Bayesian network

A Bayesian network (also known as a Bayes network, belief network, or decision network) is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Bayesian networks are ideal for taking an event that occurred and predicting the likelihood that any one of several possible known causes was the contributing factor. For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases.

Efficient algorithms can perform inference and learning in Bayesian networks. Bayesian networks that model sequences of variables (e.g. speech signals or protein sequences) are called dynamic Bayesian networks. Generalizations of Bayesian networks that can represent and solve decision problems under uncertainty are called influence diagrams.

Graphical model

Formally, Bayesian networks are directed acyclic graphs (DAGs) whose nodes represent variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes that are not connected (no path connects one node to another) represent variables that are conditionally independent of each other. Each node is associated with a probability function that takes, as input, a particular set of values for the node's parent variables, and gives (as output) the probability (or probability distribution, if applicable) of the variable represented by the node. For example, if {\displaystyle m}m parent nodes represent {\ displaystyle m}m Boolean variables, then the probability function could be represented by a table of {\displaystyle 2^{m}}2^{m} entries, one entry for each of the {\displaystyle 2^{m}}2^{m} possible parent combinations. Similar ideas may be applied to undirected, and possibly cyclic, graphs such as Markov networks.

Inferring unobserved variables

Because a Bayesian network is a complete model for its variables and their relationships, it can be used to answer probabilistic queries about them. For example, the network can be used to update knowledge of the state of a subset of variables when other variables (the evidence variables) are observed. This process of computing the posterior distribution of variables given evidence is called probabilistic inference. The posterior gives a universal sufficient statistic for detection applications, when choosing values for the variable subset that minimize some expected loss function, for instance the probability of decision error. A Bayesian network can thus be considered a mechanism for automatically applying Bayes' theorem to complex problems.

The most common exact inference methods are: variable elimination, which eliminates (by integration or summation) the non-observed non-query variables one by one by distributing the sum over the product; clique tree propagation, which caches the computation so that

many variables can be queried at one time and new evidence can be propagated quickly; and recursive conditioning and AND/OR search, which allow for a space–time tradeoff and match the efficiency of variable elimination when enough space is used. All of these methods have complexity that is exponential in the network's treewidth. The most common approximate inference algorithms are importance sampling, stochastic MCMC simulation, mini-bucket elimination, loopy belief propagation, generalized belief propagation and variational methods.

Parameter learning

In order to fully specify the Bayesian network and thus fully represent the joint probability distribution, it is necessary to specify for each node X the probability distribution for X conditional upon X's parents. The distribution of X conditional upon its parents may have any form. It is common to work with discrete or Gaussian distributions since that simplifies calculations. Sometimes only constraints on a distribution are known; one can then use the principle of maximum entropy to determine a single distribution, the one with the greatest entropy given the constraints. (Analogously, in the specific context of a dynamic Bayesian network, the conditional distribution for the hidden state's temporal evolution is commonly specified to maximize the entropy rate of the implied stochastic process.)

Often these conditional distributions include parameters that are unknown and must be estimated from data, e.g., via the maximum likelihood approach. Direct maximization of the likelihood (or of the posterior probability) is often complex given unobserved variables. A classical approach to this problem is the expectation-maximization algorithm, which alternates computing expected values of the unobserved variables conditional on observed data, with maximizing the complete likelihood (or posterior) assuming that previously computed expected values are correct. Under mild regularity conditions this process converges on maximum likelihood (or maximum posterior) values for parameters.

A more fully Bayesian approach to parameters is to treat them as additional unobserved variables and to compute a full posterior distribution over all nodes conditional upon observed data, then to integrate out the parameters. This approach can be expensive and lead to large dimension models, making classical parameter-setting approaches more tractable.

ref. https://en.wikipedia.org/wiki/Bayesian_network