# LINK STRENGTH IN BAYESIAN NETWORKS

by

BRENT BOERLAGE

B.A.Sc., The University of British Columbia, 1983

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF COMPUTER SCIENCE

We accept this thesis as conforming

to the required standard

……

……

THE UNIVERSITY OF BRITISH COLUMBIA

October 1992

# Abstract

This thesis introduces the concept of a *connection strength* (CS) between the nodes in a propositional Bayesian network (BN). Connection strength generalizes node independence from a binary property to a graded measure. The connection strength from node A to node B is a measure of the maximum amount that the belief in B will change when the truth value of A is learned. If the belief in B does not change, they are independent (zero CS), and if it changes a great deal, they are strongly connected (high CS).

Another concept introduced is the *link strength* (LS) between two adjacent nodes, which is an upper bound on that part of their connection strength which is due only to the link between them (and not other paths which may connect them). Calculating connection strengths is computationally expensive, while calculating link strengths is not. A linear complexity algorithm is provided which finds a bound on the connection strength between any two nodes by combining link strengths along the paths connecting them. Such an algorithm lends substance to notions of an "effect" or "influence" flowing along paths, and "effect" being attenuated by "weak" links, which is terminology that has appeared often in the literature, but only as an intuitive idea.

An algorithm for faster, approximate BN inference is presented, and connection strengths are used to provide bounds for its error. A system is proposed for BN diagrams to be drawn with strong links represented by heavy lines and weak links by fine lines, as a visualization aid for humans. Another visualization aid which is explored is the *CS contour map*, in which connection strengths from one node to the rest are represented as contour lines super-imposed on a regular BN diagram, allowing the viewer to quickly assess which nodes that node influences the most (or which nodes influence it the most). A non-trivial example BN is presented, some of its connection strengths are calculated, CS contour maps are constructed for it, and it is displayed with link strength indicated by line width.

# Contents

# Acknowledgments

I would like to thank:

David Lowe - My supervisor, who allowed enough flexibility in my program to really get the education I wanted and find the field that interested and benefited me most.

David Poole - Who introduced me to probabilistic reasoning, and who created a group of people studying Bayesian networks at UBC.

The Bayesian network people at UBC - David Poole, Mike Horsch, Nevin (Lianwen) Zhang, Yang Xiang, Keiji Kanazawa, Runping Qi, and Greg Provan for useful discussions, and David Lowe, David Poole and Mike Horsch for useful comments on my thesis.

The many graduate students at UBC who I had the pleasure of knowing during my *long* stay there. If I named them I would have a long list, and I'm sure I would miss some, so I'll resist the temptation to put even one name.

My parents - Who have always greatly encouraged and aided me in my education.

My wife, Christine - For patience and support.

# 1 Introduction

*Bayesian networks* (BNs), also called belief networks or probabilistic causal networks, consist of graphs in which each node represents a variable of interest, and the links between the nodes indicate the probabilistic dependencies between them. This thesis is restricted to BNs composed only of binary nodes (which are nodes representing variables that take on one of two values), and generally speaking we will interpret their value to mean that some proposition is TRUE or FALSE. Using a BN we can capture the relationships between our uncertain beliefs in the propositions, and then if we learn the truth value of one or more of the propositions, we can use BN inference algorithms to find updated beliefs for each of the other propositions, and updated relationships between the propositions.

Uncertain reasoning is very common for humans, and will conceivably be common in future machines. Chapter 2 provides some examples of situations requiring uncertain reasoning. Then it makes an argument in favor of using probabilities for such reasoning (in machines), and provides a well-known set of axioms for doing so. However, probabilistic reasoning can be extremely computationally expensive, and even quite small problems can be outside the range of practicality, unless we have some technology for taking advantage of the fact that generally our beliefs in some propositions are independent of our beliefs in others (Pearl88, Cooper90). This is the primary purpose of the BN graph. Chapter 2 goes on to show how BNs represent these independencies, and provides a non-trivial example of a BN. Some BN inference algorithms are also discussed, and it is pointed out that the BN inference problem is NP-hard (Cooper90).

*Connection strength* (CS) is a generalization of independence. Instead of simply indicating whether one proposition is independent of another, it provides a graded measure of how much our belief in one proposition can change when we learn the truth or falsity of another. Chapter 3 defines connection strength and explores some of its properties. It also introduces the concept of *link strength* (LS), which is a sort of "local connection strength" between two adjacent nodes of a BN. Link strength provides an upper bound on that part of the connection strength between two adjacent nodes that is due to the single link between them, and not due to any other paths from one of them to the other.

Computing the connection strength between two nodes is generally even more computationally expensive than regular BN inference, but a link strength can be found very quickly using only the conditional probabilities stored at a single node. Chapter 4 presents an algorithm which uses link strength values to find a bound for the connection strength between any two nodes in time linear in the number of links in the BN. The algorithm can be viewed as a summation over alternative paths between the nodes, which lends substance to notions of an "effect" flowing along paths, and "effect" being attenuated by "weak" links, which is terminology that has often appeared in the literature as an intuitive idea, but which has never been substantially formalized.

Using independence information allows BN inference algorithms to solve medium-sized BN problems in a reasonable amount of time. Using connection strengths we can determine which nodes are *nearly* independent, and then by assuming that they *are* independent, we can solve larger-sized BN problems in reasonable time, while obtaining approximate results. In Chapter 5 an algorithm is given which quickly provides bounds on the maximum error made during such an approximation.

Another application of link strengths explored in Chapter 5, is to display them on BN diagrams as a visualization aid for humans. For example, the width of the line representing a link can be used to indicate its link strength, with finer lines for weaker links, and thicker lines for stronger links. The example BN of Chapter 2 is redrawn in such a manner to illustrate this. BNs have been praised as a great tool for humans to visualize probabilistic relations, and displaying link strength extends that tool by providing graded, rather than binary, independence information.

Another visualization aid for BNs are *CS contour maps*, which indicate how much some node (termed the "origin node") can effect each of the other nodes in the network, and are created by drawing "iso-CS" lines over the BN diagram. Each line separates nodes which are more strongly connected to the origin node, from those less strongly connected. Chapter 5 contains a CS contour map for the example BN of Chapter 2. It also contains a contour map based on CS bounds calculated by the algorithm developed in Chapter 4. By comparing the two contour maps, the bounds may be compared with the true values.

Wellman90 introduces the concept of *qualitative probabilistic networks* (QPNs), which are networks with the same underlying topology as BNs, and whose purpose is to determine the *direction* of change in belief of one proposition when we learn the truth of another. We can consider connection strength as determining the maximum *magnitude*, and QPNs as determining the *sign*, of the same quantity. In fact, many of Wellman's results can be obtained from the connection strength equations, simply by modifying them to retain sign information (e.g. removing absolute value functions). This is briefly discussed in the "Further Work" section at the end of the thesis.

Notation is explained as it is introduced, but it is also summarized in Appendix A. Readers who are already familiar with BNs can skip the next chapter and go straight to chapter 3, using Appendix A as a guide for notation definitions they may have missed.

# 2 Bayesian Networks

This chapter provides a brief introduction to probabilistic reasoning and Bayesian networks (BNs), and states a number of well-known results which will be used later. It does not contain any original results, and so it may be skipped by the knowledgeable reader. Two good introductory books for Bayesian networks and related topics are Pearl88 (a "must read" for a thorough introduction) and Neapolitan90 (easier to read, has more how-to information and has more recent results, but generally doesn't have the depth of analysis).

## 2.1 When do we Reason With Uncertainty?

Reasoning with uncertainty is the process of combining items of uncertain knowledge to obtain uncertain conclusions. Uncertain knowledge is knowledge which one would be willing to retract, or consider less certain, upon receiving knowledge (certain or uncertain) to the contrary. There are not many things we know that we wouldn't be willing to retract given enough evidence to the contrary, so much of our reasoning can be considered reasoning with uncertainty.

In this thesis, the main purpose of studying uncertain reasoning will be to produce computer-based automated systems, although some of the results may apply to other intelligent agents. In constructing an automated system, we must decide which of its information it should treat as uncertain. We may want it to treat some information as certain, even though it doesn't hold in all cases (or we don't know if it does), just to make the system simpler, or give it a more predictable behavior. On the other hand, we may want it to consider much of what it learns, based on limited or imperfect observations, as uncertain information. Generally speaking, as the

sophistication of our automated systems increase, a larger percentage of their knowledge should be treated as uncertain, since many of these systems will be more adaptable, will be learning more, and will be working in less well-defined domains in an autonomous manner.

There are numerous particular situations where an automated system may need to reason with uncertainty. Any approximate measurement is information with uncertainty. So combining uncertain or approximate observations, such as physical measurements, which have a redundancy in the observations for the purposes of increasing the accuracy or detecting a totally erroneous observation, requires some form of reasoning with uncertainty. The reasoning may be as simple as taking the mean of the set of measurements, or it may involve a complex analysis. Any situation in which we have information coming from multiple sources, which may agree or conflict to varying degrees, requires reasoning with uncertainty. Examples are sensor fusion in a robot (i.e. combining sensory data), or merging news reports from different agencies.

In some reasoning situations, much of the knowledge involved is nearly certain, and we can gain huge computational savings by treating it the same way we treat knowledge that is certain. However, when we learn something that casts doubt on some piece of it, we may have to revert the status of that piece back to "uncertain", or even to "false", and suitably modify the status of related pieces. The methods of *default reasoning* or *nonmonotonic logic* have traditionally been used to do this.

Some "inverse" problems (such as diagnosis, machine vision, machine hearing, and other recognition problems) don't have a unique solution. Reasoning with uncertainty can help to find the most probable solutions to these problems.

Problems in which an agent learns generalizations from case data are examples of reasoning with uncertainty. Usually, the more cases the agent sees the more certain he becomes about the generalizations. When the generalizations are applied to predict unknown values in a new case, more reasoning with uncertainty is required, both because the generalization is uncertain, and because its applicability to the new case may be uncertain.

We may even use reasoning with uncertainty for problems that can be stated in purely logical (i.e. certain) terms, such as theorem proving. Often, these types of problems can be considered

to involve some kind of search, and they can be solved far more quickly if we use suitable heuristics to guide the search to examine the most probable candidates first. But as the use of heuristics becomes more sophisticated, it becomes difficult to combine conflicting heuristics, and so it is useful to think of the heuristics as uncertain knowledge. Then, we can use reasoning with uncertainty to direct the search of the theorem prover (or other strictly logical reasoning-with-certainty system).

## 2.2 Using Probability for Uncertain Reasoning

There are a number of mathematical systems available for reasoning with uncertainty, and there has been considerable controversy over which is "best". These systems include subjective probability, fuzzy logic, belief functions (e.g. Dempster-Shafer), certainty factors, non-numerical probabilities, and default logic. This thesis uses a system based on subjective probability, that is often called "Bayesian probabilistic reasoning", and some approximations to Bayesian reasoning will also be considered.

de Finetti provides an argument in favor of subjective probabilities based on the notion of *coherence* (F. P. Ramsey and L. J. Savage have also done similar work). An agent is offered a number of betting options and his choices are analyzed. If he acts as though his beliefs were governed by rules other than those of probability, it is possible to arrange a series of bets with him (called a Dutch book) in which he is *guaranteed* to lose money regardless of how events unfold.

Cox46 derives the rules of probability without any of the machinery of gambling or decisions, based only on meeting a list of desirable properties for an ideal agent to reason with uncertainty. The proof is more complex than the de Finetti proof, but seems to have broader applicability. Below is the list of desired properties from which all the axioms of probability theory can be derived. Since this thesis is built upon the theory of probability, these may be considered the assumptions about reasoning with uncertainty made by this thesis.

1. **Clarity**: Propositions must be defined precisely enough so that it would be theoretically possible to determine whether they are indeed true or false.

2. **Scalar Continuity**: A single real number is both necessary and sufficient for representing a degree of belief.

3. **Completeness**: Some degree of belief can be meaningfully assigned to any well-defined proposition.

4. **Context dependency**: The belief assigned to a proposition can depend on the belief in other propositions.

5. **Hypothetical conditioning**: There exists some function that allows the belief in a conjunction of two propositions to be calculated from the belief in the first proposition, and the belief in the second proposition given that the first is true.

6. **Complementarity**: The belief in the negation of a proposition is a monotonically decreasing function of the belief in the proposition itself.

7. **Consistency**: There will be equal belief in propositions that are logically equivalent.

Cox46 originally proved that the probability axioms follow from these properties, but Tribus69 weakened the hypothetical conditioning requirement, and the solution of the "associativity equation" in Aczel66 may be used to remove the differentiability assumptions required by Cox. The list 1-7 was taken (with some modifications) from HorvitzHeckermanLanglotz86, who clearly state and name the desired properties, and use the results to compare existing uncertainty systems.

Some people have argued for uncertainty systems that violate one of the properties 1-7, or that have internal inconsistencies. For example, the Dempster-Shafer theory violates the combination of 2, 3, and 6. One justification for doing this is that some way of representing total ignorance about a proposition is required. Property 3 requires that we specify a belief for every proposition and property 2 requires that it be only a single number, which appears to preclude any representation of ignorance. Bayesians have pointed out that for decision theory a representation of ignorance is not required, although for learning and communication it may be. Sometimes using Bayesian probabilities for beliefs about real-world frequencies, instead of just

beliefs about the occurrence of a single event, will handle these situations. Otherwise, true Bayesian probabilities can be augmented with a confidence measure (such as an "equivalent sample size", which may be considered as roughly equivalent to the number of relevant cases used to form a learned BN), which can be combined with the probability when necessary to provide enough information for someone with different priors to update their priors. The issue remains controversial, but without doubt the Bayesian method is suitable for a very large class of uncertainty problems.

Fuzzy logic appears to violate property 1, but the primary complaint of the fuzzy logic community is that probabilities can not represent all the different types of uncertainty that arise. They distinguish between vagueness (fuzziness, haziness, etc.), and ambiguity (nonspecificity, variety, etc.). Bayesians do claim to be able to handle these types of uncertainty using only probabilities (for example, see Cheeseman86).

Expert systems based on simple evidential support may be misled when they use transitivity to chain rules. For example, if B provides support for C, and C provides support for D, a support based system may increase support for D upon observing B, which may be inappropriate. An example from Pearl88 is that "wet grass" may provide support for "rained last night", and "sprinkler on" may provide support for "wet grass". Each of these two rules work fine by themselves but chaining them suggests that "sprinkler on" supports "wet grass" which in turn supports "rained last night". However, if we see the sprinkler on we should probably *decrease* our belief that it rained last night, instead of increasing it.

A good example of the ad hoc nature of certainty factors, and the confusion resulting in trying to apply them, is the transcript of email exchanges of the MYCIN developers (BuchananS84, pp. 221-232). Later, Heckerman (1986) analyzed certainty factors and showed that (when the inconsistencies were removed) they were equivalent to using probabilities, but with certain independence assumptions being implicitly made. In cases where these assumptions are inappropriate, using certainty factors may produce misleading results. By comparing the system to a probabilistic one, it became clearer where the deficiencies were, and how serious they were.

Proponents of default reasoning are often concerned with the numbers normally used in probabilistic reasoning, and ask "Where do these numbers come from?" The numbers represent subjective beliefs, not exact real-world frequencies, so the reasoning agent is simply summarizing his beliefs with the numbers. In fact, given a series of observations from nature, it seems easier to find probabilities (either as frequencies or doing Bayesian reasoning over priors on frequencies), than to decide on a set of default rules. Nevertheless, default reasoning can be a very valuable way of reasoning with uncertainty, since it sometimes has significant computational advantages, and can reduce the information rate considerably during communication.

It is dangerous to assume that a proof like the one by Cox completely defines the "best" system to use. It is always difficult to foresee what types of systems will be successful in the future, partly because the nature of the problems being solved keeps changing. However, the proof is useful for understanding the fundamental differences between uncertainty systems, and if one is willing to accept the properties 1-7, it indicates that probability is the system to use.

Doing complete probabilistic reasoning is generally very computationally expensive (see section 2.6), so it is reasonable to look for alternatives. Sometimes the most appropriate algorithm is as simple as taking the average of a few values, or using a majority vote scheme. The approach advocated in this thesis is to start with a normative theory for combining uncertain information, and then use a simplified scheme, when appropriate, to improve the computational speed or overall complexity of the reasoning. However, it should be considered an approximation to using probabilities, and should be judged based on how similar its results are to those of full scale Bayesian reasoning. One of the applications of "link strength" is to indicate when it is appropriate to take a certain type of computational shortcut in probabilistic reasoning, and provide a bound on the resulting inaccuracy.

## 2.3 Axioms of Probability Theory

Here is a set of axioms for probability theory equivalent to those derived by Cox (and also compatible with other axiomatizations of probability, such as the Kolmogorov axioms):

1. $P(a|a) = 1$ 2.3:1
2. $P(\neg a|b) = 1 - P(a|b)$
3. $P(a, b|c) = P(a|b, c)\ P(b|c)$

where P(x,y|z) means the probability that propositions x and y are both true, given that proposition z is true. For notational convenience we use P(x) to mean P(x|T) where T is a tautology that is always true. Upper case letters refer to propositional variables, and lower case to their value. +b stands for b=TRUE, ¬b stands for b=FALSE, and sometimes +b is written simply as b if that does not result in confusion. For more notational details see Appendix A.

From these few axioms, together with propositional logic and arithmetic, we can generate the entire mathematical structure of probability theory. If we are willing to accept using these 3 simple axioms for a system to reason with uncertainty, and we require that it is consistent, then we are bound to accept using probability theory.

Below is a list of a few theorems that I will use later. For their proofs, or alternate (but equivalent) axiomatizations of probability theory, see an elementary text such as Lindley65. This is *Bayes theorem:*

$$P(a|b, c) = P(b|a, c)\frac{P(a|c)}{P(b|c)} \qquad 2.3:2$$

This is the *reasoning by cases theorem:*

$$P(a|c) = P(a|b, c)\ P(b|c) + P(a|\neg b, c)\ P(\neg b|c) \qquad 2.3:3$$

This is the *independence theorem:*

$$P(a, b|c) = P(a|c)\ P(b|c) \quad \text{iff A is independent of B given C} \qquad 2.3:4$$

where *independence* is defined as:

$$P(a|b, c) = P(a|c) \quad \text{iff A is independent of B given C, providing b and c are consistent} \qquad 2.3:5$$

**Theorem 2.3:6**: In each of the above theorems, all the probabilities are conditioned on the proposition c. Since it could be equivalent to any logical formula, they all hold with c replaced by any vector of truth values, **c**.

The *full joint probability distribution* (FJD) specifies a probability for every possible conjunction involving every proposition of interest (or its negation). For example, if an agent knew only about the propositions A, B, and C, his FJD would consist of the probabilities P(abc), P(ab¬c), P(a¬bc), P(a¬b¬c), P(¬abc), P(¬ab¬c), P(¬a¬bc), and P(¬a¬b¬c). Supplying an FJD for a problem completely specifies the problem probabilistically. Sometimes I will refer to the FJD of a problem or an agent, when I want to indicate the complete probabilistic specification, although it may not be represented in any table. Obviously, if the FJD were stored in a table, the size of the table would be exponential in the number of base propositions. We must take advantage of independencies between propositions to more efficiently represent an FJD.

## 2.4 Representing Independencies

Using just the axioms and theorems of the previous section we can do probabilistic reasoning. That is, given beliefs for a set of propositions and their conjunctions (or relations), we can combine them with new items of certain (or uncertain) knowledge about the propositions or their relations, to obtain new beliefs for any of the propositions, or any logical formula of the propositions. But unless we exploit the fact that under some conditions our belief in some of the propositions will be independent of some others, then for even a moderately sized problem, the computational cost will be astronomical. We could represent all the independencies as a list of triples, each one of the form $I(\mathbf{X}, \mathbf{Y} | \mathbf{z})$, where **X**, **Y**, and **Z** are sets of propositions, and $I(\mathbf{X}, \mathbf{Y} | \mathbf{z})$ means that if **Z** is precisely the set of propositions we know the truth value of, then **X** is independent of **Y** (i.e. further knowledge of whether the propositions in **X** are true won't change our beliefs in any of the propositions in **Y**). However, this list would be impossibly large and awkward to deal with since the number of possible sets for each of **X**, **Y**, and **Z** is exponential (the power sets) in the number of base propositions.

Once some independencies are known, generally others must follow to be consistent with the axioms of probability. So we could keep a partial list of independencies, and generate the others as needed, using the *graphoid axioms* (Pearl88):

| | |
|---|---|
| Symmetry | $I(\mathbf{X}, \mathbf{Y} \mid \mathbf{z}) \Rightarrow I(\mathbf{Y}, \mathbf{X} \mid \mathbf{z})$ |
| Decomposition | $I(\mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{z}) \Rightarrow I(\mathbf{X}, \mathbf{Y} \mid \mathbf{z})$ |
| Weak Union | $I(\mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{z}) \Rightarrow I(\mathbf{X}, \mathbf{Y} \mid \mathbf{z} \cup \mathbf{w})$ |
| Contraction | $I(\mathbf{X}, \mathbf{Y} \mid \mathbf{z}) \ \& \ I(\mathbf{X}, \mathbf{W} \mid \mathbf{z} \cup \mathbf{y}) \Rightarrow I(\mathbf{X}, \mathbf{Y} \cup \mathbf{W} \mid \mathbf{z})$ |

However using these axioms to generate required independence information would generally turn out to be a major computational task in itself. Instead we use a *dag* (directed acyclic graph) to represent the independencies, and by using a simple and very fast algorithm called *d–separation*, we can use the dag to quickly find independence information.

We require that the dag not represent any independency that isn't in the original FJD. Unfortunately, due to the limited representational power of dags, the result is that sometimes not all the independencies of the FJD can be represented by a dag. Since we are using the independence information to speed computation, the fact that it sometimes misses an independency means that sometimes we will do a little more computation than necessary, but since it never reports an independency when there isn't one, the computation will never be in error.

In the dag representation, each proposition of interest is represented by a node in a graph. Directed arcs (called *links* in this thesis) connect the nodes. The graph may be constructed by choosing any total ordering for the nodes, starting with no links, and then stepping through the ordering and adding links to each node N as it is encountered. The links to add to node N are determined by examining each node preceding node N in the ordering, and adding a link from it to N if N is not independent of that node given all the other nodes preceding N (for all values of those other nodes). This process will be examined again in section 3.8, and in section 2.5 I give an example which shows how our natural knowledge of causality normally makes the process much easier.

The dag that results can be used to regenerate the independence information via the d-separation algorithm. In general, its structure will depend on the total ordering of the nodes that we started with. In order to create the simplest model, and to represent as many independencies as possible, it is desirable to chose an initial total ordering that will minimize the number of links in the final dag. In models of physical causation, generally placing propositions about causes before propositions about their effects will result in a smaller dag. An example of a dag to represent independence appears in figure 2.5:1.

The graph created by the above algorithm never has any cycles (i.e. paths that return to their starting point following the direction of the links), but it may have loops (paths that return to their starting point ignoring the direction of the links). A graph without loops is called *singly-connected*, and one with loops is called *multiply-connected*. The nodes with links going to a node N are called the *parents* of N, and the nodes at the end of links leaving N are called the *children* of N. The definitions of *ancestors*, *descendants*, *siblings*, etc. follow in the natural way.

The recent heightened interest in using normative probabilistic systems to reason with uncertainty, and the construction of a number of practical systems, is due in a large part to exploiting the independencies represented by a dag description of the FJD. One of the contributions of this thesis is to provide a criterion for when it is appropriate to exploit "near-independencies" as well, and to generalize the d-separation algorithm to discover them.

### 2.4.1 d-Separation Algorithm

The d-separation algorithm is used to determine the independencies represented by a dag. It is an algorithm which allows dags to represent independencies in a manner consistent with the graphoid axioms (and therefore the axioms of probability).

Say **X**, **Y**, and **Z** are disjoint subsets of nodes in the dag. We will use the d-separation algorithm to determine if **X** and **Y** are independent given **Z** ("given **Z**" means the same thing as "having evidence for **Z**", which means the truth values of all the propositions in **Z** are known by the reasoning agent).

A node is termed *converging* with respect to a path, iff it is a node along the path in which both the links to enter and to leave the node are directed toward the node (e.g. the third node B from the left in figure 2.4).

A path is *blocked* by **Z** iff:

1. It has a non converging node in **Z**, or

2. It has a converging node N that is not in **Z**, and N has no descendants in **Z**. A path that is not blocked is called *active*. Figure 2.4 illustrates some blocked and active paths. In each case A is a node from **X**, C is a node from **Y**, and B is in **Z** iff it is shaded.

**Z** *d-separates* **X** from **Y** iff all paths from **X** to **Y** are blocked by **Z**. If there is any active path from **X** to **Y**, then **X** and **Y** are not d-separated by **Z**.

If **Z** d-separates **X** from **Y**, then **X** and **Y** are independent given **Z**. That is, if you know **Z**, further knowledge of **X** will not shed any light on **Y**, and vice versa. It is important that **Z** include all the nodes for which you have knowledge, since some of the nodes in **Z** may block all the paths from a node in **X** to a node in **Y**, but others may form new paths.

You can check your ability to apply the d-separation algorithm by trying to find all the active paths from node V to node Q in the BN of figure 2.6:2, where the shaded nodes are the ones for which you have knowledge. The only two active paths are the marked ones.

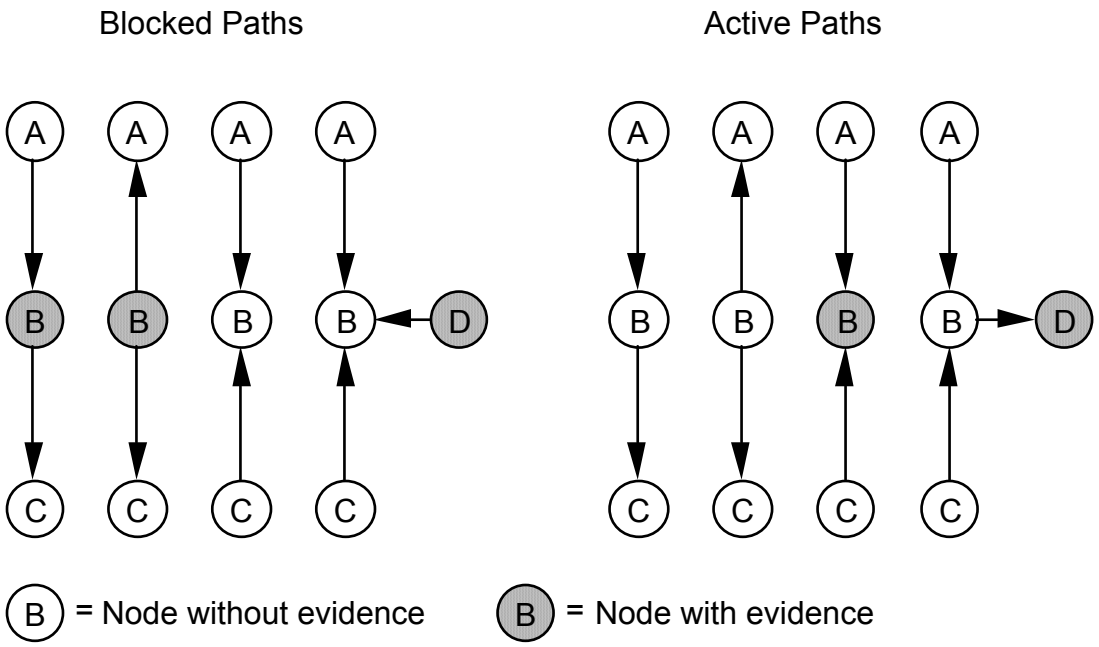


**Figure 2.4** - Each case indicates whether the displayed path between A and C is active or blocked. A node with "evidence" is one whose truth value is known. In each case there may be other links connected to A, B, C, or D, so there may be other paths between A and C which may be active or blocked.

