Data Mining 2013
Bayesian Networks (2)

Ad Feelders

Universiteit Utrecht

October 22, 2013

Learning Bayesian Networks: Overview

- structure known, complete data (done)
- structure unknown, complete data (today)
- structure known, incomplete data (today)
- structure unknown, incomplete data (beyond the scope)

Structure Known, Complete Data

The loglikelihood function for a Bayesian Network is:

$$\mathcal{L} = \sum_{i=1}^{k} \sum_{x_i, x_{pa(i)}} n(x_i, x_{pa(i)}) \log p(x_i \mid x_{pa(i)})$$

The maximum likelihood estimate of $p(x_i \mid x_{pa(i)})$ is:

$$\hat{p}(x_i \mid x_{pa(i)}) = \frac{n(x_i, x_{pa(i)})}{n(x_{pa(i)})},$$

where where $n(x_{pa(i)})$ is the number of observations (rows) with parent configuration $x_{pa(i)}$, and $n(x_i, x_{pa(i)})$ is the number of observations with parent configuration $x_{pa(i)}$ and value x_i for variable X_i.

Maximized Loglikelihood

The value of the loglikelihood function evaluated at its maximum therefore is (fill in the maximum likelihood estimates in the loglikelihood function):

$$\mathcal{L} = \sum_{i=1}^{k} \sum_{x_i, x_{pa(i)}} n(x_i, x_{pa(i)}) \log \frac{n(x_i, x_{pa(i)})}{n(x_{pa(i)})}$$

- The higher this value, the better the model fits the data.
- The saturated model (complete graph) always has the highest loglikelihood score.
- To avoid overfitting, we must penalize model complexity.

Structure Unknown, Complete Data

Scoring functions:

- $\text{AIC}(M) = \mathcal{L}^M - \dim(M)$.
- $\text{BIC}(M) = \mathcal{L}^M - \frac{\log n}{2}\dim(M)$.

where $\mathcal{L}^M$ is the maximized value of the loglikelihood function for model M and $\dim(M)$ is the number of parameters in the model.

BIC gives a higher penalty for model complexity ($n > 7$), so tends to lead to less complex models than AIC.

Note: earlier we defined $AIC(M) = 2(\mathcal{L}^{\text{sat}} - \mathcal{L}^M) + 2\dim(M)$. Dividing by -2 and ignoring the constant $\mathcal{L}^{\text{sat}}$ gives the current definition.

Optimization Problem

Given

- Training data.
- Scoring function (BIC or AIC).
- Space of possible models (all DAGs).

find the model that maximizes the score.

- Most model search algorithms do not require an a priori ordering of the variables!
- The number of labeled acyclic directed graphs on k nodes is given by the recurrence

$$a_k = \sum_{j=1}^{k} (-1)^{j-1} \binom{k}{j} 2^{j(k-j)} a_{k-j}$$

For example, $a_6 = 3,781,503$.

Heuristic Search

- Define which models are neighbors of a given model (typically: addition, removal, or reversal of an arc).
- Traverse search space looking for high-scoring models, e.g. by greedy hill-climbing.

Score Decomposes

The loglikelihood score

$$\mathcal{L} = \sum_{i=1}^{k} \sum_{x_i, x_{pa(i)}} n(x_i, x_{pa(i)}) \log \frac{n(x_i, x_{pa(i)})}{n(x_{pa(i)})}$$

must be computed many times for different models in structure learning.

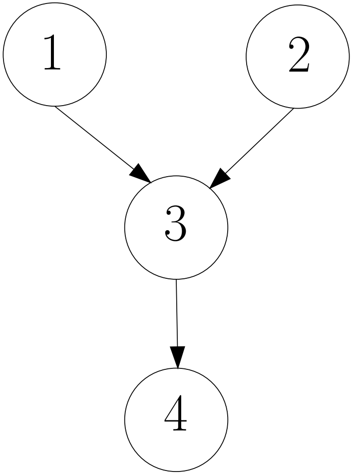
Luckily, it is a sum of terms, where each term contains the variables $\{i\} \cup \text{pa}(i)$.

