you should understand the following concepts

- the parameter learning task for Bayes nets
- the structure learning task for Bayes nets
- maximum likelihood estimation
- Laplace estimates
- $m$-estimates
- missing data in machine learning
  - hidden variables
  - missing at random
  - missing systematically
- the EM approach to imputing missing values in Bayes net parameter learning
- the Chow-Liu algorithm for structure search
Learning Bayes Networks: Parameters
The parameter learning task

• Given: a set of training instances, the graph structure of a BN

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>A</th>
<th>J</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</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>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Do: infer the parameters of the CPDs
The structure learning task

• Given: a set of training instances

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>A</th>
<th>J</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</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>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Do: infer the graph structure (and perhaps the parameters of the CPDs too)
Parameter learning and MLE

- **maximum likelihood estimation** (MLE)
  - given a model structure (e.g. a Bayes net graph) $G$ and a set of data $D$
  - set the model parameters $\theta$ to maximize $P(D \mid G, \theta)$

- i.e. make the data $D$ look **as likely as possible** under the model $P(D \mid G, \theta)$
Consider trying to estimate the parameter $\theta$ (probability of heads) of a biased coin from a sequence of flips (1 stands for head)

$$x = \{1, 1, 1, 0, 1, 0, 0, 1, 0, 1\}$$

The likelihood function for $\theta$ is given by:

$$L(\theta : x_1, \ldots, x_n) = \theta^{x_1} (1 - \theta)^{1-x_1} \cdots \theta^{x_n} (1 - \theta)^{1-x_n}$$

$$= \theta^{\sum x_i} (1 - \theta)^{n-\sum x_i}$$

What’s MLE of the parameter?
MLE in a Bayes net

\[ L(\theta : D, G) = P(D \mid G, \theta) = \prod_{d \in D} P(x_1^{(d)}, x_2^{(d)}, \ldots, x_n^{(d)}) \]

\[ = \prod_{d \in D} \prod_{i} P(x_i^{(d)} \mid \text{Parents}(x_i^{(d)})) \]

\[ = \prod_{i} \left( \prod_{d \in D} P(x_i^{(d)} \mid \text{Parents}(x_i^{(d)})) \right) \]
MLE in a Bayes net

\[ L(\theta : D, G) = P(D \mid G, \theta) = \prod_{d \in D} P(x_1^{(d)}, x_2^{(d)}, \ldots, x_n^{(d)}) \]

\[ = \prod_{d \in D} \prod_{i} P(x_i^{(d)} \mid \text{Parents}(x_i^{(d)})) \]

\[ = \prod_{i} \left( \prod_{d \in D} P(x_i^{(d)} \mid \text{Parents}(x_i^{(d)})) \right) \]

independent parameter learning problem for each CPD
now consider estimating the CPD parameters for $B$ and $J$ in the alarm network given the following data set

\[\begin{array}{cccccc}
B & E & A & J & M \\
\hline
f & f & f & t & f \\
f & t & f & f & f \\
f & f & f & t & t \\
t & f & f & f & t \\
f & f & t & t & f \\
f & f & t & f & t \\
f & f & t & t & t \\
f & f & t & t & t \\
f & f & t & t & t \\
\end{array}\]

\[P(b) = \frac{1}{8} = 0.125\]

\[P(\neg b) = \frac{7}{8} = 0.875\]
now consider estimating the CPD parameters for $B$ and $J$ in the alarm network given the following data set

<table>
<thead>
<tr>
<th>$B$</th>
<th>$E$</th>
<th>$A$</th>
<th>$J$</th>
<th>$M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</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>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

$P(b) = \frac{1}{8} = 0.125$

$P(\neg b) = \frac{7}{8} = 0.875$

$P(j \mid a) = \frac{3}{4} = 0.75$

$P(\neg j \mid a) = \frac{1}{4} = 0.25$

$P(j \mid \neg a) = \frac{2}{4} = 0.5$

$P(\neg j \mid \neg a) = \frac{2}{4} = 0.5$
suppose instead, our data set was this…

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>A</th>
<th>J</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</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>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</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>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

\[ P(b) = \frac{0}{8} = 0 \]

\[ P(\neg b) = \frac{8}{8} = 1 \]

do we really want to set this to 0?
Laplace estimates

• instead of estimating parameters strictly from the data, we could start with some prior belief for each

• for example, we could use Laplace estimates

\[ P(X = x) = \frac{n_x + 1}{\sum_{v \in \text{Values}(X)}(n_v + 1)} \]

• where \( n_v \) represents the number of occurrences of value \( v \)
M-estimates

a more general form: \textit{m-estimates}

\[
P(X = x) = \frac{n_x + p_x m}{\sum_{v \in \text{Values}(X)} n_v} + m
\]

prior probability of value \(x\)

number of "virtual" instances
M-estimates example

Now let's estimate parameters for $B$ using $m=4$ and $p_b=0.25$.