## 2.5 Bayesian Networks

We can use the dag representation of independence introduced in the previous section, together with a set of conditional probabilities at each node, to provide a complete representation of any particular FJD. We call the resulting structure a Bayesian network (BN).

In the common definition of a BN, nodes represent any kind of variable of interest, but since this thesis is restricted to the study of binary nodes, our nodes represent propositions (or variables that can take on one of two states).

To construct a BN, a dag may be determined as described in the previous section. Then, a number of conditional probabilities called the *node conditional probabilities* (NCPs), are attached to each node. These are the probabilities that the proposition of the node is TRUE given each of the different TRUE/FALSE combinations of its parents. For example, if the node B had the single parent A, its NCP would be {P(b|a), P(b|¬a)}, if it had parents A and W its NCP would be {P(b|aw), P(b|a¬w), P(b|¬aw), P(b|¬a¬w)}, and if it had no parents its NCP would be {P(b)}.

These probabilities are *subjective* probabilities, which are unique for the reasoning agent constructing and using the BN. They measure to what degree the agent would believe the child proposition if he knew the truth about the parent propositions. When appropriate, they correspond to frequencies in the real world, but that is not required. The purpose of BN inference is simply to tell the agent how to change beliefs when he observes certain evidence. The structure of the links is often called the *topology* of the BN to distinguish it from the conditional probability information.

The FJD is easily reconstructed from the BN representation via the equation:

$$P(\mathbf{v}) = \prod_{X \in V} P(x|\pi(X)) \qquad 2.5$$

where **V** is the set of nodes in the BN, P(**v**) is a probability from the FJD for the vector of truth values **v** (one value for each node in **V**), X is a node, $\boldsymbol{\pi}(X)$ is a vector of values (consistent with **v**) for the parents of X, and $P(x|\boldsymbol{\pi}(X))$ is an NCP for X, where both x and $\boldsymbol{\pi}(X)$ are consistent with **v**. In other words, the joint probability of a setting of TRUE/FALSE values for all the nodes, is simply the product over the nodes, of the NCP from each node which is consistent with the setting.

A single FJD can generally be represented by several different BNs, but if the BN must satisfy a given total ordering for the nodes, then it will be unique. Any propositional FJD can be represented by some BN having one node for each propositional variable of the FJD. A BN uniquely (and therefore unambiguously) determines an FJD. Every possible BN determines some FJD, so it is impossible to construct an inconsistent BN no matter what its NCPs or its topology (providing its acyclic).

The direction of each link in a BN is significant. A link from node A to node B may always be reversed in direction, but if it was already in its optimal orientation, then the reversal usually requires the addition of extra links from the parents of A going to B, and from the parents of B going to A, to avoid indicating independencies that don't exist. Once the extra links have been added, the BN doesn't represent all the independencies it did before the link reversal. Also, with the links in a non-optimal direction our knowledge is less modular, in that adding new variables

(nodes) will generally require adding a great many new links. In those problems where the variables are causally related, the optimal direction for links is generally the direction of causality (e.g., the direction of time).

An example BN is shown in figure 2.5:1, and its NCPs are shown in figure 2.5:2. This BN represents the relationships between the beliefs of Jim, who is a fictitious character that lives in a small community which also contains Hank, Tom, Molly, and Gale. The NCPs are the subjective probabilities of Jim only. For example, the NCPs of node MT represent Jim's belief of what Molly thinks (notice that in this example it mirrors what Jim himself believes, since the NCPs of node MT are the same as the NCPs of node TD, but this is not required).

Each node should specify a proposition precisely to the user of the BN, in order to satisfy the clarity condition (see section 2.2) of probabilistic reasoning. For example, "Tom" must refer to a particular person, so including a last name may help, "a big donation" must refer to a particular range of donation sizes, so including dollar amounts may help, and "lots of cars" must refer to a particular range in number of cars, so providing numbers may help. Perfectly describing each node to someone completely unfamiliar with the situation may require an endlessly long description, but any description is adequate as long as it produces in the mind of the BN builder and the BN users a proposition of adequate preciseness for the task at hand.

It may appear that most of the important causes for many of the nodes have not been included. For example "Park is approved", has only "Molly elected mayor" as a parent. Surely there are many more important factors, such as the need for a park, the existence of necessary funds, the availability of land, etc. However, the purpose of using probabilities and reasoning with uncertainty is to be able to reason without *explicitly* accounting for all these factors. The probabilities themselves summarize the missing factors (if all factors were accounted for, perhaps the NCPs would consist of only 0s and 1s). In building a BN we need include only those factors we know of and suspect are relevant, and the uncertainty in the inference results will reflect our lack of knowledge. If Jim were an actual person he would probably know of many more factors that were relevant to his real-life questions involving these nodes, so he could add nodes and links for them to expand our example BN, and thereby generally increase the expected certainty of his conclusions.

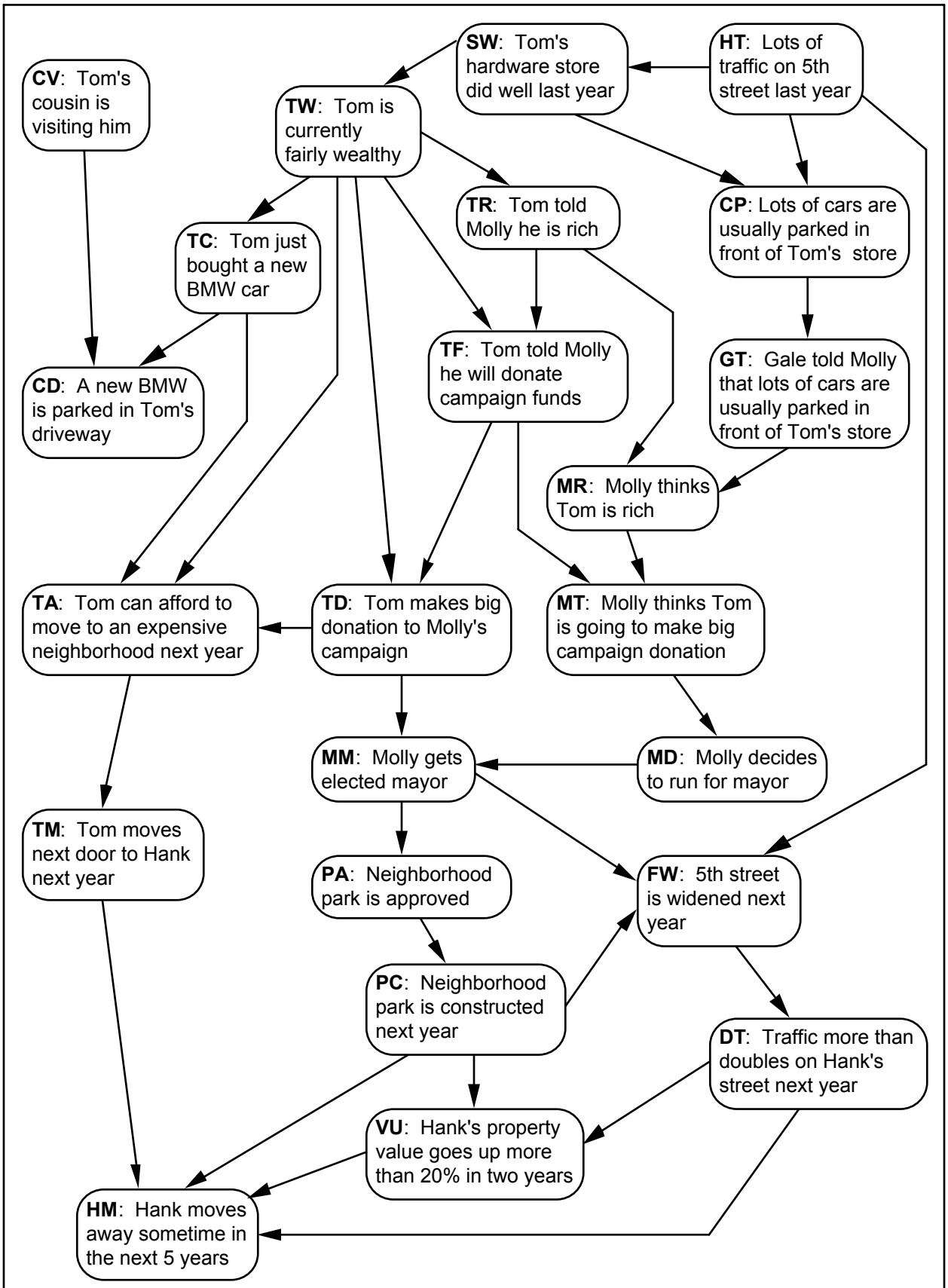


**Figure 2.5 -** Example BN.

```
P(ht) = 0.7

P(sw|+ht) = 0.7
P(sw|-ht) = 0.6

P(cp|+sw,+ht) = 0.8
P(cp|+sw,-ht) = 0.7
P(cp|-sw,+ht) = 0.7
P(cp|-sw,-ht) = 0.2

P(gt|+cp) = 0.1
P(gt|-cp) = 0.002

P(tw|+sw) = 0.7
P(tw|-sw) = 0.3

P(tr|+tw) = 0.1
P(tr|-tw) = 0.05

P(tf|+tr,+tw) = 0.60
P(tf|+tr,-tw) = 0.15
P(tf|-tr,+tw) = 0.20
P(tf|-tr,-tw) = 0.05

P(mr|+tr,+gt) = 0.71
P(mr|+tr,-gt) = 0.70
P(mr|-tr,+gt) = 0.31
P(mr|-tr,-gt) = 0.30

P(mt|+tf,+mr) = 0.8
P(mt|+tf,-mr) = 0.5
P(mt|-tf,+mr) = 0.1
P(mt|-tf,-mr) = 0.02

P(td|+tf,+tw) = 0.8
P(td|+tf,-tw) = 0.5
P(td|-tf,+tw) = 0.1
P(td|-tf,-tw) = 0.02

P(tc|+tw) = 0.3
P(tc|-tw) = 0.1

P(cv) = 0.01

P(cd|+cv,+tc) = 0.95
P(cd|+cv,-tc) = 0.90
P(cd|-cv,+tc) = 0.90
P(cd|-cv,-tc) = 0.05

P(ta|+tw,+tc,+td) = 0.800
P(ta|+tw,+tc,-td) = 0.802
P(ta|+tw,-tc,+td) = 0.810
P(ta|+tw,-tc,-td) = 0.812
P(ta|-tw,+tc,+td) = 0.300
P(ta|-tw,+tc,-td) = 0.302
P(ta|-tw,-tc,+td) = 0.310
P(ta|-tw,-tc,-td) = 0.312

P(tm|+ta) = 0.3
P(tm|-ta) = 0.05

P(md|+mt) = 0.7
P(md|-mt) = 0.5

P(mm|+md,+td) = 0.5
P(mm|+md,-td) = 0.3
P(mm|-md,+td) = 1e-7
P(mm|-md,-td) = 1e-7

P(pa|+mm) = 0.7
P(pa|-mm) = 0.4

P(pc|+pa) = 0.9
P(pc|-pa) = 1e-5

P(fw|+ht,+mm,+pc) = 0.52
P(fw|+ht,+mm,-pc) = 0.50
P(fw|+ht,-mm,+pc) = 0.42
P(fw|+ht,-mm,-pc) = 0.40
P(fw|-ht,+mm,+pc) = 0.15
P(fw|-ht,+mm,-pc) = 0.15
P(fw|-ht,-mm,+pc) = 0.12
P(fw|-ht,-mm,-pc) = 0.10

P(dt|+fw) = 0.8
P(dt|-fw) = 0.2

P(vu|+pc,+dt) = 0.80
P(vu|+pc,-dt) = 0.82
P(vu|-pc,+dt) = 0.50
P(vu|-pc,-dt) = 0.51

P(hm|+pc,+vu,+dt,+tm) = 0.12
P(hm|+pc,+vu,+dt,-tm) = 0.13
P(hm|+pc,+vu,-dt,+tm) = 0.10
P(hm|+pc,+vu,-dt,-tm) = 0.11
P(hm|+pc,-vu,+dt,+tm) = 0.11
P(hm|+pc,-vu,+dt,-tm) = 0.12
P(hm|+pc,-vu,-dt,+tm) = 0.09
P(hm|+pc,-vu,-dt,-tm) = 0.10
P(hm|-pc,+vu,+dt,+tm) = 0.31
P(hm|-pc,+vu,+dt,-tm) = 0.33
P(hm|-pc,+vu,-dt,+tm) = 0.30
P(hm|-pc,+vu,-dt,-tm) = 0.31
P(hm|-pc,-vu,+dt,+tm) = 0.31
P(hm|-pc,-vu,+dt,-tm) = 0.32
P(hm|-pc,-vu,-dt,+tm) = 0.29
P(hm|-pc,-vu,-dt,-tm) = 0.30
```

**Figure 2.5:2** - Node conditional probabilities (NCPs) for the example BN in figure 2.5:1.

This BN can be used to find Jim's initial beliefs in each proposition, and what those beliefs become if he finds out the truth value of one or more of the nodes. For example, if Jim found out that Tom's hardware store did well last year, we can use it to find his new beliefs in each proposition (for example, whether Hank is going to move away in the next 5 years). Then, if he later read in the newspaper that Molly was elected mayor, we could obtain a new set of beliefs for each node (for example, whether Tom told Molly he will donate campaign funds). The actual beliefs calculated for this example are given at the end of the next section.

## 2.6 Bayesian Network Inference

The most studied BN inference problem is: Given a BN and evidence for some of its nodes, what are the posterior probabilities of its other nodes? *Evidence* is some ideal observation, or the receiving of some certain information, on the truth of one or more node propositions (uncertain evidence is dealt with in section 2.7). With respect to a particular BN, evidence items may be inconsistent with each other. For example, if B is a child of A with P(b|a) = 1, and we obtain evidence a=TRUE and b=FALSE, we say the evidence is inconsistent. This indicates that the evidence is not possible given the BN model (which often indicates a fault with the model, not the evidence). Evidence will never be inconsistent for a BN which does not have any zeros in its NCPs. Inconsistent evidence is not allowed in BN inference, and throughout this thesis, whether it is explicitly stated or not, we assume evidence is consistent.

Occasionally in this thesis I will refer to BN inference in a dynamic sense, which implies a situation where a stream of evidence items constantly arrive to a BN, and we update the beliefs of each node as they arrive. So we may speak of the belief at a node as rising and falling through time. A primary feature of this situation is that evidence always accumulates, no item is ever retracted. As the amount of evidence monotonically increases, some quantities of interest will monotonically increase (or decrease), while others will vary up and down.

Once we have received evidence for a BN, we can use one of two different systems for taking it into account. We can do *evidence absorption* to modify the BN to one specific for that evidence state, by modifying the network topology and NCPs (often extensively), so that the nodes for which we have evidence no longer appear in the network, and the new network represents the original BN conditioned on the evidence. Or we can use the system of *belief updating*, which leaves the BN structure unchanged, but marks the nodes for which we have evidence with the state of the evidence (these become known as *evidence nodes* and are usually drawn shaded on a BN diagram), and then uses algorithms designed to handle the evidence "in place" to find posterior probabilities. These two systems of dealing with evidence are entirely equivalent in semantics; the only differences are ones of representation and computation.

The normal way of creating a new BN by evidence absorption is through link reversals and node absorption (Shachter86, Shachter88). First, all links pointing to evidence nodes are reversed, one by one, using a system based on Bayes rule, which modifies the NCPs at (only) the two nodes at each end of the link. Each reversal may result in new links pointing to the evidence node, since during a reversal the nodes at each end of the link gain all the parents the other has. Eventually though, each evidence node is guaranteed to have only links leaving it (links between evidence nodes may simply be deleted), and then that evidence node is absorbed. That is, it is removed from the network and all the NCPs of its child nodes are collapsed by one dimension to the value of the evidence. If it is desired to remove a node which does *not* have evidence, we reverse links so that it has only links *entering* it, and none leaving it, and then we just delete the node. Later a process called *probability propagation* may be used on this new BN to calculate the posterior probabilities (i.e. belief at each node).

There are a number of methods for belief updating, which computes the new beliefs after receiving evidence without creating a new BN. Generally, they take much better advantage of independencies than evidence absorption, and so require less computation. Some of them use a compiled *secondary structure* for efficiency, and some of them operate on just the original BN. Some of them produce approximate results, and some exact. Pearl developed a fast algorithm for BN updating when the network is singly-connected called *belief propagation* (Pearl88), which can find posteriors in O(N) time, where N is the number of nodes in the network, and as a parallel algorithm with a processor for each node, in O(d) time, in which d is the diameter of the network. Multiply-connected networks pose a much greater computational problem.

The *reasoning by assumptions* algorithm (Pearl88) finds a set of nodes (called cut nodes), such that if they were instantiated with evidence some active paths would become blocked and the network would become singly-connected. Then it solves the singly-connected problem multiple times, once for each possible instantiation of the cut nodes. The final beliefs are found as a weighted sum of the beliefs in each of these sub-calculations, with the weighting for each instantiation being the probability that the cut nodes would be instantiated that way (using equation 2.3:3). Since all combinations of evidence at the cut nodes must be considered, the algorithm is exponential in the number of cut nodes. It is not very efficient, but is mentioned here for its conceptual value.

Some of the most efficient and popular exact algorithms currently known are the *clique tree* algorithms (Lauritzen&Spiegelhalter88), especially those using a *junction tree* (Jensen&OA90). They create a secondary structure which is a singly-connected graph in which each node corresponds to the Cartesian product of a few nodes from the original BN. Evidence propagation may be accomplished by message passing between the nodes of this new tree. The computational complexity depends very much on the state-space size of the new nodes, which in turn depends on the connectivity of the original BN.

Cooper has shown general BN inference to be NP-hard by reducing the 3SAT problem to a BN inference problem (Cooper90). This is a worst-case result, but often even the average case requires exponential time to find exact results. So for large BN applications, some sort of approximation algorithm is often necessary for BN inference.

Although the BN inference algorithms described above will find the posterior probabilities for any node, given a set of evidence at any other nodes, it is useful to dissect BN inference into *predictive reasoning*, *diagnostic reasoning*, and *intercausal reasoning* ("intercausal" is derived from HenrionDruzdel90). Predictive reasoning finds the belief at a node which is a descendent of a node with evidence, and diagnostic reasoning finds the belief at an ancestor of an evidence node. Intercausal reasoning propagates the effect of evidence between common parents of a node with evidence. Figure 2.6:1 illustrates the three types.

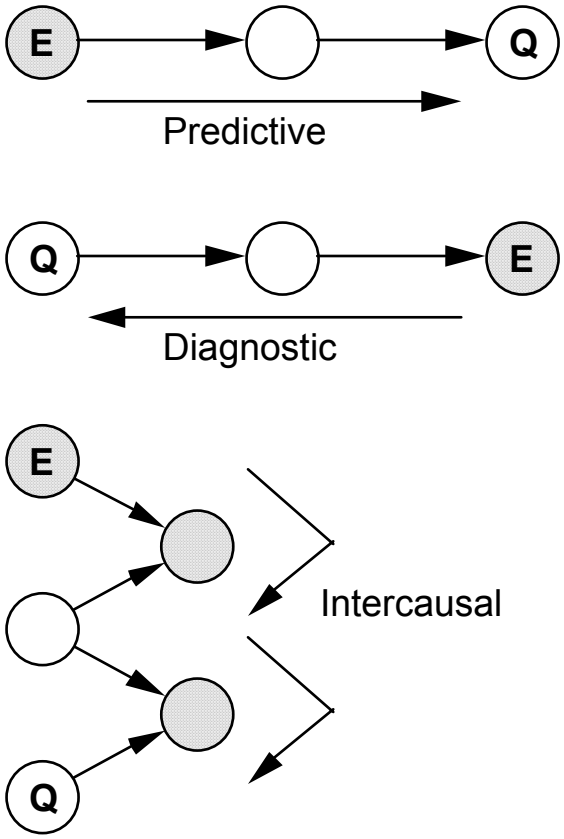


**Figure 2.6:1** - The three types of reasoning in BN inference. The shaded nodes are nodes with evidence. In each case we wish to find the change in belief at node Q due to evidence at node E.

Any of these three types of reasoning may be combined for more complex inference. Although BN inference is not normally subdivided along the paths from an evidence node to a *query node* (i.e. a node whose updated belief we wish to find), it is sometimes useful to do so. Suermondt92 does this to generate explanations of BN inference, and I will do it to study BN sensitivity. Figure 2.6:2 shows the paths of reasoning from node V to query node Q, given that some previous evidence has arrived to the network at the shaded nodes.

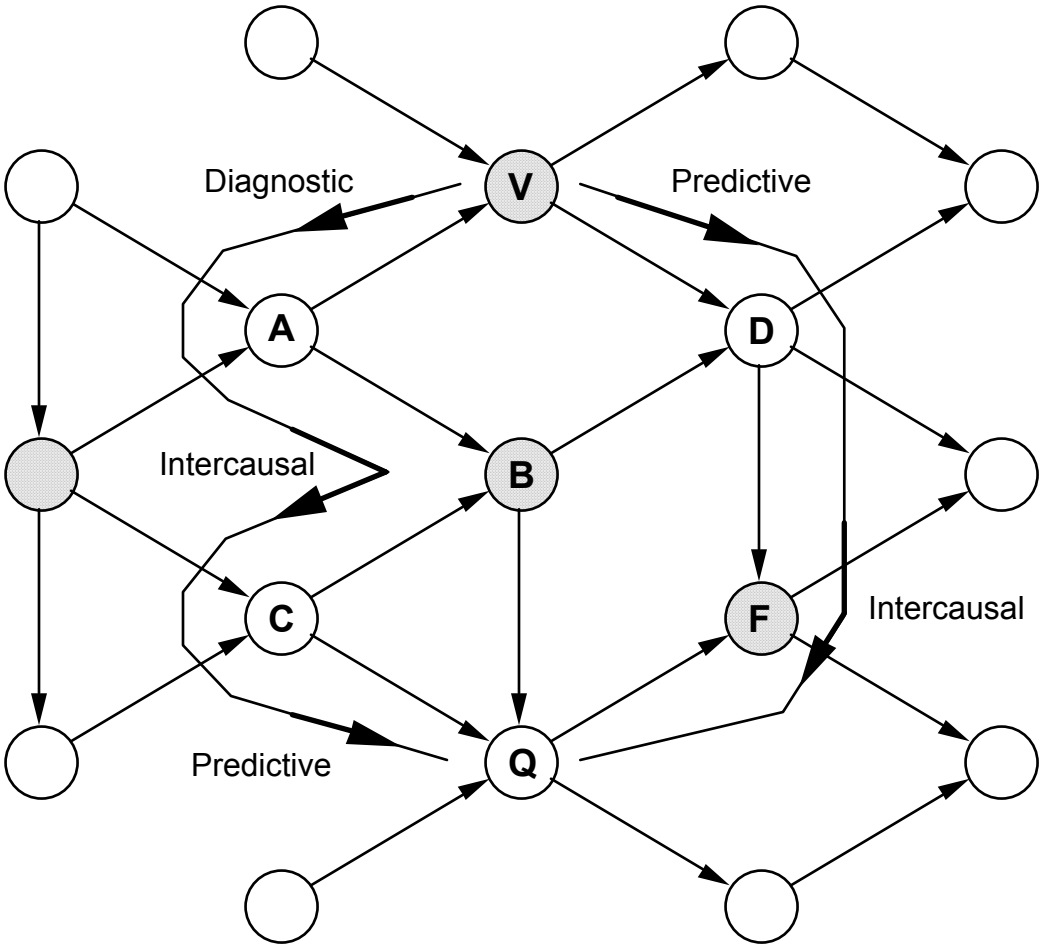


**Figure 2.6:2** - Combination of different types of reasoning. There are only two active paths from V to Q: the path V,A,B,C,Q and the path V,D,F,Q. The first path contains diagnostic, intercausal, and predictive reasoning, while the second contains only predictive and intercausal reasoning. The shaded nodes are nodes with evidence, V is a varying node (or node with new evidence), and Q is a query node.

### 2.6.1 Virtual Evidence

In cases where we obtain evidence for a node, but the evidence is not certain, we can make use of the concept of *virtual evidence* to handle it using the regular machinery of BN updating (Pearl88). For instance, suppose we wanted to process uncertain information that node A is true. We make a new node that is a child of node A and call it $A_v$. To add this node we must supply two new probabilities: the probability that $A_v$ is TRUE given that A is TRUE, and the probability that $A_v$ is TRUE given that A is FALSE. These two probabilities measure the degree of uncertainty of the evidence. If one of them is 0, then we have the limiting case of the evidence

being certain; We don't allow them both to be 0, which would be inconsistent. To do updating with the virtual evidence, node $A_V$ is marked as a node with certain evidence of TRUE, and regular BN updating is used to adjust the beliefs of the rest of the nodes in the network.

We may have several independent pieces of uncertain evidence for the node A, which may conflict or agree with each other. In this case we simply add a new child node, and its two probabilities, for each piece of evidence (see figure 2.6.1). Then regular BN updating will handle finding the belief for node A and the rest of the nodes in the network whether the evidence items conflict with each other or support each other.

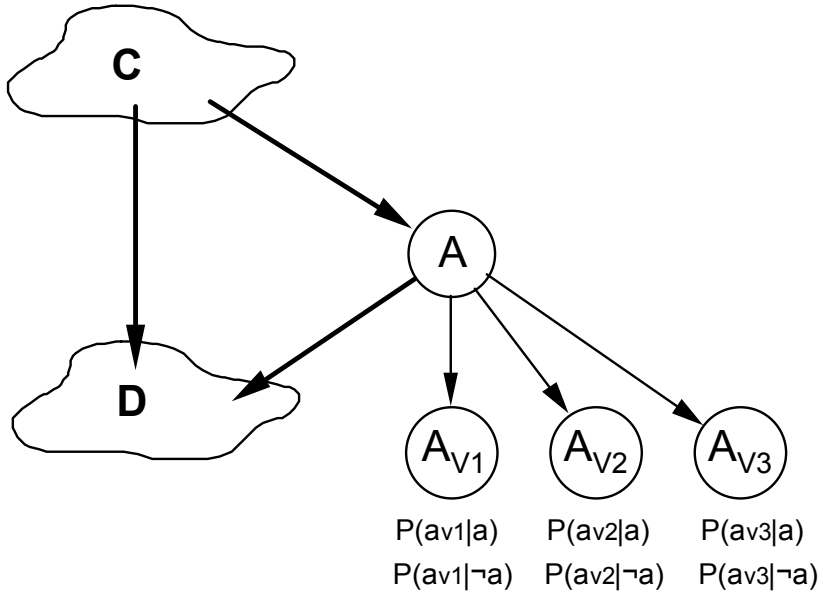


**Figure 2.6.1** - Three items of virtual evidence for node A are represented as three child nodes. The two parameters for each child specify the certainty of the evidence. The subnetworks C and D simply illustrate that A is part of a larger BN.