Hence, when making a change to the model, we only have to recompute the score for those variables for which the parent set has changed!

Example Data Set

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

Score this model



Relevant Data For Scoring Node 1

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

Score node 1 $= 5 \log \frac{5}{10} + 5 \log \frac{5}{10}$

Relevant Data For Scoring Node 2

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

Score node 2 $= 6 \log \frac{6}{10} + 4 \log \frac{4}{10}$

Relevant Data For Scoring Node 3

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

Score node 3 $= 2\log\frac{2}{3} + \log\frac{1}{3}$

Relevant Data For Scoring Node 3

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

Score node 3 $= 2\log\frac{2}{3} + \log\frac{1}{3} + 2\log 1$

Relevant Data For Scoring Node 3

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

Score node 3 $= 2\log\frac{2}{3} + \log\frac{1}{3} + 2\log 1 + \log\frac{1}{3} + 2\log\frac{2}{3}$

Relevant Data For Estimating Scoring Node 3

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

$$\text{Score node 3} = 2\log\frac{2}{3} + \log\frac{1}{3} + 2\log 1 + \log\frac{1}{3} + 2\log\frac{2}{3} + \log\frac{1}{2} + \log\frac{1}{2}$$

Relevant Data For Scoring Node 4

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

Score node 4 $= 2\log\frac{2}{4} + \log\frac{1}{4} + \log\frac{1}{4}$

Relevant Data For Scoring Node 4

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

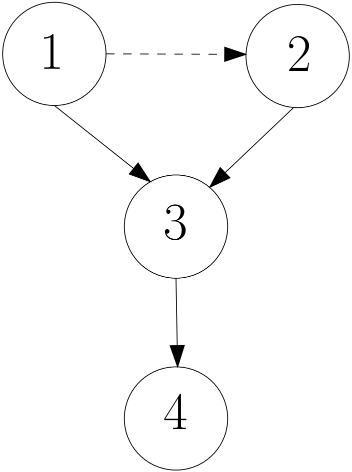
Score node 4 $= 2\log\frac{2}{4} + \log\frac{1}{4} + \log\frac{1}{4} + 2\log\frac{2}{6} + \log\frac{1}{6} + 3\log\frac{3}{6}$

Total Score

Summing the likelihood score over all nodes, we get:

$$
\begin{aligned}
\mathcal{L} = {} & 5 \log \frac{5}{10} + 5 \log \frac{5}{10} + 6 \log \frac{6}{10} + 4 \log \frac{4}{10} \\
& + 2 \log \frac{2}{3} + \log \frac{1}{3} + 2 \log 1 + \log \frac{1}{3} + 2 \log \frac{2}{3} \\
& + \log \frac{1}{2} + \log \frac{1}{2} + 2 \log \frac{2}{4} + \log \frac{1}{4} + \log \frac{1}{4} \\
& + 2 \log \frac{2}{6} + \log \frac{1}{6} + 3 \log \frac{3}{6} \approx -29.09
\end{aligned}
$$

Add an edge from X_1 to X_2



Score is Decomposable

$$
\begin{aligned}
\mathcal{L} \;=\;& 5\log\frac{5}{10} + 5\log\frac{5}{10} + \boxed{6\log\tfrac{6}{10} + 4\log\tfrac{4}{10}} \\
&+2\log\frac{2}{3} + \log\frac{1}{3} \\
&+2\log 1 + \log\frac{1}{3} + 2\log\frac{2}{3} \\
&+\log\frac{1}{2} + \log\frac{1}{2} \\
&+2\log\frac{2}{4} + \log\frac{1}{4} + \log\frac{1}{4} \\
&+2\log\frac{2}{6} + \log\frac{1}{6} + 3\log\frac{3}{6} \approx -29.09
\end{aligned}
$$

- When we add an edge from X_1 to X_2, only the parent set of node 2 changes.
- Therefore, only the score of node 2 (the boxed part) has to be recomputed.

