Adaptive Bayesian Networks: Exploratory Overview

Matt Piekenbrock – Wright State University

March 20, 2016

I. Abstract

Bayesian networks are directed acyclic graphs (DAGS) that represent dependencies between variables in a probabilistic model. Adaptive Bayesian Networks are often used for *general world modeling*, where legitimate conclusions can be drawn for any state in the world being modeling, given the current knowledge about the world. First, a brief introduction on conditional probability rules and Bayesian networks construction methods is given. The exact formulation of how Bayesian Networks are constructed is examined in the Methodology section. A small conclusion section adds a small discussion some of the recent work that has been done in terms of model selection.

II. Introduction

Probabilistic Graphical Models–using graphs to express the conditional dependency structure between random variables– has been called "the marriage of probability theory and graph theory"[1]. Whereas the phenomenon of complex systems can be modeled through modern fields such as Network Science, Bayesian Networks are graphical models that allow for efficient representations of the dependency structures that drive such systems. Modern BNMs constitute a framework for *general world modeling* in such a way that is not only mathematically rigorous, but is also *adaptable* and *scalable* to modern day problems. Bayesian Networks have been used to forecast short-term traffic patterns before[2][3]. They have also been proved useful in diagnosing medical diseases [4], predicting stock prices fluctuations [5], and in several classification applications [6]. Bayesian Network Analysis techniques have even been used to analyze flight delays before[7]. In the following section, I review the basics of a Bayesian Network representation. I establish some of the basic definitions used within the field, and then I introduce the two primary methodologies used to automatically train a Bayesian Network. I then go over some of the popular algorithmic implementations of such training methods, and discuss the advantages and disadvantages of both. Finally, I conclude with with a small discussion of the general methods preferred for model-selection.

A. Bayesian Network Representation

Predictive modeling is fundamentally rooted in the field of probability theory. A simple way of modeling the states of a world would be through the use of a joint probability distribution that accounts for every state of every combination of every random variable (knowledge) within a system. For n discrete random variables $X_1, X_2, \dots, X_n$, the joint probability mass function would be defined as:

$$P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$$

Knowing the joint probability distribution (JPD) of a set of *random variables* (RV) allows one to model any outcome of the world encoded by those random variables. That is, given a JPD, one can answer all possible inference queries by marginalization. Unfortunately, the joint probability distribution of even a small number of RVs is computationally expensive to manipulate and usually too large to store in memory [8]. Furthermore, due to the extraordinary larger possible number of combinations that are intrinsically a result of computing joint probabilities (2^n possible RV combinations for n binary-valued random variables), the resulting probabilities for many RV assignments are extremely small, translating to an unintuitive solution when modeling future events[8]. It is possible, however, to simplify the representation of a JPD if there exists independence between RVs:

$$P(X = x, Y = y) = P(X = x)P(Y = y)$$

Using the chain rule of probability, a JPD can also be represented as the product of conditional probability functions, as follows:

$$P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) =$$

$$P(X_1 = x_1) \times$$

$$P(X_2 = x_2 | X_1 = x_1) \times$$

$$P(X_3 = x_3 | X_2 = x_2, X_1 = x_1) \times$$

$$P(X_n = x_n | X_1 = x_1, X_2 = x_2, \dots, X_{n-1} = x_{n-1})$$

This idea can be extended when one RV can be used to "explain away" [9] another RV by exploiting *conditional independence*. That is, X and Y are conditionally independent given Z if:

$$P(X = x, Y = y | Z = z) =$$

$$P(X = x | Z = z) \cdot P(Y = y | Z = z)$$

But collectively utilizing these factorization rules, a compact representation of the joint probability distribution can be constructed by learning the conditional independence structure

over a set of random variables[1]. These conditional independence (CI) assumptions can be encoded into a DAG G, where nodes represent random variables and directed edges represent the conditional dependencies between them. The network structure along with the product of each random variable's conditional probability distribution (CPD) is called a **Bayesian Network (BN)**.

