

# Appendix B

---

## *Software Packages*

---

### **B.1 Introduction**

The rapid development of Bayesian network research over the past 15 years has been accompanied by a proliferation of BN software tools. These tools have been built to support both these research efforts and the applications of BNs to an ever-widening range of domains. This appendix is intended as a resource guide to those software tools.

We incorporate Kevin Murphy's listing of software packages for Bayesian networks and graphical models (§B.3), which has been built up and maintained over a number of years.\* We also note other websites for BN software and other resources.

Next, we describe some of the major software packages – those with the most functionality, or with a particular feature of interest – in more detail. Most of these packages have a long list of features that we cannot even list here, so our survey is by no means exhaustive. We endeavor to point out particular features that relate to issues we have raised earlier in this text. Note that we have personal experience, through teaching, research projects or application development, with the following software: BNT, BUGS, CBeN, CaMML, Hugin, IDEAL, Netica, SMILE and TETRAD II.

In general, all the packages with GUIs include the advanced GUI features (e.g., menu options, short-cut icons, drag-and-drop, online help) that have become the norm in recent years. In this resource review, we will generally ignore GUI aspects, unless there is some feature that stands out. Instead, we will concentrate on aspects of the functionality.

We make no attempt to give any sort of ranking of these packages, so our survey presents the packages alphabetically. The Murphy listing notes whether the products are free or commercial but available in a restricted form. Otherwise we do not make any comments on the cost of commercial products.

---

\* Our thanks to Kevin Murphy for giving us permission to use this listing.

---

## B.2 History

The development of the first BN software, beyond algorithm implementation, occurred concurrently with the surge of BN research in the late 1980s. Hugin [6] was developed at the University of Aalborg in Denmark (see §B.4.5 below) and went on to become a commercial product now widely used. The Lisp-base IDEAL (Influence Diagram Evaluation and Analysis in Lisp) test-bed environment was developed at Rockwell [267].<sup>†</sup>

Another early BN inference engine was CABeN (a Collection of Algorithms for Belief Networks) [58], which contains a library of routines for different stochastic simulation inference algorithms. Lumina Decision Systems, Inc., was founded in 1991 by Max Henrion and Brian Arnold, which produce Analytica. The development of Netica, now produced by Norsys Ltd, was started in 1992.

---

## B.3 Murphy's Software Package Survey

<http://www.ai.mit.edu/~murphyk/Software/BNT/bnsoft.html>

This survey is given in Tables B.1 and B.2. The first table gives the software package name, its producers, and where it is available online. If the software is commercial but the company has links with particular institutions or BN researchers, those are also noted. The second table covers basic feature information such technical information about source availability, platforms, GUI and API, very high level functionality such as types of nodes supported (i.e., discrete and/or continuous), whether decision networks are supported, whether undirected graphs are supported. It includes whether the software allows learning (parameters and/or structure), describes the main inference algorithms (if this information is available), and indicates whether the software is free or commercial. The details of the meaning of each column is given in Figure B.1.

Google's list of tools is available at:

[http://directory.google.com/Top/Computers/Artificial\\_Intelligence/Belief\\_Networks/Software/](http://directory.google.com/Top/Computers/Artificial_Intelligence/Belief_Networks/Software/)

The Bayesian Network Repository contains examples of BNs, plus datasets for learning them:

<http://www.cs.huji.ac.il/labs/combio/Repository>

---

<sup>†</sup>One of the authors used it for her Ph.D. research.

**Src** Is the source code included? **N=no**. If yes, what language? **J** = Java, **M** = Matlab, **L** = Lisp, **C**, **C++**, **R** language names.

**API** Is an application program interface included?

**N** means the program cannot be integrated into your code, i.e., it must be run as a standalone executable. **Y** means it can be integrated.

**Exec** The **executable** runs on: **W** = Windows (95/98/2000/NT), **U** = Unix, **M** = Mac-Intosh, **-** = Any machine with a compiler.

**GUI** Is a **Graphical User Interface** included? **Y=Yes,N=No**.

**D/C** Are continuous-valued nodes supported (as well as discrete)? **G** = (conditionally) Gaussians nodes supported analytically, **Cs** = continuous nodes supported by sampling, **Cd** = continuous nodes supported by discretization, **Cx** = continuous nodes supported by some unspecified method, **D** = only discrete nodes supported.

**DN** Are decision networks/influence diagrams supported? **Y=Yes,N=No**.

**Params** Does the software functionality include parameter learning? **Y=Yes,N=No**.

**Struct** Does the software functionality include structure learning? **Y=Yes,N=No**.

**CI** means **Y**, using conditional independency tests (see § 6.3)

**K2** means **Y**, using Cooper & Herskovits' K2 algorithm (see § 8.2)

**D/U** What kind of graphs are supported? **U** = only undirected graphs, **D** = only directed graphs, **UD** = both undirected and directed, **CG** = chain graphs (mixed directed/undirected).