### 2.6.2 Inference on BN Example

We can use any one of the algorithms described earlier to find beliefs for the example BN given in the last section. Initially Jim does not know the truth value of any of the nodes in the example. By doing probability propagation we can find his belief in any node:

| | | |
|---|---|---|
| P(sw) = 0.670 | P(tf) = 0.160 | P(gt) = 0.0695 |
| P(mm) = 0.179 | P(pc) = 0.40821 | P(hm) = 0.2304 |

If he finds out that Tom's hardware store did well last year, then we can find his new beliefs for each node by doing evidence updating:

| | | |
|---|---|---|
| P(sw\|sw) = 1 | P(tf\|sw) = 0.184 | P(gt\|sw) = 0.0777 |
| P(mm\|sw) = 0.183 | P(pc\|sw) = 0.409 | P(hm\|sw) = 0.2302 |

If he reads in the newspaper that Molly was elected mayor (and believes it with certainty), then once again we can find his new beliefs for each node by doing evidence updating:

| | | |
|---|---|---|
| P(sw\|sw,mm) = 1 | P(tf\|sw,mm) = 0.286 | P(gt\|sw,mm) = 0.0778 |
| P(mm\|sw,mm) = 1 | P(pc\|sw,mm) = 0.630 | P(hm\|sw,mm) = 0.188 |

It may seem strange that finding out Tom's hardware store did well would change Jim's belief in something so distantly related as whether Hank was going to move or not (even though it only changed from 0.2304 to 0.2302). Whenever there is an active path from one node of a BN to another, evidence at one of the nodes can change the belief at the other node. In a very large BN, it is quite reasonable for every node to be connected to every other node, with many of the connections being active paths. Knowledge of different subject areas may be connected along their boundaries. In the example BN, the links between the nodes were quite natural, and it would be natural for there to be a link from "Tom's hardware store did well" to a node called "Tom opens grocery store" to "Tom eats more lettuce" to, etc., providing nodes ever more weakly connected to "Hank moves", yet if Jim received evidence for any one of these nodes, his belief in all of them will change somewhat. The point is that the beliefs at very distant nodes will change almost imperceptibly, and if we have a way of measuring what that change will be, or of guaranteeing that it is less than some bound, we may want to ignore finding new beliefs for those distant nodes, thereby greatly simplifying the computational burden.

# 3 Connection and Link Strengths

## 3.1 Connection Strength Definition

Given two nodes, A and B, in a propositional Bayesian network, the *connection strength* (CS) from node A to node B is defined as the difference in the resulting belief at node B, between the situation where A receives evidence TRUE, and the situation where A receives evidence FALSE. Formally:

$$CS(A,B) = d(P(b|+a),\ P(b|\neg a)) \qquad 3.1{:}1$$

where d is a distance measure to determine the amount of change, or degree of difference, between two probabilities (examples will be given), which for all $x, y, z \in [0,1]$, satisfies:

1. Zero: $d(x, x) = 0$ 3.1:2
2. Symmetry: $d(x, y) = d(y, x)$
3. Triangle Inequality: $d(x, z) \leq d(x, y) + d(y, z)$
4. Monotonicity: For any given x, d(x, y) increases monotonically as y moves away from x. Formally: $x \leq y \leq z \Rightarrow d(x, y) \leq d(x, z)$ and $d(y, z) \leq d(x, z)$
5. No-maxima: Along any line segment in (X,Y), d(x,y) reaches its maximum at an endpoint. Formally: $d(\lambda a + (1-\lambda)b, \lambda c + (1-\lambda)d) \leq \max(d(a,c), d(b,d))$ for all $a,b,c,d,\lambda \in [0,1]$.

The first three conditions are the well known Hausdorff postulates of many distance semi-metrics, and are the requirements of a *probability metric* as defined in Zolotarev83. There has been some research on probability metrics (see Zolotarev83) but it appears to be mostly

concerned with distributions defined over continuous variables. Actually the zero and symmetry requirements are not needed for the crucial proofs. Without symmetry, all theorems except 3.1 and the lower bound of 3.4:3 are valid. The "zero" requirement may be removed, but it is very useful, since terms that equal zero are dropped from equations in the path based methods of section 4.4 and beyond. A further condition on d which is recommended, based on the semantics of probability, is: Invertability: $d(x, y) = d(1 - x, 1 - y)$. Specific distance measures will be discussed later.

### 3.1.1 Connection Strength and Virtual Evidence

CS(A,B) was defined as the difference in belief at B as the belief in A switched from true to false (due to evidence at A). But what if the belief at A changed from partway between certainly-true and certainly-false to somewhere else between true and false? How much would the belief at B change? If the change in belief at A was due to evidence at nodes that are independent of B given A, we can guarantee that the change in belief at B will be less than (or equal) to CS(A,B) as defined by equation 3.1:1. An example of evidence that is independent of B given A, is virtual evidence for the node A. So, as various items of virtual evidence arrive for node A, the belief in A will vary between true and false, and this will cause the belief in B to vary somewhere between true and false, but the maximum variation at B will always be less than (or equal) CS(A,B). Formally:

**Theorem 3.1**: Equation 3.1:1 defining connection strength is equivalent to:

$$CS(A,B) = \max_{a_{v1}, a_{v2}} d(P(b|a_{v1}), P(b|a_{v2})) \qquad 3.1:3$$

and:

$$CS(A,B) = \sup_{a_{v1}, a_{v2}} d(P(b|a_{v1}), P(b|a_{v1}, a_{v2})) \qquad 3.1:4$$

where $a_{v1}$ is some virtual (or nonexistent) evidence for A, $a_{v2}$ is other evidence (possibly virtual) for A, "sup" means the least upper bound, and the distance measure d is assumed continuous for 3.1:4. The equivalence of 3.1:1, 3.1:3, and 3.1:4 is proved in Appendix C.

These equations allow a slightly different defining statement for connection strength. CS(A,B) is a measure of the most the belief in B could be changed by receiving new (possibly virtual) evidence at A, whatever that evidence is, and whatever other (possibly virtual) evidence has already been received at A.

## 3.2 ΔP Connection Strength

To completely define connection strength we must supply a probability distance measure d. One possibility which satisfies the requirements (3.1:2) is simply the difference function:

$$d_p(P_1, P_2) = |P_1 - P_2| \qquad 3.2:1$$

This definition is a degenerate case (applied to a binary variable, rather than a mulitstate or continuous variable) of what is called the Kolmogorov metric, also known as the uniform metric or variation norm (see Zolotarev83).

Connection strength defined using this distance measure for probabilities will be called ΔP connection strength, and denoted $CS_p$.

### 3.2.1 Single Link Example

First we consider a very simple BN consisting of only two binary nodes and a single link between them:

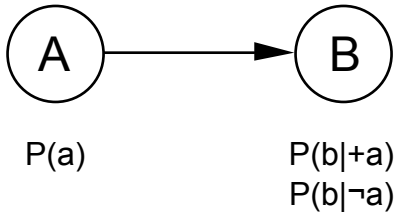


**Figure 3.2** - Simple BN example.

Three numerical parameters are needed to define this BN: P(a), P(b|+a) and P(b|¬a). As an example we take P(a) = 0.3, P(b|+a) = 0.5 and P(b|¬a) = 0.75.

If evidence TRUE is observed at A, then the belief at B (by this we mean the probability that B is TRUE given the evidence) will be P(b|+a), and if evidence FALSE is observed then it will be P(b|¬a). Using equation 3.1.4 for connection strength, and the distance measure of equation 3.2:1, for this simple BN we obtain:

$$CS_p\,(A,B) = |\, P(b|+a) - P(b|\neg a)\,| \qquad 3.2{:}2$$

This is a measure of "the most node A can affect the belief at node B". As a visualization aid one can imagine a situation in which a steady stream of virtual evidence is arriving for A, but no evidence arrives for B. As each piece of evidence arrives, our belief in A will change, and as a consequence our belief in B will also change. In general, the belief at B will rise and fall, but will be restricted within a certain envelope. $CS_p$ measures the maximum width of this envelope for any series of virtual evidence items at A. For the example parameters, $CS_p\,(A, B) = |\, 0.75 - 0.5\,| = 0.25$, so we would say the connection strength from A to B was 0.25.

### 3.2.2 Range of ΔP Connection Strengths

Consider a BN in which P(b|+a) = P(b|¬a). In that case, whether evidence of TRUE, evidence of FALSE, or no evidence, is observed at A, the belief at B remains constant at P(b|+a) = P(b|¬a) = P(b), providing no evidence is observed for B. Also, whatever evidence is observed at B, providing there is no evidence observed for A, the belief at A will remain constant at P(a). So B is actually independent of A, and normally the BN would be drawn without a link connecting the two nodes. The connection strength in this case is $CS_p(A,B) = 0$. ΔP connection strength between two nodes is zero if and only if the two nodes are independent of each other.

Consider a BN in which P(b|+a) = 1, and P(b|¬a) = 0. Then, observing evidence TRUE at A results in a belief of 1 at B (i.e., the knowledge that B is TRUE). Observing evidence FALSE at A results in the knowledge that B is FALSE. In this case the connection strength is 1. ΔP connection strength is 1 if and only if direct evidence for the first node deterministicly defines the value of the second.

ΔP connection strength varies from 0 for the weakest connections (actually independence) to 1 for the strongest connections (deterministic dependence):

$$0 \leq CS_p(X, Y) \leq 1 \qquad 3.2{:}3$$

## 3.3 ΔO Connection Strength

We have defined a connection strength based on the absolute difference of probabilities, $d(P_1, P_2) = |P_1 - P_2|$. But sometimes an absolute difference of probabilities does not capture what we want as a measure of the "distance" between two probabilities. For example, suppose we are calculating a probability approximately, and we want to measure how close our estimation, P*, is to the value calculated exactly, P. We will call the distance between these two probabilities "the error" of the estimation, $e = d(P^*, P)$. We may want to specify an upper bound $e_m$ on this error, and look for an algorithm which is guaranteed to calculate the estimate with an error less than this bound. This is an application of connection strength that we will be considering.

We can use the absolute difference function as a distance measure for calculating an error in probabilities, but sometimes this will lead to unsatisfactory results. For example, suppose an approximate algorithm estimated the probability for rain tomorrow as 0.61, when an exact algorithm with the same information would have calculated the probability to be 0.60. We would likely consider the approximate algorithm to have performed well, and the action that we take based on its estimate will not be misguided. However, if the approximate algorithm estimated the probability for severe flood each month in some area as 0.0001, when the exact algorithm yielded 0.01, we would say the approximate algorithm failed, and the action we took based on its result – building a house which will probably be destroyed in about 10 years – was misguided.

In both cases the absolute difference between the estimate and the exact value is about 0.01, but that difference is more significant in the case of the smaller probabilities (of course it is because the maximum utility values are larger in the small probability example, but that is often the case). It seems that what we need for this problem is a relative measure of error.

One might imagine using a "percent difference" distance measure, such as $d(P_1, P_2) = |P_1 - P_2| / P_1$. Then the error of the approximate algorithm estimating the probability for rain would be only 1.7%, whereas that of flood probability would be 99%, which nicely distinguishes

between the two cases. However it suffers from a couple of problems. The proposition whose probability we are estimating may just as well have been defined in the logical inverse, Q' = ¬Q, so that the probability associated with it becomes one minus what it would otherwise be, P(Q') = 1 – P(Q). Therefore, the distance measure should treat probabilities which are close to 1 in the same way that it treats probabilities close to 0, i.e. $d(P_1, P_2) = d(1 - P_1, 1 - P_2)$. For example, the approximate algorithm estimated the probability for no flood at 0.9999, while the exact algorithm yielded 0.9900, which is a percent difference of only 1%.

One way to deal with this would be to use the percent difference in P for probabilities less than 0.5, and use the percent difference in 1 – P for probabilities greater than 0.5. However, this is somewhat messy, especially when measuring differences between probabilities which straddle 0.5. A more elegant solution is to use the percent difference of odds ratios (there are also other important reasons to use odds ratio which will be discussed later). The *odds ratio* of proposition A is defined as the probability that A is TRUE, divided by the probability A is FALSE:

$$O(a) = \frac{P(a)}{P(\neg a)} = \frac{P(a)}{1 - P(a)} \qquad 3.3{:}1$$

$$O(b|a) = \frac{P(b|a)}{P(\neg b|a)} = \frac{P(b|a)}{1 - P(b|a)}$$

Odds ratios apply only to propositions, but occasionally we will loosely refer to the odds ratio of a probability P, by which we mean the quantity $P / (1 - P)$.

As two probabilities, $P_1$ and $P_2$, approach 0, the percent difference in the odds ratios of $P_1$ and $P_2$ approaches the percent difference of $P_1$ and $P_2$, and as $P_1$ and $P_2$ approach 1, the percent difference in the odds ratios of $P_1$ and $P_2$ approaches the percent difference of $1 - P_1$ and $1 - P_2$. Percent difference of odds ratios satisfies $d(P_1, P_2) = d(1 - P_1, 1 - P_2)$, and is numerically close to percent difference of probabilities for small probabilities.

We need to make another refinement to our relative distance measure. Instead of a percent difference we use a *factor* difference, which is the ratio of the two quantities. For example, a 5% difference corresponds to a factor difference of 1.05. The advantage of this is symmetry. If quantity $X_1$ changes by 5% in going to $X_2$, it doesn't change by –5% going from $X_2$ to $X_1$ (it

changes by 1/1.05 – 1 = –4.76 %). However, if $X_1$ changes by a factor of 1.05 in going to $X_2$, then $X_2$ changes by a factor of 1/1.05 going to $X_1$. To define a distance measure we are interested only in the magnitude of the change in going from one quantity to the other, not its direction, so if a factor difference is less than 1, we use its inverse instead. We denote the factor difference between $P_1$ and $P_2$ as $\|P_1/P_2\|$, where the double vertical bars are similar to the vertical bars of absolute value (since $\|x\| = \max(x, 1/x)$ is to multiplication what $|x| = \max(x, -x)$ is to addition).

This gives us a proposed relative distance measure for two probabilities $P_1$ and $P_2$ as:

$$d_c(P_1, P_2) = \left\| \frac{P_1}{1-P_1} \Big/ \frac{P_2}{1-P_2} \right\| = \left\| \frac{P_1(1-P_2)}{P_2(1-P_1)} \right\| \qquad 3.3{:}2$$

Finally, we take the logarithm of $d_c$ to provide a true distance measure, which we will denote as $d_o$. $d_o$ and $d_c$ values can always be recovered from each other, and $d_o$ satisfies 3.1:2, so it is a suitable distance measure ($d_c$ does not satisfy the zero condition or the triangle inequality condition). $d_o$ turns out to be the absolute difference of log odds ratios:

$$d_o(P_1, P_2) = |\log d_c(P_1, P_2)| = \left| \log \frac{P_1}{1-P_1} - \log \frac{P_2}{1-P_2} \right| \qquad 3.3{:}3$$

Absolute difference of log odds ratio has been in widespread use for variety of purposes. The reason for the long discussion in getting to here was to show some of the reasons it is better than some of the alternatives.

Any base of logarithm may be used, as long as the usage is consistent. This thesis assumes that natural logarithms are used, but makes a note on how an equation or result should modified to accommodate some other logarithm base in the few cases where this is necessary. To get a feel for natural logarithm $d_o$ values, when they are small, they are close to percent values. This is because for small x, $\log_e(x) \approx 1 + x$. Thus, a $d_o$ of 0.05 corresponds to approximately a 5.1% difference in odds ratio. If the probabilities are small, this in turn corresponds to approximately a 5.1% difference in probability.

If one of the probabilities $P_1$ or $P_2$ is 0 or 1, the denominator of a fraction in 3.3:3 will become 0, or we will end up with log 0, which is also undefined. Throughout this thesis I use arithmetic defined on the real numbers augmented by infinity, i.e. the set $\Re \cup \{\infty\}$. The following rules are observed:

$x / 0 = \infty$, for all $x > 0$, $\qquad x / \infty = 0$, for all $x \neq \infty$,

$0 / 0$, $\infty / \infty$, $\infty - \infty$, and $\infty * 0$ are undefined,

and in general for a function f(x), we define $f(\infty)$ as $\lim_{x \to \infty} f(x)$

Furthermore, the $d_o$ measure between equal probabilities is defined to always be 0, even if the probabilities are both 0 or both 1. With these refinements we can express $d_o$ as:

$$d_o(P_1, P_2) = \begin{cases} 0 & P_1 = P_2 \\ \infty & P_1=0, P_2=1 \text{ or } P_1=1, P_2=0 \\ \left| \log \frac{P_1}{1 - P_1} - \log \frac{P_2}{1 - P_2} \right| & \text{otherwise} \end{cases} \qquad 3.3:4$$

The definition of $d_o$ satisfies the requirements of a probability metric as described at the beginning of section 3.2. Connection strength defined using the $d_o$ distance measure for probabilities (i.e. the absolute difference of log odds ratios) will be called $\Delta O$ connection strength, and denoted $CS_o$.

### 3.3.1 Single Link Example

We return to the example of figure 3.2 to calculate the $\Delta O$ connection strength from A to B. In section 3.2 we found $CS_p(A, B) = 0.25$. By equations 3.3.4 and 3.1:4 we find:

$$CS_o(A, B) = \left| \log \frac{P(b|a)}{1 - P(b|a)} - \log \frac{P(b|\neg a)}{1 - P(b|\neg a)} \right| \qquad 3.3:5$$

$$CS_o(A, B) = \left| \log \frac{0.75}{0.25} - \log \frac{0.5}{0.5} \right| \approx 1.10$$

### 3.3.2 Range of ΔO Connection Strengths

ΔO connection strength varies from 0 for the weakest connections (actually independence) to ∞ for the strongest connections (potentially deterministic dependence):

$$0 \leq CS_o(X, Y) \leq \infty \qquad 3.3{:}6$$

Note that if $CS_o(X, Y) = \infty$, Y may be deterministic for only one value of X (such as in the case: given that X is true, Y is true, but given X is false, Y is uncertain). This differs from ΔP connection strength, where a strength of 1 indicates complete deterministic dependency.

Although log odds ratio is a monotonic function of probability, ΔO connection strength is not a monotonic function of ΔP connection strength. Two nodes with a weak ΔP connection strength ($CS_p$ close to 0) may have a strong ΔO connection strength ($CS_o$ very large). More precisely, for any two nodes X and Y, in any BN:

$$2 \log \frac{1 + CS_p(X, Y)}{1 - CS_p(X, Y)} \leq CS_o(X, Y) \leq \infty \qquad 3.3{:}7$$

where the $CS_o$ values can be anywhere in the range, depending on the particular BN. Conversely:

$$0 \leq CS_p(X, Y) \leq \frac{e^{CSo(X, Y)/2} - 1}{e^{CSo(X, Y)/2} + 1} \qquad 3.3{:}8$$

If $CS_o$ was defined using a logarithm of some base other than e, then in the equation above, e should be substituted with the actual base used.

For small values of $CS_p$ we can make the approximation:

$$CS_o \approx 4\, CS_p \qquad \text{for small } CS_p \qquad 3.3{:}9$$

$$CS_o \geq 4\, CS_p \qquad \text{for all } CS_p$$

## 3.4 An Alternate Definition of Connection Strength

The connection strength from node A to node B could have been defined as: The maximum change in belief at B as we go from a state of no evidence at A to a state with some evidence at A, that is:

$$CS'(A,B) = \max\ (d(P(b), P(b|+a)),\ d(P(b), P(b|\neg a))) \qquad 3.4{:}1$$

which is equivalent to:

$$CS'(A,B) = \max_{a_{v1}} d(P(b),\ P(b|a_{v1})) \qquad 3.4{:}2$$

where $a_{v1}$ is virtual evidence for A.

This quantity can be bounded above and below using the original definition of connection strength, and the original definition is more workable in most situations, so it is preferred.

**Theorem 3.4**: For any two propositional variables A and B, and for any evidence **e** (or no evidence **e**), connection strength defined by equation 3.4:1 can be bounded above and below as:

$$\frac{1}{2}\ CS(A,B|\mathbf{e}) \leq CS'(A,B|\mathbf{e}) \leq CS(A,B|\mathbf{e}) \qquad 3.4{:}3$$

This is proved in Appendix C.

## 3.5 Conditional Strength and Maximal Strength

In a situation where a stream of evidence arrives for a BN (items of evidence arrive sequentially through time), it is sometimes useful to consider connection strength dynamically. The *conditional connection strength* CS (A, B|**e**) is a measure of the maximum effect on node B of evidence at A given the (possibly virtual) evidence **e** that has already arrived to other nodes of the network, and no other evidence. More formally:

$$CS(A,B|\mathbf{e}) = d(P(b|+a, \mathbf{e}),\ P(b|\neg a, \mathbf{e})) \qquad 3.5{:}1$$

$$CS(A,B|\mathbf{e}) = \sup_{a_{v1} \cdot a_{v2}} d(P(b|a_{v1}, \mathbf{e}),\ P(b|a_{v1}, a_{v2}, \mathbf{e})) \qquad 3.5{:}2$$

where **e** is the evidence seen thus far, and, as with the CS definition, $a_{v1}$ is some (possibly virtual) evidence for A, $a_{v2}$ is some other consistent evidence (possibly virtual) for A, and d is a continuous distance measure satisfying 3.1:2. Equation 3.5:2 is equivalent to 3.5:1 (the proof is similar to that of theorem 3.1).

We define *maximal* CS as the maximum value that conditional CS may take upon receiving further evidence consistent with the evidence already received, and denote it as CSM:

$$CSM(A,B|\mathbf{e}) = \max_{\mathbf{e}_+ \approx \mathbf{e}} CS(A,B|\mathbf{e},\mathbf{e}_+) \qquad 3.5{:}4$$

where **e** is the evidence seen thus far and $\mathbf{e}_+$ is evidence consistent with **e** (the ≈ symbol means "consistent with").

Conditional strength may increase or decrease upon gathering more evidence, but maximal strength always remains constant or decreases. At all points in time conditional CS is always less than or equal to maximal CS. The following are some easily proved relationships, which are true for all nodes A and B, and consistent evidence **e** and $\mathbf{e}_+$, in any BN:

$$CS(A,B|\mathbf{e}) \leq CSM(A,B|\mathbf{e}) \leq CSM(A,B) \qquad 3.5{:}5$$

$$CSM(A,B|\mathbf{e}, \mathbf{e}_+) \leq CSM(A,B|\mathbf{e}) \qquad 3.5{:}6$$

$$\max_{\mathbf{e}} CS(A,B|\mathbf{e}) = \max_{\mathbf{e}} CSM(A,B|\mathbf{e}) = CSM(A,B) \qquad 3.5{:}7$$

$$CSM(A,B|\mathbf{e}) = \sup_{a_{v1}, a_{v2}, \mathbf{e}_+ \approx \mathbf{e}} d(P(b|a_{v1}, \mathbf{e}, \mathbf{e}_+),\ P(b|a_{v1}, a_{v2}, \mathbf{e}, \mathbf{e}_+)) \qquad 3.5{:}8$$

## 3.6 Connection Strength in Complex BNs

I gave an example of calculating connection strength in a simple two node BN, but how do we find the connection strength between the nodes A and B in a complex BN, where there are

multiple paths between A and B, and each path consists of multiple links? Connection strength may be computed from its definition (3.5:1) by the methods of BN inference:

$$CS(A,B|\mathbf{e}) = d\,(P(b|+a,\mathbf{e}),\ P(b|\neg a,\mathbf{e}))$$

We instantiate the BN with evidence **e** and +a, and then using the techniques of BN updating discussed in section 2.6, we find the posterior probability $P(b|+a,\mathbf{e})$. We repeat this, but instead with evidence **e** and ¬a, to find $P(b|\neg a,\mathbf{e})$, and then use the above equation to find $CS(A,B|\mathbf{e})$.

Since BN inference can be NP-hard (Cooper90), calculating CS in this manner can be quite expensive. Alternatively, we can find bounds on CS quickly by local calculations using the concept of link strength. This will be explored in Chapter 4.

## 3.7 Commutivity of Connection Strength

Given a value for the ΔP connection strength from node A to node B, can we use only this information to determine what the connection strength from node B to node A is? It turns out that the ΔP CS in one direction does not even constrain the ΔP CS in the reverse direction (unless it is 0 or 1, in which case the CS in the reverse direction will be the same). More precisely, for any value of $CS_p(A,B)$ in (0,1), and any value $CS_p(B,A)$ in (0,1), there is some BN with these connection strengths.

However, the ΔO connection strength in one direction always equals the strength in the reverse direction. In other words:

**Theorem 3.7**: ΔO connection strength is commutative (proved in Appendix C):

$$CS_o(A, B) = CS_o(B, A) \qquad 3.7{:}2$$

This means that for any two nodes A and B, in any BN, the maximum amount that evidence at node B can effect the belief at node A, is the same as the maximum amount that evidence at node A can effect the belief at node B, provided "effect" is measured by $d_o$. This is a very useful result and later we will see how it gives $CS_o$ significant advantages over $CS_p$. Conditional and

maximal connection strength based on odds ratio is also commutative (these are proved in Appendix C):

$$CS_o(A, B \mid \mathbf{e}) = CS_o(B, A \mid \mathbf{e}) \qquad 3.7:3$$

$$CSM_o(A, B \mid \mathbf{e}) = CSM_o(B, A \mid \mathbf{e}) \qquad 3.7:4$$

As a stream of evidence is obtained for a network, $CS_o(B,A|\mathbf{e})$ may change, but $CS_o(A,B|\mathbf{e})$ will also change so that they will remain equal. In extreme cases the new evidence may d-separate A and B (so $CS_o$ goes from some number to 0), or it may connect A and B with an active path when they were d-separated (so it goes from 0 to some larger number, possibly even infinity). In any case $CS_o$ will remain commutative.

## 3.8 Link Strength Definition

For each link in a BN we can supply a single number, called the *link strength* (LS), which is a measure of the maximum amount that evidence at the parent node of the link can effect the belief at the child, given that all the other parents of the child have some evidence. So the link strength of $A \rightarrow B$ is defined as:

$$LS(A \rightarrow B) = \max_{\mathbf{c} \in \prod(C(B) - \{A\})} CS(A, B|\mathbf{c}) \qquad 3.8:1$$

where C(B) is the set of parents of B. The strength of the $A \rightarrow B$ link in figure 3.8 is the maximum value of connection strength from A to B as all the parents $C_1, C_2, ..., C_n$ take on various evidence. Computing the LS from equation 3.8:1 can be done completely locally; it requires knowledge only of the NCP at B.

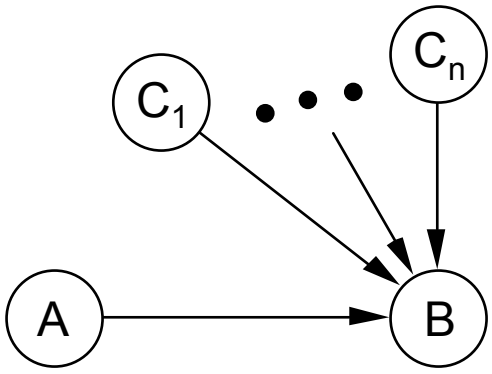


**Figure 3.8** - These nodes may be embedded in a larger BN. The only nodes shown are B and all its parents, since these are the nodes required to define the link strength of the link from A to B.

