Bayes Nets

Prepared by Dhruv Malik and Andy Palan

March 17, 2017

Adapted from Pieter Abbeel's and Dan Klein's lectures.

Background

Before jumping into the theory of Bayes Nets, we first remind ourselves of a few basic concepts from probability.

Definition: A is **conditionally independent** of B given C (we write $A \perp B|C$) if

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

This is equivalent to having $P(A, B|C) = P(A|C) \cdot P(B|C)$.

Chain Rule: For any set of random variables $\{X_1, X_2, \ldots, X_n\}$, we have that

$$P(X_1, \ldots, X_n) = P(X_1) \cdot P(X_2|X_1) \cdot \ldots \cdot P(X_N|X_1, X_2, \ldots, X_{n-1})$$

This last result is fairly trivial to prove with Bayes Rule.

The Basics of Bayes Nets

Bayes Nets are a technique for describing complex joint distributions using simple, conditional (also known as, local) distributions.

A natural question is why do Bayes Nets offer an improvement on describing joint distributions in their entirety?

1. For settings where they are many variables interacting together, the joint distribution table is too big to express explicitly.

   If we had n variables, each with domain size at most d, the size of the joint distribution table is in the order of $\mathcal{O}(n^d)$. It's not too hard to see that for even modestly-sized d and n, it becomes infeasible to store the joint distribution table explicitly.

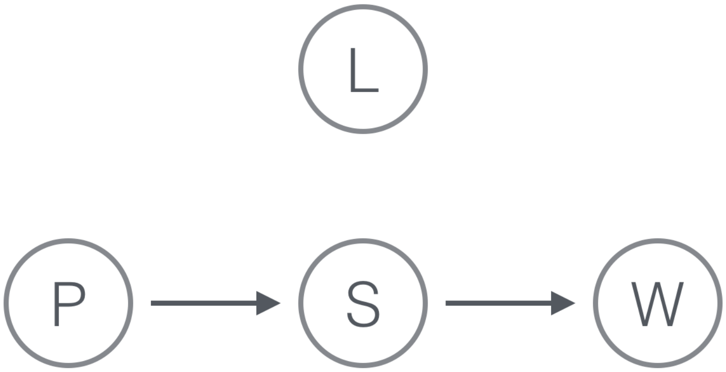
2. From a practical perspective, it is often very difficult to learn anything about more than a few variables at a time.

   It is often easier empirically to study relationships between variables locally (i.e. how they interact directly with a small number of each other). This is in turn sufficient to understand the problem in its full scale, as we can chain together these local relationships to learn about any global, indirect relationships (more on this later).

Intuitively, it should be clear that it is rather difficult to isolate relationships between variables when a large number (e.g. 50) of them are acting together.

It is much easier to do so when dealing with systems containing 2-3 variables.

Now, let's look at an example. We consider a simple Bayes Net between four variables; let L represent the event that Liverpool win their next game, P represent the event that Tom goes to a party on a given night, S represent the event that Tom gets a full night's sleep that night and W represent the event that Tom wakes up for his exam the next morning.



A Bayes Net consists of the following:

1. **Nodes**: A Bayes Net contains a set of nodes, one for each variable. In the Net above, L, P, S and W are all nodes, each representing a variable.

2. **Arcs**: Arcs (or arrows) in a Bayes net indicate influence between variables. In the Net above, we note that there are arcs between P and S and again between S and W; this suggests that there is some influence between those variables.

3. **Conditional Distribution for Each Node**: Each node has an associated conditional distribution that depends on its parent nodes in the Net. For example, in the Bayes Net above, the node W will have an associated conditional probability distribution, $P(W|P)$.

This will be further discussed in the next section; see: "Encoding Of Conditional Distributions".

In this example, there is an arc between P and S because Tom's decision whether or not to attend a party will undoubtably affect his ability to get a full night's sleep. Similarly, there is an arc between S and W because there exists a relationship between the two variables.

