A Direct Measure for the Efficacy of Bayesian Network Structures Learned from Data*

Gary F. Holness

Quantum Leap Innovations
3 Innovation Way
Newark, DE. 19711
gfh@quantumleap.us

Abstract. Current metrics for evaluating the performance of Bayesian network structure learning includes order statistics of the data likelihood of learned structures, the average data likelihood, and average convergence time. In this work, we define a new metric that directly measures a structure learning algorithm's ability to correctly model causal associations among variables in a data set. By treating membership in a Markov Blanket as a retrieval problem, we use ROC analysis to compute a structure learning algorithm's efficacy in capturing causal associations at varying strengths. Because our metric moves beyond error rate and data-likelihood with a measurement of stability, this is a better characterization of structure learning performance. Because the structure learning problem is NP-hard, practical algorithms are either heuristic or approximate. For this reason, an understanding of a structure learning algorithm's stability and boundary value conditions is necessary. We contribute to state of the art in the data-mining community with a new tool for understanding the behavior of structure learning techniques.

1 Introduction

Bayesian networks are graphical models that compactly define a joint probability over domain variables using information about conditional independencies between variables. Key to the validity of a Bayesian Network is the Markov Condition [11]. That is, a network that is faithful to a given distribution properly encodes its independence axioms. Inducing Bayesian networks from data requires a scoring function and search over the space of network structures [10]. As a consequence of the Markov Condition, structure learning means identifying a network that leaves behind few unmodeled influences among variables in the modeled joint distribution.

* We acknowledge and thank the funding agent. This work was funded by the Office of Naval Research (ONR) Contract number N00014-05-C-0541. The opinions expressed in this document are those of the authors and do not necessarily reflect the opinion of the Office of Naval Research or the government of the United States of America.

P. Perner (Ed.): MLDM 2007, LNAI 4571, pp. 601–615, 2007.

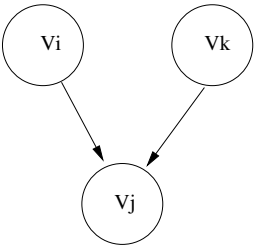


Fig. 1. Graphical representation for Bayesian Network

A Bayesian network takes form as a directed acyclic graph $G = (V, E)$ where the nodes $V_i \in V$ represent the variables in a data-set and the directed edges $(V_i, V_j) \in E$ encode the causal relationship between V_i and V_j. Dependencies among variables are modeled by a directed edge. As such, if edge $(V_i, V_j) \in E$, then V_j depends causally upon V_i or, similarly, V_i is the parent of V_j and V_j is the child of V_i. The graphical model for a simple Bayesian network appears in Figure 1. In this example, we have three nodes V_i, V_j, and V_k where V_j is causally dependent upon V_i and V_k. Causality is implied in edge directedness. Through its network structure, a Bayesian network model encodes the independence axioms of a joint distribution. Given a Bayesian network $G = (V, E)$ we compute the full joint distribution for variables $V_i \in V$ using the chain rule:

$$p(V_1, \ldots, V_n) = \prod_{i=1}^{n} p(V_i | pa_i)$$

where pa_i are the set of variables that are the parents of V_i. By expressing the joint distribution in terms of its conditionally independent factors, marginalization and inference are made more tractable.

Inference in Bayesian networks is well known to be an NP-hard problem both in the exact and approximate cases [2, 4]. Construction of Bayesian networks structures from data is also an NP-hard problem. The major classes of techniques for learning Baysian networks falls into two major categories. The first considers network construction as a constraint satisfaction problem [11, 14]. These methods compute independence statistics such as χ^2 test, KL-divergence, or entropy over variables and build networks that represent computed associations. The second considers network construction as an optimization problem. These methods search among candidate network structures for the optimum [3, 5, 15].

The search problem over Bayesian network structures is also an NP-hard problem. Heuristic approaches such as the $K2$ algorithm impose simplifying assumptions on the network in order to make learning and inference tractable [3]. In $K2$, nodes are are assumed to have a causal ordering. That is, a node appears later in an ordering than the nodes on which it depends. Additionally, the $K2$ algorithm also bounds the number of parent dependencies a node may have. In the recent $K2GA$ approach, the author employs a genetic algorithm to perform stochastic search simultaneously over the space of node orderings and network structures

for an extension of the $K2$ algorithm. $K2GA$ has been found to perform competitively with respect to ground truth networks on benchmark data-sets [7]. Additional search techniques include greedy hill-climbing, simulated annealing and Markov Chain Monte Carlo (MCMC) [6, 13]. Other approaches make the problem more tractable by pruning the search space. For example, the sparse candidate algorithm uses mutual information between variables to prune the search space so that only a reduced set of potential parents are considered for each variable [9]. Another approach that has enjoyed success performs greedy search over equivalence classes of DAG patterns instead of the full DAG space representation [1].