Relevant Data For Re-scoring Node 2

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

New score node 2 $= 3 \log \frac{3}{5} + 2 \log \frac{2}{5}$

Relevant Data For Re-scoring Node 2

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

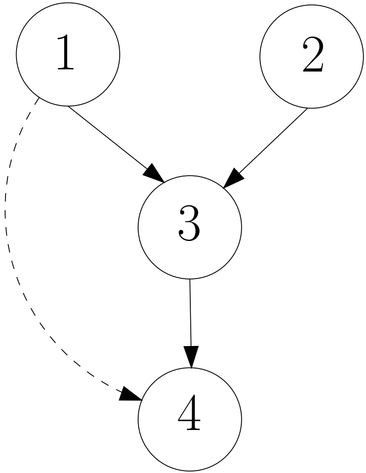
New score node 2 $= 3\log\frac{3}{5} + 2\log\frac{2}{5} + 3\log\frac{3}{5} + 2\log\frac{2}{5}$

Score Decomposes

$$
\begin{aligned}
\mathcal{L} = {} & 5 \log \frac{5}{10} + 5 \log \frac{5}{10} + \boxed{3 \log \tfrac{3}{5} + 2 \log \tfrac{2}{5} + 3 \log \tfrac{3}{5} + 2 \log \tfrac{2}{5}} \\
& + 2 \log \frac{2}{3} + \log \frac{1}{3} \\
& + 2 \log 1 + \log \frac{1}{3} + 2 \log \frac{2}{3} \\
& + \log \frac{1}{2} + \log \frac{1}{2} \\
& + 2 \log \frac{2}{4} + \log \frac{1}{4} + \log \frac{1}{4} \\
& + 2 \log \frac{2}{6} + \log \frac{1}{6} + 3 \log \frac{3}{6} \approx -29.09
\end{aligned}
$$

The boxed part is the new contribution of node 2 to the score.

Add an edge from X_1 to X_4



Score Decomposes

$$
\begin{aligned}
\mathcal{L} \quad = \quad & 5\log\frac{5}{10} + 5\log\frac{5}{10} + 6\log\frac{6}{10} + 4\log\frac{4}{10} \\
& + 2\log\frac{2}{3} + \log\frac{1}{3} \\
& + 2\log 1 + \log\frac{1}{3} + 2\log\frac{2}{3} \\
& + \log\frac{1}{2} + \log\frac{1}{2} \\
& + \boxed{2\log\tfrac{2}{4} + \log\tfrac{1}{4} + \log\tfrac{1}{4}} \\
& + \boxed{2\log\tfrac{2}{6} + \log\tfrac{1}{6} + 3\log\tfrac{3}{6}} \approx -29.09
\end{aligned}
$$

- When we add an edge from X_1 to X_4, only the parent set of node 4 changes.
- Therefore, only the score of node 4 (the boxed part) has to be recomputed.

Relevant Data For Re-scoring Node 4

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

New score node 4 $= 2 \log 1$

Relevant Data For Re-scoring Node 4

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

New score node 4 $= 2 \log 1 + 2 \log \frac{2}{3} + \log \frac{1}{3}$

Relevant Data For Re-scoring Node 4

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

New score node 4 $= 2\log 1 + 2\log\frac{2}{3} + \log\frac{1}{3} + \log\frac{1}{2} + \log\frac{1}{2}$

Relevant Data For Re-scoring Node 4

obs	X_1	X_2	X_3	X_4
1	1	1	1	1
2	1	1	1	1
3	1	1	2	1
4	1	2	2	1
5	1	2	2	2
6	2	1	1	2
7	2	1	2	3
8	2	1	2	3
9	2	2	2	3
10	2	2	1	3