Given its parents, B is independent of the rest of its ancestors, so equation 3.8:1 is equivalent to:

$$LS(A \rightarrow B) \;=\; \max_{\mathbf{a} \in \prod(C^{+}(B) - \{A\})} CS(A, B|\mathbf{a}) \qquad 3.8:2$$

where $C^{+}(B)$ is the set of all ancestors of B.

We can still speak of an $A \rightarrow B$ link strength, even if there is no link from A to B (that is, A is not a parent of B). If A is a descendent of B, then adding a link from A to B would create a cycle, and so in that case $LS(A \rightarrow B)$ is undefined. But, if A is not a descendent of B, and is not a parent of B, it will be independent of B given B's parents, so by equation 3.8:1, $LS(A \rightarrow B)$ is zero. In this case we sometimes say there is a *null link* from A to B, although of course it wouldn't count as a link to the d-separation algorithm, and normally we wouldn't draw it on a BN diagram.

The range of conceivable LS values is the same as that of CS, so $LS_p$ values vary from 0 for null links to 1 for the strongest links (potentially deterministic dependence), and $LS_o$ values vary from 0 for null links to infinity for the strongest links:

$$0 \le LS_p \le 1 \qquad\qquad 0 \le LS_o \le \infty \qquad 3.8:3$$

Recalling the algorithm described in section 2.4 used to construct the dag of a BN, we can view it slightly differently now. We are given a FJD and we wish to construct a BN to represent it. First we choose a total ordering for the nodes. Using equation 3.8:1 on the FJD and the ordering,

we can find the LS between any pair of nodes. We put a link between those nodes (in the direction from the node earlier in the ordering to the later one) if the LS is greater than zero. After doing this for every pair of nodes we have the completed dag.

## 3.9 Comparing Link and Connection Strengths

A fundamental difference between CS and LS is that CS is defined with respect to a FJD, whereas LS is defined with respect to a FJD and an ordering on the nodes. For a given probabilistic specification, the CS between two nodes will be the same regardless of what BN is used to represent that specification. But LS values between two nodes will generally be different depending on the particular BN used to represent the FJD.

Figure 3.9:1 shows an example of the invariance of CS, and the dependence of LS, on the particular BN ordering. Both BNs represent the same FJD, but the total order used in (a) is A, B, C while the order in (b) is A, C, B. The $\Delta O$ link strengths, which are written along the links, are different between (a) and (b), but the connection strengths, which appear to the right, are the same for (a) and (b).

Notice that link $B \rightarrow C$ has reversed in going from (a) to (b) and its link strength remains the same. If any BN is modified by reversing a link, say $B \rightarrow C$, the only link strengths that will change are links going to B or to C, and if the $\Delta O$ measure for link strength is used, the strength of the link being reversed won't change.

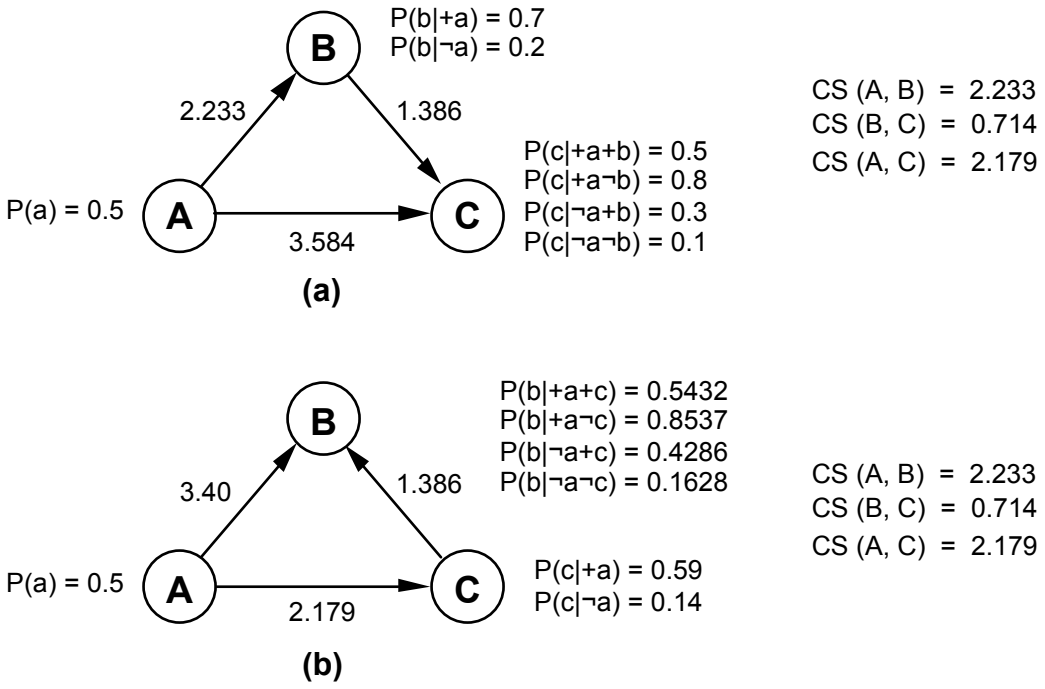


**Figure 3.9:1** - These two BNs represent the same FJD. Link strengths are different in each, but connection strengths are the same. The link strengths ($LS_o$) are displayed alongside the link, and the connection strengths ($CS_o$) are to the right. The total order in (a) is A, B, C, while in (b) the B → C link has been reversed, so it is A, C, B.

Another fundamental difference between CS and LS is that LS is local measure which can be computed very quickly, while CS is a global measure which may be very difficult to compute. Consider the BN in figure 3.9:2.

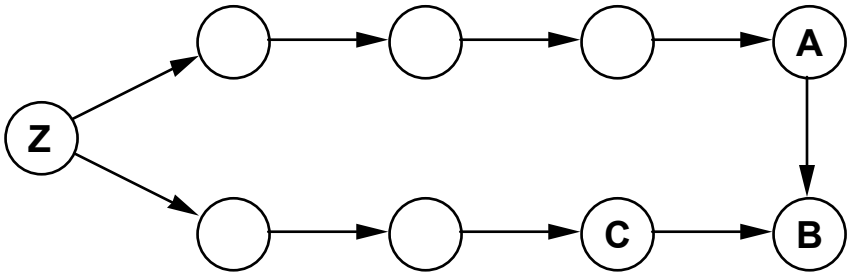


**Figure 3.9:2** - A BN to illustrate the global nature of CS and the local nature of LS.

Calculating an exact value for CS(A,B) involves all the nodes in the network, since evidence at A can change the belief at B by the active path through Z. However, calculating LS(A → B)

involves only the nodes A, B, and C since taking the maximum with C having evidence blocks this active path. In fact, it really only involves the NCP at B.

For any BN, calculating a precise value for the unconditional CS between two nodes involves all their ancestors, and calculating a precise value for the conditional CS between two nodes involves all their ancestors, and all the ancestors of every node with evidence. Also, in general, this calculation is of exponential complexity. Calculating the LS between two nodes involves only the NCP of the child node, and the complexity is linear with the number of NCP probabilities (or better than this, depending on how the NCP probabilities are represented).

# 4 Using Link Strength to Bound Connection Strength

## 4.1 ΔP Serial Combination

Link strengths may be combined by local operations to provide bounds on the connection strength of widely separated nodes. This allows for fast algorithms to find a limit on the degree to which evidence at one node can affect the belief at another. For a simple example, consider the following BN with two links in series:

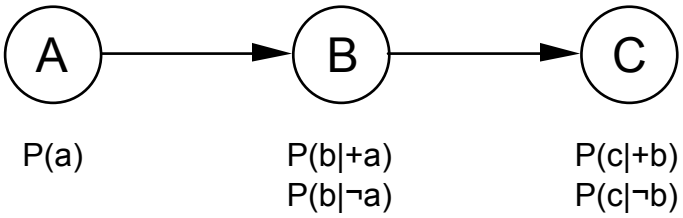


**Figure 4.1** - A three node BN in which we want to find the CS from A to C given the link strengths of $A \rightarrow B$ and $B \rightarrow C$.

We wish to find a bound on the ΔP connection strength from A to C given only the ΔP strengths of the two links $A \rightarrow B$ and $B \rightarrow C$. Actually, in this case we can find an exact relation, not just a bound (the proof is in Appendix C):

**Theorem 4.1**: For any three propositional variables A, B, and C, where A and C are independent given B:

$$CS_p(A, C) = CS_p(A, B)\ CS_p(B, C) \qquad 4.1$$

If node A is the only parent of B, and B is the only parent of C (as in figure 4.1), then the definition of link strength matches that of connection strength and we obtain:

$$CS_p(A, C) = LS_p(A \rightarrow B)\ LS_p(B \rightarrow C)$$

This rule was reported in Henrion89 in a different form. It also corresponds to the "shrinkage factor" of Markov net analysis (Howard71, p. 20). It can be chained for longer paths as long as the arrows all point in the same direction. Since each ΔP link strength is less than one, the effect of evidence can only get attenuated (or unchanged) by intermediate links; it can never be amplified.

## 4.2 ΔO Serial Combination

Consider the BN in figure 4.1 once again. This time we want to find a bound on the ΔO connection strength from A to C, given only information on the ΔO strengths from A to B, and from B to C. Below is the tightest bound that can be placed with only this information (see Appendix C for the derivation):

**Theorem 4.2**: For any three propositional variables A, B, and C, where A and C are independent given B:

$$\tanh\left(\tfrac{1}{4}\ CS_o(A, C)\right) \leq \tanh\left(\tfrac{1}{4}\ CS_o(A, B)\right)\ \tanh\left(\tfrac{1}{4}\ CS_o(B, C)\right) \qquad 4.2{:}1$$

where tanh is the hyperbolic tangent function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad 4.2{:}2$$

When $x \geq 0$, which is the domain of ΔO connection and link strengths, tanh(x), stays within the range [0, 1], and increases strictly monotonically as x increases (see figure 4.2:1). If the

logarithms used to define $CS_o$ and $LS_o$ are not natural logarithms, the multiplying constants 1/4 in equation 4.2:1 change to another constant dependent on the logarithm base, but otherwise the equation remains the same.

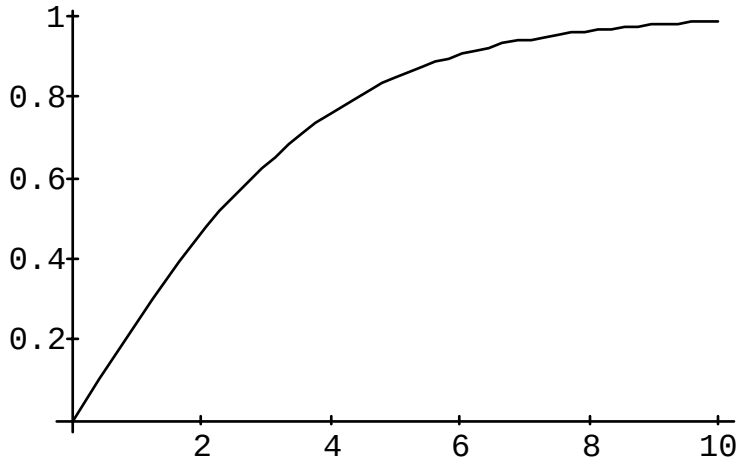


**Figure 4.2:1** - Graph of tanh(x/4). It starts out quite linear and then asymptotically approaches 1.

If node A is the only parent of B, and B is the only parent of C (as in figure 4.1), then the definition of link strength matches that of connection strength and we obtain:

$$\tanh\left(\tfrac{1}{4}\ CS_o(A, C)\right) \ \leq \ \tanh\left(\tfrac{1}{4}\ LS_o(A \rightarrow B)\right) \ \ \tanh\left(\tfrac{1}{4}\ LS_o(B \rightarrow C)\right)$$

First, consider a case in which $LS_o(B \rightarrow C) = 0$ in the BN of figure 4.1, so B is independent of C. Then, by equation 4.2:1 $\tanh\left(\tfrac{1}{4}\ CS_o(A, C)\right) = 0$, which implies $CS_o(A, C) = 0$, so C is independent of A. Second, consider a case in which $LS_o(B \rightarrow C) = \infty$, so B and C are potentially deterministicly related. Then, $\tanh\left(\tfrac{1}{4}\ LS_o(B \rightarrow C)\right) = 1$, and so $CS_o(A, C) \leq LS_o(A \rightarrow B)$. In general, when links $A \rightarrow B$ and $B \rightarrow C$ are in series, we can consider the link $B \rightarrow C$ as attenuating the effect of the link $A \rightarrow B$ on C. The degree of attenuation is always between that of the first case (independence leading to complete attenuation), and that of the second case (determinism leading to no attenuation).

Equation 4.2:1 can be chained for longer paths, resulting in a product with one term for each link. Since tanh(x), stays within the range [0, 1], the effect of evidence can only get attenuated (or unchanged) by intermediate links.

For small x, tanh (x) $\approx$ x, and for all $x \geq 0$, tanh (x) $\leq$ x, so from equation 4.2:1 we can form the following bound:

$$CS_o(A, C) \leq \frac{1}{4} LS_o(A \rightarrow B) \; LS_o(B \rightarrow C) \qquad 4.2:3$$

which is valid for all values of $LS_o$, but only forms a reasonably tight bound when both $LS_o$ values are small (say $LS_o \leq 2$). Using equation 3.3:9, we can see that the equation above, and therefore the $\Delta O$ serial link strength equation (4.2:1), approaches the $\Delta P$ serial link strength equation (4.1) when the links are weak (small $LS_o$ values).

### 4.2.1 Empirical Test of Bound Tightness

Some bounds provided in the field of computer science are notorious for being so far above values actually obtained in practice, that the bound is almost useless. A simulation was done in which 200000 BNs of the topology in figure 4.1 were generated, with NCP values drawn from a uniform distribution on [0,1]. In each case the bound on CS(A,C) given by equation 4.2:1 was compared to the true value of CS(A,C), and frequency histograms of the results appear in figures 4.2:2 and 4.2:3. Of course, in some applications the actual distribution of NCPs may be very different, which could lead to very different results on the tightness of the bound, but at least we can see that for this example application, the bound for CS given by equation 4.2:1 is usually not far above the true value of CS.

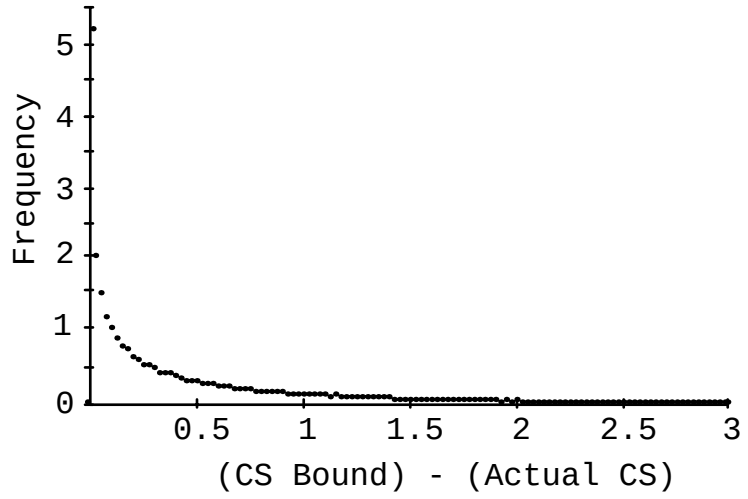


**Figure 4.2:2** - Frequency histogram of the amount that the bound calculated from LS(A $\rightarrow$ B) and LS(B $\rightarrow$ C) exceeded the actual value of CS(A,C) in 200000 cases of random BNs with the structure of Figure 4.1 and uniformly generated NCPs. Notice the point on the vertical axis at a height above 5, and the point to the left of the axis at a height of zero. The same graph is drawn with different scales below.

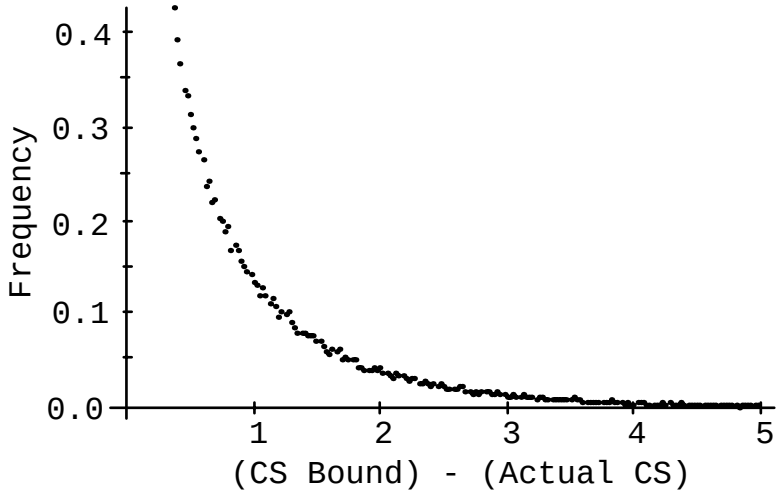


**Figure 4.2:3** - Expanded scales of graph in Figure 4.2:2.

## 4.3 Fundamental Sensitivity Equation

Now we wish to find a method we can use to find bounds on connection strength in a BN of arbitrary complexity. Sometimes we can break a probability problem down into two simpler problems by assuming that some proposition is true, solving the problem, then assuming it is false, and re-solving the problem. Then we say the solution for the case where we don't know

the value of the proposition is somewhere between the solutions for the proposition being true and being false.

For example, this method will work to find posterior probabilities. The posterior probability for a proposition Q in the case that we don't know the value of proposition Z, will be between the values it would have if we believed Z were true or Z were false. In fact the weighted average of these two bounds (weighted by the probabilities of Z being true or false) is actually the probability of Q for the case we don't know Z, which is the basis for the reasoning by assumptions method described in section 2.6. So we always have the following two relations:

$$P(q|+v\neg z) \leq P(q|+v) \leq P(q|+v+z) \quad \text{or} \quad P(q|+v+z) \leq P(q|+v) \leq P(q|+v\neg z) \qquad 4.3{:}1$$

$$P(q|\neg v\neg z) \leq P(q|\neg v) \leq P(q|\neg v+z) \quad \text{or} \quad P(q|\neg v+z) \leq P(q|\neg v) \leq P(q|\neg v\neg z) \qquad 4.3{:}2$$

We might be tempted to think connection strength will behave in the same way, since it is just the distance between the two posterior probabilities bounded in the two expressions above. That is, since 4.3:1 and 4.3:2 hold, we might expect the following to hold:

$$d\,(P(q|+v), P(q|\neg v)) \leq d\,(P(q|+v+z), P(q|\neg v+z)) \quad \text{or}$$

$$d\,(P(q|+v), P(q|\neg v)) \leq d\,(P(q|+v\neg z), P(q|\neg v\neg z)) \qquad \text{[false]} \qquad 4.3{:}3$$

or equivalently:

$$d\,(P(q|+v), P(q|\neg v)) \leq \max_z\ d\,(P(q|+vz),\ P(q|\neg vz)) \qquad \text{[false]} \qquad 4.3{:}4$$

$$CS\,(V, Q) \leq \max_z\ CS\,(V, Q|z) \qquad \text{[false]} \qquad 4.3{:}5$$

But these equations do not hold. The reason I've presented this line of thought is to warn that the connection strength for the case when a proposition Z is unknown is not bounded by the strengths for the cases when Z is true and when Z is false, as it is for a number of other probabilistic quantities (like belief, odds ratio belief, expected utility, most probable explanation, etc.). A trivial example to illustrate is shown in figure 4.3:1. When Z is unknown CS(V,Q) may be nonzero, but when Z is known, whether it is true or false, it d-separates V from Q so CS(V,Q|+z) and CS(V,Q|¬z) are zero.

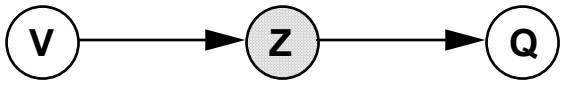


**Figure 4.3:1** - CS(V,Q) may be nonzero, but CS(V,Q|+z) and CS(V,Q|¬z) are both zero, so they don't provide bounds for CS(V,Q).

However, there is an equation similar to equation 4.3:5 that we can use to decompose connection strength problems.

**Theorem 4.3:** For any three propositional variables V, Q, and Z, we can decompose the connection strength from V to Q on cases of Z as follows:

$$CS(V,Q) \leq \max_z CS(V,Q|z) + CS(V,Z) * \min(CS(Z,Q|+v),\ CS(Z,Q|\neg v),\ CS(Z,Q)) \qquad 4.3{:}6$$

where * is a generalized multiplication corresponding to the serial combination rule. If the ΔP measure is used for connection strength, it is regular multiplication, but for the ΔO measure it is the ΔO serial combination rule:

$$x * y = 4 \tanh^{-1}(\tanh(\tfrac{1}{4} x)\ \tanh(\tfrac{1}{4} y)) \qquad 4.3{:}8$$

Since all connection strength problems can be decomposed using equation 4.3:6, it is called the *fundamental sensitivity equation* within this thesis. It is proved in Appendix C. Notice that it is similar to the more traditional style of decomposition equation given by 4.3:5, but with the addition of an extra term.

To help visualize the fundamental equation, consider the BN of figure 4.3:2. Varying evidence at node V will create a varying belief at the query node Q. We are interested in decomposing the effect on Q by cases of Z. We can consider equation 4.3:6 as composed of two parts, each corresponding to one of the active paths from V to Q. The first part is $\max_z$ CS(V,Q|z), and this corresponds to the link from V to Q, since Z is given and therefore the other path is blocked. The second part is CS(V,Z) * CS(Z,Q), which corresponds to the serial connection of the two links $V \rightarrow Z$ and $Z \rightarrow Q$. Together they provide a limit on how much the node V can effect the belief at Q.

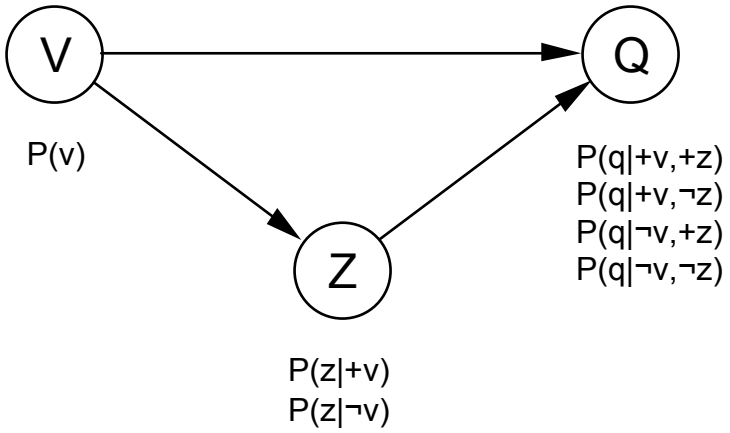


**Figure 4.3:2** - A BN to help visualize the terms of the fundamental equation. This equation can be considered to consist of a term for the $V \rightarrow Q$ path in parallel with a serial combination of terms for $V \rightarrow Z$ and $Z \rightarrow Q$.

The fundamental equation applies to any 3 variables of any BN. They may be connected like the BN in figure 4.3:2, or they may be scattered through a large network with no links between them.

## 4.4 Example of Finding CS by Fundamental Equation

Suppose we have the BN of figure 4.4:1 and we wish to find the connection strength from node $X_1$ to node $X_3$.

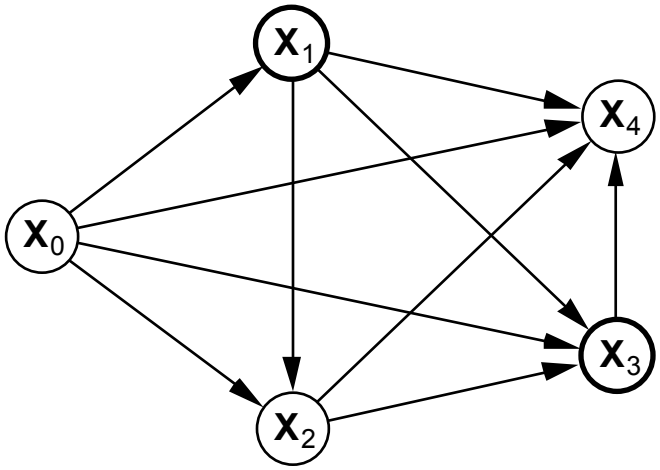


**Figure 4.4:1** - A BN for which we wish to find the CS from node $X_1$ to $X_3$.

The fundamental equation implies both the following two equations, which we will use to solve this problem:

$$CS(V, Q) \leq \max_{z} CS(V, Q|z) + CS(V, Z) * CS(Z, Q) \qquad 4.4:1$$

$$CS(V, Q) \leq \max_{z} CS(V, Q|z) + CS(V, Z) * \max_{v} CS(Z, Q|v) \qquad 4.4:2$$

Using equation 4.4:2 with $X_1$ as V, $X_3$ as Q, and $X_0$ as Z:

$$CS(X_1, X_3) \leq \max_{x_0} CS(X_1, X_3|x_0) + CS(X_1, X_0) * \max_{x_1} CS(X_0, X_3|x_1) \qquad 4.4:4$$

Now we find bounds on the three terms of the equation above, one by one, by using the fundamental equation repeatedly. For the first term:

$$\max_{x_0} CS(X_1, X_3|x_0) \leq \qquad 4.4:5$$

$$\max_{x_0, x_2} CS(X_1, X_3|x_0,x_2) + \max_{x_0} CS(X_1, X_2|x_0) * \max_{x_0, x_1} CS(X_2, X_3|x_0,x_1)$$

The first term in the equation above (4.4:5) matches the definition of link strength for the $X_1 \rightarrow X_3$ link:

$$LS(X_1 \rightarrow X_3) = \max_{\mathbf{c} \in \prod(C(X_3) - \{X_1\})} CS(X_1, X_3|\mathbf{c}) = \max_{x_0, x_2} CS(X_1, X_3|x_0,x_2) \qquad 4.4:6$$

The other terms in equation 4.4:5 are also link strengths, so we may rewrite equation 4.4:5 as:

$$\max_{x_0} CS(X_1, X_3|x_0) \leq LS(X_1 \rightarrow X_3) + LS(X_1 \rightarrow X_2) * LS(X_2 \rightarrow X_3) \qquad 4.4:7$$

The second term of equation 4.4:4 is CS ($X_1$, $X_0$). If we are using the $\Delta O$ measure of CS, then this is equivalent to CS ($X_0$, $X_1$), which is the link strength from $X_0$ to $X_1$. For the last term of equation 4.4:4 we get:

$$\max_{x_1} CS(X_0, X_3|x_1) \leq \qquad 4.4:8$$

$$\max_{x_1, x_2} CS(X_0, X_3|x_1,x_2) + \max_{x_1} CS(X_0, X_2|x_1) * \max_{x_0, x_1} CS(X_2, X_3|x_0,x_1)$$

Each of the terms in the above equation is a link strength, so we may write it as:

$$\max_{x_1} CS(X_0, X_3|x_1) \leq LS(X_0 \rightarrow X_3) + LS(X_0 \rightarrow X_2) * LS(X_2 \rightarrow X_3) \qquad 4.4:9$$

Combining all of this we finally get our bound for $CS(X_1, X_3)$:

$$\begin{aligned} CS(X_1, X_3) \leq \quad & LS(\overrightarrow{X_1X_3}) + \\ & LS(\overrightarrow{X_1X_2}) * LS(\overrightarrow{X_2X_3}) + \\ & LS(\overleftarrow{X_1X_0}) * (LS(\overrightarrow{X_0X_3}) + LS(\overrightarrow{X_0X_2}) * LS(\overrightarrow{X_2X_3})) \end{aligned} \qquad 4.4:10$$

By examining the BN in figure 4.4:1, we can see how the equation above can be considered as having terms for all of the paths from $X_1$ to $X_3$. The first line corresponds to the link straight from $X_1$ to $X_3$. The second line corresponds to the path from $X_1$, through $X_2$, to $X_3$ (i.e. the $X_1 \rightarrow X_2$ and $X_2 \rightarrow X_3$ links in series). The last line corresponds to the paths from $X_1$ through $X_0$, then either straight to $X_3$ or from $X_0$ to $X_2$, then to $X_3$.

