Bayes nets: understanding the best newest thing

| By Steven Struhl

snapshot

In this first of a two part series, Steven Struhl explains what Bayes nets are and why researchers should care about them

Just when we were starting to feel comfortable with our state of confusion about statistics, another remarkably powerful set of analytical methods with new rules has appeared. They are variously called Bayes nets, Bayesian networks, Bayesian belief networks and probabilistic structural equations—and generally, computational tools to model uncertainty.

Importantly for us, they work in many practical applications and often work better than other methods.

Bayes nets can quickly focus on important variables, show how variables are connected, what the importances of variables are with terrific accuracy, and build models with strong predictive power. They even bring us closer to what may approximate the Holy Grail in research: showing clear linkages between survey data and market share.

Bayes nets have many remarkable capabilities and, inevitably and unfortunately, some new terminology. We will need two installments to cover their basic functions and to give some examples of how they work. In this month's article we cover some background, including what Bayes nets do and what makes a Bayes net Bayesian. We will touch on some of the ground rules and then discuss the central concept of conditional probability.

To show how conditional probability can solve problems that really elude us intuitively, we will see how reliable a witness actually is when he says he saw an accident. Then we will solve the famous (or infamous) Monty Hall Let's Make a Deal problem in which you will get to decide whether to stick with the door you chose or switch to a different one.

The answers will surprise you!

We will round out this installment with some more basics, discussing networks and the value of information, and finally will compare networks with regressions.

In the next installment we will get to the examples, showing the remarkable powers of these networks in practical applications. We will first show how a network automatically found logical and informative patterns of relationships in questions from a typical big, messy questionnaire. We will conclude with a network linking questionnaire questions to market share with over 70 percent correct prediction levels. This example strongly suggests that Bayes nets may well be the next new thing, greatly expanding our ability to understand variables and their effects.

A proven method

Networks are new to research and the social sciences but they have proven themselves in many other fields. They have served for years as reliable and valuable additions to the analytical armamentarium. So the bugs have been worked out and there are a host of highly useful applications. Work that was directly applicable to the development of Bayes nets goes back at least to the 1940s. Judea Pearl's *Probabilistic Reasoning in intelligent Systems: Networks of Plausible Inference*, which discusses principles that underlie these nets, dates back to 1988.

A network can be simple, like the example in Figure 1, which shows the relationship of cancer, bronchitis and abnormal X-rays. (Set as it is, it shows what we can expect in the other areas if a person has bronchitis.) Networks can

Quirks.com/articles ID 20130306

become so complex that reading them becomes quite difficult. Below in Figure 2 is one that is nearly unintelligible, but that still works. It has a genuinely serious application, namely deciding whether to launch a missile. This at least should give us confidence that these Nets are in fact (and not just in metaphor) "battle tested."

Bayes Nets are a powerful and flexible set of approaches that can solve many problems. The variety of uses can range from brainstorming to highly sophisticated modeling and forecasting systems.

Figure 1

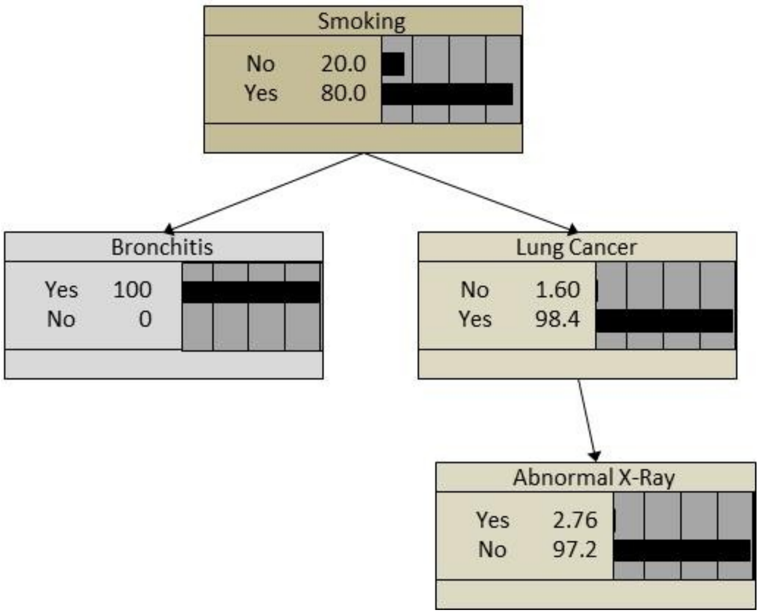
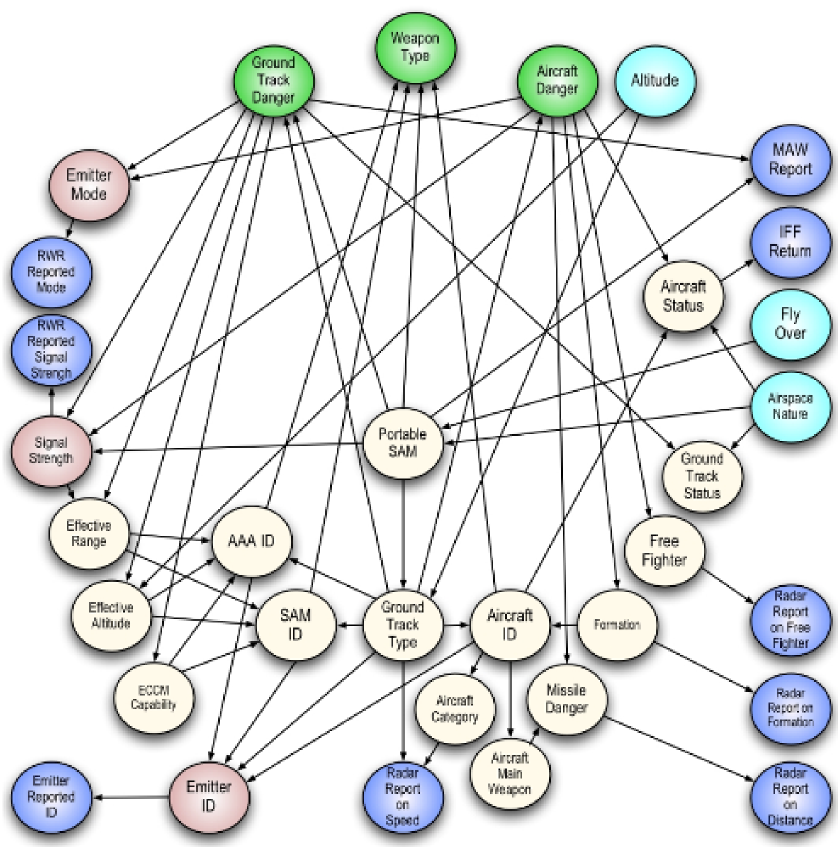


Figure 2



Here are some applications:

- Automatically finding meaningful patterns and connections among variables;
- Getting accurate measures of variable strengths (for drivers analysis);
- Screening large numbers of variables quickly (for data mining);
- Linking questionnaire data to data from the outside world, such as market share;
- Developing models of cause and effect (in the right circumstances);
- Incorporating expert judgment into data-driven models.

Everything Bayesian refers back to the work of the reverend Thomas Bayes, who lived an apparently quiet life in Tunbridge Wells, England in the 18^{th} century. He published two books in the 1730s, but never the "Bayes' theorem" that bears his name.

Bayes' formulation itself is simple. We should add that he never called it a theorem, and that any reasonably literate person could easily understand it in its entirety, aside from what one writer[1] astutely calls the "goggle-making" formulation often used to represent it.

Starting from Bayes' straightforward assertion and arriving at many of the types of analyses that bear his name likely would have caused the good reverend to take on a strange hue. The portrait at the left shows just what happened.

