Introduction to Artificial Intelligence

Lecture 14 – Information gathering

CS/CNS/EE 154
Andreas Krause
Announcements

- Homework 2 due today
- Homework 3 out later this week
- Final project due December 1
- Code released on Monday (Nov 15)

- Note on midterm grades (Avian Asker)
Sampling based inference

- So far: deterministic inference techniques
  - Variable elimination
  - (Loopy) belief propagation

- Will now introduce stochastic approximations
  - Algorithms that “randomize” to compute expectations
  - In contrast to the deterministic methods, guaranteed to converge to right answer (if wait looong enough..)
  - More exact, but slower than deterministic variants
Forward sampling from a BN

Sample:
\[ D = E, I = L, G = [80, 100], S = B, L = P \]
Rejection sampling

Collect samples over all variables

\[ \hat{P}(X_A = x_A \mid X_B = x_B) \approx \frac{\text{Count}(x_A, x_B)}{\text{Count}(x_B)} \]

Throw away samples that disagree with \( x_B \)

Can be problematic if \( P(X_B = x_B) \) is rare event
Sample complexity for probability estimates

- Absolute error:

\[
Prob\left( \left| \hat{P}(x) - P(x) \right| > \varepsilon \right) \leq 2 \exp\left(-2N\varepsilon^2\right)
\]

- Relative error:

\[
Prob\left( \hat{P}(x) < (1 + \varepsilon)P(x) \right) \leq 2 \exp\left(-NP(x)\varepsilon^2/3\right)
\]

if \( P(x) \) exponentially small, need \( N \) exponentially large
Sampling from rare events

- Estimating conditional probabilities $P(X_A \mid X_B = x_B)$ using rejection sampling is hard!
  - The more observations, the unlikelier $P(X_B = x_B)$ becomes

- Want to directly sample from posterior distribution!
Gibbs sampling

- Start with initial assignment $x^{(0)}$ to all variables
- For $t = 1$ to $\infty$ do
  - Set $x^{(t)} = x^{(t-1)}$
  - For each variable $X_i$
    - Set $v_i = $ values of all $x^{(t)}$ except $x_i$
    - Sample $x^{(t)}_i$ from $P(X_i \mid v_i)$

- For large enough $t$, sampling distribution will be “close” to true posterior distribution!

**Key challenge:** Computing conditional distributions $P(X_i \mid v_i)$
Gibbs Sampling

Gibbs sampling \( P(D,I,G,S,L \mid J = 1) \)

<table>
<thead>
<tr>
<th>Iter</th>
<th>D</th>
<th>I</th>
<th>G</th>
<th>S</th>
<th>L</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>∗</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

...
Example: (Simple) image segmentation

\[
P(x) = \frac{1}{Z} \prod_{i} \Phi(x_i) \prod_{(j,k) \in E} \Psi(x_j, x_k)
\]

\[
\Phi(x_i) = \exp \left\{ -\frac{(y_i - \mu x_i)^2}{2\sigma^2 x_i} \right\}
\]

\[
\Psi(x_i, x_j) = \exp \left\{ -\beta (x_i - x_j)^2 \right\}
\]

[see Singh ‘08]
Gibbs Sampling iterations
Convergence of Gibbs Sampling

$P(X_1 = 1)$

True posterior

Iteration #
Summary: Inference

- For tree-structured Bayes nets, can compute exact marginals
  - Variable elimination
  - Belief propagation (efficiently computes all marginals)
- For loopy networks, can use approximate inference
  - Loopy belief propagation (may not converge)
  - Gibbs sampling (will converge, but may take long time)
Information gathering

So far:

- Bayesian networks for quantifying uncertainty in real world environments
- Exact and approximate algorithms for inference in Bayesian networks (e.g., compute $P(Pit | Breezes)$)

Now:

- Selecting most “informative” variables for making effective predictions / decisions
Why does my car not start?

- Selectively run tests to diagnose cause of failure
Clinical diagnosis?

- Patient either healthy or ill
- Can choose to treat or not treat

<table>
<thead>
<tr>
<th></th>
<th>healthy</th>
<th>ill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment</td>
<td>-$$</td>
<td>$</td>
</tr>
<tr>
<td>No treatment</td>
<td>0</td>
<td>-$$ $$</td>
</tr>
</tbody>
</table>

- Only know distribution $P(\text{ill} \mid \text{observations})$
- Can perform costly medical tests to reveal aspects of the condition

Which tests should we perform to most cost-effectively diagnose?
Autonomous robotic exploration

- Limited time for measurements
- Limited capacity for rock samples

Need optimized information gathering!
A robot scientist

King et al, Nature ‘04
How do people gather information?

[Renninger et al, NIPS ’04]
How do people gather information?
[Renninger et al, NIPS ’04]
How do people gather information?