The experiments described in this paper grew out of the need to characterize the performance of an implementation of $K2GA$. This work goes beyond measures of model fit and convergence time as typical in the Bayesian network literature to include measurements of stability. While we use $K2GA$ as the target system for evaluation, our techniques are generally applicable to any Bayesian network structure learning algorithm. In recent related work Shaughnessy and Livingston introduce a method for evaluating the causal explanatory value of structure learning algorithms [12]. Their approach begins with randomly generated ground truth networks involving three-valued discrete variables. Next they sample from them to produce small synthetic data-sets that are input to a structure learning algorithm. Finally, precision-recall measures are made from edge level statistics, such as *false positive edge count*, comparing the learned and ground truth networks. While this method evaluates different types of causal dependencies it cannot vary the strength of such dependencies and requires a sufficient number of samples. Because $K2GA$ is a stochastic algorithm, we set out to test if initial conditions and noise in the data affect the structure learner's ability to correctly capture variable dependencies.

In the sections that follow, we begin with a high level description of the stochastic algorithm K2GA. Then, we outline a method for testing how well the Bayesian network has modeled dependencies among variables. In doing so, we treat variable dependence as a retrieval problem and apply an ROC technique for measuring performance stability. Lastly, we describe our experiments and discuss results.

2 Structure Learning Using K2GA

$K2GA$ makes use of an alternate Bayesian network representation that encodes a DAG in terms of its undirected skeleton and the causal ordering of the nodes. Let $X = \{X_1, \ldots, X_N\}$ be a set of variables, $\Theta = \{\Theta_1, \ldots, \Theta_N\}$ be the ordering of nodes (where $\Theta_i \in [0, 1]$), and $\mathcal{B}$ be the adjacency matrix for the undirected skeleton such that $\mathcal{B}_{ij} = 1$ if and only if X_i is related to X_j. Skeleton, $\mathcal{B}$, describes the dependency between two variables while Θ defines the edge directedness. For example, in the situation where X_j is causally dependent on X_i, we have $\mathcal{B}_{ij} = 1$ and $\Theta_i < \Theta_j$.

Given the exponential space of DAGS, a number of simplifying assumptions have been made to reduce the complexity of the search space to polynomial in the number of nodes. These include causal ordering of variables that participate in the model along with bounded in-degree between a node an its parents. The topological ordering, $\prec$, of graph nodes $\{X_1, \ldots, X_N\}$ is such that

$$\bigvee_{i,j} X_j \prec X_i \rightarrow X_j \in Ancestors(X_i)$$

Structure learning algorithms that assume the $K2$ heuristic search within a family of DAGS possible from fixed causal orderings. Given topological ordering, $\prec$, the set of all possible skeletons $\mathcal{S} = \{\mathcal{B}_1, \ldots, \mathcal{B}_L\}$ is defined by the number of unique skeletons that can be defined from the upper triangle of $\mathcal{B}$. Given N-variables, $|S| = 2^{\frac{N(N-1)}{2}}$. Since there are $N!$ orderings, this results in substantial reduction from a total of $N! \left(2^{\frac{N(N-1)}{2}}\right)$ possible DAG patterns. While a factorial reduction in search space is significant, the issue of which ordering to search remains. The $K2GA$ algorithm performs simultaneous search of the space of topological orderings and connectivity matrices. For more detailed descriptions of $K2GA$, We direct the reader to the original work [7].

3 Markov Blanket Retrieval: An Efficacy Measure

By extending the definition of a document in information retrieval, verification of a Bayesian network is treated as a retrieval problem where the information need is the set of causal dependencies for a given variable. This corresponds to the Markov blanket that most closely resembles the ground truth blanket for a given node. In using a vector space approach and ranking, we allow for partial similarity. This is particularly important for variables with weak dependence relationship.

This approach differs from traditional methods for verification of Bayesian networks in that we do not rely on samples from a hand constructed *gold standard network* for verification. Because such techniques rely on samples from the specified network, a sufficient sample size is required. Moreover, for nontrivial real-world problems, apriori knowledge of variable dependencies is difficult. Consider a real world complex data-set such as manufacturing or supply chain modeling scenario involving 100's or $1,000$'s of variables. It might be the case that the Bayesian network structure learned from data is correct, but its performance is discounted by a faulty hand constructed gold standard network. By exercising precise control over variable dependence and measuring resulting performance, we provide characterization of a Bayesian network learner's modeling stability using ROC analysis.

3.1 ROC Curves

More than just its raw performance numbers, an algorithm's quality is also measured in terms of sensitivity and specificity. A predictor's sensitivity measures

the proportion of the cases picked out from a data set relative to the total number of cases that satisfy some test. Sensitivity is also called the true positive rate. A predictor's specificity measures its ability to pick out cases that do not satisfy some test. Specificity is also called the true negative rate. A receiver operating characteristic (ROC) curve is related to likelihood ratio tests in statistics and expresses how the relationship between sensitivity and specificity changes with system parameters [8].