B. Learning the conditional independence structure

Constructing a Bayesian Network is fundamentally done in one of two ways:

1) Manually specifying the relational structure (CPDs) between the RVs of interest and then estimating the optimal parameters for each distribution, or
2) Learning the CI network structure through applied structure-learning analysis $of\ the\ data\ itself$

While the first is simpler to implement and may lead to better results if a domain expert is involved or if the data is considered vastly incomplete, the latter is preferred when one of the goals is knowledge discovery[8]. Additionally, in the context of structure-based learning methods, there are three primary approaches that are used to learn the structure of a Bayesian Network, given a dataset:

1) Constraint-based learning
2) Score-based learning
3) Bayesian model averaging

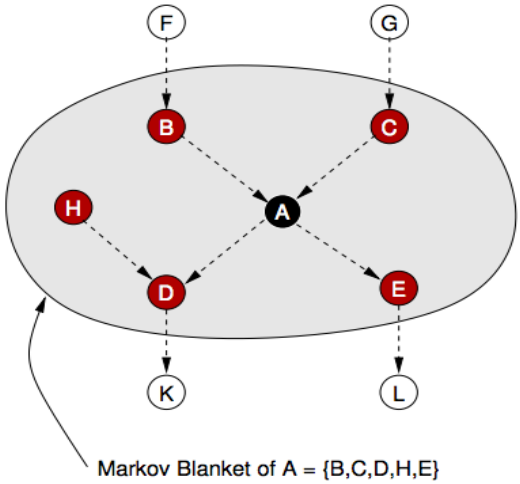
It's important to mention that sometimes a hybrid approach is taken to BN construction, where some conditional (in)dependencies are "blacklisted" or "whitelisted" based on some *a priori* domain knowledge. Furthermore, the third approach to BN structure-learning is called *Bayesian model averaging*; this approach does not attempt to learn a single structure, but rather when probability queries (predictions) are generated, they are averaged over a set of possible Bayesian Network structures (the set is determined by varying the CI test(s) α levels). Koller & Friedman note this in their book, stating: "The space of Bayesian Networks is a combinatorial space, consisting of a superexponential number of structures – $2^{O(n^2)}$. ...Since the number of structures is immense, performing this task seems impossible. For some classes of models this can be done efficiently, and for others we need to resort to approximations." [8] For these reasons, only a subset of the constraint-based and score-based structure learning algorithms were used of this project, as outlined more in detail in the experimental results portion of this report.

C. Markov Blanket

Learning the structure of a BN primarily involves finding a specific substructure for each RV in the JPD called the *Markov Blanket*[10]. In order to be a proper Bayesian Network, for some JPD P over a set of random variables V and a DAG $G = (V, E)$, (G, P) must satisfy the Markov condition for every variable $X \in V$ such that $\{X\}$ is conditionally independent of the set of all of its non-descendants ND_X given the set of all of its parents PA_X [9]. That is:

$$I_P(\{X\}, ND_X \mid PA_X)$$

Put more simply, a node is conditionally independent of its non-descendants, given its parents. An example of what a Markov Blanket would look like for a RV A in a Bayesian Network is given in figure 1 below.



(e) Final Markov Blanket of A : $\{B, C, D, H, E\}$.

Figure 1: Markov Blanket of RV A

Knowing the optimal (minimal) Markov Blanket for the complete set of RVs is extremely important, as all of the RVs within the Markov blanket of a given RV encompass all of knowledge needed to predict the behavior of the latter RV. This is discussed in both "Learning Bayesian Networks"[9] and "Probabilistic Graphical Models"[8] in further detail. Thus, in the BN represented in figure 1, prediction of A only requires previous knowledge (data) of RVs $\{B, C, D, H, E\}$. Removing nodes, even (non-parent) ancestors, outside a specific target RVs' Markov Blanket will not effect predictions of the future behavior of the target RV, because the target RV (A) is "shielded" from the *downward propagation* of conditional probability queries. The intuitive reasoning behind this is discussed more in the *Constraint-based Learning* section below.

III. Methodology

A Bayesian Network is defined as a directed acyclic graph (DAG) $G(V, E)$ whose nodes (V) represent the random variables (RV) in some domain of interest and whose edges (E) encode the conditional dependencies between them. Directed

[1]In both primary types of structure learning (see below), hypothesis tests are used to test for conditional *independence*, as opposed to causal *dependence*.

edges, thus, represent the direct influence of RVs onto other RVs. The structure of a Bayesian Network also encodes the set of conditional independence assumptions within a JPD. Learning the conditional independence (CI) structure of a BN is not a trivial process, and there are multiple algorithms that have been developed to do so. The two primary types of CI structure learning are presented below: Constraint-Based Learning, and Score-Based Learning.