[Renninger et al, NIPS ’04]
How do people gather information?

[Renninger et al, NIPS ’04]
How do people gather information?

[Renninger et al, NIPS ’04]
How do people gather information?

[Renninger et al, NIPS ’04]
Running example: Detecting fires

Want to place sensors to detect fires in buildings
Monitoring using Bayesian Networks

Joint probability distribution
\[ P(X_1, \ldots, X_n, Y_1, \ldots, Y_n) = P(X_1, \ldots, X_n) \cdot P(Y_1, \ldots, Y_n | X_1, \ldots, X_n) \]

\( X_s \): temperature at location \( s \)
\( Y_s \): sensor value at location \( s \)
\( Y_s = X_s + \text{noise} \)
Making observations

Less uncertain $\Rightarrow$ Reward[$\ P(X|Y_1=\text{hot})\] = 0.2
Making observations

Reward \[ P(X|Y_3=\text{hot}) \] = 0.4
A different outcome...

\[ \text{Reward}[ P(X|Y_3=\text{cold})] = 0.1 \]
Reducing uncertainty

- Want to select observations that maximize reduction in uncertainty

- Can quantify uncertainty using Shannon entropy:
  \[ H(X) = - \sum_x P(X = x) \log_2 P(X = x) \]

- For discrete variables \( 0 \leq H(X) \leq \log_2 |\text{dom}(X)| \)
  - \( P(X = x) = \frac{1}{n} \Rightarrow H(X) = -n \cdot \frac{1}{n} \log_2 \frac{1}{n} = \log_2 n \)

- Thus, can use Reward[ \( P(X) \) ] = \(-H(X) = \sum_x P(x) \log_2 P(x)\)
Making observations

Prior entropy: $H(X) \approx 4.2$
Posterior entropy

Entropy before observations:

\[ H(X) = - \sum_x P(X = x) \log_2 P(X = x) \]

Entropy after observing \( Y = y \):

\[ H(X \mid Y = y) = - \sum_x P(X = x \mid Y = y) \log_2 P(X = x \mid Y = y) \]
Making observations

Posterior entropy: $H(X \mid Y_3 = \text{hot}) \approx 2.7$

Reward: $H(X) - H(X \mid Y_3 = \text{hot}) \approx 1.5$
A different outcome...

Posterior entropy \( H(X \mid Y_3 = \text{cold}) \approx 3.2 \)

Reward: \( H(X) - H(X \mid Y_3 = \text{cold}) \approx 1.0 \)
Information gain

- Entropy after observing $Y = y$:

$$H(X \mid Y = y) = -\sum_x P(X = x \mid Y = y) \log_2 P(X = x \mid Y = y)$$

- Don’t know value of $y$ before observing it!

- Conditional entropy:

$$H(X \mid Y) = \sum_y P(y) H(X \mid Y = y)$$

- Expected information gain (aka mutual information):

$$I(X; Y) = H(X) - H(X \mid Y)$$
Properties of entropy and infogain

\[ P \text{ (mle) } = P(x \mid y) = P(y) \cdot P(y \mid x) \]

\[ H(x \mid y) = H(x) + H(y \mid x) \]

\[ H(x, \ldots, x_n) = H(x_1) + H(y_2(x_2)) + H(x_3 \mid x_2, 1) + \ldots + H(x_n \mid x_1, \ldots, x_{n-1}) \]

\[ H(x) \geq 0 \]

\[ \Rightarrow H(x) \geq H(x \mid y) \quad \text{“information never hurts” (on average)} \]

\[ I(x; y) \geq 0 \]

\[ I(x; y) = 0 \quad \text{if } x \perp y \]

\[ I(x; y) = H(x) - \mathbb{E}[H(x \mid y)] = H(x) + H(y) - \mathbb{E}[H(y \mid x) + H(x \mid y)] \]

\[ = I(y; x) \]
Maximizing information gain

- Given: finite set $V$ of locations
- Want: $A^* \mu V$ such that $A^* = \text{argmax}_{A} F(A)$ with $|A| \leq k$

Typically NP-hard!

Greedy algorithm:
- Start with $A = \{\}$
- For $i = 1$ to $k$
  - $s^* := \text{argmax}_{s} F(A \cup \{s\})$
  - $A := A \cup \{s^*\}$

How well can this simple heuristic do?

$F(A) = I(X; Y_{A})$
Performance of greedy

- Greedy empirically close to optimal. Why?

Temperature data from sensor network
Key observation: Diminishing returns

Placement A = \{Y_1, Y_2\}

Placement B = \{Y_1, ..., Y_5\}

Adding \(Y'\) will help a lot!

