WIKIPEDIA

Bayesian network

A **Bayesian network** (also known as a **Bayes network**, **Bayes net**, **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.

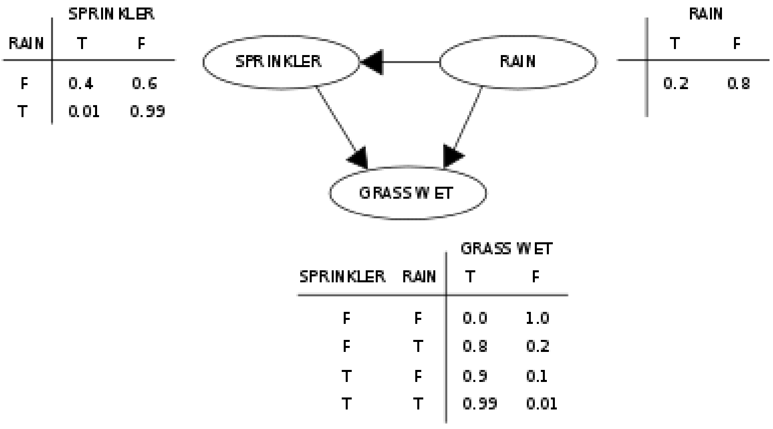
Contents

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 m parent nodes represent m Boolean variables, then the probability function could be represented by a table of 2^m entries, one entry for each of the 2^m possible parent combinations. Similar ideas may be applied to undirected, and possibly cyclic, graphs such as Markov networks.

Example

Two events can cause grass to be wet: an active sprinkler or rain. Rain has a direct effect on the use of the sprinkler (namely that when it rains, the sprinkler usually is not active). This situation can be modeled with a Bayesian network (shown to the right). Each variable has two possible values, T (for true) and F (for false).



	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

A simple Bayesian network with conditional probability tables

The joint probability function is, by the chain rule of probability,

$$\Pr(G,S,R)=\Pr(G\mid S,R)\Pr(S\mid R)\Pr(R)$$

where *G* = "Grass wet (true/false)", *S* = "Sprinkler turned on (true/false)", and *R* = "Raining (true/false)".

The model can answer questions about the presence of a cause given the presence of an effect (so-called inverse probability) like "What is the probability that it is raining, given the grass is wet?" by using the conditional probability formula and summing over all nuisance variables:

$$\Pr(R=T\mid G=T)=\frac{\Pr(G=T,R=T)}{\Pr(G=T)}=\frac{\sum_{x\in\{T,F\}}\Pr(G=T,S=x,R=T)}{\sum_{x,y\in\{T,F\}}\Pr(G=T,S=x,R=y)}$$

Using the expansion for the joint probability function $\Pr(G,S,R)$ and the conditional probabilities from the conditional probability tables (CPTs) stated in the diagram, one can evaluate each term in the sums in the numerator and denominator. For example,

$$\begin{aligned}\Pr(G=T,S=T,R=T)&=\Pr(G=T\mid S=T,R=T)\Pr(S=T\mid R=T)\Pr(R=T)\\&=0.99\times 0.01\times 0.2\\&=0.00198.\end{aligned}$$

Then the numerical results (subscripted by the associated variable values) are

$$\Pr(R=T\mid G=T)=\frac{0.00198_{TTT}+0.1584_{TFT}}{0.00198_{TTT}+0.288_{TTF}+0.1584_{TFT}+0.0_{TFF}}=\frac{891}{2491}\approx 35.77\%.$$

To answer an interventional question, such as "What is the probability that it would rain, given that we wet the grass?" the answer is governed by the post-intervention joint distribution function

$$\Pr(S,R\mid \text{do}(G=T))=\Pr(S\mid R)\Pr(R)$$

obtained by removing the factor $\Pr(G\mid S,R)$ from the pre-intervention distribution. The do operator forces the value of G to be true. The probability of rain is unaffected by the action:

$$\Pr(R\mid \text{do}(G=T))=\Pr(R).$$

To predict the impact of turning the sprinkler on:

$$\Pr(R,G\mid \text{do}(S=T))=\Pr(R)\Pr(G\mid R,S=T)$$

with the term $\Pr(S=T\mid R)$ removed, showing that the action affects the grass but not the rain.

These predictions may not be feasible given unobserved variables, as in most policy evaluation problems. The effect of the action $\text{do}(x)$ can still be predicted, however, whenever the back-door criterion is satisfied.[1][2] It states that, if a set *Z* of nodes can be observed that *d*-separates[3] (or blocks) all back-door paths from *X* to *Y* then

$$\Pr(Y,Z\mid \text{do}(x))=\frac{\Pr(Y,Z,X=x)}{\Pr(X=x\mid Z)}.$$

A back-door path is one that ends with an arrow into *X*. Sets that satisfy the back-door criterion are called "sufficient" or "admissible." For example, the set *Z* = *R* is admissible for predicting the effect of *S* = *T* on *G*, because *R* *d*-separates the (only) back-door path *S* ← *R* → *G*. However, if *S* is not observed, no other set *d*-separates this path and the effect of turning the sprinkler on (*S* = *T*) on the grass (*G*) cannot be predicted from passive observations. In that case *P*(*G* | do(*S* = *T*)) is not "identified". This reflects the fact that, lacking interventional data, the observed dependence between *S* and *G* is due to a causal connection or is spurious (apparent dependence arising from a common cause, *R*). (see Simpson's paradox)

