Scaling-Up Bayesian Network Learning to Thousands of Variables Using Local Learning Techniques

Ioannis Tsamardinos, Ph.D., Constantin F. Aliferis, M.D., Ph.D., Alexander Statnikov, M.S., Laura E. Brown, M.S.

Department of Biomedical Informatics, Vanderbilt University, Nashville, TN

ABSTRACT

State-of-the-art Bayesian Network learning algorithms do not scale to more than a few hundred variables; thus, they fall far short from addressing the challenges posed by the large datasets in biomedical informatics (e.g., gene expression, proteomics, or text-categorization data). In this paper, we present a BN learning algorithm, called the Max-Min Bayesian Network learning (MMBN) algorithm that can induce networks with tens of thousands of variables, or alternatively, can selectively reconstruct regions of interest if time does not permit full reconstruction. MMBN is based on a local algorithm that returns targeted areas of the network and on putting these pieces together. On a small dataset MMBN outperforms other state-of-the-art methods. Subsequently, its scalability is demonstrated by fully reconstructing from data a Bayesian Network with 10,000 variables using ordinary PC hardware. The novel algorithm pushes the envelope of Bayesian Network learning (an NP-complete problem) by about two orders of magnitude.

1. Introduction

Bayesian Networks (BN) is a formalization that has proved itself a useful and important tool in medicine for building decision support systems [2] and in bioinformatics for discovering gene expression pathways [4]. Automatically learning BNs from observational data has been an area of intense research for more than a decade yielding practical algorithms and tools. The BN representation and learning algorithms naturally lend themselves causal modeling and causal discovery [5].

Despite the great advances in BN learning techniques, they have not yet proved themselves up to the challenge posed in a number of current domains, such as gene expression, proteomics, and text categorization. Current techniques only scale up to a few hundred variables in the best case. For example, the public available versions of the **PC** [10] and the **TPDA** (also called PowerConstructor) [6] algorithms accept datasets with only 100 and 255 variables respectively, indicating the expectations of the inventors regarding their scalability.

In this paper, we present a BN learning algorithm called Max-Min Bayesian Network (**MMBN**) that is able to scale up to tens of thousands of variables, thus pushing the envelope of BN learning by about two orders of magnitude. When compared with a variety of state-of-the-art methods in the field on a small dataset, it exhibits improved quality of output. In addition, the performance of the algorithm is still excellent in reconstructing from data a BN with 10,000 variables using 1000 training instances.

MMBN has an additional important advantage over the previous methods: it is an anytime algorithm in the sense that one can stop it at any time and recover only an area of interest around a target variable *T* instead of the full BN. The longer the algorithm is allowed to run, the larger the reconstructed area around *T* will be. This property allows the algorithm to learn at least parts of extremely large networks; this is empirically demonstrated with a proof-of-concept experiment.

2. The Max-Min Bayesian Network (MMBN) Algorithm

Bayesian Networks (BN) [10] are mathematical objects that compactly represent a joint probability distribution *J* among a set of random variables Φ (also called nodes) using a directed acyclic graph *G*.

As a first step, **MMBN** discovers the edges of the BN, and in the second step orients them. However, for the rest of the paper we only focus on the first step of edge discovery. The orientation phase of the **PC** algorithm or a constrained hill-climbing search-and-score in the fashion of the Sparse Candidate [3] can then be used for edge orientation. Edge identification is important by itself since an edge between variables *X* and *Y* under certain conditions corresponds to a direct causal relation between the two variables, i.e., *X* directly causing *Y* or vice versa [5, 9, 10]. Thus, edges may be used to generate accurate hypotheses for causal discovery.

Since there may be many BNs that can capture the data joint distribution, the question is which one will **MMBN** discover? In [10] it is proved that all BN that are faithful to the same joint data distribution have the same set of edges. This unique set of edges is the one discovered by **MMBN**. *A BN is faithful to a joint distribution if all and only the independencies in the distribution are entailed by the Markov Condition.*

```
MMBN(Target T; Data D)
"Returns the edges in the BN that faithfully captures the
joint data distribution"
1. For all variables X
2.    Run MMPC(X, D)
3. End For
5. Edges = {edges (X, Y) such that $X \in$ MMPC(Y, D) or
      $Y \in$ MMPC(X, D)}
6. Return Edges

MMPC(Target T ; Data D)
"Returns the parents and children of T"
Phase I: Forward
7.  CPC = $\varnothing$
8.  Repeat
9.     For every variable X find
10.        minassocset(X)=subset s of CPC that minimizes
              assoc(X;T|s)
11.    End For
12.    F=variable of $\Phi - (\{T\} \cup$ CPC) that maximizes
              assoc(F;T|minassocset(F))
13     If $\neg$Ind(F;T|minassocset(F))
14.        CPC = CPC $\cup$ F /* Add X  to CPC */
15.    End If
16. Until CPC has not changed

Phase II: Backward
17. For all $X \in$ CPC
18.    If $\exists$s$\subseteq$CPC s.t. Ind(X;T|s)
19.        CPC = CPC – X  /* Remove X from CPC */
20.    End If
21. End For
22. Return CPC
```

