charles river analytics

Beyond Rules:

Getting More Out of Data with Belief Networks

Because Belief Networks greatly condense the information and relationships needed to create a decision framework, they can be applied in situations previously too complex to be automated with rules-based expert and decision-support systems.

This paper describes Belief Networks, the next step beyond rules-based expert systems. It covers how Belief Networks can turn missing or uncertain data into valuable, actionable information. It starts by addressing the rigidities and associated limitations of rule-based systems, and how Belief Networks condense the needed information and relationships for decision-making. The paper concludes with what you can do to start bringing Belief Networks to work for your organization.

If you are responsible for purchasing and implementing Expert Systems in your company, read this paper to understand how Belief Networks can expand the strategic benefits of automated systems and decision support systems into areas that are too complex to benefit from business rules.

If you are a software vendor or consultant read this paper to understand how to incorporate Belief Networks into your products and projects to offer new ways for your customers to gain the benefits of expert and decision support systems in areas that are too complex to benefit from business rules.

Can you afford all the black-and-white rules you need… to cover the gray areas?

Rules-based expert systems use key insights to turn data into information and decisions. However, these systems have a critical characteristic that restricts their use to only the simplest of problems: they only allow yes-or-no, if-this-then-that rules. When there are more than just a few contributing factors for a decision, or if some factors involve gradations or uncertain information, the number of rules needed to fully define the decision making process can grow exponentially. Additional rules are needed to account for situations where data is missing, or where inputs come in the form of possibilities or positions in a spectrum, not facts.

Do I really want to carry an umbrella every time I miss the weather report?

Consider a simple system that would help you decide whether or not to carry an umbrella when you leave a building. You have several inputs you probably use in making this decision:

- The most recent weather report you heard or saw
- What the sky looks like (or looked like when last you saw it)
- If you have achy joints, you may feel a twinge when the weather is about to change

There are significant complexities and chances for uncertainty within even this small set of inputs. The weather reporters, always hedging their bets, usually put a percentage on the chance of rain. You might miss the report if you're in a hurry. Perhaps the weather isn't changing fast enough for you to feel it in your achy joints. It may be cloudy, but not so much as last time it rained…

To understand how these additional complexities can cause the size of the system to grow unmanageably large, consider constructing a useful rule system for this situation. A number of simplifying assumptions will have to be made to incorporate information that comes in the form of a spectrum (like the % chance of rain, or how cloudy it really is). In addition, you will have to manage a growing number of rules as you incorporate more of the important inputs and potential uncertainties.

The weather report is easy to tackle:

`If the chance of rain > 80%.....Then always bring an umbrella`

But, how do you know what to do if you miss the weather report?

`If it is cloudy.....Then always bring an umbrella?`

Well, how do you define cloudy? It would seem extreme to bring an umbrella every time it was partly cloudy in the morning. Is there a way to match this up with whether or not your joints hurt and what you might hear in the weather report?

OK, let's try to make some rules.

- Aching joints have two states: either they hurt or they don't
- The weather report usually comes in some % chance of rain. Let's break it into 4 ranges to make things easier: 0-40%, 41-70%, 71-90%, and 90-100%
- Again, keeping things simple, let's define 3 states for the sky: sunny, partly cloudy and cloudy

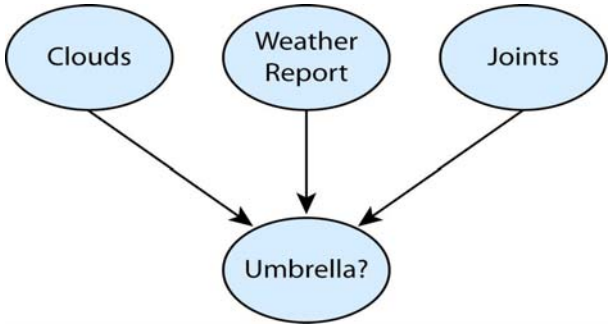
Now combine these rules. That means…2 x 4 x 3 or 24 rules. You will need even more if you want to include cases like missing the weather report, or not being able see the sky because you're indoors. Suddenly, the list of rules is getting long. Even if you decide to carry an umbrella no matter what if the weather report says chance of rain 90-100%, that only cuts out a few.

One Belief Network is Worth Thousands of Rules (or More)

In the case of the Umbrella Decision System, using a belief network instead of rules offers a chance to condense the large number of required rules into a simple model. In fact, a belief network even allows the user to incorporate prior knowledge and uncertain information without adding rules or complexity.