We can formulate Bayes' idea in a variety of ways. Let's start with this more practical formulation:

> We start with "prior" (existing) beliefs, and we can update or modify these by using information on likelihoods which we get from data we observe. Adding this information gives us a new and more accurate "posterior" estimate. From this posterior estimate, we draw conclusions.

That's really all there is to it.

However, it is usual to encounter this formulation, which can indeed make many readers' eyes goggle:

$$P(Bi|A) = P(A|Bi)P(Bi)/\{P(Bi)P(A|Bi)\}$$

Yet this notation simply reflects what we said in the modest paragraph above.

The Bayesian approach also includes the idea of "conditional probability." This phrase appears prominently in many discussions. However, a probability that is "conditional" on the data is no more than what we just described: an estimate of probability that is revised based on including information from data into some prior estimate or belief.

The ground rules for networks

Diagrams of variables are key

A Bayes Net calculates relationships among variables and shows how they fit together. A diagram of how variables relate to each other therefore is integral. Somewhat more formally, these networks are based on "graph theory" and on "probability theory"—so grasping their workings requires both a diagram and calculations with it.

Looking at a network, you will see a familiar type of diagram, if you have experience with structural equation models (SEMs) or with partial least squares (PLS) regression path models. Variables are connected with arrows, or *arcs*, showing pathways between them, and these lead to target variables.

Directions are important

In a Bayes Net, there must be directions between the variables and there cannot be any circular or "cyclic" pathways where a variable points back to itself. This is why these networks are sometimes called "directed acyclic diagrams"—or as you may encounter in the literature, DAGs.

The arrows or arcs have a specific meaning in these diagrams. However, this is largely intuitive. A variable at the start of an arrow leads to another variable, and in certain conditions we even can say that the starting variable *causes* the variable at the end. Arrows can lead to or from a target variable.

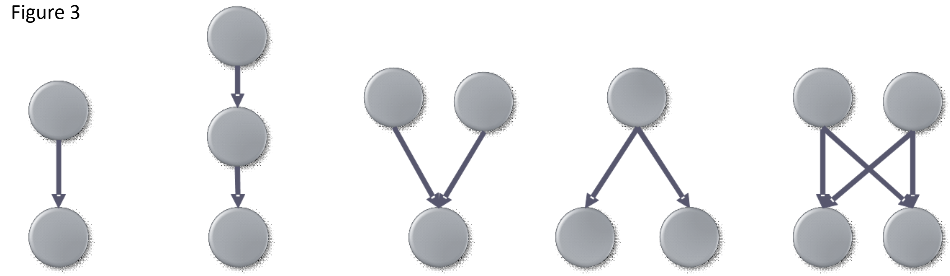
Terms and phrases: It's all in the family

There is of course some terminology to learn. (See Figure 3) Fortunately, much of this too is largely intuitive, and rather uncharacteristically warm and fuzzy for statistics.

- The variable at the start of an arrow is called a "parent."
- The variable at the end is called a "child" of the parent.
- Children can have several parents and parents can have several children.
- If there are two or more parents, they are called "spouses."
- A parent of a parent is a grandparent, and so on.
- Variables are "dependent" only if they are directly connected. Children and parents are dependent on each other. Children are independent of grandparents and other variables further away.

Whether variables are dependent on each other becomes important when screening variables for inclusion in a model. One powerful screening technique is to include only those variables that are dependent on the target variable (its parents and children) and the other parents of any children.

Figure 3



Closely connected variables have stronger effects, so this quickly eliminates less important variables where there are many—as in data mining applications. This set of variables has a name also: the "Markov blanket." (How Mr. Markov enters consideration is something to talk about another time.)

Related to whether variables are dependent, there is another item of terminology that you may encounter. The variables have "edges" that go with the arrows. The "edge" of a child node points toward a parent node.

Everything is connected: changes move through the whole network. Regardless of dependencies, all variables in a network change when one changes. This sometimes is described as "information propagating through the network." The whole network is connected. And indeed, as we will soon see, understanding networks as conveying information is critical to their practical applications.

This connectedness throughout the network also makes estimates of variables' importances much more powerful and accurate than those we can get from regression-based models.

Using regressions, we need to assume that when we change one variable, all others remain constant. This can happen if we set up an experiment, but with real-world data, this is hardly ever the case.

However, with a Bayes Net, any effect we see takes into account all the connections to all the other variables. We are considering the entire system when we measure any effects.

Network building ranges from simple to complex. As follows, when we are modeling relationships among variables, our choices in how the network gets put together are of prime importance. There are many, many ways to get a network assembled automatically. At their simplest, there are methods in which all variables get fit directly to the target variable as well as possible. This is very much like a regression where all variables are put in without screening to see which ones belong.

Networks at their most complex result from countless attempts to fit the data—finding how variables best fit together to predict or explain the target variable. These methods use sophisticated tests to ensure that the network does not seize upon a connection that is good "locally" (where a variable is being added) but not good for the overall network.

Did somebody say we can figure out causality? Finally, we did mention that we can, at times, see whether one variable actually *causes* another. The idea of finding causality in networks is quite intriguing. However, when we build a network from the data we typically use, we often discover that some arrows work as well in either direction. These directions are "equivalencies" and we must decide how the arrows point.

Only if an arrow *must* point in one direction can we say that one variable causes another. There are tests that determine this.

Unfortunately, rarely does the data we find in surveys have completely definite directions among the variables.

Conditional probability

Bayes Nets involve conditional probabilities, which in practical terms means how the distribution of values in each variable fits with the distribution of values in other variables.

More practically, networks can solve problems quickly that are difficult or elusive to solve using other methods. The workings of conditional probability can be difficult to envisions, so hold on and we will try two small examples:

- The yellow taxi-while taxi problem
- The "three door" Monty Hall problem[2]

The yellow taxi—white taxi problem

There is an accident involving a taxicab. A witness reports that the cab involved was white in color. In this city, 85% of cabs are yellow and 15% are white. The police actually test the witness out on a street corner and find that he is 80% accurate at getting the cab's color.

What are the odds that the cab actually was white? We can solve this with a simple Bayes Net.

Setting up the taxicab problem is straightforward. Recall that we can make a network ourselves by linking up variables, just as we can create a network from a data file. Here we will form the network by linking two events: the color of the cab and what the witness reports as the color. Each event is called a "node." (Figure 4)

We understand that the actual color of the cab leads to what the witness reports as its color, so we will draw a small network with the color of the cab leading to what the witness reports.

First we set up the node showing the odds of a taxicab being yellow. (Figure 5) Next we set up the second node showing the odds of the witness being right about each type of cab. (Figure 6)

Now what happens when the witness says he definitely saw a white cab?

In the diagram, we will change the value of "What the witness says" to 100% for white. Since cab color and what the witness says are linked (as we showed in the diagram), if we change the value of one node, the other will change along with it.

So even though the arrow points from cab to witness, if we change the value for "witness", we will see the change *flow back* to the likelihood of the cab being a given color.