**Inf** Which inference algorithm is used? (See Chapter 3)

**JT** = **Junction Tree**, **VE** = **variable (bucket) elimination**, **PT** = **Pearl's poly-tree**, **E** = **Exact inference** (unspecified), **MH** = **Metropolis Hastings**, **MC** = **Markov chain Monte Carlo (MCMC)**, **GS** = **Gibbs sampling**, **IS** = **Importance sampling**, **S** = **Sampling**, **O** = **Other** (usually special purpose), **++** = **Many** methods provided, **?** = Not specified, **N** = None, the program is only designed for structure learning from completely observed data.

**NB**: Some packages support a form of sampling (.e.g., **likelihood weighting**, **MDMC**), in addition to their exact algorithm; this is indicated by **(+S)**.

**Free** Is a free version available? **O**=Free (though possibly only for academic use) **\$** = Commercial (although most have free versions which are restricted in various ways, e.g., the model size is limited, or models cannot be saved, or there is no API.)

**FIGURE B.1**

Description of features used in Murphy's BN software survey, in Table B.2.

**TABLE B.1**  
Software packages: name, web location and developers

Name	Web Location	Authors
Analytica	<a href="http://www.luminna.com">http://www.luminna.com</a>	Luminna (Hennron)
Bassist	<a href="http://www.cs.helsinki.fi/research/fdk/bassist">http://www.cs.helsinki.fi/research/fdk/bassist</a>	U. Helsinki
Bayda	<a href="http://www.cs.helsinki.fi/research/cosoc/Projects/NONE/SW/">http://www.cs.helsinki.fi/research/cosoc/Projects/NONE/SW/</a>	U. Helsinki
BayesBuilder	<a href="http://www.mbfys.kun.nl/sm/Research/bayesbuilder/">http://www.mbfys.kun.nl/sm/Research/bayesbuilder/</a>	Nijman (U. Nijmegen)
Bayesial ab	<a href="http://www.bayesia.com">http://www.bayesia.com</a>	Bayesia Ltd
Bayesware Discoverer	<a href="http://www.bayesware.com">http://www.bayesware.com</a>	Bayesware (Open Univ., UK)
B-course	<a href="http://b-course.cs.helsinki.fi">http://b-course.cs.helsinki.fi</a>	U. Helsinki
BN power constructor	<a href="http://www.cs.ualberta.ca/~cheng/bnpce.htm">http://www.cs.ualberta.ca/~cheng/bnpce.htm</a>	Cheng (U. Alberta)
BNT	<a href="http://www.ai.mtl.edu/~mumphyk/Software/BNT/bnt.html">http://www.ai.mtl.edu/~mumphyk/Software/BNT/bnt.html</a>	Murphy (prev U.C. Berkeley, now MIT)
BNI	<a href="http://bndev.sourceforge.net/">http://bndev.sourceforge.net/</a>	Hsu (Kansas)
BucketElim	<a href="http://www.jcs.uci.edu/~trinar">http://www.jcs.uci.edu/~trinar</a>	Rish (U.C. Irvine)
BUGS	<a href="http://www.mrc-bsu.cam.ac.uk/bugs">http://www.mrc-bsu.cam.ac.uk/bugs</a>	MRC/Imperial College
Business Navigator 5	<a href="http://www.data-digest.com">http://www.data-digest.com</a>	Data Digest Corp
CABen	<a href="http://www-pod.stanford.edu/cousins/caben-1.1.tar.gz">http://www-pod.stanford.edu/cousins/caben-1.1.tar.gz</a>	Cousins et al. (Wash. U.)
CAMML	<a href="http://www.damining.monash.edu.au/software/camml">http://www.damining.monash.edu.au/software/camml</a>	Wallace, Korb (Monash U.)
CoCo+Xisp	<a href="http://www.math.auc.dk/~jhb/COCO/information.html">http://www.math.auc.dk/~jhb/COCO/information.html</a>	Badsberg (U. Aalborg)