Figure 1: Algorithms **MMBN** and **MMPC**.

MMBN (Figure 1) is based on a local discovery algorithm called Max-Min Parents and Children (**MMPC**) that identifies the parents and children set of (i.e., all variables with an edge to or from) a target variable of interest *T*. **MMBN** calls **MMPC** for all variables as targets and returns the edges discovered.

The real work in the global learning algorithm is done by **MMPC** (Figure 1). **MMPC** is based on tests of conditional independence and measures of association. By definition, *X* is independent of *T* given **Z** *iff* $P(X;T|\mathbf{Z}) = P(X|\mathbf{Z})P(T|\mathbf{Z})$ and is denoted as **Ind**(*X;T*|**Z**). The implementation of conditional tests of independence is based on the χ^2 test of the G^2 statistic and is explained in detail in [10], also used in [8]. The standard 5% threshold on *p*-values was used to reject independency and accept dependency.

In a BN faithful to its joint distribution, a variable *X* has an edge to or from *T* if and only if it is conditionally dependent with *T* given any other subset **Z** [10]. **MMPC** provides an efficient way of testing whether the above condition holds.

MMPC discovers the parents and children of *T* using a two-phase scheme In phase I, the forward phase, variables enter sequentially a candidate parents and children set, from now on denoted as **CPC**, by use of a heuristic function. The heuristic is admissible in the sense that variables with an edge to or from *T* and possibly more will enter **CPC**. In phase II, the backward phase, **MMPC** removes all false positives that entered in the first phase.

MMPC first includes in **CPC** the variable with the highest univariate association with *T*. **MMPC** chooses to include next into **CPC** the variable that exhibits the maximum association with *T* conditioned on the subset of **CPC** that achieves the minimum association possible for this variable. Intuitively the heuristic is justified as follows: *select the variable that, despite our best efforts to make it independent of T* (i.e., considering the minimum association conditioned on all possible subsets of **CPC)** *has the highest minimum association with T among all other candidate variables.* In the second phase false positives are removed when a subset of **CPC** is discovered conditioned on which they become independent of *T*.

As a measure of association, (function *assoc* in the pseudo-code of Figure 1), we used the negative *p*-value returned by the χ^2 test of independence: the smaller the *p*-value, the higher the association. Any other reliable statistical test of independence and measure of association can be employed.

In general, the tests of conditional independence and measures of association contain an exponential number of free parameters to be estimated to the size of the conditioning set. Unless this size is restricted, the estimation of the parameters and the tests themselves become unreliable. In our implementation, we do not perform any tests or measures of association when there are less than five training instances per parameter (cell in the G^2 table) to be estimated, in a similar fashion as in [10] and the **PC** algorithm. This restriction limits the number of subsets that have to be tried at Lines 10 and 18 in Figure 1 and significantly speeds up the algorithm. However, it may allow false positives to enter the output.

The unrestricted **MMBN** returns all and only the true edges under two assumptions: (i) there exist a BN faithful to the joint probability distribution of the data, and (ii) the statistical tests of independence and measure of associations are reliable. Its correctness is a corollary of the correctness of **MMPC** in [1].

3. Simulating Large Bayesian Networks

Typically, BN learning algorithms are tested on randomly generated networks, or networks that have been used in real Decision Support Systems, so that the structure of the network is known and can serve as a rigorous gold standard. Networks from real systems are expected to be a better representative sample of distributions likely to be encountered in practice.

Algorithm	Sensitivity	Specificity	Distance
PC	98%	93%	7%
TPDA	91%	96%	9%
SC/MI	98%	94%	6%
SC/Sc	96%	94%	7%
MMBN	98%	95%	5%

Table 1: Results on ALARM. SC/MI and SC/Sc stand for Sparse Candidate with the Mutual Information and Score heuristic respectively. *k=10* was used for Sparse Candidate (slightly worse results were obtained for *k=5*).