New score node 4 $= 2\log 1 + 2\log\frac{2}{3} + \log\frac{1}{3} + \log\frac{1}{2} + \log\frac{1}{2} + 3\log 1$

Score Decomposes

$$
\begin{aligned}
\mathcal{L} \quad = \quad & 5\log\frac{5}{10} + 5\log\frac{5}{10} + 6\log\frac{6}{10} + 4\log\frac{4}{10} \\
& +2\log\frac{2}{3} + \log\frac{1}{3} \\
& +2\log 1 + \log\frac{1}{3} + 2\log\frac{2}{3} \\
& +\log\frac{1}{2} + \log\frac{1}{2} \\
& +\boxed{2\log 1 + 2\log\tfrac{2}{3} + \log\tfrac{1}{3}} \\
& +\boxed{\log\tfrac{1}{2} + \log\tfrac{1}{2} + 3\log 1} \approx -22.16
\end{aligned}
$$

The boxed part is the new contribution of node 4 to the score.

Counting Parameters

The number of parameters of a Bayesian network is:

$$\sum_{i=1}^{k}(d_i - 1) \prod_{j \in pa(i)} d_j$$

where k is the number of variables in the network, and d_i is the number of possible values of X_i.

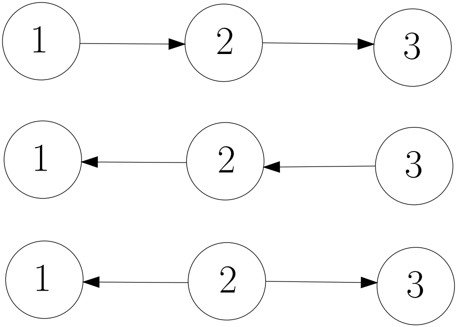
If X_i has no parents, the number of parent configurations should be taken to be 1, so it contributes $d_i - 1$ parameters.

A Simple Structure Learning Algorithm

Algorithm 1 BN Structure Learning

$G \leftarrow$ initial graph
max $\leftarrow$ score(G)
repeat
    nb $\leftarrow$ neighbours(G)
    for all $G' \in$ nb **do**
        if score $(G') >$ max **then**
            max $\leftarrow$ score(G')
            $G \leftarrow G'$
        end if
    end for
until no change to G
return G

Interpretation: warning!



These models can not be distinguished from data alone.
They represent the same independencies!

AIC and *BIC* give equivalent networks the same score.

Example Analysis

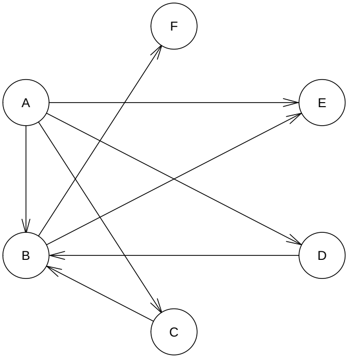
We analyze a data set concerning risk factors for coronary heart disease. For a sample of 1841 car-workers, the following information was recorded

Variable	Description
A	Does the person smoke?
B	Is the person's work strenuous mentally?
C	Is the person's work strenuous physically?
D	Systolic blood pressure < 140mm?
E	Ratio of beta to alfa lipoproteins < 3?
F	Is there a family history of coronary heart disease?

Example Analysis

For learning Bayesian networks, we use the *bnlearn* package in R.
Hill-climbing with the BIC score function:

```
> coronary.hc <- hc(coronary)
> plot(coronary.hc)
```



