Artificial Intelligence **Ali Raza Syed**
Lecture 6: Probabilistic Reasoning

Introduction

The last lecture introduced probability theory and the notions of independence and conditional independence. We move on to study how these are applied to reasoning given uncertainty.

Bayesian Network

The joint probability distribution specifies all the information about our domain, but it is also hard to apply as it becomes very large with increasing number of variables. A Bayesian network is a more compact representation of the joint probability distribution.

A Bayesian network is a directed acyclic graph (DAG). Each node represents a random variable and an edge connects two dependent nodes. If an arrow points from node X to node Y, we say that X is a parent of Y. This means (informally) that X influences Y.

Given its parents, each node X_i has an associated conditional probability distribution: $P(X_i|Parents(X_i))$. This specifies the effect that $Parents(X_i)$ have on X_i. (Note that, by definition, a DAG has no directed cycles.)

Alarm example

We went over the burglar alarm example (slide 6). An alarm may go off due to a burglary or an earthquake. If the alarm goes off, two neighbors, John and Mary will call the owner. However, John confuses a phone ring with the alarm, while Mary often doesn't hear the alarm at all.

We represent this network with nodes B (Burglary), E (Earthquake), A (Alarm), J (JohnCalls), and M (MaryCalls). B, E are parents of A.

A is a parent of J, M. Each variable has a conditional probability distribution specified in a conditonal probability table (CPT): $P(B)$, $P(E)$, $P(A)$, $P(J)$ and $P(M)$. Each row specifies the node's value given the parents' condititioning values. For Boolean variables, we usually don't specify the second value since $P(\neg a) = 1 - P(a)$.

Through this representation we can cut down the number of probabilities we deal with. Rather than 2^k rows of calculation for a node with k parents (with Boolean values). In the real word, we often find that connections between nodes are sparse, and this makes our calculations relatively easier.

Meaning of Bayesian networks

The network allows us to calculate probability distributions as a product of the the conditional probabilities. This factorization allows us to simply read off the joint probability distribution: $P(x_1, \ldots, x_n) = \prod P(x_i|parents(X_i))$. These are the global semantics of the Bayesian network(slide 8).

Each node is conditionally independent of its non-descendants given the parents. These are the local semantics of the network. Once we fix u_t, u_m, knowing z has no effect on x (slide 9). Judea Pearl showed that these two semantics are equivalent. A Markov blanket for a node (slide 10) consists of the node's parents, children and children's parents. Given the Markov blanket, the node is conditionally independent of all other nodes.

Inference

Given an observed event (e), we want to find the posterior probability distribution for the variables of interest (X), $P(X|e)$. The remaining variables we don't observe (y) are called hidden variables. We can find this by computing the sum of the factored products of conditional probabilities in the network: $P(X|e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$.

However, the resulting algorithm has a high coplexity. With n boolean variables, the complexity is $O(n2^n)$. This can be improved with some optimizations. Constant terms can be removed from the summation, then using depth first recursion to evaluate the expressions at each node. There are other techniques that can be employed such as caching and clustering (slide 21, and section 14.4 in the book for details).

Approximate Inference

Since exact inference can be computationally intractable, we can resort to approximation methods using stochastic simulation. Directly sampling from the distribution at the nodes, we can traverse the network and obtain probability distributions for our variables of interest (slide 31).

Rejection sampling simulates values from prior distributions and rejects outcomes that do no correspond to what was observed. The final estimate is obtained by counting the values for the variable of interest. A complete algorithm is presented in figure 14.13.

Importance sampling fixes the values for the observed variables and simulates the relevant variables. The final counts are weighted by likelihood of the event, which is the product of conditional probabilities for the evidence variables. The algorithm is presented in figure 14.14 and an example on slides 36-38.