In comparing Bayesian networks, we would like a single measure of predictive quality. Area under the curve (AUC) is a non-parametric approach for measuring predictive quality. AUC is simply the area under the ROC curve. This gives us a standard means of comparing performance. AUC varies in the closed interval $(0, 1)$ on the real number line and is interpreted as the rate of correct prediction.

As one could imagine, a good predictor is one that can correctly identify cases in the data that actually have the phenomenon under test. This corresponds to an AUC that is closer to 100%. AUC results are typically compared to the random performance. In an example where true positive and false negative are assumed equally likely, the ROC curve is a straight line with slope 45-degrees and AUC of 50%. Any method that cannot outperform random performance is not worth deployment.

For Bayesian network structures, in order to convert performance measures into likelihood ratio tests for the purpose of ROC analysis, we must compare structures learned from data with some notion of ground truth. This allows us to define what it means to have a true positive or a true negative.

3.2 Markov Blanket

A Markov blanket of a node, A, is defined as A's parents, children and spouses (the parents of A's children). The Markov blanket is the minimal set of nodes that give A conditional independence.

$$P(A|MB(A), B) = P(A|MB(A))$$

That is, A is conditionally independent of any node $B \notin MB(A)$ given $MB(A)$. The Markov blanket gives complete description of the variables upon which A depends. As depicted in Figure 2, these are the nodes that partition A from the rest of the nodes in the network.

The Markov blanket is related to d-separation in that given the set $Z = \{Z_i \in MB(A)\}$ and $C = \{X - A - MB(A)\}$ where X is the set of variables, it is the case that A is d-separated from C given $MB(A)$. Thus, the Markov blanket gives us the dependence relationship between a node and all other nodes in the Bayesian network. We use this to test $K2GA$'s efficacy in correctly modeling causal dependencies.

3.3 Ground Truth Causal Dependence

Controlling variable dependencies is accomplished by augmenting a data-set with synthetic variables. We treat synthetic variables X_{new} as queries. Because

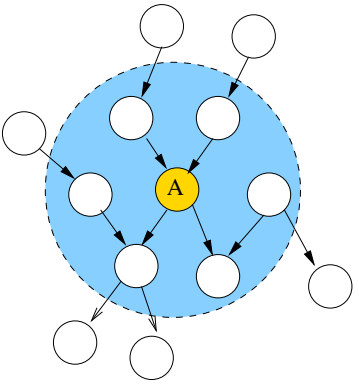


Fig. 2. A node and its Markov blanket

variables $X_1, \ldots, X_k$ are used to compute the synthetic variable, we know ground truth that $X_1, \ldots, X_k$ are in X_{new}'s Markov blanket. A synthetic variable also depends on a noise process ϵ used to control the strength of dependence on the ground truth parent variables. For strong dependence, the contribution to X_{new} by the noise process is dominated by $X_1, \ldots, X_k$. For weak dependence, the contribution to X_{new} by the ground truth parent variables is dominated by the noise process. Using random variable $A \sim Bernoulli(\alpha)$ taking on values $a \in \{0, 1\}$ we select

$$X_{new} = \begin{cases} f_{\boldsymbol{w}}(X_1, \ldots, X_k) & \text{if } a = 1 \\ \epsilon & o.w. \end{cases}$$

where parameter α is defined in the closed interval $(0, 1)$ on the real number line. Thus, α regulates the strength of causal dependence. Regardless of the strength of causal dependence, we know ground truth that $\{X_1, \ldots, X_k\} \subset MB_{\mathcal{B}}(X_{new})$.

The amount by which the synthetic variable depends on each of its ground truth parents is determined by a vector of weights. Given k ground truth parent variables, we have weight vector $\boldsymbol{w} =< w_1, \ldots, w_k >$ computed by uniform sampling from the unit simplex in k-dimensions. That is the series of weights from the set $\{< w_1, \ldots, w_k > | w_1 + \ldots + w_k = 1, 0 <= w_i <= 1, i = 1, \ldots, k\}$. Given Q-samples, this gives us representative coverage across the range of associations a dependent variable can have on k-parent variables. The dependent variable takes on values drawn from the union of the domains of its parents. In Figure 3 we list the values of the domain for three parents in rectangles along the top row and domain values of the dependent variable in rectangles along the bottom row. In this example, we have three parent variables whose domain sets have values $\{v_1, v_2\}$, $\{v_3, v_4, v_5\}$, and $\{v_6, v_7, v_8, v_9\}$ respectively. The dependent variable draws its values from the set $\{v_1, \ldots, v_9\}$ (Figure 3). This allows us to interpret the weight vector as the relative proportion of cases for which the value of the dependent variable is dictated by a given parent. An example of this appears in Figure 4. We list values for four cases by repeating the pair of rows