Figure 4

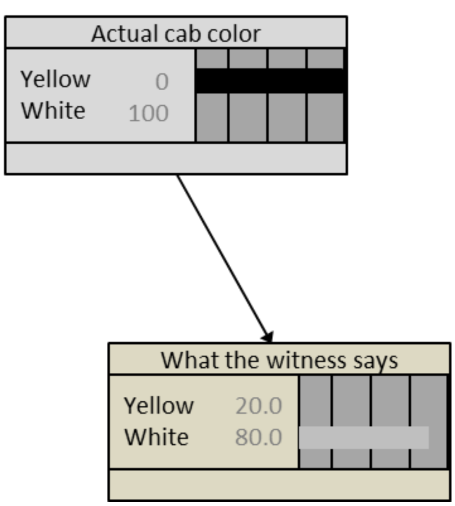


Figure 5

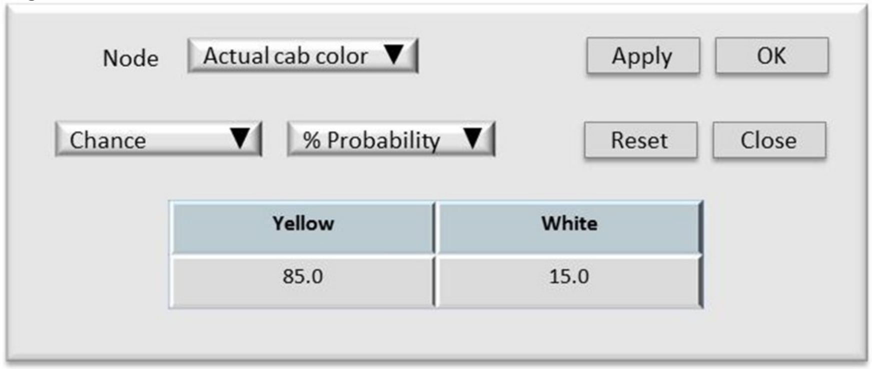
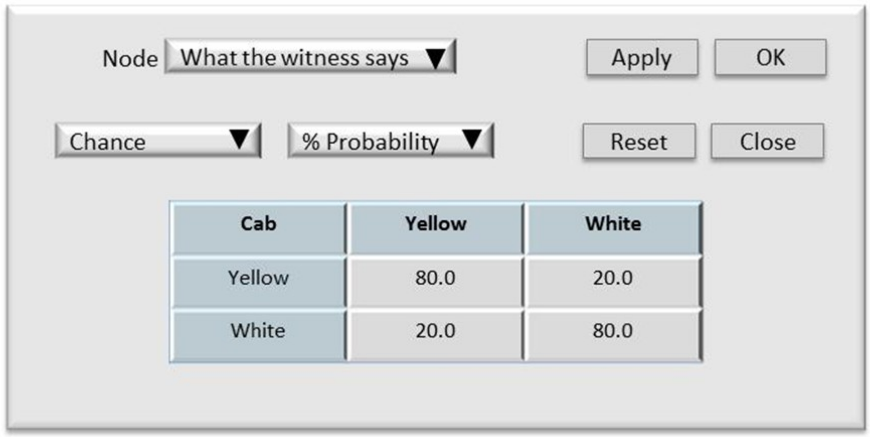


Figure 6

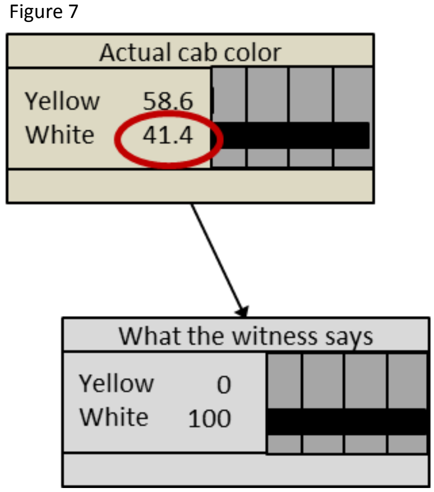


As we can see, the odds of the cab actually being white is about 41.4%, given that 85% of cabs are yellow and the witness is 80% right in identifying colors.

We have just come upon something difficult about Bayes Nets. We have just solved a problem that would have eluded most of us, neatly and simply. And yet, the correct answer seems strange. This is a difficulty we have with conditional probability. As Yudkowsky[3] aptly puts it: "Bayesian reasoning is very counterintuitive."

In sum, we have an approach that is powerful and hard to work out in our heads.

Figure 7



The three door "Monty Hall" problem

This is a classic that has generated numerous arguments among scientists, statisticians, random onlookers, and fans/foes of *Parade* magazine puzzle columnist Marilyn vos Savant. Here's the problem:

There is a prize behind one of three doors. You pick a door. The sneaky game host does not tell you whether your door has won. Rather, he opens another door where there is NO prize. Then he asks whether you would rather switch doors OR stay with your door.

What do you do?

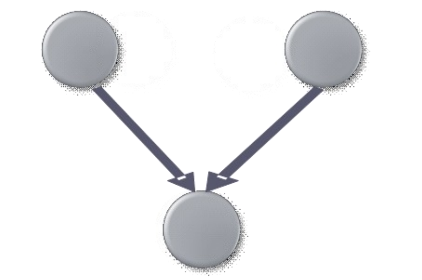
A statistician has shown on the Web how to get this nearly correct by setting up 10,000 simulation runs in SPSS.[4] We will solve it with a simple three-node network

Setting up is critical. First we have two independent events (Figure 8): the door you pick, and the door that wins. However, the door that is opened depends on both your choice and the winner. So these two independent events are now linked by the event of the door being opened (it depends on both of them).

Each of the independent events has a probability of 33% for each door. This part is very straightforward. (Figure 9).

Now on to the key: Which door gets opened, based on which one you have chosen. This is going to take some thinking, and echoing Mr. Yudkowsky above, this part is not completely intuitive.

Figure 8



We have made a table (Figure 10) showing what happens with your choice and which door wins. Here goes. The table starts by saying you could choose Door 1, Door 2, or Door 3. For each of your choices, the winner could be any of the doors. So we have Doors 1, 2 and 3 within each of the three doors as the headers for the rows, going down the right side.

Now we have to cross these rows with which door gets opened. So at the top of the table, the column headers are "Open 1," "Open 2," and "Open 3." We now start to see some surprising relationships.

Right away we can see that the odds of any door getting opened do not match. Just looking at this table, it seems lopsided, not what we would expect if the chances are even. A table with even chances would look completely symmetrical.