To determine whether a causal relation is identified from an arbitrary Bayesian network with unobserved variables, one can use the three rules of "*do*-calculus"[1][4] and test whether all *do* terms can be removed from the expression of that relation, thus confirming that the desired quantity is estimable from frequency data.[5]

Using a Bayesian network can save considerable amounts of memory over exhaustive probability tables, if the dependencies in the joint distribution are sparse. For example, a naive way of storing the conditional probabilities of 10 two-valued variables as a table requires storage space for $2^{10}=1024$ values. If no variable's local distribution depends on more than three parent variables, the Bayesian network representation stores at most $10\cdot 2^3=80$ values.

One advantage of Bayesian networks is that it is intuitively easier for a human to understand (a sparse set of) direct dependencies and local distributions than complete joint distributions.

Inference and learning

Bayesian networks perform three main inference tasks:

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 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.

Structure learning

In the simplest case, a Bayesian network is specified by an expert and is then used to perform inference. In other applications, the task of defining the network is too complex for humans. In this case, the network structure and the parameters of the local distributions must be learned from data.

Automatically learning the graph structure of a Bayesian network (BN) is a challenge pursued within machine learning. The basic idea goes back to a recovery algorithm developed by Rebane and Pearl[6] and rests on the distinction between the three possible patterns allowed in a 3-node DAG:

Junction patterns

Pattern	Model
Chain	$X \rightarrow Y \rightarrow Z$
Fork	$X \leftarrow Y \rightarrow Z$
Collider	$X \rightarrow Y \leftarrow Z$

The first 2 represent the same dependencies (X and Z are independent given Y) and are, therefore, indistinguishable. The collider, however, can be uniquely identified, since X and Z are marginally independent and all other pairs are dependent. Thus, while the *skeletons* (the graphs stripped of arrows) of these three triplets are identical, the directionality of the arrows is partially identifiable. The same distinction applies when X and Z have common parents, except that one must first condition on those parents. Algorithms have been developed to systematically determine the skeleton of the underlying graph and, then, orient all arrows whose directionality is dictated by the conditional independences observed.[1][7][8][9]

An alternative method of structural learning uses optimization-based search. It requires a scoring function and a search strategy. A common scoring function is posterior probability of the structure given the training data, like the BIC or the BDeu. The time requirement of an exhaustive search returning a structure that maximizes the score is superexponential in the number of variables. A local search strategy makes incremental changes aimed at improving the score of the structure. A global search algorithm like Markov chain Monte Carlo can avoid getting trapped in local minima. Friedman et al.[10][11] discuss using mutual information between variables and finding a structure that maximizes this. They do this by restricting the parent candidate set to *k* nodes and exhaustively searching therein.

A particularly fast method for exact BN learning is to cast the problem as an optimization problem, and solve it using integer programming. Acyclicity constraints are added to the integer program (IP) during solving in the form of cutting planes.[12] Such method can handle problems with up to 100 variables.

In order to deal with problems with thousands of variables, a different approach is necessary. One is to first sample one ordering, and then find the optimal BN structure with respect to that ordering. This implies working on the search space of the possible orderings, which is convenient as it is smaller than the space of network structures. Multiple orderings are then sampled and evaluated. This method has been proven to be the best available in literature when the number of variables is huge.[13]

Another method consists of focusing on the sub-class of decomposable models, for which the MLE have a closed form. It is then possible to discover a consistent structure for hundreds of variables.[14]

Learning Bayesian networks with bounded treewidth is necessary to allow exact, tractable inference, since the worst-case inference complexity is exponential in the treewidth k (under the exponential time hypothesis). Yet, as a global property of the graph, it considerably increases the difficulty of the learning process. In this context it is possible to use K-tree for effective learning.[15]

Statistical introduction

Given data x and parameter θ, a simple Bayesian analysis starts with a prior probability (*prior*) $p(\theta)$ and likelihood $p(x \mid \theta)$ to compute a posterior probability $p(\theta \mid x) \propto p(x \mid \theta) p(\theta)$.

Often the prior on θ depends in turn on other parameters φ that are not mentioned in the likelihood. So, the prior $p(\theta)$ must be replaced by a likelihood $p(\theta \mid \varphi)$, and a prior $p(\varphi)$ on the newly introduced parameters φ is required, resulting in a posterior probability

$$p(\theta, \varphi \mid x) \propto p(x \mid \theta)p(\theta \mid \varphi)p(\varphi).$$

This is the simplest example of a *hierarchical Bayes model*.

The process may be repeated; for example, the parameters φ may depend in turn on additional parameters ψ , which require their own prior. Eventually the process must terminate, with priors that do not depend on unmentioned parameters.