There is no arc between L and any of the remaining variables because there is no (obvious) relationship between Liverpool's performance in their next game and any of Tom's life events. In other words, L is independent of P, S, and W.

Now, note that there is no arrow directly between P and W, although they are both connected to S. In other words, P is not an immediate parent of W. In such cases, we assume that W is **conditionally independent** of P given S (i.e. its parent).

The astute reader will by now have noticed that a Bayes Net is a directed acylic graph.

This property of a Bayes Net leads to some nice topological properties that can be exploited; however, this is not something we will concern ourselves with in this brief note.

Encoding Of Conditional Distributions

As mentioned in the tail-end of the last section, the absence of an arc between two nodes implies that we are assuming that the two nodes are conditionally independent. What this suggests is that Bayes Nets (and their arcs in particular) really encode our conditional independence assumptions. In other words, we are assuming that any node X_i is conditionally independent of its non-parent nodes *given its parent nodes*.

Hence, each node in a Bayes Net is imbued with a conditional (i.e. local) distribution that depends only its parents i.e. for each node X_i, we have an associated conditional distribution

$$P(X_i|X_1, X_2, \ldots, X_{i-1}) = P(X_i|\text{parents}(X_i))$$

This result falls directly from the definition of conditional independence

Armed with this fact, it becomes fairly easy and cheap to compute the probability of any set of observations occuring:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i))$$

Below is a brief sketch of the proof of this result:

$$\begin{aligned} P(x_1, x_2, \ldots, x_n) &= P(x_1|x_2, \ldots, x_n) \cdot P(x_2, \ldots, x_n) \\ &\overset{(a)}{=} P(x_1|\text{parents}(X_1)) \cdot P(x_2, \ldots, x_n) \\ &= P(x_1|\text{parents}(X_1)) \cdot P(x_2|\text{parents}(X_2)) \cdot P(x_3, \ldots, x_n) \\ &\overset{(b)}{=} \ldots \\ &= \prod_{i=1}^{n} P(x_i|\text{parents}(X_i)) \end{aligned}$$

(a) by conditional independence

(b) proceeding inductively

Hence, we can say that Bayes Nets implicitly encode joint distributions as a product of local conditional distributions. We will discuss further the process of finding desired joint distributions (and more generally, the process of inference) from a Bayes Net in the next section.

Inference

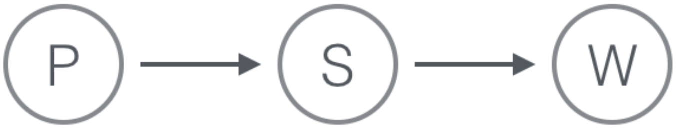
Now that we understand what Bayes Nets are and what they represent, the next logical question is how can we gain useful insight from the Net? This question forms the basis of the following discussion on inference, the act of extracting information from a joint probability distribution.

In general, we want to find the probability that some event occurs given that some evidence from the system. Mathematically speaking, we are concerned with a probability of the form $P(Q|e_1, \ldots, e_k)$, where Q is our query, the event we are primarily concerned with; each e_i is an observation of some variable E_i, which we refer to as evidence variables; and all other variables in the net, which are not of immediate importance to us, are called hidden variables, and arbitrarily labelled H_j.

There are two primary techniques for inference: inference by enumeration and variable elimination. They both follow almost directly from Bayes Rule and vary only slightly, in the order in which certain operations are carried out.

Inference by Enumeration

As discussed earlier, each node in a Bayes Net is imbued with a conditional distribution that depends only on its parents; we refer to each such distribution as a **factor**.



For example, in the Bayes Net given above (which is identical to that from the previous section, except without the L node), we have three factors, $P(P)$, $P(S|P)$, $P(W|S)$. In this simple example, we assume that all variables are binary (i.e. $P = +p$ or $P = -p$ only; similar for S and W).