Figure 9

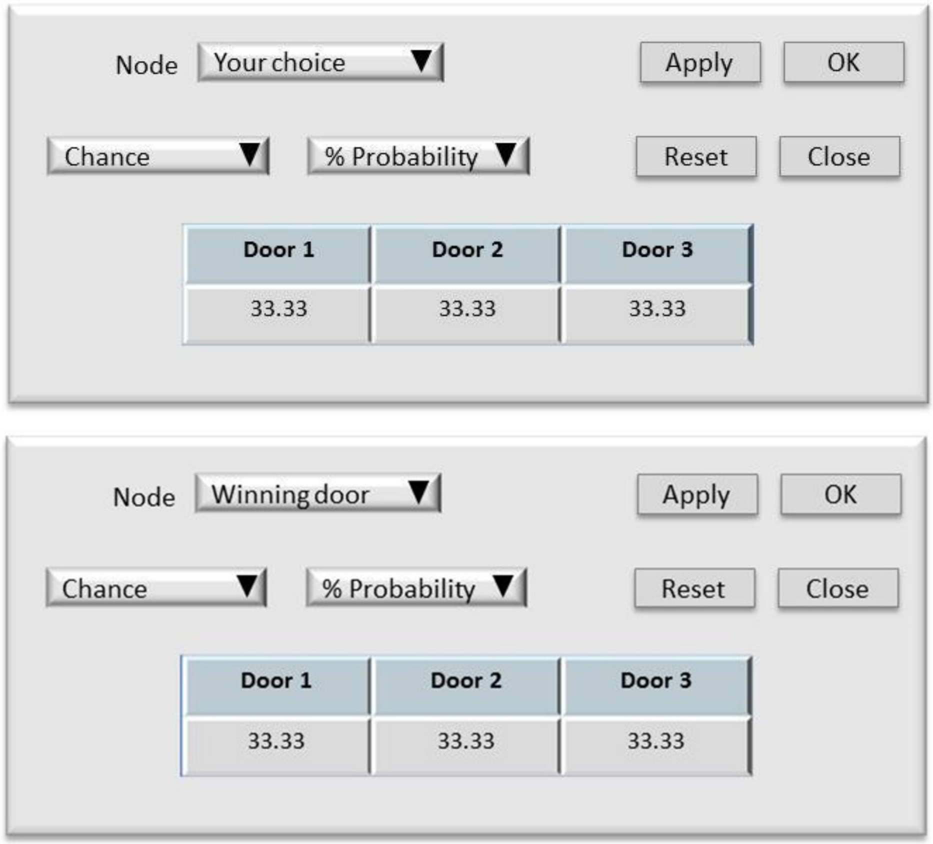
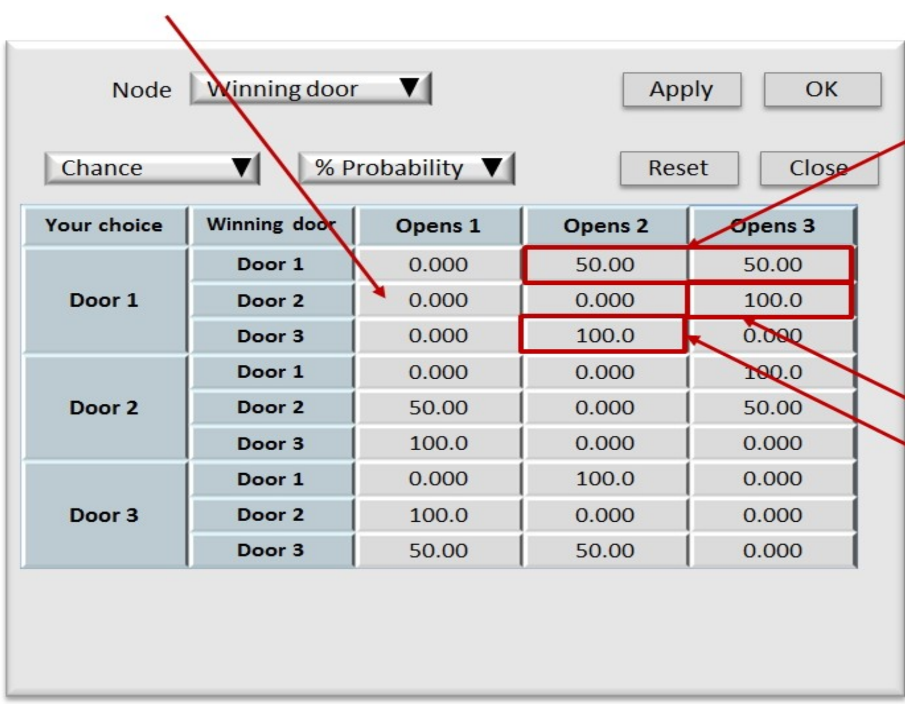


Figure 10

Door 1 NEVER gets open if you pick it—it doesn't matter which door is the winner



Your choice	Winning door	Opens 1	Opens 2	Opens 3
Door 1	Door 1	0.000	50.00	50.00
	Door 2	0.000	0.000	100.0
	Door 3	0.000	100.0	0.000
Door 2	Door 1	0.000	0.000	100.0
	Door 2	50.00	0.000	50.00
	Door 3	100.0	0.000	0.000
Door 3	Door 1	0.000	100.0	0.000
	Door 2	100.0	0.000	0.000
	Door 3	50.00	50.00	0.000

If you picked door 1 and it was the winner, the host has an even chance of picking door 2 or door 3

But if you picked door 1 and door 2 was the winner, then the host MUST pick door 3. The same happens if you picked door 1 and the winner was door 2.

Once you realize how door 1 is set up, it Is relatively simple to do the same for doors 2 and 3.

Figure 11

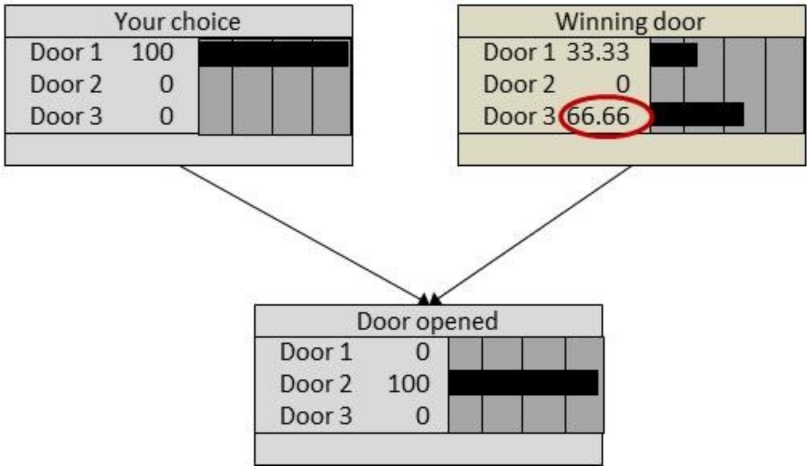
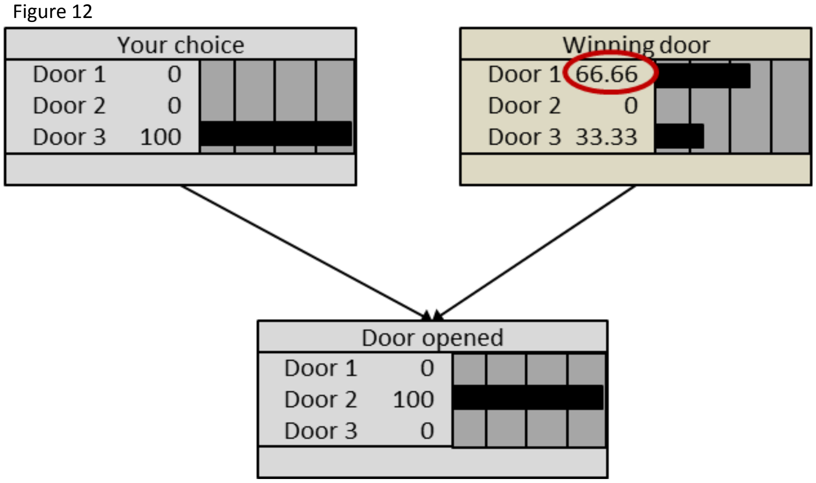


Figure 12



Networks gain more power if they use "value of information" as a basis for understanding structure. Information has value inversely proportional to its probability. That is, describing high probability events has low information. Conversely, high levels of information consist of describing low probability events accurately.

This is not statistics as we have known it. Rather, testing balances the value of information vs. the length of description in machine language.

As you read about networks, you will encounter the Minimum description length (MDL) principle. It is based on the idea that any regularity in a data set can be used to "compress the data"—to describe it "using fewer symbols than needed to describe the data literally."[5]

Therefore, the best explanation is the one that that minimizes description length while conveying the most information. As a general practice, information theory has sought to keep cost of describing the data equal to or less than the value of the information in the data.

This is an excellent idea with very large data sets. There is plenty of data and it makes sense to balance how much information we gain precisely against how much effort it takes to describe that information. Using survey data and typical sample sizes, though, we may need to explore different ratios of description vs. information to see patterns clearly. the variance or pattern in the dependent variable.

Anyhow, the (perhaps not) surprising answer to the three door problem: You should switch!

The diagram in Figure 11 (following) may take a little explaining, but it shows the result. In it, we have moved your choice to "Door 1." Now door 1 cannot be the one opened, so the sneaky host opens Door 2. The bar in "Door the host opens" gets moved to reflect that. "Your first choice" and "Door the host opened" have taken on a grayish color to show that they are being changed.

Not only should you switch, but the odds favor you switching by 2:1 (66.6% for Door 1 vs. 33.3% for staying with your door). The red arrow points to the door with the best odds of having the prize—namely, the other door.

Now just to confirm that this is not a fluke, we change the doors. Now you choose Door 3 and the host opens Door 2. Once again, the odds favor a switch: 66.6% to 33.3% for staying.

If this seems confusing, if not impossible, you are in good company. About 10,000 people wrote to Marilyn vos Savant when she published the correct answer, saying she was wrong. About 1000 of those people had doctorates.