Notice that $X_4$ was not involved at all in finding $CS(X_1, X_3)$. For any BN, finding $CS(X_i, X_j)$, where i and j indicate the position of the nodes in the total order, does not involve any nodes $X_k$, where k>i and k>j, since there are no active paths from $X_i$ to $X_j$ through $X_k$ (they all have at least one converging node).

Because of this, equation 4.4:10 can be used to find the connection strength from node 1 to node 3 in any BN. The nodes that come after node 3 are irrelevant. Equation 4.4:10 was developed for a *fully connected* network (i.e. every two nodes are connected by a link). If we wish to use it for a network that is not fully connected, then we just use a link strength of 0 between nodes with no link connecting them, as described in section 3.9.

For example, we can use equation 4.4:10 to find a bound on $CS(X_1,X_3)$ for the BN in figure 4.4:2. Setting $LS(\overrightarrow{X_1X_2}) = 0$ and $LS(\overrightarrow{X_0X_3}) = 0$, we obtain:

$$CS(X_1, X_3) \leq LS(\overrightarrow{X_1X_3}) + LS(\overleftarrow{X_1X_0}) * LS(\overrightarrow{X_0X_2}) * LS(\overrightarrow{X_2X_3}) \qquad 4.4:11$$

This consists of two parallel paths, one straight from $X_1$ to $X_3$, and the other consisting of 3 links in serial: $X_1$ to X0, then to $X_2$, and finally to $X_3$. The path from $X_1$ through $X_4$ to $X_3$ does not appear because it is blocked by the converging node $X_4$.

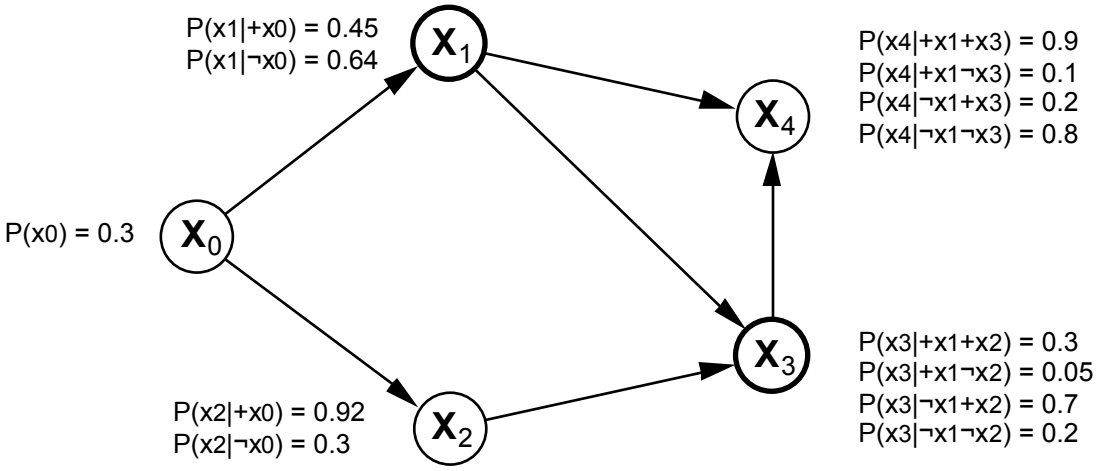


**Figure 4.4:2** - A BN like that in figure 4.4:1, but with fewer links. We can use the equation developed for bounding $CS(X_1,X_3)$ of the BN in 4.4:1 to bound $CS(X_1,X_3)$ of this BN.

Using the numbers provided we obtain:

$$LS_o(X_1 \leftarrow X_0) = LS_o(X_0 \rightarrow X_1) = d_o(P(x_1|+x_0), P(x_1|\neg x_0)) = d_o(0.45, 0.64) = 0.776$$

$$LS_o(X_0 \rightarrow X_2) = d_o(P(x_2|+x_0), P(x_2|\neg x_0)) = d_o(0.92, 0.30) = 3.29$$

$$\begin{aligned} LS_o(X_1 \rightarrow X_3) &= \max(d_o(P(x_3|+x_1+x_2), P(x_3|\neg x_1+x_2)),\ d_o(P(x_3|+x_1\neg x_2), P(x_3|\neg x_1\neg x_2))) \\ &= \max(d_o(0.3, 0.7),\ d_o(0.05, 0.2)) = 1.69 \end{aligned}$$

$$\begin{aligned} LS_o(X_2 \rightarrow X_3) &= \max(d_o(P(x_3|+x_2+x_1), P(x_3|\neg x_2+x_1)),\ d_o(P(x_3|+x_2\neg x_1), P(x_3|\neg x_2\neg x_1))) \\ &= \max(d_o(0.3, 0.05),\ d_o(0.7, 0.2)) = 2.23 \end{aligned}$$

$$\begin{aligned} CS_o(X_1, X_3) &\leq LS_o(X_1 \rightarrow X_3) + LS_o(X_1 \leftarrow X_0) * LS_o(X_0 \rightarrow X_2) * LS_o(X_2 \rightarrow X_3) \\ &= 1.69 + 0.776 * 3.29 * 2.23 \\ &= 1.69 + 4 \tanh^{-1}(\tanh(0.776 / 4) \tanh(3.29 / 4) \tanh(2.23 / 4)) \\ &= 1.95 \end{aligned}$$

So the bound we calculate is $CS_o(X_1, X_3) \leq 1.95$. In actual fact $CS_o(X_1, X_3) = 1.543$.

We can repeat the calculation to find a bound on ΔP connection strength, but the calculation of the "backwards" connection strength $CS_p(X_1, X_0)$ is more difficult than in the ΔO case, since in the ΔO case it was just the same as $CS_o(X_0, X_1)$ which is $LS_o(X_0 \rightarrow X_1)$. Actually, this time we

can easily find $CS_p(X_1, X_0)$ using Bayes rule, but in the general case finding "backwards" $CS_p$ link strength involves nonlocal calculations.

## 4.5 Path Based Methods

Most researchers have been reluctant to attach formal significance to the paths of a BN. It seems that on an intuitive level they make use of ideas like "effect" or "influence" "flowing" along the paths of a BN, and will speak of "weak paths", "strong links", etc., but do not formally define these concepts.

One reason for this may be that paths are not intrinsic to the probabilistic model (the FJD), but are an artifact of the BN factoring process. So if a BN had been constructed with a different total ordering for the nodes, its path structures could have turned out completely different. One can observe the same effect in the operation of "link reversal", which does not change the FJD represented by a BN, but can add or remove links, thereby changing the paths of the BN.

Another reason for avoiding paths could be due to concerns that they lead to an over-simplistic view of the BN. There is a tendency for beginners to think of a BN as a sort of constraint network, with the belief of a child node given as a function of the beliefs of its parents. Or, if they are a bit more sophisticated, they may think the belief in a node can be given as a function of the beliefs in the nodes of its Markov boundary (i.e. its parents, children, and parents of its children).

Actually, to express the belief in a node as a function, it must be expressed as a function of the *joint* beliefs of its Markov boundary nodes (i.e. the beliefs in the Cartesian product of their values). Thinking in terms of paths can obscure this. For example, consider the BN of figure 4.5:1. When there is no evidence, the beliefs at A, B, C, and D are all 1/2. If we get evidence TRUE for A, the beliefs at B and C remain at 1/2, but the belief at D changes to 3/4. Thinking in terms of a constraint network, or "flow of influence along paths", it is hard to see how a change at A can create a change at D without changing the beliefs at B or C. Of course, it is the joint belief in B and C which has changed (BEL(+b+c) changes from 1/4 to 3/8, BEL(+b¬c) changes from 1/4 to 1/8, etc.). So we must be careful with the path concept.

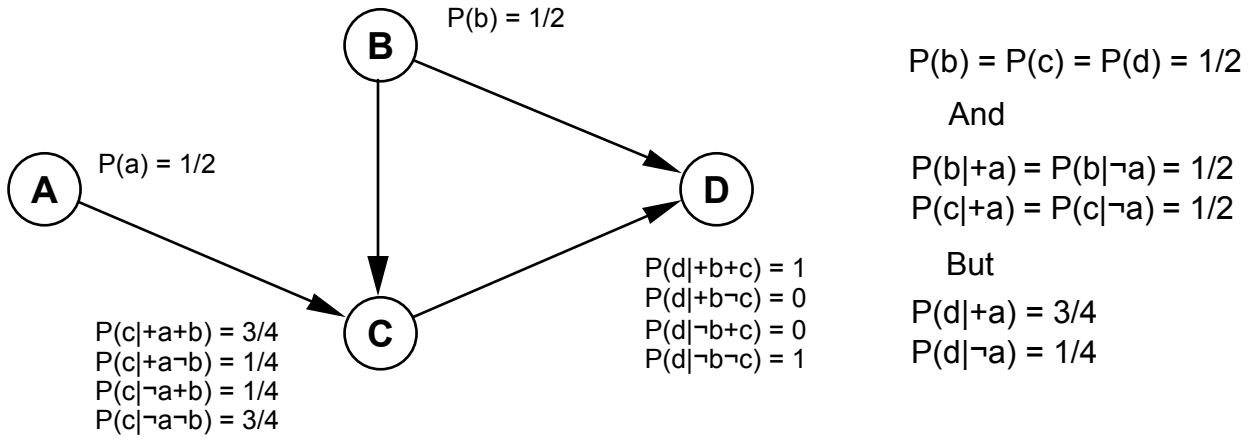


**Figure 4.5:1** - Evidence at A changes the belief at D, but not at B or C.

There are only a few examples of previous research that make extensive use of paths in BNs. The most obvious example is the d-separation algorithm itself. It finds independence information by tracing active and blocked paths. Since connection strength can be considered the "degree of independence" it is not surprising that bounds for it can be found using a path based method as well.

Suermondt92 uses paths to generate explanations of BN inference for a human user. He considers some paths more significant than others if breaking a link along one of them results in a significant change in the inference result. That way he can prioritize "chains of reasoning" in presenting the explanation. Intuitively he considers something "flowing" along paths, although he is wary of going to far with this line of thought, as is evident from the quote, "Such an image, in which probabilistic updates are treated analogously to electrical currents, is simplistic, and is invalid in many cases; we cannot predict in a definitive manner the combined effects of evidence transmission along multiple chains by analyzing the chains separately, since there are often unpredictable synergistic effects". He doesn't arrive at any of the central results of this thesis, since he measures changes in belief due to breaking a link by doing full Bayesian inference with the link in place, then again with the link broken, and then compares the beliefs produced in each case (similar to the method mentioned in section 3.6, but breaking links instead of instantiating nodes). This can be very expensive when there are many links to try breaking, and worse when one considers combinations of links. Of course his method is much more precise than just the

bounds calculated in this thesis, but the methods of this thesis could be used as a pre-screening phase to eliminate obviously weak paths from consideration (since the main purpose is to leave out very weak paths from the explanation).

Wellman90 uses the paths in a BN to do qualitative probabilistic inference. The purpose is to answer the question, "if the belief in this node increases (say through virtual evidence), will the belief in this other node increase or decrease?" He traces "influence" along paths to arrive at the answer. This is discussed in greater detail in section 6.1:1.

### 4.5.1 The CS Path Algorithm

We can simplify using the fundamental equation to find connection strength bounds, by using it to develop a path-based algorithm. The resulting formulation is also intuitively more appealing.

We will find an expression for a bound on the $\Delta O$ connection strength from node $X_v$ to node $X_q$, that is $CS_o(X_v,X_q)$, for a fully connected BN. The expression will be entirely in terms of link strengths. Later the solution for any BN can be generated simply by setting the link strengths of missing links to 0 in that expression.

We start by providing a total ordering for the nodes of the BN consistent with its dag, and we label the nodes $X_0$, $X_1$, ..., $X_n$ where a lower index corresponds to earlier in the total order. Since we are assuming the BN is fully connected, the parents of node $X_i$ will be $\{X_j \mid 0 \leq j < i\}$. To find $CS_o(X_v,X_q)$ we first apply the fundamental equation (version 4.4:1) to it, with $Z = X_0$, and obtain:

$$CS(X_v, X_q) \leq CS(X_v, X_0) * CS(X_0, X_q) + \max_{x_0} CS(X_v, X_q|x_0)$$

Now we apply version 4.4:1 of the fundamental equation again, this time with $Z = X_1$, to the last term of the above equation. Then we repeat the process until its been done with $Z = X_j$, for j=0 to j=v–1. The result is the expansion shown below. Each line corresponds to one application of the fundamental equation (used with $Z = X_j$, j having the value shown in the rightmost column) to the last term of the line above it. The resulting bound is the sum of products in the left hand column.

$CS(X_v, X_q) \leq$

| | | |
|---|---|---|
| $CS(X_v, X_0) * CS(X_0, X_q) +$ | $\max_{x_0} CS(X_v, X_q \mid x_0)$ | j=0 |
| $\max_{x_0} CS(X_v, X_1 \mid x) * \max_{x_0} CS(X_1, X_q \mid x) +$ | $\max_{x_0, x_1} CS(X_v, X_q \mid x_0, x_1)$ | j=1 |
| $\max_{\mathbf{x} = (x_0, x_1)} CS(X_v, X_2 \mid \mathbf{x}) * \max_{\mathbf{x} = (x_0, x_1)} CS(X_2, X_q \mid \mathbf{x}) +$ | $\max_{\mathbf{x} = (x_0, x_1, x_2)} CS(X_v, X_q \mid \mathbf{x})$ | j=2 |
| • • • | | |
| $\max_{\mathbf{x} = (x_0, .., x_{v-2})} CS(X_v, X_{v-1} \mid \mathbf{x}) * \max_{\mathbf{x} = (x_0, .., x_{v-2})} CS(X_{v-1}, X_q \mid \mathbf{x}) +$ | $\max_{\mathbf{x} = (x_0, x_1, .., x_{v-1})} CS(X_v, X_q \mid \mathbf{x})$ | j=v–1 |

We can express the above sum of products as:

$$CS(X_v, X_q) \leq \sum_{j=0}^{v-1} \max_{\mathbf{x} = (x_0, .., x_{j-1})} CS(X_v, X_j \mid \mathbf{x}) * \max_{\mathbf{x} = (x_0, .., x_{j-1})} CS(X_j, X_q \mid \mathbf{x}) + \max_{\mathbf{x} = (x_0, .., x_{v-1})} CS(X_v, X_q \mid \mathbf{x}) \qquad 4.5{:}1$$

Since this derivation is for $\Delta O$ connection strength, which is commutative, we can reverse the order of the CS arguments in the first factor of the first line. Also, we can include the second line in the sum by increasing the range of its index, so we get:

$$CS(X_v, X_q) \leq \sum_{j=0}^{v} \max_{\mathbf{x} = (x_0, .., x_{j-1})} CS(X_j, X_v \mid \mathbf{x}) * \max_{\mathbf{x} = (x_0, .., x_{j-1})} CS(X_j, X_q \mid \mathbf{x}) \qquad 4.5{:}2$$

The CS expressions in the above equation are all of the form $\max_{\mathbf{x} = (x_0, .., x_{j-1})} CS(X_j, X_q \mid \mathbf{x})$, some with j or q replaced by v. We can generate an expansion to solve for these expressions in the same way that we did for $CS(X_v, X_q)$. It appears below, with each line formed by expanding the last term of the line above it using version 4.4:2 of the fundamental equation (with $Z = X_k$, k having the value shown in the rightmost column), to form the sum of products shown in the left hand column.

$$\max_{\mathbf{x}=(x_0,.,x_{j-1})} CS(X_j,X_q|\mathbf{x}) \leq$$

$$\begin{array}{l|l|l}
\max_{\mathbf{x}=(x_0,x_1,.,x_{j-1})} CS(X_j,X_{j+1}|\mathbf{x}) * \max_{\mathbf{x}=(x_0,x_1,.,x_j)} CS(X_{j+1},X_q|\mathbf{x}) + & \max_{\mathbf{x}=(x_0,x_1,.,x_{j-1},x_{j+1})} CS(X_j,X_q|\mathbf{x}) & k=j+1 \\
\max_{\mathbf{x}=(x_0,.,x_{j-1},x_{j+1})} CS(X_j,X_{j+2}|\mathbf{x}) * \max_{\mathbf{x}=(x_0,.,x_{j+1})} CS(X_{j+2},X_q|\mathbf{x}) + & \max_{\mathbf{x}=(x_0,.,x_{j-1},x_{j+1},x_{j+2})} CS(X_j,X_q|\mathbf{x}) & k=j+2 \\
\bullet \quad \bullet \quad \bullet & & \\
\max_{\mathbf{x}=(x_0,.,x_{j-1},x_{j+1},.,x_{q-2})} CS(X_j,X_{q-1}|\mathbf{x}) * \max_{\mathbf{x}=(x_0,.,x_{q-2})} CS(X_{q-1},X_q|\mathbf{x}) + & \max_{\mathbf{x}=(x_0,.,x_{j-1},x_{j+1},.,x_{q-1})} CS(X_j,X_q|\mathbf{x}) & k=q-1
\end{array}$$

We can express the above sum of products as:

$$\max_{\mathbf{x}=(x_0,.,x_{j-1})} CS(X_j,X_q|\mathbf{x}) \leq \sum_{k=j+1}^{q-1} \max_{\mathbf{x}=(x_0,.,x_{j-1},x_{j+1},x_{k-1})} CS(X_j,X_k|\mathbf{x}) * \max_{\mathbf{x}=(x_0,.,x_{k-1})} CS(X_k,X_q|\mathbf{x}) +$$

$$\max_{\mathbf{x}=(x_0,.,x_{j-1},x_{j+1},.,x_{q-1})} CS(X_j,X_q|\mathbf{x})$$

Once again we must evaluate the two factors in the sum of products above. The first factor of the product matches the link strength definition. The second factor is of the same form as the expression being bounded, but with the j index at least one larger, so it may be bounded recursively using the same equation. Folding the second line into the sum, we obtain the following recursive equation:

$$\max_{\mathbf{x}=(x_0,.,x_{j-1})} CS(X_j,X_q|\mathbf{x}) \leq \begin{cases} \sum_{k=j+1}^{q} LS(\overrightarrow{x_j x_k}) * \max_{\mathbf{x}=(x_0,.,x_{k-1})} CS(X_k,X_q|\mathbf{x}) & j<q \\ \infty & j=q \end{cases} \qquad 4.5:3$$

This equation can be used to bound each of the CS expressions of equation 4.5:2.

Now we modify equations 4.5:2 and 4.5:4 to suit the case of a network that is not fully connected, by setting the appropriate link strengths to zero. The range of the summation is reduced to only include nonzero terms. Also, we use a notation in which $C^+(X)$ are the ancestors of X, C*(X) are the ancestors of X and the node X, and S(X) are the successors (children) of X.

$$CS(X_v, X_q) \leq \sum_{X_j \in C^*(X_v) \cap C^+(X_q)} \max_{\mathbf{x} \in \prod(C^+(X_j)} CS(X_j,X_v|\mathbf{x}) * \max_{\mathbf{x} \in \prod C^+(X_j)} CS(X_j,X_q|\mathbf{x})$$

$$\max_{\mathbf{x} \in \prod C^+(X_j)} CS(X_j, X_q|\mathbf{x}) \;\leq\; \begin{cases} \sum_{X_k \in S(X_j) \cap C^*(X_q)} LS(\overrightarrow{X_j X_k})^* \max_{\mathbf{x} \in \prod C^+(X_k)} CS(X_k, X_q \mid \mathbf{x}) & X_j \in C^+(X_q) \\ \infty & X_j = X_q \end{cases}$$

The expression $\max_{\mathbf{x} \in \prod C^+(X_j)} CS(X_j, X_q|\mathbf{x})$, appears repeatedly in the above equations so we define a quantity termed "the strength of all forward paths from Xj to Xq", and denote it with the letter F, as follows:

$$F(X_j, X_q) \;=\; \max_{\mathbf{x} \in \prod C^+(X_j)} CS(X_j, X_q|\mathbf{x}) \qquad 4.5{:}6$$

We substitute this into our two CS bounding equations. Also, since they no longer rely on the total order, and we don't need integer indexes for the nodes, we can rename all the nodes from $X_a$ style to A style for notational aesthetics. This gives:

$$CS(V, Q) \;\leq\; \sum_{J \in C^*(V) \cap C^+(Q)} F(J,V) \;*\; F(J,Q) \qquad Q \notin C^*(V) \qquad 4.5{:}7$$

$$F(X,Y) \;\leq\; \begin{cases} \sum_{K \in S(X) \cap C^*(Y)} LS(\overrightarrow{XK})^* \; F(K,Y) & X \neq Y \\ \infty & X = Y \end{cases} \qquad 4.5{:}8$$

With the two equations above we can now bound the connection strength between any two nodes of any BN, using only link strength values. The following written description may help to make the above equations more intuitive.

**Specification 4.5**: A bound on the ΔO connection strength from node V to node Q, when Q is not an ancestor of V, is given by:

The strength of all forward paths from V to Q, plus the sum over every node, J, which is an ancestor of both V and Q, of the strength of all backwards paths from V to J, multiplied by the strength of all forward paths from J to Q.

The "strength of all forward paths from X to Y" is bounded by the sum over all X's children, K, which are ancestors of Y (or Y itself), of:

The strength of the link from X to K, multiplied by the strength of all forward paths from K to Y.

The "strength of all backward paths from V to J" is the same as the "strength of all forward paths from J to V". In the above, the term "multiply" is used in its generalized sense to mean serial combination as described in section 4.2. To find a bound on $CS_o(V,Q)$ when Q *is* an ancestor of V, we use the above algorithm to find a bound on $CS_o(Q,V)$, which is equal to $CS_o(V,Q)$.

## 4.6 Complexity of Path Algorithm

Using equations 4.5:7 and 4.5:8 directly in a recursive manner to find a bound for connection strength results in an algorithm of exponential worst case complexity. However, the same values are being repeatedly calculated and so by doing a "bottom up" evaluation instead, the complexity becomes linear in the number of links.

The bottom up algorithm to find a bound for CS(V,Q) involves calculating F(J,V) and F(J,Q) values for a number of nodes J, and storing those values with the nodes to aid in calculating further F(J,V) and F(J,Q) values. Once these values have been calculated for all the required nodes, equation 4.5:7 is used to combine them, yielding the desired bound. This yields the following algorithm, in which the descendants of a set of elements is defined as the set of all the descendants of the elements (and C* represents all ancestors, and S* all descendents):

**Algorithm 4.6** - To find a bound on $CS_o(V,Q)$:

1. If Q is an ancestor of V, switch V and Q, and find the equivalent connection strength CS(Q,V).

2. Starting with J = Q, and working J backwards through a total order on the nodes consistent with the dag, calculate F(J,Q) values for all J falling in the set: $S^*(C^*(V)) \cap C^*(Q)$ using equation 4.5:8.

3. Starting with J = V, and working J backwards through the total order, calculate F(J,V) values for all the nodes falling in the set: $S^*(C^*(Q)) \cap C^*(V)$ using equation 4.5:8.

4. Use equation 4.5:7 to sum up products of F(J,V) and F(J,Q) values, yielding the bound on CS(V,Q).

The following property holds for the above algorithm (proved in Appendix C):

**Lemma 4.6:** When calculating each new F(J,Q) in step 2, all the subcalculations of F(K,Q) that are required, will already be calculated. The same holds for step 3.

To determine the complexity of algorithm 4.6, a good estimate is provided by the number of generalized multiplications required. The CS(V,Q) bound is formed only by generalized multiplications and additions, and the number of additions will be slightly less than the number of multiplications, since the only additions that are required are those to add one product to another. The following theorem supplies the number of multiplications needed (proved in Appendix C):

**Theorem 4.6**: The number of generalized multiplications required to find a bound on $CS_0(V,Q)$ using algorithm 4.6, is the number of links between the ancestors of Q which are also descendants of ancestors of V, plus the number of links between the ancestors of V which are also descendants of ancestors of Q, that is:

$$\text{Number multiplies} = \text{Number links between } (S^*(C^*(Q)) \cap C^*(V)) + \text{Number links between } (S^*(C^*(V)) \cap C^*(Q))$$

Clearly this will be less than the number of links between ancestors of V, plus the number of links between ancestors of Q.

Depending on the representation of the BN, extra computation may be required to determine which nodes are the required descendants, ancestors, etc., but even when doing this, the overall complexity is linear in the number of links between ancestors of V and Q.

The space requirements are minimal. On top of what is required for the BN and control of the algorithm, only $| S^*(C^*(V)) \cap C^*(Q) |$ + $| S^*(C^*(Q)) \cap C^*(V) |$ real numbers must be stored (i.e. linear in the number of ancestor nodes of V and Q).

## 4.7 Dealing With Evidence

We may wish to find a bound for a conditional connection strength, that is, a bound for a connection strength after the BN has received some evidence. If we are automatically modifying the network to incorporate the evidence as it arrives using evidence absorption (see section 2.6), then we can simply use the methods of the previous section on the new BN. However, if we want to find a CS bound when knowing that the evidence is present, but not propagated, we need some new techniques.

Link strengths were defined by maximizing CS over all states of the other parents of the child in the link. If the evidence received at the BN is for one of those parents, that maximum may be reduced. Consider the example BN in figure 4.7:1:

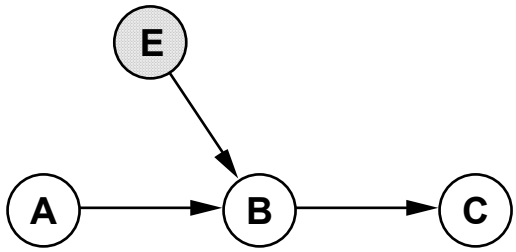


**Figure 4.7:1** - Evidence at node E restricts the CS(A,C) bound.

Say we are finding a bound on CS(A,C) with no evidence at node E. This is provided by

$$CS(A,C) \leq LS(A \rightarrow B) * LS(B \rightarrow C) = \max_{e} CS(A,B|e) * CS(B,C)$$

where * is the serial combination operator. However, if we receive evidence that E is true, then a bound for the conditional connection strength, CS(A,C|+e) is provided by:

$$CS(A,C|+e) \leq LS(A \overset{\rightarrow}{} B|+e) * LS(B \overset{\rightarrow}{} C|+e) = CS(A,B|+e) * CS(B,C)$$

This new bound for CS will be less than or the same as the original bound since $CS(A,B|+e) \leq \max(CS(A,B|+e), CS(A,B|\neg e))$. Often when evidence is received it will improve CS bounds in this manner.

