Learning Bayesian Networks
(part 1)

Mark Craven
Computer Sciences 760
Fall 2016

www.biostat.wisc.edu/~craven/cs760/

Some of the slides in these lectures have been adapted/borrowed from materials developed by Tom Dietterich, Pedro Domingos, Tom Mitchell, David Page, and Jude Shavlik

Goals for the lecture

you should understand the following concepts

• the Bayesian network representation
• inference by enumeration
• 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
Bayesian network example

- Consider the following 5 binary random variables:
  - $B$ = a burglary occurs at your house
  - $E$ = an earthquake occurs at your house
  - $A$ = the alarm goes off
  - $J$ = John calls to report the alarm
  - $M$ = Mary calls to report the alarm

- Suppose we want to answer queries like what is $P(B \mid M, J)$?
Bayesian networks

- a BN consists of a Directed Acyclic Graph (DAG) and a set of conditional probability distributions

- in the DAG
  - each node denotes random a variable
  - each edge from $X$ to $Y$ represents that $X$ directly influences $Y$
  - formally: each variable $X$ is independent of its non-descendants given its parents

- each node $X$ has a conditional probability distribution (CPD) representing $P(X | Parents(X))$

Bayesian networks

- using the chain rule, a joint probability distribution can be expressed as

  $$P(X_1, \ldots, X_n) = P(X_1) \prod_{i=2}^{n} P(X_i | X_1, \ldots, X_{i-1})$$

- a BN provides a compact representation of a joint probability distribution

  $$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Parents(X_i))$$
Bayesian networks

\[ P(B, E, A, J, M) = P(B) \times P(E) \times P(A \mid B, E) \times P(J \mid A) \times P(M \mid A) \]

- a standard representation of the joint distribution for the Alarm example has \(2^5 = 32\) parameters
- the BN representation of this distribution has 20 parameters

Bayesian networks

- consider a case with 10 binary random variables

- How many parameters does a BN with the following graph structure have?

\[ = 42 \]

- How many parameters does the standard table representation of the joint distribution have? \(= 1024\)
Advantages of the Bayesian network representation

- Captures independence and conditional independence where they exist
- Encodes the relevant portion of the full joint among variables where dependencies exist
- Uses a graphical representation which lends insight into the complexity of inference

The inference task in Bayesian networks

**Given**: values for some variables in the network (*evidence*), and a set of *query* variables

**Do**: compute the posterior distribution over the query variables

- variables that are neither evidence variables nor query variables are *hidden* variables
- the BN representation is flexible enough that any set can be the evidence variables and any set can be the query variables
Inference by enumeration

• let $a$ denote $A=true$, and $\neg a$ denote $A=false$

• suppose we’re given the query: $P(b \mid j, m)$
  “probability the house is being burglarized given that John and Mary both called”

• from the graph structure we can first compute:

$$P(b, j, m) = \sum_{e, \neg e, a, \neg a} P(b)P(E)P(A \mid b, E)P(j \mid A)P(m \mid A)$$

sum over possible values for $E$ and $A$ variables ($e, \neg e, a, \neg a$)

Inference by enumeration

$$P(b, j, m) = \sum_{e, \neg e, a, \neg a} P(b)P(E)P(A \mid b, E)P(j \mid A)P(m \mid A)$$

$$\sum_{e, \neg e, a, \neg a} P(b)P(E)P(A \mid b, E)P(j \mid A)P(m \mid A)$$

$$= P(b)\sum_{e, \neg e, a, \neg a} P(E)P(A \mid b, E)P(j \mid A)P(m \mid A)$$

$$= 0.001 \times (0.001 \times 0.95 \times 0.9 \times 0.7 + e, a$$

$$0.001 \times 0.05 \times 0.05 \times 0.01 + e, \neg a$$

$$0.999 \times 0.94 \times 0.9 \times 0.7 + \neg e, a$$

$$0.999 \times 0.06 \times 0.05 \times 0.01) \neg e, \neg a$$

<table>
<thead>
<tr>
<th>$B$</th>
<th>$E$</th>
<th>$P(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>t</td>
<td>0.95</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>0.94</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>0.29</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$A$</th>
<th>$P(B)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.9</td>
</tr>
<tr>
<td>f</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Inference by enumeration