As we said above, the conditional logic used in networks is extremely powerful, but not always intuitive, even if it gets the correct answer quickly.

To review, what tends to trick us is that the actions of the host have caused *a link* between two events that are otherwise independent. Now we see two points: (1) these events are connected, and (2) the choice the host makes connects them. We then can understand how change will *flow through* when one of the events changes.

From a the perspective of looking at a network, changes in probability must *propagate (flow) through* the network regardless of the way the arrows point.

New statistics involved

Networks can use standard statistical tests to determine structure, but these tests may pose problems. In fact, many problems related to networks are called "hard" in statistical language. Sometimes you will see the term "NP-Hard" (which also may describe the reading that follows). Practically speaking, NP-Hard problems can be insoluble—and that definitely would slow down your work.

In a network, if we rely on tests of significance, it is often difficult to choose the appropriate tests and good thresholds for those tests, because relationships can be numerous and highly complex. We also might be forced to reduce the number of statistical tests or reduce the number of variables processed, in an attempt to increase the tests' reliability.

May seem puzzling

Regressions are equivalent to a "Naïve Bayesian" network. In a Naïve Bayesian network all the variance in the target variable is portioned out to the dependent variables. And indeed, this is just what happens in a regression.

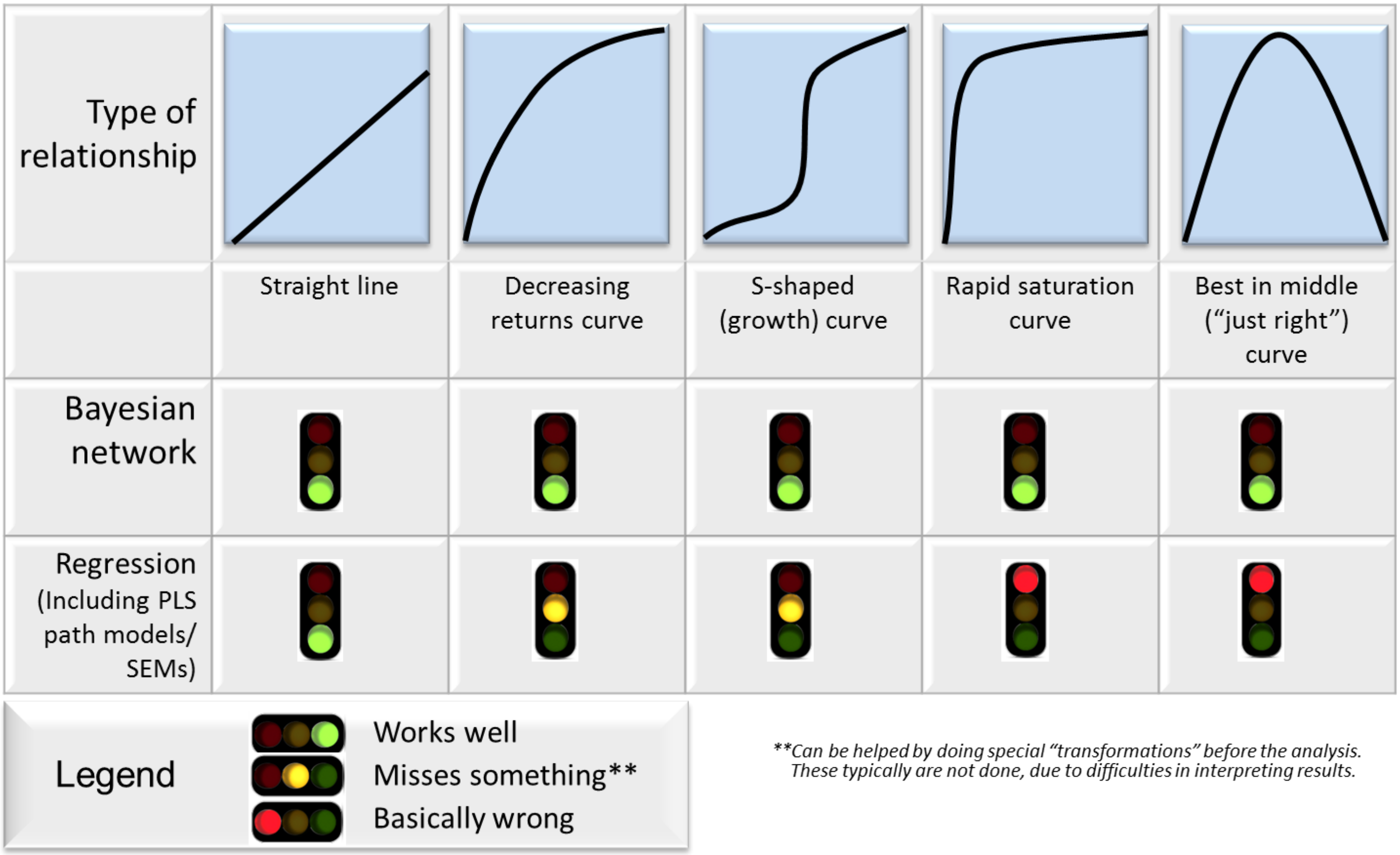
This may seem puzzling (is this becoming a refrain?), but one way to describe the dependent variables in a regression is as *explaining* the dependent. That is, each independent variable accounts for some of the variance in the target or dependent variable.

We have become accustomed to talking about the independent variables as "predicting" or "driving" the independent. However, if we think of the coefficients in a regression, we can see that each independent variable actually makes up some part of the total value of the dependent variable. We sum up the contribution of each variable times its coefficient, and that total is the predicted value of the dependent.

Bayes Nets of course differ from regressions in important ways. Regressions are supposed to use continuous dependent variables (or with logistic regression, binary ones). Bayes Nets were developed for use with discrete (categorical) variables. Many programs that create and analyze networks still only can handle variables of this type.

However, in recent years, networks have

Figure 13



***Can be helped by doing special “transformations” before the analysis. These typically are not done, due to difficulties in interpreting results.*

been extended and now can analyze continuous variables as well. The math involved is very abstruse indeed—but it works.

Regressions are supposed to have straight line relationships among variables. This means all relationships, not only between the dependent and the independents, but among all the independent variables as well. Bayes Nets can handle any regular relationship between variables, whether it is linear or not. Figure 13 (helpfully provided by a Mr. L. da Vinci), shows where each method is relatively likely to find success.

Will find logical patterns

We will return next month with two remarkable examples of the practical uses of Bayes Nets, showing how they find logical patterns even in messy data, and then giving an example of how they linked questionnaire questions to market share with remarkable predictive activity. In short, we will see two demonstrations that show just how these Nets may be the “best newest thing” in understanding data. Stay tuned.

Steven Struhl is principal at
Converge Analytic, Philadelphia, PA.
He can be reached at smstruhl@gmail.com
or smstruhl@convergeanalytic.com

REFERENCES

[1] Stanners, W. (1999). “Essay on Bayes,” *Game Theory and Information*

[2] Thanks to Lionel Jouffe for the examples

[3] http://yudkowsky.net/rational/bayes

[4] http://www.uvm.edu/~dhowell/StatPages/More_Stuff/ThreeDoor.html

[5] Perhaps the definitive work on this is Grünwald, P. (2007) The Minimum Description Length Principle, (Cambridge, MA: MIT Press). That’s all we can say about it in this paper.

Bayes nets: understanding the best newest thing

Part 2

| by Steven Struhl

This is the second installment discussing the remarkable realm of Bayes Nets. This month, we will show aspects of what makes them so notable, giving two practical examples dealing with the type of messy data we often find at the end of a research study.

In last month's section, we covered some background, including what Bayes Nets do and what makes a Bayes Net Bayesian. We touched on some of the ground rules, and then discussed the central concept of conditional probability.

We also showed how conditional probability can solve problems that elude us intuitively, including the famous (or infamous) Monty Hall "Let's Make a Deal" problem, in which you will get to decide whether to stick with the door you chose or switch.