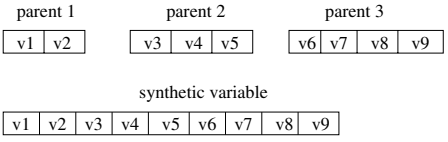


Fig. 3. The domain of a synthetic variable

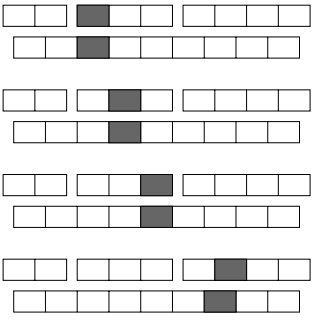


Fig. 4. Illustration of synthetic variable causally dependent on parents

from Figure 3 once for each case. The values taken by the three parents and the dependent variable are illustrated by shading in the appropriate positions in each row.

As can be seen in Figure 4, for the synthetic dependent variable, the first, second, and third cases are causally dependent on the second parent while the fourth is causally dependent on the third parent. These four example cases would correspond to a weight vector of $< 0.0, 0.75, 0.25 >$ with $\alpha = 1.0$. For $\alpha < 1$, we incorporate a noise process ϵ by selecting the dependent variable's value from its domain by sampling uniform at random for $(1-\alpha)$ percent of the cases. We treat X_{new}'s Markov blanket computed from Bayesian network $\mathcal{B}$ as a document. The causal dependency set for each of the X_i is also treated as a document. This results in a collection of documents, one for X_{new} and each X_i.

3.4 The Retrieval Problem

A Markov blanket describes the complete set of dependencies for a given variable. By definition, the Markov blanket is a subset of the variables over the modeling domain. Let each variable, $X_1, \ldots, X_d$ (including X_{new}) in a data-set be an indexing term. A Markov blanket then becomes a simple document containing a subset of indexing terms. Define weight w_{ij} as the number of occurrences of term-i in document j. For a Markov blanket, because a variable occurs at most once, we have that $w_{ij} \in \{0, 1\}$. Given d-variables, the Markov blanket $MB(X_j)$ for variable X_j is compactly described by weight vector

$$MB(X_j) =< w_{1j}, \ldots, w_{dj} >$$

A ranking function $R(q_i, d_j)$ outputs a value along the real number line that defines an ordering of documents in terms of their relevance to the given information need. Define ranking function R:

$$R(d_j, q_i) = \frac{\sum_{k=1}^{d} w_{kj} q_{ki}}{|X|}$$

That is the proportion of the variables in the Markov blanket that satisfy the query. With this definition, we rank the d Markov blankets in a Bayesian network and select the Markov blanket *document* associated with the highest rank. The top ranked Markov blanket corresponds to the variable dependencies that are most relevant to the query. We then measure quality in modeling ground truth variable dependences using ROC curves.

We expand a query for X_{new} into known ground truth causal dependencies in vector form and search for the most relevant document in the collection. In our procedure, we create X_{new} randomly. Given Bayesian network $\mathcal{B}$ learned from a data set augmented with the synthetic variable, compute documents $d_i = MB_{\mathcal{B}}(X_i)$. Define the f-blanket for X_{new}, $MB_f(X_{new}) = \{X_1, \ldots, X_k\}$. Given a query expansion, q_i, the most relevant document, d_r, is returned:

$$d_r = argmax_j R(d_j, q_i)$$

That is the document with the highest rank. This corresponds to the Markov blanket in the learned network that most closely resembles the ground truth f-blanket. In using a vector space approach and ranking, we allow for partial similarity with a given query. This is particularly important for synthetic variables that are weakly dependent on their parents. By adding a set of synthetic variables whose dependence on $X_1, \ldots, X_d$ varies in the number parent nodes and strength of dependence, we can use the true positive and false positive rate for retrieval to measure the Bayesian network's ability to accurately model true causal dependencies.

