A Tutorial on Bayesian Belief Networks

Mark L Krieg

Surveillance Systems Division
Electronics and Surveillance Research Laboratory

DSTO–TN–0403

ABSTRACT

This tutorial provides an overview of Bayesian belief networks. The subject is introduced through a discussion on probabilistic models that covers probability language, dependency models, graphical representations of models, and belief networks as a particular representation of probabilistic models. The general class of causal belief networks is presented, and the concept of d-separation and its relationship with independence in probabilistic models is introduced. This leads to a description of Bayesian belief networks as a specific class of causal belief networks, with detailed discussion on belief propagation and practical network design. The target recognition problem is presented as an example of the application of Bayesian belief networks to a real problem, and the tutorial concludes with a brief summary of Bayesian belief networks.

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Edinburgh, South Australia, Australia 5111

Telephone:	*(08) 8259 5555*
Facsimile:	*(08) 8259 6567*

AR No. 012–084
December, 2001

A Tutorial on Bayesian Belief Networks

EXECUTIVE SUMMARY

A Bayesian belief network is a graphical representation of a probabilistic dependency model. It consists of a set of interconnected nodes, where each node represents a variable in the dependency model and the connecting arcs represent the causal relationships between these variables. Each node or variable may take one of a number of possible states or values. The belief in, or certainty of, each of these states is determined from the belief in each possible state of every node directly connected to it and its relationship with each of these nodes. The belief in each state of a node is updated whenever the belief in each state of any directly connected node changes.

Bayesian belief networks are particularly suited to the target recognition problem, where the category, identity and class of a target track are to be determined. Each of these three track attributes may be modelled by a hypothesis node, in which each state represents a different hypothesis. Evidence, such as Identification Friend or Foe (IFF) reports, Electronic Support (ES) data and track dynamics, is applied to the network through evidence nodes. On receipt of evidence, the belief in the state of the evidence node changes, causing changes in the belief of all nodes to ripple through the entire network, including the hypothesis nodes. In this way, the evidence updates the beliefs for each category, identity and class, and possibly the most likely state of each.

Author

Mark L Krieg
Surveillance Systems Division

Mark Krieg joined the Defence Science and Technology Organisation (DSTO) Australia in 1976 as a radio apprentice. From 1981 until 1987 he worked as a technical officer in the Communications and Electronic Engineering Division in the areas of communication networks, and control and instrumentation.

Dr Krieg obtained his BE(Elect) from the University of Adelaide in 1992 and his PhD from the same institution in 1998. He joined the Microwave Radar Division in 1992 where he worked in the radar signal and data processing area. He is currently a Senior Research Scientist attached to the Tracking and Sensor Fusion group of the Surveillance Systems Division, where he is pursuing research into multi-sensor tracking and fusion for defence applications.

Contents

Appendices

Figures

Glossary

acyclic graph A graph that contains no cycles, that is, at most one path exists between each pair of nodes in the graph.

anticipatory node A leaf node that has not been instantiated, that is, one awaiting evidence.

arc A link or edge joining two nodes of a graph.

Bayesian belief networks Directed acyclic graphs that represent probabilistic dependency models. The nodes represent stochastic variables and the arcs represent the Bayesian probabilistic relationships between these variables.

Bayesian probabilities Probabilities that are based on a person's belief of the likelihood of an event.

belief The posterior probability of each possible state of a variable, that is, the state probabilities after considering all the available evidence.

belief network A graphical representation of a model that captures the relationships between the model's variables.

binary variable A variable that has only two possible states.

causal belief networks Graphical networks that are represented by directed graphs.

causal polytree A singly connected tree in which each node may have multiple parent nodes.

causal support Evidential information that propagates through the Bayesian belief network in the direction of the arcs.

causation The influence that the state of one event has on the state of another event.

child A child of a variable is any other variable whose state is directly influenced by that variable. That is, immediate neighbours of the variable that are connected by arcs directed away from the variable.

chordal graph An undirected graph in which every cycle of length four or more contains a chord, that is, an edge connecting two non-consecutive nodes.

classical probabilities Probabilities that are determined from the results of repeated trials.

clique A set of variables that are all pairwise linked.

cluster tree A tree in which each node is a cluster of variables from the domain of interest and each variable in the domain appears in at least one node of the cluster tree.

clustering A technique in which variables are grouped into *clusters* to form singly connected networks or trees.

conditioned A variable is conditioned on another variable when its state is dependent on knowledge of the state of that other variable.

conditioning A technique where variables are selectively instantiated to break loops in a belief network.

cyclic graph A graph that contains at least one cycle, that is, multiple paths must exist between at least two of the nodes in the graph.

d-separation The blocking of the flow of evidence or information between nodes or sets of nodes in a network. It is used to represent independence in causal belief networks.

DAG isomorph A dependency model for which there exists a directed acyclic graph that is a perfect map of that model relative to d-separation.

decomposable model A probability model with a chordal minimal independency map.

dependency map A graph in which all connected nodes correspond to dependent variables in the model that it represents.

dependency model A rule that assigns truth values to the predicate that **X** is independent of **Y** given **Z** for the disjoint sets of variables **X**, **Y** and **Z**.

diagnostic support Evidential information or support that propagates through the Bayesian belief network against the direction of the arcs.

directed acyclic graph A graph that has directed arcs and no cycles.

directed graph A graph containing directed arcs between nodes.

dummy node A node representing evidence that bears upon another node of interest.

evidence node A node with a single value that has been observed with probability one.

explaining away Some knowledge of the state of a child of a variable may explain (or fail to explain) or provide some knowledge of the state of that variable. This may alter the confidence in the assumed state of that variable.

graph A set of nodes or vertices connected by a set of arcs or edges.

graphoid A graph that conforms to the symmetry, decomposition, weak union, contraction and intersection axioms for probabilistic dependencies.

hard evidence *see* specific evidence

hypothesis node A node whose state represents one of a number of alternative hypotheses.

independency map A graph in which all separated nodes correspond to independent variables in the model that it represents.

inference The calculation of posterior marginal distributions for variables of interest, given a set of observations or evidence.

instantiated node *see* evidence node

intermediate node A node inserted before a hypothesis variable to allow uncertainty to be modelled.

join tree A tree structure in which the nodes are the cliques of the graph that it represents.

leaf node A node whose arcs are directed toward it.

likelihood How often a particular event is expected to occur.

Markov networks Graphical networks that are represented by undirected graphs.

Markov tree A Markov network that has a tree structure.

maximal clique A maximal set of variables that are all pairwise linked.

minimal independency map An independency map that fails to be an independency map if any of its arcs are removed.

neighbours The neighbours of a node are those nodes that are directly connected to it.

parent A parent of a variable is any other variable that directly influences the state of that variable. That is, the immediate neighbours of the variable that are connected by arcs directed toward the variable.

perfect map A graph that is both a dependency and an independency map of the model that it represents.

predictive support *see* causal support

probabilistic model An encoding of probabilistic information that allows the computation of every well–formed sentence or proposition in accordance with the axioms of the probability language.

probability language A collection of semantics for describing the uncertainty relationships between variables.

relevance Two events become relevant if a common consequence is observed.

retrospective support *see* diagnostic support

root node A node whose arcs are directed away from it.

singly connected A network or graph in which each possible pair of variables or nodes is connected by a single path.

singly connected DAG A directed acyclic graph whose undirected graph is a tree.

soft evidence *see* virtual evidence

specific evidence Information that provides knowledge of the state of a variable with no uncertainty.

stochastic simulation A technique for computing probabilities by measuring how frequently specific events occur during simulation runs.

terminal node A node whose arcs are either directed away from or toward it, but not both.

triangulation The process of converting an undirected graph into a chordal graph.

uncertain evidence Specific evidence that does not directly identify the state of a variable.

undirected graph A graph in which the arcs have no explicit direction, that is, they exhibit bidirectional influence between nodes.

universe The set of all variables in a model representing all knowledge in the domain of interest.

virtual evidence Information that provides knowledge of the probability that a variable is in each of its possible states.

Abbreviations and Acronyms

BBN	Bayesian belief network
D-map	dependency map
DAG	directed acyclic graph
ES	Electronic Support
FLIR	Forward Looking Infrared
I-map	independency map
IFF	Identification Friend or Foe
NCTR	Non-Cooperative Target Recognition
pdf	probability density function
pmf	probability mass function
SIF	Selective Identification Function

Symbols

$\mathbf{1}$	The vector containing all ones.
$\mathbf{Bel}(A)$	The vector of belief values for each possible state of variable A.

$\mathrm{Bel}\,(a)$	The belief that variable A is in state a, given all the evidence.
E	An evidence node or variable.
$\mathbf{E}$	The set of all evidence nodes or variables.
$\mathbf{e}$	The set of all evidence, that is, the values of set of all evidence nodes or variables.
e	The value of an evidence node or variable.
$\mathbf{e}_X^+$	The evidence introduced through the parents of the variable X.
$\mathbf{e}_{W_jX}^+$	The evidence introduced through the arc linking X to its parent W_j.
$\mathbf{e}_X^-$	The evidence introduced through the children of the variable X.
$\mathbf{e}_{XY_i}^-$	The evidence introduced through the arc joining X to its child Y_i.
$\mathtt{G}\,(\mathbf{V}, \mathbf{E})$	The undirected graph containing the set of vertices $\mathbf{V}$ and edges $\mathbf{E}$.
$\mathtt{G}\left(\mathbf{V}, \vec{\mathbf{E}}\right)$	The directed graph with nodes $\mathbf{V}$ and directed edges $\vec{\mathbf{E}}$.
$\mathtt{G}_M\,(\mathbf{V}, \mathbf{E})$	The graph representing the dependency model M.
$\mathtt{I}\,(X, Z, Y)_M$	The mathematical representation of the dependency model M, in which X and Y are conditionally independent given Z.
$\mathbf{P}\,(A)$	The probability table containing the prior probabilities $\mathtt{p}\,(a)$.
$\mathtt{p}\,(a)$	Shorthand for $\mathtt{p}\,(A = a)$. Used where A is obvious from the context.
$\mathtt{p}\,(A = a)$	The probability that the random variable A takes the value a as its state.
$\mathbf{P}\,(A \mid B)$	The probability table containing the conditional probabilities $\mathtt{p}\,(a \mid b)$.
$\mathtt{p}\,(a \mid b)$	The conditional probability that $A = a$, given that $B = b$.
$pa(A)$	The set of parent variables of variable A.
$\mathbf{U}$	The universe or set of all variables in a model.
α	A normalising constant that ensures the total probability sums to unity.
β	An arbitrary constant.
$\boldsymbol{\delta}_{xx'}$	An evidence vector containing a single one and zeros elsewhere. The one indicates the state to which the variable is instantiated.
$\boldsymbol{\lambda}(X)$	The vector of evidential diagnostic or retrospective support for each possible state of X.
$\lambda(x)$	The likelihood representing diagnostic or retrospective support for the proposition $X = x$. λ messages are propagated through the BBN against the direction of the arcs.

$\boldsymbol{\lambda}_Y(X)$	The diagnostic support received from Y for all states of X.
$\lambda_Y(x)$	The diagnostic support received from Y for the proposition $X = x$.
$\boldsymbol{\pi}(X)$	The vector of evidential causal or predictive support for each possible state of X.
$\pi(x)$	The likelihood representing the causal or predictive support for the proposition $X = x$.
$\boldsymbol{\pi}_Y(X)$	The predictive support for all states of X that is sent to Y.
$\pi_Y(x)$	The predictive support for the proposition $X = x$ that is sent to Y.
$\langle X \mid Z \mid Y \rangle_G$	The graph G in which Z is the subset of all nodes intercepting all paths between the subsets of nodes X and Y.

1 Introduction

When making important decisions, it is prudent to seek advice from others who have expertise in the relevant field or domain of knowledge. These *experts* or *consultants* base their advice on accumulated knowledge, experience and any other available source of information [Jensen 1996]. There is considerable interest in automating this human reasoning process. Such automated systems, known as *expert systems*, model the knowledge domain of interest and its relationship with the available data. Current approaches to expert systems include rule–based, decision tree and artificial neural network representations for modelling the knowledge, and density estimation, classification, regression and clustering techniques for analysing these models [Heckerman 1996].

Probabilistic models have also also received considerable attention, particularly as they are able to cater for *uncertain* expert knowledge in their representation and reasoning. The knowledge in these models is represented by causal relationships between variables and their associated probability measures. These probability measures may be based on *classical probabilities*, *Bayesian probabilities* or a combination of both. Classical probability relies upon repeated trials to determine physical probabilities for particular events. This requires the processing of substantial quantities of data that may not always be available [Heckerman 1996]. The alternative Bayesian approach uses the degree of a person's belief that an event will occur, rather than the actual probability that the event will occur. These probabilities are known as Bayesian probabilities and are properties of the person, not the event. Prior probabilities may be determined for each event without the expense and difficulties associated with running repeated trials [Heckerman 1996].

Belief networks are graphical models that effectively model the knowledge domain, as opposed to rule based systems that model the expert. They provide a useful tool for inferring hypotheses from the available information by transmitting probabilistic information from the data to various hypotheses [Suermondt 1992]. *Bayesian belief networks*[1] are graphical models that use Bayesian probabilities to model the dependencies within the knowledge domain [Jensen 1996]. They are used to determine or infer the posterior marginal probability distributions for the variables of interest, given the observed information [Suermondt 1992]. As such, they do not include decisions or utilities that represent the preferences of the user,[2] rather it is left to the user to make decisions based on these probability distributions [Suermondt 1992]. The causal relationships in Bayesian belief networks allow the correlation between variables to be modelled and predictions to be made, even when direct evidence or observations are unavailable. Bayesian belief networks also blend prior knowledge with the data through the use of both prior probabilities and conditional probabilities. These characteristics of Bayesian belief networks give them an advantage over classical probability techniques [Heckerman 1996].

An example where Bayesian belief networks may be applied is in solving the target recognition problem. In this problem, Identification Friend or Foe (IFF) transponders provide incomplete and uncertain identity measurements and Electronic Support (ES)

[1]Bayesian belief networks are also referred to as Bayesian networks, Bayesian nets, probability influence diagrams and causal probability networks [Suermondt 1992].

[2]Although the standard representation does not include utility functions, these are often included in practical applications. Utilities and actions (decisions) are discussed by Jensen [1996, Ch. 6], and a brief comparison of Decision Theory and Bayesian belief networks is provided in Appendix E of this tutorial.

receivers provide uncertain class measurements. Identification information may also be inferred from the ES measurements and classification information may be inferred from the IFF measurements.

This tutorial is arranged as follows. This introductory section is followed by a section on probabilistic models (Section 2) that includes probability languages, dependency models, graphical representations of dependency models and belief networks. This leads into Section 3, which discusses causal networks, and Section 4, which covers Bayesian belief networks. The propagation of information through Bayesian belief networks is described in Section 5, and network design is discussed in Section 6. The application of Bayesian belief networks to the target recognition problem is discussed in Section 7, and a summary is presented in Section 8. Appendix A contains a description of the axioms of dependency models, Appendix B covers Markov networks, and Appendix C presents belief table algebra. An overview of adaptive networks and learning from data is provided in Appendix D, and a comparison of Bayesian belief networks with decision theory is provided in Appendix E.

2 Probabilistic Models

A *probabilistic model* is an encoding of the probabilistic information in some domain of interest, such that each well formed sentence, statement or proposition can be calculated in accordance with the axioms of a probability language [Pearl 1988, p. 33]. These sentences are represented by interrelated *random* or *stochastic variables* within the model. For example, the proposition

Target 1 is a F/A-18C

may be represented by a variable labelled *Target 1* that is in state F/A-18C. The list of possible states of *Target 1* may also include F-111, MiG-29 and P-3C, representing alternate propositions or hypotheses. A probabilistic model M is defined by the joint probability of all its variables or *universe*, that is, $M \triangleq \mathtt{p}(\mathbf{U})$.

