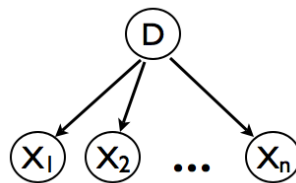


Bayesian networks, continued

1 Applications

Bayes nets have been applied to a wide variety of problems, and here we outline just a few. An early example was PATHfinder, a system which diagnosed pathologies in lymph nodes. The earliest versions of this system were based on a system of formal rules, but later versions used Bayes nets. The first Bayes net structure that was incorporated into PATHfinder is called **Naive Bayes**. A Naive Bayes network for medical diagnosis has a single node D which represents whether or not the patient has a particular disease, and all of the other variables X_1, X_2, \dots, X_n are direct children of the disease node. These variables represent symptoms, and we suppose that all symptoms are independent given the presence or absence of the disease. Here is such a network:



Applying our general definition of Bayes nets, the joint distribution over all of the variables is given by:

$$P(D, X_1, \dots, X_n) = P(D) \prod_{i=1}^n P(X_i | D).$$

This structure makes it easy to compute the conditional probability of a disease given the presence or absence of each of the symptoms:

$$\begin{aligned}
 P(d^1 \mid x_1, \dots, x_n) &= \frac{P(d^1, x_1, \dots, x_n)}{P(x_1, \dots, x_n)} \\
 &= \frac{P(d^1, x_1, \dots, x_n)}{P(d^1, x_1, \dots, x_n) + P(d^0, x_1, \dots, x_n)} \\
 &= \frac{P(d^1)P(x_1 \mid d^1) \cdots P(x_n \mid d^1)}{P(d^1)P(x_1 \mid d^1) \cdots P(x_n \mid d^1) + P(d^0)P(x_1 \mid d^0) \cdots P(x_n \mid d^0)}
 \end{aligned}$$

The conditional probability of a disease given its symptoms can, therefore, be computed in linear time.

The next version of the PATHfinder network eliminated 10% of all of the misdiagnoses of this network just by eliminating all of the CPT entries which were assigned the value 0. No amount of evidence can make an event seem possible if it had a prior probability of zero. More generally, unless an event is absolutely impossible, it is usually a bad idea to assign it a CPT entry of 0 in a Bayes net.

Later, PATHfinder was expanded into a full Bayes net, which could take into account not only symptoms, but other factors such as family history or behavior, which could affect the prior probability of having a disease. The overall results with this full Bayes net were equivalent to saving 1 life in 1000.

PATHfinder was shown to outperform human pathologists in many situations because:

- Bayes nets incorporate the prior probabilities of various diseases in a principled way. Often, people have trouble precisely weighting prior probabilities against the evidence. For example, psychology studies have shown that human physicians tend to weight the prior probability less than they should. Instead, they tend to focus on the probability of the symptoms given the disease, thereby assigning high likelihoods to very rare diseases.
- Bayes nets are better at incorporating all of the different pieces of evidence available. Humans have a hard time keeping more than 7-9 pieces of evidence in their heads at a time, while Bayes nets can easily consider dozens of pieces of evidence.

With Bayes nets, one can also determine which further piece of evidence would be most useful to observe, possibly taking into account that it may cost different amounts to observe different variables. In medical diagnosis,

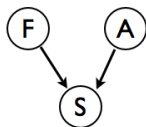
this helps avoid unnecessary medical tests, thereby saving time and money, and possible even sparing the patient from painful procedures.

As another example, Microsoft uses a Bayes net to diagnose printer errors. Using a Bayes net rather than a set of hard-coded rules allows the system to make good predictions even if the user chooses not to enter particular information, such as the results of printing out a test page.

2 Parameter learning

We have discussed the semantics of Bayes nets and how to perform inference. We now discuss how to come up with the Bayes net in the first place. There are actually two problems: **structure learning** (determining the BN graph structure) and **parameter learning** (assigning values to the CPT entries). Typically, we specify a Bayes net structure by hand rather than use a learning algorithm. There are various algorithms for learning Bayes net structures from data, but it is far more common to hand-specify the graph structure, and so we won't discuss these algorithms here. Instead, we will focus on the more important problem of parameter learning.

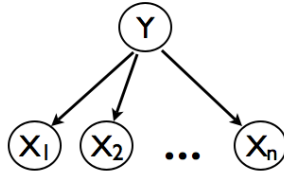
Let's return to our earlier flu network example. Recall that we had three random variables: flu (F), allergy (A), and sinus trouble (S). The network structure was as follows:



Suppose we have a database of patients, and we know the values of F , S , and A for each of them. Then, we would estimate $P(f^1)$ as the fraction of people in this database who had the flu. To estimate $P(s^1 | f^1, a^1)$, we would use the fraction of people with the flu and allergies who had a sinus infection.¹

Finally, it's worth noting that the Naive Bayes network mentioned above is often used as a supervised learning algorithm. More specifically, assume we have n discrete-valued random variables X_1, X_2, \dots, X_n , and we are trying to predict the value of a discrete-valued target variable Y . Suppose we have a training set $S_{\text{train}} = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$. We apply the Naive Bayes network structure:

¹As an exercise, try to justify this method using maximum likelihood.



We estimate all of the parameters for this network just as we did above. Then, given a new example $X = (X_1, X_2, \dots, X_n)$, we predict

$$\begin{aligned}
 \arg \max_y P(y \mid x_1, \dots, x_n) &= \arg \max_y \frac{P(y, x_1, \dots, x_n)}{P(x_1, \dots, x_n)} \\
 &= \arg \max_y P(y, x_1, \dots, x_n) \\
 &= \arg \max_y P(y) \prod_{i=1}^n P(x_i \mid y). \quad (1)
 \end{aligned}$$

Finally, an implementation note. If we were to compute (1) directly, we might have problems with numerical underflow, especially if any of the probabilities are very small. Instead, we take the logarithm of each of the terms, and choose as our prediction:

$$\arg \max_y \log P(y) + \sum_{i=1}^n \log P(x_i \mid y).$$

This can be safely computed without numerical underflow.

Naive Bayes often does not perform as well as logistic regression or a well-designed decision tree, but it is such a simple algorithm that it is worth using in many cases.