CISpace	<a href="http://www.cs.ubc.ca/labs/lei/CISpace/">http://www.cs.ubc.ca/labs/lei/CISpace/</a>	Poole et al. (UBC)
Deal	<a href="http://www.math.auc.dk/novo/deal">http://www.math.auc.dk/novo/deal</a>	Botcher et al.
Ergo	<a href="http://www.noeticsystems.com">http://www.noeticsystems.com</a>	Noetic systems
GDA/Gsim	<a href="http://www.staff.ncl.ac.uk/d.j.wilkinson/software/gdagsim/">http://www.staff.ncl.ac.uk/d.j.wilkinson/software/gdagsim/</a>	Wilkinson (U. Newcastle)
GMRFSim	<a href="http://www.math.rutn.no/~trane/GMRFSim/">http://www.math.rutn.no/~trane/GMRFSim/</a>	Rue (U. Trondheim)
GENIE/SMILE	<a href="http://www.sis.pitt.edu/~genie">http://www.sis.pitt.edu/~genie</a>	U. Pittsburgh (Druzdel)
GMTK	<a href="http://ssl.lee.washington.edu/~hlnes/gmtk/">http://ssl.lee.washington.edu/~hlnes/gmtk/</a>	Blimes (UW), Zweig (IBM)
gR	<a href="http://www.r-project.org/gR">http://www.r-project.org/gR</a>	Lauritzen et al.
Grappa	<a href="http://www.stats.bris.ac.uk/~peter/Grappa/">http://www.stats.bris.ac.uk/~peter/Grappa/</a>	Green (Bristol)
Hugin	<a href="http://www.hugin.com">http://www.hugin.com</a>	Hugin Expert (U. Aalborg, Lauritzen/Jensen)
Hydra	<a href="http://software.bioslat.washington.edu/statsoft/MCMC/Hydra">http://software.bioslat.washington.edu/statsoft/MCMC/Hydra</a>	Warnes (U. Wash.)
Ideal	<a href="http://yoda.cis.temple.edu:8080/ideal/">http://yoda.cis.temple.edu:8080/ideal/</a>	Rockwell (Srinivas)
Java Bayes	<a href="http://www.cs.cmu.edu/~javabayes/Home/">http://www.cs.cmu.edu/~javabayes/Home/</a>	Cozman (CMU)
MIM	<a href="http://www.hypergraph.dk/">http://www.hypergraph.dk/</a>	HyperGraph Software
MSBNx	<a href="http://research.microsoft.com/adapt/MSBNx/">http://research.microsoft.com/adapt/MSBNx/</a>	Microsoft
Netica	<a href="http://www.norsys.com">http://www.norsys.com</a>	Norsys (Boerlage)
PMT	<a href="http://people.bu.edu/vladimir/pmt/index.html">http://people.bu.edu/vladimir/pmt/index.html</a>	Pavlovic (BU)
PNL	<a href="http://www.ai.mtl.edu/~mumphyk/Software/PNL/pnl.html">http://www.ai.mtl.edu/~mumphyk/Software/PNL/pnl.html</a>	Eruhinov (Inel)
Pulcinella	<a href="http://irdia.ualb.ac.be/pulcinella/Welcome.html">http://irdia.ualb.ac.be/pulcinella/Welcome.html</a>	IRIDIA
RISO	<a href="http://sourceforge.net/projects/riso">http://sourceforge.net/projects/riso</a>	Dodier (U. Colorado)
TETRAD	<a href="http://www.phl.cmu.edu/tetrad/">http://www.phl.cmu.edu/tetrad/</a>	CMU
UnBBayes	<a href="https://sourceforge.net/projects/unbbayes/">https://sourceforge.net/projects/unbbayes/</a>	?
Vibes	<a href="http://www.inference.phy.cam.ac.uk/jmw39/">http://www.inference.phy.cam.ac.uk/jmw39/</a>	Winn & Bishop (U. Cambridge)
Web Weaver	<a href="http://snowhite.cis.ouguelpk.ca/faculty_jinfo/xyiang/ww3/">http://snowhite.cis.ouguelpk.ca/faculty_jinfo/xyiang/ww3/</a>	Xiang (U. Regina)
WinMine	<a href="http://research.microsoft.com/~dmax/WinMine/tooldoc.htm">http://research.microsoft.com/~dmax/WinMine/tooldoc.htm</a>	Microsoft
XBAIES 2.0	<a href="http://www.staff.city.ac.uk/~Tgc/wepages/xbpape.html">http://www.staff.city.ac.uk/~Tgc/wepages/xbpape.html</a>	Cowell (City U.)