Another way in which receiving evidence can lower the CS bound, is if the evidence is for a nonconverging node which is right on an active path. The evidence blocks the active path, so the CS contribution from that path drops to zero, giving a lower overall CS value, and a lower value for the CS bound.

Although receiving evidence often lowers CS values, it may increase CS values by creating new active paths through converging nodes which have received the evidence. Consider the example BN in figure 4.7:2:

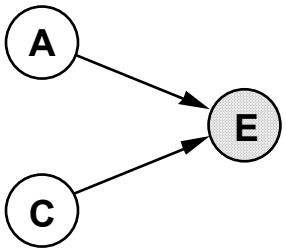


**Figure 4.7:2** - BN with evidence at E, for which we wish to find the CS from A to C via the intercausal path through E.

Without any evidence at E, CS(A,C) = 0, and the CS(A,C) bound calculated by active paths is also zero. But once evidence TRUE arrives for E, an active path from A to C is created. The strength of this path is d(P(c|+a+e), P(c|¬a+e)), and it is termed an *intercausal link strength*, since reasoning from A to C is intercausal reasoning, as defined in section 2.6. For ΔO connection strength it turns out to be (proved in Appendix C):

$$CS_o(A,C|+e) \;=\; LS_o(A^{\rightarrow}{+}e^{\leftarrow}C) \;=\; \left|\log \frac{P(+e|+a+c)\,P(+e|\neg a\neg c)}{P(+e|+a\neg c)\,P(+e|\neg a+c)}\right| \qquad 4.7{:}1$$

where the $LS(A^{\rightarrow}{+}e^{\leftarrow}C)$ is a notation invented to denote an intercausal link strength from A to C when E has evidence TRUE. It is defined by the equation above.

As an example, we can put all these techniques together to find a bound on CS(A,D|e,f) in the BN of figure 4.7:3:

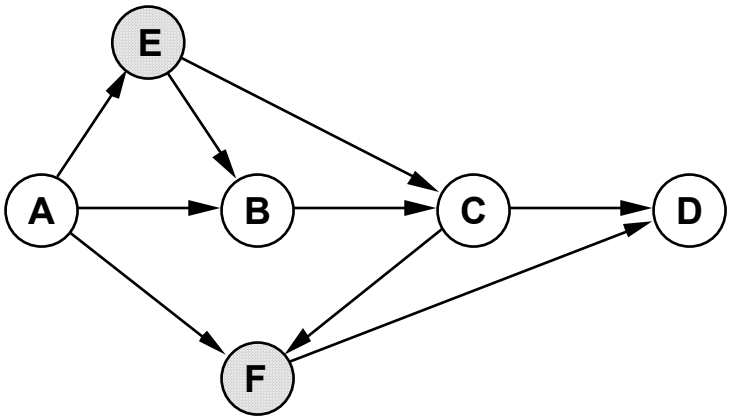


**Figure 4.7:3** - BN with evidence at E and F, for which we wish to find a bound on the CS from A to D.

The active paths from A to D are A,B,C,D and A,F,C,D. Each one of them forms one of the two terms in the bounding equation below:

$$CS(A,D|e,f) \leq [(LS(A \rightarrow B|e) * LS(B \rightarrow C|e) + LS(A \rightarrow f \leftarrow C))] * LS(C \rightarrow D|f)$$

# 5 Applications

## 5.1 BN Link Display

The first application of link strength that we consider is as a visualization aid for humans. BN diagrams have been praised as a great tool for people to visualize probabilistic relationships, and this tool may be improved by displaying the links according to their strengths. That way someone viewing the BN doesn't just get information about node independence, but also about a "degree of independence". An extremely weak link is very nearly an independence, but this information is lost if it is drawn exactly the same as the rest of the links. One possibility is to draw stronger links with thicker (or darker) lines and weaker links with thinner (or lighter) lines. Viewing a BN can be much more meaningful if one sees a skeleton of very heavy lines corresponding to definitions and constraints, followed by slightly lighter lines for less certain rules, and so on, down to faint lines corresponding to very weak dependencies, and of course the absence of links indicating independencies.

When using the $\Delta O$ link strength measure, we must represent the zero-to-infinity scale of $LS_o$ with a finite width line. Graphical display of infinite scales is commonly accomplished using the mapping: $W = x / (x + \alpha)$, where $W$ is proportional to the width of the line, and $\alpha$ is an adjustable scale parameter. This is approximately linear in x for $x \ll \alpha$, and approaches 1 as x approaches infinity.

Another possibility is to use $W = \tanh (x / \alpha)$, which is also linear in x for $x \ll \alpha$, and approaches 1 as x approaches infinity (see graph of figure 4.2:1). This mapping is

recommended (with $\alpha = 4$), since it corresponds closely to serial combinations of links. That way the human viewer can easily imagine the minimum attenuation of a chain of links as the product of the attenuations of each of the links. So two 50% width lines in series correspond to a 25% width line (or smaller) connection.

Using this scheme, it takes a bit more work to mentally combine the $\Delta O$ strength of parallel paths. For example two 50% width lines would combine to form, at most, an 80% width connection ($2 \times 4\tanh^{-1}(0.5) = 4\tanh^{-1}(0.8)$). However, for the finer lines, simple addition of line width can be used to approximately combine parallel paths. For example, two 25% width lines combine in parallel to form, at most, a 47% width connection, which is very nearly 50%.

This measure used for the width of the line turns out to be the absolute value of the statistical measure of association known as the *coefficient of colligation*, or *Yule's Y* (not to be confused with Yule's Q, which is in more common usage). The original invention of this measure had nothing to do with the chaining between variables that we use it for here, or so it appears from reading the paper in which it was introduced. For a description of Yule's Y see Appendix B, and for the original paper, see Yule1912.

Figure 5.1 shows the same BN as figure 2.5, but with the link strengths printed beside each link, and the links drawn in different widths to show their strengths according to the formula:

$$\text{Width} = \begin{cases} 0 & LS_o = 0 \\ \max((0.2\text{mm}),\ (2\text{mm}) \times \tanh(LS_o / 4)) & \text{otherwise} \end{cases}$$

At a glance one can get an idea of which dependencies are always of minor importance. It must be remembered that each link strength represents the amount that the parent node can effect the child, *maximized over all possible beliefs for the other parents*. Furthermore, $\Delta O$ link strength was used, so any effects that bring a belief close to 0 or 1 are considered very significant.

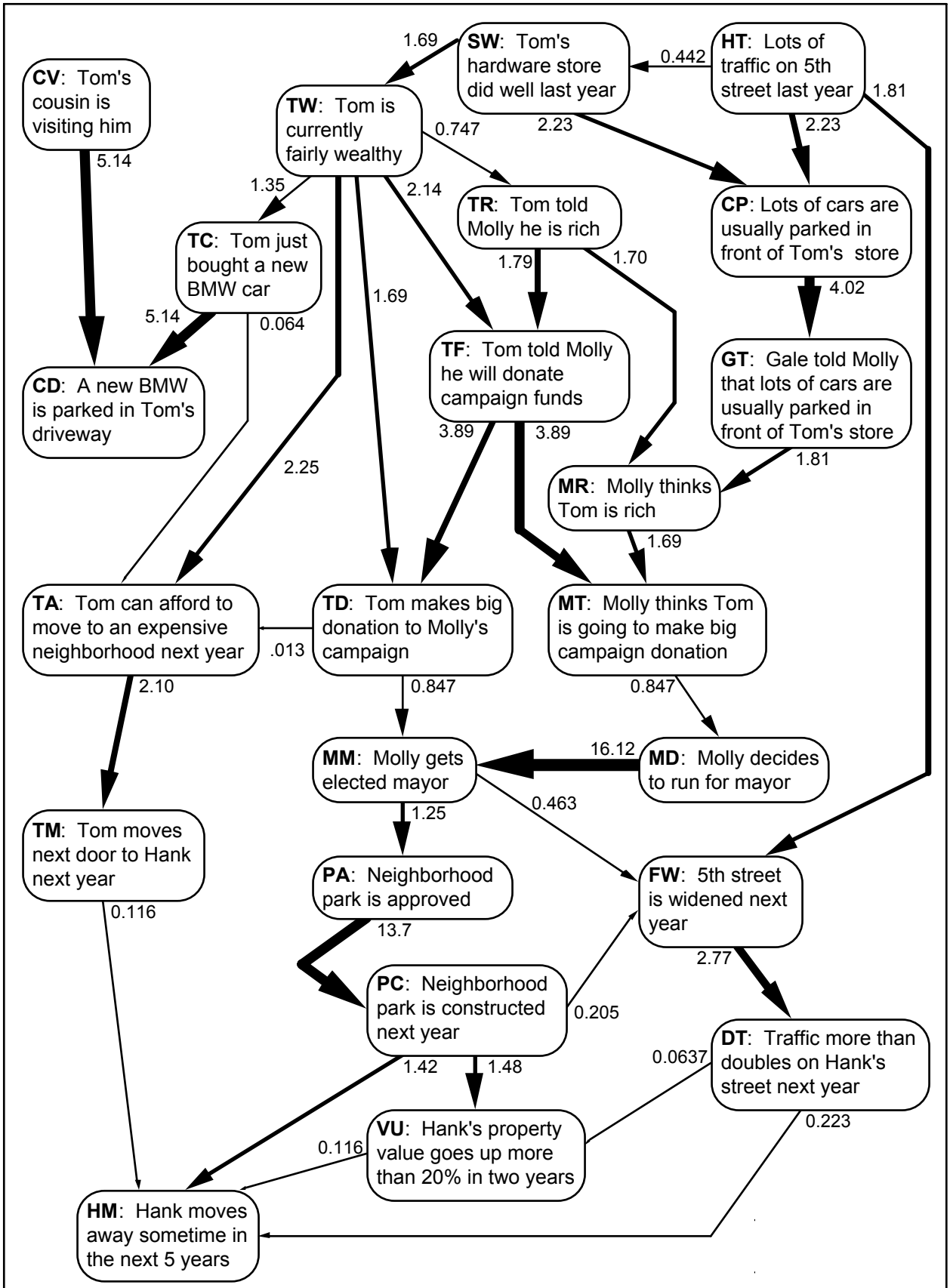


**Figure 5.1 -** Example BN showing link strengths. This is the same BN as in figure 2.5, except the links have been drawn thicker to indicate LS as: width = (2mm) * tanh (LS/4), with a 0.2mm minimum. The LS values also appear beside the link.

## 5.2 Connection Strength Contours

If there is a particular node for which evidence may arrive (called the *origin* node), we may draw an *iso-CS contour map* over the BN to indicate the maximum effect which that evidence can have on each of the other nodes. Each contour represents a particular value of CS, and is drawn so as to separate nodes on its one side which are more strongly connected to the origin node (i.e. have a greater CS), from those on its other side which are less strongly connected to the origin node (i.e. have a lesser CS). The purpose is for a human to quickly assess which nodes could be affected by the evidence, and by how much. It also helps to visualize the "neighborhood" of a node, and provide a sense of locality. Figure 5.2:1 shows an iso-CS contour map with the node SW as the origin node. Of course, the map would be different if some other node was the origin.

If $\Delta O$ connection strength is used to draw the contours, then the contour map may also be interpreted in another way. Instead of measuring the degree to which evidence at the origin node effects each of the other nodes, it can be interpreted to measure the degree to which evidence at each of the other nodes can effect the origin node. This is due to the commutivity of $\Delta O$ connection strength. One possible application of this is the following. We have a query node and we want to know which nodes to gather evidence at to best form a belief for the query node. Gathering evidence at a node that has very little effect on the query node is generally useless. So we can use the contours as indicators of levels of desirability for gathering evidence at each of the nodes (and trade that off with the cost of gathering evidence at that node).

Using the methods of the previous chapter, a contour map may be drawn that is based on the CS bounds calculated from link strengths, instead of the actual CS values. Such a contour map may be constructed very quickly (of complexity linear in the product of number of links and number of nodes). It may be used to immediately eliminate parts of the network as being irrelevant to some particular evidence, or some particular query, given a desired level of accuracy. Figure 5.2:2 shows such a contour map calculated for the node SW, using bounds produced by algorithm 4.6. It is interesting to compare it with figure 5.2:1, which is based on the exact CS values. For each node the actual CS value is less than the bound, as we would expect. For nodes close to SW the bound is very close to the actual value (equal for TW and HT), whereas for distant nodes the bound is less tight (greater by a factor of about 12 for the most distant node,

HM). It is interesting to notice that, at least for this example, even in areas where the bound is loose, the shapes of the contours for the bound are quite similar to the contours for the actual value.

This example does not show it well, but if the BN is composed of a dendritic skeleton of strong links with the "flesh" filled in by a network of weak links, then the contours will tend to follow the skeleton in a manner similar in appearance to the contours of a topographic map following the valleys of a dendritic river system, with the origin node at the point where the river meets the ocean.

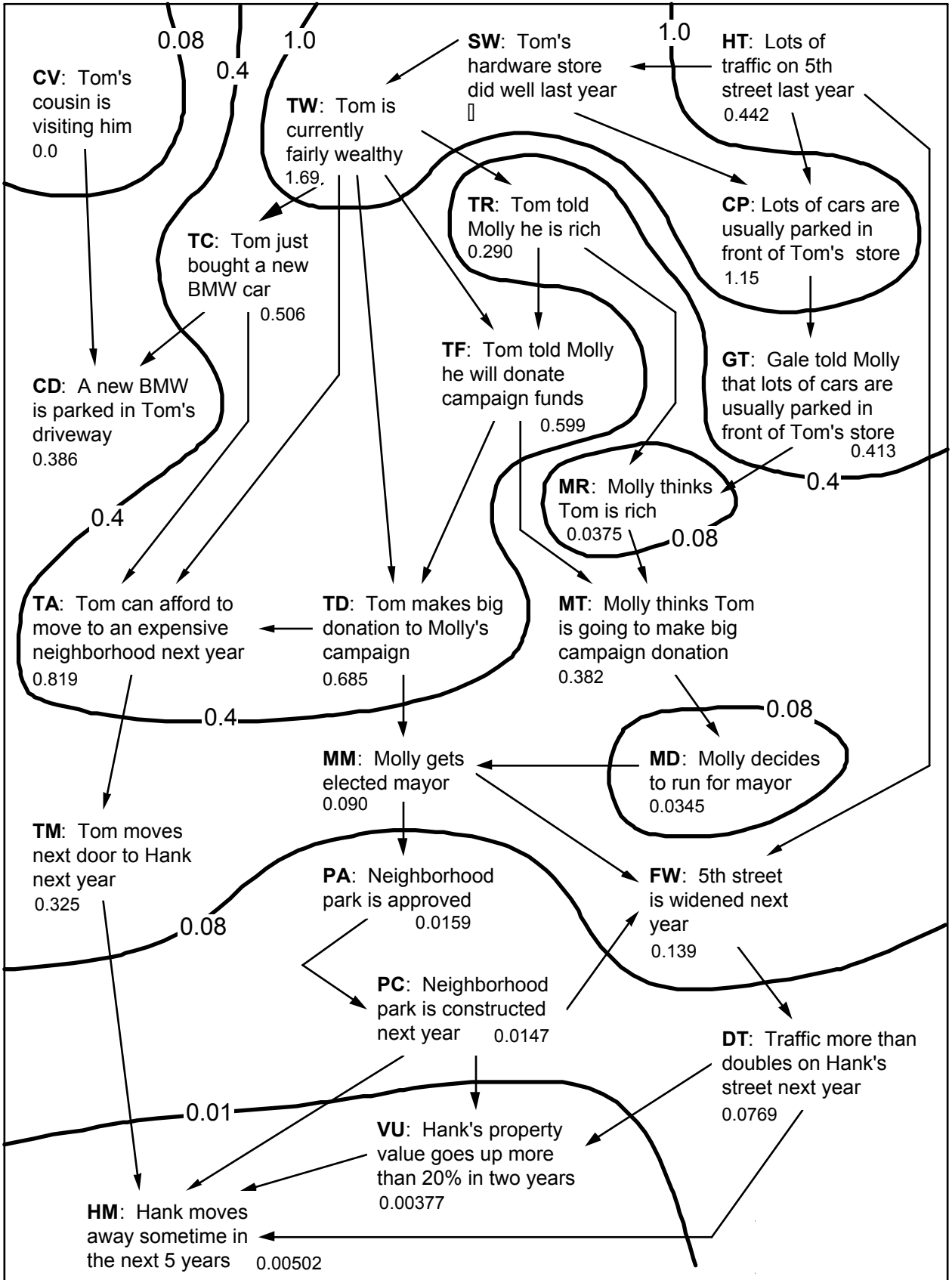


**Figure 5.2:1 -** Example BN showing the connection strength contours for the node SW. This is the same BN as in figure 5.1, with contours drawn to show the CS between each node and the node SW. The CS values appear below each node.

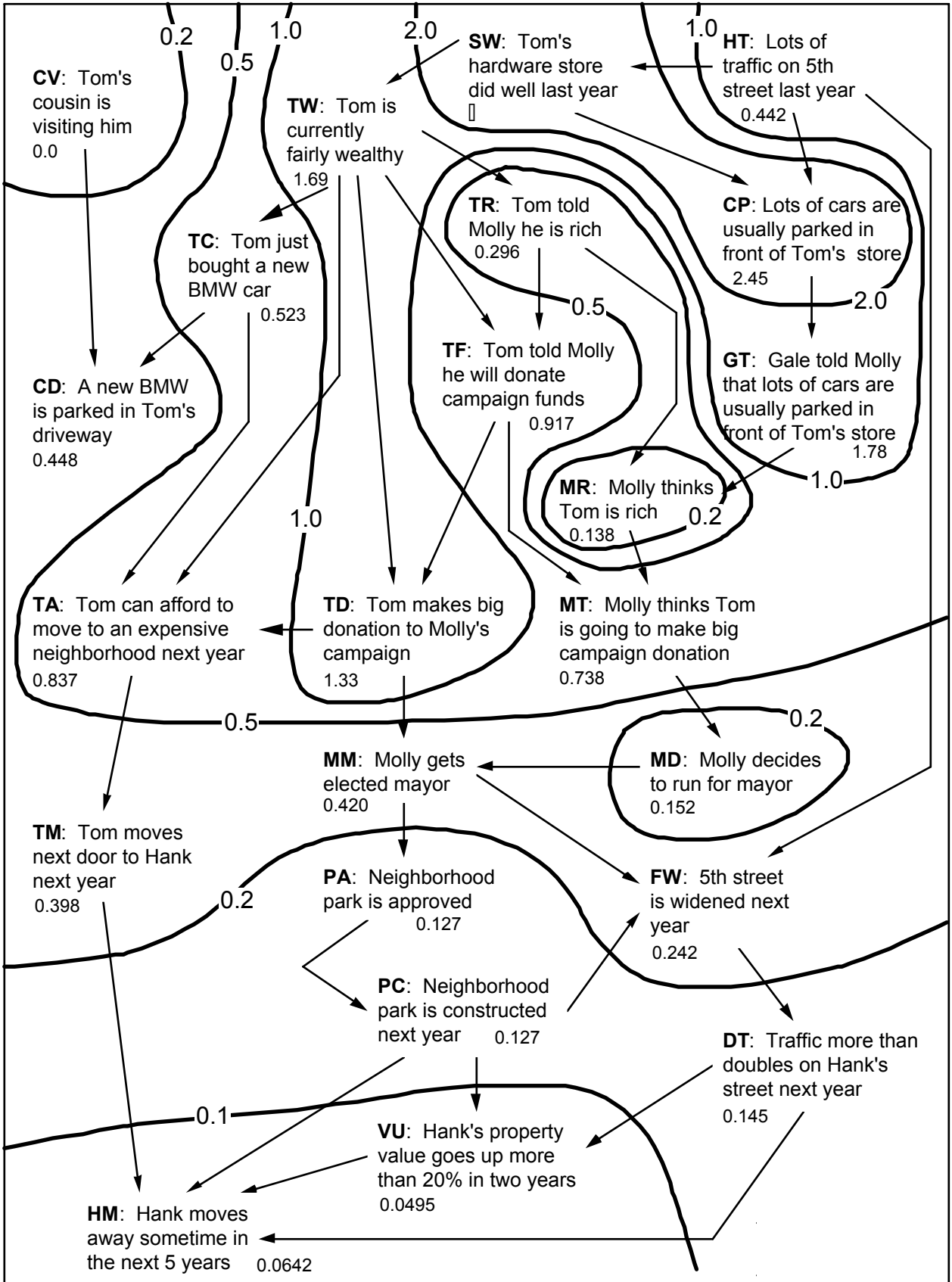


**Figure 5.2:2 -** The example BN of figure 5.1, with contours showing the connection strength bound for the node SW. The contours are drawn for the calculated CS bound between each node and the node SW. The CS bound values appear below each node.

## 5.3 Approximate Inference

Recall the "reasoning by assumptions" algorithm for BN inference, introduced in section 2.6. In it we instantiate some node (say node Z) to one of its values in order to simplify the network (generally to reduce the active path connectivity), do the BN inference with the simplified network, repeat the process with Z instantiated to its other value, and then combine the two solutions by taking their weighted average (weighted by the probabilities that Z would take on each of its two values).

But now suppose that node Z is distant enough from the nodes whose belief we wish to find (i.e. the query nodes), that instantiating Z to some value has almost no effect on their beliefs. In that case the two solutions will be almost the same, and their weighted average won't be that different from either one of them. So we could save some time by only computing one of the solutions, and recognizing that the beliefs that it provides are approximate. This suggests the following algorithm:

**Algorithm 5.3:1**: To compute the posterior probabilities of the nodes in the set **Q** given evidence **e** for the nodes in **E**, instantiate the node Z to one of its values, then use any standard BN inference algorithm to find $P(q_i|z,\mathbf{e})$, and finally consider it an approximation for $P(q_i|\mathbf{e})$, for all $Q_i \in \mathbf{Q}$. Node Z can be any node not in **E**, and will normally be chosen so that the particular BN inference algorithm to be used can find $P(q_i|z,\mathbf{e})$ more quickly than $P(q_i|\mathbf{e})$, which is often the case if, for example, Z blocks active paths from nodes in **Q** to their ancestors or nodes in **E**.

When we use an algorithm that produces approximate results, we usually need some kind of bound on how accurate we can expect the approximation to be. That is provided by the following theorem (which follows directly from theorem 3.4):

**Theorem 5.3:1**: When using algorithm 5.3.1, a bound on the error of the approximation, $e = d(P(q|z,\mathbf{e}), P(q|\mathbf{e}))$, is:

$$e \leq CS(Z, Q|\mathbf{e})$$

for any node $Q \in \mathbf{Q}$, where d is the distance measure used for the definition of CS.

Although instantiating one node may result in faster BN inference, usually we want to instantiate a set of nodes to block many active paths. There are two different advantages we could gain from this. We could make the network singly connected (or more singly connected), and/or we could prune off large parts of the network. For example consider the BN in figure 5.3:

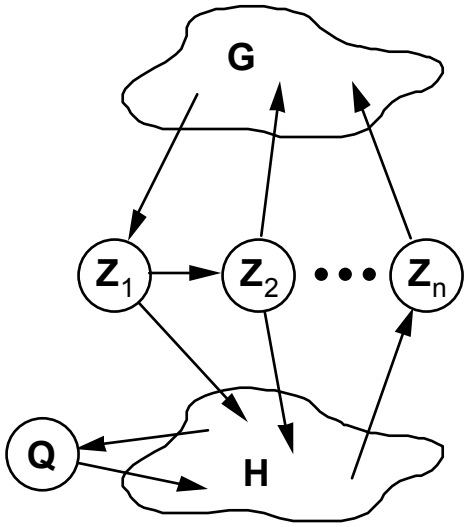


**Figure 5.3** - $Z_1$, $Z_2$,..., $Z_n$ block all active paths from subnetwork G to subnetwork H.

Instantiating the nodes $Z_1$, $Z_2$,..., $Z_n$ cuts off the subnetwork **G** from Q and its subnetwork **H**. So a BN inference algorithm finding the belief of Q can ignore the whole subnetwork **G**. Clearly, this may result in a major computational saving. In fact, the computational savings may be arbitrarly large, depending on the size and complexity of **G**. An approximation algorithm which instantiates multiple nodes is algorithm 5.3:2, which is essentially the same as algorithm 5.3:1, except **Z** is now a set of nodes.

**Algorithm 5.3:2**: To compute the posterior probabilities of the nodes in the set **Q** given evidence **e** for the nodes in **E**, instantiate all the nodes in **Z** to one of their values, creating the tuple of values **z**, then use some standard BN inference algorithm to find $P(q_i|\mathbf{z},\mathbf{e})$ and consider it an approximation for $P(q_i|\mathbf{e})$, for all $Q_i \in \mathbf{Q}$. The nodes in **Z** can be any nodes not in **E**, and will normally be chosen so that the particular BN inference algorithm to be used can find $P(q_i|\mathbf{z},\mathbf{e})$ more quickly than $P(q_i|\mathbf{e})$.

**Theorem 5.3:2**: When using algorithm 5.3:2, a bound on the error of the approximation:

$$e = d(P(q|z_1,z_2,...z_n,\mathbf{e}),\ P(q|\mathbf{e}))$$

is given by:

$$e \leq CS(Z_1, Q|\mathbf{e}) + CS(Z_2, Q|z_1,\mathbf{e}) + ... + CS(Z_n, Q|z_1,z_2,...,z_{n-1},\mathbf{e})$$

for any node $Q \in \mathbf{Q}$, where d is the distance measure used for the definition of CS.

This theorem is proved in Appendix C.

# 6 Conclusion

The major contributions of this thesis have been to introduce and explore connection and link strengths, to show the commutivity of $\Delta O$ connection strength, to find an algorithm (and its complexity) which determines connection strength bounds based only on link strength values and the BN graph topology, to explore the significance of paths in a BN, to provide an algorithm for approximate inference based on near independencies (and its error bound), to introduce and demonstrate the use of link strengths to display degree-of-independence on a BN diagram, and to introduce and demonstrate connection strength contour maps.

## 6.1 Further Work

### 6.1.1 Qualitative Probabilistic Networks

Algorithm 4.6 provides a way to calculate a bound on CS values, that is, it calculates the maximum magnitude of the change of belief at one node due to evidence at another. But we may also be interested in the direction of the change; do the beliefs increase or decrease? By removing the absolute value signs in the definition of $CS_o$ and $CS_p$ we can retain the information on the direction of the change. Equations 4.5:7 and 4.5:8, used by algorithm 4.6, must be modified to handle the signs of LS and CS values separately from the magnitudes. They must produce a number of the same magnitude as they do now, but the combination of signs must be as follows: A positive plus a positive is a positive, a negative plus a negative is a negative, a positive plus a negative is an unknown sign, and an unknown sign plus anything is an unknown

sign. For the serial combination rule: A positive times a positive is a positive, a positive times a negative is a negative, a negative times a negative is a positive, and an unknown times anything is an unknown.

In the above additions, the sign is handled separately from the magnitude, because actually interval arithmetic is being performed, where the intervals are always from zero to the positive or negative CS values (since they are actually *bounds*, the true value may be anywhere from zero to the bound). Exploiting this, we could modify equations 4.5:7 and 4.5:8 to produce a tighter bound in cases which involve the sum of a positive CS with a negative CS, since the magnitude of their sum will be bounded by the maximum of their magnitudes, which is less than the sum of their magnitudes (which is what must be used in the absence of sign information).