Introductory examples

Given the measured quantities $x_1, \dots, x_n$ each with normally distributed errors of known standard deviation σ,

$$x_i \sim N(\theta_i, \sigma^2)$$

Suppose we are interested in estimating the θ_i. An approach would be to estimate the θ_i using a maximum likelihood approach; since the observations are independent, the likelihood factorizes and the maximum likelihood estimate is simply

$$\theta_i = x_i.$$

However, if the quantities are related, so that for example the individual θ_i have themselves been drawn from an underlying distribution, then this relationship destroys the independence and suggests a more complex model, e.g.,

$$\begin{aligned} x_i &\sim N(\theta_i, \sigma^2), \\ \theta_i &\sim N(\varphi, \tau^2), \end{aligned}$$

with improper priors $\varphi \sim \text{flat}$, $\tau \sim \text{flat} \in (0, \infty)$. When $n \geq 3$, this is an *identified model* (i.e. there exists a unique solution for the model's parameters), and the posterior distributions of the individual θ_i will tend to move, or *shrink* away from the maximum likelihood estimates towards their common mean. This *shrinkage* is a typical behavior in hierarchical Bayes models.

Restrictions on priors

Some care is needed when choosing priors in a hierarchical model, particularly on scale variables at higher levels of the hierarchy such as the variable τ in the example. The usual priors such as the Jeffreys prior often do not work, because the posterior distribution will not be normalizable and estimates made by minimizing the expected loss will be inadmissible.

Definitions and concepts

Several equivalent definitions of a Bayesian network have been offered. For the following, let *G* = (*V*,*E*) be a directed acyclic graph (DAG) and let *X* = (X_v), *v* ∈ *V* be a set of random variables indexed by *V*.

Factorization definition

X is a Bayesian network with respect to G if its joint probability density function (with respect to a product measure) can be written as a product of the individual density functions, conditional on their parent variables:[16]

$$p(x)=\prod_{v\in V} p\left(x_v \,\big|\, x_{\operatorname{pa}(v)}\right)$$

where pa(v) is the set of parents of v (i.e. those vertices pointing directly to v via a single edge).

For any set of random variables, the probability of any member of a joint distribution can be calculated from conditional probabilities using the chain rule (given a topological ordering of X) as follows:[16]

$$\operatorname{P}(X_1=x_1,\ldots,X_n=x_n)=\prod_{v=1}^n \operatorname{P}(X_v=x_v \mid X_{v+1}=x_{v+1},\ldots,X_n=x_n)$$

Using the definition above, this can be written as:

$$\operatorname{P}(X_1=x_1,\ldots,X_n=x_n)=\prod_{v=1}^n \operatorname{P}(X_v=x_v \mid X_j=x_j \text{ for each } X_j\, \text{ that is a parent of } X_v\,)$$

The difference between the two expressions is the conditional independence of the variables from any of their non-descendants, given the values of their parent variables.

Local Markov property

X is a Bayesian network with respect to G if it satisfies the *local Markov property*: each variable is conditionally independent of its non-descendants given its parent variables:[17]

$$X_v \perp\!\!\!\perp X_{V \,\smallsetminus\, \operatorname{de}(v)} \mid X_{\operatorname{pa}(v)} \quad\text{for all } v\in V$$

where de(v) is the set of descendants and V \ de(v) is the set of non-descendants of v.

This can be expressed in terms similar to the first definition, as

$$\begin{aligned}&\operatorname{P}(X_v=x_v \mid X_i=x_i \text{ for each } X_i \text{ that is not a descendant of } X_v\,)\\={}&P(X_v=x_v \mid X_j=x_j \text{ for each } X_j \text{ that is a parent of } X_v\,)\end{aligned}$$

The set of parents is a subset of the set of non-descendants because the graph is acyclic.

Developing Bayesian networks

Developing a Bayesian network often begins with creating a DAG G such that X satisfies the local Markov property with respect to G. Sometimes this is a causal DAG. The conditional probability distributions of each variable given its parents in G are assessed. In many cases, in particular in the case where the variables are discrete, if the joint distribution of X is the product of these conditional distributions, then X is a Bayesian network with respect to G.[18]

Markov blanket

The Markov blanket of a node is the set of nodes consisting of its parents, its children, and any other parents of its children. The Markov blanket renders the node independent of the rest of the network; the joint distribution of the variables in the Markov blanket of a node is sufficient knowledge for calculating the distribution of the node. *X* is a Bayesian network with respect to *G* if every node is conditionally independent of all other nodes in the network, given its Markov blanket.[17]

d-separation

This definition can be made more general by defining the "d"-separation of two nodes, where d stands for directional.[1] We first define the "d"-separation of a trail and then we will define the "d"-separation of two nodes in terms of that.