Adding \(Y'\) doesn’t help much

New sensor \(Y'\)

Submodularity:

For \(A \mu B\), \(F(A \cup \{Y'\}) - F(A) \geq F(B \cup \{Y'\}) - F(B)\)
One reason submodularity is useful

**Theorem** [Nemhauser et al ‘78]
Greedy algorithm gives constant factor approximation

\[ F(A_{\text{greedy}}) \geq (1-1/e) F(A_{\text{opt}}) \]

- Greedy algorithm gives near-optimal solution!
- Is information gain submodular?
Non-submodularity of information gain

$Y_1, Y_2 \sim \text{Bernoulli}(0.5)$

$X = Y_1 \text{ XOR } Y_2$

Let $F(A) = I(Y_A; X) = H(X) - H(X|Y_A)$

$X \sim B(0.5) \quad H(X) = 1$

$X|Y_1 = y \sim B(0.5) \quad H(X|Y_1) = 1$

$X = Y_1 \text{ XOR } Y_2 \quad H(X|Y_1, Y_2) = 0$

$F(\emptyset) = H(X) - H(X) = 0$

$F(\{Y_1, Y_2\}) = H(X) - H(X|Y_1) = 0$
Example: Submodularity of info-gain

\[ Y_1, \ldots, Y_m, X_1, \ldots, X_n \text{ discrete RVs} \]
\[ F(A) = I(X; X_A) = H(Y) - H(Y \mid X_A) \]

However, NOT always submodular

**Theorem**
If \( Y_i \) are all conditionally independent given \( X \), then \( F(A) \) is submodular!

Hence, greedy algorithm works!

In fact, NO algorithm can do better than \((1-1/e)\) approximation!
Case study: Building a Sensing Chair

- Activity recognition in assistive technologies
- Seating pressure as user interface

Equipped with 1 sensor per cm²!
Costs $6,000!

Can we get similar accuracy with fewer, cheaper sensors?

82% accuracy on 10 postures!
How to place sensors on a chair?

- Sensor readings at locations V as random variables
- Predict posture X using probabilistic model $P(Y,V)$
- Pick sensor locations $A^* \in V$ to minimize entropy:

\[
A^* = \arg\max_{|A| \leq k} I(X; Y_A)
\]

Placed sensors, did a user study:

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before</td>
<td>82%</td>
<td>$6,000</td>
</tr>
</tbody>
</table>
Adaptive Optimization

- So far: Search for a most informative \textit{set} of variables (e.g., sensor placement).
- In many applications, want to adaptively choose observations:

Interested in a \textit{policy} (decision tree), not a \textit{set}. 
Adaptive greedy algorithm

- Expected benefit of adding test $s$ after we’ve seen $Y_A = y_A$.

$$\Delta(s \mid y_A) = H(X \mid y_A) - \sum_{y_s} P(y_s \mid y_A) H(X \mid y_A, y_s)$$

Adaptive Greedy algorithm:

- Start with $A = \emptyset$
- For $i = 1:k$
  - Pick $s_k \in \arg \max_s \Delta(s \mid y_A)$
  - Observe $Y_{s_k} = y_{s_k}$
  - Set $A \leftarrow A \cup \{s_k\}$
Gathering information for making decisions

- So far: Selecting variables which decrease the uncertainty the most

- Often, want to gather information to take the right action
Value of information

Should we raise a fire alert?

<table>
<thead>
<tr>
<th>Actions</th>
<th>Temp. X</th>
<th>Fiery hot</th>
<th>normal/cold</th>
</tr>
</thead>
<tbody>
<tr>
<td>No alarm</td>
<td>-$$\text{$}$$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Raise alarm</td>
<td>$$\text{$}$$</td>
<td>-$$\text{$}$$</td>
<td></td>
</tr>
</tbody>
</table>

Only have belief about temperature $P(X = \text{hot} \mid \text{obs})$

$\Rightarrow$ choose $a^* = \arg\max_a \sum_x P(x \mid \text{obs}) \ U(x,a)$

**Decision theoretic value of (perfect) information**

Reward$[ P(X \mid \text{obs}) ] = \text{MEU}(X \mid \text{obs}) = \max_a \sum_x P(x \mid \text{obs}) \ U(x,a)$
For a set $A$ of variables, its expected value of information is

$$F(A) = \sum_{y_A} P(y_A) \text{MEU}[X \mid y_A]$$

Observations made by sensors $A$ Max. expected utility when observing $Y_A = y_A$

Unfortunately, value of information is not submodular
Greedy algorithm can fail arbitrarily badly
Can do better with look-ahead
Maximizing value of information

[Mathematical expression]

A* = \arg\max_{|A| \leq k} F(A)

Theorem: Complexity of optimizing value of information

For chains (HMMs, etc.)
Optimally solvable in polytime 😊

For trees:
NP^P^P^P complete 😞