CSE 473: Artificial Intelligence
Autumn 2014

Bayesian Networks – Learning II

Dan Weld

Slides adapted from Jack Breese, Dan Klein, Daphne Koller, Stuart Russell, Andrew Moore & Luke Zettlemoyer

473 Topics

- Search
  - Problem Spaces
  - BFS, DFS, UCS, A* (tree and graph)
  - Completeness and Optimality
  - Heuristics: admissibility and consistency
- CSPs
  - Constraint graphs, backtracking search
  - Forward checking, AC3 constraint propagation, ordering heuristics
- Games
  - Minimax, Alpha-beta pruning, Expectimax, Evaluation Functions
- MDPs
  - Bellman equations
  - Value iteration
- Reinforcement Learning
  - Exploration vs. Exploitation
  - Model-based vs. model-free
  - Q-learning
  - Linear value function approx.
- Hidden Markov Models
  - Markov chains
  - Forward algorithm
  - Particle Filter
- Bayesian Networks
  - Basic definition, independence (d-sep)
  - Variable elimination
  - Sampling (rejection, importance)
- Learning
  - BN parameters with data complete & incomplete (Expectation Maximization)
  - Search thru space of BN structures
Search thru a Problem Space / State Space

- **Input:**
  - Set of states
  - Operators [and costs]
  - Start state
  - Goal state [test]

- **Output:**
  - Path: start $\Rightarrow$ a state satisfying goal test
  - [May require shortest path]
  - [Sometimes just need state passing test]

Graduation?

- Getting a BS in CSE as a search problem?
  
  *(don’t think too hard)*

- Space of States
- Operators
- Initial State
- Goal State
Topics

- Another Useful Bayes Net
  - Hybrid Discrete / Continuous
- Learning Parameters for a Bayesian Network
  - Fully observable
    - Maximum Likelihood (ML),
    - Maximum A Posteriori (MAP)
  - Bayesian
    - Hidden variables (EM algorithm)
- Learning Structure of Bayesian Networks

Bayes Nets

<table>
<thead>
<tr>
<th>Earthquake</th>
<th>Burglary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nbr1Calls</td>
<td>Nbr2Calls</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Radio</th>
<th>Alarm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr(E=t) Pr(E=f)</td>
<td>0.01 0.99</td>
</tr>
<tr>
<td>Pr(A</td>
<td>E,B)</td>
</tr>
<tr>
<td></td>
<td>e,b</td>
</tr>
<tr>
<td></td>
<td>e,b</td>
</tr>
<tr>
<td></td>
<td>e,b</td>
</tr>
</tbody>
</table>
Continuous Variables

Pr(E=t) Pr(E=f)  
0.01 0.99

Earthquake

So far: assuming variables have discrete values  
Could also allow continuous values, E ∈ R,
How specify probabilities? (explicit CPT would be infinitely large)

\[ P(x \mid \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]

Continuous Variables

Pr(E=t) Pr(E=f)  
0.01 0.99

Earthquake

So far: assuming variables have discrete values  
Could also allow continuous values, E ∈ R,
And specify probabilities using a continuous distribution, such as a Gaussian

\[ P(x \mid \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]
Continuous Variables

- Earthquake
  - \( \text{Pr(E=x)} \)
  - Mean: \( \mu = 6 \)
  - Variance: \( \sigma = 2 \)

So far: assuming variables have discrete values
Could also allow continuous values, \( E \in \mathbb{R} \),
And specify probabilities using a continuous distribution, such as a Gaussian

\[
P(x \mid \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]

Continuous Variables

- Aliens
  - \( \text{Pr}(A=t) \text{Pr}(A=f) \)
    - 0.01 0.99

- Earthquake

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( \text{Pr}(E \mid A) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>( \mu = 6 )</td>
</tr>
<tr>
<td></td>
<td>( \sigma = 2 )</td>
</tr>
<tr>
<td>( \bar{a} )</td>
<td>( \mu = 1 )</td>
</tr>
<tr>
<td></td>
<td>( \sigma = 3 )</td>
</tr>
</tbody>
</table>
Learning Bayes Networks

