Lecture 2: Simple Bayesian Networks

Simple Bayesian inference is inadequate to deal with more complex models of prior knowledge. Consider our measure:

$$Catness = |(R_l - R_r)/R_r| + |(S_i - 2 \times (R_l + R_r))/R_r|$$

We are currently weighting the two terms equally, but perhaps this is not a good idea. Moreover we may want new terms, for example fur colour around the putative eyes. A more complex catness measure might be:

$$Catness = \alpha|(R_l - R_r)/R_r| + \beta|(S_i - 2\,(R_l + R_r))/R_r| + \gamma(ColourMatch) + \&c.$$

α, β and γ are constants to be determined. The whole process becomes very heuristic and we need to look for better methods for representing our prior models. Consider the case where we have evidence from more than one source. We could write Bayes' theorem as follows:

$$P(D|S_1\&S_2\&S_3\cdots\&S_n) = \frac{P(D)P(S_1\&S_2\cdots S_n|D)}{P(S_1\&S_2\&S_3\cdots S_n)}$$

Already we have a problem. The term $P(S_1\&S_2\cdots S_n|D)$ is of little use for inference since for large n we are unlikely to be able to estimate it. To get round this problem we normally make the assumption that the S_i are independent given D, this allows us to write:

$$P(S_1\&S_2..S_n|D) = P(S_1|D)P(S_2|D)\cdots P(S_n|D)$$

This has the advantage that each individual term $P(S_i|D)$ can be estimated from data. However, as we shall see later, this assumption however has consequences. The term $P(S_1\&S_2\&...\&S_n)$ can be eliminated by normalisation, and therefore does not cause a problem. The inference equation we can obtain from Bayes' theorem is therefore:

$$P(D|S_1\&S_2..\&S_n) = \alpha P(D)P(S_1|D)P(S_2|D)\cdots P(S_n|D)$$

We can represent this equation graphically as shown in Figure 1. Variables (measured or hypothesised) are represented by circles and variables are joined to their parents by conditional probabilities. Returning to our problem of recognising cats using computer vision, we can express our knowledge about cats using the network shown in figure 2. This corresponds to the inference equation:

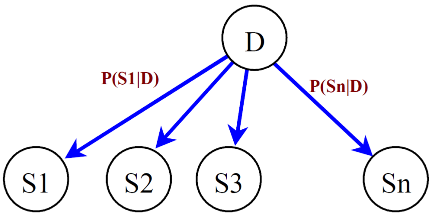
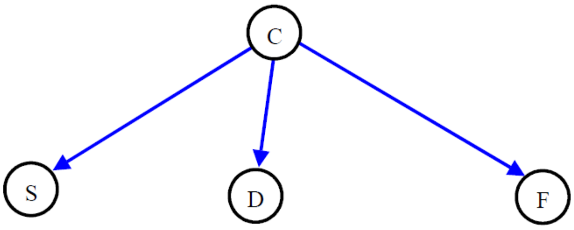


Figure 1: A naive Bayesian network

$$P(C|S\&D\&F) = \alpha P(C)P(S|C)P(D|C)P(F|C)$$

Our variables (hypothesis or evidence) fall into one of two categories, discrete and continuous. Discrete variables take one of a finite number fixed values or states. The states could be taken to be an integer number, or possible a range of values. Continuous variables can take any value within some range, and can be treated as real numbers. For the most part we will be dealing with discrete variables, though continuous variables can also be incorporated in Bayesian Networks. The measures that we developed in the last lecture are good examples of different variable types. If we wanted to make an estimate of the fur colour around the putative eyes, we could simply take a histogram of hue values of pixels in a small area. This would be a discrete variable. On the other hand the separation of the eyes is a continuous variable (although we could only measure it to pixel precision). If we change the formula slightly by removing the ”mod” and allowing positive and negative values (as shown in Figure 2), we have a measure might vary plausibly from -1.5 (eyes very close) to 1.5 (eyes very far apart).