• now do equivalent calculation for \( P(\neg b, j, m) \)
• and determine \( P(b \mid j, m) \)

\[
P(b \mid j, m) = \frac{P(b, j, m)}{P(j, m)} = \frac{P(b, j, m)}{P(b, j, m) + P(\neg b, j, m)}
\]

Comments on BN inference

• inference by enumeration is an exact method (i.e. it computes the exact answer to a given query)
• it requires summing over a joint distribution whose size is exponential in the number of variables
• in many cases we can do exact inference efficiently in large networks
  – key insight: save computation by pushing sums inward
• in general, the Bayes net inference problem is NP-hard
• there are also methods for approximate inference – these get an answer which is “close”
• in general, the approximate inference problem is NP-hard also, but approximate methods work well for many real-world problems
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>
</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>
</tbody>
</table>

- Do: infer the graph structure (and perhaps the parameters of the CPDs too)
Parameter learning and maximum likelihood estimation

- **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 \( P(D \mid G, \theta) \)

Maximum likelihood estimation

consider trying to estimate the parameter \( \theta \) (probability of heads) of a biased coin from a sequence of flips

\( x = \{1,1,1,0,1,0,1,0,1,1\} \)

the likelihood function for \( \theta \) is given by:

\[
L(\theta : x_1, \ldots, x_n) = \theta^{x_1}(1-\theta)^{1-x_1} \cdots \theta^{x_n}(1-\theta)^{1-x_n} = \theta^{\sum x_i}(1-\theta)^{n-\sum x_i}
\]

for \( h \) heads in \( n \) flips
the MLE is \( \frac{h}{n} \)
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

Maximum likelihood estimation

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>( 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>

\( P(b) = \frac{1}{8} = 0.125 \)
\( P(\neg b) = \frac{7}{8} = 0.875 \)
\( P(j \mid a) = \frac{3}{4} = 0.75 \)
\( P(\neg j \mid a) = \frac{1}{4} = 0.25 \)
\( P(j \mid \neg a) = \frac{2}{4} = 0.5 \)
\( P(\neg j \mid \neg a) = \frac{2}{4} = 0.5 \)
Maximum likelihood estimation

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>
</tbody>
</table>

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

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

do we really want to set this to 0?

Maximum a posteriori (MAP) estimation

• 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 \( v \)
Maximum a posteriori estimation

a more general form: \( m \)-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 \( p_B=0.25 \)

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

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

<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>B</th>
<th>E</th>
<th>( P(B) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>f</td>
<td>0.1</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>
</tbody>
</table>

<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>?</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>?</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>?</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>?</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>?</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>?</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>?</td>
<td>t</td>
<td>t</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>?</td>
<td>f</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>?</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>?</td>
<td>f</td>
<td>t</td>
</tr>
</tbody>
</table>
example: E-step

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

\[
\begin{array}{c|c|c|c|c}
B & E & A & J & M \\
\hline
f & f & f & f & f \\
0.0069 & 0.0069 & 0.0931 & 0.0931 & 0.0931 \\
0.098 & 0.098 & 0.02 & 0.02 & 0.02 \\
f & f & t & f & f \\
f & t & 0.2 & f & t \\
t & t & 0.3 & t & f \\
f & f & 0.2 & f & t \\
t & t & 0.997 & t & t \\
f & f & 0.0069 & f & f \\
f & f & 0.2 & f & t \\
f & f & 0.2 & f & t \\
f & f & 0.2 & f & t \\
\end{array}
\]