- **Learning Structure of Bayesian Networks**
  - Search thru space of BN structures

- **Learning Parameters for a Bayesian Network**
  - Fully observable variables
    - Maximum Likelihood (ML), MAP & Bayesian estimation
    - Example: Naïve Bayes for text classification
  - Hidden variables
    - Expectation Maximization (EM)

### Summary

<table>
<thead>
<tr>
<th></th>
<th>Prior</th>
<th>Hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Likelihood Estimate</td>
<td>Uniform</td>
<td>The most likely</td>
</tr>
<tr>
<td></td>
<td>Any</td>
<td>The most likely</td>
</tr>
<tr>
<td>Bayesian Estimate</td>
<td>Any</td>
<td>Weighted combination</td>
</tr>
</tbody>
</table>

- Easy to compute
- Still easy to compute
- Incorporates prior knowledge
- Minimizes error
- Great when data is scarce
- Potentially much harder to compute

12/3/14
Bayesian Learning

Use Bayes rule:

\[ P(Y \mid X) = \frac{P(X \mid Y) P(Y)}{P(X)} \]

Or equivalently: 

\[ P(Y \mid X) \propto P(X \mid Y) P(Y) \]

What Prior to Use?

- Two common priors for continuous variables
  - Binary variable Beta
    - Posterior distribution is binomial
    - Easy to compute posterior
    - Easy to compute MAP estimate
      - MAP \( E[\text{Beta}(a, b)] = a / (a + b) \)
  - Discrete variable Dirichlet
    - Posterior distribution is multinomial
    - Easy to compute posterior
Estimation: Laplace Smoothing

- Laplace’s estimate:
  pretend you saw every outcome once more than you actually did

\[ P_{LAP}(x) = \frac{c(x) + 1}{\sum_x[c(x) + 1]} \]

\[ P_{ML}(X) = \]

\[ P_{LAP}(X) = \]

Another name for computing the MAP estimate with Dirichlet priors (Bayesian justification)

How Learn Continuous CPTs?
Maximum Likelihood
Mean of Single Gaussian

\[ U_{ml} = \text{argmin}_u \sum_i (x_i - u)^2 \]

Learning with Continuous Variables

\[ \hat{\mu}_{MLE} = \frac{1}{N} \sum_{i=1}^{N} x_i \]

\[ \hat{\sigma}^2_{MLE} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{\mu})^2 \]
Output of Learning

Did Learning Work Well?

Can easily calculate $P(\text{data})$ for learned parameters
Topics

- Another Useful Bayes Net
  - Hybrid Discrete / Continuous
- Learning Parameters for a Bayesian Network
  - Fully observable
    - Maximum Likelihood (ML),
    - Maximum A Posteriori (MAP)
  - Bayesian
  - Hidden variables (EM algorithm)
- Learning Structure of Bayesian Networks

Why Learn Hidden Variables?
How Learn Hidden Variables?

- If we knew whether patient had disease
  - It would be easy to learn CPTs
  - But we can’t observe states, so we don’t!

- If we knew CPTs
  - It would be easy to predict if patient had disease
  - But we don’t, so we can’t!

Chicken & Egg Problem
Continuous Variables

\[
\Pr(A=t) \Pr(A=f) = 0.01 \quad 0.99
\]

Aliens

Earthquake

\begin{tabular}{|c|c|}
\hline
Pr(E|A) & \\
\hline
a & \(\mu = 6\) \\
\sigma & 2 \\
\hline
\bar{a} & \(\mu = 1\) \\
\sigma & 3 \\
\hline
\end{tabular}

Simplest Version

- Mixture of two distributions

- Know: form of distribution & variance, \(\sigma = 0.5\)
- Just need \textit{mean} of each distribution
Input Looks Like

We Want to Predict

Naturally Caused

Aliens Caused

Slide by Daniel S. Weld
Chicken & Egg

Note that coloring instances would be easy

if we knew Gausians….

