Learning the structure of Bayesian belief networks

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Learning of BBN

Learning:
• Learning of parameters of conditional probabilities
• Learning of the network structure

Variables:
• Observable – values present in every data sample
• Hidden – they values are never observed in data
• Missing values – values sometimes present, sometimes not

Next: All variables are observable
1. Learning of parameters of BBN
2. Learning of the model (BBN structure)
Learning of BBN parameters. Example.

**Example:**

- **Pneumonia**

  - **P(Pneumonia)**
    - T | F | ? | ?
  - **P(HWBC|Pneum)**
    - Pn | T | F
      - T | ? | ?
      - F | ? | ?

- **Paleness**
- **Fever**
- **Cough**
- **High WBC**

  - **P(Paleness|Pneum)**
  - **P(Fever|Pneum)**
  - **P(Cough|Pneum)**

  - ? | ? | ?

Learning of BBN parameters. Example.

**Data D (different patient cases):**

<table>
<thead>
<tr>
<th>Pal</th>
<th>Fev</th>
<th>Cou</th>
<th>HWB</th>
<th>Pneu</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

CS 2750 Machine Learning
Estimates of parameters of BBN

- Much like multiple coin toss or roll of a dice problems.
- A “smaller” learning problem corresponds to the learning of exactly one conditional distribution
- Example: $P(Fever \mid Pneumonia = T)$
- Problem: How to pick the data to learn?

Learning of BBN parameters. Example.

Learn: $P(Fever \mid Pneumonia = T)$

Step 1: Select data points with Pneumonia=T

<table>
<thead>
<tr>
<th>Pal</th>
<th>Fev</th>
<th>Cou</th>
<th>HWB</th>
<th>Pneu</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Learning of BBN parameters. Example.

**Learn:**  \( P(Fever \mid Pneumonia = T) \)

**Step 1:** Ignore the rest

<table>
<thead>
<tr>
<th>Pal</th>
<th>Fev</th>
<th>Cou</th>
<th>HWB</th>
<th>Pneu</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Learning of BBN parameters. Example.

**Learn:**  \( P(Fever \mid Pneumonia = T) \)

**Step 2:** Select values of the random variable defining the distribution of Fever

<table>
<thead>
<tr>
<th>Pal</th>
<th>Fev</th>
<th>Cou</th>
<th>HWB</th>
<th>Pneu</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Learning of BBN parameters. Example.

Learn: \( P(Fever \mid Pneumonia = T) \)

Step 2: Ignore the rest

Learning of BBN parameters. Example.

Learn: \( P(Fever \mid Pneumonia = T) \)

Step 3a: Learning the ML estimate

\[
P(Fever \mid Pneumonia = T)
\]

\[
\begin{array}{cc}
T & F \\
0.6 & 0.4 \\
\end{array}
\]
Learning of BBN parameters. Bayesian learning.

Learn: \( P(Fever \mid Pneumonia = T) \)

Step 3b: Learning the Bayesian estimate

Assume the prior

\[
\theta_{Fever \mid Pneumonia = T} \sim \text{Beta}(3,4)
\]

\( \begin{align*}
\text{Fev} & : F \\
\text{F} & : F \\
\text{T} & : T \\
\text{T} & : T
\end{align*} \)

Posterior:

\[
\theta_{Fever \mid Pneumonia = T} \sim \text{Beta}(6,6)
\]

Model selection

- BBN has two components:
  - Structure of the network (models conditional independences)
  - A set of parameters (conditional child-parent distributions)

We already know how to learn the parameters for the fixed structure

But how to learn the structure of the BBN?
Learning the structure

Criteria we can choose to score the structure $S$

- **Marginal likelihood**
  
  maximize $P(D \mid S, \xi)$

  $\xi$ - represents the prior knowledge

- **Maximum posterior probability**

  maximize $P(S \mid D, \xi)$

\[
P(S \mid D, \xi) = \frac{P(D \mid S, \xi) P(S \mid \xi)}{P(D \mid \xi)}
\]

How to compute marginal likelihood $P(D \mid S, \xi)$?