**TABLE B.2**  
Murphy's feature comparison of software packages

Name	Src	API	Exec	GUI	D/C	DN	Params	Struct	D/U	Infer	Free
Analytica	N	Y	WM	Y	G	Y	N	N	D	S	\$
Bassist	C++	Y	U	G	N	N	Y	N	D	MH	O
Bayda	J	Y	WUM	Y	G	N	Y	N	D	?	O
BayesBuilder	N	N	W	Y	D	N	N	N	D	?	O
BayesiaLab	N	N	-	Y	Cd	N	Y	Y	CG	JT,G	\$
Bayeware	N	N	WUM	Y	Cd	N	Y	Y	D	?	\$
B-course	N	N	WUM	Y	Cd	N	Y	Y	D	?	O
BNPC	N	Y	W	Y	D	N	Y	CI	D	?	O
BNT	M/C	Y	WUM	N	G	Y	Y	Y	DU	S,E(++)	O
BNJ	J	Y	-	Y	D	N	N	Y	D	JT,IS	O
BucketElim	C++	Y	WU	N	D	N	N	N	D	VE	O
BUGS	N	N	WU	Y	Cs	N	Y	N	Ds	GS	O
BusNav	N	N	W	Y	Cd	N	Y	Y	D	JT	\$
CABeN	C	Y	WU	N	D	N	N	N	D	S(++)	O
CaMML	C	N	U	N	Cx	N	Y	Y	D	N	O
CoCo+Xlisp	C/L	Y	U	Y	D	N	Y	CI	U	JT	O
CIspace	J	N	WU	Y	D	N	N	N	D	VE	O
Deal	R	-	-	Y	G	N	M	Y	D	N	O
Ergo	N	Y	WM	Y	D	N	N	N	D	JT(+S)	\$
GDAGsim	C	Y	WUM	N	G	N	N	N	D	E	O
GeNIe/SMILE	N	Y	WU	Y	D	Y	N	N	D	JT(+S)	O
GMRFSim	C	Y	WUM	N	G	N	N	N	U	MC	O
GMTk	N	Y	U	N	D	N	Y	Y	D	JT	O
gR	R	-	-	-	-	-	-	-	-	-	O
Grappa	R	Y	-	N	D	N	N	N	D	JT	O
Hugin	N	Y	WU	Y	G	Y	Y	CI	CG	JT	\$
Hydra	J	Y	-	Y	Cs	N	Y	N	DU	MC	O
Ideal	L	Y	WUM	Y	D	Y	N	N	N	JT	\$
Java Bayes	J	Y	WUM	Y	D	Y	N	N	N	JT,VE	O
MIM	N	N	W	Y	G	N	Y	Y	CG	JT	\$
MSBNx	N	Y	W	Y	D	Y	N	N	D	JT	O
Netica	N	Y	WUM	Y	G	Y	Y	N	D	JT	\$
PMT	M/C	Y	-	N	D	N	Y	N	D	O	O
PNL	C++	-	-	N	D	N	Y	Y	UD	JT	O
Pulcinella	L	Y	WUM	Y	D	N	N	N	D	?	O
RISO	J	Y	WUM	Y	G	N	N	N	D	PT	O
TETRAD	N	N	WU	N	G	N	Y	CI	UD	N	O
UnBBayes	J	Y	-	Y	D	N	N	Y	D	JT	O
Vibes	J	Y	WU	Y	Cx	N	Y	N	D	V?	O
Web Weaver	J	Y	WUM	Y	D	Y	N	N	D	?	O
WinMine	N	N	W	Y	Cx	N	Y	Y	UD	N	O
XBAIES 2.0	N	N	W	Y	G	Y	Y	Y	CG	JT	O

---

## B.4 BN software

In this section we review some of the major software packages for Bayesian and decision network modeling and inference. The information should be read in conjunction with the feature summary in Table B.2. The additional aspects we consider are as follows.

**Development:** Any background information of the developers or the history of this software.

**Technical:** Further information about platforms or products (beyond the summary given in Table B.2).

**Node Types:** Relating to discrete/continuous support (see §9.3.1.4).

**CPTs:** Support for elicitation (§9.3.3), or local structure (§7.4, §9.3.4).

**Inference:** More details about the inference algorithm(s) provided, and possible user control over inference options (Chapter 3). Also whether computes MPE and P(E) (§3.7).

**Evidence:** Whether negative and likelihood evidence are supported, in addition to specific evidence (§3.4).

**Decision networks:** Information about decision network evaluation (if known), whether expected utilities for all policies are provided, or just decision tables (§4.3.4). Whether precedence links between decision nodes are determined automatically if not specified by the knowledge engineer (§4.4). Also, whether value of information is supported directly (§4.4.4).

**DBNs:** Whether DBN representation and/or inference is supported (§4.5).

**Learning:** What learning algorithms are used (Chapters 6-8).

**Evaluation:** What support, if any, for evaluation? (Chapter 10). In particular, sensitivity analysis (§10.3) and statistical validation methods (§10.5).

**Other features:** Functionality not found in most other packages.

### B.4.1 Analytica

Lumina Decision Systems, Inc.  
26010 Highland Way, Los Gatos, CA 95033  
<http://www.lumina.com>

**Development:** As noted earlier in the history outline, Lumina Decision Systems, Inc., was founded in 1991 by Max Henrion and Brian Arnold. The emphasis in Analytica is on using influence diagrams as a statistical decision support tool. Analytica does not use Bayesian network terminology, which can lead to difficulties in identifying aspects of its functionality.

