CSE 473: Artificial Intelligence

Bayesian Networks - Learning

Dan Weld

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

What action next?

Static vs. Dynamic

Environment

Percepts

Actions

Fully vs. Partially Observable

Perfect vs. Noisy

Deterministic vs. Stochastic

Instantaneous vs. Durative
AI 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 & policy iteration
  - RTDP, LAO* & UCT
  - POMDPs
- 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
  - Gibbs sampling
- Learning
  - BN parameters with data complete & incomplete (Expectation Maximization)
  - Search thru space of BN structures

Search thru a Problem Space / State Space

Ex. Proving a trig identity, e.g. \( \sin^2(x) = \frac{1}{2} - \frac{1}{2} \cos(2x) \)

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

• Output:
  - Path: start ⇒ a state satisfying goal test
  - [May require shortest path]
  - [Sometimes just need state passing test]
Today

- Bonus Topic – Hybrid Bayes Nets
- Learning
  - Parameter Learning & Priors
  - Expectation Maximization
  - Structure Learning

Bayes Nets

<table>
<thead>
<tr>
<th>Event</th>
<th>Earthquake</th>
<th>Burglary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio</td>
<td>e,b</td>
<td>0.9 (0.1)</td>
</tr>
<tr>
<td>Alarm</td>
<td>e,B</td>
<td>0.2 (0.8)</td>
</tr>
<tr>
<td>Nbr1Calls</td>
<td>e,b</td>
<td>0.85 (0.15)</td>
</tr>
<tr>
<td>Nbr2Calls</td>
<td>e,B</td>
<td>0.01 (0.99)</td>
</tr>
</tbody>
</table>

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

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

Earthquake

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

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

Aliens

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

Earthquake

| Pr(E|A)  | $\mu$ = 6 | $\sigma$ = 2 |
|--------|----------|--------------|
| $\text{a}$ | $\mu$ = 1 | $\sigma$ = 3 |
Supremacy of Machine Learning

- Machine learning is preferred approach to
  - Speech recognition, Natural language processing
  - Web search – result ranking
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - Computational biology
  - Sensor networks
  - ...
- This trend is accelerating
  - Improved machine learning algorithms
  - Improved data capture, networking, faster computers
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for self-customization to user, environment
What is Machine Learning?

Machine Learning

Study of algorithms that
- improve their **performance**
- at some **task**
- with **experience**

?? Reinforcement Learning ??
Machine Learning

Study of algorithms that
- improve their performance
- at some task
- with experience

Learning Bayes Networks

- 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)
- Learning Structure of Bayesian Networks
The Origin of Bayes Nets

Earthquake → Alarm → Nbr1Calls
Burglary → Alarm → Nbr2Calls
Radio → Alarm

| Pr(A|E,B) | e,b | e,b | e,b | e,b |
|----------|-----|-----|-----|-----|
|          | 0.9 (0.1) | 0.2 (0.8) | 0.85 (0.15) | 0.01 (0.99) |

Pr(B=t) Pr(B=f) 0.05 0.95

Learning Bayes Nets

Suppose …
1. Know structure & get complete observations of every var
2. Know structure & get observations only of some vars
   Others are hidden (learn with EM)
3. Don’t even know structure!
Parameter Estimation and Bayesian Networks

We have:
- Bayes Net structure and observations
- We need: Bayes Net parameters

\[
P(B) = \frac{1}{1 + e^{-\gamma}}
\]

\[
P(\neg B) = 1 - P(B)
\]

<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>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Parameter Estimation and Bayesian Networks

\[
P(A|E,B) = ?
\]
\[
P(A|E,\neg B) = ?
\]
\[
P(A|\neg E,B) = ?
\]
\[
P(A|\neg E,\neg B) = 0.5
\]
Parameter Estimation and Bayesian Networks

P(A|E,B) = ?
P(A|E,¬B) = ?
P(A|¬E,B) = ?
P(A|¬E,¬B) = ?

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

Parameter Estimation and Bayesian Networks

Coin
Coin Flip

\[
P(H|C_1) = 0.1 \\ P(H|C_2) = 0.5 \\ P(H|C_3) = 0.9
\]

Which coin will I use?

\[
P(C_1) = 1/3 \\ P(C_2) = 1/3 \\ P(C_3) = 1/3
\]

Prior: Probability of a hypothesis before we make any observations

Uniform Prior: All hypotheses are equally likely before we make any observations
Experiment 1: Heads

Which coin did I use?

\[ P(C_1|H) = \ ? \quad P(C_2|H) = \ ? \quad P(C_3|H) = \ ? \]

\[ P(C_1|H) = \frac{P(H|C_1)P(C_1)}{P(H)} \]

\[ P(H) = \sum_{i=1}^{3} P(H|C_i)P(C_i) \]

\[ P(H|C_1) = 0.1 \quad P(H|C_2) = 0.5 \quad P(H|C_3) = 0.9 \]

\[ P(C_1) = 1/3 \quad P(C_2) = 1/3 \quad P(C_3) = 1/3 \]