Let *P* be a trail from node *u* to *v*. A trail is a loop-free, undirected (i.e. all edge directions are ignored) path between two nodes. Then *P* is said to be *d*-separated by a set of nodes *Z* if any of the following conditions holds:

- *P* contains (but does not need to be entirely) a directed chain, $u\cdots\leftarrow m\leftarrow\cdots v$ or $u\cdots\rightarrow m\rightarrow\cdots v$, such that the middle node *m* is in *Z*,
- *P* contains a fork, $u\cdots\leftarrow m\rightarrow\cdots v$, such that the middle node *m* is in *Z*, or
- *P* contains an inverted fork (or collider), $u\cdots\rightarrow m\leftarrow\cdots v$, such that the middle node *m* is not in *Z* and no descendant of *m* is in *Z*.

The nodes *u* and *v* are *d*-separated by *Z* if all trails between them are *d*-separated. If *u* and *v* are not d-separated, they are d-connected.

X is a Bayesian network with respect to *G* if, for any two nodes *u*, *v*:

$$X_u \perp\!\!\!\perp X_v \mid X_Z$$

where *Z* is a set which *d*-separates *u* and *v*. (The Markov blanket is the minimal set of nodes which *d*-separates node *v* from all other nodes.)

Causal networks

Although Bayesian networks are often used to represent causal relationships, this need not be the case: a directed edge from *u* to *v* does not require that X_v be causally dependent on X_u. This is demonstrated by the fact that Bayesian networks on the graphs:

$$a\rightarrow b\rightarrow c \qquad \text{and} \qquad a\leftarrow b\leftarrow c$$

are equivalent: that is they impose exactly the same conditional independence requirements.

A causal network is a Bayesian network with the requirement that the relationships be causal. The additional semantics of causal networks specify that if a node *X* is actively caused to be in a given state *x* (an action written as do(*X* = *x*)), then the probability density function changes to that of the network obtained by cutting the links from the parents of *X* to *X*, and setting *X* to the caused value *x*.[1] Using these semantics, the impact of external interventions from data obtained prior to intervention can be predicted.

Inference complexity and approximation algorithms

In 1990, while working at Stanford University on large bioinformatic applications, Cooper proved that exact inference in Bayesian networks is NP-hard.[19] This result prompted research on approximation algorithms with the aim of developing a tractable approximation to probabilistic inference. In 1993, Paul Dagum and Michael Luby proved two surprising results on the complexity of approximation of probabilistic inference in Bayesian networks.[20] First, they proved that no tractable deterministic algorithm can approximate probabilistic inference to within an absolute error $\varepsilon < 1/2$. Second, they proved that no tractable randomized algorithm can approximate probabilistic inference to within an absolute error $\varepsilon < 1/2$ with confidence probability greater than 1/2.

At about the same time, Roth proved that exact inference in Bayesian networks is in fact #P-complete (and thus as hard as counting the number of satisfying assignments of a conjunctive normal form formula (CNF)) and that approximate inference within a factor $2^{n^{1-\varepsilon}}$ for every $\varepsilon > 0$, even for Bayesian networks with restricted architecture, is NP-hard.[21][22]

In practical terms, these complexity results suggested that while Bayesian networks were rich representations for AI and machine learning applications, their use in large real-world applications would need to be tempered by either topological structural constraints, such as naïve Bayes networks, or by restrictions on the conditional probabilities. The bounded variance algorithm[23] developed by Dagum and Luby was the first provable fast approximation algorithm to efficiently approximate probabilistic inference in Bayesian networks with guarantees on the error approximation. This powerful algorithm required the minor restriction on the conditional probabilities of the Bayesian network to be bounded away from zero and one by $1/p(n)$ where $p(n)$ was any polynomial on the number of nodes in the network n.

Software

Notable software for Bayesian networks include:

- Just another Gibbs sampler (JAGS) – Open-source alternative to WinBUGS. Uses Gibbs sampling.
- OpenBUGS – Open-source development of WinBUGS.
- SPSS Modeler – Commercial software that includes an implementation for Bayesian networks.
- Stan (software) – Stan is an open-source package for obtaining Bayesian inference using the No-U-Turn sampler (NUTS),[24] a variant of Hamiltonian Monte Carlo.
- PyMC3 – A Python library implementing an embedded domain specific language to represent bayesian networks, and a variety of samplers (including NUTS)
- WinBUGS – One of the first computational implementations of MCMC samplers. No longer maintained.

History

The term Bayesian network was coined by Judea Pearl in 1985 to emphasize:[25]

- the often subjective nature of the input information
- the reliance on Bayes' conditioning as the basis for updating information
- the distinction between causal and evidential modes of reasoning[26]

In the late 1980s Pearl's *Probabilistic Reasoning in Intelligent Systems*[27] and Neapolitan's *Probabilistic Reasoning in Expert Systems*[28] summarized their properties and established them as a field of study.