Wellman90 introduces *qualitative probabilistic networks*, which have the same dag structure as BNs, but contain only sign information along the links instead of NCPs (and may optionally contain hyperedges providing the sign of synergies, etc.). Their purpose is to predict in which direction a belief at one node will change given evidence at another node. It appears that adding sign information to CS as described above, will produce the same qualitative results as those of Wellman (while also providing magnitudes), but more work remains if it is desired to account for synergies in the way Wellman does.

### 6.1.2 Greater Computation for Tighter Bounds

This thesis presented a way to find a bound on CS, by combining quantities that could be calculated locally at the scale of a single link. But if we separate the nodes of a BN into small disjoint groups, then the interactions between two groups may be expressed solely in terms of the links between the nodes in the Markov boundaries of the two groups. For each group, we can do full BN reasoning to find exact connection strengths between the nodes of the group, then use the methods of this thesis to find bounds on CS between the nodes in the Markov boundaries of the different groups. That way we can find a bound on the CS between any two nodes of the original BN. The larger we make the groups, the tighter the bound will be, but the longer it will take to compute (because full BN inference will be required on larger groups). If the groups are so small they are just single nodes, the CS bounds calculated will be those of algorithm 4.6. If a

group is so large that it includes the whole BN, then the CS values will be exact. By varying the group size we will be provided with a continuum of CS bounding algorithms from completely global and slow algorithms which produce exact results, to completely local and fast algorithms which produce loose bounds.

### 6.1.3 Multistate Nodes

An obvious next step would be to try to extend the results of this thesis to BNs composed of nodes that can take on more than two values. Any multistate node BN can be modeled by a binary one by replacing each multistate node with a number of binary nodes, each binary node representing the proposition that the multistate node takes on one of its values. So, many of the proofs in this thesis will extend to BNs composed of multistate nodes, but whether practical algorithms and reasonably tight bounds can be produced has yet to be determined. Also, it may be desirable to generalize the definition of CS for multistate nodes in some other way.

# 7 Bibliography


Aczel, J. (1966) *Lectures on Functional Equations and Their Applications*, Academic Press, New York.

Agosta, John M. (1991) "'Conditional inter-causal independent' node distributions, a property of 'noisy-or' models" in *Uncertainty in Artificial Intelligence, Proc. of the Seventh Conf.* (July, UCLA), Bruce D'Ambrosio, et al, eds., Morgan Kaufmann, San Mateo, CA.

Boerlage, Brent (1992) Link Strength in Bayesian Networks, MSc Thesis, Dept. Computer Science, Univ. of British Columbia, BC.

Buchanan, Bruce G. and Edward H. Shortliffe (1984) *Rule Based Expert Systems: The MYCIN Experiments*, Addison-Wesley, Reading, MA.

Cheeseman, Peter (1986) "Probabilistic versus fuzzy reasoning" in *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmer (Eds.), North-Holland, Amsterdam, pp. 85-102.

Cooper, Gregory F. (1990) "The computational complexity of probabilistic inference using belief networks" in *Artificial Intelligence*, **42**, 393-405.

Heckerman, David E. (1986) "Probabilistic interpretations for MYCIN's certainty factors" in *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmer (Eds.), North-Holland, Amsterdam, pp. 167-196.

Henrion, Max (1989) "Some practical issues in constructing belief networks" in *Uncertainty in Artificial Intelligence 3*, Laveen N. Kanal, T. S. Levitt and J. F. Lemmer (Eds.), North-Holland, Amsterdam.

Henrion, Max and M. J. Druzdel (1990) "Qualitative propagation and scenerio-based explanation of probabilistic reasoning" in *Proc. of the Sixth Conf. on Uncertainty in Artificial Intelligence* (July 27-29, MIT), GE Corporate Research and Development.

Horvitz, Eric J. (1989) "Reasoning about beliefs and actions under computational resource constraints" in *Uncertainty in Artificial Intelligence 3*, Laveen N. Kanal, T. S. Levitt and J. F. Lemmer (Eds.), North-Holland, Amsterdam.

Howard, Ron A. (1971) *Dynamic Probabilistic Systems*, John Wiley & Sons, New York.

Jensen, Finn V., Kristian G. Olesen and Stig K. Andersen (1990) "An algebra of Bayesian belief universes for knowledge based systems" in Networks, **20**(5), 637-659.

Lauritzen, Steffen L. and David J. Spiegelhalter (1988) "Local computations with probabilities on graphical structures and their application to expert systems" in *J. Royal Statistics Society B*, **50**(2), 157-194.

Lindley, D. V. (1965) *Introduction to Probability and Statistics*, Cambridge University Press, Cambridge, MA.

Neapolitan, Richard E. and James R. Kenevan (1990) "Computation of variances in causal networks" in *Proc. of the Sixth Conf. on Uncertainty in Artificial Intelligence* (July 27-29, 1990 MIT), GE Corporate Research and Development.

Neapolitan, Richard E. (1990) *Probabilistic Reasoning in Expert Systems: Theory and Algorithms* , John Wiley & Sons, New York.

Neapolitan, R. and J. Kenevan (1991) "Investigation of variances in belief networks" in *Uncertainty in Artificial Intelligence, Proc. of the Seventh Conf.* (July, UCLA), Bruce D'Ambrosio, et al, eds., Morgan Kaufmann, San Mateo, CA.

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

Poole, David (1991) "Representing Bayesian networks within probabilistic Horn abduction" in *Uncertainty in Artificial Intelligence, Proc. of the Seventh Conf.* (July, UCLA), Bruce D'Ambrosio, et al, eds., Morgan Kaufmann, San Mateo, CA.

Suermondt, H. Jacques (1992) Explanation in Bayesian Belief Networks, PhD thesis (Report No. STAN-CS-92-1417), Departments of Computer Science and Medecine, Stanford Univ., CA.

Shachter, Ross D. (1986) "Evaluating influence diagrams" in *Operations Research*, **34**(6), 871-882.

Shachter, Ross D. (1988) "Probabilistic inference and influence diagrams" in *Operations Research*, **36**(4), 589-604.

Tribus, M. (1969) "What do we mean by rational?" in *Rational Descriptions, Decisions and Designs*, Pergamon Press, New York.

Wellman, Michael P. (1990) "Fundamental concepts of qualitative probabilistic networks" in *Artificial Intelligence*, **44**, 257-303.

Yule, G. Udny (1912) "On the methods of measuring association between two attributes" in *J. of the Royal Statistical Society*, **75**, 579-642.

Zolotarev, V. M. (1983) "Probability Metrics" in *Theory of Probability and its Applications*, **28**(2), 278-302.

# A Notation and Nomenclature

When naming nodes in a Bayesian net, upper case letters, such as "A", refer to single nodes, and bold upper case, such as "**A**", to a set of zero or more nodes. Since each node represents a propositional random variable, the names of the random variables are also denoted upper case, and the values that it can take on are labeled "TRUE" and "FALSE". "+a" denotes that the value of node A is TRUE, "¬a" denotes that the value of node A is FALSE, and "a" stands for the value of node A (TRUE or FALSE). Sometimes "+a" is written simply as "a", if that does not result in confusion. A vector of values for all the nodes in the set **E**, is written bold lower case, as "**e**".

Conditional probabilities are written in the form "P(+b|¬a,+c)", which means "the probability that B is TRUE, given that A is FALSE and C is TRUE". "O(+b|¬a,+c)" is the odds ratio that B is TRUE, given that A is FALSE and C is TRUE, i.e. O(+b|¬a,+c) = P(+b|¬a,+c) / P(¬b|¬a,+c). If we say "the belief at node B is x" we mean P(b=TRUE|**e**) = x, where **e** is the evidence seen so far.

"C(B)" denotes the set of parents (conditional predecessors) of node B, "$C^+$(B)" is the set of all ancestors of B, and "$C^*$(B)" is the set of all ancestors of B, including B. Likewise "S(B)" is the set of B's children (successors), and "$S^*$(B)" the set of all descendents. "$C^*$(**B**)", where **B** is a set of nodes, is defined as the set of all descendents of all the nodes in **B**. Likewise for $C^+$(**B**), $S^*$(**B**), etc.

A BN link from node A to node B is denoted as "A→B" or "B←A" or "$\overrightarrow{AB}$" or "$\overleftarrow{BA}$." If A is not a parent of B, the preceeding link notation may still be used, providing that adding a link from A to B does not create a cycle in the BN, and the link represented is considered a *null link* from A to B.

The Cartesian product is formed with the "Π" symbol, which takes a set of variables as its argument, and represents the set of all vectors formed with each of the variables taking on one of its possible values. As an example, for propositional variables A and B:

$$\Pi \{A, B\} = \{+a+b, +a\neg b, \neg a+b, \neg a\neg b\}.$$

# Abbreviations

BN - Bayesian net.

CS - Connection strength.

FJD - Full joint (probability) distribution. Consists of a probability for every conjunction consisting of all the primitive propositions (nodes). The term is also used to indicate the complete probabilistic model.

LS - Link strength.

NCP - Node conditional probability (ies). The probability of a node proposition being true, conditioned on its parents. Also called the "link matrix" in Pearl88.

# B Conventional Statistical Measures of Association

There is a broad array of standard statistical measures of association. Historically they have been defined simply by searching through equations to find one that meets certain desiderata (although in the last few decades there has been a move to define measures according to some optimality criterion). The following have been mentioned by the statistical community as desirable qualities for a measure of association:

1. Range: The measure of association should range from 0 to 1, or -1 to 1.
2. Endpoints: A measure of association of 0 should correspond to independence, while 1 should indicate full association (and -1 indicate reverse full association, if that value can be obtained). Full association is defined by some to mean deterministic dependence (all conditional probabilities of the contingency table are 0 or 1), and by others to mean that at least one of the conditional probabilities of the contingency table is 0 or 1.
3. Monotonicity: The measure of association should vary monotonically and continuously with P(xy)-P(x)P(y).
4. Symmetry: For two binary variables X and Y, the measure of association given as a function of P(x|y) and P(x|¬y), should be the same as the same function of P(y|x) and P(y|¬x).

Here are the most common measures of association as they would be applied to the case of two binary variables, X and Y, and written in the probabilistic notation used in this thesis.

Cross Ratio: $C = \frac{P(y|x)\,P(\neg y|\neg x)}{P(\neg y|x)\,P(y|\neg x)} = \frac{P(y|x)\,[1 - P(y|\neg x)]}{[1 - P(y|x)]\,P(y|\neg x)}$

Log cross ratio: $L = \log(C)$

Coefficient of association (Yule's Q): $Q = \frac{C - 1}{C + 1}$

Coefficient of colligation (Yule's Y): $Y = \frac{\sqrt{C} - 1}{\sqrt{C} + 1}$

Root mean square contingency: $r = \sqrt{\frac{\chi^2}{N}} = \sqrt{[P(x|y) - P(x|\neg y)]\,[P(y|x) - P(y|\neg x)]}$

Coefficient of contingency (Pearson): $c = r / \sqrt{r^2 + 1}$

Difference coefficient (J. H. Edwards): $E = P(y|x) - P(y|\neg x)$

Ratio coefficient (J. H. Edwards): $F = P(y|x) / P(y|\neg x)$

Mutual information: $I = P(x)\,P(y|x) \log \frac{P(y|x)}{P(y)} + (1 - P(x))\,P(y|\neg x) \log \frac{P(y|\neg x)}{P(y)} +$

$$P(x)\,(1 - P(y|x)) \log \frac{1 - P(y|x)}{1 - P(y)} +$$

$$(1 - P(x))\,(1 - P(y|\neg x)) \log \frac{1 - P(y|\neg x)}{1 - P(y)}$$

# C Proofs

## Theorem 3.1 - Equivalence of CS Definitions

**Theorem 3.1**: The following 3 definitions of CS are equivalent:

$$CS(A,B) = d(P(b|+a),\ P(b|\neg a)) \qquad 3.1{:}1$$

$$CS(A,B) = \max_{a_{v1}, a_{v2}} d(P(b|a_{v1}),\ P(b|a_{v2})) \qquad 3.1{:}3$$

$$CS(A,B) = \sup_{a_{v1}, a_{v2}} d(P(b|a_{v1}),\ P(b|a_{v1}, a_{v2})) \qquad 3.1{:}4$$

where $a_{v1}$ is some virtual (or nonexistent) evidence for A, $a_{v2}$ is other consistent evidence (possibly virtual) for A, "sup" means the least upper bound, and the distance measure d is assumed continuous for 3.1:4.

To prove that 3.1:3 and 3.1:4 are equivalent to 3.1:1, it is first useful to prove the following property of the distance measure:

**Lemma 3.1:5**: For the distance measure defined in 3.1:2 and any a, c, x, y $\in$ [0,1]:

$$\text{If} \quad a \le x \le c \quad \text{and} \quad a \le y \le c \quad \text{then} \quad d(x, y) \le d(a, c) \qquad 3.1{:}5$$

**Proof of lemma 3.1:5**:

By the monotonicity requirement on d:

$$a \le x \le c \quad \Rightarrow \quad d(a, x) \le d(a, c)$$

$$a \le y \le c \quad \Rightarrow \quad d(a, y) \le d(a, c)$$

Either: $a \le x \le y$ or $a \le y \le x$

So, by monotonicity and symmetry of d:

Either $d(x,y) \le d(a,y)$ or $d(y,x) = d(x,y) \le d(a,x)$

But both $d(a, y)$ and $d(a, x)$ are $\le d(a, c)$, so $d(x, y) \le d(a, c)$. ■

**Proof that 3.1:3 is equivalent to 3.1:1**:

Decomposing $P(b|a_{V1})$ on cases of A (by 2.3:3) we get:

$$P(b|a_{V1}) = P(b|a, a_{V1})\, P(a|a_{V1}) + P(b|\neg a, a_{V1})\, P(\neg a|a_{V1})$$

Since $a_{V1}$ is virtual evidence for A, B is independent of $a_{V1}$ given A. So $P(b|a,a_{V1}) = P(b|a)$. Also substitute $\alpha = P(a|a_{V1})$, to get:

$$P(b|a_{V1}) = P(b|a)\, \alpha + P(b|\neg a)\, (1 - \alpha)$$

$\alpha = P(a|a_{V1})$ is restricted to [0,1], and as it varies from 0 to 1, $P(b|a_{V1})$ will vary linearly from $P(b|\neg a)$ to $P(b|+a)$. So at all times it is bounded by these limits:

$$P(b|\neg a) \le P(b|a_{V1}) \le P(b|+a) \quad \text{or} \quad P(b|+a) \le P(b|a_{V1}) \le P(b|\neg a)$$

By an identical argument for $a_{V2}$:

$$P(b|\neg a) \le P(b|a_{V2}) \le P(b|+a) \quad \text{or} \quad P(b|+a) \le P(b|a_{V2}) \le P(b|\neg a)$$

By lemma 3.1:5:

$$d(P(b|a_{V1}),\ P(b|a_{V2})) \le d(P(b|+a),\ P(b|\neg a))$$

So CS(A,B) defined by 3.1:3 is always less than or equal CS(A,B) defined by 3.1:1. But in 3.1:3 the max runs over all values of $a_{V1}$ and $a_{V2}$, which includes the possibility $a_{V1} = +a$ and $a_{V2} =$

¬a, which will give it the value of CS(A,B) defined by 3.1:1, and therefore its maximum value. So each equation assigns the same value to CS(A,B). ■

**Proof that 3.1:4 is equivalent to 3.1:1**:

This proof is the same as the one above, with $a_{v1}, a_{v2}$ substituted for $a_{v2}$, except for the last paragraph, which becomes: CS(A,B) defined by 3.1:4 is always less than or equal CS(A,B) defined by 3.1:1. In 3.1:4 the max runs over all values of $a_{v1}$ and $a_{v2}$, which *doesn't* include the possibility $a_{v1} = +a$ and $a_{v1}, a_{v2} = \neg a$, because that would be inconsistent evidence.

However, $a_{v1}$ and $a_{v2}$ can come arbitrarily close to this, and since there are no discontinuities in the system (the probability equations are linear and the distance measure was required to be continuous for this proof), d (P(b|$a_{v1}$), P(b|$a_{v1}$, $a_{v2}$)) can come arbitrarily close to CS (A,B) defined by 3.1:1, with the appropriate choice of $a_{v1}$ and $a_{v2}$. By replacing "max" by "sup" to mean the lowest upper bound, we can write the expression as an equality, and have CS(A,B) defined by 3.1:4 exactly equivalent to CS(A,B) defined by 3.1:1. ■

# Theorem 3.4 - Alternate CS Definition

**Theorem 3.4**: If an alternate connection strength CS' is defined as:

$$CS'(A,B|\mathbf{e}) = \max\left(d(P(b|\mathbf{e}), P(b|+a,\mathbf{e}), d(P(b|\mathbf{e}), P(b|\neg a,\mathbf{e}))\right) \quad 3.4:1$$

then for any two propositional variables A and B, and for any evidence **e** (or no evidence **e**), connection strength defined by equation 3.4:1 can be bounded above and below as:

$$\frac{1}{2} CS(A,B|\mathbf{e}) \leq CS'(A,B|\mathbf{e}) \leq CS(A,B|\mathbf{e}) \quad 3.4:3$$

To prove the lower bound the following lemma is useful:

**Lemma 3.4**: For a distance measure, d, satisfying the triangle inequality (of 3.1:2):

$$\max(d(x,y), d(y,z)) \geq \frac{1}{2} d(x,z)$$

**Proof of lemma 3.4**:

Whether d(x,y) or d(y,z) is greater:

$$\max (d(x,y), d(y,z)) \geq \min (d(x,y), d(y,z))$$

Adding max (d(x,y), d(y,z)) to each side:

$$2 \max (d(x,y), d(y,z)) \geq \max (d(x,y), d(y,z)) + \min (d(x,y), d(y,z))$$

But:

$$\max (d(x,y), d(y,z)) + \min (d(x,y), d(y,z)) = d(x,y) + d(y,z)$$

By the triangle inequality of 3.1:2:

$$d(x,y) + d(y,z) \geq d(x,z)$$

Combining the above:

$$2 \max (d(x,y), d(y,z)) \geq d(x,y) + d(y,z) \geq d(x,z)$$

Dividing each side by 2:

$$\max (d(x,y), d(y,z)) \geq \frac{1}{2} d(x,z) \quad \blacksquare$$

**Proof of lower bound in 3.4:3**:

If we substitute P(b|+a,**e**) for x, P(b|**e**) for y, and P(b|¬a,**e**) for z in lemma 3.4, we obtain:

$$\max (d(P(b|+a,\mathbf{e}), P(b|\mathbf{e})), d(P(b|\mathbf{e}), P(b|\neg a,\mathbf{e}))) \geq \frac{1}{2} d(P(b|+a,\mathbf{e}), P(b|\neg a,\mathbf{e}))$$

By the symmetry of d (required by 3.1:2)

$$\max (d(P(b|\mathbf{e}), P(b|+a,\mathbf{e})), d(P(b|\mathbf{e}), P(b|\neg a,\mathbf{e}))) \geq \frac{1}{2} d(P(b|+a,\mathbf{e}), P(b|\neg a,\mathbf{e}))$$

Substituting, by the definition of CS (i.e. 3.1:1), and the definition of CS' (i.e. 3.4:1):

$$CS'(A,B|\mathbf{e}) \geq \frac{1}{2} CS(A,B|\mathbf{e}) \quad \blacksquare$$

**Proof of upper bound in 3.4:3**:

Decomposing P(b|**e**) on cases of A (by 2.3:3) we get:

$$P(b|\mathbf{e}) = P(b|+a,\mathbf{e})\,P(+a|\mathbf{e}) + P(b|\neg a,\mathbf{e})\,(1 - P(+a|\mathbf{e}))$$

P(+a|**e**) is restricted to [0,1], and as it varies from 0 to 1, P(b|**e**) will vary linearly from P(b|+a,**e**) to P(b|¬a,**e**). So at all times it is bounded by these limits:

$$P(b|+a,\mathbf{e}) \leq P(b|\mathbf{e}) \leq P(b|\neg a,\mathbf{e}) \quad \text{or} \quad P(b|\neg a,\mathbf{e}) \leq P(b|\mathbf{e}) \leq P(b|+a,\mathbf{e})$$

By the monotonicity requirement on d ( 3.1:2)

$$d(P(b|\mathbf{e}), P(b|+a,\mathbf{e})) \leq d(P(b|+a,\mathbf{e}), P(b|\neg a,\mathbf{e})) \quad \text{and}$$

$$d(P(b|\mathbf{e}), P(b|\neg a,\mathbf{e})) \leq d(P(b|+a,\mathbf{e}), P(b|\neg a,\mathbf{e}))$$

Since both left hand sides in the above are less than d(P(b|+a,**e**), P(b|¬a,**e**)), the maximum of them must also be less than d(P(b|+a,**e**), P(b|¬a,**e**)):

$$\max(d(P(b|+a,\mathbf{e}), P(b|\mathbf{e})),\ d(P(b|\mathbf{e}), P(b|\neg a,\mathbf{e}))) \leq d(P(b|+a,\mathbf{e}), P(b|\neg a,\mathbf{e}))$$

Substituting, by the definition of CS (i.e. 3.1:1), and the definition of CS' (i.e. 3.4:1):

$$CS'(A,B|\mathbf{e}) \leq CS(A,B|\mathbf{e}) \quad \blacksquare$$

# Theorem 3.7 - Commutivity of $CS_o$

**Theorem 3.7:3**: For any two propositional variables A and B, and any evidence **e**:

$$CS_o(A, B|\mathbf{e}) = CS_o(B, A|\mathbf{e}) \qquad 3.7:3$$

and from this it also follows that:

$$CS_o(A, B) = CS_o(B, A) \qquad 3.7:2$$

$$CSM_o(A, B|\mathbf{e}) = CSM_o(B, A|\mathbf{e}) \qquad 3.7:4$$

**Proof of 3.7:3 and 3.7:2**: By definition of CS (equations 3.5:1 and 3.3:3):

$$CS_o(A, B|\mathbf{e}) = |\log \frac{O(+b|+a, \mathbf{e})}{O(+b|\neg a, \mathbf{e})}|$$

By the definition of odds ratio (equation 3.3:1):

$$CS_o(A, B|\mathbf{e}) = |\log \frac{P(+b|+a, \mathbf{e})\, P(\neg b|\neg a, \mathbf{e})}{P(\neg b|+a, \mathbf{e})\, P(+b|\neg a, \mathbf{e})}|$$

By 4 applications of Bayes rule (equation 2.3:2):

$$CS_o(A, B|\mathbf{e}) = |\log \frac{P(+a|+b, \mathbf{e}) \frac{P(+b|\mathbf{e})}{P(+a|\mathbf{e})} P(\neg a|\neg b, \mathbf{e}) \frac{P(\neg b|\mathbf{e})}{P(\neg a|\mathbf{e})}}{P(+a|\neg b, \mathbf{e}) \frac{P(\neg b|\mathbf{e})}{P(+a|\mathbf{e})} P(\neg a|+b, \mathbf{e}) \frac{P(+b|\mathbf{e})}{P(\neg a|\mathbf{e})}}|$$

Canceling common factors:

$$CS_o(A, B|\mathbf{e}) = |\log \frac{P(+a|+b, \mathbf{e})\, P(\neg a|\neg b, \mathbf{e})}{P(+a|\neg b, \mathbf{e})\, P(\neg a|+b, \mathbf{e})}|$$

By the definition of odds ratio (equation 3.3:3):

$$CS_o(A, B|\mathbf{e}) = |\log \frac{O(+a|+b, \mathbf{e})}{O(+a|\neg b, \mathbf{e})}|$$

By definition of CS (equation 3.5:1):

$$CS_o(A, B|\mathbf{e}) = CS_o(B, A|\mathbf{e})$$

Which proves 3.7:3. Note that when $CS_o(A, B|\mathbf{e})$ is infinity (using infinity as described in section 3.3), the result still holds. There is a problem when $P(+a|\mathbf{e}) = 0$ or $P(\neg a|\mathbf{e}) = 0$ in the third step of the proof, since then these factors cannot be canceled from the numerator and denominator. However, in that case we simply define the $CS_o$ values in both directions as 0, since the node A is independent of all other nodes in the network (in one direction this follows from $d_o(0, 0) = 0$, and in the other it is equivalent to saying $P(b|+a) = P(b|\neg a)$ when one of them is undefined).