A. Constraint-based Learning

Constraint-based learning methods focus on finding the *minimal Markov blanket* of some *target* RV through a series of conditional independence (CI) tests. That is, the markov blanket of a variable of interest T is the minimal set of other RVs T is conditioned on such that all other RVs not in the markov blanket are conditionally independent of T. There are two constraint-based learning algorithms that were used to construct a BN for this project:

1) Grow-Shrink [11] := Markov blanket detection test that consists of a growing phase (where RVs are admitted if they fail the CI tests) and a shrinking phase (where previously RVs are removed due to "shielding" effect). CI testing done in an arbitrary order.
2) Incremental-Association [12] := Markov blanket detection test that performs multiple CI tests between RVs, iteratively removing RVs in the Markov Blanket in an interleaved fashion.

Grow-Shrink Algorithm: The Grow-Shrink algorithm is a two-phase algorithm that relies on pairwise CI tests to *grow* the Markov Blanket for some target variable T, and then in the end the Markov Blanket is *shrunk* to remove RVs that were "explained away" from a RV introduced towards the end of the growing phase. To illustrate this, here is a summary is the grow-shrink algorithm on some target RV $T = A$ over a set of RVs $V = \{B, F, G, C, K, D, H, E, L\}$:

1) Given a RV A with an empty Markov blanket $S =$, begin a pairwise conditional independence (CI) hypothesis test over V (order is arbitrary)
2) Test $A \perp\!\!\!\perp B \mid \{\}$. If the test fails, then add B to S, otherwise move to next RV. Assume in this case the test fails (A and B are not conditionally independent).
3) Next test $A \perp\!\!\!\perp F \mid \{B\}$; if the test fails, add F to A's Markov blanket S
4) Continue pairwise CI tests until all other RVs have been tested

The previous algorithm is known as the *growing phase*, since RVs are constantly being added the Markov Blanket of A. It is noticeable that, since each CI test depends on the previous RVs that were tested for conditional independence, the order in which RVs are added to A's Markov blanket depends on the order that they are tested. Furthermore, because of this, it is possible that RVs that were previously added to the Markov blanket are conditionally "explained away"[2]. The dependency that may have existed given a different set of RVs (given alternative prior knowledge) may or may not be uncovered. Because of this, a *shrinking phase* is needed to do a final pass to uncover A's minimal Markov Blanket:

1) Test $A \perp\!\!\!\perp X \mid \{S \setminus X\}$ where X is one of the RVs in S (except in the current test)
2) If the test returns true, remove X from S

Finally, this process is repeated for all other RVs. It's important to note that this algorithm assumes $Faithfulness$, where the CI test with a pre-specified p-value is assumed to be an accurate depiction of the conditional independence that exists between the RVs being tested. A more detailed explanation of the grow-shrink algorithm can be found in Dimitris' original PhD thesis[11].

Incremental Association: Another popular constraint-based algorithm that was used in this project is called the incremental association algorithm (IAMB). The incremental association algorithm theoretically resembles the Grow-Shrink algorithm in that it consists of two-phases, one of which adds RVs to the Markov Blanket and another that removes (called the backward-conditioning) RVs from the Markov Blanket, however in comparison with the Grow-Shrink (GS) algorithm, IAMB iteratively interleaves backward-conditioning for each forward-selection phase (CI testing + Markov Blanket expansion) to reduce the type I error associated with the Markov Blanket being constructed for the target node. This was motivated by the fact that, "the smaller the conditioning test given a finite sample of fixed size, the more accurate the statistical tests of independence and the measure of associations."[11]. It's noted further than one visual improvement one might see using IAMB compared to GS is that strong (potential) spouses that have common children are entered into the Markov Blanket earlier on, improving the strength of the subsequent CI tests. The pseudo-code for the IAMB algorithm is given below:

Phase I (forward)
$CMB = \emptyset$
While CMB has changed:
  Find the feature X in $V - CMB - \{T\}$
    that maximizes $f(X; T \mid CMD)$
  If not $I(X; T \mid CMB)$
    Add X to CMB
  End If
End While
Phase II (backwards)
Remove from CMB all variables X,
  for which $I(X; T \mid CMB - \{X\})$
Return CMB

B. Score-based Algorithms

An alternative approach to automatically learning the structure of Bayesian Networks is through the use of score-based learning algorithms. Score-based approaches treat the problem of BN learning as an optimization problem over a *hypothesis space*–a set of possible BN networks. Each [Bayesian] network

[2]See the "Bayesian Network Representation" section of this report for a better explanation of a variable being "explained away"

in the hypothesis space is scored against some *objective function*[8]. There are two primary categories of scoring functions, either *Bayesian* or *Information-theoretic* scoring functions.