Algorithm	Sens.	Spec.	Dist.	Time
MMBN	81%	99.9%	18%	62 hours

Table 2: Results of **MMBN** on a 10,000 variables tiled ALARM and 1000 training instances. Statistics reported are on the task of edge rediscovery. Time measured on a 2.4GHz Intel Pentium Xeon with 2GB of memory.

Unfortunately, there are currently no publicly available Bayesian Networks used in real systems of the sizes required for our experimental purposes.

Instead of relying on randomly generated BNs, we invented a new method for generating large real-like BN. The method tiles multiple copies of smaller real BNs in a way that guarantees the structural and probabilistic properties of the tiles remain the same as in the originating networks.

The structural properties are maintained by only adding relatively few randomly chosen edges to interconnect the tiles. Preserving the probabilistic properties of the tiles is guaranteed as follows. If a tile with variables Φ has a joint distribution in the originating network denoted by P(Φ) and a marginal joint distribution in the large tiled network denoted by P′(Φ), then we impose the constraint that P(Φ)= P′(Φ). In our implementation the free parameters of the new conditional probability tables, generated by the addition of new edges are selected with uniform probability while respecting the constraints P(Φ)= P′(Φ). The details of the method are omitted due to lack of space.

4. Experimental Results

Experiment 1: MMBN Outperforms State-of-the-art BN Learning Algorithms.

The first set of experiments compares **MMBN** with state-of-the-art BN algorithms, namely **PC** [10], **TPDA** [6], and the Sparse Candidate algorithm [3]. **MMBN** is implemented using Matlab 6.5, while for the rest of the algorithms we used the publicly available versions and default values. One thousand training instances were generated by randomly sampling from the distribution of ALARM, a BN used in a medical diagnosis decision support system [2]; the data were then fed to the algorithms.

As a measure of comparison we used the *sensitivity* and *specificity* in edge discovery. The *sensitivity* of an algorithm is the ratio of correctly identified edges over the total number of edges in the original network. The *specificity* is the ratio of edges correctly identified as not belonging in the graph over the true number of edges not present in the original network.

An algorithm can achieve perfect sensitivity or specificity by including or excluding respectively all edges from the output. Thus, a combined measure of these statistics is needed. One such possible measure is the Euclidean distance of the sensitivity and specificity from the perfect score of 1:

$$d = \sqrt{(1 - sensitivity)^2 + (1 - specificity)^2}$$

The area under the ROC curve could not be used because the Sparse Candidate does not have a suitable parameter to vary and create the corresponding curve, while the rest of the algorithms provide few points on the curve for a large number of different thresholds.

The results are shown in Table 1. All algorithms took less than a couple of minutes to complete on a Pentium Xeon with 2.4GHz. **MMBN** outperforms all other state-of-the-art global BN learning algorithms on this task.

Experiment 2: MMBN Reconstructs a 10,000 Variables BN with Excellent Quality.

The second experiment demonstrates the scalability of **MMBN**. A network with approximately 10,000 variables was created by tiling 270 copies of ALARM as described in Section 3. Again, a thousand training instances were randomly generated from the network and **MMBN** was run to identify the edges in the networks. The results are shown in Table 2.

The first observation is that **MMBN** scales up very well to a large network with relatively small decrease in quality (keeping constant the size of the training sample). With **MMBN**, ordinary hardware is enough for experimentation with networks of the size encountered in a number of challenging biomedical domains. This is especially true considering **MMBN** is implemented in Matlab and has not yet been optimized for performance. Additionally, **MMBN** is an easily parallelizable algorithm.

Another observation is that specificity increases as the number of variables increase. Other preliminary results not reported here also confirm this observation. Increasing the number of variables in relatively sparse networks increases the number of true negatives. Thus, the results suggest that the rate of increase in false positives (that reduce specificity) is lower than the rate of increase of true negatives.

Depth	Sens.	Spec.	Dist.
1	79.17%	100.00%	20%
2	67.48%	100.00%	32%
3	59.56%	100.00%	40%
4	52.84%	100.00%	53%

Table 3: Results of **MMBN** on a 10,000 variables tiled ALARM and 1000 training instances. Statistics reported are for the task of reconstructing an area of radius *d* around a target *T*. The statistics shown are the averages over all variables.

