# Lecture 5: Building networks from data

## Expert knowledge

Having established a way to make inferences with Bayesian networks, we now turn to the question of how to obtain the correct network structure for a given application. In previous lectures we assumed that the network structure would be known, or readily obtainable from an expert (or possibly from common sense). We then used the available data to find the conditional probabilities in the link matrices. Thus the network structure is highly subjective, and reliant on the expert advice that is available. The alternative, purely objective, approach is to use data to build the network.

## Spanning Tree Algorithms

The idea of a spanning tree is to start with a set of nodes to which we add arcs until a complete network is created. We use the fact that a Bayesian Network expresses the dependencies and independencies of a set of variables. Those joined by arcs are dependent, those not are at least conditionally independent. Thus our strategy in adding arcs is to connect the variables that are most dependent. For this we need a dependency measure. Right at the start of the course we noted that for two variables $A$ and $B$:

$P(A\&B) = P(A)P(B)$ if they are independent;

$P(A\&B) = P(A)P(B|A)$ if they have some dependency.

A conditional probability, such as $P(B|A)$, may be smaller or larger than the unconditional probability ($P(B)$). In the limit if $P(B)$ = $P(B|A)$ then $A$ and $B$ are independent. This leads us immediately to a dependency measure:

$$Dep(A, B) = |P(A\&B) - P(A)P(B)|$$

If the depencency between A and B is large then we may conclude that they should be joined by an arc in the network. Intuitively we choose to join the arcs with the largest dependencies which leads to the outline spanning tree algorithm

1. For every pair of variables calculate the dependency: $Dep(A, B)$;
2. Join the nodes in dependency order, providing the resulting structure has no loops.

Lets take a closer look at how we calculate the dependency measure. Firstly, since each variable has many states we need to sum the measure over the states:

$$Dep(A, B) = \sum_{A\times B} |P(a_i\&b_j) - P(a_i)P(b_j)|$$

To find the values of $P(a_i\&b_j)$ from our data set, we first construct a co-occurrence matrix. For example:

| | | A | | | |
|---|---|---|---|---|---|
| | | $a_1$ | $a_2$ | $a_3$ | $a_4$ |
| B | $b_1$ | 5 | 2 | 0 | 4 |
| | $b_2$ | 7 | 0 | 0 | 1 |
| | $b_3$ | 3 | 12 | 0 | 0 |
| | $b_4$ | 0 | 4 | 6 | 0 |
| | $b_5$ | 4 | 2 | 5 | 1 |
| | $b_6$ | 6 | 7 | 3 | 2 |

$[a_2, b_3]$ occurs 12 times in the data set

This is converted into probabilities by dividing by the number of data points, which, in the case of the above example is 74. The resulting matrix is the joint probability distribution over nodes $A$ and $B$, and the probability distributions $P(A)$ and $P(B)$ can be found by summing the columns and rows. This is the process known as marginalisation. A common notation for marginalisation is shown in the figure below. For example $\sum_B P(A\&B)$ means: for each state of $A$ sum over the states of $B$.

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $\Sigma_A$ |
|---|---|---|---|---|---|
| $b_1$ | 0.07 | 0.03 | 0.00 | 0.05 | 0.15 |
| $b_2$ | 0.09 | 0.00 | 0.00 | 0.01 | 0.11 |
| $b_3$ | 0.04 | 0.16 | 0.00 | 0.00 | 0.20 |
| $b_4$ | 0.00 | 0.05 | 0.08 | 0.00 | 0.14 |
| $b_5$ | 0.05 | 0.03 | 0.07 | 0.01 | 0.16 |
| $b_6$ | 0.08 | 0.09 | 0.04 | 0.03 | 0.24 |
| $\Sigma_B$ | 0.33 | 0.36 | 0.19 | 0.10 | |

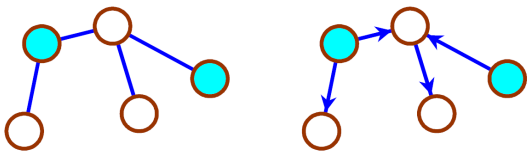
Summing the rows gives the probability distribution over B: P(B)