See also

- Bayesian epistemology
- Bayesian programming
- Causal inference
- Causal loop diagram
- Chow–Liu tree
- Computational intelligence
- Computational phylogenetics
- Deep belief network
- Dempster–Shafer theory – a generalization of Bayes' theorem
- Expectation–maximization algorithm
- Factor graph
- Hierarchical temporal memory
- Kalman filter
- Memory-prediction framework
- Mixture distribution
- Mixture model
- Naive Bayes classifier
- Polytree
- Sensor fusion
- Sequence alignment
- Structural equation modeling
- Subjective logic
- Variable-order Bayesian network

Notes

1. Pearl, Judea (2000). *Causality: Models, Reasoning, and Inference* (https://books.google.com/books?id=LLkhAwAAQBAJ). Cambridge University Press. ISBN 978-0-521-77362-1. OCLC 42291253 (https://www.worldcat.org/oclc/42291253).
2. "The Back-Door Criterion" (http://bayes.cs.ucla.edu/BOOK-2K/ch3-3.pdf) (PDF). Retrieved 2014-09-18.
3. "d-Separation without Tears" (http://bayes.cs.ucla.edu/BOOK-09/ch11-1-2-final.pdf) (PDF). Retrieved 2014-09-18.
4. Pearl J (1994). "A Probabilistic Calculus of Actions" (http://dl.acm.org/ft_gateway.cfm?id=2074452&ftid=1062250&dwn=1&CFID=161588115&CFTOKEN=10243006). In Lopez de Mantaras R, Poole D (eds.). *UAI'94 Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*. San Mateo CA: Morgan Kaufmann. pp. 454–462. arXiv:1302.6835 (https://arxiv.org/abs/1302.6835). Bibcode:2013arXiv1302.6835P (https://ui.adsabs.harvard.edu/abs/2013arXiv1302.6835P). ISBN 1-55860-332-8.
5. Shpitser I, Pearl J (2006). "Identification of Conditional Interventional Distributions". In Dechter R, Richardson TS (eds.). *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*. Corvallis, OR: AUAI Press. pp. 437–444. arXiv:1206.6876 (https://arxiv.org/abs/1206.6876).
6. Rebane G, Pearl J (1987). "The Recovery of Causal Poly-trees from Statistical Data". *Proceedings, 3rd Workshop on Uncertainty in AI*. Seattle, WA. pp. 222–228. arXiv:1304.2736 (https://arxiv.org/abs/1304.2736).

7. Spirtes P, Glymour C (1991). "An algorithm for fast recovery of sparse causal graphs" (http://repository.cmu.edu/cgi/viewcontent.cgi?article=1316&context=philosophy) (PDF). *Social Science Computer Review*. **9** (1): 62–72. doi:10.1177/089443939100900106 (https://doi.org/10.1177%2F089443939100900106). S2CID 38398322 (https://api.semanticscholar.org/CorpusID:38398322).
8. Spirtes P, Glymour CN, Scheines R (1993). *Causation, Prediction, and Search* (https://books.google.com/books?id=VkawQgAACAAJ) (1st ed.). Springer-Verlag. ISBN 978-0-387-97979-3.
9. Verma T, Pearl J (1991). "Equivalence and synthesis of causal models" (https://books.google.com/books?id=ikuuHAAACAAJ). In Bonissone P, Henrion M, Kanal LN, Lemmer JF (eds.). *UAI '90 Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*. Elsevier. pp. 255–270. ISBN 0-444-89264-8.
10. Friedman N, Geiger D, Goldszmidt M (November 1997). "Bayesian Network Classifiers" (https://doi.org/10.1023%2FA%3A1007465528199). *Machine Learning*. **29** (2–3): 131–163. doi:10.1023/A:1007465528199 (https://doi.org/10.1023%2FA%3A1007465528199).
11. Friedman N, Linial M, Nachman I, Pe'er D (August 2000). "Using Bayesian networks to analyze expression data". *Journal of Computational Biology*. **7** (3–4): 601–20. CiteSeerX 10.1.1.191.139 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.191.139). doi:10.1089/106652700750050961 (https://doi.org/10.1089%2F106652700750050961). PMID 11108481 (https://pubmed.ncbi.nlm.nih.gov/11108481).
12. Cussens J (2011). "Bayesian network learning with cutting planes" (https://dslpitt.org/papers/11/p153-cussens.pdf) (PDF). *Proceedings of the 27th Conference Annual Conference on Uncertainty in Artificial Intelligence*: 153–160. arXiv:1202.3713 (https://arxiv.org/abs/1202.3713). Bibcode:2012arXiv1202.3713C (https://ui.adsabs.harvard.edu/abs/2012arXiv1202.3713C).
13. Scanagatta M, de Campos CP, Corani G, Zaffalon M (2015). "Learning Bayesian Networks with Thousands of Variables" (https://papers.nips.cc/paper/5803-learning-bayesian-networks-with-thousands-of-variables). *NIPS-15: Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates. pp. 1855–1863.
14. Petitjean F, Webb GI, Nicholson AE (2013). *Scaling log-linear analysis to high-dimensional data* (http://www.tiny-clues.eu/Research/Petitjean2013-ICDM.pdf) (PDF). International Conference on Data Mining. Dallas, TX, USA: IEEE.
15. M. Scanagatta, G. Corani, C. P. de Campos, and M. Zaffalon. Learning Treewidth-Bounded Bayesian Networks with Thousands of Variables. (http://papers.nips.cc/paper/6232-learning-treewidth-bounded-bayesian-networks-with-thousands-of-variables) In NIPS-16: Advances in Neural Information Processing Systems 29, 2016.
16. Russell & Norvig 2003, p. 496.
17. Russell & Norvig 2003, p. 499.
18. Neapolitan RE (2004). *Learning Bayesian networks* (https://books.google.com/books?id=OlMZAQAAIAAJ). Prentice Hall. ISBN 978-0-13-012534-7.
19. Cooper GF (1990). "The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks" (https://stat.duke.edu/~sayan/npcomplete.pdf) (PDF). *Artificial Intelligence*. **42** (2–3): 393–405. doi:10.1016/0004-3702(90)90060-d (https://doi.org/10.1016%2F0004-3702%2890%2990060-d).
20. Dagum P, Luby M (1993). "Approximating probabilistic inference in Bayesian belief networks is NP-hard". *Artificial Intelligence*. **60** (1): 141–153. CiteSeerX 10.1.1.333.1586 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.333.1586). doi:10.1016/0004-3702(93)90036-b (https://doi.org/10.1016%2F0004-3702%2893%2990036-b).
21. D. Roth, On the hardness of approximate reasoning (http://cogcomp.cs.illinois.edu/page/publication_view/5), IJCAI (1993)