The First Step: Create a Network and Generate a Table of Likelihoods

A simple diagram of potentially useful inputs and the results they drive is the first step. You can now build a table that lays out which circumstances mean you should bring an umbrella.



Conditional Probability Table

	Parents			Child	
	Joints Ache?	Is It Clou...	Weather Report	Need Umbrella?	
				YES	NO
1	Yes	Completely	Will Rain	1.000	0.000
2	Yes	Completely	Will Not Rain	0.000	1.000
3	Yes	Partly	Will Rain	1.000	0.000
4	Yes	Partly	Will Not Rain	0.500	0.500
5	Yes	No	Will Rain	1.000	0.000
6	Yes	No	Will Not Rain	0.500	0.500
7	No	Completely	Will Rain	0.800	0.200
8	No	Completely	Will Not Rain	0.500	0.500
9	No	Partly	Will Rain	0.800	0.200
10	No	Partly	Will Not Rain	0.200	0.800
11	No	No	Will Rain	0.800	0.200
12	No	No	Will Not Rain	0.100	0.900

Notice that the weather report can be listed as either "will rain" or "will not rain" - we don't need to split up the whole range of possible % chances of rain because a belief network will allow us to enter a % chance directly. That is, ***you don't have to be sure: the information that it* might *rain is just as useful to us as the information that it* will *rain!***

Squeezing all you can out of what you already know:

If you can build a table like the one above for deciding to bring an umbrella, can't you build similar ones for each of the other inputs? You do know about how often it is cloudy, about how often the weather report calls for rain, and about how often your joints hurt. Can't you somehow use this information to fill in the blanks when you happen to miss the weather report, or you are leaving for an evening meeting and can't tell if it is cloudy?

Yes. With a belief network, you can fill in for what you don't know today with what you know usually happened in the past.

Historical chances that it will be cloudy	**Cloudy**	**Partly Cloudy**	**Sunny**
	30%	40%	30%
Usual Chance of Rain	10%		
How likely your joints are to hurt on any given day	15%		

Granted, this information about previous experience is not as good as knowing exactly what is going on right now, but *it is a lot better than using no information at all* to make a decision.

Other advantages of Belief Networks

Beyond the reduction in the expense of building a decision or expert system, Belief Networks offer a number of other advantages:

- It is easy to explain how a system arrived at a particular recommendation, decision, or action
- It is possible to diagnose problems: Belief Networks can be run in multiple directions
- With sensitivity analysis, it is easy to understand how changing inputs can affect overall results

Where can I apply Belief Networks?

Belief networks have already been applied in a number of business areas. In each of the examples below, the system described would have been impossible with business rules.

- Selecting loan opportunities with low default risk
- Estimating when equipment will need service, parts or retirement based on sensor input and records
- Quickly resolving customer problems via online troubleshooting tools
- Diagnosing and troubleshooting on-site cellular networks in real-time
- Filtering email to accurately control spam and to highlight critical messages
- Controlling autonomous vehicles and their navigation (wheeled, aquatic, aerospace, and others)
- Assessing risks and rewards of R&D projects to move resources into the best opportunities
- Many more…

Charles River Analytics' BNet® Products: The Benefits of Belief Network Expertise and Experience

Charles River Analytics, a leader in complex custom belief networks, has developed the BNet tool set for building belief networks and incorporating them into larger software applications.

The BNet product family includes:

- **BNet.Builder™** for rapidly building Belief Networks, entering information, and *getting results*
- **BNet.EngineKit™** for incorporating and running Belief Networks in your applications
- **BNet.Interface Kit** for graphically building and modifying Belief Networks in your applications

These tools offer a number of **advantages** for both users and developers:

- A user-friendly GUI
- Intuitive and fast information entry in information tables (conditional probability tables)
- Written entirely in Java for ease of integration
- Easy customization to mesh seamlessly with the look and feel of your own applications

If you or your colleagues wish to pre-purchase BNet products or participate in Beta tests of BNet:Builder, contact:

info@cra.com
Charles River Analytics
625 Mount Auburn St.
Cambridge, MA 02138
Tel: (617) 491-3474
Fax: (617) 868-0780

Charles River Analytics

- 20+ years developing belief network tools for defense, aerospace, security, and other clients
- Stable and Established:
    - 20+ years in business
    - $6+ million in revenue
- Expert in software and science:
    - 15 PhDs in Computer Science, Cognitive Science, Engineering, and other related fields
    - 37 MSs in Computer Science, Cognitive Science, Engineering, and other related fields