Bayesian Networks: Approximate Inference

Approaches to inference

- Exact inference
  - Variable elimination
  - Join tree algorithm
- Approximate inference
  - Simplify the structure of the network to make exact inference efficient (variational methods, loopy belief propagation)
- Probabilistic methods
  - Stochastic simulation / sampling methods
  - Markov chain Monte Carlo methods

Network simplification

Typical simplifications:
- Remove parts of the network
- Remove edges
- Reduce the number of values (value abstraction)
- Replace a sub-network with a simpler one (model abstraction)
- These simplifications are often w.r.t. to the particular evidence and query

Inference by sampling

- Want to compute $P(e)$
- Suppose we can sample instances $<x_1,\ldots,x_n>$ according to $P(X_1,\ldots,X_n)$
- The probability that a random sample $<x_1,\ldots,x_n>$ satisfies $e$ is approximately $P(e)$
- We can view each sample as tossing a biased coin with probability $P(e)$ of “Heads”

Sampling a Bayesian Network

- If $P(X_1,\ldots,X_n)$ is represented by a Bayesian network, how can we efficiently sample from it?
- Idea: sample according to structure of the network
  - Write distribution using the chain rule, and then sample each variable given its parents
**BN sampling**

- Let $X_1, \ldots, X_n$ be order of variables consistent with arc direction
- for $i = 1, \ldots, n$ do
  - sample $x_i$ from $P(X_i | Pa(X_i))$
  - (Note: since $Pa(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$, we already assigned values to them)
- return $x_1, \ldots, x_n$

**Sampling a complete instance is linear in number of variables**
- Regardless of structure of the network
- However, if $P(e)$ is small, we need many samples to get a decent estimate
Can we sample from $P(X_1, ..., X_n | e)$?

- If evidence is in roots of network, easily
- If evidence is in leaves of network, we have a problem
  - Our sampling method proceeds according to order of nodes in graph
- Rejection sampling: keep those instantiations that are consistent with the values of the evidence variables
- Estimate $P(X|e)$ by $N(X,e) / N(e)$ where $N(.)$ counts the number of times an event was sampled.

Markov chain Monte Carlo sampling

- Generates events by making random changes to the state variable.
- The next state is generated by sampling a value for one of the nonevidence variables conditioned on the current values.

Gibbs sampling example

- Consider a 2 variable network:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>0.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

- Initialize randomly
- Sample variables alternately

Learning Bayesian Networks

Known Structure, Complete Data

- Network structure is specified
  - Learning algorithm needs to estimate parameters
- Data does not contain missing values
Unknown Structure, Complete Data

- Network structure is not specified
  - Algorithm needs to select edges & estimate parameters
- Data does not contain missing values

Known Structure, Incomplete Data

- Network structure is specified
- Data contains missing values

Known Structure / Complete Data

- Given a network structure G
  - And choice of parametric family for $P(X_i|Pa(X_j))$
- Learn parameters for network

Goal

- Construct a network that is "closest" to probability that generated the data

Benefits of Learning Structure

- Discover structural properties of the domain
  - e.g.: Relevance
- Identifying independencies $\Rightarrow$ faster inference
- Predict effect of actions
  - Involves learning causal relationship among variables

Learning Parameters for a Bayesian Network

- Training data has the form:

$$D = \begin{bmatrix}
\vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}$$

Why Worry about Accurate Structure?

- Increases the number of parameters to be fitted
- Wrong assumptions about causality and domain structure
- Cannot be compensated by accurate fitting of parameters
- Also misses causality and domain structure
Approaches to Learning Structure

- Score based
  - Define a score that evaluates how well the (in)dependencies in a structure match the observations
  - Search for a structure that maximizes the score
- Pros & Cons
  - Statistically motivated
  - Computationally hard

Search for a good structure

- Define a search space:
  - nodes are possible structures
  - edges denote adjacency of structures
  - Traverse this space looking for high-scoring structures
- Search techniques:
  - Greedy hill-climbing
  - Best first search
  - Simulated Annealing

Search (cont.)

- Typical operations:
  - Add C → D
  - Delete C → E
  - Reverse C → E

Greedy Hill-Climbing

- Simplest heuristic local search
  - Start with a network
    - empty network
    - a random network
  - At each iteration
    - Evaluate all possible changes
    - Apply change that leads to best improvement in score
    - Iterate
  - Stop when no modification improves score
  - Each step requires evaluating approximately $n^2$ new changes
  - Involves the standard pitfalls of hill-climbing

Applications of BN

- Medical diagnosis
- Troubleshooting of hardware/software systems