**Technical:** Analytica 2.0 GUI is available for Windows and Macintosh. The Analytica API (called the Analytica Decision Engine) is available for windows 95/98 or NT 4.0, and runs in any development environment with COM or Automation support.

**CPTs:** Analytica supports many continuous and discrete distributions, and provides a large number of mathematical and statistical functions.

**Inference:** Analytica provides basic MDMC sampling, plus median latin hypercube (the default method) and random latin hypercube, and allows the sample size to be set. The Analytica GUI provides many ways to view the results of inference, through both tables and graphs: statistics, probability bands, probability mass (the standard for most other packages), cumulative probability, and the actual samples generated by the inference.

**Evidence:** Specific evidence can only be entered for variables previously set up as “input nodes”.

**DBNs:** Analytica provides dynamic simulation time periods by allowing the user to specify both a list of time steps and which variables change over time. Note: Analytica does not use DBN terminology or show the “rolled-out” network.

**Evaluation:** Analytica provides what it calls “importance analysis”, which is an absolute rank-order correlation between the sample of output values and the sample for each uncertain input. This can be used to create so-called importance variables. Analytica also provides a range of sensitivity analysis functions, including “whatif” and scatterplots.

**Other features:** Analytica supports the building of large models by allowing the creation of a hierarchical combination of smaller models, connected via specified input and output nodes.

## B.4.2 BayesiaLab

BAYESIA

6, rue Lonard de Vinci - BP0102,53001 Laval Cedex, France

<http://www.bayesia.com>

**Technical:** BayesiaLab GUI is available for all platforms supporting JRE. There is also a product “BEST”, for using BNs for diagnosis and repair.

**Node Types:** Continuous variables must be discretized. When learning variables from a database, BayesiaLab supports equal distance intervals, equal frequency intervals, and a decision tree discretization that chooses the intervals depending on the information they contribute to a specified target variable.

**CPTs:** Some types of node such as logical-or are supported. BayesiaLab supports entry of the CPT through normalization, completing entries, offering multiple entries and will also generate random entries.

**Inference:** Inference in BayesiaLab is done in what they call “validation” mode (compared to the modeling mode for changing the network). A junction tree algorithm is the default used, with a MDMC Gibbs sampling algorithm available.

**Evidence:** Only specific evidence supported (entered through the so-called “monitor”).

**DBNs:** BayesiaLab supports DBN specification and roll-out (with the temporal layout *down* rather than across).

**Learning:** BayesiaLab apparently does parameter learning using a version of the Spiegelhalter & Lauritzen parameterization algorithm. It has three methods for structural learning (which they call “association discovery”): SopLEQ, which uses properties of equivalent Bayesian networks and two versions of Taboo search. BayesiaLab also provides clustering algorithms

**Evaluation:** BayesiaLab provides a number of functions (with graphical display support) for evaluating a network, including the strength of the arcs (used also by the automated graph layout), the amount of information brought to the target node, the type of probabilistic relation, generation of analysis reports, and causal analysis (based on an idea of “essential graphs”, showing arcs that can’t be reversed without changing the probability distribution represented), and arc reversal. It supports simulation of “What-if” scenarios, and provides sensitivity analysis (through their so-called “adaptive questionnaires”), lift curves and confusion matrices.

**Other features:** BayesiaLab supports hidden variables, importing from a database, and export a BN for use by their troubleshooting product, BEST.

### B.4.3 Bayes Net Toolbox (BNT)

Kevin Murphy

MIT AI lab, 200 Technology Square, Cambridge, MA 02139

<http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html>

**Development:** This package was developed during Kevin Murphy’s time at U.C. Berkeley as a Ph.D. student, where his thesis [198] addressed DBN representation, inference and learning. He also worked on BNT while an intern at Intel.

**Technical:** The Bayes Net Toolbox is for use with only with Matlab, a widely used and powerful mathematical software package. Its lack of a GUI is made up for by Matlab’s visualization features. This software is distributed under the Gnu Library General Public License.

**CPTs:** BNT support the following conditional probability distributions: tabular (multinomial), Gaussian, Softmax (logistic/ sigmoid), Multi-layer perceptron (neural network), Noisy-or, Deterministic.



**Inference:** BNT supports many different exact and approximate inference algorithms, for both ordinary BNs and DBNs, including all the algorithms described in this text.

**DBNs:** The following dynamic models can be implemented in BNT: Dynamic HMMs, Factorial HMMs, coupled HMMs, input-output HMMs, DBNs, Kalman filters, ARMAX models, switching Kalman filters, tree-structured Kalman filters, multiscale AR models.

**Learning:** BNT parameter learning methods are: (1) Batch MLE/MAP parameter learning using EM (different M and E methods for each node type); (2) Sequential/batch Bayesian parameter learning (for tabular nodes only).