Equation 3.7:2 follows from the simple case when the evidence **e** is absent or irrelevant. ■

**Proof of 3.7:4**: By equation 3.7:3

For any $\mathbf{e}_+ \approx \mathbf{e}$: $CS_o(A, B|\mathbf{e}, \mathbf{e}_+) = CS_o(B, A|\mathbf{e}, \mathbf{e}_+)$

So:

$$\max_{\mathbf{e}_+ \approx \mathbf{e}} CS(A, B|\mathbf{e}, \mathbf{e}_+) = \max_{\mathbf{e}_+ \approx \mathbf{e}} CS(B, A|\mathbf{e}, \mathbf{e}_+)$$

But by equation 3.5:4:

$$\max_{\mathbf{e}_+ \approx \mathbf{e}} CS(A, B|\mathbf{e}, \mathbf{e}_+) = CSM_o(A, B|\mathbf{e})$$

So:

$CSM_o(A,B|\mathbf{e}) = CSM_o(B,A|\mathbf{e})$ ■

# Theorem 4.1 - ΔP Serial Chaining

**Theorem 4.1**: For any three propositional variables A, B, and C, where A and C are independent given B:

$$CS_p(A, C) = CS_p(A, B)\ CS_p(B, C) \qquad 4.1$$

**Proof of Theorem 4.1**: Using the reasoning-by-cases theorem (2.3:3):

$P(c|a) = P(c|a,+b)\ P(+b|a) + P(c|a,\neg b)\ P(\neg b|a)$

Since I(A,C|b), we know that $P(c|a,b) = P(c|b)$:

$P(c|a) = P(c|+b)\ P(+b|a) + P(c|\neg b)\ P(\neg b|a)$

If we subtract a version of this equation with a=FALSE, from a version of it with a=TRUE we obtain:

$P(c|+a) - P(c|\neg a) = P(c|+b)\ (P(+b|+a) - P(+b|\neg a)) + P(c|\neg b)\ (P(\neg b|+a) - P(\neg b|\neg a))$

Replacing $P(\neg b|a)$ with $1-P(+b|a)$, and simplifying, we obtain:

$$P(c|+a) - P(c|\neg a) \;=\; (P(c|+b) - P(c|\neg b))\;\; (P(+b|+a) - P(+b|\neg a))$$

Taking the absolute value of each side:

$$|\, P(c|+a) - P(c|\neg a)\,| \;=\; |\, P(c|+b) - P(c|\neg b)\,| \quad |\, P(+b|+a) - P(+b|\neg a)\,|$$

By the definition of ΔP connection strength this equation is equivalent to:

$$CS_p\,(A, C) \;=\; CS_p\,(A, B)\;\; CS_p\,(B, C) \quad \blacksquare$$

# Theorem 4.2 - ΔO Serial Chaining

**Theorem 4.2**: For any three propositional variables A, B, and C, where A and C are independent given B:

$$\tanh\left(\tfrac{1}{4}\; CS_o(A, C)\right) \;\leq\; \tanh\left(\tfrac{1}{4}\; CS_o(A, B)\right)\;\; \tanh\left(\tfrac{1}{4}\; CS_o(B, C)\right) \qquad 4.2{:}1$$

**Proof of Theorem 4.2**: Using the reasoning-by-cases theorem (2.3:3):

$$P(c|a) \;=\; P(c|a,+b)\;\; P(+b|a) \;+\; P(c|a,\neg b)\;\; P(\neg b|a)$$

Since I(A,C|b), we know that P(c|a,b) = P(c|b):

$$P(c|a) \;=\; P(c|+b)\;\; P(+b|a) \;+\; P(c|\neg b)\;\; P(\neg b|a)$$

Now we convert from probabilities to odds ratio, using O=P/(1–P), divide an A=TRUE version of the resulting equation with an A=FALSE version, and then simplify:

$$\frac{O(c|+a)}{O(c|\neg a)} \;=\; (1 + O(b|\neg a) + O(c|b) + O(b|\neg a)\; O(c|\neg b))$$
$$(O(b|a)\; O(c|b) + O(c|\neg b) + O(c|b)\; O(c|\neg b) + O(b|a)\; O(c|b)\; O(c|\neg b)) \;/$$
$$((1 + O(b|a) + O(c|b) + O(b|a)\; O(c|\neg b))$$
$$(O(b|\neg a)\; O(c|b) + O(c|\neg b) + O(c|b)\; O(c|\neg b) + O(b|\neg a)\; O(c|b)\; O(c|\neg b)))$$

We define O(x|+y)/O(x|¬y) as C(x|y), substitute where appropriate in the above equation, and simplify:

$$C(c|a) = (1 + C(c|b)\,O(b|a) + O(c|b) + O(b|a)\,O(c|b))$$
$$(C(b|a)\,C(c|b) + C(c|b)\,O(b|a) + C(b|a)\,C(c|b)\,O(c|b) + O(b|a)\,O(c|b)) /$$
$$((C(b|a) + C(c|b)\,O(b|a) + C(b|a)\,O(c|b) + O(b|a)\,O(c|b))$$
$$(C(c|b) + C(c|b)\,O(b|a) + C(c|b)\,O(c|b) + O(b|a)\,O(c|b)))$$

The above expression for C(c|a) can be broken down into the composition of two functions:

$$C(c|a) = \frac{(C(b|a)\,C(c|b)\,f1 + 1)\,(f1 + 1)}{(C(b|a)\,f1 + 1)\,(C(c|b)\,f1 + 1)} \qquad \text{C4.2:1}$$

where

$$f1 = \frac{1 + O(c|b)}{(C(c|b) + O(c|b))\,O(b|a)} \qquad \text{C4.2:2}$$

Now we must find the maximum value that C(c|a) can take for a given C(b|a) and C(c|b), so we maximize it with respect to O(b|a) and O(c|b). We are really interested in the maximums of $CS_o$, which is the *absolute value* of the log of C(c|a), so we want to find all maxima of C(c|a) which are greater than 1, *and* all minima less than 1. We assume C(b|a) and C(c|b) are finite and nonzero, and we will check the infinity and zero cases for maxima and minima at the end. First we maximize with respect to O(b|a), then with respect to O(c|b).

First we check the boundaries of O(b|a). At O(b|a) = 0 and O(b|a) = ∞, we get C(c|a) = 1, so there are no maxima or minima of interest at the boundaries. Next we check for discontinuities (i.e. the numerator or denominator equals zero). All the quantities in f1 are greater than or equal 0, so f1 = (1 + O(c|b)) / (( C(c|b) + O(c|b)) O(b|a)) is greater than or equal 0. The denominator of C(c|a) is (C(b|a) f1 + 1) ( C(c|b) f1 + 1) which therefore must be 1 or greater. The numerator of C(c|a) is ( C(b|a) C(c|b) f1 + 1) (f1 + 1) which must also be 1 or greater. So there are no discontinuities in our domain of interest. Also, since it is a "rational" function, the derivative won't have discontinuities in places where the function doesn't.

Next we take the derivative with respect to O(b|a), and check the zeros of the derivative for maxima. We do this in two steps using the chain rule for derivatives:

$$C(c|a) = g1(C(b|a),\ C(c|b),\ f1(C(c|b), O(b|a), O(c|b)))$$

$$\frac{d[C(c|a)]}{d[O(b|a)]} = \frac{d[C(c|a)]}{d[f1]}\,\frac{d[f1]}{d[O(b|a)]}$$

So the maxima/minima will occur when either derivative equals zero.

$$\frac{d[f1]}{d[O(b|a)]} = -\frac{1 + O(c|b)}{(O(b|a))^2 (C(c|b) + O(c|b))}$$

This is zero when O(b|a) is infinity (which is at the boundary and has already been checked). It is not zero when O(c|b) is infinity (there it is -1). It is zero when O(c|b) = -1, but that is not an allowed value for O(c|b), so there are no maxima of interest when this derivative is zero. We try the other derivative in the product:

$$\frac{d[C(c|a)]}{d[f1]} = -\frac{(C(b|a) - 1) (C(c|b) - 1) (C(b|a) C(c|b) f1^2 - 1)}{(1 + C(b|a) f1)^2 (1 + C(c|b) f1)^2}$$

Zeros are of this derivative are at $C(b|a) = 1$, $C(c|b) = 1$, $f1 = \infty$, and $C(b|a) C(c|b) f1^2 = 1$. When $C(b|a) = 1$ or $C(c|b) = 1$, we find that $C(c|a) = 1$, so they do not correspond to maxima of interest. f1 is infinite only if $O(b|a) = 0$ (which we have already considered). So the only interesting roots are at $C(b|a) C(c|b) f1^2 = 1$. One of the roots of this equation is always negative, so it doesn't correspond to a valid solution. The other root is:

$$f1_m = \frac{\sqrt{C(b|a) C(c|b)} (1 + O(c|b))}{C(c|b) + O(c|b)}$$

Substituting this back in equation C4.2:1, yields:

$$C(c|a)_m = \left[\frac{1 + \sqrt{C(b|a)} \sqrt{C(c|b)}}{\sqrt{C(b|a)} + \sqrt{C(c|b)}}\right]^2 \qquad \text{C4.2:4}$$

which is a suitable maxima.

We have left to the end the task of checking for possible maxima or minima at $C(b|a)$ or $C(c|b)$ taking a value of 0 or $\infty$. When we substitute $C(b|a) = 0$ into the formula for $C(c|a)$ we get $C(c|a) = 1/ C(c|b)$ and for $C(b|a) = \infty$ we get $C(c|a) = C(c|b)$. These are the same values we get from C4.2:4, so that formula will do in all cases. So the value of C(c|a) given by equation C4.2:4 truly is the maximum value C(c|a) can take given values of C(b|a) and C(c|b), and we may rewrite it as an inequality which always holds:

$$C(c|a) \leq \left[\frac{1+\sqrt{C(b|a)}\sqrt{C(c|b)}}{\sqrt{C(b|a)}+\sqrt{C(c|b)}}\right]^2 \qquad \text{C4.2:5}$$

We can use equation 3.3:3 to express things in terms of $CS_o$:

$$CS_o(A, C) = |\log C(c|a)|$$

$$CS_o(A, B) = |\log C(b|a)|$$

$$CS_o(B, C) = |\log C(c|b)|$$

If we make the above substitutions in equation C4.2:5, and then simplify, we obtain:

$$\tanh\left(\tfrac{1}{4} CS_o(A, C)\right) \leq \tanh\left(\tfrac{1}{4} CS_o(A, B)\right) \tanh\left(\tfrac{1}{4} CS_o(B, C)\right)$$

which is the equation to be proved. ■

# Theorem 4.3 - Fundamental Equation

**Theorem 4.3**: For any three propositional variables V, Q, and Z, we can decompose the connection strength from V to Q on cases of Z as follows:

$$CS(V, Q) \leq \max_z CS(V, Q|z) + CS(V, Z) * \min_v CS(Z,Q|v)$$

where * is a generalized multiplication corresponding to the serial combination rule for the particular distance measure, d, used to define CS.

**Lemma 4.3**: For any real numbers $x_1$, $x_2$, $y_1$, $y_2$, and z, if

$$z \leq x_1 + y_1 \qquad \text{and} \qquad z \leq x_2 + y_2$$

then

$$z \leq \max(x_1, x_2) + \min(y_1, y_2)$$

**Proof of Lemma 4.3**:

There are four possible cases. If $x_1 \geq x_2$ and $y_1 \leq y_2$, then

$$\max(x_1, x_2) + \min(y_1, y_2) = x_1 + y_1 \geq z \quad \text{by the first given equation}$$

If $x_1 \geq x_2$ and $y_1 \geq y_2$, then

$$\max(x_1, x_2) + \min(y_1, y_2) = x_1 + y_2 \geq x_2 + y_2 \geq z \quad \text{by the second given equation}$$

and similarily for the other two cases. ■

**Proof of Theorem 4.3**:

We can analyze the propositional variables V, Q, and Z by constructing a BN for them, which appears in figure C.1. Since this is a fully connected BN, it can represent any probabilistic relationship between the 3 variables, with the right choice of NCP values. So even if these three variables are originally from a different BN where they are connected up in a different way, and perhaps with many other nodes involved, the BN of figure C.1 can represent their relationships within the other BN with no loss of generality. So we need only prove equation 4.3:6 for the BN of figure C.1.

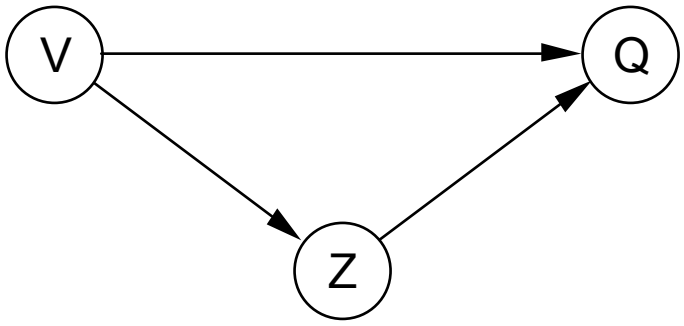


**Figure C.1** - Fully connected BN representing the probabilistic relationship between V, Z, and Q with no loss of generality.

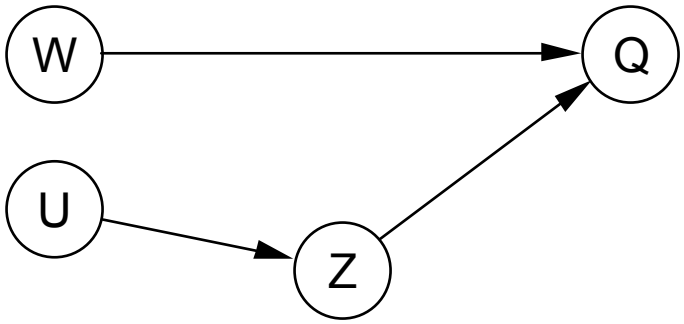


**Figure C.2** - A new BN which is the same as the one in figure C.1, except the V node has been split into U and W. This BN is defined to have the same NCPs as the one in figure C.1.

The BN in figure C.2 is the same as C.1 (has the same connections and the same NCPs), except the node V has been replaced with two nodes: U and W. In general C.2 will produce different inference results from those produced by C.1, because the active path from Q to Z through V has been broken. However, in those cases where U and W both have evidence, and that evidence is the same for both of them, then inference using C.2 will produce the same beliefs as C.1 (with V receiving the same evidence as U and W). Since in the calculation of CS(V,Q) the only values of interest are the beliefs at Q when V has evidence, we will obtain the same values from C.2 (giving both U and W the same evidence as V). Even though some of the intermediate calculations may be different, the results will be the same because $P(q|+v) = P(q|+u,+w)$ and $P(q|\neg v) = P(q|\neg u,\neg w)$. So we need only prove equation 4.3:6 for the BN of figure C.2.

By the definition of CS (3.1:1):

$$CS(V,Q) = d(P(q|+v), P(q|\neg v)) = d(P(q|+u,+w), P(q|\neg u,\neg w))$$

By the triangle inequality of d (3.2:2):

$$d(P(q|+u,+w), P(q|\neg u,\neg w)) \leq d(P(q|+u,+w), P(q|+u,\neg w)) + d(P(q|+u,\neg w), P(q|\neg u,\neg w))$$

Substituting in the above the definition of conditional CS (3.5:1):

$$CS(V,Q) \leq CS(W,Q|+u) + CS(U,Q|\neg w)$$

Similarily we can derive:

$$CS(V,Q) \leq CS(W,Q|\neg u) + CS(U,Q|+w)$$

Invoking lemma 4.3 on the two equations above:

$$CS(V,Q) \leq \max_u CS(W,Q|u) + \min_w CS(U,Q|w) \qquad \text{C4.3:1}$$

The second term in the above equation can be evaluated as a three node BN consisting of two links in serial, similar in form to the BN of figure 4.1. The techniques of section 4.1 or section 4.2 can be used (or similar techniques for other distance measures) to provide a bound for it. We express that bound in the general form:

$$\min_w CS(U,Q|w) \leq \min_w [CS(U,Z|w) * CS(Z,Q|w)]$$

where * is the serial combination rule, which depends on the particular distance measure being used. Since W is independent of U and Z, $CS(U,Z|w) = CS(U,Z)$, and the * operator is monotonically increasing w.r.t. both arguments, we may move the min operator in:

$$\min_w CS(U,Q|w) \leq CS(U,Z) * \min_w CS(Z,Q|w)$$

Returning to notation using V instead of U and W:

$$\min_w CS(U,Q|w) \leq CS(V,Z) * \min_v CS(Z,Q|v) \qquad \text{C4.3:2}$$

Now we examine the first term of C4.3:1. Expressing it in terms of the d measure:

$$\max_u CS(W,Q|u) = \max_u d(P(q|+w,u), P(q|\neg w,u)) \qquad \text{C4.3:3}$$

We evaluate $d(P(q|+w,u), P(q|\neg w,u))$ by reasoning by cases on Z (2.3:3):

$$d(P(q|+w,u), P(q|\neg w,u)) = d(P(q|+w,+z,u)\,P(+z|+w,u) + P(q|+w,\neg z,u)\,P(\neg z|+w,u),\; P(q|\neg w,+z,u)\,P(+z|\neg w,u) + P(q|\neg w,\neg z,u)\,P(\neg z|\neg w,u))$$

Q is independent of U given Z, so $P(q|w,z,u) = P(q|w,z)$. Also, Z is independent of W given U so $P(z|w,u) = P(z|u)$. Substituting these in:

$$d(P(q|+w,u), P(q|\neg w,u)) = d(P(q|+w,+z)\,P(+z|u) + P(q|+w,\neg z)\,P(\neg z|u),\; P(q|\neg w,+z)\,P(+z|u) + P(q|\neg w,\neg z)\,P(\neg z|u))$$

Substituting $\lambda$ for $P(+z|u)$, and $1-\lambda$ for $P(\neg z|u)$, we obtain:

$$d(P(q|+w,u), P(q|\neg w,u)) = d(P(q|+w,+z)\,\lambda + P(q|+w,\neg z)\,(1-\lambda),\ P(q|\neg w,+z)\,\lambda + P(q|\neg w,\neg z)\,(1-\lambda))$$

By the no-maxima requirement on d (3.1:2):

$$d(P(q|+w,u), P(q|\neg w,u)) \leq \max\,(d(P(q|+w,+z), P(q|\neg w,+z)), d(P(q|+w,\neg z), P(q|\neg w,\neg z))$$

By the definition of conditional CS (3.5:1):

$$d(P(q|+w,u), P(q|\neg w,u)) \leq \max_z CS(W,Q|z)$$

Substituting it back into C4.3:3, and switching W to V notation, we obtain:

$$\max_u CS(W,Q|u) \leq \max_z CS(V,Q|z) \qquad \text{C4.3:4}$$

Combining C4.3:1, C4.3:2, and C4.3:4, gives us the result to be proved:

$$CS(V,Q) \leq \max_z CS(V,Q|z) + CS(V,Z) * \min_v CS(Z,Q|v) \quad \blacksquare$$

# Theorem 4.6 - Path Complexity

**Lemma 4.6:** When calculating each new F(J,Q) in step 2 of algorithm 4.6, all the subcalculations of F(K,Q) that are required, will already be calculated. The same holds for step 3.

**Theorem 4.6**: The number of generalized multiplications required to find a bound on $CS_o(V,Q)$ using algorithm 4.6, is the number of links between the ancestors of Q which are also descendants of ancestors of V, plus the number of links between the ancestors of V which are also descendants of ancestors of Q, that is:

$$\text{Number multiplies} = \text{Number links between } (S^*(C^*(Q)) \cap C^*(V)) + \text{Number links between } (S^*(C^*(V)) \cap C^*(Q))$$

**Proof of Lemma 4.6:** Each time we use equation 4.5:8 to find a value for F(X,Y), we end up finding values for F(X,Y), and each F(K,Y) where $K \in S(X) \cap C^*(Y)$ (whether or not X=Y). If we apply the equation recursively, we end up finding F(K,Y) values for all $K \in \mathbf{K}$, where **K** is given by:

$$\mathbf{K} = X \cup (S(X) \cap C^*(Y)) \cup (S(S(X) \cap C^*(Y)) \cap C^*(Y)) \cup \ldots \qquad 4.6:3$$

It is a property of ancestor/descendent relations, that if a node W isn't an ancestor of another node Z, it can't have a successor or descendent which is an ancestor of W. That is:

$$W \notin C^*(Z) \quad \text{implies} \quad S(W) \notin C^*(Z) \quad \text{and} \quad S^*(W) \cap C^*(Z) = \emptyset \qquad 4.6:4$$

Using 4.6:4, we can simplify 4.6:3 to:

$$\mathbf{K} = S^*(X) \cap C^*(Y) \qquad 4.6:5$$

When we use equation 4.5:7 to find a bound on CS(V,Q), we have to find a value of F(J,Q) for all $J \in C^*(V) \cap C^+(Q)$. Since we will find these values using equation 4.5:8, that requires using equation 4.5:8 to find F(K,Q) for all $K \in \mathbf{K_Q}$ where $\mathbf{K_Q}$ is given by 4.6:5, substituting $C^*(V) \cap C^+(Q)$ for X, and Q for Y:

$$\mathbf{K_Q} = S^*(C^*(V) \cap C^+(Q)) \cap C^*(Q) \qquad 4.6:6$$

We can simplify the above using 4.6:4 to get:

$$\mathbf{K_Q} = S^*(C^*(V)) \cap C^*(Q) \qquad 4.6:7$$

So when finding all the F(J,Q) values in step 2, all the recursive F(K,Q) values that need to be found will be in the set above. Since step 2 specifies that we find the F(J,Q) values in reverse order of J, and equation 4.5:8 finds F(J,Q) values using only F(K,Q) values for which K succeeds J, completion of step 2 will require only F(K,Q) values from the set above which have already been found.

The same type of argument, but with V substituted for Q, and Q for V, serves to show that step 3 requires finding only F(K,V) values in which K is an element of the set $\mathbf{K_V}$ below:

$$\mathbf{K_V} = S^*(C^*(Q)) \cap C^*(V) \qquad 4.6:8$$

Furthermore, step 3 also specifies that we find the F(J,V) values in reverse order of J, so by the same reasoning as the step 2 case, completion of step 3 will require only F(K,V) values from the set above which have already been found. ■

**Proof of Theorem 4.6**: Equation 4.5:8 is invoked to find each value of F(X,Q) where $X \in S^*(C^*(V)) \cap C^*(Q)$ as stated by equation 4.6:7. Each time it is invoked it performs one multiply for each link leaving X and going to a node in C*(Q). Since the node that the link goes to will be in S*(C*(V)) as well (because by definition it must go to a successor), we can say that one multiply is performed for each link leaving a node in $S^*(C^*(V)) \cap C^*(Q)$ and going to a node in $S^*(C^*(V)) \cap C^*(Q)$. Or, in other words, the number of multiplies required is the number of links between the nodes in $S^*(C^*(V)) \cap C^*(Q)$.

Equation 4.5:8 must also be invoked to find each value of F(X,V) where $X \in S^*(C^*(Q)) \cap C^*(V)$ as stated by equation 4.6:8. By reasoning similar to the last paragraph, we can say that the number of multiplies required to do this is the number of links between the nodes in $S^*(C^*(Q)) \cap C^*(V)$.

Finally, we must add on the number of multiplies required by equation 4.5:7 in step 4. This will be $|C^*(V) \cap C^+(Q)|$, which is one multiply for each term of the sum. However, if we don't count all those multiplications that are with a connection strength from a node to itself, done in steps 2 and 3 above (which don't really need to be done, since they yield the original number, and are present simply to terminate the recursion), then this quantity will be absorbed, leaving us with the sum of the two quantities from the two paragraphs above:

Number multiplies = Number links between ( $S^*(C^*(Q)) \cap C^*(V)$ ) +
Number links between ($S^*(C^*(V)) \cap C^*(Q)$)

■

# Theorem 4.7:1 - Intercausal Link Strength

**Theorem 4.7:1**: If A and C are both parents of E, and I(A,C|), then if E receives evidence TRUE, the CS from A to C is bounded by:

$$CS_o(A,C|+e) \;=\; |\log \frac{P(+e|+a+c)\,P(+e|\neg a\neg c)}{P(+e|+a\neg c)\,P(+e|\neg a+c)}| \qquad 4.7{:}1$$

**Proof of Theorem 4.7:1**: By Bayes rule (2.3:2):

$$P(c|a,e) \;=\; P(e|a,c)\,\frac{P(c|a)}{P(e|a)}$$

If we form the ratio of this equation with c=TRUE, to it with c=FALSE we obtain:

$$\frac{P(+c|a,e)}{P(\neg c|a,e)} \;=\; \frac{P(e|a,+c)}{P(e|a,\neg c)}\;\frac{P(+c|a)\,P(e|a)}{P(\neg c|a)\,P(e|a)}$$

Canceling P(e|a) and using the definition of odds ratio, we obtain the following well known (e.g. Pearl88) equation, which holds for either value of A and either value of E:

$$O(+c|a,e) \;=\; \frac{P(e|a,+c)}{P(e|a,\neg c)}\;O(+c|a)$$

If we form the ratio of this equation with a=TRUE, to it with a=FALSE we obtain:

$$\frac{O(+c|+a,e)}{O(+c|\neg a,e)} \;=\; \frac{P(e|+a,+c)\,P(e|\neg a,\neg c)}{P(e|+a,\neg c)\,P(e|\neg a,+c)}\;\frac{O(+c|+a)}{O(+c|\neg a)}$$

Since I(A,C|), we know that $O(+c|+a) = O(+c|\neg a)$, so we can cancel these factors.

$$\frac{O(+c|+a,e)}{O(+c|\neg a,e)} \;=\; \frac{P(e|+a,+c)\,P(e|\neg a,\neg c)}{P(e|+a,\neg c)\,P(e|\neg a,+c)}$$

Next we take the absolute value of the logarithm of both sides.

$$|\log \frac{O(+c|+a,e)}{O(+c|\neg a,e)}| \;=\; |\log \frac{P(e|+a,+c)\,P(e|\neg a,\neg c)}{P(e|+a,\neg c)\,P(e|\neg a,+c)}|$$

By the definition of CS using the $d_o$ measure, the left hand side is CS(A,C|e):

$$CS(A,C|e) \;=\; |\log \frac{P(e|+a,+c)\,P(e|\neg a,\neg c)}{P(e|+a,\neg c)\,P(e|\neg a,+c)}|$$

The equation we set out to prove is just a special case of the above with e=TRUE. ■

# Theorem 5.3:2 - Approx. Inference Error Bound

**Theorem 5.3:2**: When using algorithm 5.3:2, a bound on the error of the approximation:

$$e = d(P(q|z_1,z_2,...z_n,\mathbf{e}), P(q|\mathbf{e}))$$

is given by:

$$e \leq CS(Z_1, Q|\mathbf{e}) + CS(Z_2, Q|z_1,\mathbf{e}) + ... + CS(Z_n, Q|z_1,z_2,...,z_{n-1},\mathbf{e})$$

for any node $Q \in \mathbf{Q}$, where d is the distance measure used for the definition of CS.

**Lemma 5.3**: For any two propositional variables, Z and Q, and any evidence **e**:

$$CS(Z,Q|\mathbf{e}) \geq d(P(q|\mathbf{e}), P(q|z,\mathbf{e}))$$

**Proof of Lemma 5.3**: By the definition for an alternate CS (3.4:1):

$$CS'(Z,Q|\mathbf{e}) = \max(d(P(q|\mathbf{e}), P(q|+z,\mathbf{e})), d(P(q|\mathbf{e}), P(q|\neg z,\mathbf{e})))$$

The result of a "max" function is greater than either of its arguments, so:

$$CS'(Z,Q|\mathbf{e}) \geq d(P(q|\mathbf{e}), P(q|z,\mathbf{e}))$$

By theorem 3.4, $CS(Z,Q|\mathbf{e}) \geq CS'(Z,Q|\mathbf{e})$, so:

$$CS(Z,Q|\mathbf{e}) \geq d(P(q|\mathbf{e}), P(q|z,\mathbf{e}))$$ ■

**Proof of Theorem 5.3:2**: By the triangle inequality on d (required by 3.1:2):

$$d(P(q|\mathbf{e}), P(q|z_1,...,z_{n-1},\mathbf{e})) + d(P(q|z_1,...,z_{n-1},\mathbf{e}), P(q|z_1,...,z_n,\mathbf{e})) \geq d(P(q|\mathbf{e}), P(q|z_1,...,z_n,\mathbf{e}))$$

The first term of the above can be substituted with an upper bound (i.e. a number guaranteed to be greater or equal) provided by the same equation with an index of n that is one lower, to yield.

$$d(P(q|\mathbf{e}), P(q|z_1,...,z_{n-2},\mathbf{e})) + d(P(q|z_1,...,z_{n-2},\mathbf{e}), P(q|z_1,...,z_{n-1},\mathbf{e})) + d(P(q|z_1,...,z_{n-1},\mathbf{e}), P(q|z_1,...,z_n,\mathbf{e})) \geq d(P(q|\mathbf{e}), P(q|z_1,...,z_n,\mathbf{e}))$$

This process can be repeated n-2 times to yield:

$$\sum_{i=1}^{n} d(P(q|z_1,...,z_{i-1},\mathbf{e}), P(q|z_1,...,z_i,\mathbf{e})) \geq d(P(q|\mathbf{e}), P(q|z_1,...,z_n,\mathbf{e}))$$

Each term of the sum can be bounded by lemma 5.3, by substituting $Z_i$ for Z and $z_1 \& ... \& z_{i-1} \& \mathbf{e}$ for **e**, to yield:

$$\sum_{i=1}^{n} CS(Z_i,Q|z_1,...,z_{i-1},\mathbf{e}) \geq d(P(q|\mathbf{e}), P(q|z_1,...,z_n,\mathbf{e}))$$

If we define: $e = d(P(q|z_1,z_2,...z_n,\mathbf{e}), P(q|\mathbf{e}))$, then we obtain the bound:

$$e \leq CS(Z_1, Q|\mathbf{e}) + CS(Z_2, Q|z_1,\mathbf{e}) + ... + CS(Z_n, Q|z_1,z_2,...,z_{n-1},\mathbf{e}) \quad \blacksquare$$