We call our approach Markov blanket retrieval (MBR). The algorithm for MBR appears in Figure 5. Input parameters to MBR are a data-set $\mathbf{X}$ dependence strength α, and parent set size k. We begin by computing the number of cases and variables in steps 1 and 2. Measurements are made for a fixed number of Q queries (step 3). Each query consists of a synthetic variable whose k ground truth parents are selected randomly (step 4). For each selected parent set, we choose their dependence strengths by sampling from the unit simplex (step 5). Before constructing the synthetic variable, we first create its domain set by taking the union of the domains of its k-parent set (step 6). Then, looping over each of the N cases (step 7) we compute the value, $x_{i,new}$ of the synthetic variable using the mixture weights and the dependence strength α (step $8, 9, 10$). This gives us a new column of data corresponding to the synthetic variable V_{new}. The augmented data-set $\mathbf{X}'$ is then constructed by including the column of values, X_{new}, for the synthetic variable among the columns $\{X_1, \ldots, X_L\}$ of the original data set (step 11). We run structure learning on the augmented data set and obtain a Bayesian network $\mathcal{B}$ (step 12). For each variable in the augmented

```
MARKOV-BLANKET-RETRIEVAL(k, α, X)
 1  N ← |X|
 2  L ← num-variables(X)
 3  for q ← 1 to Q
 4  do sample {V_1, ..., V_k} ∈ X
 5      sample < w_1, ..., w_k > from simplex
 6      domain(V_new) = ⋃_{j=1}^{k} domain(V_j)
 7      for i ← 1 to N
 8      do sample A ~ Bernoulli(α)
 9          if a = 1 then x_{i,new} ← f(x_{i,1}, ..., x_{i,k}) //using mixture weights
10          else x_{i,new} ← ε
11          X' = {X_1, ..., X_L, X_new} //augment data-set
12      B ← learn-structure(X')
13      for i ← 1 to L + 1
14      do
15          d_i = compute-document(MB_B(X_i))
16      q = compute-document(MB_{ground−truth}(X_new))
17      d_r = argmax_l R(d_l, q)
18      record ROC data
```

Fig. 5. algorithm for Markov blanket retrieval analysis of structure learner

data-set, we obtain the Markov blanked computed by the structure learner and compute a document (steps 13, 14, and 15). Given the ground truth Markov blanket for the synthetic variable, we expand it into a query (step 16). We then rank Markov blanket *documents* from step 14 and return the highest ranking document (step 17). We then record whether or not our result is a true positive, true negative, false positive or false negative and continue to the next query iteration (step 18).

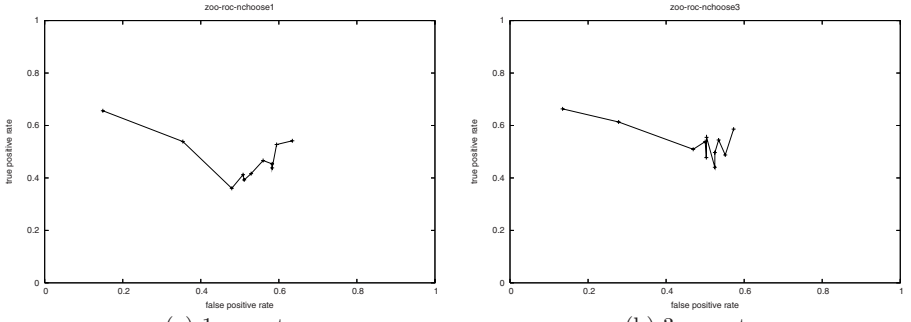
4 Experiments

Our experimental goal was to uncover how $K2GA$'s ability to model causal dependence changed as we varied the genetic algorithm's population size and number of generations across data-sets of different complexities. We ran experiments using three data-sets from the UCI machine learning repository. We selected one nominal (zoo), one mixed nominal-integer (lymphoma), and one real valued (sonar) data set for experiments in order to have representation across different types of data-sets. We rank data-sets by their complexity defined in terms of the number of variables and the number of instances (Table 1).

In our ranking, we include the class label in our variable counts. Since optimization based approaches such as $K2GA$ bound the maximum in-degree of nodes in the Bayesian network, it is important demonstrate how in-degree for causal dependence affects performance. This means measuring performance as more parents nodes are recruited. We tested variable dependence by running experiments for f-blankets of size 1,2, and 3.

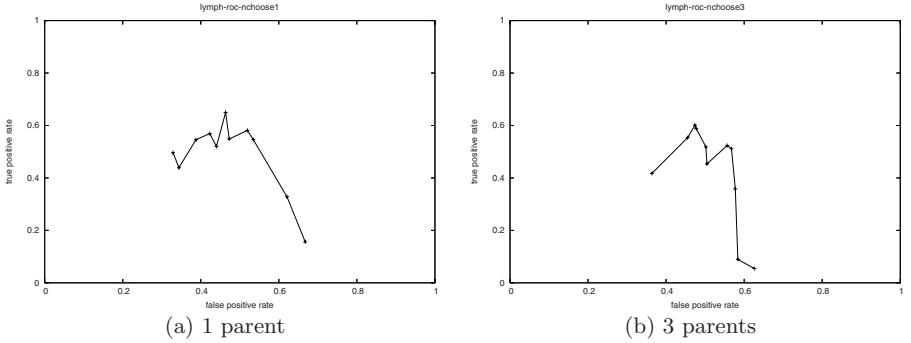
Table 1. Data set complexities

rank	data-set	number of variables	number of instances
1	zoo	17	101
2	lymphoma	19	148
3	sonar	61	208