The Search Process

```
> coronary.hc <- hc(coronary, debug=T)
----------------------------------------------------------------
* starting from the following network:
  model:
   [A][B][C][D][E][F]

* current score: -7061.714
* caching score delta for arc A -> B (17.531166).
* caching score delta for arc A -> C (9.981480).
* caching score delta for arc A -> D (1.757126).
* caching score delta for arc A -> E (4.941129).
* caching score delta for arc A -> F (-3.224701).
* caching score delta for arc B -> C (264.272873).
* caching score delta for arc B -> D (2.313656).
* caching score delta for arc B -> E (21.030213).
* caching score delta for arc B -> F (2.303571).
* caching score delta for arc C -> D (-3.711314).
* caching score delta for arc C -> E (4.577177).
* caching score delta for arc C -> F (-3.673929).
* caching score delta for arc D -> E (2.645583).
* caching score delta for arc D -> F (-3.197133).
* caching score delta for arc E -> F (-2.257169).
```

The Search Process

- The initial model (the mutual independence model `[A] [B] [C] [D] [E] [F]`) has a BIC score of `−7061.714`.
- The output gives the *change* in score between the current model and its neighbors.
- Why is the score of only 15 of the 30 neighbors computed? (e.g. `A -> B`, but not `B -> A`)?

The Search Process

- The initial model (the mutual independence model `[A][B][C][D][E][F]`) has a BIC score of `-7061.714`.
- The output gives the *change* in score between the current model and its neighbors.
- Why is the score of only 15 of the 30 neighbors computed? (e.g. `A -> B`, but not `B -> A`)?
- The neighbors `A -> B` and `B -> A` are equivalent, and therefore have the same score.
- Adding `B -> C` causes the largest positive change in score so we move to that neighbor.

The Search Process

```
* best operation was: adding B -> C .
* current network is :

  model:
   [A][B][D][E][F][C|B]

* current score: -6797.441
* caching score delta for arc A -> C (9.975823).
* caching score delta for arc B -> C (-264.272873).
* caching score delta for arc D -> C (-1.472731).
* caching score delta for arc E -> C (-6.587044).
* caching score delta for arc F -> C (-6.059896).
```

The Search Process

- We don't have to recompute the change in score caused by, for example, adding `A -> B`, because the parent set of `B` is the same as in the previous iteration.
- Therefore, adding `A -> B` now will cause the same score change as in the previous iteration.
- Only the parent set of `C` has changed, so we only have to recompute the change in score caused by adding arcs `X -> C`.
- Adding `B -> E` causes the largest positive change in score so we move to that neighbor.
- The current model becomes: `[A][B][D][F][C|B][E|B]`.

Learning Bayesian Networks: Overview

- structure known, complete data (done)
- structure unknown, complete data (done)
- structure known, incomplete data (today)
- structure unknown, incomplete data (beyond the scope)

Structure Known, Incomplete Data

Suppose for observation j some variables are unobserved. We write

$$X^{(j)} = (X^{(j)}_{obs}, X^{(j)}_{mis}).$$

The marginal probability of the observed part of $X^{(j)}$ is obtained by *summing out* the missing part, i.e.:

$$P(X^{(j)}_{obs}) = \sum_{X^{(j)}_{mis}} P(X^{(j)}_{obs}, X^{(j)}_{mis})$$

Sum rule of probability: $P(X) = \sum_y P(X, Y)$.

Example 1

If we have three binary variables $X = (X_1, X_2, X_3)$, and we have an observation $X^{(j)} = (1, 0, ?)$, then $X_{obs}^{(j)} = (X_1, X_2)$ and $X_{mis}^{(j)} = (X_3)$.

The marginal probability of the observed part is obtained by summing over all possible values of the missing data:

$$P(1, 0, ?) = P(1, 0, 0) + P(1, 0, 1)$$

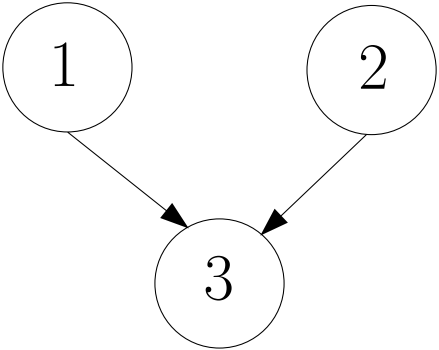
Example 2

If we have three binary variables $X = (X_1, X_2, X_3)$, and we have an observation $X^{(j)} = (?, 1, ?)$, then $X^{(j)}_{obs} = (X_2)$ and $X^{(j)}_{mis} = (X_1, X_3)$.