$$P(b) = \frac{0 + 0.25 \times 4}{8 + 4} = \frac{1}{12} = 0.08$$

$$P(\neg b) = \frac{8 + 0.75 \times 4}{8 + 4} = \frac{11}{12} = 0.92$$

<table>
<thead>
<tr>
<th></th>
<th>$B$</th>
<th>$E$</th>
<th>$A$</th>
<th>$J$</th>
<th>$M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t</td>
<td>t</td>
<td>t</td>
<td></td>
</tr>
</tbody>
</table>
EM Algorithm
Missing data

• Commonly in machine learning tasks, some feature values are missing

• some variables may not be observable (i.e. hidden) even for training instances

• values for some variables may be missing at random: what caused the data to be missing does not depend on the missing data itself
  • e.g. someone accidentally skips a question on an questionnaire
  • e.g. a sensor fails to record a value due to a power blip

• values for some variables may be missing systematically: the probability of value being missing depends on the value
  • e.g. a medical test result is missing because a doctor was fairly sure of a diagnosis given earlier test results
  • e.g. the graded exams that go missing on the way home from school are those with poor scores
Missing data

- hidden variables; values *missing at random*
  - these are the cases we’ll focus on
  - one solution: try impute the values

- values *missing systematically*
  - may be sensible to represent “missing” as an explicit feature value
Imputing missing data with EM

Given:
- data set with some missing values
- model structure, initial model parameters

Repeat until convergence
- *Expectation* (E) step: using current model, compute expectation over missing values
- *Maximization* (M) step: update model parameters with those that maximize probability of the data (MLE or MAP)
Example: EM for parameter learning

suppose we’re given the following initial BN and training set

\[
\begin{array}{c|c|c|c|c|c|c|c}
B & E & A & J & M \\
\hline
f & f & ? & f & f \\
f & f & ? & t & f \\
t & f & ? & t & t \\
f & f & ? & f & t \\
f & t & ? & t & f \\
f & f & ? & f & t \\
t & t & ? & t & t \\
f & f & ? & f & f \\
f & f & ? & t & f \\
f & f & ? & f & t
\end{array}
\]
Example: E-step

\[ P(a \mid \neg b, \neg e, \neg j, \neg m) \]
\[ P(\neg a \mid \neg b, \neg e, \neg j, \neg m) \]

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>A</th>
<th>J</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.0069</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>t: 0.98</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f: 0.02</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>t: 0.2</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f: 0.8</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f: 0.2</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f: 0.0069</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>f: 0.8</td>
<td>f</td>
<td>t</td>
</tr>
</tbody>
</table>

\[ P(B) \]
\[ P(E) \]

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>P(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>t</td>
<td>0.9</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>0.6</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>0.3</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>P(J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.9</td>
</tr>
<tr>
<td>f</td>
<td>0.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>P(M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.8</td>
</tr>
<tr>
<td>f</td>
<td>0.1</td>
</tr>
<tr>
<td>f</td>
<td>0.2</td>
</tr>
<tr>
<td>f</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Example: E-step

\[
\begin{align*}
P(a \mid \neg b, \neg e, \neg j, \neg m) &= \frac{P(-b, \neg e, a, \neg j, \neg m)}{P(-b, \neg e, a, \neg j, \neg m) + P(-b, \neg e, \neg a, \neg j, \neg m)} \\
&= \frac{0.9 \times 0.8 \times 0.2 \times 0.1 \times 0.2}{0.9 \times 0.8 \times 0.2 \times 0.1 \times 0.2 + 0.9 \times 0.8 \times 0.8 \times 0.8 \times 0.9} \\
&= \frac{0.00288}{0.4176} = 0.0069
\end{align*}
\]

\[
\begin{align*}
P(a \mid \neg b, \neg e, j, \neg m) &= \frac{P(-b, \neg e, a, j, \neg m)}{P(-b, \neg e, a, j, \neg m) + P(-b, \neg e, \neg a, j, \neg m)} \\
&= \frac{0.9 \times 0.8 \times 0.2 \times 0.9 \times 0.2}{0.9 \times 0.8 \times 0.2 \times 0.9 \times 0.2 + 0.9 \times 0.8 \times 0.2 \times 0.2 \times 0.9} \\
&= \frac{0.02592}{0.1296} = 0.2
\end{align*}
\]
Example: M-step

Re-estimate probabilities using expected counts

\[ P(a \mid b, e) = \frac{E\#(a \land b \land e)}{E\#(b \land e)} \]

\[ P(a \mid b, e) = \frac{0.997}{1} \]

\[ P(a \mid b, \neg e) = \frac{0.98}{1} \]

\[ P(a \mid \neg b, e) = \frac{0.3}{1} \]

\[ P(a \mid \neg b, \neg e) = \frac{0.0069 + 0.2 + 0.2 + 0.2 + 0.0069 + 0.2 + 0.2}{7} \]