Joint Probability distribution P(A&B)

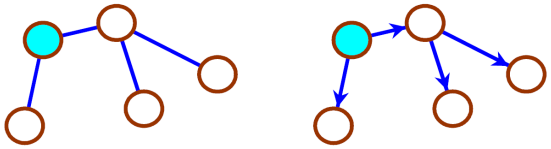
Summing the columns gives the probability distribution over A: P(A)

## Adding causal directions

Computing the dependency measure is now straightforward, and we can write a spanning algorithm but it does not give us any causal information. For cases with one hypothesis node we assume that the hypothesis is the cause and point all the arrows away from it. Similarly if we know that two nodes are to be considered root nodes then we propagate the arrows from them. This may result in a singly connected network (with some nodes having multiple parents) rather than a simple decision tree. There are methods of determining causal directions from the data, which we will look at next lecture, but these are not very effective. On the whole cause is a semantic entity that needs human intervention to determine.



(a) Choosing one root creates a decision tree



(b) Choosing two roots creates a singly connected network

Having now found a network with causal information, the same data can be used directly to find the conditional probabilities just as we discussed in the previous lectures. It is a matter of counting the frequency of occurrences of different states in the data.

$$P(ai|bj) = \frac{\text{Occurrences of } [a_i, b_j, \cdots]}{\text{Occurrences of } [\ ., b_j, \cdots]}$$

An alternative method of finding the conditional probability matrix is to make use of the joint probability matrix that we calculated in order to find the dependency. If we adopt the convetion that $P(B|A)$ has columns for each state of $A$ and rows for each state of $B$, then we start with the joint probability matrix $P(A\&B)$ written in the same format, and normalise each column so that it sums to 1. As we saw above marginalising over $B$ is achieved by summming each column. If we now divide the column by its sum we are in effect using the equation:

$$P(B|A) = P(B\&A)/P(A)$$

## Other measures of dependency

The measure we defined and used is sometimes called the un-weighted L1 metric. Notice that as the probabilities become small they contribute less to the dependency, and this effect is acceptable since we have little information on rare events. Another metric, which further reduced the dependency for low probability values is the weighted L1 metric:

$$Dep(A,B) = \sum_{A\times B} P(a_i\&b_j) \times |P(a_i\&b_j) - P(a_i)P(b_j)|$$

A very similar measure, which is more generally used, is the L2 metric:

$$Dep(A,B) = \sum_{A\times B} (P(a_i\&b_j) - P(a_i)P(b_j))^2$$

and it also has a weighted form:

$$Dep(A,B) = \sum_{A\times B} P(a_i \& b_j) \times (P(a_i \& b_j) - P(a_i)P(b_j))^2$$

A different measure, which is the most widely used is called mutual entropy, but it also goes under a variety of other names including mutual information, co-entropy, and the Kullback-Leibler divergence.

$$Dep(A,B) = \sum_{A\times B} P(a_i \& b_j) log_2((P(a_i \& b_j)/(P(a_i)P(b_j)))$$

## Properties of Mutual Entropy

Mutual entropy has three properties which make it an attractive measure for dependency between variables.

1. It is zero when two variables are completely independent.
2. It is positive and increasing with dependency when applied to probability distributions.
3. It is independent of the actual value of the probability.

We will give an illustration of mutual entropy through a simple example. Given three variables $A$, $B$ and $C$, and four data points:

$$[a_1, b_1, c_1][a_2, b_2, c_1][a_2, b_2, c_2][a_1, b_1, c_2]$$

We can calculate the joint probability matrix for A and B as:

| | $a_1$ | $a_2$ | $P(B)$ |
|---|---|---|---|
| $b_1$ | 0.5 | 0 | 0.5 |
| $b_2$ | 0 | 0.5 | 0.5 |
| $P(A)$ | 0.5 | 0,5 | |

We see that $A$ and $B$ are exactly correlated and therefore totally dependent

$$\sum_{A\times B} P(a_i \& b_j) log_2((P(a_i \& b_j)/(P(a_i)P(b_j))) = 0.5 log_2 2 + 0.5 log_2 2 = 1$$