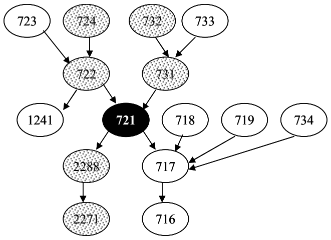

Figure 2: Reconstructed area of radius 2 by **MMBN** around target 721 in 10,000 variable tiled ALARM. The textured nodes are true positives.

Experiment 3: MMBN for Selective Local Reconstruction

When the number of variables is extremely large, e.g., half a million or so as in certain proteomics datasets, then not even **MMBN** is able to fully reconstruct the BN within a reasonable amount of time. In such cases, selective reconstruction of an area around a target node *T* may be the only option for a practitioner who wants to develop a causal theory targeted around *T*.

For the third set of experiments we modified **MMBN** to return the area of a BN of radius *d* edges away from a node *T* (denoted as *A(T, d)*), for all variables *T* and depth *d=1,2,3,4*. This is done using a breadth-first-search staring with *T*, then discovering the parents and children of *T*, then the parents and children of the parents and children of *T* and so on recursively.

The sensitivity of the algorithm is the ratio of correctly identified edges as belonging in *A(T, d)* over the true number of edges in *A(T, d)*. The specificity is ratio of the edges correctly identified as not belonging in *A(T, d)* over the true number of edges not in *A(T, d)*. Notice, that the true number of edges not in *A(T, d)* is any edges not in this region but also any possible edge among variables not in *A(T, d)*. The results are shown in the Table 3.

The first observation is that specificity is almost 100% (up to four digits accuracy). This is because there is a quadratic number of potential edges, in this case 10,000(10,000-1)/2, most of which are true negatives for any small area *A(T, d)* and the algorithm returns only a small fraction of these as false positives. Secondly, the average sensitivity (# of edges – # of false negatives)/(# of edges) of 81% (Table 2) is slightly different than the average of sensitivity of 79.17% (Table 3) over all targets for depth 1 (# of edges to/from *T* – # of false negatives around *T*)/(# edges to/from *T*).

Sensitivity for depth one is encouraging for the experimentalists: on average about 80% of direct causes and direct effects were identified with purely computational methods and no experiments. However, sensitivity quickly deteriorates with depth. The explanation is presented with an example. Figure 2 shows the reconstructed area around variable with index 721 in the network. The textured nodes and the edges between them are the ones correctly identified by **MMBN** using a breadth-first-search strategy. At depth 1 **MMBN** misses only a single node (717) when identifying the parents and children of node 721 and has sensitivity of ¾ =75%. However, missing the single node 717 is an error that propagates. Since, **MMBN** could not discover that the node belongs in the desired area to be reconstructed it did not expanded it, causing it in turn to miss the edges among nodes 718, 719, 734, and 716. The sensitivity at depth two is 6 correctly identified edges over 14 edges (42%), which is much lower than the sensitivity for depth one (75%). Had **MMBN** ran **MMPC** for node 717 (as it would have for a full reconstruction) the edges to variables 718, 719, 734, and 716 would have been discovered. Because errors propagate in general, we expect the error of selective local reconstruction to increase exponentially with increasing depth.

Summary of Experimental Results

The experiments constitute a proof-of-concept that **MMBN** outperforms current BN learning algorithms and in addition it scales up to very large networks with minimal quality degradation when the task is undirected edge identification. By focusing on edge identification and taking a local approach, selective reconstruction of targeted areas is also possible.

For an experimentalist our evaluation suggests that using **MMBN** (i) discovery of complete causal theories is possible with excellent quality for very large networks like the ones encountered in biomedicine, (ii) discovery of direct causal relations around a target variable *T* is possible with excellent quality for very large networks, and (iii) selective reconstruction is possible, albeit more difficult because of error propagation and accumulation.

5. Discussion and Conclusions

In this paper, we introduce the new algorithm **MMBN** for BN learning that in preliminary experiments outperforms other state-of-the-art algorithms and additionally, scales up to large BN with minimal degradation of quality. In addition, we show how **MMBN** can be used to selectively reconstruct targeted areas around a variable *T*.

The first provably correct local BN learning algorithm is the Grow-Shrink algorithm for discovering Markov Blankets instead of parents and children sets in [8]. Like **MMPC** it was then used as part of a global BN learning algorithm. The Incremental Association Markov Blanket algorithm [12] improves over the Grow-Shrink by employing a dynamic variable ordering heuristic. MMPC however, solves a different problem than Markov Blanket discovery, that of parents and children identification. LCD2 [7] is another local algorithm but in a different sense: it identifies a subset of the edges in the network but does not focus on a targeted area.