Posterior: Probability of a hypothesis given data

P(C_1|H) = 0.066 \quad P(C_2|H) = 0.333 \quad P(C_3|H) = 0.6
Using Prior Knowledge

- Should we always use a **Uniform Prior**?
- **Background knowledge:**
  Heads $\Rightarrow$ we have to buy Dan chocolate
  Dan *likes* chocolate…
  $\Rightarrow$ Dan is more likely to use a coin biased in his favor

\[
\begin{align*}
C_1 & : P(H|C_1) = 0.1 \\
C_2 & : P(H|C_2) = 0.5 \\
C_3 & : P(H|C_3) = 0.9
\end{align*}
\]

Using Background Knowledge

We can encode it in the **prior**:

\[
\begin{align*}
P(C_1) & = 0.05 \\
P(C_2) & = 0.25 \\
P(C_3) & = 0.70
\end{align*}
\]
Experiment 1: Heads

Which coin did I use?

\[ P(C_1|H) = 0.006 \quad P(C_2|H) = 0.165 \quad P(C_3|H) = 0.829 \]

Compare with ML posterior after Exp 1:

\[ P(C_1|H) = 0.066 \quad P(C_2|H) = 0.333 \quad P(C_3|H) = 0.600 \]

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Prior</th>
<th>Bayesian Estimate</th>
<th>Probabilistic Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>The most likely</td>
<td>Uniform</td>
<td>Easy to compute</td>
<td>Incorporates prior knowledge</td>
</tr>
<tr>
<td>The most likely</td>
<td>Any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weighted combination</td>
<td>Any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Minimizes error
Great when data is scarce
Potentially much harder to compute
Bayesian Learning

Use Bayes rule:

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

Or equivalently:

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

Really? Only 3 Coins?

- \( C_1 \): P(H|C_1) = 0.1
- \( C_2 \): P(H|C_2) = 0.5
- \( C_3 \): P(H|C_3) = 0.9

More Likely….
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

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

\[
= \frac{c(x) + 1}{N + |X|}
\]

$PLAP(H) = (2+1)/(3+2) = 3/5$

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

Did Learning Work Well?

Can easily calculate $P(data)$ for learned parameters
Topics

- Another Useful Bayes Net
  - Hybrid Discrete / Continuous
- Learning Parameters for a Bayesian Network
  - Fully observable
  - Hidden variables (EM algorithm)
- Learning Structure of Bayesian Networks

Why Learn Hidden Variables?

- Diagram showing the relationships between variables like Smoking, Diet, Exercise, and Heart Disease with symptoms.
**How Learn Hidden Variables?**

- Smoking
- Diet
- Exercise
- HeartDisease
- Symptom
- Symptom
- Symptom

**Chicken & Egg Problem**

- 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!
Face It...
Continuous Variables

\[ \frac{\Pr(A=t) \Pr(A=f)}{0.01 \ 0.99} \]

Aliens

Earthquake

Pr(E|A)

\begin{array}{c|cc}
\hline
\alpha & \mu = 6 & \\
\bar{\alpha} & \mu = 1 & \sigma = 3 \\
\hline
\end{array}

Learning with Continuous Variables

Pr(E=x)

mean: \( \mu = ? \)

variance: \( \sigma = ? \)

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

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

\[ \text{Pr}(A=t) \cdot \text{Pr}(A=f) \]

\[ \begin{array}{cc} 
0.01 & 0.99 
\end{array} \]

hidden

Aliens

Earthquake

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

Simplest Version

- Mixture of two distributions

- Know: form of distribution & variance, \( \sigma = .5 \)
- Just need \textit{mean} of each distribution
Input Looks Like

We Want to Predict

Naturally Caused

Aliens Caused
Chicken & Egg

Note that coloring instances would be easy if we knew Gaussians….

And finding Gaussian parameters would be easy if we knew the coloring.
Expectation Maximization (EM)

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

---

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
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

[Slide by Daniel S. Weld]

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

[Slide by Daniel S. Weld]
**ML Mean of Single Gaussian**

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

---

**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

Slide by Daniel S. Weld

---

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

Slide by Daniel S. Weld
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)
  - 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 $P(\text{data})$

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

- Can represent strictly more P distributions
- Can represent NOISE in training data
- Often preforms WORSE on test data

Augment Score Function

- Bayesian Information Criterion (BIC)
  - P(D | BN) – penalty
  - Penalty = $\alpha$ complexity
  - $\alpha$ [½ (# parameters) Log (# data points)]

Instance of “regularization”
Solves problem of “overfitting”
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**