And finding the Gausians would be easy

If we knew the coloring

Slide by Daniel S. Weld
Expectation Maximization (EM)

- Pretend we do know the parameters
  - Initialize randomly: set $\theta_1 = ?; \quad \theta_2 = ?$

### Expectation Maximization (EM)

- Pretend we do know the parameters
  - Initialize randomly
  - \[E\text{ step}\] Compute probability of instance having each possible value of the hidden variable
Expectation Maximization (EM)

- Pretend we do know the parameters
  - Initialize randomly
- [E step] Compute probability of instance having each possible value of the hidden variable

---

**[M step]** Treating each instance as fractionally having both values compute the new parameter values
MLE Mean of Single Gaussian

$$U_{ml} = \arg \min_u \sum_i (x_i - u)^2$$

Expectation Maximization (EM)

[M step] Treating each instance as fractionally having both values compute the new parameter values.
Expectation Maximization (EM)

- **[E step]** Compute probability of instance having each possible value of the hidden variable

- **[M step]** Treating each instance as fractionally having both values compute the new parameter values
Expectation Maximization (EM)

- **[E step]** Compute probability of instance having each possible value of the hidden variable.

- **[M step]** Treating each instance as fractionally having both values compute the new parameter values.

Topics

- Another Useful Bayes Net
  - Hybrid Discrete / Continuous

- Learning Parameters for a Bayesian Network
  - Fully observable
    - Maximum Likelihood (ML),
    - Maximum A Posteriori (MAP)
    - Bayesian
  - Hidden variables (EM algorithm)

- Learning Structure of Bayesian Networks
What if we *don’t* know structure?

Learning The Structure of Bayesian Networks

<table>
<thead>
<tr>
<th>E</th>
<th>B</th>
<th>R</th>
<th>A</th>
<th>J</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

...
Learning The Structure of Bayesian Networks

- Search thru the space...
  - of possible network structures!

- For each structure, learn parameters
  - As just shown...

- Pick the one that fits observed data best
  - Calculate $P(data)$
Two problems:
- Fully connected will be most probable
- Exponential number of structures

Learning The Structure of Bayesian Networks

- Search thru the space…
  - of possible network structures!
- For each structure, learn parameters
  - As just shown…
- Pick the one that fits observed data best
  - Calculate P(data)

Two problems:
- Fully connected will be most probable
  - Add penalty term (regularization) $\propto$ model complexity
- Exponential number of structures
  - Local search
Score Functions

- **Bayesian Information Criterion (BIC)**
  - $P(D \mid BN)$ – penalty
  - Penalty = $\alpha$ complexity
  - $= \alpha \left[ \frac{1}{2} \text{(parameters)} \log(\text{data points}) \right]$
Tuning on Held-Out Data

- Now we’ve got two kinds of unknowns
  - Parameters: the probabilities \( P(Y|X), P(Y) \)
  - Hyperparameters, like
    - the amount of smoothing to do: \( k \), or
    - regularization penalty, \( \alpha \)

- Where to learn?
  - Learn parameters from training data
  - Must tune hyperparameters on different data
    - Why?
  - For each value of the hyperparameters, train and test on the held-out data
  - Choose the best value and do a final test on the test data

Baselines

- First step: get a baseline
  - Baselines are very simple “straw man” procedures
  - Help determine how hard the task is
  - Help know what a “good” accuracy is

- Weak baseline: most frequent label classifier
  - Gives all test instances whatever label was most common in the training set
  - E.g. for spam filtering, might label everything as ham
  - Accuracy might be very high if the problem is skewed
  - E.g. calling everything “spam” gets 86%, so a classifier that gets 90% isn’t very good...

- For real research, usually use previous work as a (strong) baseline