Learning of BBNs

- **Notation:**
  
  - $i$ ranges over all possible variables $i=1,..,n$
  - $j=1,..,q$ ranges over all possible parent combinations
  - $k=1,..,r$ ranges over all possible variable values
  - $\Theta$ - parameters of the BBN

  $\Theta_{ij}$ is a vector of $\Theta_{ijk}$ representing parameters of the conditional probability distribution; such that $\sum_{k=1}^{r} \Theta_{ijk} = 1$

  $N_{ijk}$ - a number of instances in the dataset where parents of variable $X_i$ take on values $j$ and $X_i$ has value $k$

  $N_j = \sum_{k=1}^{r} N_{ijk}$

  $\alpha_{ijk}$ - prior counts (parameters of Beta and Dirichlet priors)

  $\alpha_{ij} = \sum_{k=1}^{r} \alpha_{ijk}$
Marginal likelihood

- Integrate over all possible parameter settings

\[ P(D \mid S, \xi) = \int_{\Theta} P(D \mid S, \Theta, \xi) p(\Theta \mid S, \xi) d\Theta \]

- Using the assumption of parameter and sample independence

\[
P(D \mid S, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}
\]

- We can use log-likelihood score instead

\[
\log P(D \mid S, \xi) = \sum_{i=1}^{n} \left( \sum_{j=1}^{q_i} \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right)
\]

Score is decomposable along variables !!!

---

Marginal likelihood

- **From the iid assumption:**

\[
P(D \mid S, \Theta) = \prod_{h=1}^{N} \prod_{i=1}^{n} P(x_{ih} \mid \text{parents}_{ih}, \Theta)
\]

- Let \( r_i = \) number of values that attribute \( x_i \) can take
  \( q_i = \) number of possible parent combinations
  \( N_{ijk} = \) number of cases in \( D \) where \( x_i \) has value \( k \) and parents with values \( j \).

\[
= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} P(x_i = k \mid \text{parents}_i = j, \Theta)^{N_{ijk}}
\]

\[
= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \Theta_{ijk}^{N_{ijk}}
\]
Marginal likelihood

- From parameter independence

\[ p(\Theta \mid S, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\Theta_{ij} \mid S, \xi) \]

- Priors for \( p(\Theta_{ij} \mid S, \xi) \)
  - \( \Theta_{ij} = (\Theta_{ij1}, \ldots, \Theta_{ijr}) \) is a vector of parameters;
  - we use a Dirichlet distribution with parameters \( \alpha \) to represent it

\[ P(\Theta_{ij} \mid S, \xi) = P(\Theta_{ij1}, \ldots, \Theta_{ijr} \mid S, \xi) = Dirichlet(\Theta_{ij1}, \ldots, \Theta_{ijr} \mid \alpha) \]

\[
\frac{\Gamma\left(\sum_{k=1}^{r_j} \alpha_{jk}\right)}{\prod_{k=1}^{r_j} \Gamma(\alpha_{jk})} \prod_{k=1}^{r_j} \Theta_{ijk}^{a_{jk}-1} \]

Marginal likelihood

- Combine things together:

\[
P(D \mid S_0) = \int \int P(D \mid S_0, \Theta) P(\Theta \mid S_0)d\Theta \\
= \int \left[ \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_j} \Theta_{ijk}^{N_{ijk}} \cdot \frac{\Gamma\left(\sum_{k=1}^{r_j} \alpha_{jk}\right)}{\prod_{k=1}^{r_j} \Gamma(\alpha_{jk})} \prod_{k=1}^{r_j} \Theta_{ijk}^{a_{jk}-1} \right] d\Theta \\
= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma\left(\sum_{k=1}^{r_j} \alpha_{jk}\right)}{\prod_{k=1}^{r_j} \Gamma(\alpha_{jk})} \int \prod_{k=1}^{r_j} \Theta_{ijk}^{N_{ijk}+a_{jk}-1} d\Theta \\
= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{yj})}{\prod_{k=1}^{r_j} \Gamma(\alpha_{yk})} \cdot \prod_{k=1}^{r_j} \frac{\Gamma(a_{yk} + N_{ijk})}{\Gamma(a_{yk} + N_{ijk})} \\
\]
An alternative way to compute the marginal likelihood

