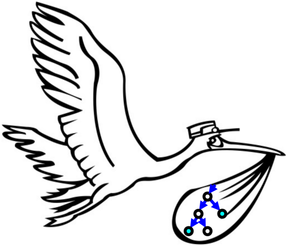
Lecture 5

Where do Bayesian Nets come from?



Building Networks from Data

Expert knowledge

- In previous lectures we used the following methodology:

  1. Consult an expert to obtain the structure
  2. Use available data to find the conditional probabilities in the link matrices

- This approach dating from the 1980s assumes a subjective structure, but objective parameters.
- However, should we trust expert opinion?

Expert giving advice to a cat

Drawing by Kliban

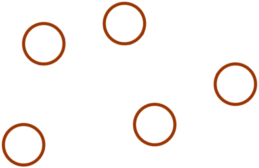
Expert giving advice to a cat

Drawing by Kliban

Perhaps an objective approach might be better.

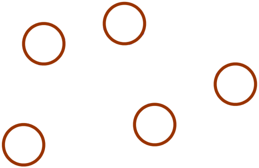
Spanning Tree Algorithm

Given a set of variables (nodes) and a data set:

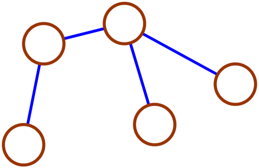


Spanning Tree Algorithm

Given a set of variables (nodes) and a data set:

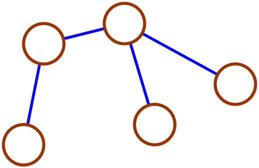


Find the most dependent variables and join them:



Spanning Tree Algorithm

Find the most dependent variables and join them:



Bayesain networks represent dependency. We expect that nodes joined by arcs have some dependency (though mathematically this need not be the case).

Here's the catch - Nodes that are not joined are assumed conditionally independent.

How can we measure dependency?

At the start of the course we observed that, if two variables are independent:

$$P(D\&S) = P(D) \times P(S)$$

but if they are dependent in any way:

$$P(D\&S) = P(D) \times P(S|D)$$

Comparing $P(S\&D)$ with $P(S) \times P(D)$ is the basis of all dependency measures.

Dependency Measurement using an L1 metric

A joint probability, such as $P(B\&A)$, may be smaller or larger than the product of the individual probabilities $(P(B) \times P(B))$. For the simplest dependency measure we take the magnitude of the difference.

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

Intuitively we choose to join the arcs with the largest dependencies which leads to the maximum weighted spanning tree algorithm.

The Spanning Tree Algorithm

Assume we have a large data set over our variables, ie for discrete variables A, B, C and D we have a number of data points:

$$\begin{array}{cccc} a_1 & b_2 & c_1 & d_2 \\ a_2 & b_1 & c_1 & d_3 \\ a_1 & b_1 & c_2 & d_1 \\ a_1 & b_1 & c_3 & d_2 \end{array}$$

etc.

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

Summing over the joint states

In order to compute:

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

we need to sum over the joint states of the variables:

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

This means we need to compute the distributions $P(A\&B)$, $P(A)$ and $P(B)$ from the data set.

Computing $P(A\&B)$

From the data set we first find the co-occurence matrix:

$$\begin{array}{cc|cccc} & & & \text{A} & & \\ & & a_1 & a_2 & a_3 & a_4 \\ \hline & b_1 & 5 & 2 & 0 & 4 \\ & b_2 & 7 & 0 & 0 & 1 \\ \text{B} & b_3 & 3 & 12 & 0 & 0 \\ & b_4 & 0 & 4 & 6 & 0 \\ & b_5 & 4 & 2 & 5 & 1 \\ & b_6 & 6 & 7 & 3 & 2 \end{array}$$

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

We convert this to probabilities by dividing by the number of data points. (74 in this example)

Computing $P(A)$ and $P(B)$ by marginalisation

	a_1	a_2	a_3	a_4	Σ_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
Σ_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)

Problem

Calculate the dependency between the variables A and B for the following two data sets:

1. Independent

a_1	b_1
a_1	b_2
a_2	b_1
a_2	b_2

2. Dependent

a_1	b_1
a_1	b_1
a_2	b_2
a_2	b_2

Solution: Independent

Co-occurences:

	a_1	a_2
b_1	1	1
b_2	1	1

Probabilities:

	a_1	a_2	Σ
b_1	0.25	0.25	0.5
b_2	0.25	0.25	0.5
Σ	0.5	0.5	

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

$$Dep(A,B) = 0$$

Solution: Dependent case

Co-occurences:

	a_1	a_2
b_1	2	0
b_2	0	2

Probabilities:

	a_1	a_2	Σ
b_1	0.5	0	0.5
b_2	0	0.5	0.5
Σ	0.5	0.5	

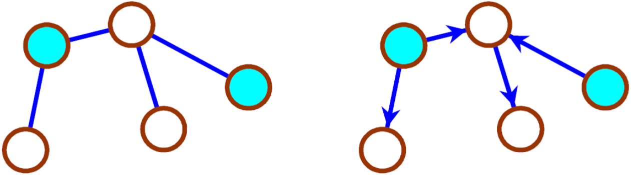
$$Dep(A,B) = \sum_{A \times B} |P(a_i \& b_j) - P(a_i)P(b_j)|$$

$$Dep(A,B) = 0.25 + 0.25 + 0.25 + 0.25 = 1$$

Nota Bene - dependencies can be bigger than 1.