(a) 1 parent (b) 3 parents

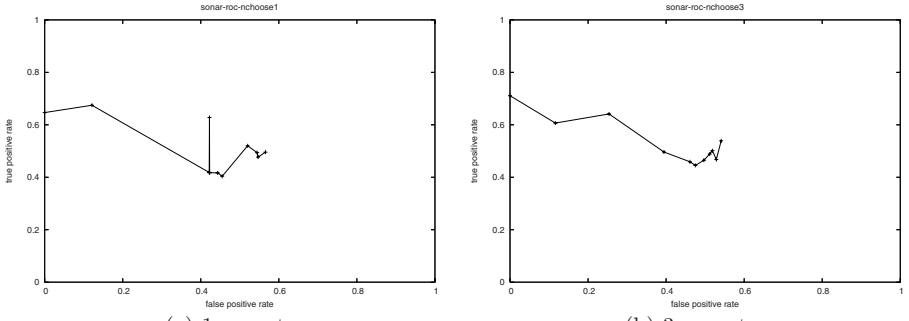
Fig. 6. ROC curve for zoo data-set with various parental causal dependencies for $K2GA$ at 50 generations and population size 10



(a) 1 parent (b) 3 parents

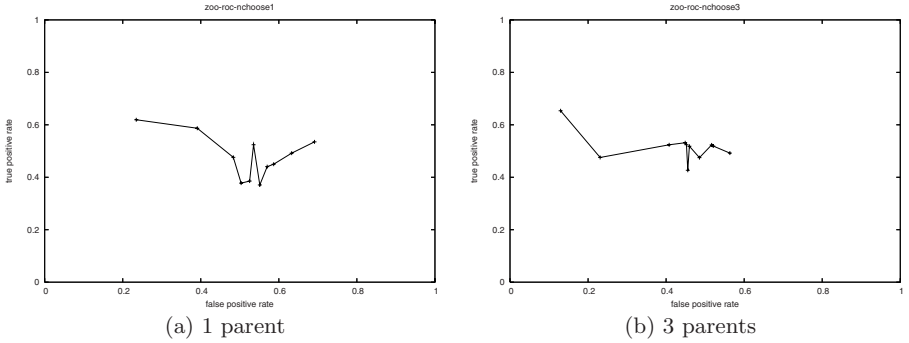
Fig. 7. ROC curve for lymphoma data-set with various parental causal dependencies for $K2GA$ at 50 generations and population size 10

A positive test instance is a synthetic variable for which a true causal dependency exists and a negative is a synthetic variable for which a dependence does not exist. We generated synthetic variables with 50% priors over positive instances. For the remaining 50%, we set thresholds for strength of causal dependence in regular increments for $\alpha = 0.0, 0.1, \ldots, 1.0$. Across all settings of α the expected generation rate for positive instances is $0.5 + \sum_{\alpha=0.0}^{1.0} 0.5\alpha = 0.75$.



(a) 1 parent (b) 3 parents

Fig. 8. ROC curve for sonar data-set with various parental causal dependencies for $K2GA$ at 50 generations and population size 10

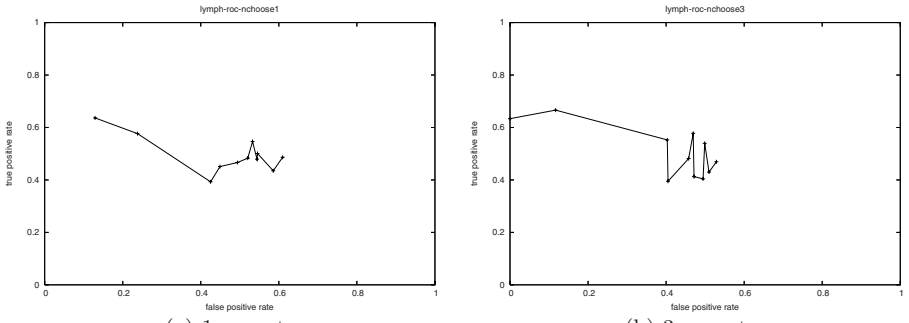


(a) 1 parent (b) 3 parents

Fig. 9. ROC curve for zoo data-set with various parental causal dependencies for $K2GA$ at 100 generations and population size 20

