Bayesian networks – Learning

Pekka Parviainen

University of Helsinki

Prob. Models, Spring 2012
Where do the Bayesian networks come from?

- An expert can construct the network
- Or we can learn the network from data
- Goals of learning
  - Density estimation
  - Specific prediction tasks
  - Knowledge discovery
Learning tasks

- Learning the parameters of a Bayesian network
  - Marginalizing over all parameters.
  - Equivalent to choosing the expected parameters.
- Learning the structure of a Bayesian network
  - Marginalizing over the structures not computationally feasible.
  - Model selection.
Learning the parameters

- Given the data $D$ and the structure $G$, how should I fill the conditional probability tables?

- Bayesian answer:
  - You should not. If you do not know them, you will have priori and posteriori distributions for them.
  - There are many possible distributions, but once again, the independence comes to rescue.
  - Once you have distribution of parameters, you can do the prediction by model averaging.
  - Very similar to Bernoulli case.
A Bayesian network

```
θ_{C}
<table>
<thead>
<tr>
<th>Cloudy=no</th>
<th>Cloudy=yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

θ_{S|C}
<table>
<thead>
<tr>
<th>Sprinkler=on</th>
<th>Sprinkler=off</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>C=no</td>
</tr>
<tr>
<td>S</td>
<td>C=yes</td>
</tr>
</tbody>
</table>

θ_{R|C}
<table>
<thead>
<tr>
<th>Rain=yes</th>
<th>Rain=no</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>C=no</td>
</tr>
<tr>
<td>R</td>
<td>C=yes</td>
</tr>
</tbody>
</table>

θ_{W|S,R}
<table>
<thead>
<tr>
<th>WetGrass=yes</th>
<th>WetGrass=no</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>S=on,R=no</td>
</tr>
<tr>
<td>W</td>
<td>S=on,R=yes</td>
</tr>
<tr>
<td>W</td>
<td>S=off,R=no</td>
</tr>
<tr>
<td>W</td>
<td>S=off,R=yes</td>
</tr>
</tbody>
</table>
```
A Bayesian network as a generative model

\[ \Theta_C \]

\[ \Theta_{S|C} \]

\[ \Theta_{W|S,R} \]

\[ \Theta_{R|C} \]

Cloudy

Sprinkler

Rain

Wet Grass
A Bayesian network as a generative model
A Bayesian network as a generative model

▷ Consider network $G$ with $n$ variables. Let $q_i$ be the number of possible value combinations of the parents of variable $i$ and let $r_i$ be the cardinality of $i$.

▷ Parameters are independent a priori:

\[
P(\theta) = \prod_{i=1}^{n} P(\theta_i)
\]

\[
= \prod_{i=1}^{n} \prod_{j=1}^{q_i} P(\theta_{i|j}),
\]

where $P(\theta_{i|j}) = \text{Dir}(\alpha_1, \ldots, \alpha_{r_i})$. 
Generating a data set

▶ l.i.d. samples
Plate notation
**Likelihood** $P(D \mid G, \theta)$

- For one data vector the likelihood is

$$P(x_1, x_2, \ldots, x_n \mid G) = \prod_{i=1}^{n} P(x_i \mid pa_G(X_i)),$$

or

$$P(d_1 \mid G, \theta) = \prod_{i=1}^{n} \theta_{d_1_i \mid pa_1_i},$$

where $d_1_i$ and $pa_1_i$ are the value and the parent configuration of the variable $i$ in data vector $d_1$. 
Likelihood \( P(D \mid G, \theta) \)

- For a data set of \( N \) i.i.d. samples, the likelihood is

\[
P(D \mid G, \theta) = P(d_1, d_2, \ldots, d_N \mid G, \theta)
= \prod_{l=1}^{N} \prod_{i=1}^{n} \theta_{d_{li} \mid pa_{li}}
= \prod_{i=1}^{n} \prod_{k=1}^{r_i} \prod_{j=1}^{q_i} \theta_{i k \mid j}^{N_{ijk}} = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{i j k}^{N_{ijk}},
\]

where \( N_{ijk} \) is the number of data vectors with a parent configuration \( j \) when variable \( i \) has the value \( k \): Further, \( r_i \) and \( q_i \) are the numbers of values and parent configurations of the variable \( i \), respectively.