- Example: E-step

\[ P(a \mid \neg b, \neg e, \neg j, \neg m) = \frac{P(\neg b, \neg e, a, j, \neg m)}{P(\neg b, \neg e, a, j, \neg m) + P(\neg b, \neg e, a, j, \neg m)} \]
\[ = \frac{0.9 \times 0.8 \times 0.2 \times 0.1 \times 0.2}{0.9 \times 0.8 \times 0.2 \times 0.1 \times 0.2 + 0.9 \times 0.8 \times 0.8 \times 0.9} \]
\[ = 0.0069 \approx 0.0069 \]

\[ P(a \mid b, \neg e, j, \neg m) = \frac{P(b, \neg e, a, j, \neg m)}{P(b, \neg e, a, j, \neg m) + P(b, \neg e, a, j, \neg m)} \]
\[ = \frac{0.9 \times 0.8 \times 0.2 \times 0.1 \times 0.2}{0.9 \times 0.8 \times 0.2 \times 0.1 \times 0.2 + 0.9 \times 0.8 \times 0.8 \times 0.9} \]
\[ = 0.02592 = 0.02592 \]

\[ P(a \mid b, \neg e, j, \neg m) = \frac{P(b, \neg e, a, j, \neg m)}{P(b, \neg e, a, j, \neg m) + P(b, \neg e, a, j, \neg m)} \]
\[ = \frac{0.1 \times 0.8 \times 0.2 \times 0.1 \times 0.2}{0.1 \times 0.8 \times 0.2 \times 0.1 \times 0.2 + 0.1 \times 0.8 \times 0.8 \times 0.9} \]
\[ = 0.03456 = 0.03456 \]

\[ P(a \mid b, \neg e, j, \neg m) = \frac{P(b, \neg e, a, j, \neg m)}{P(b, \neg e, a, j, \neg m) + P(b, \neg e, a, j, \neg m)} \]
\[ = \frac{0.1 \times 0.8 \times 0.2 \times 0.1 \times 0.2}{0.1 \times 0.8 \times 0.2 \times 0.1 \times 0.2 + 0.1 \times 0.8 \times 0.8 \times 0.9} \]
\[ = 0.0352 = 0.0352 \]

- Example: E-step
example: M-step

re-estimate probabilities using expected counts

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

\[
\begin{align*}
P(a \mid b, e) &= \frac{0.997}{1} \\
P(a \mid b, \neg e) &= \frac{0.98}{1} \\
P(a \mid \neg b, e) &= \frac{0.3}{1} \\
P(a \mid \neg b, \neg e) &= \frac{0.0069 + 0.2 + 0.2 + 0.2 + 0.0069 + 0.2 + 0.2}{7} \\
\end{align*}
\]

<table>
<thead>
<tr>
<th>B</th>
<th>E</th>
<th>P(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>t</td>
<td>0.997</td>
</tr>
<tr>
<td>t</td>
<td>f</td>
<td>0.98</td>
</tr>
<tr>
<td>f</td>
<td>t</td>
<td>0.3</td>
</tr>
<tr>
<td>f</td>
<td>f</td>
<td>0.145</td>
</tr>
</tbody>
</table>

re-estimate probabilities for \( P(J \mid A) \) and \( P(M \mid A) \) in same way

\[
\begin{align*}
P(j \mid a) &= \frac{E \#(a \land j)}{E \#(a)} \\
P(j \mid \neg a) &= \frac{0.8 + 0.02 + 0.2 + 0.2 + 0.3 + 0.2 + 0.997 + 0.0069 + 0.2 + 0.2}{7} \\
\end{align*}
\]

\[
\begin{align*}
P(j \mid a) &= \frac{E \#(a \land j)}{E \#(a)} \\
P(j \mid \neg a) &= \frac{0.8 + 0.02 + 0.7 + 0.003 + 0.8}{0.9931 + 0.8 + 0.02 + 0.8 + 0.7 + 0.8 + 0.003 + 0.9931 + 0.8 + 0.8} \\
\end{align*}
\]
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)