Structure learning methods are: (1) Bayesian structure learning, using MCMC or local search (for fully observed tabular nodes only); (2) Constraint-based structure learning (IC/PC and IC\*/FCI).

#### B.4.4 GeNIe

Decision Systems Laboratory, University of Pittsburgh  
B212 SLIS Building, 135 North Bellefield Avenue, Pittsburgh, PA 15260, USA  
<http://www.sis.pitt.edu/~genie/>

**Development:** Developed by Druzzdel's decision systems group, GeNIe's support of decision networks, in addition to BNs, reflects their teaching and research interests in decision support and knowledge engineering. GeNIe 1.0 was released in 1998, and GeNIe 2.0 is due for release in mid-2003.

**Technical:** GeNIe (Graphical Network Interface) is a development environment for building decision networks, running under Windows. SMILE (Structural Modeling, Reasoning, and Learning Engine) is its portable inference engine, consisting of a library of C++ classes currently compiled for Windows, Solaris and Linux. GeNIe is an outer shell to SMILE. Here we focus on describing GeNIe.

**CPTs:** Supports chance nodes with General, Noisy OR/MAX and Noisy AND distribution, as well as graphical elicitation of probabilities.

**Inference:** GeNIe's default BN inference algorithm is the junction tree clustering algorithm, however a polytree algorithm is also available, plus several approximate algorithms that can be used if the networks get too large for clustering (logic sampling, likelihood weighting, self importance and heuristic importance sampling, backward sampling). GeNIe 2.0 provides more recent state-of-the-art sampling algorithms, AIS-BN [43] and EPIS-BN[?].

**Evidence:** Only handles specific evidence.

**Decision networks:** GeNIe offers two decision network evaluation algorithms: a fast algorithm [250] that provides only the best decision, and a slower algorithm that use an inference algorithm to evaluate the BN part of the network, then computes the expected utility for all possible policies.

If the user does not specify the temporal order of the decision nodes, it will try to infer it using causal considerations, otherwise it will decide an order arbitrarily. To simplify the displayed model, GeNIe does not require the user to create temporal arcs, inferring them from the temporal order among the decision nodes.

Viewing results: The value node will show the expected utilities of all combinations of decision alternatives. The decision node will show the expected utilities of its alternatives, possibly indexed by those decision nodes that precede it.

GeNIe provides the expected value of information, i.e., the expected value of observing the state of a node before making a decision.

**Evaluation:** GeNIe supports simple sensitivity analysis in graphical models, through the addition of a variable that indexes various values for parameters in question. GeNIe computes the impact of these parameter values on the decision results (showing both the expected utilities and the policy). Using the same index variable, GeNIe can display the impact of uncertainty in that parameter on the posterior probability distribution of any node in the network.

**Other features:** GeNIe allows submodels and a tree view. It can handle other BN file formats (Hugin, Netica, Ergo). GeNIe provides integration with MS. Excel, including cut and paste of data into internal spreadsheet view of GeNIe, and supports for diagnostic case management.

GeNIe also supports what they call “**relevance reasoning**” [173], allowing users to specify nodes that are of interest (so-called target nodes). Then when updating computations are performed, only the nodes of interest are guaranteed to be fully updated; this can result in substantial reductions in computation.

### B.4.5 Hugin

Hugin Expert, Ltd

ADDRESS

<http://www.hugin.com>

**Development:** The original Hugin shell was initially developed by a group at the Aalborg University, as part of an ESPRIT project which also produced MUNIN system [9]. Hugin’s development continued through another Lauritzen-Jensen project called ODIN. Hugin Expert was established to start commercializing the Hugin tool. The close connection between Hugin Expert and the Aalborg research group has continued, including co-location and personnel moving between the two. This has meant that Hugin Expert has consistently contributed to and taken advantage of the latest BN research. In 1998 Hewlett-Packard purchased 45% of Hugin Expert; one consequence of this seems to have been the tailored development of Hugin to support trouble-shooting.

**Technical:** The Hugin API is called the “Hugin Decision Engine”. It is available for the languages C++, Java and as an ActiveX-server, and runs on the operating systems: Sun Solaris (Sparc and i x86), HP-UX, Linux, and Windows. Versions are available for single and double-precision floating-point operations. The Hugin GUI (called “Hugin”) is available for Sun Solaris (sparc, x86) Windows, and Linux red-hat.

Hugin also offers “Hugin Advisor” for developing trouble shooting applications, and “Hugin Clementine” for integrating Hugin’s learning with datamining in SPSS’s Clementine system.

**Node Types:** Good support for continuous variable modeling, and combining discrete and continuous nodes, following on from research in this area [211].

**CPTs:** CPTs can be specified with expressions as well as through manual entry. The CPTs don’t have to sum to one; entries that don’t sum to one are normalized.