In Bayesian scoring methods, the set of possible Bayesian Networks makes up a *prior* probability distribution, and the best Bayesian Network B is the one that maximizes the posterior probability given the dataset T, $\max(P(B|T))$.

In *information-theoretic* scoring functions, rather than maximizing the posterior probability of the hypothesis space of BNs given the data $\max(P(B|T))$, the information content of T induced by the distribution of BNs B is measured as a heuristic. All information-theoretic scoring functions were based on Claude Shannon's original definition of information and his related concept of *Entropy*[13]. In this research, only information-theoretic were considered due to their effectiveness and familiarity. There are several objective functions often used in scoring functions:

1) Log-likelihood
2) Akaike Information Criterion [14]
3) Minimum Description Length/Bayesian Information Criterion [15]

One oft-used information-theoretic scoring function is the log-likelihood function (LL):

$$LL(T \mid B) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$$

r_i represents the number of states of a RV X_i, q_i is the product of every r_i for every possible combination of parent RVs to the current RV ($q_i = \prod_{X_j \in \prod X_i} r_j$), N_{ijk} represents the number of instances in the data T where the variable X_i takes on the k-th value x_{ik} *and* the RVs in the parent set take on some j-th configuration of $\prod_{X_i} (1 \leq j \leq q_i)$. For further details, see the original paper[16] by Alexandra M. Carvalho that summarizes these measures and the effects they have on constructing Bayesian Networks.

All three of mentioned measures can be used interchangeably with the previous equation to measure the "quality" of the space of potential BN configurations through a general form:

$$\phi(B \mid T) = LL(B \mid T) - f(N)|B|$$

Where $f(N)$ represents the penalization function. If $f(N) = 1$, the BNs are evaluated according to their Akaike Information Criterion (AIC), if $f(N) = \frac{1}{2}\log(N)$, the optimization is with respect to the Bayesian Information Criterion (BIC), and if $f(N) = 0$, then it's just the normal log-likelihood evaluation.

IV. Conclusion

The concludes a brief overview of the concepts of automatically training a basic Bayesian Network, including some of the research efforts that have been done on model selection and training strategies.

References

[1] Z. Ghahramani, "Learning dynamic bayesian networks," in *Adaptive processing of sequences and data structures*, pp. 168–197, Springer, 1998.

[2] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 124–132, 2006.

[3] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Predicting traffic flow using bayesian networks," *Transportation Research Part B: Methodological*, vol. 42, no. 5, pp. 482–509, 2008.

[4] D. Nikovski, "Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 12, no. 4, pp. 509–516, 2000.

[5] K. Eisuke, M. Harada, and T. Mizuno, "Application of bayesian network to stock price prediction," *Artificial Intelligence Research*, vol. 1, no. 2, p. p171, 2012.

[6] J. Cheng and R. Greiner, "Comparing bayesian network classifiers," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 101–108, Morgan Kaufmann Publishers Inc., 1999.

[7] N. Xu, K. B. Laskey, C.-H. Chen, S. C. Williams, and L. Sherry, "Bayesian network analysis of flight delays," in *Transportation Research Board 86th Annual Meeting, Washington, DC*, 2007.

[8] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[9] R. E. Neapolitan *et al.*, "Learning bayesian networks," 2004.

[10] J. Pearl, "Morgan kaufmann series in representation and reasoning. probabilistic reasoning in intelligent systems: Networks of plausible inference," 1988.

[11] D. Margaritis, *Learning Bayesian network model structure from data*. PhD thesis, US Army, 2003.

[12] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov, "Algorithms for large scale markov blanket discovery.," in *FLAIRS conference*, vol. 2, 2003.

[13] C. Shannon, "A mathematical theory of communication, bell system technical journal 27: 379-423 and 623–656," *Mathematical Reviews (MathSciNet): MR10, 133e*, 1948.

[14] H. Akaike, "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.

[15] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[16] A. M. Carvalho, "Scoring functions for learning bayesian networks inesc-id tec. rep. 54/2009 apr 2009," 2009.