Each variable in a probabilistic model may take any one of a set of mutually exclusive and collectively exhaustive states. Discrete valued variables have a finite number of possible states, for example, the set of known aircraft types. The probability distribution of a discrete variable is represented by a *probability mass function* (pmf), defined as

$$\mathtt{p}(A = a_1, a_2, \dots, a_n) = (x_1, x_2, \dots, x_n) \ : \ x_i \geq 0, i = 1, 2, \dots, n \quad \sum_{i=1}^{n} x_i = 1, \tag{2.1}$$

where x_i is the probability of variable A being in state a_i. This probability is denoted $\mathtt{p}(A = a_i)$, or simply $\mathtt{p}(a_i)$ if the variable is obvious from the context. Continuous valued variables take their state from an infinite set of possible values. For example, the speed of an aircraft may take any value between zero and the maximum speed for that type of aircraft. The probability distribution of a continuous variable is represented by a *probability density function* (pdf). As the probability of a continuous variable taking a specific value is zero, probabilities are expressed for a range of values, for example, $\mathtt{p}(A \leq a)$, $\mathtt{p}(a < A \leq b)$, etc.

In this tutorial, variables are denoted by uppercase letters, and their values or states by the corresponding lower case letter. For example, the variables X, Y, X_i and Θ take the values x, y, x_i and θ, respectively. Sets of variables, such as all air targets, and their values are denoted by bold-face upper case and lower case letters. For example, the states of the variables in sets $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{X}_i$, $\boldsymbol{\Theta}$ are denoted $\mathbf{x}$, $\mathbf{y}$, $\mathbf{x}_i$, $\boldsymbol{\theta}$, respectively.

2.1 Probability Language

Probability languages are well suited to reasoning under uncertainty; they provide a suitable framework for processing the uncertainty relationships between the variables of a probabilistic model. They promote consistency of inferred results and the knowledge base through their axiomatic basis, and they provide a convenient mechanism for presenting uncertain results [Suermondt 1992].

To illustrate how a probability language facilitates reasoning under uncertainty, consider the target identification application, in particular, the response received from interrogating an IFF transponder. The variable associated with this event may take one of *friend*, *neutral* or *no response* as its state. Each of these states will occur with some probability that depends on the state of other events, such as whether the aircraft belongs to the *own*, a *neutral* or the *enemy* force. In a probability language context, the deterministic statement:

The IFF will return a *no response* if the aircraft is hostile;

is replaced by the probabilistic statement:

The IFF will return a *no response*, with probability x, if the aircraft is hostile.

The IFF response may also affect the probability of other events that are influenced by it, for example, the *threat* status of the detected aircraft.

A probability language contains the four primitive relationships of *likelihood*, *conditioning*, *relevance* and *causation* [Pearl 1988]. The likelihood of an event is a measure of how likely or probable it is that the event will occur, that is, the chances of its occurrence. An event is conditional on a second event if its likelihood is changed by the knowledge of the second event's state. In other words, it is influenced by the state of the second event. Events A and B are said to be relevant to each other in the context of event C if adding the knowledge of C to the knowledge of B changes the likelihood of A. For example, two events become relevant when common consequence is observed, that is, A and B are relevant if both are causes of C. Causation conveys a pattern of dependence between events. For example, two consequences become independent upon learning the cause.

The following definitions provide an axiomatic basis for probabilistic dependencies [Pearl 1988]. Let $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ denote any three disjoint subsets of variables in the universe

$\mathbf{U}$. Defining $\mathtt{I}(\mathbf{X},\mathbf{Z},\mathbf{Y})_p$ as the representation of the relation $\mathbf{X}$ *is independent of* $\mathbf{Y}$ *given* $\mathbf{Z}$ in some probabilistic model p, that is,

$$\mathtt{I}(\mathbf{X},\mathbf{Z},\mathbf{Y})_p \quad \text{iff} \quad \mathtt{p}(\mathbf{x} \mid \mathbf{y},\mathbf{z}) = \mathtt{p}(\mathbf{x} \mid \mathbf{z}) \quad \text{and} \quad \mathtt{p}(y) > 0, \tag{2.2}$$

the following relationships hold:

$$\mathtt{I}(\mathbf{X},\mathbf{Z},\mathbf{Y})_p \Leftrightarrow \mathtt{p}(\mathbf{x},\mathbf{y} \mid \mathbf{z}) = \mathtt{p}(\mathbf{x} \mid \mathbf{z})\,\mathtt{p}(\mathbf{y} \mid \mathbf{z})\,; \tag{2.3a}$$

$$\mathtt{I}(\mathbf{X},\mathbf{Z},\mathbf{Y})_p \Leftrightarrow \mathtt{p}(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathtt{p}(\mathbf{x} \mid \mathbf{z})\,\mathtt{p}(\mathbf{y},\mathbf{z})\,; \tag{2.3b}$$

$$\mathtt{I}(\mathbf{X},\mathbf{Z},\mathbf{Y})_p \Leftrightarrow \exists\, f, g \colon \mathtt{p}(\mathbf{x},\mathbf{y},\mathbf{z}) = f(\mathbf{x},\mathbf{z})\, g(\mathbf{y},\mathbf{z})\,; \tag{2.3c}$$

where f and g represent arbitrary functions. Therefore, the joint probability of conditionally independent sets of events is the product of the individual conditional probabilities of these sets (2.3a). The joint probability of all conditionally independent sets of events and the conditioning sets may be written as the conditional probability of one set of conditionally independent events multiplied by the joint probability of all other events (2.3b) or, more generally, as the product of functions of each conditionally independent set of events together with the conditioning set (2.3c).

Each independency in a model p, $\mathtt{I}(\mathbf{X},\mathbf{Z},\mathbf{Y})_p$, must also satisfy the four independent axioms for probabilistic dependencies, namely *symmetry*, *decomposition*, *weak union* and *contraction*, as described in Appendix A. If the probabilities are strictly positive and the variables in $\mathbf{U}$ are not constrained by logical properties, then the independency $\mathtt{I}(\mathbf{X},\mathbf{Z},\mathbf{Y})_p$ must also satisfy the axiom of *intersection* [Pearl 1988].

2.2 Dependency Models

A *dependency model* is a probabilistic model in which the relationship between each of the variables is captured. That is, it captures the dependencies between the various events and data that the model represents. In particular, if $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are three disjoint sets of variables representing propositions in the knowledge domain, then a dependency model is a rule that determines the subset of triplets $(\mathbf{X},\mathbf{Z},\mathbf{Y})$ for which the assertion $\mathbf{X}$ *is independent of* $\mathbf{Y}$ *given* $\mathbf{Z}$ *is true* [Pearl 1988, p. 91]. The dependency model M is denoted $\mathtt{I}(\mathbf{X},\mathbf{Z},\mathbf{Y})_M$. Each of these sets of variables may also contain local dependencies that may be modelled in the same way, for example, $\mathbf{X}$ may contain the model $\mathtt{I}(\mathbf{X}_1,\mathbf{X}_3,\mathbf{X}_2)_{\mathbf{X}}$. This allows the hierarchical development of complex dependency models using local dependencies. All probability distributions are dependency models.

2.3 Graphical Representation of Dependency Models

A graph, denoted $\mathtt{G}(\mathbf{V},\mathbf{E})$, is a set of nodes or vertices $\mathbf{V}$ connected by a set of arcs or edges $\mathbf{E}$. A graph may be *undirected* or *directed*. The arcs of an undirected graph have no explicit direction; they represent bidirectional influence between the connected nodes. A directed graph has unidirectional or directed arcs, implying influence from one node to the other, but not vice versa. Directed arcs provide the mechanism for representing causation. Graphs may also be either *cyclic* or *acyclic*. An acyclic graph contains no cycles, that is, there is at most one path from any node that returns to that same node. A cyclic graph must contain at least one cycle. Undirected graphs and directed acyclic graphs are both *graphoids*, that is, they conform to the axioms for probabilistic dependencies, namely symmetry, decomposition, weak union, contraction and intersection [Pearl 1988].[3]

Each arc of a graph may have some *weight* or value associated with it, indicating the strength of the interaction between the two nodes that it connects. The nature of this weight is application dependent. For example, it may represent a cost associated with a particular action, the strength of a connection between two nodes or, in the case of probabilistic models, the probability that a particular event will occur.

Graphs may be used to represent dependency models. For example, the dependency model M comprising the variables $\mathbf{U}$ is represented by the graph $G \triangleq \mathtt{G}_M(\mathbf{U}, \mathbf{E})$. As seen in this example, the nodes of the graph correspond directly to the variables in the dependency model. The set of arcs $\mathbf{E}$ represent the conditional dependencies between these variables, that is, the joint probability function of M is encoded in $\mathbf{E}$. The topology of G reflects some of the properties of the model M, using vertex separation to represent various dependencies and independencies of M. The graph topology representing the dependency model $\mathtt{I}(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_M$ is denoted $\langle \mathbf{X} \mid \mathbf{Z} \mid \mathbf{Y} \rangle_G$, with $\mathbf{Z}$ representing the subset of all nodes that intercept all paths between the subsets of nodes $\mathbf{X}$ and $\mathbf{Y}$. In other words, $\mathbf{Z}$ separates $\mathbf{X}$ from $\mathbf{Y}$ [Pearl 1988].

Not all the dependencies and independencies of the model may be represented by the graph's topology. If all the connected nodes in G correspond to dependent variables in M, then G is a *dependency map* (D-map) of M. However, some dependent variables in M may appear as separated nodes in G, therefore the separations between the nodes of G do not necessarily represent independent variables in M. Conversely, G is an *independency map* (I-map) of M if all the separated nodes in G correspond to independent variables in M. In this case, some independent variables in M may appear as connected nodes in G and the connections between nodes in G do not always represent dependencies between the variables in M. An I-map of G is a *minimal I-map* if G ceases to be an I-map of M if any arc is removed. Every dependency model that satisfies the axioms of symmetry, decomposition and intersection (Appendix A) has a unique minimal I-map [Pearl 1988]. A *perfect map* of M is both a D-map and an I-map of M, with all connections and separations in the graph corresponding to the dependencies and independencies in the model, respectively. The necessary and sufficient conditions for a dependency model to have a perfect map are given in [Pearl 1988, p. 94].

2.4 Belief Networks

Belief networks are graphical representations of models that capture the relationships between the model's variables. They identify the variables that interact directly and simplify belief updating by *limiting each variable's processing to its local neighbourhood*, that is, those variables to which it is directly connected. This approach is based on the premise that what a variable does not directly see does not matter to it.

[3]Structures that satisfy just the symmetry, decomposition, weak union and contraction axioms are referred to as *semi-graphoids*. Unlike graphoids, semi-graphoids may represent probability models that are not strictly positive [Pearl 1988, p. 101].

Belief networks may use either directed or undirected graphs to represent a dependency model. The directed acyclic graph (DAG) provides a better representation, is more flexible, and is able to represent a wider range of probabilistic independencies than undirected graphs, particularly conceptually meaningful ones. For example, induced and transitive dependencies are unable to be modelled accurately by undirected graphs, but are easily represented in DAGs.[4]

Every dependency model that is isomorphic to a chordal graph is also isomorphic to a DAG.[5] Therefore, there is a class of probabilistic dependencies that can be represented by both DAGs and undirected graphs, namely those that form decomposable models. Such probability distributions have perfect maps in chordal graphs. This is further illustrated by the similarity of the axioms for DAGs and undirected graphs. DAGs satisfy the same axioms as probabilistic dependencies, namely symmetry, decomposition, intersection, weak union and contraction. They also satisfy the axioms of composition, weak transitivity and chordality.[6] The undirected graph implies all the DAG axioms except chordality; weak union is implied by strong union, composition and contraction are implied by intersection and strong union, and weak transitivity is implied by transitivity. This property is exploited in some belief network propagation schemes [Pearl 1988].

Networks that are represented by undirected graphs are referred to as *Markov networks*; these are described in Appendix B. *Causal networks* are represented by directed graphs and are discussed in Section 3.

3 Causal Belief Networks

A causal belief network is a directed acyclic graph (DAG) that represents a dependency model of the knowledge in a domain or universe of interest. The directional arcs in the network represent causal influences, and this causal nature allows independencies to be represented by the network. A dependency model M is causal, or a *DAG isomorph*, if there exists a DAG that is a perfect map of M relative to d-separation (Section 3.1). A necessary condition for a dependency model to be causal is that it be consistent with the axioms for DAGs, namely the axioms of symmetry, decomposition, composition, intersection, weak union, contraction, weak transitivity and chordality, as described in Appendix A.

The concept of causality or directionality in a network introduces a hierarchical structure to the network. This hierarchy is usually expressed in a manner similar to a family tree. Variables or nodes that are causes of a particular node are referred to as *parents* of that node. A *child* of a node is influenced directly by that node, that is it is a consequence of that node. A variable's parents and their parents, etc. are referred to as its *ancestors* and its children and their children, etc. are referred to as its *descendants*.

A causal belief network is made up of various types of nodes. A *root* node has no parents, and a *leaf* node has no children. Both root and leaf nodes are *terminal* nodes.

[4]Multiple causes are irrelevant to each other when the state of the common consequence is unknown, but all are relevant to that consequence. Therefore, multiple causes have a non-transitive relationship. When the state of the consequence is observed, a dependency is induced between the multiple causes. Non-transitive dependencies always induce dependencies between causes [Pearl 1988, p. 19].

[5]A dependency model is isomorphic to a graph when a graph exists that is a perfect map of that model.

[6]Not every probabilistic model is a DAG isomorph.

A *hypothesis* node is any node whose state represents one of a number of alternative hypotheses. An *evidence* node, also known as a *finding node*, *instantiated node*, *observed node*, or *fixed value node*, represents a variable with a single value that has been observed with probability one [Suermondt 1992]. Such a node is said to have been instantiated with *specific* or *hard evidence*. *Virtual evidence*, also known as *intangible* or *soft evidence*, only provides a probabilistic indication of a node's state. Although not generally the case, virtual evidence is usually assumed to be independent of evidence from other sources and is expressed in terms of likelihood ratios or relative magnitudes [Pearl 1988]. It is entered into a node by adding an evidence node as a child and choosing conditional probabilities that reflect the virtual evidence. This child node is referred to as a *dummy node*. An example of virtual evidence would be the prior probability that a *no response* from an IFF interrogation indicates a hostile identity with probability 0.8 and a friend identity with probability 0.2 to account for equipment failure and stealth operation. *Uncertain evidence*, although certain, does not directly identify the state of a variable. It too must be entered through a dummy variable [Pearl 1988]. For example, an IFF interrogation may produce a particular Selective Identification Function (SIF) code indicating a neutral identity. However, this does not mean that the target is neutral, only that a neutral response has been received. This response may have been the result of a hostile target imitating a neutral target to avoid identification. Therefore, an *intermediate variable*, representing the type of response received, must be inserted before the target identity variable to allow these uncertainties to be modelled. It is assumed that only leaf nodes can be instantiated. This is not a restrictive assumption, as evidence can be entered at any node using a dummy leaf node. An *anticipatory* node is a leaf node that has not been instantiated. Evidence is denoted $e = (e_1, e_2, \dots)$, where e_n denotes the probability that the evidence variable E is in the n^{th} state. Specific or hard evidence is represented by

$$e = (0, \dots, 0, 1, 0, \dots, 0) \triangleq \boldsymbol{\delta}_{xx'} = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{if } x \neq x' \end{cases}. \tag{3.1}$$

The position of the '1' indicates the state to which the variable is instantiated. The set of all evidence variables is denoted $\mathbf{E}$ with values $\mathbf{e}$.