**Inference:** The basic algorithm is the junction tree algorithm, with options to choose between variations. The junction tree may be viewed. There is the option to vary the triangulation method, and another to turn on compression (of zeros in the junction tree) (see Problem 3, Chapter 3). An approximate version of the junction tree algorithm is offered, where all probabilities less than a specified threshold are made zero (see Problem 5, Chapter 3).

In addition Hugin GUI computes P(E), the data conflict measure [131, 146], described in §3.7.2, and the MPE.

**Evidence:** Specific, negative and virtual evidence are all supported.

**Decision networks:** Hugin requires the existence of a directed path including all decision variables. It gives the expected utility of each decision option in the decision table.

**Learning:** The parameter learning is done with EM learning (see §7.3.2.2), and Spiegelhalter & Lauritzen sequential learning (adaptation) and fading are also supported. Structure learning is done using the PC algorithm (see §6.3.2)..

**Other features:** Supports object-oriented BNs(see §9.3.5.2).

## B.4.6 JavaBayes

Fabio Gagliardi Cozman

Escola Politecnica, University of So Paulo

<http://www.cs.cmu.edu/~javabayes/Home/>

<http://www.pmr.poli.usp.br/ltd/Software/javabayes/> (recent versions)

**Development:** JavaBayes was the first BN software produced in Java and is distributed under the GNU License.

**Other features:** JavaBayes provides a set of parsers for importing Bayesian networks in several proposed so-called “interchange” formats.

JavaBayes also offers Bayesian **robustness analysis** [62], where sets of distributions are associated to variables: the size of these sets indicates the “uncertainty” in the modeling process. JavaBayes can use models with sets of distributions to calculate intervals of posterior distributions or intervals of expectations. The larger these intervals, the less robust are the inferences with respect to the model.

#### B.4.7 MSBNx

Microsoft

<http://research.microsoft.com/adapt/MSBNx/>

**CPTs:** MSBNx supports the construction of the usual tables, as well as local structure in the form of context-sensitive independence (CSI)[30], (see §9.3.4), and classification trees (see §7.4.3).

**Inference:** A form of junction tree algorithm is used.

**Evidence:** Supports specific evidence only.

**Evaluation:** MSBNx can recommend what evidence to gather next. If given cost information, MSBNx does a cost-benefit analysis, otherwise it makes recommendations based on an entropy-based value of information measure (note: prior to 2001, this was a KL-divergence based measure).

#### B.4.8 Netica

Norsys Software Corp.

3512 W 23<sup>rd</sup> Ave., Vancouver, BC, Canada V6S 1K5

<http://www.norsys.com>

**Development:** Netica’s development was started in 1992, by Norsys CEO Brent Boerlage, who had just finished a Masters degree at the University of British Columbia, where his thesis looked at quantifying and displaying “link strengths” in Bayesian networks [23]. Netica became commercially available in 1995, and is now widely used.

**Technical:** The Netica API is available for languages C and Java, to run on Mac OSX, Sun Sparc, Linux and Windows. The GUI is available for Mac and Windows. There is also a COM interface for integrating the GUI with other GUI applications and Visual Basic programming.

**Node Types:** Netica can learn node names from variable names in a data file (called a case file). Netica discretizes continuous variables but allows control over the discretization.

**CPTs:** There is some support for manual entry of probabilities, with functions for checking that entries sum to 100 (Netica has a default option to use numbers out of 100, rather than probabilities between 0 and 1), automatically filling in

the final probability, and normalizing. Equations can also be used to specify the CPT, using a large built-in library of functions and continuous and discrete probability distributions, and there is support for noisy-or, noisy-and, noisy-max and noisy-sum nodes.

**Inference:** Netica's inference is based on the elimination junction tree method (see §3.10). The standard compilation uses a minimum-weight search for a good elimination order, while an optimized compilation option searches for the best elimination order using a combination of minimum-weight search and stochastic search. Both the junction tree and the elimination order may be viewed. Netica also reports both the probability of the most recent evidence, and the probability of all evidence currently entered, and provides the MPE and its probability (but not for networks containing decision nodes). Netica can generate random samples by junction tree or logic sampling.

**Evidence:** Netica supports specific (which they call "positive"), negative and likelihood evidence. Multiple likelihood evidence may be incorporated for the same nodes. Netica also handles sets of evidence (cases) by case files and direct database access.

**Decision networks:** Netica infers a temporal order for decision network, if it can. DN evaluation gives the expected utilities for a one-off decision, but only the decision table for sequential decision making.