22. D. Roth, On the hardness of approximate reasoning (http://cogcomp.cs.illinois.edu/papers/hardJ.pdf), Artificial Intelligence (1996)
23. Dagum P, Luby M (1997). "An optimal approximation algorithm for Bayesian inference" (https://web.archive.org/web/20170706064354/http://www1.icsi.berkeley.edu/~luby/PAPERS/bayesian.ps). *Artificial Intelligence*. **93** (1–2): 1–27. CiteSeerX 10.1.1.36.7946 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.7946). doi:10.1016/s0004-3702(97)00013-1 (https://doi.org/10.1016%2Fs0004-3702%2897%2900013-1). Archived from the original (http://icsi.berkeley.edu/~luby/PAPERS/bayesian.ps) on 2017-07-06. Retrieved 2015-12-19.
24. Hoffman, Matthew D.; Gelman, Andrew (2011). "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". arXiv:1111.4246 (https://arxiv.org/abs/1111.4246). Bibcode:2011arXiv1111.4246H (https://ui.adsabs.harvard.edu/abs/2011arXiv1111.4246H).
25. Pearl J (1985). *Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning* (http://ftp.cs.ucla.edu/tech-report/198_-reports/850017.pdf) (UCLA Technical Report CSD-850017). Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA. pp. 329–334. Retrieved 2009-05-01.
26. Bayes T, Price (1763). "An Essay towards solving a Problem in the Doctrine of Chances". *Philosophical Transactions of the Royal Society*. **53**: 370–418. doi:10.1098/rstl.1763.0053 (https://doi.org/10.1098%2Frstl.1763.0053).
27. Pearl J (1988-09-15). *Probabilistic Reasoning in Intelligent Systems* (https://books.google.com/books?id=AvNID7LyMusC). San Francisco CA: Morgan Kaufmann. p. 1988. ISBN 978-1558604797.
28. Neapolitan RE (1989). *Probabilistic reasoning in expert systems: theory and algorithms* (https://books.google.com/books?id=7X5KLwEACAAJ). Wiley. ISBN 978-0-471-61840-9.

References

- Ben Gal I (2007). "Bayesian Networks" (http://www.eng.tau.ac.il/~bengal/BN.pdf) (PDF). In Ruggeri F, Kennett RS, Faltin FW (eds.). *Support-Page*. *Encyclopedia of Statistics in Quality and Reliability*. John Wiley & Sons. doi:10.1002/9780470061572.eqr089 (https://doi.org/10.1002%2F9780470061572.eqr089). ISBN 978-0-470-01861-3.
- Bertsch McGrayne S (2011). *The Theory That Would not Die* (https://archive.org/details/theorythatwouldn0000mcgr). New Haven: Yale University Press.
- Borgelt C, Kruse R (March 2002). *Graphical Models: Methods for Data Analysis and Mining* (http://fuzzy.cs.uni-magdeburg.de/books/gm/). Chichester, UK: Wiley. ISBN 978-0-470-84337-6.
- Borsuk ME (2008). "Ecological informatics: Bayesian networks". In Jørgensen, Sven Erik, Fath, Brian (eds.). *Encyclopedia of Ecology*. Elsevier. ISBN 978-0-444-52033-3.
- Castillo E, Gutiérrez JM, Hadi AS (1997). "Learning Bayesian Networks". *Expert Systems and Probabilistic Network Models*. Monographs in computer science. New York: Springer-Verlag. pp. 481–528. ISBN 978-0-387-94858-4.
- Comley JW, Dowe DL (June 2003). "General Bayesian networks and asymmetric languages" (http://www.csse.monash.edu.au/~dld/David.Dowe.publications.html#ComleyDowe2003). *Proceedings of the 2nd Hawaii International Conference on Statistics and Related Fields*.
- Comley JW, Dowe DL (2005). "Minimum Message Length and Generalized Bayesian Nets with Asymmetric Languages" (http://www.csse.monash.edu.au/~dld/David.Dowe.publications.html#ComleyDowe2005). In Grünwald PD, Myung IJ, Pitt MA (eds.). *Advances in Minimum Description Length: Theory and Applications*. Neural information processing series. Cambridge, Massachusetts: Bradford Books (MIT Press) (published April 2005). pp. 265–294. ISBN 978-0-262-07262-5. (This paper puts decision trees in internal nodes of Bayes

networks using Minimum Message Length (http://www.csse.monash.edu.au/~dld/MML.html) (MML).