- The counts \( N_{ijk} \) are sufficient statistics.
Bayesian network learning

\[ P(D|G, \theta) = \prod_{i=1}^{n} \prod_{k=1}^{r_i} \prod_{j=1}^{q_i} \theta_{i k|j}^{N_{ijk}} \]

- \( i \) picks the variable (table)
- \( j \) picks the row
- \( k \) picks the column

\[ N_{S|C} (q_S=2, r_S=2) \]

\[ N_{C} (q_C=1, r_C=2) \]

\[ N_{R|C} (q_R=2, r_R=2) \]

\[ N_{W|S,R} (q_W=4, r_W=2) \]
Bayesian network learning

Data: \((C, S, R, W) = [(\text{no, on, yes, yes}), (\text{no, on, no, no})]\)

\[
P(D|G, \theta) = \prod_{i=1}^{n} \prod_{k=1}^{r_i} \prod_{j=1}^{q_i} \theta_{ikj}^{N_{ijk}}
\]

- \(i\) picks the variable (table)
- \(j\) picks the row
- \(k\) picks the column
- \(r_i\), number of columns in table \(i\)
- \(q_i\), number of rows in table \(i\)
Bayesian network learning after a while (20 data vectors)

\[
P(D|G, \theta) = \prod_{i=1}^{n} \prod_{k=1}^{r_i} \prod_{j=1}^{q_i} \theta_{ikj}^{N_{ikj}}
\]

- i picks the variable (table)
- j picks the row
- k picks the column
- \(r_i\) number of columns in table i
- \(q_i\) number of rows in table i

\[
N_C
\]

\[
N_{S|C}
\]

- Sprinkler=on
  - Sprinkler=on
    - Cloudy=off
      - 10
    - Cloudy=on
      - 1
  - Sprinkler=off
    - 6
    - 3

\[
N_{S|C=\text{yes}}
\]

\[
N_{R|C}
\]

- Rain=yes
  - 3
  - 4
- Rain=no
  - 13
  - 0

\[
N_{W|S,R}
\]

- WetGrass=yes
  - 2
  - 6
- WetGrass=no
  - 3
  - 8
Maximum likelihood

- Since the parameters occur separately in likelihood we can maximize the terms independently:

$$P(D \mid G, \theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$$

- Maximum likelihood parameters:

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{k' = 1}^{r_i} N_{ijk'}}$$

- So you simply normalize the rows in the sufficient statistics tables to get ML-parameters.

- But these parameters may have zero probabilities...
Bayesian approach

- A priori, parameters are independently Dirichlet:

\[
P(\theta \mid \alpha) = \prod_{i=1}^{n} P(\theta_i)
= \prod_{i=1}^{n} \prod_{j=1}^{q_i} P(\theta_i \mid j)
= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \frac{\Gamma \left( \sum_{k=1}^{r_i} \alpha_{ijk} \right)} {\prod_{k=1}^{r_i} \Gamma \left( \alpha_{ijk} \right)} \theta_{ijk}^{\alpha_{ijk} - 1}
\]

- Likelihood is compatible with conjugate prior:

\[
P(D \mid G, \theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}
\]
Bayesian approach

- Yields a simple posteriori

\[
P(\theta \mid D, G, \alpha) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \frac{\Gamma(\sum_{k=1}^{r_i} N_{ijk} + \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(N_{ijk} + \alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk} + \alpha_{ijk} - 1}
\]

\[
= \prod_{i=1}^{n} \prod_{j=1}^{q_i} P(\theta_{ij} \mid N_{ij}, \alpha_{ij}),
\]

where \( P(\theta_{ij} \mid N_{ij}, \alpha_{ij}) = \text{Dir}(N_{ij} + \alpha_{ij}). \)
Predictive distribution $P(d \mid D, G, \alpha)$

- Predictive distribution:

$$P(d \mid D, G, \alpha) = \int_{\theta} P(d, \theta \mid D, G, \alpha) d\theta$$

$$= \int_{\theta} P(d \mid \theta) P(\theta \mid D, G, \alpha) d\theta$$

$$= \int_{\theta} \prod_{i=1}^{n} P(d_i \mid \theta_i, pa_i) P(\theta_i \mid D, G, \alpha) d\theta$$

$$= \prod_{i=1}^{n} \int_{\theta_{ijk}} \theta_{ijk} P(\theta_{ijk} \mid N_{ij}, G, \alpha_{ij}) d\theta_{ijk}$$

$$= \prod_{i=1}^{n} \frac{N_{ijk} + \alpha_{ijk}}{\sum_{k'=1}^{r_i}(N_{ijk'} + \alpha_{ijk'})},$$

where $j$ is the parent configuration and $k$ is the value of $i$ (in $d$).
Predictive distribution

- This means that predictive distribution

\[
P(d \mid D, G, \alpha) = \prod_{i=1}^{n} \frac{N_{ijk} + \alpha_{ijk}}{\sum_{k'=1}^{r_i}(N_{ijk'} + \alpha_{ijk'})}
\]

can be achieved by just setting

\[
\theta_{ijk} = \frac{N_{ijk} + \alpha_{ijk}}{N_{ij} + \alpha_{ij}},
\]

where \( N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \) and \( \alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk} \)

- So just gather counts \( N_{ijk} \), add \( \alpha_{ijk} \) to them and normalize.
Further readings

- Neapolitan (2004), Chapters 6-7
- Koller & Friedman (2009), Chapter 17