The general outline for inference by enumeration is as follows:

1. **Select all known values as given by the evidence**. i.e. if we are given that $S = +s$, in all our initial factors that involve S (here, this would be $P(S|P)$ and $P(W|S)$), we may concern ourselves only with those rows where $S = +s$ and ignore all other entries that contain over instances of S.

2. **Join all the factors together to get the joint distribution that**

includes all variables (i.e. over the query, evidence and hidden variables). Put in mathematical notation, we essentially want to compute $P(Q, h_1, \dots, h_r, e_1, \dots, e_k)$; this can be done trivially using the chain rule.

3. Once we have the joint, we want to **eliminate all the hidden variables and get the joint distribution over only the query and evidence variables**. (This step is at times referred to as marginalizing the joint distribution.) We can do so by summing up over all possible values of all hidden variables. Put mathematically, we want to compute $P(Q, e_1, \dots, e_k) = \sum_{h_1,\dots,h_r} P(Q, h_1, \dots, h_r, e_1, \dots, e_k)$.
4. Finally, to compute the conditional probability of the query variable given the evidence variable, we **apply Bayes Rule** and find

$$P(Q|e_1, \dots, e_k) = \frac{P(Q, e_1, \dots, e_k)}{\sum_q P(q, e_1, \dots, e_k)}$$

The term on the denominator on the right hand side is often referred to as the **normalization constant**; as the name suggests, this constant should be pre-computed and stored to speed up computations.

Note that this really is just a result of applying Bayes Rule even though it may not immediately look like it.

$$\begin{aligned} P(Q|e_1, \dots, e_k) &= \frac{P(Q, e_1, \dots, e_k)}{P(e_1, \dots, e_k)} \\ &= \frac{P(Q, e_1, \dots, e_k)}{\sum_q P(q, e_1, \dots, e_k)} \end{aligned}$$

As an exercise, the reader should attempt to write out the steps that would be required to compute $P(W|P)$ and $P(W)$ using the basic technique of inference by enumeration.

Inference by Enumeration is a very intuitive but unfortunately, naive approach at inference in Bayes Nets.

The computational complexity of the process of inference depends on the largest factor being generated (where size is determined by the number of entries in the joint distribution table). In the simple case where we are dealing with n binary variables, the size of the largest factor generated through this approach in 2^n, which for even modest-sized n, is rather large. The question then arises, can we do better? The answer is yes.

Variable Elimination

The large factor sizes in inference by enumeration arise because of the fact we join all the variables together into a single joint distribution before marginalizing any variable. An alternative approach would be to interweave the process of joining and marginalizing factors; this would (likely) keep the size of the largest factor down as we would never have to deal with a joint distribution table containing all n variables together.

The general outline for variable elimination is very much like that of inference by enumeration, with one major modification:

1. Follow Step 1 of inference by enumeration.
2. While there are still hidden variables:
   a. Pick a hidden variable, H.
   b. Join all factors that mention H. (This is similar to Step 2 of inference by enumeration, except you only join those factors that mention H).
   c. Eliminate H by summing the resulting joint distribution over all possible values of H (as in Step 3 of inference by enumeration)
3. Join all remaining factors.
4. Normalize to arrive at the desired conditional probability.

As an exercise, the reader should attempt to write out the steps that would be required to compute $P(W|P)$ and $P(W)$ using variable elimination.

VARIABLE ELIMINATION, like inference by enumeration, is still an NP-hard problem, but is often much faster than inference by enumeration for the reasons outlined above.

A natural question to ask about variable elimination is "Does the order in which the factors are joined and marginalized affect the efficiency of the algorithm (by keeping factors small)?" Unfortunately, an efficient ordering does not always exist, although there are certain cases where they definitely do.

One such case is where the Bayes Net is in fact a polytree (a directed graph with no undirected cycles).

And hence, one potential (and perhaps naive) approach to speeding up inference computations in a Bayes Net is to simply remove a set of variables from the Net such that the nodes left behind form a polytree, allowing us to find an efficient ordering.

This method may be effective but it should not be hard to see that this method carries with it significant drawbacks.