EECS 391: Introduction to AI

Soumya Ray

Website: http://engr.case.edu/ray_soumya/eecs391_sp15/
Email: sray@case.edu
Office: Olin 516
Office hours: F 11:30-2pm (email if stopping by between 12:30-2) or by appointment
Bayesian Networks

Pr(B=true) = 0.2

Burglary

Pr(E=true) = 0.3

Earthquake

Alarm

Phone Call

<table>
<thead>
<tr>
<th>Alarm</th>
<th>P=true</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>0.25</td>
</tr>
<tr>
<td>True</td>
<td>0.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Burglary</th>
<th>Earthquake</th>
<th>A=true</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>0.2</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>0.5</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>0.75</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Inference in Bayesian Networks

• How to answer queries $\Pr(V=v \mid E=e)$, given a BN?

• Two kinds of algorithms:
  – Exact
    • Always returns exact answer, but may take a long time
  – Approximate
    • Returns approximate answer. More time=better answers ("anytime")
Variable Elimination

• Suppose we had the BN:

• And we want $Pr(D)$
Variable Elimination

• Steps (elimination ordering A, B, C, D):

\[
\Pr(D) = \sum_{A,B,C} \Pr(A, B, C, D)
\]

\[
= \sum_{A,B,C} \Pr(A) \Pr(B \mid A) \Pr(C \mid B) \Pr(D \mid C)
\]

\[
= \sum_{C} \Pr(D \mid C) \sum_{B} \Pr(C \mid B) \sum_{A} \Pr(B \mid A) \Pr(A)
\]

Each term here is a table, called a “factor”. A factor may not be a probability distribution (though in this case it is). Notice that factors are computed by eliminating variables. The efficiency of VE comes from “pushing in” the sums as far as possible.
Variable Elimination

• **Order the variables** in the network with the variable(s) in the query coming last

• For each elimination variable in the ordering
  – **Multiply all the tables** involving this variable
  – Then **sum out** this variable by adding all the rows where this variable is the only one changing and the others are fixed
  – Store the resulting “factor” or “potential”
Variable Elimination

Pr(B=true)=0.2
Burglary

Pr(E=true)=0.3
Earthquake

Alarm

Phone Call

<table>
<thead>
<tr>
<th>Burglary</th>
<th>Earthquake</th>
<th>A=true</th>
<th>A=false</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Alarm | P=true | P=false

| False | 0.25 | 0.75 |
| True  | 0.75 | 0.25 |

Find $Pr(P)$ using VE
Variable Elimination

• The efficiency of this procedure depends on the order of the variables
  – Finding an optimal order is NP-complete
Incorporating evidence

• If we know the value of a variable, just select that value instead of summing out
Variable Elimination

Find $Pr(P|B)$ using VE

### Pr(B=true) = 0.2
- Burglary

### Pr(E=true) = 0.3
- Earthquake

### Alarm

#### Phone Call

<table>
<thead>
<tr>
<th>Alarm</th>
<th>P=true</th>
<th>P=false</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>0.25</td>
<td>0.75</td>
</tr>
<tr>
<td>True</td>
<td>0.75</td>
<td>0.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Burglary</th>
<th>Earthquake</th>
<th>A=true</th>
<th>A=false</th>
</tr>
</thead>
<tbody>
<tr>
<td>False</td>
<td>False</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>

$Pr(B=true) = 0.2$, $Pr(E=true) = 0.3$
Approximate Inference in BNs
Approximate Inference

• Sometimes a BN can be very complex
• Sometimes we don’t really need the exact probabilities

• In these cases, we can use *sampling* methods to answer queries
  – Often very fast, very easy to implement
  – But convergence is only asymptotic in general
Approximate Inference 1 (Monte Carlo)

• Idea: Topologically sort the variables according to the graph structure
• Sample each according to the conditional distribution (well-defined due to the sorting)
• Count the samples with desired values
• Easy!
  – Right?
Approximate Inference 2: Incorporating Evidence

• What if we have evidence?
• Well, let’s just throw away the samples that have the evidence variables wrong
  – “Rejection Sampling”
Reasoning over time (15.5)

• Suppose an agent needs to reason about how the states of the world evolve over time, but it is uncertain about the states.

• This situation can be represented with *dynamic Bayesian networks (DBNs)*
  – Similar to BNs, but explicit representation of time
    • (Sort of like one BN for each time slice, repeated ad infinitum)
Application: Autonomous Vehicles

• Inference in DBNs can be done with a variation of Monte Carlo sampling, called “Sequential Monte Carlo” or Particle Filtering

• Particle filters are widely used in autonomous vehicles for localization, object detection and tracking

• Also widely used in object recognition in video
Summary

- Bayesian networks
- Exact Inference in BNs through variable elimination
- Approximate inference through Monte Carlo and rejection sampling
- Next: Machine Learning (Ch 18.1-18.2, 20.1-20.2.3)