Finding structure with a Bayes Net

As mentioned back in the first part, Bayes Nets find patterns and connections in data. They seek out the best connections among sets of variables, showing which variables are most closely related and how all the variables in a data set work together to predict the dependent variable.

In this example, we will show how one network found a compelling structure in a very messy data set. This was a typical sprawling questionnaire, where everybody involved had a pet question or two (or four), and then of course expected their questions to appear in the final model.

The more thoughtful people of course knew that all these questions were not needed—some items even could be spotted right away as near duplicates—but this did not alter the strong general expectation that everything would get fit into the model.

This type of data set often leads to considerable frustration for the person charged with making some sense of it all. This is especially the case if developing a true multivariate model becomes an objective only *after* those involved say collectively, "There is too much to make any sense of here!" following many manful struggles with stacks of devotedly prepared cross-tabulations.

There are only a few options for dealing with such unruly data sets and making a model with all or most of the variables. Both structural equation models (SEMs) and partial least squares (PLS) path models have been turned toward this use. However, both of these methods struggle with a large burden of variables to process.

PLS path models require a great deal of testing and retesting, grouping variables and moving variables from group to group. Finally, with a large number of variables, none of the predictors seems to have much effect.

Structural equation models share many of these complications, and also may fail to run at all. As these models grow large, they also need to include fleets of "unknown" quantities, typically not shown in any final discussion, but essential for the model to run. The large number of items and connections can be difficult to juggle.

However, Bayes Nets do nearly all the hard work themselves. We will see part of what one revealed. The entire network was monstrous, with 54 variables connected to the dependent measure ("I would use this again due to quality" on a likelihood scale).

The excerpt in Figure 1 is still rather staggering. It comes from a study of a pharmaceutical product. The diagram following in Figure 2 shows how the network found logical arrangements in these variables

So hold on—here is the list of variables and the structure we found.

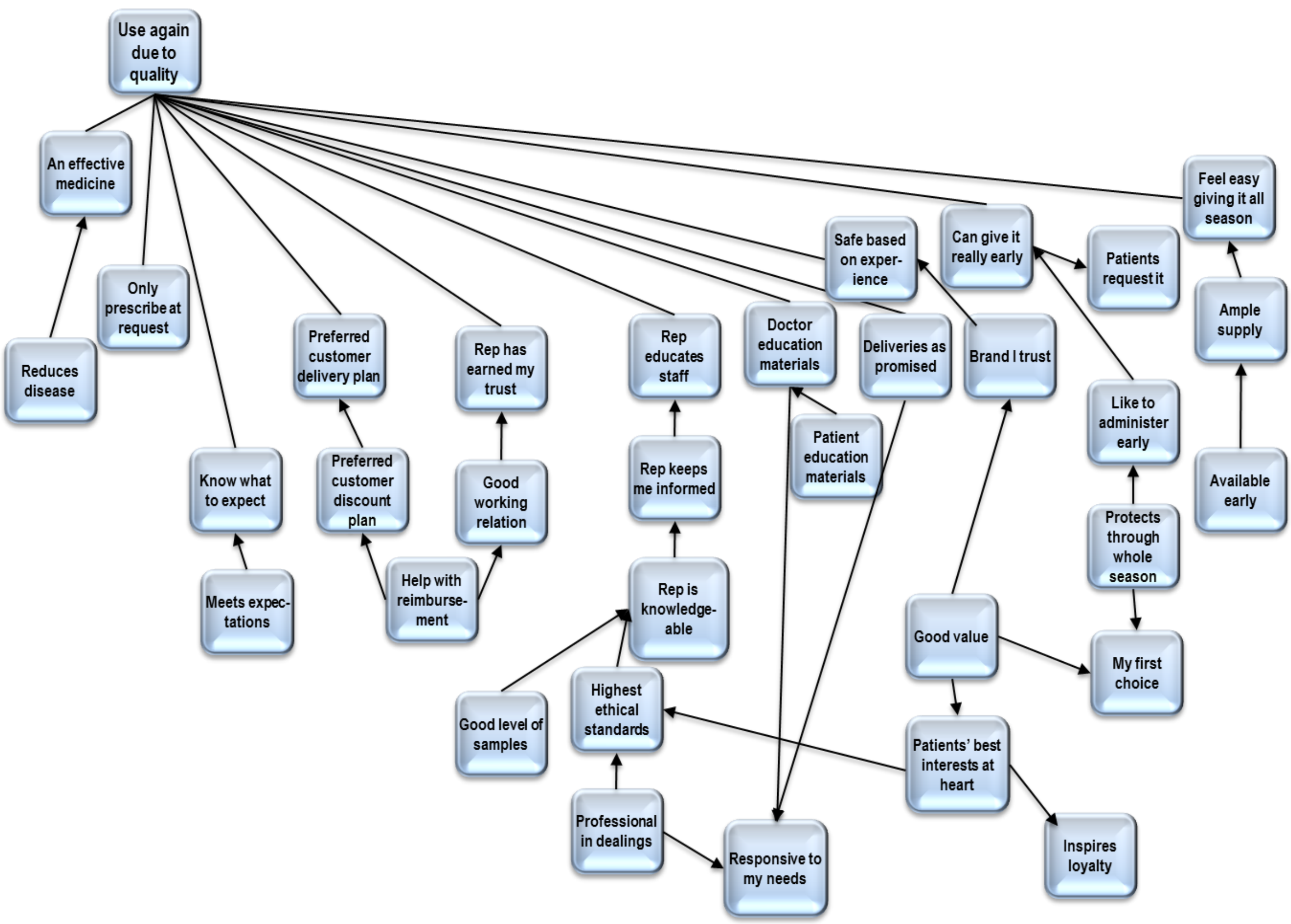
Figure 1

An effective medicine
Only prescribe at request
Product reduces disease
I know what to expect
Meets expectations
Preferred customer delivery plan
Preferred customer discount plan
Help with reimbursement
Rep has earned my trust
Good working relationship
Rep educates staff

Rep keeps me informed
Rep is knowledgeable
Good level of samples
Highest ethical standards
Professional in dealings
Responsive to my needs
Patient education materials
Medication safe based on experience
Provides good value
Safe based on experience
Deliveries as promised

Protects through whole season
Like to administer early
Available early
Ample supply
Feel easy giving it all season
Brand I trust
Patients' best interests at heart
Inspires loyalty
My first choice

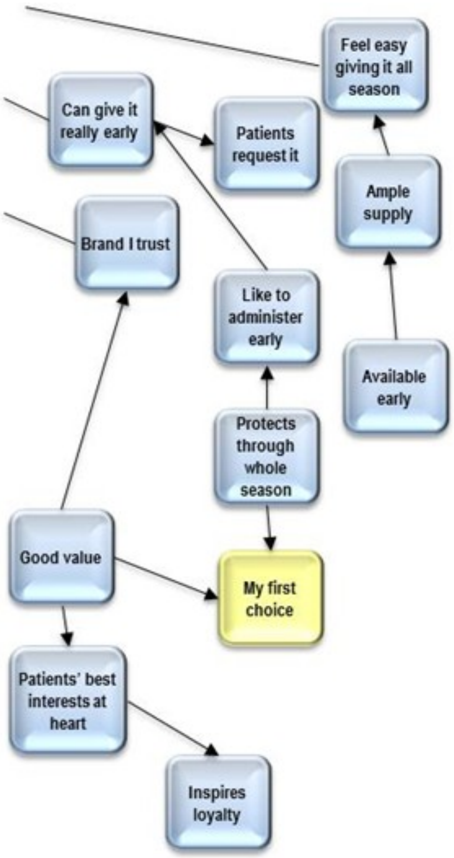
Figure 2



The complexity here can be daunting. Yet when we look carefully, one section at a time, there are many surprising yet intuitive relationships. For instance, starting in the lower right corner, note how "inspires loyalty" comes from the idea that the company "has patients' best interests at heart" and that "good value" in turn supports "patients' best interests." Having the patients' best interests at heart leads in turn to "has the highest ethical standards." Good value also leads to "my first choice" and "brand I trust." "Safe based on experience" is another variable with very close ties to "brand I trust."