The marginal probability of the observed part is obtained by summing over all possible values of the missing data:

$$P(?, 1, ?) = P(0, 1, 0) + P(0, 1, 1) + P(1, 1, 0) + P(1, 1, 1)$$

A Simple Bayesian Network



This network corresponds to the factorisation:

$$P(X_1, X_2, X_3) = p_1(X_1)p_2(X_2)p_{3|12}(X_3|X_1, X_2)$$

Example 1 (easy)

According to this network, the probability of $(1, 0, ?)$ is

$$\begin{aligned} P(1, 0, ?) &= P(1, 0, 0) + P(1, 0, 1) \\ &= p_1(1)p_2(0)p_{3|12}(0|1, 0) + \\ &\quad\; p_1(1)p_2(0)p_{3|12}(1|1, 0) \\ &= p_1(1)p_2(0) \end{aligned}$$

since $p_{3|12}(0|1, 0) + p_{3|12}(1|1, 0) = 1$.

Example 1 (continued)

Suppose we observe the following data:

X_1	X_2	X_3	$n(X_1, X_2, X_3)$
0	0	0	10
0	0	1	40
1	0	0	20
1	0	1	20
0	1	0	20
0	1	1	30
1	1	0	10
1	1	1	90
0	0	?	10
0	1	?	10
1	0	?	40
1	1	?	0

Log-likelihood function

The corresponding log-likelihood function is:

$$\begin{aligned}
\mathcal{L} &= 120 \log p_1(0) + 180 \log(1 - p_1(0)) \\
&+ 140 \log p_2(0) + 160 \log(1 - p_2(0)) \\
&+ 10 \log p_{3|12}(0|0,0) + 40 \log(1 - p_{3|12}(0|0,0)) \\
&+ 20 \log p_{3|12}(0|1,0) + 20 \log(1 - p_{3|12}(0|1,0)) \\
&+ 20 \log p_{3|12}(0|0,1) + 30 \log(1 - p_{3|12}(0|0,1)) \\
&+ 10 \log p_{3|12}(0|1,1) + 90 \log(1 - p_{3|12}(0|1,1)).
\end{aligned}$$

Estimation of $p_{3|12}(0|0,0)$

$$\frac{\partial \mathcal{L}}{\partial p_{3|12}(0|0,0)} = \frac{10}{p_{3|12}(0|0,0)} - \frac{40}{1 - p_{3|12}(0|0,0)}$$

Equate to zero

$$\frac{10}{p_{3|12}(0|0,0)} = \frac{40}{1 - p_{3|12}(0|0,0)}$$

Solve for $p_{3|12}(0|0,0)$:

$$\hat{p}_{3|12}(0|0,0) = \frac{10}{50} = 0.2$$

The observations with X_3 missing are irrelevant to the estimation of this parameter.

Example 2 (hard)

Suppose however that we have an observation $(1, ?, 0)$. Its probability according to the network is:

$$\begin{aligned} P(1,?,0) &= P(1,0,0) + P(1,1,0) \\ &= p_1(1)p_2(0)p_{3|12}(0|1,0) \\ &+ p_1(1)p_2(1)p_{3|12}(0|1,1) \end{aligned}$$

- This expression can't be simplified.
- We get a sum of parameters inside the log, making analytical maximization impossible!

ML Estimation with Incomplete Data

Direct maximization of the observed data likelihood is complicated: in most cases there is no closed form solution of the ML estimates as in the complete data case.

There is however an ingenious iterative scheme to compute the ML estimates, called Expectation Maximization (EM).