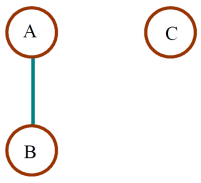
Now consider A and C, we extract the joint distribution from the data and get:

| | $a_1$ | $a_2$ | $P(C)$ |
|---|---|---|---|
| $c_1$ | 0.25 | 0.25 | 0.5 |
| $c_2$ | 0.25 | 0.25 | 0.5 |
| $P(A)$ | 0.5 | 0,5 | |

Looking at the matrix $P(C|A)$ we see that that A and C are completely independent. The mutual information confirms this:

$$\sum_{A\times B} P(a_i \& b_j) log_2((P(a_i \& b_j)/(P(a_i)P(b_j))) = 0.25 log_2 1 + 0.25 log_2 1 + 0.25 log_2 1 + 0.25 log_2 1 = 0$$

B and C similarly have no dependency, and the resulting network is simply:

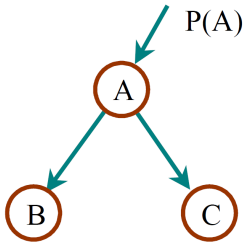



In general the Kullback-Leibler divergence is an appropriate measure for comparing two probability distributions (in the case above the two are $P(A \& B)$ and $P(A)P(B)$). A justification for the spanning tree algorithm is that it minimises the divergence between the network joint probability and the data joint probability.

$$I(P_N, P_D) = \sum_X P_D(X) log_2[P_D(X)/P_N(X)]$$

Any other network with the same nodes is further from the data distribution.

*Joint Probability Distribution of a Bayesian Network*
The joint probability of any Bayesian network is the product of the conditional probabilities, given the parents, with the prior probabilities of the roots. For a simple case:




$$P(A\&B\&C) = P(A) * P(B\&C|A)$$

And since B and C are conditionally independent given A

$$P(A\&B\&C) = P(A) * P(B|A) * P(C|A)$$

*Joint probability distribution of real data*
Given a data set of 3 variables with $N$ data points, the joint probability distribution of the data is found by frequency. If we choose any one combination of states $[a_i, b_j, c_k]$ we can calculate:

$$P(ai, bj, ck) = (\text{Number of Occurrences of } [a_i, b_j, c_k])/N$$

## Correlation

A more intuitive (to anybody who has studied statistics) measure of dependency is correlation. A correlation coefficient varies between -1 and 1, with the meaning that 0 means there is no correlation between the variables, 1 means there is an exact linear dependency between them, and -1 means that there is an inverse linear dependency between them. Correlation measures only "linear" dependency, whereas mutual information can characterise higher order dependencies more accurately. To use it we must be able to represent our states by integers. We can do this for example in cases where the states were created by quantising continuous variables Suppose for example the data set in the previous example:

$$[a_1, b_1, c_1][a_2, b_2, c_1][a_2, b_2, c_2][a_1, b_1, c_2]$$

becomes:

$$[1, 1, 1][2, 2, 1][2, 2, 2][1, 1, 2]$$

given that the variance of $A$ is defined as:

$$\sigma_A = \sum_{i=1}^{N} (\overline{a} - a_i)^2/(N-1)$$

and the covariance of A and B is defined as:

$$\Sigma_{AB} = \sum_{i=1}^{N} (\overline{a} - a_i)(\overline{b} - b_i)/(N-1)$$

The correlation between $A$ and $B$ is defined as:

$$C(A, B) = \Sigma_{AB}/\sqrt{\sigma_A \sigma_B}$$

Applying this to the data set above we have that:

$$\overline{a} = \overline{b} = \overline{c} = 1.5$$
$$\sigma_a = \sigma_b = \sigma_c = 1/3$$
$$\Sigma_{AB} = (0.25 + 0.25 + 0.25 + 0.25)/3 = 1/3$$
$$\Sigma_{AC} = (0.25 - 0.25 + 0.25 - 0.25)/3 = 0$$
$$\text{Dependence(A,B)} = |C(A, B)| = 1$$
$$\text{Dependence(A,C)} = |C(A, C)| = 0$$

So in this toy example it gives the same result as mutual entropy.