Let's zoom in and pull aside just the variables closely related just to "my first choice." (Figure 3). Note that its closest connections are "good value" and "protects through the whole season." These two most likely would have the strongest relationship with "my first choice." These both seem to be more "hard edged" objective criteria. But then, the full picture involves still more connections. Good value is linked directly to "(has the) patients' best interests at heart." So a drug is not really a good value unless the doctor believes the

Figure 3



maker cares about patients.

Also back in the larger diagram, note all the variables that are *not* directly connected with "first choice," including the more relationship-oriented "responsive to my needs." That is its own focal point, in fact (at the bottom somewhat to the right of center), where (rather neatly), "professional in dealings" and "deliveries as promised" converge with "doctor education materials." For some pharmaceuticals, such as vaccines, it is key to doctors that they get the right materials that explain the applications and even storage of the product.

Throughout the network there are other very logical-seeming connections. Anywhere we look, the chains of variables, and how they connect, makes sense.

Note that not all the arrows go toward the dependent. But recall that variables will have effects on the dependent or target variable, regardless of which way the arrows point.

The fit is very good, with 63% correct prediction of the target variable—very strong considering we used a stringent "hold out" testing method called "cross-folded" validation. We can discuss "cross-folding" at another time, but it does mean that the correct prediction level has been reached with stiff testing requirements.

Importantly, we did not have to struggle, trying to fit together this model ourselves, as we would have needed to do with the SEM and PLS path modeling methods. We did have to give the model all of 20 seconds to run and do all its validation testing. Certainly there was advance time to make sure the data set was clean and ready to analyze.

But beyond that, we just needed to follow a few relatively straightforward steps:

1. Testing several alternative network building methods and seeing what they produced;
2. Finding if adjusting the threshold for "value of information" affected results;
3. Checking to see if further restrictions on the number of connections between parent and child nodes improved structure;
4. Testing a few connections to see if they could be reversed to clarify relationships;
5. Trimming a few connections that were extraneous. There are a few characteristic forms of connections within networks that, based on experience, can be safely eliminated.

At each step we checked overall network performance based two criteria. The first, correct classification levels, will be familiar to those having experience with discriminant analysis. That is, we determine how well each value of the dependent variable is predicted. The dependent variable in this example could take 10 values.

We also tested information scores, not accepting a solution where the information score was appreciably worse than the best we obtained.

However, paramount was whether the network made substantive sense. Did the connections not only predict the dependent variable, but did they convey a coherent story about the data? Here we have both a coherent story and very strong overall predictive power.

"Drivers" and linkage to share

Understanding patterns in variables is valuable, but then other questions follow, most importantly:

- What are the relative effects of the variables on the dependent variable, and
- How strongly does changing each variable affect the dependent?

Here we have an example where, thankfully, the predictor variables were heavily weeded. This is based on an information technology (or IT) product and a very long questionnaire. The network is represented more simply to fit on the page. (Figure 4)

Correct prediction levels were very strong indeed for fitting questionnaire questions to behavior. This model was 74% correct with the same stringent "validation" of results.

Those of us who have tried using questionnaire questions to predict actual behavior know that this almost invariably has poor results when using regression-based models. While networks do not always do as well as this one, they typically have outperformed regression-based models on overall measures of model fit with a behavioral target variable, such as actual use levels, buying the item in question, and so on.

Recall that the network also determines the importances of the variables and sensitivity of the dependent variable to changes in the independents. Adjacent on the page is perhaps the best variable-importance-measurement chart that your author has seen. (Figure 5). It compares the effects of changing each variable to directly changing the dependent variable. This chart shows that, for instance, changing "gives me a competitive advantage" has about 45% of the effect of directly changing the dependent.

Beyond this, we have actual sensitivity of the dependent variable to changes in the independents. In Figure 6, we can see the range of expected effects associated with these importance scores. Here again it is really important to restate that *this model does not show causation*. So we cannot say, for instance, that changing the score on "gives a competitive advantage" to 10 will increase signing up again by 12% (going from 76% to 88%).

We *can* say that if signing up ever reached 88%, then we would expect an average of 10 out of 10 points on this measure. Neither event seems that likely!

We also can see that there is considerable "downside" risk to letting these scores slip. That is, very low scores on this measure are consistent with a very low rate of signing up again.

Most areas have considerable "downside" risk—very low shares correspond with their lowest scores. This would be expected where baseline score in each area are high and so could fall