MMBN scales-up over **PC** and **TPDA** (two other constraint-based algorithms) for two reasons: it uses a powerful heuristic to identify potential members of the parents and children, and a different strategy for the order of performing tests of independence. In addition, compared to Bayesian search-and-score methods such as the Sparse Candidate or greedy hill-climbing, it first solves a relaxed version of the BN learning problem; that of structure identification. In contrast, current search-and-score methods identify and direct edges simultaneously.

Scalable algorithms are essential for two reasons. The first is that a number of significant datasets in biomedicine uses sizes of variable sets outside the current reach of BN learning algorithms. The second reason, indicated by our experiments, is that full BN learning may be necessary in order to achieve acceptable quality of local reconstruction, in particular regarding sensitivity.

In turn, local algorithms are important because in the worst case, full reconstruction is impossible rendering local edge identification the only currently available way of forming a local causal theory. For discovery of direct causal relations (depth one) the preliminary results are excellent and deteriorate only for indirect causality. Finally, local algorithms are important for variable selection. The Markov Blanket of a target variable, as shown in [11] is the minimum variable set that achieves optimal classification accuracy under certain conditions. The Markov Blanket of *T* is contained within an area of radius two edges around the target.

The sizes of datasets that are currently encountered in gene expression reach 12,000 different genes, while the true number of genes in the human genome is estimated in the neighborhood of 30K to 50K. In proteomics, the estimated number of different proteins in the human body is of the order of half a million, and in text categorization there are tens of thousands of different words in most corpora. Finally, in time stamped data the number of variables is the number of observed quantities is multiplied by the number of time steps. The presented algorithm is a first step in addressing the challenges these datasets pose for BN modeling and causal discovery.

Experiments are under way to determine the quality of directing the edges locally and globally using **MMBN**, superiority over other BN algorithms, and behavior with different sample sizes, as well as a theoretical comparison with other constraint-based algorithms. **MMPC** is one of a family of provably correct local algorithms [1]. The exploration of different heuristics, tests of independence, and measure of associations is also under way.

References

[1] Aliferis, C.F. and I. Tsamardinos, *Algorithms for Large-Scale Local Causal Discovery and Feature Selection in the Presence of Limited Sample or Large Causal Neighborhoods*. 2002, Department of Biomedical Informatics, Vanderbilt University.Technical Report DSL-02-08.http://discover1.mc.vanderbilt.edu/discover/public

[2] Beinlich, I.A., H. Suermondt, et al. *The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks.* in *Second European Conference in Artificial Intelligence in Medicine*. 1989.

[3] Friedman, N., I. Nachman, and D. Pe'er. *Learning Bayesian Network Structure from Massive Datasets: The ``Sparse Candidate" Algorithm*. in *Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*. 1999.

[4] Friedman, N., M. Linial, et al., *Using Bayesian Networks to Analyze Expression Data.* Computational Biology, 2000. **7**: p. 601-620.

[5] Glymour, C. and G.F. Cooper, eds. *Computation, Causation, and Discovery*. 1999, AAAI Press/The MIT Press: Menlo Park, California, Cambridge, Massachusetts, London, England.

[6] Jie, C., R. Greiner, et al., *Learning Bayesian Networks from Data: An Information-Theory Based Approach.* Artificial Intelligence, 2002. **137**: p. 43-90.

[7] Mani, S. and G.F. Cooper. *A study in causal discovery from population-based infant birth and death records*. in *American Medical Informatics Association (AMIA)*. 1999.

[8] Margaritis, D. and S. Thrun. *Bayesian Network Induction via Local Neighborhoods*. in *Advances in Neural Information Processing Systems 12 (NIPS)*. 1999.

[9] Pearl, J., *Causality: Models, Reasoning, and Inference*. 2000: Cambridge University Press.

[10] Spirtes, P., C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. Second ed. 2000, Cambridge, Massachusetts, London, England: The MIT Press.

[11] Tsamardinos, I. and C.F. Aliferis. *Towards Principled Feature Selection: Relevancy, Filters, and Wrappers*. in *Ninth International Workshop on Artificial Intelligence and Statistics*. 2003. Key West, Florida, USA.

[12] Tsamardinos, I., C.F. Aliferis, and A. Statnikov. *Algorithms for Large Scale Markov Blanket Discovery*. in *The 16th International FLAIRS Conference*. 2003. St. Augustine, Florida, USA.