We could divide it into any number of states, but for ease of data handling it is preferable to keep the number of states small. We might adopt seven states, with good resolution close to zero, as follows:



Variable	Interpretation	Type	Value
C	Cat	Discrete (2 states)	True or False
S	Separation of the eyes	Continuous	$S = (S_i - 2 * (R_l + R_r))/R_r$
D	Difference in eye size	Continuous	$\|(R_l - R_r)/R_r\|$
F	Fur colour	Discrete (20 states)	Coarse histogram of pixel hues

Figure 2: A Bayesian network with discrete and continuous variables

$$[below - 1.5][-1.5 \cdots - 0.75][-0.75 \cdots - 0.25][-0.25 \cdots 0.25][0.25 \cdots 0.75][0.75 \cdots 1.5][above 1.5]$$

We can quantise variables in a large number of ways, and indeed this forms an important area of research. However, we wont give an extensive treatment of that here, but just note that every variable can be made discrete in a reasonable way to fit the application, and we will assume that the difference in eye size can be similarly quantised into four discrete states.

Each arc in a simple network is represented by a matrix, called a link matrix (or conditional probability matrix). The link matrix that joins node D to node C contains a conditional probability for every pair of states.

$$\boldsymbol{P}(\boldsymbol{D}|\boldsymbol{C}) = \begin{bmatrix} P(d_1|c_1) & P(d_1|c_2) \\ P(d_2|c_1) & P(d_2|c_2) \\ P(d_3|c_1) & P(d_3|c_2) \\ P(d_4|c_1) & P(d_4|c_2) \end{bmatrix}$$

Note that matrix notation (bold) $\boldsymbol{P}(\boldsymbol{D}|\boldsymbol{C})$ should not be confused with the scalar value implied by $P(D|C)$ which we use, for example, in Bayes' theorem. Notice also that each column is a probability distribution and therefore sums to 1. We can find the values of the conditional probabilities in the link matrices from a typical data set. To do this we need a large number of cases in which we know the values of all the variables. This can be supplied by processing real pictures for the leaf nodes, S, D and F, and getting expert advice on the state of C. The link matrices derived in this way are objective probabilities. If we process a large number of images, and we find that c_2 (cat in image) occurs in $N(c_2)$ images and both c_2 and d_4 occur in $N(c_2\&d_4)$ images, then we write

$$P(d_4|c_2) = N(c_2\&d_4)/N(c_2)$$

Generally there will be a large number of conditional probabilities. In our simple example there are 62. We therefore need a very large data set to make a reasonable estimate. Notice that the use of the network gives us a much more accurate way of expressing how each term in the catness measure relates to the presence of a cat. Networks of the sort we have considered so far are referred to by a number of names:

Bayesian Classifier
Naive Bayesian Network
Simple Bayesian Network

They are in many ways the most useful form of network and should be used wherever possible.

Instantiation

Instantiation means setting the value of a node. To make an inference with the simple network of figure 2, we instantiate variables S, D and F by using the measurements from the image and the quantisation rules that we defined. We then look up the conditional probability values for a state of C, from the link matrices and multiply them together. When we have done this for each state of C we multiply by the prior probability and normalise the results so that they sum to 1 to get the probability of a cat, given that data.

Tree Structured Classifiers

Reasoning about our variables we could argue that, given there was a cat in the picture, the separation and the difference variables might not be completely independent. In particular, we could argue that S and D might be linked as variables indicating eyes.

Thus we might refine our network into a more complex structure as shown in figure 3. The new structure gives us a better model since it includes eyes as a semantic entity, which might be present but not caused by a cat. The node E (eyes) can be seen as a common cause of the separation and difference nodes, getting round the problem that they may not be independent variables. In adding a new node we have to decide how many states it has. It could be simply binary (true or false), but for better generality we could have three states in this particular case:

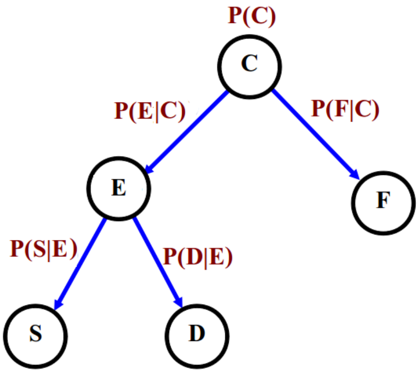


Figure 3: A Bayesian classifier

e_1 interpreted as probably not eyes
e_2 interpreted as could be eyes
e_3 interpreted as probably eyes

The link matrices are found as before, but this time we need expert advice on both the non terminal node E and the hypothesis node C. To analyse the network using Bayes' theorem, we can begin with the eyes node.