The fixed conditional probabilities assigned to each arc, $\mathtt{p}(a \mid b)$, are stored in a probability table $\mathbf{P}(A \mid B)$ at the child node. Root nodes have unconditional prior probabilities $\mathtt{p}(a)$ that are stored in the probability table $\mathbf{P}(A)$ at that node. The conditional and prior probabilities are part of the design of the network and do not change during inference. The dynamic values of the node probabilities shall be denoted $\mathrm{Bel}(a)$. This latter value reflects the overall belief in the proposition $A = a$ given all the evidence, that is,

$$\mathrm{Bel}(a) \triangleq \mathtt{p}(a \mid \mathbf{e}). \tag{3.2}$$

As such, the belief values represent the posterior probabilities. The posterior probability of any variable may be obtained through marginalisation. A vector of belief values for all possible states of A is denoted $\mathbf{Bel}(A)$.

3.1 Dependencies in Causal Networks

The concept of *d-separation* is used to represent conditional independence in causal networks. The rules for d-separation in basic networks, as discussed in this section, can be used to determine dependence and independence in larger more complex networks. If two variables are not d-separated, then they are considered to be *d-connected* and dependent [Jensen 1996].

Serial Connections Figure 3.1 shows a serial connection consisting of an allegiance variable A, a platform variable P and a radar variable R. It shows that the type of radar is determined by the platform on which it is housed, which is in turn determined by the allegiance of that platform. Obviously, knowing the allegiance will increase our belief in the type of platform and subsequently the type of radar. Similarly, knowing the type of radar allows us to infer the type of platform and the likely allegiance, because we have knowledge of the types of platforms carrying that type of radar and the allegiance of each type of platform. However, if the type of platform is known, knowing the allegiance does not increase our knowledge of the radar, this being dependent only on the type of platform. Similarly, we are able to infer the allegiance from the type of platform; the type of radar does not add any additional information to the inference of the allegiance. Knowledge of the platform type has blocked the flow of information between A and R. Therefore, if P is instantiated, A and R are d-separated by P, or conditionally independent given P.

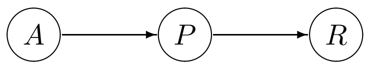


Figure 3.1: A serial belief network connection

Diverging Connections Where a node has multiple child nodes, influence can pass between all the child nodes of the parent node. That is, knowledge of the state of the parent node can be inferred if the state of any child node is known. In turn, as the parent node is a cause of its children, knowledge of the state of the other child nodes can be inferred from knowledge of the state of the parent node. If the state of the parent node is known, then there is no inference to be made from its children, and the belief in the state of any child node is not influenced by knowledge of the state of any other child of that parent node.

For example, consider the network in Figure 3.2, where a platform P produces a primary radar return R, a secondary radar (IFF) response (or no response) S and an emitter classification from an ES receiver C. If the platform type is not known, it can be inferred from any of its child nodes, if their states are known. This inferred state of P can then be used to infer the state of any unknown child node. However, if the platform type is known, then the state of each child node can be inferred from P, and any knowledge of the state of any other child node does not influence the belief in the state of that node. Therefore, if P is instantiated, R, S and C are d-separated by P, or conditionally independent given P.

Converging Connections The converging connection is more complex. Consider the example in Figure 3.3, where variables $T_1, T_2, \ldots, T_n$ represent the presence of targets 1

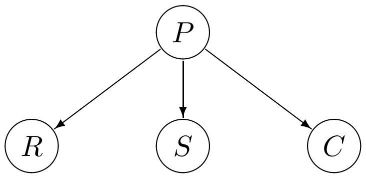


Figure 3.2: A diverging belief network connection

to n. Variable R, represents the event that a radar has detected a target. Assuming that a target must be present for a radar detection (ignoring false alarms), R depends on all of $T_1, T_2, \ldots, T_n$, even though the presence of each target is independent of the others. However, if we know that a radar detection has occurred and we also know that T_1 is present, then our belief that any of $T_2, T_3, \ldots, T_n$ are also present reduces. That is, if we have some knowledge of the state of R, then some knowledge of $T_2, T_3, \ldots, T_n$ may be inferred from the state T_1 and the knowledge of R. Therefore, $T_1, T_2, \ldots, T_n$ are no longer independent or d-separated; they are d-connected, or conditionally dependent given R. This phenomenon is known as *explaining away* [Jensen 1996]. D-connection in converging connections requires some knowledge of the state of the connection variable, that is, it or at least one of its descendents must have received either hard or soft evidence.

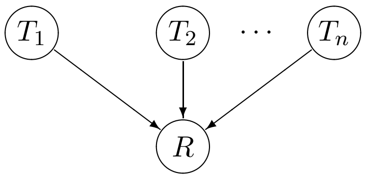


Figure 3.3: A converging belief network connection

Summary The above three cases cover all the ways that evidence may be transmitted through a variable. The above rules are summarised as follows:

> Two variables A and B in a causal network are *d-separated* if, for all paths between A and B, there is an intermediate variable V such that either the connection is serial or diverging and the state of V is known, or the connection is converging and we have no knowledge of V and its descendents.

As a result of d-separation, an evidence node does not pass information between its children, or between its children and parents. Also, a node cannot pass information between its parents unless it is an evidence node or at least one of its descendants is an evidence node [Suermondt 1992].

4 Bayesian Belief Networks

A Bayesian belief network (BBN) is a specific type of causal belief network. As for any causal belief network, the nodes represent stochastic variables and the arcs identify direct causal influences between the linked variables [Suermondt 1992, p. 12]. It is the use of *Bayesian* calculus to determine the state probabilities of each node or variable from the predetermined conditional and prior probabilities that distinguishes Bayesian belief networks from other causal belief networks.

The directed acyclic graph $\mathtt{G}\left(\mathbf{U},\vec{\mathbf{E}}\right)$ is a Bayesian belief network if, and only if, it is a minimal I-map of the model M representing the probability distribution $\mathtt{p}\left(\mathbf{U}\right)$, where $\mathbf{U}$ is the set of all variables in M. For any ordering $X_1, X_2, \ldots, X_n$ of the variables in the probability distribution $\mathtt{p}\left(\mathbf{U}\right)$, a DAG in which a minimal set of predecessor variables are designated as parents of each variable X_i, such that

$$\mathtt{p}\left(x_i \mid \mathbf{W}\right) \qquad \mathbf{W} \in \{X_1, X_2, \ldots, X_{i-1}\}, \tag{4.1}$$

is a Bayesian belief network of that probability distribution. If the distribution is strictly positive, then all of the parent sets are unique for that ordering [Pearl 1988, p. 119]. Although the actual structure of a Bayesian network depends on the particular ordering of the variables used, each such network is an I-map of P, and the resulting DAG represents many of the independencies in M. All orderings that are consistent with the direction of the arrows in a DAG produce the same network topology. For a DAG to be a Bayesian network of M, it is necessary and sufficient that each variable be conditionally independent of all of its non-descendents given its parents, and that no proper subset of its parents also satisfies this condition [Pearl 1988, p. 120]. Although many different Bayesian belief networks may be used to represent the same probabilistic model, causation imposes standard orderings on the variables when representing experimental knowledge, producing identical network topologies for the same experimental knowledge.

In practice, we usually only have an intuitive understanding of the major constraints in a domain of interest. However, we can easily identify the variables that directly influence other variables, that is, we find it easier to understand the local interaction of variables than the domain as a whole. Bayesian belief networks are conducive to this understanding because they are constructed such that the parents of each variable are direct causes. This simplifies the process of determining the weights for the arcs of the network; it is only necessary to assess the conditional probabilities [Pearl 1988, p. 124]. This is easier than quantifying a Markov network because the conditional probabilities relate to conceptual relationships in one's mind.

Bayesian calculus is the probability language used for Bayesian belief networks. It is framed on the basic axioms of probability:

$$0 \leq \mathtt{p}\left(a\right) \leq 1; \tag{4.2a}$$

$$\mathtt{p}\left(S\right) = 1 \qquad \text{where } S \text{ is the sample space;} \tag{4.2b}$$

$$\mathtt{p}\left(a \cup b\right) = \mathtt{p}\left(a\right) + \mathtt{p}\left(b\right) \qquad \text{if } A \text{ and } B \text{ are mutually exclusive.} \tag{4.2c}$$

The basic concept in the Bayesian treatment of uncertainties in causal networks is *conditional probability*, that is:

Given the event B, the probability of the event A is x.

Mathematically, this is equivalent to $\mathtt{p}\,(a \mid b) = x$. This means that if $B = b$, and *everything* else known is irrelevant for $A = a$, then $\mathtt{p}\,(a) = x$. The fundamental rule for probability calculus is

$$\mathtt{p}\,(a \mid b)\,\mathtt{p}\,(b) = \mathtt{p}\,(a, b)\,. \tag{4.3}$$

In the real world, all events are conditioned by some context $C = c$, therefore (4.3) may be written

$$\mathtt{p}\,(a \mid b, c)\,\mathtt{p}\,(b \mid c) = \mathtt{p}\,(a, b \mid c)\,, \tag{4.4}$$

which can also be considered an axiom of probability calculus. Bayes' rule may also be conditioned on C to give

$$\mathtt{p}\,(b \mid a, c) = \frac{\mathtt{p}\,(a \mid b, c)\,\mathtt{p}\,(b \mid c)}{\mathtt{p}\,(a \mid c)}. \tag{4.5}$$

The left hand side of (4.5) is the posterior probability of b, derived from the likelihood $\mathtt{p}\,(a \mid b, c)$ and prior probability $\mathtt{p}\,(b \mid c)$. $\mathtt{p}\,(a \mid c)$ is a normalising factor and, as it is often not directly available, is replaced by $\int_b \mathtt{p}\,(a \mid b, c)\,\mathtt{p}\,(b \mid c)\,db$ and $\sum_b \mathtt{p}\,(a \mid b, c)\,\mathtt{p}\,(b \mid c)$ for continuous and discrete distributions, respectively. To illustrate this, consider the hypothesis that event $A = a$ given that $B = b$. Let H represent the event $A = a$ and let the evidence $\mathbf{e}$ be $B = b$. The *posterior probability* of the hypothesis H given the evidence $\mathbf{e}$, that is, the probability after considering the evidence, is written

$$\mathtt{p}\,(H \mid \mathbf{e}) = \frac{\mathtt{p}\,(\mathbf{e} \mid H)\,\mathtt{p}\,(H)}{\mathtt{p}\,(\mathbf{e})}\,, \tag{4.6}$$

where $\mathtt{p}\,(\mathbf{e} \mid H)$ is the likelihood of the evidence $\mathbf{e}$ occurring if the hypothesis H is true, $\mathtt{p}\,(\mathbf{e})$ is the *prior probability* of the evidence and $\mathtt{p}\,(H)$ is the prior probability that H is the correct hypothesis in the absence of evidence. As H is a cause of $\mathbf{e}$, this illustrates how Bayes' rule may be employed to propagate the probabilities of parent events given the states of their children, that is, backward propagation.

Each variable A with parents $B_1, B_2, \ldots, B_n$ in a Bayesian belief network has an attached conditional probability table $\mathbf{P}\,(A \mid B_1, B_2, \ldots, B_n)$. This table contains the probabilities for all possible combinations of the states of A and its parents. To reflect the interactions between the influence of multiple parents, a single conditional probability is assigned to each variable. For example, variable A with parents B_1 and B_2 is assigned the probability table $\mathbf{P}\,(A \mid B_1, B_2)$, and not two separate probability tables $\mathbf{P}\,(A \mid B_1)$ and $\mathbf{P}\,(A \mid B_2)$. Variables without parents are assigned *unconditional prior* probabilities. Although prior probabilities have been criticised as a source of unwanted bias, they are an integral part of human uncertainty reasoning [Jensen 1996, p. 19].

The joint probability distribution of the network is the joint probability of all variables or nodes in the network. Using the *chain rule*, this may be expressed as the product of the prior probabilities of all the root nodes and the conditional probabilities of all the other nodes in the network. That is, for a Bayesian belief network of variables $\mathbf{U} = (A_1, A_2, \ldots, A_n)$, and denoting $pa(A_i)$ as the parent set of A_i,[7]

$$\mathtt{p}(\mathbf{U}) = \mathtt{p}(A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n) = \prod_i \mathtt{p}(a_i \mid pa(A_i)) \, . \tag{4.7}$$

5 Propagation in Bayesian Belief Networks

A Bayesian belief network represents a complete probabilistic model of all the variables in a specific domain. Therefore, it contains all the information required to answer any probabilistic question about any variable in that domain. These questions are usually restricted to determining the most likely hypothesis or, more specifically, the belief in, or probability of, each possible hypothesis, given the available observations or evidence. The application of evidence is ongoing as the observations continue to arrive over a period of time. Therefore, the belief network is dynamic and the most likely hypotheses, and their probabilities, are likely to change.

The effect of the available evidence on any variable (hypothesis) may be ascertained by marginalising the joint probability of the entire network to obtain the posterior probability of that variable. This technique, known as *node elimination*, is computationally expensive, provides no insight on the impact of the evidence on the hypotheses, and spurious dependencies may be created by the removal of variables through marginalisation [Pearl 1988]. An alternative that overcomes these limitations is for each node to update the posterior probabilities for each of its hypotheses on the receipt of data messages from its immediate neighbours. The effect of evidence is transmitted throughout the network by allowing each variable to continually update its state by comparing the state of its neighbours against local constraints; if the local constraints are satisfied, no activity takes place, otherwise action is taken to correct the constraint violation and messages are sent to the immediate neighbours of that variable. For example, consider the hypothesis variable I representing track identity, which is dependent on an observation variable P representing the observed platform type, as illustrated in Figure 5.1. The enemy force may not contain F-14s, therefore a local constraint for the track identification node is: if the platform type P is F–14, then the identity cannot be *hostile.* If P informs I that it is in state F–14 with some probability, then I will change its belief in each of its possible states to reflect a low likelihood for the *hostile* state.

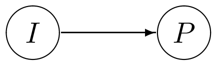


Figure 5.1: Bayesian belief network for track identification from platform data

Under certain conditions, the propagation of evidence through the belief network may prematurely terminate, that is, the network may become *deadlocked.* The removal of cycles from the belief network to form acyclic network topologies reduces the likelihood

[7]The parent set is empty for root nodes, that is, $pa(A_i) = \phi$.

of deadlock, and it can be eliminated entirely by careful selection of the local constraints [Pearl 1988]. Tree structures provide useful acyclic networks in which deadlock is easily eliminated. In these structures, information is propagated bi-directionally through the tree until the local constraints for each variable are met.

The posterior probability, or *belief*, for variable $X = x$ is $\mathrm{Bel}(x) \triangleq \mathtt{p}(x \mid \mathbf{e})$. The evidence influencing X is separated into two disjoint subsets, $\mathbf{e}_X^-$ denoting the evidence introduced through the arcs between X and its children, and $\mathbf{e}_X^+$ denoting the evidence introduced through the arcs between X and its parents. These may be further divided into $\mathbf{e}_{XY_i}^-$ for the evidence introduced through the arc to child Y_i, and $\mathbf{e}_{W_jX}^+$ as that introduced through the arc from parent W_j. If X is a root node, $\mathbf{e}_X^+$ is the background knowledge or prior probability. Alternatively, if X is a leaf node, then $\mathbf{e}_X^-$ is $\boldsymbol{\delta}_{xx'}$ if it is instantiated with x' or $\alpha\mathbf{1}$ otherwise.[8] Applying Bayes' Rule conditioned on $\mathbf{e}_X^+$ (4.5), and assuming that $\mathbf{e}_X^-$ and $\mathbf{e}_X^+$ are independent, the belief may be written

$$\mathrm{Bel}(x) = \mathtt{p}\left(x \mid \mathbf{e}_X^-, \mathbf{e}_X^+\right) = \frac{\mathtt{p}\left(\mathbf{e}_X^- \mid x, \mathbf{e}_X^+\right)\mathtt{p}\left(x \mid \mathbf{e}_X^+\right)}{\mathtt{p}\left(\mathbf{e}_X^- \mid \mathbf{e}_X^+\right)} = \frac{\mathtt{p}\left(\mathbf{e}_X^- \mid x\right)\mathtt{p}\left(x \mid \mathbf{e}_X^+\right)}{\mathtt{p}\left(\mathbf{e}_X^-\right)}. \tag{5.1}$$

Defining $\pi(x) = \mathtt{p}\left(x \mid \mathbf{e}_X^+\right)$ and $\lambda(x) = \mathtt{p}\left(\mathbf{e}_X^- \mid x\right)$, the belief becomes

$$\mathrm{Bel}(x) = \alpha\,\pi(x)\lambda(x), \tag{5.2}$$

where α represents a normalising constant, in this case

$$\alpha = \left[\mathtt{p}\left(e_X^-\right)\right]^{-1} = \left[\sum_x \pi(x)\lambda(x)\right]^{-1}. \tag{5.3}$$

$\lambda(x)$ is the *likelihood* representing *diagnostic* or *retrospective* support for the proposition $X = x$. $\boldsymbol{\lambda}(X)$ vectors or messages, where each element of $\boldsymbol{\lambda}(X)$ corresponds to a different state of X, are passed up the network in the opposite direction to the arrows on the arcs. Conversely, $\pi(x)$ may be considered as the prior certainty of x given the the prior evidence $\mathbf{e}_X^+$, and it represents the *causal* or *predictive* support for $X = x$. $\boldsymbol{\pi}(X)$ messages, also representing each possible state of X, are passed down the network in the direction of the arrows. For simplicity, it is assumed that all evidence is introduced into the network through leaf nodes, that is, all evidence on non-leaf nodes is represented by adding child leaf nodes.