EM for Bayesian Networks

Algorithm sketch:

1. Pick starting values $\hat{p}^{(0)}$ for parameters.

Repeat until convergence:

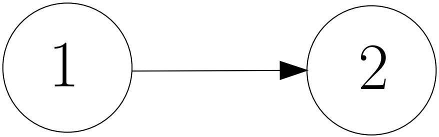
2. E-step: Compute expected value of sufficient statistics using $\hat{p}^{(t)}$ and observed data (inference in network required).
3. M-step: Compute $\hat{p}^{(t+1)}$ using the expected values of the sufficient statistics from the last E-step (closed form!).

$\hat{p}^{(0)}, \hat{p}^{(1)}, \ldots$ converges to a maximum likelihood estimate for the observed data likelihood.

The sufficient statistics are the counts from the data that are required to estimate the network parameters.

EM for Bayesian Networks: Example

Simple BN for EM example.



Corresponding factorisation:

$$P(X_1, X_2) = p(X_1)p(X_2|X_1)$$

EM for Bayesian Networks: Example

Initial values for the network parameters: $\hat{p}^{(0)}(X_1 = 1) = 0.8$, $\hat{p}^{(0)}(X_2 = 1|X_1 = 1) = 0.6$, $\hat{p}^{(0)}(X_2 = 1|X_1 = 0) = 0.2$.

This gives joint distribution:

x_1, x_2	$\hat{P}^{(0)}(x_1, x_2)$
(0,0)	$0.2 \times 0.8 = 0.16$
(0,1)	$0.2 \times 0.2 = 0.04$
(1,0)	$0.8 \times 0.4 = 0.32$
(1,1)	$0.8 \times 0.6 = 0.48$

EM for Bayesian Networks: Example

x_1, x_2	count		
(0,0)	12		
(0,1)	8		
(1,0)	20		
(1,1)	40		
(0,?)	2	$\hat{P}^{(0)}(X_2 = 0 \mid X_1 = 0) = 0.8$	$\hat{P}^{(0)}(X_2 = 1 \mid X_1 = 0) = 0.2$
(1,?)	8	$\hat{P}^{(0)}(X_2 = 0 \mid X_1 = 1) = 0.4$	$\hat{P}^{(0)}(X_2 = 1 \mid X_1 = 1) = 0.6$
(?,0)	6	$\hat{P}^{(0)}(X_1 = 0 \mid X_2 = 0) = 0.33$	$\hat{P}^{(0)}(X_1 = 1 \mid X_2 = 0) = 0.67$
(?,1)	4	$\hat{P}^{(0)}(X_1 = 0 \mid X_2 = 1) = 0.077$	$\hat{P}^{(0)}(X_1 = 1 \mid X_2 = 1) = 0.923$

For example:

$$\hat{P}^{(0)}(X_1 = 1 \mid X_2 = 0) = \frac{\hat{P}^{(0)}(X_1 = 1, X_2 = 0)}{\hat{P}^{(0)}(X_2 = 0)} = \frac{0.32}{0.32 + 0.16} = 0.67.$$

Expected Values of Sufficient Statistics

Sufficient statistics are the counts needed from the data to compute the parameter estimates. For example

$$
\begin{aligned}
\hat{n}_1(1) &= n(1,0) + n(1,1) + n(1,?) + \\
&+ n(?,0) \times \hat{P}(X_1 = 1 | X_2 = 0) + n(?,1) \times \hat{P}(X_1 = 1 | X_2 = 1)
\end{aligned}
$$

The expected values of the sufficient statistics are:

$$
\begin{aligned}
\hat{n}_1(1) &= 20 + 40 + 8 + 6 \times 0.67 + 4 \times 0.923 = 75.7 \\
\hat{n}_1(0) &= 100 - 75.7 = 24.3 \\
\hat{n}_{12}(0,0) &= 12 + 2 \times 0.8 + 6 \times 0.33 = 15.6 \\
\hat{n}_{12}(0,1) &= 24.3 - 15.6 = 8.7 \\
\hat{n}_{12}(1,0) &= 20 + 8 \times 0.4 + 6 \times 0.67 = 27.2 \\
\hat{n}_{12}(1,1) &= 75.7 - 27.2 = 48.5
\end{aligned}
$$