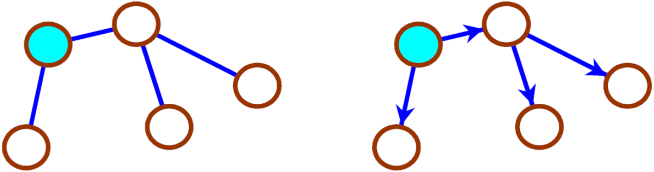
Causal Directions

The maximum weighted spanning tree algorithm does not give us any information on causal directions. One way to determine these is to assume that the root nodes represent fundamental causes and therefore all arrows propogate from them:



Causal Directions

If there is just one known root the resulting structure is a decision tree. If there are many roots then a singly connected network is created.



Causal Directions

- Deciding which nodes are roots normally requires expert intervention, and is therefore a subjective process.
- There are algorithms for estimating causal directions from data which we will discuss next lecture.

Computing the Conditional Probabilities

We can compute the conditional probabilities from the data as described in the first lecture:

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

However, we have already computed every pair of joint probability matrices. For a joint probability matrix we can compute the corresponding conditional probability matrix by normalising the columns to sum to 1, eg:

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

Other measures of dependency

- The measure we have used so far is an unweighted L1 metric.
- Its characteristic is that as the probabilities become small they contribute less to the dependency.
- This effect reflects the fact that we have little information on rare events

The Weighted L1 Metric

Another metric is formed by weighting the difference in magnitude by the joint probability:

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

The effect is to further reduce the contribution to the dependency measure where probabilities are low.

The L2 Metric

L2 metrics use the squared differences:

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

There is a weighted form:

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

Mutual Entropy

The most widely used measure in comparing probability distributions is Mutual Entropy. It is also called Mutual Information or the Kullback-Liebler divergence.

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

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

Justification of Mutual Entropy

The chief justification for using mutual entropy for the spanning tree algorithm is that it minimises the difference between the joint probability distribution calculated from the data and the joint probability of the network.

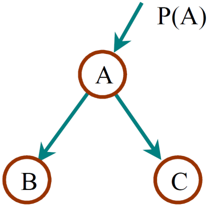
$$I(P_N, P_D) = \sum_X P_D(X) log_2[P_D(X)/P_N(X)]$$

Joint Probability distribution of a 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 three node tree:

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



Joint Probability distribution of a data set

Given a data set of 3 variables(A, B and C) with N data points, the joint probability distribution of the data is found by frequency.

For each possible combination of states $[a_i, b_j, c_k]$ we can calculate:

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

Example: Building a Spanning Tree

Given variables A, B and C and a data set:

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

The co-occurrence probability matrix for A and B is:

	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	

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

$$= 0.5 log_2 2 + 0.5 log_2 2 = 1$$

A and B are completely dependent.

Example: Building a Spanning Tree

Next Consider A and C:

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

The co-occurrence probability matrix for A and C is:

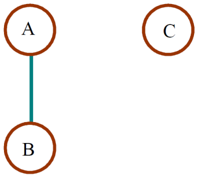
	a_1	a_2	$P(C)$
c_1	0.25	0.25	0.5
c_2	0.25	0.25	0.5
$P(A)$	0.5	0.5	

$$Dep(A, C) =$$
$$0.25 log_2 1 + 0.25 log_2 1 + 0.25 log_2 1 + 0.25 log_2 1 = 0$$

A and C are completely independent.

Example: Building a Spanning Tree

Similarly we find that B and C are completely independent, thus our spanning tree takes the form:



We need further information (expert advice) to determine the direction of the link between A and B.

Dependency and Correlation

An alternative dependency measure is correlation (more intuitive to anybody who has studied statistics).

However, correlation only measures linear dependency in a numerical context.

To use it we must represent our states by integers in a meaningful way. In effect we need an inverse quantization that maps our states to numeric values. This may not be an easy thing to do, but we could well have numerical (as opposed to state) data for a given application.

Dependency and Correlation

Consider the data set in the previous example:

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

Let us suppose that it can be mapped to numeric values as follows:

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

Dependency and Correlation

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:

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

Dependency and Correlation

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

Using correlation on the above data set we get:

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

In this toy example correlation gives the same result as mutual entropy.

Correlation vs Mutual Information

Correlation only identifies linear relationships

Mutual information measures dependency and so it is more general in encapsulating non-linear relationships.

However it does not tell us anything about how the variables are related