**DBNs:** Netica supports DBN specification and roll-out.

**Learning:** Netica supports parameter learning only. It uses the Spiegelhalter & Lauritzen parameterization algorithm, allows missing values, and allows the specification of a weighting to the original probabilities, providing a form of adaptation. Netica can also do EM learning and gradient descent learning, to handle large amounts of missing data, or latent (unobserved) variables. It also supports fading, with the user able to specify a factor from 0 (no new learning) to 1 (removes all previous learning).

**Evaluation:** Netica supports sensitivity to findings, as described earlier in §10.3.1. It also provides a number of measures for statistical validation including a count form of predictive accuracy, a confusion matrix, the error rate, scoring rule results, logarithmic loss and quadratic loss, and spherical pay off, calibration results, and a "times surprised" table (indicating when the network was confident of its beliefs but was wrong).

---

## B.5 Bayesian statistical modeling

### B.5.1 BUGS

MRC/Imperial College

<http://www.mrc-bsu.cam.ac.uk/bugs>

BUGS Bayesian inference Using Gibbs Sampling provides Bayesian analysis of complex statistical models using Markov Chain Monte Carlo (MCMC). Its most advanced version is WinBUGS, which is available in a free educational version. Other versions are available for unix systems (Sun, SGI and Linux). The web pages provide thorough documentation for the different versions of the program. A BUGS email discussion list is also maintained.

The Markov Chain Monte Carlo (MCMC) methods available include univariate Gibbs sampling (BUGS 0.6) and a more sophisticated univariate Metropolis sampler (WinBUGS). Applications include generalized linear mixed models, latent variable models, modeling measurement error, handling missing data, and Bayesian updating.

BUGS provides a language for specifying a Bayesian network. A compiler then processes the model and data and sets up the sampling distributions required for the Gibbs sampling. Finally, appropriate sampling algorithms are implemented to simulate values of the unknown quantities in the model.

### B.5.2 First Bayes

Dept. of Probability and Statistics  
University of Sheffield  
<http://www.shef.ac.uk/~st1ao/lb.html>

First Bayes is program to assist people in learning elementary Bayesian Statistics. It is meant to *supplement* other educational material and is not a self-contained tutoring system. It runs under MS-Windows. First Bayes is offered free to anyone interested in teaching or learning Bayesian Statistics, provided it is not used for profit.

---

## B.6 Causal discovery programs

### B.6.1 Bayesware Discoverer

Bayesware Ltd.  
<http://www.bayesware.com>

Bayesware Discoverer is a new commercial causal discovery tool running on the MS Windows platforms. The program uses the Cooper and Herskovits formula of §8.2 to construct a Bayesian metric, adjusted to deal with incomplete data by employing probability intervals. It has a “Wizard-like” interface to assist.

This software is a commercial version of the Bayesian Knowledge Discoverer, developed at the Knowledge Media Institute of The Open University (UK), based on the work of Ramoni and Sebastiani (e.g., [231]). A limited educational version is available free.

### B.6.2 CaMML

Chris Wallace, Kevin Korb  
School of Computer Science, Monash University, Victoria 3800 Australia

<http://www.datamining.monash.edu.au/software/camml/index.shtml>

CaMML (Causal discovery via MML) was described in Chapter 8 in some detail. It is freely available as an executable download for Linux from the above web site. There are two different versions: CaMML-L, which learns linear Gaussian models, and CaMML, which learns discrete causal models. Both use the Metropolis sampling search algorithm. These versions have a fairly crude ASCII command-line interface. There is a project to reimplement CaMML inside the Monash CDMS (Core Data Mining Software) project, providing CaMML with a GUI interface and data visualization capabilities. When ready, this will be available from the above web site, as will any future developments from the CaMML project.

### B.6.3 TETRAD

Peter Spirtes, Clark Glymour and Richard Scheines  
Dept of Philosophy, Carnegie Mellon University  
<http://www.phil.cmu.edu/tetrad/>

TETRAD II was the first commercially available causal discovery program. Its PC algorithm is described in Chapter 6. It is available for purchase, with relevant information available on the website. TETRAD III adds Gibbs sampling to the functionality of TETRAD II, but retains the command-line interface. TETRAD IV has a graphical interface, running under the Java Runtime Environment. TETRAD III and IV are available for free download.

### B.6.4 WinMine

David Max Chickering  
Microsoft Research, Redmond, Washington 98052  
<http://research.microsoft.com/~dmax/WinMine/tooldoc.htm>

WinMine is a set of causal discovery programs for Windows 2000/NT/XP [47]. The majority of the programs are command-line executables that can be run in scripts. It includes GUIs for viewing learned or modeled Bayesian networks and for displaying classification trees (“decision trees”). It is freely downloadable, if it is not going to be used for commercial purposes.

WinMine can learn discrete Bayesian networks from sample data. It supports prior information in the form of partial variable orderings and forbidden arcs. There is also support for evaluating the learned models.