$$P(E|S\&D) = \alpha P(E)P(S|E)P(D|E)$$

Immediately we have a problem, since we do not have a direct estimate for $P(E)$, the prior probability of the eyes. E is an intermediate variable that we compute and we are not measuring it. However, we can still calculate a likelihood value of E. A likelihood value can be thought of as a probability based on measured values alone, ignoring any prior information. Given some values for S and D. We can write:

$$L(E|S\&D) = P(S|E)P(D|E)$$

Liklihoods do not normally have the property of probability distribution that they sum to 1. If we choose to normalise them, we can do so as before in which case the Likelihood becomes a probability distribution over the states of E calculated under the assumption that the prior probability of each state of E is equal. $P(e_1) = P(e_2) = P(e_3) = 1/3$. Now we can turn to the root node.

$$P(C|E\&F) = \alpha P(C)P(E|C)P(F|C)$$

If we have a measurement for F, say $F = f_5$, then we can look up $P(F|C)$ from the link matrix. However we dont have a value for the state of E, just and estimate of the likelihood of each state of $E(L(E))$. In order to estimate $P(E|C)$ we take an average of the link matrix entries weighted according to this distribution. We can do this as follows:

$$P(e|c_1) = P(e_1|c_1)L(e_1) + P(e_2|c_1)L(e_2) + P(e_3|c_1)L(e_3)$$
$$P(e|c_2) = P(e_1|c_2)L(e_1) + P(e_2|c_2)L(e_2) + P(e_3|c_2)L(e_3)$$

So finally we can calculate the probability distribution over C

$$P'(c_1) = P(c_1|e\&f_5) = \alpha P(c_1)\{P(e_1|c_1)L(e_1) + P(e_2|c_1)L(e_2) + P(e_3|c_1)L(e_3)\}P(f_5|c_1)$$
$$P'(c_2) = P(c_2|e\&f_5) = \alpha P(c_2)\{P(e_1|c_2)L(e_1) + P(e_2|c_2)L(e_2) + P(e_3|c_2)L(e_3)\}P(f_5|c_2)$$

We calculate α by normalisation as before. We use P' to mean the posterior probability, that is the probability of a variable given whatever information is known (in this case the values of F, S and D)

Although we don't have a prior probability for node E, it is still possible to estimate one from our knowledge of the evidence for C, and the link matrix $P(E|C)$. The evidence for C can be divided into two parts:

1. Evidence coming from E and its sub-tree
2. Evidence coming from everywhere else.

We only use the second type of evidence to estimate a prior of E. Let:

$$P_E(C) = \alpha P(C)P(F|C)$$

be the probability of C given the evidence from everywhere except E and its subtree. Then we can estimate a prior for E using the vector equation:

$$\boldsymbol{P(E) = P(E|C)P_E(C)}$$

Note that this is a vector equation, not a scalar equation like all the others used so far. Vectors and matrices will be shown in bold face. To clarify the notation we write $\boldsymbol{P(E|C)}$ for the link matrix, $P(e_1|c_2)$ for a specific scalar value taken from the link matrix and $P(E|C)$ to indicate a scalar entry in the link matrix parameterised by variables E and C. Assuming that $\boldsymbol{P_E(C)}$ is $[0.4, 0.6]$, the equation expands to:

$$\begin{bmatrix} P(e_1) \\ P(e_2) \\ P(e_3) \end{bmatrix} = \begin{bmatrix} P(e_1|c_1) & P(e_1|c_2) \\ P(e_2|c_1) & P(e_2|c_2) \\ P(e_3|c_1) & P(e_3|c_2) \end{bmatrix} \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix} = \begin{bmatrix} 0.4P(e_1|c_1) + 0.6P(e_1|c_2) \\ 0.4P(e_2|c_1) + 0.6P(e_2|c_2) \\ 0.4P(e_3|c_1) + 0.6P(e_3|c_2) \end{bmatrix}$$

Notice that just as the columns of the link matrices sum to 1, so does the calculated value for $\boldsymbol{P(E)}$.

Now, for a given set of measurements, say $[s_3, d_2]$ we can compute a posterior probability distribution over the states of E:

$$P'(e_1) = \alpha P(e_1)P(s_3|e_1)P(d_2|e_1)$$
$$P'(e_2) = \alpha P(e_2)P(s_3|e_2)P(d_2|e_2)$$
$$P'(e_3) = \alpha P(e_3)P(s_3|e_3)P(d_2|e_3)$$

and using

$$P'(e_1) + P'(e_2) + P'(e_3) = 1$$

we can eliminate α.

All this looks very cumbersome, and is going to become intractable in a large network, so next time we will look at a systematic way of calculating probabilities from which we can develop algorithmic methods.