• Integrate over all possible parameter settings
\[ P(D \mid S, \xi) = \int_{\Theta} P(D \mid S, \Theta, \xi) p(\Theta \mid S, \xi) d\Theta \]

• Posterior of parameters, given data and the structure
\[ p(\Theta \mid D, S, \xi) = \frac{P(D \mid \Theta, S, \xi) p(\Theta \mid S, \xi)}{P(D \mid S, \xi)} \]

Trick
\[ P(D \mid S, \xi) = \frac{P(D \mid \Theta, S, \xi) p(\Theta \mid S, \xi)}{p(\Theta \mid D, S, \xi)} \]

• Gives the solution
\[ P(D \mid S, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{N} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \]

Learning the structure

• Likelihood of data for the BBN (structure and parameters)
\[ P(D \mid S, \Theta, \xi) \]
measures the goodness of fit of the BBN to data

• Marginal likelihood (for the structure only)
\[ P(D \mid S, \xi) \]

• Does not measure only a goodness of fit. It is:
  – different for structures of different complexity
  – Incorporates preferences towards simpler structures, implements Occam’s razor !!!!
Occam’s Razor

• Why there is a preference towards simpler structures?

Rewrite marginal likelihood as

\[
P(D \mid S, \xi) = \frac{\int P(D \mid S, \Theta, \xi) p(\Theta \mid S, \xi) d\Theta}{\int p(\Theta \mid S, \xi) d\Theta}
\]

We know that \( \int p(\Theta \mid S, \xi) d\Theta = 1 \)

**Interpretation:** in more complex structures there are more ways parameters can be set badly

– **The numerator:** count of good assignments
– **The denominator:** count of all assignments

Approximations of probabilistic scores

Approximations of the marginal likelihood and posterior scores

• **Information based measures**
  – Akaike criterion
  – Bayesian information criterion (BIC)
  – Minimum description length (MDL)

• Reflect the tradeoff between the fit to data and preference towards simpler structures

Example: **Akaike criterion.**

**Maximize:** \( \text{score}(S) = \log P(D \mid S, \Theta_{ML}, \xi) - \text{compl}(S) \)

**Bayesian information criterion (BIC)**

**Maximize:** \( \text{score}(S) = \log P(D \mid S, \Theta_{ML}, \xi) - \frac{1}{2} \text{compl}(S) \log N \)
Optimizing the structure

Finding the best structure is a \textit{combinatorial optimization} problem

- A good feature: the score is decomposable along variables:

\[
\log P(D \mid S, \xi) = \sum_{i=1}^{n} \left\{ \sum_{j=1}^{q_i} \log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r} \log \frac{\Gamma(\alpha_{jk} + N_{jk})}{\Gamma(\alpha_{jk})} \right\}
\]

\textbf{Algorithm idea:} Search the space of structures using local changes (additions and deletions of a link)

\[
\begin{array}{c}
\text{Algorithm idea:} \quad \text{Search the space of structures using local changes (additions and deletions of a link)} \\
\begin{array}{c}
\text{Algorithm idea:} \quad \text{Search the space of structures using local changes (additions and deletions of a link)} \\
\end{array}
\end{array}
\]

\textbf{Advantage:}
- we do not have to compute the whole score from scratch
- Recompute the partial score for the affected variable

---

Optimizing the structure. Algorithms

- \textbf{Greedy search}
  - Start from the structure with no links
  - Add a link that yields the best score improvement

- \textbf{Metropolis algorithm (with simulated annealing)}
  - Local additions and deletions
  - Avoids being trapped in “local” optimal