- Darwiche A (2009). *Modeling and Reasoning with Bayesian Networks* (http://www.cambridge.org/9780521884389). Cambridge University Press. ISBN 978-0521884389.
- Dowe, David L. (2011-05-31). "Hybrid Bayesian network graphical models, statistical consistency, invariance and uniqueness" (http://www.csse.monash.edu.au/~dld/Publications/2010/Dowe2010_MML_HandbookPhilSci_Vol7_HandbookPhilStat_MML+hybridBayesianNetworkGraphicalModels+StatisticalConsistency+InvarianceAndUniqueness_pp901-982.pdf) (PDF). *Philosophy of Statistics* (https://books.google.com/books?id=mPG5RupkTX0C). Elsevier. pp. 901–982 (http://www.csse.monash.edu.au/~dld/Publications/2010/Dowe2010_MML_HandbookPhilSci_Vol7_HandbookPhilStat_MML+hybridBayesianNetworkGraphicalModels+StatisticalConsistency+InvarianceAndUniqueness_pp901-982.pdf). ISBN 9780080930961.
- Fenton N, Neil ME (November 2007). "Managing Risk in the Modern World: Applications of Bayesian Networks" (https://web.archive.org/web/20080514044436/http://www.agenarisk.com/resources/apps_bayesian_networks.pdf) (PDF). *A Knowledge Transfer Report from the London Mathematical Society and the Knowledge Transfer Network for Industrial Mathematics*. London (England): London Mathematical Society. Archived from the original (http://www.agenarisk.com/resources/apps_bayesian_networks.pdf) (PDF) on 2008-05-14. Retrieved 2008-10-29.
- Fenton N, Neil ME (July 23, 2004). "Combining evidence in risk analysis using Bayesian Networks" (https://web.archive.org/web/20070927153751/https://www.dcs.qmul.ac.uk/~norman/papers/Combining%20evidence%20in%20risk%20analysis%20using%20BNs.pdf) (PDF). *Safety Critical Systems Club Newsletter*. Vol. 13, no. 4. Newcastle upon Tyne, England. pp. 8–13. Archived from the original (https://www.dcs.qmul.ac.uk/~norman/papers/Combining%20evidence%20in%20risk%20analysis%20using%20BNs.pdf) (PDF) on 2007-09-27.
- Gelman A, Carlin JB, Stern HS, Rubin DB (2003). "Part II: Fundamentals of Bayesian Data Analysis: Ch.5 Hierarchical models" (https://books.google.com/books?id=TNYhnkXQSjAC&pg=PA120). *Bayesian Data Analysis* (https://books.google.com/books?id=TNYhnkXQSjAC). CRC Press. pp. 120–. ISBN 978-1-58488-388-3.
- Heckerman, David (March 1, 1995). "Tutorial on Learning with Bayesian Networks" (https://web.archive.org/web/20060719171558/http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-95-06). In Jordan, Michael Irwin (ed.). *Learning in Graphical Models*. Adaptive Computation and Machine Learning. Cambridge, Massachusetts: MIT Press (published 1998). pp. 301–354. ISBN 978-0-262-60032-3. Archived from the original on July 19, 2006. Retrieved September 15, 2006.:Also appears as Heckerman, David (March 1997). "Bayesian Networks for Data Mining". *Data Mining and Knowledge Discovery*. **1** (1): 79–119. doi:10.1023/A:1009730122752 (https://doi.org/10.1023%2FA%3A1009730122752). S2CID 6294315 (https://api.semanticscholar.org/CorpusID:6294315).

  An earlier version appears as , Microsoft Research, March 1, 1995. The paper is about both parameter and structure learning in Bayesian networks.

- Jensen FV, Nielsen TD (June 6, 2007). *Bayesian Networks and Decision Graphs* (https://books.google.com/books?id=cWLaBwAAQBAJ). Information Science and Statistics series (2nd ed.). New York: Springer-Verlag. ISBN 978-0-387-68281-5.
- Karimi K, Hamilton HJ (2000). "Finding temporal relations: Causal bayesian networks vs. C4.5" (http://www.kamran-karimi.com/pubs/khISMIS2000.pdf) (PDF). *Twelfth International Symposium on Methodologies for Intelligent Systems*.
- Korb KB, Nicholson AE (December 2010). *Bayesian Artificial Intelligence* (https://books.google.com/books?id=LxXOBQAAQBAJ). CRC Computer Science & Data Analysis (2nd ed.).