Propagation will be discussed for trees, that is, singly connected DAGs where each node has at most one parent. This will be followed by a description of approaches for more general network structures.

5.1 Causal Trees

Consider the serial or chain network in Figure 5.2, where $\boldsymbol{\lambda}(Y) = \mathtt{p}\left(\mathbf{e}_Y^- \mid Y\right)$ will take either $\boldsymbol{\delta}_{xx'}$ or $\alpha\,\mathbf{1}$ as its value. Now,

$$\lambda(x) = \mathtt{p}\left(\mathbf{e}_X^- \mid x\right) = \sum_y \mathtt{p}(y \mid x)\,\mathtt{p}\left(\mathbf{e}_Y^- \mid y\right) \Rightarrow \boldsymbol{\lambda}(X) = \boldsymbol{\lambda}(Y) \cdot \mathbf{P}(Y \mid X), \tag{5.4}$$

[8] $\mathbf{1} \triangleq [1, 1, \ldots, 1]^{\mathsf{T}}$ represents equi-likely states, and α is a normalising constant.

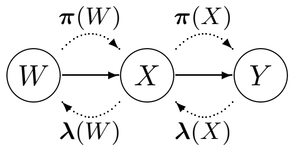


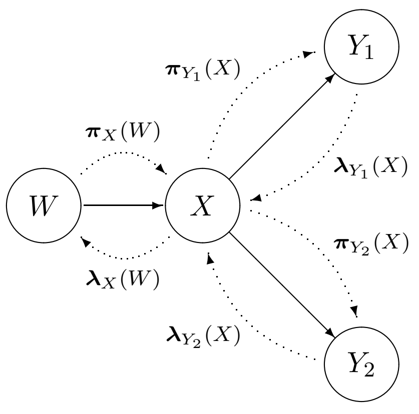
Figure 5.2: Propagation in a chain network

where $\mathbf{A} \cdot \mathbf{B}$ is defined in (C4) of Appendix C.1. Similarly, $\boldsymbol{\lambda}(W) = \boldsymbol{\lambda}(X) \cdot \mathbf{P}(X \mid W)$. Note that $\boldsymbol{\lambda}(X)$ is calculated from the probability table $\mathbf{P}(Y \mid X)$ attached to node Y and is calculated at node Y. To complete the belief propagation, consider the propagation of the $\boldsymbol{\pi}$ messages, starting at node W. Node W is a root node and its only *evidence* is its prior probability distribution. Therefore, $\pi(w) = \mathtt{p}\left(w \mid \mathbf{e}_W^+\right) = \mathtt{p}(w)$, and

$$\pi(x) = \mathtt{p}\left(x \mid e_X^+\right) = \sum_w \mathtt{p}(x \mid w)\, \mathtt{p}\left(w \mid \mathbf{e}_W^+\right) \Rightarrow \boldsymbol{\pi}(X) = \boldsymbol{\pi}(W) \cdot \mathbf{P}(X \mid W). \qquad (5.5)$$

Similarly, $\boldsymbol{\pi}(Y) = \boldsymbol{\pi}(X) \cdot \mathbf{P}(Y \mid X)$. The $\boldsymbol{\pi}$ messages are propagated as illustrated in Figure 5.2, and the belief at each node can be calculated using (5.2). The appropriate $\boldsymbol{\lambda}$ and $\boldsymbol{\pi}$ messages are re-propagated if evidence is added or changed.

Now consider a case of multiple child nodes, as illustrated in Figure 5.3. The additional problems introduced here are combining the $\boldsymbol{\lambda}$s as they are propagated up the tree and separating the $\boldsymbol{\pi}$s as they are propagated down the tree.



Y_1
$\boldsymbol{\pi}_{Y_1}(X)$
$\boldsymbol{\pi}_X(W)$
$\boldsymbol{\lambda}_{Y_1}(X)$
W
X
$\boldsymbol{\pi}_{Y_2}(X)$
$\boldsymbol{\lambda}_X(W)$
$\boldsymbol{\lambda}_{Y_2}(X)$
Y_2

Figure 5.3: Propagation in a tree network

Starting with the $\boldsymbol{\lambda}$s, the evidence on the sub-tree at X is partitioned into the disjoint sets $\mathbf{e}_{XY_1}^-$ and $\mathbf{e}_{XY_2}^-$. Therefore,

$$\lambda(x) = \mathtt{p}\left(\mathbf{e}_{XY_1}^- \mid x\right) \mathtt{p}\left(\mathbf{e}_{XY_2}^- \mid x\right) \triangleq \lambda_{Y_1}(x)\lambda_{Y_2}(x) \Rightarrow \boldsymbol{\lambda}(X) = \boldsymbol{\lambda}_{Y1}(X)\boldsymbol{\lambda}_{Y2}(X). \qquad (5.6)$$

Node X then transmits $\boldsymbol{\lambda}_X(W) = \boldsymbol{\lambda}(X) \cdot \mathbf{P}(X \mid W)$ to node W. Assuming we have $\boldsymbol{\pi}(X) = \boldsymbol{\pi}_X(W) \cdot \mathbf{P}(X \mid W)$, it is now necessary to split $\boldsymbol{\pi}(X)$ into individual messages for each child of X, that is Y_1 and Y_2. Consider the predictive support for $X = x$ that is passed to node Y_1, namely,

$$\pi_{Y_1}(x) = \mathtt{p}\left(x \mid \mathbf{e}_X^+, \mathbf{e}_{Y_2}^-\right). \tag{5.7}$$

Using Bayes' Rule,

$$\pi_{Y_1}(x) = \alpha\,\mathtt{p}\left(\mathbf{e}_{Y_2}^- \mid x\right)\mathtt{p}\left(x \mid \mathbf{e}_X^+\right) = \alpha\lambda_{Y_2}(x)\pi(x). \tag{5.8}$$

More generally, for the k^{th} child of K children of X,

$$\pi_{Y_k}(x) = \alpha \prod_{j=1\backslash k}^{K} \lambda_{Y_j}(x)\pi(x) = \alpha\frac{\mathrm{Bel}\,(x)}{\lambda_{Y_k}(x)} \Rightarrow \boldsymbol{\pi}_{Y_k}(X) = \alpha\frac{\mathbf{Bel}\,(X)}{\boldsymbol{\lambda}_{Y_k}(X)}, \tag{5.9}$$

allowing the same message to be passed to all children, namely $\mathbf{Bel}\,(X)$, for determination of $\boldsymbol{\pi}_k(X)$ at each child.

5.2 Causal Polytrees

Causal polytrees are *singly connected* trees in which each node may have multiple parents. They may be thought of as collections of causal trees, fused at the nodes where the arrows converge.

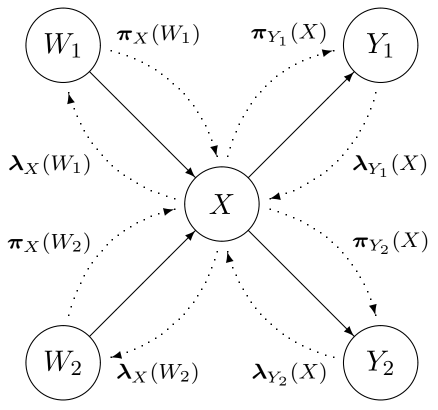

W_1
$\boldsymbol{\pi}_X(W_1)$
$\boldsymbol{\pi}_{Y_1}(X)$
Y_1
$\boldsymbol{\lambda}_X(W_1)$
$\boldsymbol{\lambda}_{Y_1}(X)$
X
$\boldsymbol{\pi}_X(W_2)$
$\boldsymbol{\pi}_{Y_2}(X)$
W_2
$\boldsymbol{\lambda}_X(W_2)$
$\boldsymbol{\lambda}_{Y_2}(X)$
Y_2

Figure 5.4: Propagation in a causal polytree network

To illustrate the propagation in this type of network, consider the example in Figure 5.4. The additional problems introduced by this type of structure are combining the $\boldsymbol{\pi}$ messages from multiple parent nodes and splitting the $\boldsymbol{\lambda}$ messages between multiple parent nodes. Consider the variable X that has two parents, W_1 and W_2, and two children, Y_1 and Y_2. The evidence obtained from the sub-tree rooted at X is the union of evidence obtained from each arc connecting it to its children, that is, $\mathbf{e}_{XY_1}^-$ and $\mathbf{e}_{XY_2}^-$. Similarly, the evidence from the rest of the network can be separated into that obtained from each arc linking X to each of its parents, namely $\mathbf{e}^{+}_{W_1X}$ and $\mathbf{e}^{+}_{W_2X}$ [Pearl 1988].

The $\boldsymbol{\lambda}$ messages, $\boldsymbol{\lambda}(X) = \boldsymbol{\lambda}_{Y_1}(X)\boldsymbol{\lambda}_{Y_2}(X)$ must be split for transmission to the parents of X, namely $\boldsymbol{\lambda}_X(W_1)$ for W_1 and $\boldsymbol{\lambda}_X(W_2)$ for W_2. Therefore, denote $\lambda_X(w_1) = \mathtt{p}\left(\mathbf{e}^{-}_{W_1X} \mid w_1\right)$ and separate the evidence $\mathbf{e}^{-}_{W_1X}$ into disjoint sets, namely that associated with the sub-tree rooted at X, $\mathbf{e}^{-}_{X}$, and that associated with all other parents of X, in this case $\mathbf{e}^{+}_{W_2X}$. Then,

$$\lambda_X(w_1) = \mathtt{p}\left(\mathbf{e}^{-}_{W_1X} \mid w_1\right) = \mathtt{p}\left(\mathbf{e}^{-}_{X}, \mathbf{e}^{+}_{W_2X} \mid w_1\right), \tag{5.10}$$

and conditioning on X and W_2 and applying Bayes' Rule gives

$$\lambda_X(w_1) = \beta \sum_x \mathtt{p}\left(\mathbf{e}^{-}_{X} \mid x\right) \sum_{w_2} \mathtt{p}\left(w_2 \mid \mathbf{e}^{+}_{W_2X}\right) \mathtt{p}\left(x \mid w_1, w_2\right), \tag{5.11}$$

where β is an arbitrary constant. In general, for n parents,

$$\lambda_X(w_i) = \beta \sum_x \lambda(x) \sum_{w_k : k \neq i} \mathtt{p}\left(x \mid w_1, w_2, \ldots, w_n\right) \prod_{k=1\backslash i}^{n} \pi_X(w_k), \tag{5.12}$$

where $\pi_X(w_k) = \mathtt{p}\left(w_k \mid \mathbf{e}^{+}_{W_kX}\right)$. The predictive support is provided through the arcs from the parents of X, that is, $\pi(x) = \mathtt{p}\left(x \mid \mathbf{e}^{+}_{X}\right) = \mathtt{p}\left(c \mid \mathbf{e}^{+}_{W_1X}, \mathbf{e}^{+}_{W_2X}\right)$. Conditioning on W_1 and W_2, this becomes

$$\pi(x) = \sum_{w_1} \sum_{w_2} \mathtt{p}\left(x \mid w_1, w_2\right) \pi_X(w_1)\pi_X(w_2). \tag{5.13}$$

In general, for n parents, this may be written,

$$\pi(x) = \sum_{w_1, w_2, \ldots, w_n} \mathtt{p}\left(x \mid w_1, w_2, \ldots, w_n\right) \prod_{i=1}^{n} \pi_X(w_i). \tag{5.14}$$

5.3 Practical Methods

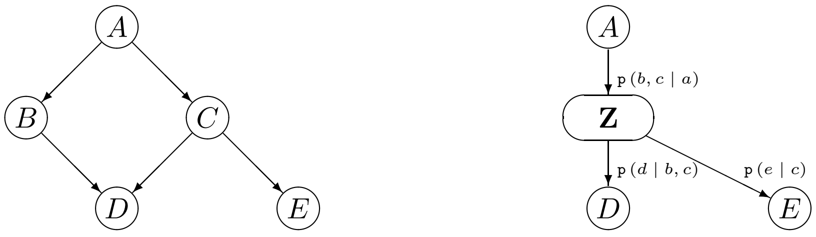
As is evident from the summations in (5.12) and (5.14), the introduction of multiple parents significantly increases the computational requirements for belief propagation. To overcome this problem, alternative methods are usually employed. These methods include *clustering* techniques that convert the network into a tree topology to take advantage of the lower complexity of propagation in trees and the *noisy-OR-gate* for binary variables.

Another problem is the presence of loops or undirected cycles in the underlying Markov network, that is, cycles that appear when the directed arcs are replaced by undirected arcs. The presence of loops indicates that the network is not singly connected and that local propagation methods are not suitable. In particular, messages may circulate indefinitely around these loops, never converging to a stable equilibrium. Techniques for dealing with this problem include *clustering*, *conditioning* and *stochastic simulation*.

The clustering, conditioning, stochastic simulation and noisy-OR-gate techniques are presented in the succeeding sections.

5.3.1 Clustering

The process of *clustering* involves grouping variables into *compound* variables to produce a singly connected network known as a *cluster tree.* A cluster tree, over a universe $\mathbf{U}$, is a tree in which the nodes are clusters of variables from $\mathbf{U}$, and each variable in $\mathbf{U}$ appears in at least one node [Pearl 1988]. As an example, the loop $ABCD$ in Figure 5.5(a) could be broken by combining B and C into a single compound variable $\mathbf{Z}$, as shown in Figure 5.5(b). Belief propagation occurs as for any graph with a tree topology, although it is necessary to marginalise the clusters to obtain the belief of variables within a cluster. For example, $\mathrm{Bel}\,(b) = \sum_c \mathrm{Bel}\,(\mathbf{Z} = (b, c))$.