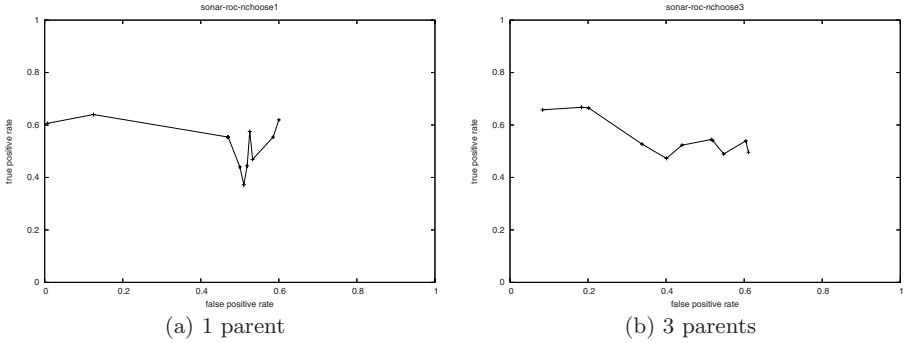
An augmented data-set has $d+1$ variables where d variables are from the original data-set, and the $d+1$-th is the synthetic variable. Since a random approach must guess uniform at random which of the $d+1$ Markov blanket *documents* matches the query, the probability of picking out the true positive is $\frac{1}{d+1}(0.5 + \sum_{\alpha=0.0}^{1.0} 0.5\alpha)$. Using the trapezoidal rule, we compute AUC for random performance as 0.5000.

K2GA performs optimization by stochastic search. K2GA is a genetic algorithm in which Bayesian network structure candidates are members of a population. Thus, the population size for $K2GA$ controls the number of frontiers along which stochastic search in the space of network structures is performed. The number of generations controls the number of optimization rounds for which search proceeds. We ran two versions of K2GA differing in population size and number of generations, one at 50 generations and population size of 10 and another at 100 generations and population size of 20. We refer to these as



(a) 1 parent (b) 3 parents

Fig. 10. ROC curve for lymphoma data-set with various parental causal dependencies for *K2GA* at 100 generations and population size 20



(a) 1 parent (b) 3 parents

Fig. 11. ROC curve for sonar data-set with various parental causal dependencies for *K2GA* at 100 generations and population size 20

K2GA-small and *K2GA-large*. We ran experiments for 100 queries for each setting of causal dependence strength using 5 fold cross validation on 10 random initializations of K2GA. Because validation is done directly on the resulting structure and not on the test set, we did not use the test set from each fold. We did this in order to train similarly to approaches that validate by partitioning data into training and testing sets.

This resulted in 5000 queries for each setting of α representing a total of $55,000$ total queries per experiment. ROC curves for f-blanket sizes 1 and 3 appear in Figure 6, 7, 8, 9, 10, 11. In each of our results, there was a dramatic decrease in the true positive rate once the false positive rate reached between 0.4 and 0.5. We compare K2GA-small with K2GA-large by their AUC scores (Table 2). We group the results of K2GA-small and K2GA-large and indicate the better performer in bold typeface.

Table 2. AUC scores for Markov-Blanket Retrieval

data-set	*K2GA* setting	f-blanket size	AUC for MBR	AUC for random
zoo	50-gen 10-pop	1	**0.5802**	0.5000
zoo	100-gen 20-pop	1	0.5482	0.5000
zoo	50-gen 10-pop	2	**0.6338**	0.5000
zoo	100-gen 20-pop	2	0.6023	0.5000
zoo	50-gen 10-pop	3	**0.6359**	0.5000
zoo	100-gen 20-pop	3	0.5927	0.5000
lymph	50-gen 10-pop	1	0.5653	0.5000
lymph	100-gen 20-pop	1	**0.5737**	0.5000
lymph	50-gen 10-pop	2	**0.6305**	0.5000
lymph	100-gen 20-pop	2	0.6174	0.5000
lymph	50-gen 10-pop	3	0.6205	0.5000
lymph	100-gen 20-pop	3	**0.6529**	0.5000
sonar	50-gen 10-pop	1	0.6356	0.5000
sonar	100-gen 20-pop	1	**0.6709**	0.5000
sonar	50-gen 10-pop	2	0.6280	0.5000
sonar	100-gen 20-pop	2	**0.6640**	0.5000
sonar	50-gen 10-pop	3	**0.6652**	0.5000
sonar	100-gen 20-pop	3	0.6186	0.5000

We found that if the data-set contained a smaller number of variables as is the case with the zoo data-set (complexity rank 1), as we increase the number of parents upon which a variable can causally depend, K2GA-small consistently had higher AUC. A Bayesian network with larger node in-degree is a more complex model. Building more complex models require a larger number of training examples. The zoo data-set contains relatively few instances. Since K2GA-large searches twice as many frontiers for twice as many optimization rounds, it tends to over-fit the data. Therefore, its performance is worse on our simplest data-set as the f-blanket size increases. On the lymphoma data-set, we see a modest increase in the number of variables and 40% increase in number of instances. K2GA-large turns in its largest favorable difference in performance over K2GA-small when the f-blanket is 3. This coincides with K2GA-large's ability to search more complex models.