Re-estimate probabilities for

\[ P(J \mid A) \] and \[ P(M \mid A) \] in same way

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>A</th>
<th>J</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.0069</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>t: 0.98</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>t: 0.3</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.0069</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2</td>
<td>f</td>
<td>t</td>
</tr>
</tbody>
</table>
Example: M-step

re-estimate probabilities

using expected counts

\[ P(j \mid a) = \frac{E\#(a \wedge j)}{E\#(a)} \]

\[ P(j \mid a) = \frac{0.2 + 0.98 + 0.3 + 0.997 + 0.2}{0.0069 + 0.2 + 0.98 + 0.2 + 0.3 + 0.2 + 0.997 + 0.0069 + 0.2 + 0.2} \]

\[ P(j \mid \neg a) = \frac{0.8 + 0.02 + 0.7 + 0.003 + 0.8}{0.9931 + 0.8 + 0.02 + 0.8 + 0.7 + 0.8 + 0.003 + 0.9931 + 0.8 + 0.8} \]

<table>
<thead>
<tr>
<th>$B$</th>
<th>$E$</th>
<th>$A$</th>
<th>$J$</th>
<th>$M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.0069 f: 0.9931</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2 f: 0.8</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>t: 0.98 f: 0.02</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2 f: 0.8</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>t: 0.3 f: 0.7</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2 f: 0.8</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>t: 0.997 f: 0.003</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.0069 f: 0.9931</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2 f: 0.8</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>t: 0.2 f: 0.8</td>
<td>f</td>
<td>t</td>
</tr>
</tbody>
</table>
Convergence of EM

- E and M steps are iterated until probabilities converge
- will converge to a maximum in the data likelihood (MLE or MAP)
- the maximum may be a local optimum, however
- the optimum found depends on starting conditions (initial estimated probability parameters)
Learning Bayes Networks: Structure
Learning structure + parameters

• number of structures is superexponential in the number of variables
• finding optimal structure is NP-complete problem
• two common options:
  • search very restricted space of possible structures (e.g. networks with tree DAGs)
  • use heuristic search (e.g. sparse candidate)
The Chow-Liu algorithm

• learns a BN with a tree structure that maximizes the likelihood of the training data

• algorithm
  1. compute weight $I(X_i, X_j)$ of each possible edge $(X_i, X_j)$
  2. find maximum weight spanning tree (MST)
  3. assign edge directions in MST
1. use mutual information to calculate edge weights

\[
I(X, Y) = \sum_{x \in \text{values}(X)} \sum_{y \in \text{values}(Y)} P(x, y) \log_2 \frac{P(x, y)}{P(x)P(y)}
\]
2. find maximum weight spanning tree: a maximal-weight tree that connects all vertices in a graph

The Chow-Liu algo always have a complete graph, but here we use a non-complete graph as the example for clarity.
Kruskal’s algorithm for finding an MST

given: graph with vertices $V$ and edges $E$

$$E_{\text{new}} \leftarrow \{ \}$$
for each $(u, v)$ in $E$ ordered by weight (from high to low)
{
remove $(u, v)$ from $E$
if adding $(u, v)$ to $E_{\text{new}}$ does not create a cycle
add $(u, v)$ to $E_{\text{new}}$
}
return $V$ and $E_{\text{new}}$ which represent an MST
Finding MST in Chow-Liu

i.

ii.

iii.

iv.
Finding MST in Chow-Liu

v.

vi.
3. pick a node for the root, and assign edge directions
The Chow-Liu algorithm

• How do we know that Chow-Liu will find a tree that maximizes the data likelihood?

• Two key questions:
  • Why can we represent data likelihood as sum of $I(X;Y)$ over edges?
  • Why can we pick any direction for edges in the tree?
Why Chow-Liu maximizes likelihood (for a tree)

data likelihood given directed edges

$$\log_2 P(D \mid G, \theta_G) = \sum_{d \in D} \sum_i \log_2 P(x_i^{(d)} \mid Parents(X_i))$$

$$E[\log_2 P(D \mid G, \theta_G)] = |D| \sum_i (I(X_i, Parents(X_i)) - H(X_i))$$

we’re interested in finding the graph $G$ that maximizes this

$$\arg \max_G \log_2 P(D \mid G, \theta_G) = \arg \max_G \sum_i I(X_i, Parents(X_i))$$

if we assume a tree, each node has at most one parent

$$\arg \max_G \log_2 P(D \mid G, \theta_G) = \arg \max_G \sum_{(X_i, X_j) \in \text{edges}} I(X_i, X_j)$$

edge directions don’t matter for likelihood, because MI is symmetric

$$I(X_i, X_j) = I(X_j, X_i)$$
THANK YOU

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, and Pedro Domingos.