(a) A multiply connected causal network

(b) Cluster tree

Figure 5.5: Simple clustering example

The problem is to cluster the nodes of the network so that the structure is optimised with no cycles and minimum size clusters. One popular approach uses *join trees* (Appendix B.2), in which the clusters are allowed to grow until a tree structure is formed. If a probabilistic model is decomposable with respect to a chordal graph, then the cliques of the graph can be arranged in a tree that is an I-map of the probabilistic model. This leads to the following procedure for forming the clusters of a Bayesian belief network:

1. Form a Markov network G by connecting all the parent nodes of the BBN that share a common child node and removing the arrows from all the arcs. Such a Markov network is illustrated in Figure 5.6(b);

2. Triangulate G to form a chordal super graph G'. Figure 5.6(b) is already chordal;

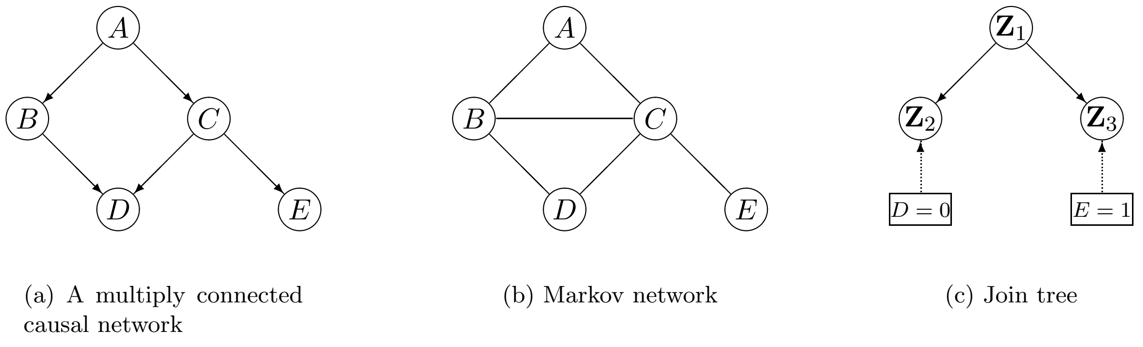
3. Identify the cliques in G' as compound variables and connect them to form a join tree T. The maximal cliques in Figure 5.6(c) are $\mathbf{Z}_1 = \{A, B, C\}$, $\mathbf{Z}_2 = \{B, C, D\}$ and $\mathbf{Z}_3 = \{C, E\}$, that is, each subset of nodes for which all nodes are adjacent;

4. The evidence nodes in the BBN are added as dummy variables to T. The $\boldsymbol{\lambda}$ messages are sent to one of the clique nodes that contains the evidence node. Figure 5.6(c) shows one such configuration;

5. Generate the link matrices for each arc in T so that evidence can be propagated throughout T and projected back to the individual variables of the BBN. The con-

ditioning is performed on all variables that are common to both the clusters joined by the arc. For example, in Figure 5.6(c),

$$\mathrm{p}\left(\mathbf{z}_2 \mid \mathbf{z}_1\right) = \mathrm{p}\left(b, c, d \mid b, c\right) = \mathrm{p}\left(d \mid b, c\right)$$
$$\mathrm{p}\left(\mathbf{z}_3 \mid \mathbf{z}_1\right) = \mathrm{p}\left(c, e \mid c\right) = \mathrm{p}\left(e \mid c\right).$$



(a) A multiply connected causal network

(b) Markov network

(c) Join tree

Figure 5.6: Clustering using join trees

Although the cardinality of **Z** has increased and the matrices on its arcs contain additional rows or columns, these matrices do not have to be stored on the arcs; they can be generated from the original link matrices. Also, although the propagated messages will have greater dimensionality in the join tree network, the computational complexity is not as great as would be expected because of the variables that are common to more than one cluster. Most computations within clusters are still performed using these common variables as mediators.

The difficult part of constructing a join tree is the triangulation step. A graph may be triangulated using node elimination; nodes are successively eliminated by adding arcs or *fill-ins* such that their neighbours are all adjacent, and then removing the node and its arcs. The original graph is triangulated by adding the fill-ins produced by the node elimination process. Although eliminating nodes in any order will produce a triangulated graph, the size of the tables (the product of the number of states of the variables) in each cluster will affect propagation performance. A good triangulation will yield small cliques. Although determining the optimal triangulation is *NP*–complete, a heuristic algorithm that uses a greedy approach gives reasonable results. At each iteration of this algorithm, the node selected for elimination must not require fill-ins or, failing that, must have the smallest table.

A variation of the join tree method uses *separator* labels on each arc. These separators consist of the variables that are common to the connected variables and, like the variables, the separators hold tables over all configurations of their variables. This approach has been adopted in the *hugin* system [Jensen 1996].

5.3.2 Conditioning

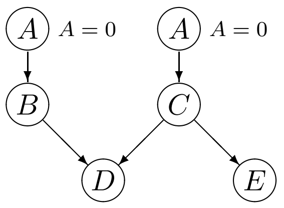
Conditioning, or *reasoning by assumptions*, works by instantiating a group of variables selected to break the loops. This is similar to human reasoning where, for complex problems, diametrically opposed assumptions are made and the final result is some weighted average of the individual results [Pearl 1988].

By way of an example, consider the network in Figure 5.6(a). As illustrated in Figure 5.7, this multiply connected network may be converted to a singly connected network by instantiating A, duplicating the network for each possible value of A, say $\{0, 1\}$, and integrating over the duplicates to get the final result. This network could also have been conditioned on C, but not on D, as instantiation does not block a converging node (Section 3.1). The belief in node B, given evidence at E, can be determined from

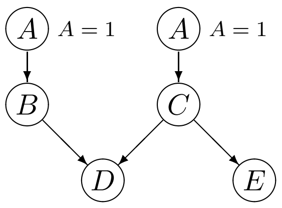
$$\mathrm{Bel}(b) = \mathtt{p}(b \mid e) = \sum_a \mathtt{p}(b \mid a, e)\, \mathtt{p}(a \mid e)\,. \tag{5.15}$$

The first term of the summation is the belief at B, obtained by propagating the influence of the evidence at E and the instantiated value of A through the network. The second is the mixing weight used in integrating the results for each value of A. This mixing weight can be calculated using Bayes' rule, that is

$$\mathtt{p}(a \mid e) = \alpha\, \mathtt{p}(e \mid a)\, \mathtt{p}(a)\,. \tag{5.16}$$



(a) $A = 0$



(b) $A = 1$

Figure 5.7: Using conditioning to remove loops

In general, for multiple evidence nodes $E_1, E_2, \ldots$, the mixing weight can be calculated recursively by considering each evidence node in turn. This may be done using

$$\mathtt{p}(a \mid e_1, \ldots, e_{i-1}, e_i) = \alpha\, \mathtt{p}(e_i \mid a, e_1, \ldots, e_{i-1})\, \mathtt{p}(a \mid e_1, \ldots, e_{i-1})\,. \tag{5.17}$$

The first term on the right represents the current belief distribution for E_i, and is available immediately on instantiation. The second is the previous mixing weight.

Obviously, this approach is expensive in terms of computational complexity, particularly if the network contains many loops and the conditioning variables have many states.

5.3.3 Stochastic Simulation

Stochastic simulation is a method of computing the probabilities by measuring how frequently specific events occur during a series of simulation runs. Causal models of a domain are used to produce random samples of hypothetical scenarios that are likely to occur in that domain [Pearl 1988].

Consider the example in Figure 5.6(a). Assuming the evidence nodes D and E have been instantiated with values d and e, respectively, the objective is to determine the most likely propositions at each of the hypothesis nodes A, B and C. Starting with an arbitrary configuration for the states of A, B and C, each of these variables are processed as follows.

1. Calculate the posterior probability distribution of the variable given the assumed states of the other variables in the network.

2. Draw a random sample from this distribution and set the variable equal to that value.

The new values for the set of all variables represent a single sample. This process is then repeated for all variables to obtain further samples. Once enough samples have been collected, the posterior probability of, say, A may be determined by calculating the percentage of times A was in each state, or by calculating the average posterior probability distribution of A from the sample distributions. Feller's Theorem guarantees convergence to the true posterior distributions, provided all the arc probabilities are positive [Pearl 1988, p. 216].[9]

Forward sampling is a simple approach in which each variable is processed in order from the top of the causal network. This method ignores the evidence until the appropriate node is reached, that is, it does not guarantee that instantiated variables will draw their instantiated values. If an instantiated variable fails to draw its instantiated value, the simulation run must be discarded. As the joint probability for the instantiated set of variables is often small, many runs must be discarded. For example, Jensen [1996] quotes in excess of 35000 runs to obtain 100 samples when the joint probability for two instantiated variables is 0.00282. An alternative is to clamp the evidence variables to their observed values. One algorithm that adopts this approach is *Gibbs sampling.* This algorithm commences with a configuration constrained to the observed data, perhaps generated by forward sampling, and then randomly changes the values of the uninstantiated variables in causal order. Once through the network, the sample is complete, and this configuration is used as the starting configuration for the next pass.

5.3.4 The Noisy-Or-Gate

The noisy-OR-gate is an approximation technique that may be employed to model the disjunctive interaction between binary variables, that is, variables with only two possible states.[10] It is used when any one of a set of conditions (parent variables) is likely to cause a certain event, and this likelihood is not reduced when other conditions in this set are activated. Each cause is considered separately, and the results are then combined using the logical *or* function.

[9]The presence of zero probabilities corresponds to *reducible* Markov chains, for which this technique has limited applicability.

[10]The dual noisy-AND-gate may be used to model conjunctive interaction between binary variables.

The noisy-OR-gate is based on the assumptions of *accountability* and *exception independence*. The accountability assumption specifies that an event is false if all of its causes are also false. The exception independence assumption implies that the inhibiting mechanism associated with one cause is independent of those associated with other causes.

Figure 5.8 illustrates the noisy-OR-gate model. Nodes W_i, $i = 1, 2, \ldots, n$ are the parent nodes of the event X, that is, they are the conditions that trigger the consequence X. All these nodes appear in the underlying Bayesian belief network. Nodes I_i are the inhibiting mechanisms for each cause; these do not appear explicitly in the Bayesian belief network, rather they are embedded in the link probabilities $\mathtt{p}\,(x \mid w_1, w_2, \ldots, w_n)$. This is a special case of the causal polytree solution, in which X and its parents $\mathbf{W}$ are binary variables.

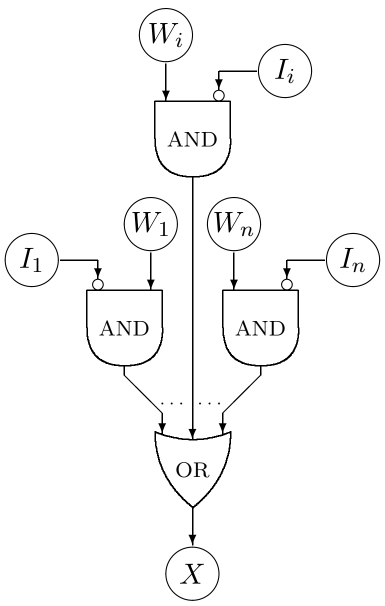


Figure 5.8: Modelling disjunctive interactions using the noisy-OR-gate

The belief in X, obtained from (5.2) and (5.14), is written as

$$\mathrm{Bel}\,(x) = \alpha\,\lambda(x) \sum_{\mathbf{w}} \mathtt{p}\,(x \mid \mathbf{w}) \prod_i \pi_X(w_i). \tag{5.18}$$

Following Pearl [1988, pp. 185–190], this can be written

$$\mathrm{Bel}\,(x) = \begin{cases} \alpha\,\lambda(X=0) \prod\limits_i (1 - (1-q_i)\pi_{iX}) & \text{for } x = 0 \\ \alpha\,\lambda(X=1) \left(1 - \prod\limits_i (1 - (1-q_i)\pi_{iX})\right) & \text{for } x = 1, \end{cases} \tag{5.19}$$

where q_i denotes that probability that the i^{th} inhibitor is active and

$$\pi_{iX} = \pi_X(W_i = 1) = 1 - \pi_X(W_i = 0), \tag{5.20}$$

that is, the predictive support for $W_i = 1$ sent from W_i to X. $(1 - q_i)$ is the probability that the output of the i^{th} AND gate in Figure 5.8 is true, and $1-(1-q_i)$ is the probability that it is false.

The $\boldsymbol{\lambda}$s are propagated to the parent nodes as follows:

$$\boldsymbol{\lambda}_X(W_i) = \beta \sum_x \boldsymbol{\lambda}(X) \sum_{\mathbf{w}_k : k \neq i} \mathtt{p}\left(x \mid w_1, w_2, \ldots, w_n\right) \prod_{k=1\backslash i}^{n} \boldsymbol{\pi}_X(W_k) \tag{5.21}$$

$$= \begin{cases} \beta\, \lambda(X=0)\, q_i^{w_i} \prod_{k=1\backslash i}^{n} (1-(1-q_k)\pi_{kX}) & \text{for } x=0,\ w_i = 0,1 \\ \beta\, \lambda(X=1) \left(1 - q_i^{w_i} \prod_{k=1\backslash i}^{n} (1-(1-q_k)\pi_{kX})\right) & \text{for } x=1,\ w_i=0,1. \end{cases} \tag{5.22}$$

If X does not receive any diagnostic evidence, then W_i will not receive any from X. Furthermore, W_i will not be affected by any other parents of X, namely W_j, $j \neq i$; this is consistent with d-separation and independence at converging arcs. Additional decoupling is provided by the noisy-OR-gate; when X is completely denied, that is, it is instantiated to zero, then any $\boldsymbol{\pi}$ messages are absorbed by X, and no $\boldsymbol{\lambda}$ messages are passed to $\mathbf{W}$. This is similar behaviour to that of anticipatory variables. Also, if one parent lends full support to $X = 1$, then no other parent receives support from X.

6 Building Networks

The designer of a Bayesian belief network must determine the most appropriate network structure and probabilities for his or her application. Both of these issues are discussed in this section, and some practical hints on the design of Bayesian belief networks are provided.

Designing a Bayesian belief network involves the following initial steps [Heckerman 1996]:

1. Identify the goals of the model;
2. Identify all possible sources of data that may be relevant to achieving these goals;
3. Select the data that are meaningful and worthwhile in the context of the model;
4. Organise the selected data into variables that have mutually exclusive and collectively exhaustive states.

For example, the goal of a model may be to determine a target's identity and class, that is, to which force it belongs and the type of platform. Possible sources of identity information may be Electronic Support (ES) reports, IFF responses, Forward Looking Infrared (FLIR) attribute data, track position and kinematic behaviour. However, the operational range of the FLIR may make it unsuitable at the ranges of most interest and track location may not provide any identity or class discrimination. Therefore, only the ES, IFF and kinematic behaviour data will be inputs to the model. The ES data could be represented by the variable *ES* with states corresponding to each type of platform and emitter known to the ES receiver plus an *unknown* state. The IFF data could be represented by the variable *IFF* with states *friend*, *neutral* and *no response*. The kinematic behaviour could be represented by two variables, *Vel*, with states *slow*, *moderate* and *fast*, and *Man*, with states *no manoeuvre*, *low-g manoeuvre* and *high-g manoeuvre*. Having made the decisions above, it is then possible to develop an appropriate structure to meet the goals using the selected data. Once the structure has been determined, the model is completed by choosing the probabilities.

6.1 Choosing the Structure

When using Bayesian belief networks, we are interested in obtaining estimates of the certainties for events that are not observable, or are observable at an unacceptable cost. These certainty estimates allow us to hypothesise the occurrence of events that are of particular interest to us. These are referred to as *hypothesis events* and correspond to the hypothesis nodes of the network. In the target identification example, the hypothesis events may be *ID*, with states *friend*, *assumed friend*, *neutral*, *suspect* and *hostile* and *Cls* with the possible platform types as its states. The hypothesis nodes of a Bayesian belief network should be selected to reflect the goals of the model.

Other available information may offer insight into the certainty of various hypothesis events, and this information may be incorporated into the network through the use of *information variables* that are represented by evidence nodes in the network. It is necessary to identify the types of information that may reveal something about a hypothesis variable and represent these sources of information as variables. Usually this information is of the type that may be expressed as *information variable X is in state x*, although softer statements are allowed, for example, *variable X is in state x with probability y*. In the target identification example, the information variables are *ES*, *IFF*, *Man* and *Vel*.