Figure 4

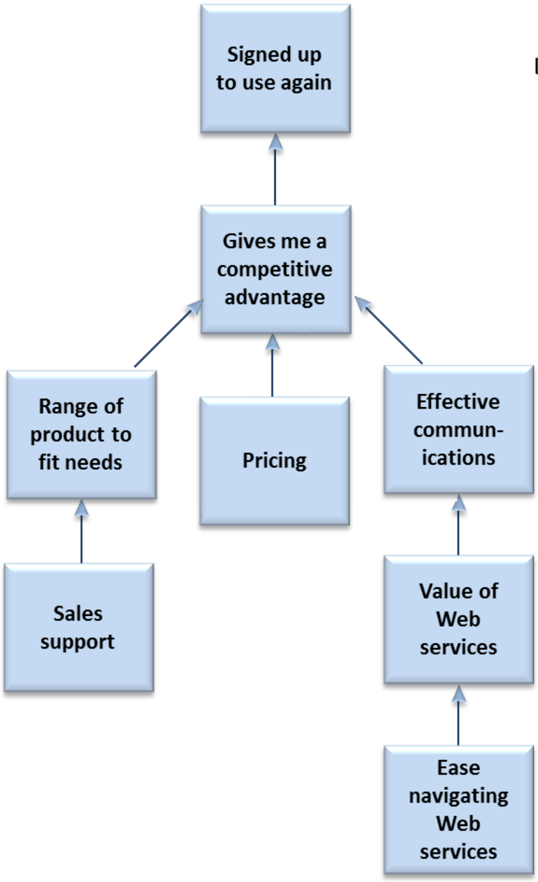


Figure 5

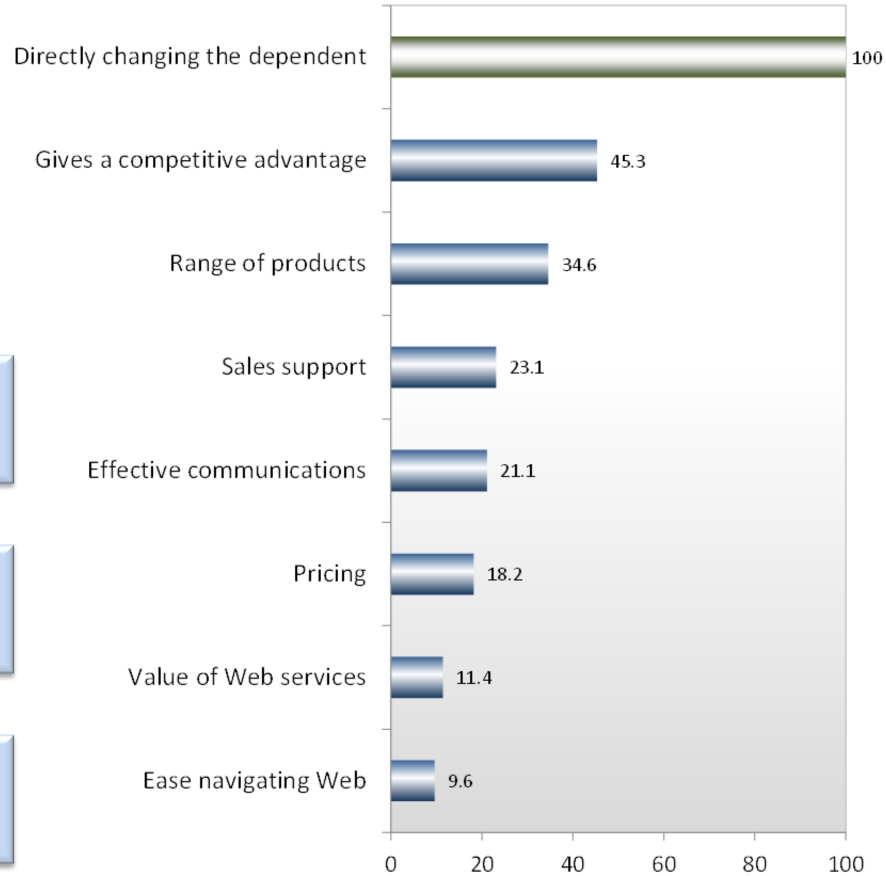
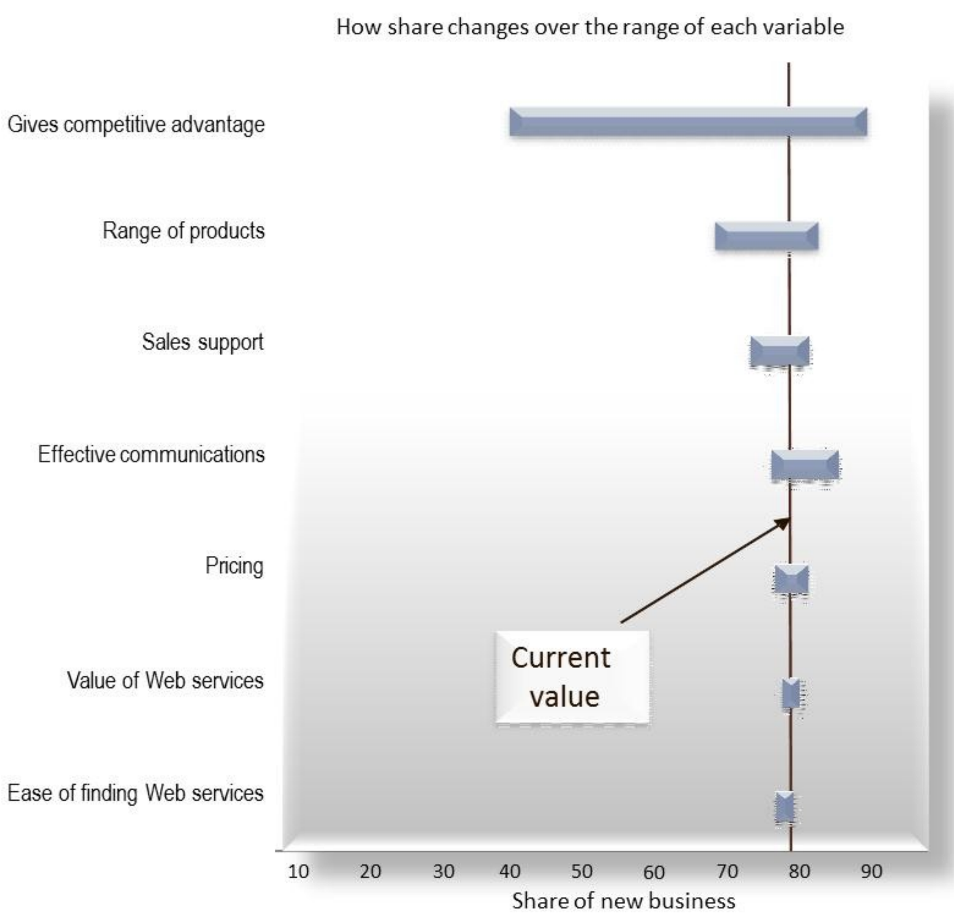


Figure 6



very sharply. We do notice that "gives a competitive" advantage and "effective communications" are the only areas that have much of an "upside"—that is, where gains would be associated with considerably higher shares. All the other areas are close to "saturation"—the point at which higher scores give minuscule incremental benefit, or actually may have reached it, as in the Web-related scores at the bottom.

Reaching back to concepts discussed in the first installment, what makes these estimates so powerful is that all parts of the network are linked and so effects in changing one variable take into account all the other variables in the network. We do not need to assume, as we do with regression-based models, that when we change one variable all other variables remain constant. That kind of assumption is very workable when we have a controlled experiment, as we do with conjoint analysis or discrete choice modeling, but it is not realistic when we are dealing with data gathered by questions and observations—as in a questionnaire.

Bayes Nets move beyond a salient limitation in analytical approaches to date when finding the effects of variables. Assuming we reach a network configuration that makes sense, we see the most realistic estimates of how variables influence each other.

What we cannot see here, unfortunately, is that the network diagram itself reflects how all variables change when we change any one of them. When we manipulate any variable, not only the target variable, but all the other variables in the network change. The network diagram shows this in real time.

The vast world of Bayes Nets software

Software can pose problems. Programs present a confusing world of choices. Applications range from free to incredibly expensive, and from remarkably capable to nearly useless. Not all the free programs are of the "nearly useless" type, and your author is not yet sure that all the really expensive ones are "remarkably capable," although we would believe so, since several have been selling at very lofty prices for years. (Unfortunately not all programs offer even limited trial versions. Fortunately, though, many do, and so you can try those before you buy.)

Most programs can solve logic problems. That is, you can build networks in them by hand, as we did earlier, and use those networks to answer specific questions. However, there are many programs with salient limitations. For instance:

- Some only some allow editing of networks made elsewhere;
- Only some can find structures in data;
- Only some allow for multiple methods of searching for structure;
- Only some give strengths of relationships.
- Some do not even accept data files. They are just for solving logic problems.

There are lists of Bayes Net programs you can refer to and explore. KBNuggets has one such list (http://www.kdnuggets.com/software/bayesian.html), and it seems to have a good number of programs. Kevin Murphy at University of British Columbia also has a list; recently updated, it has some helpful notes (http://www.cs.ubc.ca/~murphyk/Software/bnsoft.html). Google once had such a list but apparently has abandoned it.

From your author's perspective, to make full use of Bayes Nets, you need a program that allows you to do the following:

- Screen variables,
- Find structures in data,
- See the network diagram and edit it as needed,
- Incorporate expert judgment with results from data,
- Do what-if analyses, and
- Summarize the effects of variables on a target or dependent variable.

Some packages will do even more, such as:

- Deal with "censored" data (data that is deliberately omitted, as in questions that are skipped in questionnaires),
- Cluster variables based on similarities,
- Create latent variables from clusters,
- Cluster respondents based on clustered variables,
- Find "optimal policy,"
- Evaluate alternatives in terms of costs,
- Include constraints on variables,
- Include temporal (time sequence) relationships,
- Include nodes representing equations and many other non-data driven factors.

Admittedly, trying to do everything listed with one package will land you squarely at the more expensive end of the price continuum. And so far, no program has approached the "ideal" spot where costs are low and capabilities manifold and robust. Again, a number of programs offer trial versions that at least give some sense of what they can do. If cost is an issue, with careful searching, you should be able to combine two or more lower priced options to do all that you yourself find important.

Well worth the effort to learn

Overall, even if there is some need for exploration, the results are manifestly worth it. These are incredibly powerful methods and their capabilities are being expanded constantly. It may take a little time to master them, but Bayes Nets look very much like the next wave in data analysis, truly the "best newest thing." They are eminently well worth getting to know and putting to use. ●

Steven Struhl is principal at :
Converge Analytic, Philadelphia, PA.
He can be reached at smstruhl@gmail.com
Or smstruhl@convergeanalytic.com