For the sonar data-set (rank 3) , we find K2GA-large turns in a higher AUC for f-blanket sizes 1 and 2. The sonar data-set has $3x$ more variables. By searching twice as many frontiers for twice as many optimization rounds, K2GA-large is more able to consistently and stably (higher AUC) model causal linkages in complex data. When the f-blanket increases to 3, the number of instances becomes insufficient. Consider 2 Markov blankets each containing a child node with 3 parent nodes. Building network involves evaluating conditional probability tables. If each variable assumes only 2 states, we find the conditional probability table (CPT) for the child node has 2^4 entries. Across 2 Markov blankets, we have $(2^4)^2 = 256$ unique configurations. Estimating the CPTs for this example

requires more than 208 instances. Because K2GA-small involves fewer frontiers and optimization rounds, it effectively builds lower complexity models. This gives us advantage when there are too few examples because it helps against over-fitting by early stopping. As we can see K2GA-small turns in a higher AUC than K2GA-large when the f-blanket is 3.

5 Discussion

We have presented a new tool for measuring the efficacy of structure learning algorithms in finding causal dependencies that exist within data. By treating membership in a Markov blanket as a retrieval problem and controlling for ground truth causal dependencies, we are able to borrow sound principles of ROC analysis to evaluate the structure learner's performance. Our measurements go beyond error by measuring stability across a range of dependence strengths using AUC. Our method measures structure learning efficacy directly from the learned structures themselves without use of a gold standard network. We have found from our experiments that Markov Blanket Retrieval (MBR) lends insight into parameter tuning and stability of a structure learning algorithm and feel it is a valuable tool for the data-mining community.

The goal for reported experiments was the development of a tool for comparing the performance of different parameterizations of a structure learner under varying dependence strengths. In complex real-world data-sets, some of the variables are correlated. Future investigation will include measurements for the effect of correlation between parent variables on modeling efficacy. We are encouraged by results for our measure on K2GA. A logical next step is to investigate MBR's utility in making fair comparison between different structure learning techniques. We represented Markov blankets using vector space and ranked documents based on a normalized inner product. This approach allowed us to observe the proportion of variables in the augmented data-set that matched the ground truth f-blanket. In future experiments we will extend our ranking approach to include measurement of graph properties as well as other distance measures.

References

[1] Chickering, D.: Learning equivalence classes of bayesian-network structures. Journal of Machine Learning Research 2, 445–498 (2002)
[2] Cooper, G.: Probabilistic inference using belief networks is np-hard. Artificial Intelligence 42(1), 393–405 (1990)
[3] Cooper, G., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Mahcine Learning 9(4), 309–347 (1992)
[4] Dagum, P., Luby, M.: Approximating probabilistic inference in bayesian belief networks is np-hard. Artificial Intelligence 60(1), 141–153 (1993)
[5] Heckerman, D., Geiger, D., Chickering, D.: Learning bayesian networsk: The combination of knowledge and statistical data. Machine Learning 20, 197–243 (1995)
[6] Eaton, D., Murphy, K.: Bayesian structure learning using dynamic programming and mcmc. In: NIPS Workshop on Causality and Feature Selection (2006)

[7] Faulkner, E.: K2ga: Heuristically guided evolution of bayesian network structures from data. In: IEEE Symposium on Computational Intelligence and Data Mining, 3 Innovation Way, Newark, DE. 19702, April 2007, IEEE Computer Society Press, Los Alamitos (2007)
[8] Fawcett, T.: Roc graphs: Notes and practical considerations for data mining researchers. Technical Report HPL-2003-04, Hewlett Packard Research Labs (2003)
[9] Friedman, N., Nachman, I., Peér, D.: Learning bayesian network structure from massive datasets: The sparse candidate algorithm. In: Proceedings of UAI, pp. 206–215 (1999)
[10] Heckerman, D.: A tutorial on learning with bayesian networks (1995)
[11] Pearl, J., Verma, T.S.: A theory of inferred causation. In: Allen, J.F., Fikes, R., Sandewall, E. (eds.) KR'91: Principles of Knowledge Representation and Reasoning, San Mateo, California, pp. 441–452. Morgan Kaufmann, San Francisco (1991)
[12] Shaughnessy, P., Livingston, G.: Evaluating the causal explanatory value of bayesian network structure learning algorithms. In: AAAI Workshop on Evaluation Methods for Machine Learning (2006)
[13] Singh, M., Valtorta, M.: Construction of bayesian network structures from data: A brief survey and an efficient algorithm. International Journal of Approximate Reasoning 12(2), 111–131 (1995)
[14] Spirtes, P., Glymour, C., Scheines, R.: Causation, prediction, and search. Springer, Heidelberg (1993)
[15] Lam, W., Bacchus, F.: Learning bayesian belief networks: An approach based on the mdl principle. Comp. Int. 10, 269–293 (1994)