Having selected the variables for the network, it is then necessary to determine the causal structure between the variables, in particular, which variable causes a particular state in another variable. Generally, people are able to readily recognise causal relationships between variables, and these relationships typically correspond to conditional dependence between the variables [Heckerman 1996]. When determining this causality, it is important that the appropriate conditional probabilities and dependencies be established between the information variables and the hypothesis variables. This is done by positioning *intermediate* or mediating nodes between the information (evidence) and hypothesis nodes. This simplifies the determination of conditional probabilities and, in particular, allows the dependency of two variables that is not manifested through other variables to be introduced through a mediating variable that is a parent of both. Returning to the target identification example, it is obvious that *IFF* is caused by *ID*, while *ES*, *Man* and *Vel* are caused by *Cls*. However, although there is a relationship between *ID* and *Cls*, it is ambiguous as to which causes which. This problem can be readily solved by introducing a mediating variable representing the nationality (*Nat*) with states representing each nation with a defence force. Both the type of platform and its identity are influenced by the country of its origin, giving the structure shown in Figure 6.1.

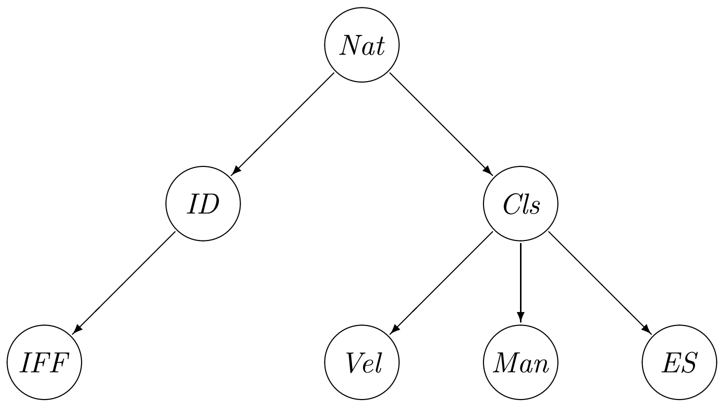


Figure 6.1: A simple target identification and classification network

6.2 The Probabilities

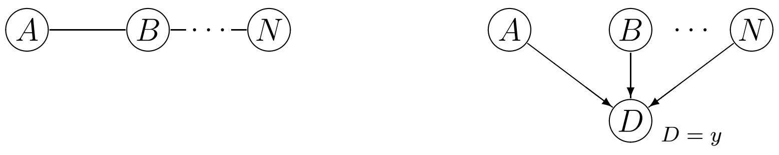
The conditional and prior probabilities in a Bayesian network may be Bayesian or physical. Bayesian probabilities are derived from prior knowledge only, whereas physical probabilities are learnt from available data [Heckerman 1996]. The process of determining the degree of belief or probability of each state of a variable is referred to as *probability assessment*. A major concern for both Bayesian and physical probability assessment is that of precision, although the decisions derived from these networks are not usually sensitive to small variations in the probabilities. The conditional probabilities may be based on statistical and empirical frequencies, subjective estimates or a combination of these. Where possible, it is preferable to base the probabilities on a theoretical basis or on actual measured data. However, in many cases, only subjective estimates may be available. It is best to use as many sources as possible, for example, experimental data and expert opinions. The prior probabilities of the variables without parents must also be determined. In the target identification example, it would be sensible to base the prior probabilities for *Nat* on the relative sizes of each defence force. However, prior information such as this may not always be available. In these cases, the prior probabilities should be chosen so that unwanted biases are minimised. For example, a diffuse prior may be selected for a continuous variable and an equi-probable condition for all states of a discrete variable may be employed.

6.3 Other Design Issues

In practice, the design of Bayesian belief networks may be complicated by difficulties in determining the dependencies and independencies, and computational constraints may impose restrictions on the design. Practical solutions and alternatives have been developed to overcome many of these problems, a few of which are presented here to assist in the design of practical Bayesian belief networks.

6.3.1 Undirected Relations

In some cases, it may be impossible to determine the direction of the arcs, that is, two or more events are dependent but neither is caused by the other, and it may not be possible or practical to identify a common cause. This may lead to networks containing a mixture of directed and undirected arcs or relations, giving rise to what are known as *chain graphs*. These undirected arcs may be removed from the network and the previously connected variables made parents of a new binary variable having states *yes* (y) and *no* (n) but instantiated with y, as illustrated in figure 6.2. The provision of evidence at D makes the other nodes dependent (Section 3.1), as was the case with the undirected arcs.

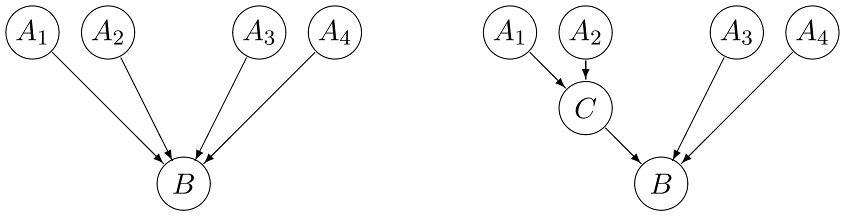


(a) Undirected arcs (b) DAG equivalent

Figure 6.2: Dealing with undirected relations

6.3.2 Divorcing

To reduce the number of combinations of states for a variable with multiple parents, and hence the number of probabilities to be calculated, the *divorcing* technique may be employed. Some of the parents of a variable are removed or divorced from that variable by introducing a mediating variable and making it a child of the divorced parents and a parent of the original child. For example, in figure 6.3, the variable C is introduced to divorce the parents A_1 and A_2 of B. For divorcing to be effective, the number of states of C must be less than the number of combinations of states of A_1 and A_2.



(a) Original (b) After divorcing

Figure 6.3: The use of divorcing to reduce complexity

6.3.3 Noisy Or

It may be difficult to determine the conditional probabilities of a variable with multiple parents. For example, a binary variable may be in the state y if any of its binary parents are also in state y. This situation is readily handled by the noisy-or approach, as discussed in Section 5.3.4.

6.3.4 Causal Independence

Under certain circumstances, it may appear that the causes of a variable A act independently, although independence in this context may not be well defined. In these cases, a change in the state of one parent may cause a change in the state of A, but this change in state is only dependent on the previous state of A, the new state of the cause and the previous state of the cause. The states of the other causes that have not changed do not influence the change of state of the variable.

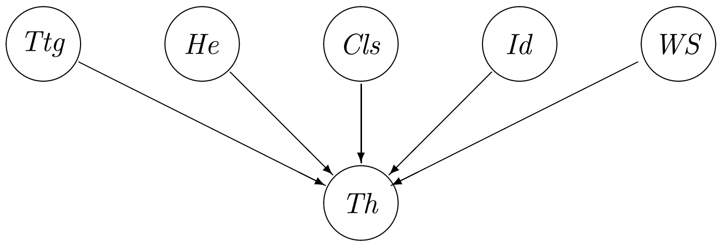


Figure 6.4: A simple threat evaluation network

To illustrate this, consider the threat analysis example in Figure 6.4, where the threat status *Th*, with ordered states *no*, *low*, *medium* and *high*, is caused by the time for the threat to reach a defended asset (time-to-go) *Ttg*, and the weapons state *WS*, heading *He*, class *Cls* and identity *Id* of the threat. If *Th* is in state medium, then an *Id* change from suspect to hostile may cause an elevation of the threat state, in this case from medium to high. The other causes have not changed and do not contribute to this change of state in *Th*, it being only dependent on the previous state of *Th* and the previous and new state of *Id*. This may be modelled by a chain of *Th* related variables, each corresponding to a single cause, as shown in Figure 6.5. In this way, the state of Th is dependent on its previous value.

7 Target Recognition Application

The target identification or recognition problem may be generally interpreted to include the determination of target category (air, surface, sub-surface, land, etc.), target identity (friend, neutral, hostile, etc.) and target class (platform type). Supporting and contradicting evidence for any category, identity or class hypothesis may be obtained from

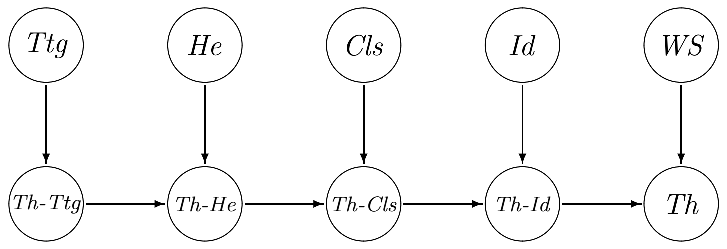


Figure 6.5: A simple threat evaluation network using causal independence

a variety of sources, including IFF reports, ES data, track dynamics, track location, Non-Cooperative Target Recognition (NCTR), radar cross-section, visual identification and intelligence reports. In some cases, the evidence may directly confirm a hypothesis with some level of certainty, for example, a friendly *mode 4* IFF response, or an unambiguous ES platform declaration. In other cases, the data may indirectly contribute, for example, no response from an IFF interrogation or a track following an air-lane.

Target recognition is improved by considering all available sources of data [Stewart & McCarty 1992]. Two approaches that account for uncertainty and are able to accommodate the disparate types of potential data are Bayesian techniques, including Bayesian belief networks [Hautaniemi, Korpisaari & Saarinen 2000] [Korpisaari & Saarinen 1999] [Stewart & McCarty 1992] and Dempster-Shafer [Bogler 1987] [Simard, Couture & Bossé 1996] [Valin, Couture & Simard 1996], with debate over which approach is superior [Braun 2000] [Leung & Wu 2000].

The ability of Bayesian belief networks to contain continuous hypothesis and evidence variables to represent kinematic information and discrete variables to represent category, identity and class information make them a suitable candidate for target tracking and recognition applications. Their ability to model the influence of disparate types of evidence on hypothesis variables through the network arcs and intermediate variables is particularly advantageous for target recognition. Bayesian belief networks may be used to propagate the relevant hypotheses through time, and evidence from sensors, remote data sources and information obtained from the track state may be used to update the likelihood of each hypothesis. Prior information, such as the numbers of each type of platform in each force, is also considered in this approach. As the probability of each hypothesis is provided by the network, some method of determining the actual category, identity and class must be implemented. This may simply be the one with the highest probability, or the decision may be deferred until one hypothesis becomes dominant.

8 Summary

Probabilistic models are a convenient tool for modelling uncertain expert knowledge. These models consist of a set of interrelated variables, where each may take any one of a number of possible states with some probability. Probability languages provide the framework for presenting and processing the uncertainty relationships between the variables, effectively addressing the issues of likelihood, conditioning, relevance and causation. Probabilistic models that capture the relationships between the variables are known as dependency models.

Directed acyclic graphs, known as causal belief networks, are convenient tools for representing dependency models. These networks consist of a set of nodes, representing the set of variables in the model, connected by a set of arcs representing the causal relationships between these variables. Independence is handled by the concept of d-separation; the flow of information is blocked by instantiating variables, except at converging connections, where the presence of evidence enables the flow of information. Bayesian belief networks are a particular class of causal networks that use Bayesian calculus for updating the node probabilities.

Bayesian belief networks contain hypothesis nodes, representing alternate hypotheses, information or evidence nodes that provide the mechanism for entering evidence, and intermediate nodes that establish the correct dependence relationships between the hypothesis and information nodes. The belief or probability that a particular node is in a particular state is determined by propagating the effect of all the evidence throughout the entire network. The belief update for each variable is local to its immediate neighbours, that is, the belief in each of its possible states is determined only by the belief in the states of the nodes to which it is directly connected. This simplifies the processing, particularly if appropriate tree structures are used. Techniques exist for converting general network structures into tree topologies, and practical strategies may be employed to further reduce the processing load. Various methods are available to determine appropriate network structures and probabilities from the available data.

Bayesian belief networks are particularly suited to the target recognition problem. Their ability to incorporate the influence of disparate types of evidence on the hypotheses for track category, identity and class, and their capacity to accommodate the uncertainty in this evidence, makes them suitable candidates for solving this problem.

References

Bogler, P. L. (1987) Shafer-Dempster reasoning with applications to multisensor target identification systems, *IEEE Trans. Systems, Man, and Cybernetics* **SMC-17**(6), 968–77.

Braun, J. J. (2000) Dempster-Shafer theory and Bayesian reasoning in multisensor data fusion, *in Sensor Fusion: Architectures, Algorithms, and Applications IV*, Vol. 4051 of *Proceedings of the SPIE*, Orlando, FL, pp. 255–66.

Hautaniemi, S. K., Korpisaari, P. T. & Saarinen, J. P. P. (2000) Target identification with Bayesian networks, *in Sensor Fusion: Architectures, Algorithms, and Applications IV*, Vol. 4051 of *Proceedings of the SPIE*, Orlando, FL, pp. 55–66.

Heckerman, D. (1996) *A Tutorial on Learning With Bayesian Networks*, Technical Report MSR-TR-95-06, Microsoft Corporation, Redmond, WA.

Jensen, F. V. (1996) *An Introduction to Bayesian Networks*, UCL Press, London UK.

Korpisaari, P. & Saarinen, J. (1999) Bayesian networks for target identification and attribute fusion with JPDA, *in 2^{nd} Int. Conf. on Information Fusion*, Vol. II, Int. Soc. of Information Fusion, Sunnyvale CA, pp. 763–9.

Leung, H. & Wu, J. (2000) Bayesian and Dempster-Shafer target identification for radar surveillance, *IEEE Trans. Aerospace and Electronic Systems* **36**(2), 432–47.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco CA.

Simard, M.-A., Couture, J. & Bossé, É. (1996) Data fusion of multiple sensors attribute information for target identity estimation using a Dempster-Shafer evidential combination algorithm, *in Signal and data processing of small targets 1996*, Vol. 2759 of *Proceedings of the SPIE*, Orlando, FL, pp. 577–88.

Stewart, L. & McCarty, Jr., P. (1992) The use of Bayesian belief networks to fuse continuous and discrete information for target recognition, tracking and situation assessment, *in Signal Processing, Sensor Fusion and Target Recognition*, Vol. 1699 of *Proceedings of the SPIE*, pp. 177–85.

Suermondt, H. J. (1992) *Explanation in Bayesian belief networks*, PhD thesis, Stanford University, Stanford, CA.

Valin, P., Couture, J. & Simard, M.-A. (1996) Position and attribute fusion of radar, ESM, IFF and datalink for AAW missions of the Canadian patrol frigate, *in Proceedings of MFI'96, 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Washington, DC, pp. 63–71.

Appendix A: Axioms for Dependency Models

The following axioms or conditions are relevant to dependency models and their representative graphs [Pearl 1988]. In the following descriptions, **W**, **X**, **Y** and **Z** are disjoint subsets of variables in the universe **U**, and A, B, C and D represent single variables in **U**.

Symmetry: If **Y** provides no new information about **X**, then **X** provides no additional information about **Y**.

$$\mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_p \Leftrightarrow \mathtt{I}\,(\mathbf{Y}, \mathbf{Z}, \mathbf{X})_p \tag{A1}$$

In graphical terms, if **Z** separates **X** from **Y**, then it also separates **Y** from **X**.

Composition: If two sets of variables are individually judged to be irrelevant to **X**, then their combination is also irrelevant to **X**.

$$\mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_p \ \&\ \mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{W})_p \Rightarrow \mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})_p \tag{A2}$$

Graphically, as illustrated in Figure A1, if **Z** separates **X** from both **W** and **Y**, then it also separates **X** from the union of **W** and **Y**.

Decomposition: If a set **S** is judged irrelevant to **X**, then each subset of **S** is also irrelevant to **X**.

$$\mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})_p \Rightarrow \mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_p \ \&\ \mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{W})_p \tag{A3}$$

Graphically, this means that if **Z** separates **X** from a set **S** ($\mathbf{S} = \{\mathbf{Y}, \mathbf{W}\}$ in Figure A2), then it also separates **X** from each subset of **S** (**W** and **Y** in Figure A2).

Strong Union: If **Y** is irrelevant to **X**, then learning new data **W** cannot help **Y** become relevant to **X**.

$$\mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{Y})_p \Rightarrow \mathtt{I}\,(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y})_p \tag{A4}$$

As shown in Figure A3, if **Z** separates **X** and **Y**, then **X** and **Y** remain separated when **Z** is augmented with **W**.

