145254.145259

[13] K. P. Murphy. (2001) Active Learning of Causal Bayes Net Structure. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.8206

[14] D. Eaton and K. Murphy, "Bayesian structure learning using dynamic programming and MCMC," in *Proceedings of the TwentyThird Conference on Uncertainty in Artificial Intelligence*, 2007, pp. 1–8.

[15] ——, "Exact bayesian structure learning from uncertain interventions," in *In Proceedings of AI & Statistics*. Press, 2007.

[16] M. Koivisto, "Advances in exact bayesian structure discovery in bayesian networks," in *UAI*. AUAI Press, 2006.

[17] S. Ordyniak and S. Szeider, "Algorithms and Complexity Results for Exact Bayesian Structure Learning," in *The 26th Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 8–11.