New Parameter Estimates

Using the expected values of the required counts, we have closed form estimates for the network parameters:

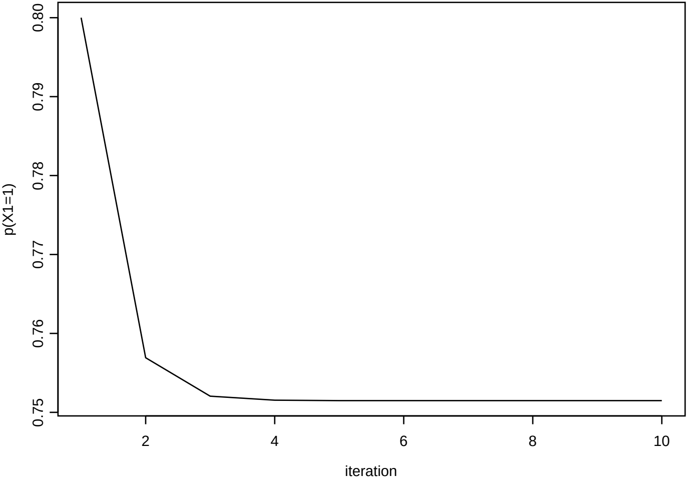
$$\begin{aligned}
\hat{p}^{(1)}(X_1 = 1) &= \frac{\hat{n}_1(1)}{n} = \frac{75.7}{100} \approx 0.76 \\
\hat{p}^{(1)}(X_2 = 1 | X_1 = 1) &= \frac{\hat{n}_{12}(1,1)}{\hat{n}_1(1)} = \frac{48.5}{75.7} \approx 0.64 \\
\hat{p}^{(1)}(X_2 = 1 | X_1 = 0) &= \frac{\hat{n}_{12}(0,1)}{\hat{n}_1(0)} = \frac{8.7}{24.3} \approx 0.36
\end{aligned}$$

New Joint Distribution

Based on these new parameter estimates, the new joint distribution becomes:

x_1, x_2	$\hat{P}^{(1)}(x_1, x_2)$
(0,0)	$0.24 \times 0.64 = 0.1536$
(0,1)	$0.24 \times 0.36 = 0.0864$
(1,0)	$0.76 \times 0.36 = 0.2736$
(1,1)	$0.76 \times 0.64 = 0.4864$

EM iterations for $\hat{p}(X_1 = 1)$



EM Pseudocode

EM(Data, Network Structure , $\varepsilon = 10^{-5}$)
$\hat{\mathbf{p}}^{(0)}$ = initial estimates of parameters
$t = 0$
Repeat
For all $x_i, x_{pa(i)}$ **do**

$$\hat{n}^{(t+1)}(x_i, x_{pa(i)}) = \sum_{j=1}^{n} \overbrace{P(X_i = x_i, X_{pa(i)} = x_{pa(i)} | X_{obs}^{(j)}, \hat{\mathbf{p}}^{(t)})}^{\text{Requires Inference in Network}}$$

$$\hat{n}^{(t+1)}(x_{pa(i)}) = \sum_{x_i} \hat{n}^{(t+1)}(x_i, x_{pa(i)})$$

$$\hat{p}^{(t+1)}(x_i | x_{pa(i)}) = \hat{n}^{(t+1)}(x_i, x_{pa(i)}) / \hat{n}^{(t+1)}(x_{pa(i)})$$

od
$t = t + 1$
Until $\sum |\hat{\mathbf{p}}^{(t)} - \hat{\mathbf{p}}^{(t-1)}| < \varepsilon$
Return $\hat{\mathbf{p}}$