Weak Union: If **Y** is irrelevant to **X**, then learning new data **W** that is also irrelevant to **X** cannot help **Y** become relevant to **X**.

$$\mathtt{I}\,(\mathbf{X}, \mathbf{Z}, \mathbf{Y} \cup \mathbf{W})_p \Rightarrow \mathtt{I}\,(\mathbf{X}, \mathbf{Z} \cup \mathbf{W}, \mathbf{Y})_p \tag{A5}$$

Graphically, if **Z** separates **X** and **Y**, then **X** and **Y** remain separated when **Z** is augmented with **W**, provided that **W** is separated from **X** by **Z** (see Figure A4).

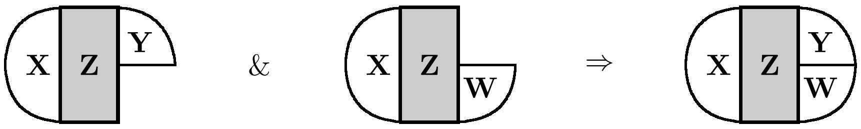


Figure A1: Composition condition

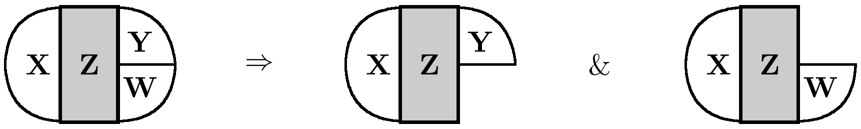


Figure A2: Decomposition condition

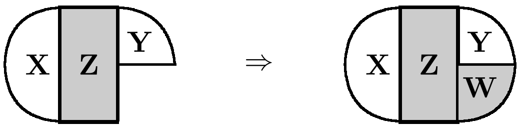


Figure A3: Strong union condition

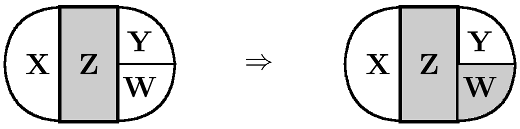


Figure A4: Weak union condition

Contraction: If **W** is judged to be irrelevant to **X** after learning information **Y** that is also irrelevant to **X**, then **W** must have been irrelevant prior to learning **Y**.

$$\mathtt{I}\left(\mathbf{X},\mathbf{Z},\mathbf{Y}\right)_p \ \&\ \mathtt{I}\left(\mathbf{X},\mathbf{Z}\cup\mathbf{Y},\mathbf{W}\right)_p \Rightarrow \mathtt{I}\left(\mathbf{X},\mathbf{Z},\mathbf{Y}\cup\mathbf{W}\right)_p \tag{A6}$$

As illustrated in Figure A5, the deletion of part of the separator set is permitted, provided that the remaining separator set separates the deleted portion from **X**.

Transitivity: If **X** and **Y** are irrelevant to each other, then each individual element or variable in the universe **U** is irrelevant to either **X** or **Y**, but not both.

$$\mathtt{I}\left(\mathbf{X},\mathbf{Z},\mathbf{Y}\right)_p \Rightarrow \mathtt{I}\left(\mathbf{X},\mathbf{Z},A\right)_p \quad \text{or} \quad \mathtt{I}\left(A,\mathbf{Z},\mathbf{Y}\right)_p \tag{A7}$$

Graphically, as shown in Figure A6, if **Z** separates **X** and **Y**, then every element in **U** is also separated from either **X** or **Y**.

Weak transitivity: If **X** and **Y** are irrelevant to each other after learning information on some variable of **U**, then that variable is irrelevant to either **X** or **Y**, but not both.

$$\mathtt{I}\left(\mathbf{X},\mathbf{Z},\mathbf{Y}\right)_p \ \&\ \mathtt{I}\left(\mathbf{X},\mathbf{Z}\cup A,\mathbf{Y}\right)_p \Rightarrow \mathtt{I}\left(\mathbf{X},\mathbf{Z},A\right)_p \text{ or } \mathtt{I}\left(A,\mathbf{Z},\mathbf{Y}\right)_p \tag{A8}$$

As shown in Figure A7, if **Z** and some single variable in **U** separates **X** and **Y**, then that variable is also separated from either **X** or **Y**.

Intersection: If **Y** is irrelevant to **X** after learning **W**, and **W** is irrelevant to **X** after learning **Y**, then neither **Y**, **W** nor their combination is relevant to **X**.

$$\mathtt{I}\left(\mathbf{X},\mathbf{Z}\cup\mathbf{W},\mathbf{Y}\right)_p \ \&\ \mathtt{I}\left(\mathbf{X},\mathbf{Z}\cup\mathbf{Y},\mathbf{W}\right)_p \Rightarrow \mathtt{I}\left(\mathbf{X},\mathbf{Z},\mathbf{Y}\cup\mathbf{W}\right)_p \tag{A9}$$

As illustrated in Figure A8, if a subset **X** of **S** can be separated from the rest of **S** by two different subsets of **S**, then the intersection of those two subsets is sufficient to separate **X** from the rest of **S**.

Chordality: If two variables, A and B, of **U** are irrelevant when two other variables, C and D, are known, and C and D are irrelevant when A and B are known, then A and B are irrelevant if only C is known or only D is known.

$$\mathtt{I}\left(A,C\cup D,B\right)_p \ \&\ \mathtt{I}\left(C,A\cup B,D\right)_p \Rightarrow \mathtt{I}\left(A,C,B\right)_p \text{ or } \mathtt{I}\left(A,D,B\right)_p \tag{A10}$$

Graphically, this is illustrated in Figure A9 where, if C and D separate A and B and A and B separate C and D, A and B are separated by either C or D.

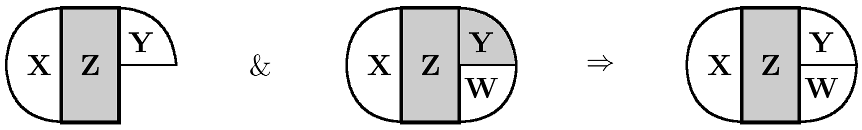


Figure A5: Contraction condition

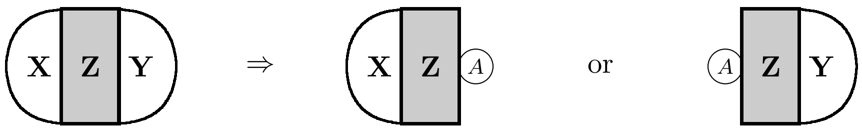


Figure A6: Transitivity condition

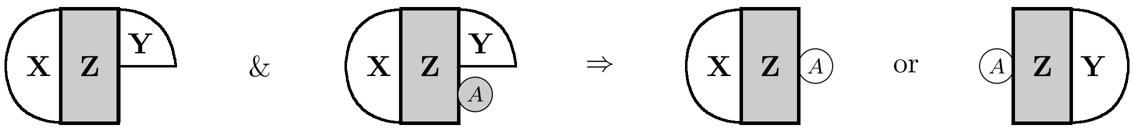


Figure A7: Weak transitivity condition

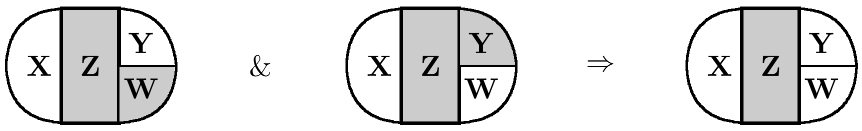


Figure A8: Intersection condition

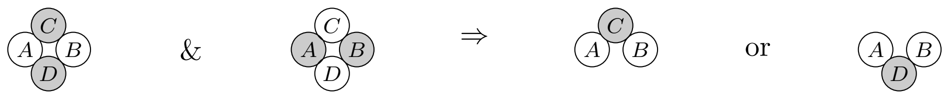


Figure A9: Chordality condition

These axioms may be used to derive rules for dependency models. Examples include the *chaining* and *mixing* rules, derived from the axioms of symmetry, decomposition, weak union, contraction and intersection [Pearl 1988, pp. 88–89].

Chaining: If knowing **Y** makes **X** and **Z** irrelevant to each other, and **X** and **Y** are irrelevant to **W** if **Z** is known, then **X** and **W** are irrelevant if **Y** is known.

$$\mathtt{I}\left(\mathbf{X},\mathbf{Y},\mathbf{Z}\right)_p \ \&\ \mathtt{I}\left(\mathbf{X}\cup\mathbf{Y},\mathbf{Z},\mathbf{W}\right)_p \Rightarrow \mathtt{I}\left(\mathbf{X},\mathbf{Y},\mathbf{W}\right)_p \tag{A11}$$

Graphically, if **Y** separates **X** and **Z** and **Z** separates **W** from **X** and **Y**, then **Y** separates **X** from **W** (see Figure A10).

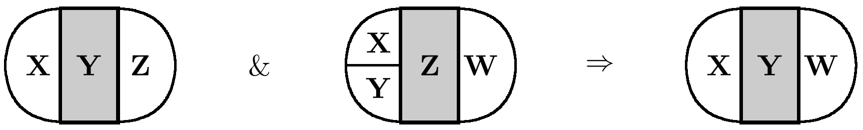


Figure A10: Chaining rule

Mixing: For each of **X**, **Y** and **W** to be independent of the other two, one of them must be independent of the other two, which must also be mutually independent.

$$\mathtt{I}\left(\mathbf{X},\mathbf{Z},\mathbf{Y}\cup\mathbf{W}\right)_p \ \&\ \mathtt{I}\left(\mathbf{Y},\mathbf{Z},\mathbf{W}\right)_p \Rightarrow \mathtt{I}\left(\mathbf{X}\cup\mathbf{W},\mathbf{Z},\mathbf{Y}\right)_p \tag{A12}$$

Figure A11 provides a graphical representation of this rule. If **Z** separates one set of variables from two other sets, and also separates those two sets, then **Z** also separates each set from the other two sets.

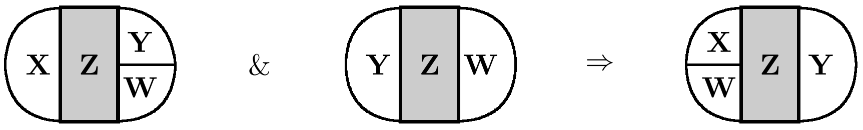


Figure A11: Mixing rule

Appendix B: Markov Networks

A *Markov Network* is an undirected graph that is a minimal I-map[11] of a dependency model. These networks are produced by connecting only the pairs of variables in a model that are not independent [Pearl 1988]. A dependency model M is a *graph isomorph* if it can be represented by an undirected graph. A necessary and sufficient condition for M to be a graph isomorph is for $\mathtt{I}\left(\mathbf{X}, \mathbf{Z}, \mathbf{Y}\right)_M$ to satisfy the axioms of symmetry, decomposition, intersection, strong union and transitivity, as defined in Appendix A.

In this type of network, each node is influenced by all those that are connected to it, either directly or indirectly through other nodes. The influence of connected nodes may be blocked by instantiating at least one node in every path connecting the two nodes. Therefore conditional probabilities in Markov networks are defined as follows:

> If the removal of a subset of nodes Z renders two nodes independent, then those two nodes are conditionally independent given Z.

In addition to representing the conditional independence properties of a model, a Markov network must also represent the strength of the interactions between variables through weights on each arc. These weights must be consistent and complete, that is, they must not lead to contradictory conclusions or prevent the reaching of conclusions through a lack of evidence. *Gibbs potential* can be used to construct a consistent and complete quantitative model without destroying the dependency structure of the graph. This method consists of the following four steps [Pearl 1988].

1. Identify the maximal cliques of the graph G.[12]
2. For each clique, assign a nonnegative function that quantifies the relative degree of compatibility between the assigned values for each variable in that clique.
3. Form the product of all the compatibility functions.
4. Normalise this product over all possible variable values to get a joint distribution that encompasses the conditional independencies in G.

The difficulty is in assigning meanings to the compatibility functions; a necessity if the model is to produce meaningful inferences. Often experts are only prepared to assess simple hypotheses that are isolated from the rest of the universe, that is, low order conditional probabilities. In general, Markov networks only allow these judgements to be taken as constraints on the joint probability distribution of the network, making it difficult to calculate the compatibility functions [Pearl 1988, p. 108]. However, some dependency Markov networks have compatibility functions that are directly related to the lower marginal probabilities on the variables in each clique of the network. The cliques in

[11]Minimal I-maps are discussed in Section 2.3.
[12]A maximal clique is the largest subgraph whose nodes are all adjacent or directly connected by arcs, that is, a maximal complete graph.

these *decomposable models* form a tree structure, allowing the joint probability distribution to be expressed as a product of conditional probabilities. A probability model P is decomposable if it has a minimal I-map that is chordal. It is decomposable relative to a graph G if G is an I-map of P and G is chordal. A graph is chordal if every cycle of four or more nodes has a chord, that is, an arc joining two non-consecutive nodes. The following triangulation algorithm may be used to determine if a graph is chordal, or to add arcs to a graph to make it chordal:

1. Use a *maximum cardinality search*[13] to assign an order (rank) to the nodes of the graph, that is, $1, 2, \ldots, |\mathbf{V}|$;

2. Starting at the node with the highest rank, recursively place arcs between any two non-adjacent neighbours of lower rank, including neighbours linked in previous steps;

3. If no arcs are added to the graph, it is chordal. Otherwise, the new graph is chordal.

Two examples of decomposable models, namely *Markov trees* and *join trees*, are discussed in the following sections.

B.1 Markov Trees

A Markov tree is a Markov network that has a tree structure, therefore it does not contain any cycles. A Markov tree also has a minimal I-map and is chordal, making it decomposable. Its product form distribution may be obtained using either the *directed trees* method or the *product division* method [Pearl 1988]. These are described by way of the following undirected tree example of seven variables.

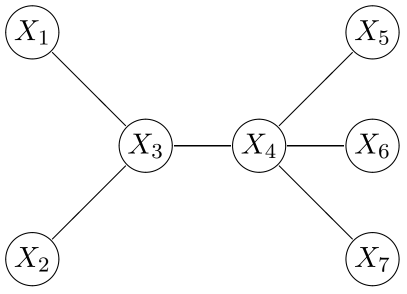


Figure B1: Undirected tree of seven variables

Using the directed trees method, the network in Figure B1 is converted to a directed tree by designating a node as the *root* and assigning arrows to the arcs such that they point away from the root. The directed tree resulting from placing the root at X_1 is shown

[13]A maximum cardinality search numbers the nodes sequentially, always assigning the next number to the node with the largest set of previously numbered neighbours.

in Figure B2. Every node in the directed tree, except the root, has a unique parent. By inspection, the product form of the distribution may be written

$$\begin{aligned}&\mathrm{p}\left(x_1, x_2, \ldots, x_7\right) = \\ &\qquad \mathrm{p}\left(x_1\right) \mathrm{p}\left(x_3 \mid x_1\right) \mathrm{p}\left(x_2 \mid x_3\right) \mathrm{p}\left(x_4 \mid x_3\right) \mathrm{p}\left(x_5 \mid x_4\right) \mathrm{p}\left(x_6 \mid x_4\right) \mathrm{p}\left(x_7 \mid x_4\right). \quad \text{(B1)}\end{aligned}$$

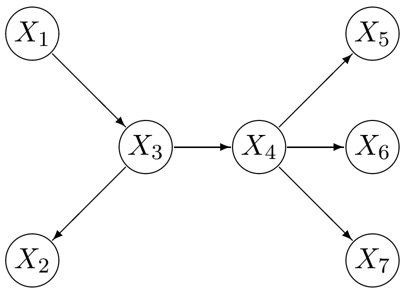


Figure B2: Directed tree with root X_1

The product division method involves dividing the product of the marginal distributions of the edges by the product of the distributions of the intermediate nodes. Therefore, the joint distribution for the example of Figure B1 becomes

$$\begin{aligned}\mathrm{p}\left(x_1, x_2, \ldots, x_7\right) &= \frac{\mathrm{p}\left(x_1, x_3\right) \mathrm{p}\left(x_2, x_3\right) \mathrm{p}\left(x_3, x_4\right) \mathrm{p}\left(x_4, x_5\right) \mathrm{p}\left(x_4, x_6\right) \mathrm{p}\left(x_4, x_7\right)}{\mathrm{p}\left(x_3\right) \mathrm{p}\left(x_3\right) \mathrm{p}\left(x_4\right) \mathrm{p}\left(x_4\right) \mathrm{p}\left(x_4\right)} \\ &= \mathrm{p}\left(x_3\right) \mathrm{p}\left(x_1 \mid x_3\right) \mathrm{p}\left(x_2 \mid x_3\right) \mathrm{p}\left(x_4 \mid x_3\right) \mathrm{p}\left(x_5 \mid x_4\right) \mathrm{p}\left(x_6 \mid x_4\right) \mathrm{p}\left(x_7 \mid x_4\right) \quad \text{(B2)}\end{aligned}$$

which is the same as would have been obtained using the directed trees method with X_3 as the root.