Chapman & Hall (CRC Press). doi:10.1007/s10044-004-0214-5 (https://doi.org/10.1007%2Fs10044-004-0214-5). ISBN 978-1-58488-387-6. S2CID 22138783 (https://api.semanticscholar.org/CorpusID:22138783).

- Lunn D, Spiegelhalter D, Thomas A, Best N (November 2009). "The BUGS project: Evolution, critique and future directions". *Statistics in Medicine*. **28** (25): 3049–67. doi:10.1002/sim.3680 (https://doi.org/10.1002%2Fsim.3680). PMID 19630097 (https://pubmed.ncbi.nlm.nih.gov/19630097). S2CID 7717482 (https://api.semanticscholar.org/CorpusID:7717482).
- Neil M, Fenton N, Tailor M (August 2005). Greenberg, Michael R. (ed.). "Using Bayesian networks to model expected and unexpected operational losses" (http://www.dcs.qmul.ac.uk/~norman/papers/oprisk.pdf) (PDF). *Risk Analysis*. **25** (4): 963–72. doi:10.1111/j.1539-6924.2005.00641.x (https://doi.org/10.1111%2Fj.1539-6924.2005.00641.x). PMID 16268944 (https://pubmed.ncbi.nlm.nih.gov/16268944). S2CID 3254505 (https://api.semanticscholar.org/CorpusID:3254505).
- Pearl J (September 1986). "Fusion, propagation, and structuring in belief networks". *Artificial Intelligence*. **29** (3): 241–288. doi:10.1016/0004-3702(86)90072-X (https://doi.org/10.1016%2F0004-3702%2886%2990072-X).
- Pearl J (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (https://books.google.com/books?id=mn2jBQAAQBAJ). Representation and Reasoning Series (2nd printing ed.). San Francisco, California: Morgan Kaufmann. ISBN 978-0-934613-73-6.
- Pearl J, Russell S (November 2002). "Bayesian Networks". In Arbib MA (ed.). *Handbook of Brain Theory and Neural Networks* (https://books.google.com/books?id=Av6qWhtw0-EC). Cambridge, Massachusetts: Bradford Books (MIT Press). pp. 157–160. ISBN 978-0-262-01197-6.
- Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (http://aima.cs.berkeley.edu/) (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.
- Zhang NL, Poole D (May 1994). "A simple approach to Bayesian network computations" (http://www.cs.ust.hk/~lzhang/paper/pspdf/canai94.pdf) (PDF). *Proceedings of the Tenth Biennial Canadian Artificial Intelligence Conference (AI-94).*: 171–178. This paper presents variable elimination for belief networks.

Further reading

- Conrady S, Jouffe L (2015-07-01). *Bayesian Networks and BayesiaLab – A practical introduction for researchers* (https://books.google.com/books?id=etXXsgEACAAJ). Franklin, Tennessee: Bayesian USA. ISBN 978-0-9965333-0-0.
- Charniak E (Winter 1991). "Bayesian networks without tears" (http://pages.cs.wisc.edu/~dyer/cs540/handouts/charniak.pdf) (PDF). *AI Magazine*.
- Kruse R, Borgelt C, Klawonn F, Moewes C, Steinbrecher M, Held P (2013). *Computational Intelligence A Methodological Introduction* (https://books.google.com/books?id=etXXsgEACAAJ). London: Springer-Verlag. ISBN 978-1-4471-5012-1.
- Borgelt C, Steinbrecher M, Kruse R (2009). *Graphical Models – Representations for Learning, Reasoning and Data Mining* (https://books.google.com/books?id=I8Fa-LKDpF0C) (Second ed.). Chichester: Wiley. ISBN 978-0-470-74956-2.

External links

- An Introduction to Bayesian Networks and their Contemporary Applications (http://www.niedermayer.ca/papers/bayesian/bayes.html)

- On-line Tutorial on Bayesian nets and probability (http://www.dcs.qmw.ac.uk/%7Enorman/BBNs/BBNs.htm)
- Web-App to create Bayesian nets and run it with a Monte Carlo method (https://web.archive.org/web/20170601002137/http://princesofserendib.com/)
- Continuous Time Bayesian Networks (http://robotics.stanford.edu/~nodelman/papers/ctbn.pdf)
- Bayesian Networks: Explanation and Analogy (https://web.archive.org/web/20090923200511/http://wiki.syncleus.com/index.php/DANN%3ABayesian_Network)
- A live tutorial on learning Bayesian networks (http://videolectures.net/kdd07_neapolitan_lbn/)
- A hierarchical Bayes Model for handling sample heterogeneity in classification problems (http://www.biomedcentral.com/1471-2105/7/514/abstract), provides a classification model taking into consideration the uncertainty associated with measuring replicate samples.
- Hierarchical Naive Bayes Model for handling sample uncertainty (http://www.labmedinfo.org/download/lmi339.pdf) Archived (https://web.archive.org/web/20070928081740/http://www.labmedinfo.org/download/lmi339.pdf) 2007-09-28 at the Wayback Machine, shows how to perform classification and learning with continuous and discrete variables with replicated measurements.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Bayesian_network&oldid=1094972270"