B.2 Join Trees

A join tree of a graph G is a tree structure in which the nodes are the cliques of G. The product form of the joint distribution may be found using the product division method, where the product of the distributions of the cliques is divided by the product of the distributions of the intersections of the cliques. This is a generalised version of that applied to the Markov tree, in which the cliques consist of two connected nodes, and the intersections are single nodes [Pearl 1988, p. 111].

For the joint probability distribution of a model to be expressed in its product form, the model must be decomposable relative to a graph G, and that graph must be chordal. As it is only necessary for G to be an I-map of P, a triangulation algorithm may be used to make G chordal, if it is not already so. The following procedure may be employed to produce a join tree from a probability model [Pearl 1988, p. 113]:

1. Use a triangulation algorithm to create a chordal graph G';

2. Identify all the cliques in G';
3. Order the cliques $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n$ by rank of the highest node in each;
4. Join each clique $\mathbf{C}_i$ to the predecessor $\mathbf{C}_j$, $j < i$, sharing the largest number of nodes with $\mathbf{C}_i$.

Appendix C: Belief Table Algebra

The following algebraic notation is defined for performing algebra on the belief tables representing the conditional probabilities of causal belief network variables [Jensen 1996, Sec. 4.1].

C.1 Multiplication and Division

If two tables over the same variables, $x, y, \ldots$, are denoted $f(x, y, \ldots)$ and $g(x, y, \ldots)$, then the product $f(x, y, \ldots)g(x, y, \ldots)$ denotes the element–by–element multiplication of the tables. For example,

$$f(x,y)g(x,y) = \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline y_1 & f_1 & f_2 \\ y_2 & f_3 & f_4 \end{array}\right] \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline y_1 & g_1 & g_2 \\ y_2 & g_3 & g_4 \end{array}\right] = \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline y_1 & f_1g_1 & f_2g_2 \\ y_2 & f_3g_3 & f_4g_4 \end{array}\right]. \tag{C1}$$

Element–by–element multiplication may also be performed over tables with different variables. In this case, the multiplication is over the common variables. For example,

$$f(x,y)g(x,z) = \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline y_1 & f_1 & f_2 \\ y_2 & f_3 & f_4 \end{array}\right] \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline z_1 & g_1 & g_2 \\ z_2 & g_3 & g_4 \end{array}\right] \tag{C2}$$

$$= \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline y_1 & (f_1g_1, f_1g_3) & (f_2g_2, f_2g_4) \\ y_2 & (f_3g_1, f_3g_3) & (f_4g_2, f_4g_4) \end{array}\right], \tag{C3}$$

where (a, b) denote the values for z_1 and z_2, respectively.

Division is performed in the same way. However, if the denominator table contains zero entries, then the numerator table must contain zeros in the same locations, with the resulting table containing ones in those locations.

Inner, or dot, products are denoted $f(x, y) \cdot g(x, z)$. In this case, the summation will always be taken over the repeated indices, that is,

$$f(x,y) \cdot g(x,z) \triangleq \sum_x f(x,y)g(x,y). \tag{C4}$$

For example,

$$f(x,y) \cdot g(x,z) = \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline y_1 & f_1 & f_2 \\ y_2 & f_3 & f_4 \end{array}\right] \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline z_1 & g_1 & g_2 \\ z_2 & g_3 & g_4 \end{array}\right] \tag{C5}$$

$$= \left[\begin{array}{c|cc} & y_1 & y_2 \\ \hline z_1 & f_1g_1 + f_2g_2 & f_3g_1 + f_4g_2 \\ z_2 & f_1g_3 + f_2g_4 & f_3g_3 + f_4g_4 \end{array}\right]. \tag{C6}$$

C.2 Marginalisation

If $f(x, y)$ is a table over x and y, then a table over just y can be produced by marginalising over x, that is,

$$g(y) = \sum_{x} f(x, y). \tag{C1}$$

For example,

$$g(y) = \sum_{x} f(x, y) = \sum_{x} \left[\begin{array}{c|cc} & x_1 & x_2 \\ \hline y_1 & f_1 & f_2 \\ y_2 & f_3 & f_4 \end{array} \right] = \left[\begin{array}{c|c} y_1 & f_1 + f_2 \\ y_2 & f_3 + f_4 \end{array} \right]. \tag{C2}$$

Appendix D: Learning

In the context of models, learning refers to semi-automatic methods that use data to construct or modify a model. In Bayesian belief networks, this learning may be either qualitative or quantitative. Qualitative learning generally corresponds to determining a suitable network structure, and quantitative learning corresponds to determining the conditional probabilities. *Batch learning* uses a database of cases to establish the model, whereas *adaptive learning* modifies a model successively as new cases or data are received [Jensen 1996].

D.1 Batch Learning

Two key issues arise in constructing a Bayesian belief network, namely how well the original table of joint probabilities can be constructed from the model and how much space is required by the model [Jensen 1996]. The original table of probabilities is constructed from observed or inferred knowledge of the system, and the corresponding table produced from the selected model should ideally be an accurate reflection of the original table. The amount of space required by the model is directly proportional to the size or complexity of the model, that is, the number of conditional probabilities in the network. The optimum model is the smallest that has an acceptable modelling error, that is, the simplest structure that adequately describes the probabilistic tables.

Distance measures are required to compare the true and approximate model networks. These distance measures are formed from the differences between the original distribution or probability table and that constructed by the model. The aim is to minimise the chosen distance measure. Two common distance measures are the *Euclidean* and *cross entropy* measures. Defining P as the probability table for the true distribution and P^M as the probability table produced from some model M, and P_c and P_c^M as the c^{th} entries in these tables, the Euclidean distance is defined as

$$\mathrm{Dist}_Q(P, P^M) = \sum_{c \in P} \left(P_c - P_c^M\right)^2 , \tag{D1}$$

and the cross entropy distance as

$$\mathrm{Dist}_L(P, P^M) = -\sum_{c \in P} P_c \frac{P_c}{P_c^M} . \tag{D2}$$

Impossible events, that is, those with zero probability, cannot be accommodated by the cross entropy measure. The size of the distance measure that can be tolerated depends on the application for which the Bayesian belief network is to be used. Therefore, the selection of a threshold for accepting a particular model is a network design issue.

The learning algorithm searches for networks within a specified distance or threshold. As several candidates will generally meet this criteria, the candidate with the smallest size is generally chosen to minimise the complexity of the model. The size of a network is defined as the sum of the spans of all variables with parents, that is,

$$\mathrm{Size}(M) = \sum_{A \in \mathbf{U}} \mathrm{Sp}(A), \tag{D3}$$

where the span of A, $\mathrm{Sp}(A)$, is the number of entries in A's probability table. An acceptance measure, taking account of both the difference between the true and model tables of probabilities and the size of the model, may then be defined as

$$\mathrm{Acc}(P, M) = \mathrm{Size}(M) + k\,\mathrm{Dist}(P, P^M), \tag{D4}$$

where k is a positive real number that is chosen to reflect the relative importance of the size and distance measure. The aim is to minimise $\mathrm{Acc}(P, M)$ subject to $\mathrm{Dist}(P, P^M)$ meeting the distance threshold.

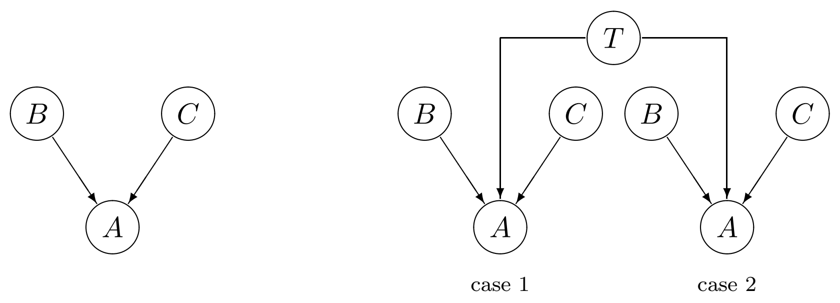
Ideally, all possible DAGs should be considered. However, this is excessively complex, and practical constraints are considered to reduce the number of potential DAGs. A typical approach is to commence with the largest model and successively delete arcs. Once a model has been rejected, further arcs do not have to be removed.

D.2 Adaptation

Adaptive learning is usually limited to updating the conditional probabilities. When there is uncertainty in the conditional probabilities of the network, the network is said to have *second-order uncertainty*. This uncertainty can be addressed by updating the conditional probabilities when new information becomes available.

Consider the simple Bayesian network in figure D1(a), in which there exists some uncertainty in the conditional probability table $\mathbf{P}(A \mid B, C)$. This can be addressed by adding variable T with probability $\mathtt{p}(t)$, as in figure D1(b). The variable T models the uncertainty in $\mathbf{P}(A \mid B, C)$ and its prior probability table $\mathbf{P}(T)$ reflects the initial credibility of expert opinion or experimental data. When new information is received, that is, case 2 in figure D1, the propagation of the probabilities yields new values for $\mathtt{p}(t)$. These new values become the new prior probability table $\mathbf{P}(T)$. This change in probability reflects the new knowledge imparted to the network by the new information.

In practice, a training data set that is representative of the real world data may be used to train the network. Once sufficient data has been processed to determine the appropriate probability tables, the network is available to process the data of interest.



(a) BBN with second order uncertainty

(b) Updating the conditional probabilities

Figure D1: Adaptive learning

Appendix E: Comparison with Decision Theory

Decision theory complements reasoning under uncertainty by providing a coherent framework for making decisions in accordance with the preferences of a *decision manager* [Suermondt 1992, pp. 20–23]. A decision is defined as an *irrevocable allocation of resources*, and the preferences that dictate the decision process represent the relative values that the decision maker places on each possible consequence of a decision. The aim is to maximise the expected utility or benefit resulting from one or more decisions. Decision theory provides an axiomatic basis for the preferences.

Decision bases are represented graphically by *decision trees* and *influence diagrams*. The focus of a decision tree is on the procedural aspects of an evaluation. The influence diagram contains probabilistic dependencies between variables, similar to the Bayesian belief network. However, influence diagrams contain decision nodes that provide choices and value nodes that provide utility measures. It is necessary to know the state of all predecessors of a decision node with certainty before making a decision.

Although influence diagrams and Bayesian belief networks are similar, they are applied to two different problems. Influence diagrams are used as a graphical aid to evaluate the interaction between various aspects of a problem, whereas Bayesian belief networks are used to infer information from available data.

DISTRIBUTION LIST

A Tutorial on Bayesian Belief Networks

Mark L Krieg

	Number of Copies
DEFENCE ORGANISATION	
S&T Program	
Chief Defence Scientist	1
FAS Science Policy	
AS Science Corporate Management	
Director General Science Policy Development	
Counsellor, Defence Science, London	Doc Data Sht
Counsellor, Defence Science, Washington	Doc Data Sht
Scientific Adviser to MRDC, Thailand	Doc Data Sht
Scientific Adviser Joint	1
Navy Scientific Adviser	Doc Data Sht
Scientific Adviser, Army	Doc Data Sht
Air Force Scientific Adviser	1
Director Trials	1
Aeronautical and Maritime Research Laboratory	
Director, Aeronautical and Maritime Research Laboratory	1
Electronics and Surveillance Research Laboratory	
Director, Electronics and Surveillance Research Laboratory	Doc Data Sht
Chief, Surveillance Systems Division	1
Research Leader, Wide Area Surveillance	1
Head, Tracking and Sensor Fusion	1
Dr Martin Oxenham, Surveillance Systems Division	1
Author	1
DSTO Research Library and Archives	
Library Fishermans Bend	Doc Data Sht
Library Maribyrnong	Doc Data Sht
Library Edinburgh	1
Australian Archives	1
Library, MOD, Pyrmont	Doc Data Sht
US Defense Technical Information Center	2
UK Defence Research Information Centre	2
Canada Defence Scientific Information Service	1

NZ Defence Information Centre	1
National Library of Australia	1
Capability Systems Staff	
Director General Maritime Development	Doc Data Sht
Director General Aerospace Development	Doc Data Sht
Knowledge Staff	
Director General Command, Control, Communications and Computers (DGC4)	Doc Data Sht
Army	
Stuart Schnaars, ABCA Standardisation Officer, Tobruk Barracks, Puckapunyal, 3662	4
SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), MILPO Gallipoli Barracks, Enoggera QLD 4052	Doc Data Sht
Intelligence Program	
DGSTA, Defence Intelligence Organisation	1
Manager, Information Centre, Defence Intelligence Organisation	1
Corporate Support Program	
Library Manager, DLS–Canberra	1
UNIVERSITIES AND COLLEGES	
Australian Defence Force Academy Library	1
Head of Aerospace and Mechanical Engineering, ADFA	1
Hargrave Library, Monash University	Doc Data Sht
Librarian, Flinders University	1
OTHER ORGANISATIONS	
NASA (Canberra)	1
AusInfo	1
State Library of South Australia	1
ABSTRACTING AND INFORMATION ORGANISATIONS	
Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts, US	1
Documents Librarian, The Center for Research Libraries, US	1
INFORMATION EXCHANGE AGREEMENT PARTNERS	
Acquisitions Unit, Science Reference and Information Service, UK	1

Library – Exchange Desk, National Institute of Standards and Technology, US	1

SPARES

DSTO Edinburgh Research Library	4

Total number of copies:	42

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. CAVEAT/PRIVACY MARKING

2. TITLE A Tutorial on Bayesian Belief Networks	3. SECURITY CLASSIFICATION Document (U) Title (U) Abstract (U)
4. AUTHOR Mark L Krieg	5. CORPORATE AUTHOR Electronics and Surveillance Research Laboratory PO Box 1500 Edinburgh, South Australia, Australia 5111

6a. DSTO NUMBER DSTO–TN–0403	6b. AR NUMBER 012–084	6c. TYPE OF REPORT Technical Note	7. DOCUMENT DATE December, 2001

8. FILE NUMBER B9505/21/217	9. TASK NUMBER JNT 00/189	10. SPONSOR	11. No OF PAGES 44	12. No OF REFS 12

13. URL OF ELECTRONIC VERSION http://www.dsto.defence.gov.au/corporate/reports/DSTO–TN–0403.pdf	14. RELEASE AUTHORITY Chief, Surveillance Systems Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved For Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SOUTH AUSTRALIA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS

No Limitations

18. DEFTEST DESCRIPTORS

Bayes theorem
Probability theory
Applications of mathematics

19. ABSTRACT

This tutorial provides an overview of Bayesian belief networks. The subject is introduced through a discussion on probabilistic models that covers probability language, dependency models, graphical representations of models, and belief networks as a particular representation of probabilistic models. The general class of causal belief networks is presented, and the concept of d-separation and its relationship with independence in probabilistic models is introduced. This leads to a description of Bayesian belief networks as a specific class of causal belief networks, with detailed discussion on belief propagation and practical network design. The target recognition problem is presented as an example of the application of Bayesian belief networks to a real problem, and the tutorial concludes with a brief summary of Bayesian belief networks.

Page classification: UNCLASSIFIED