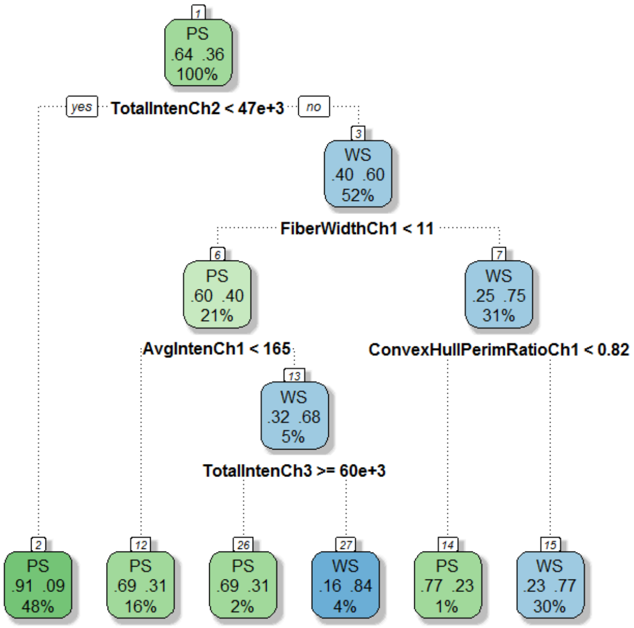
Southern New Hampshire University

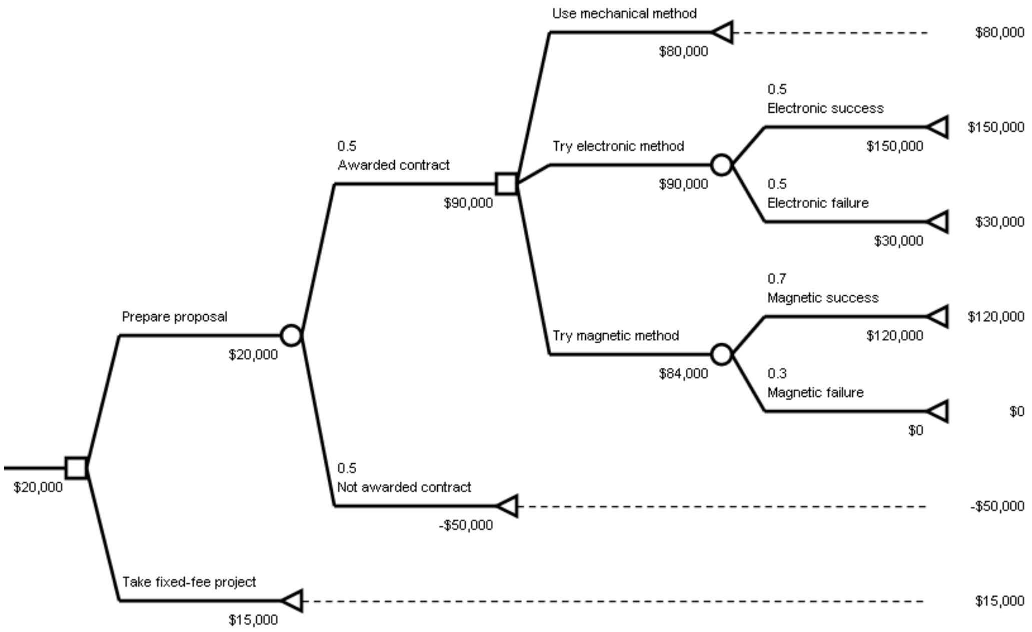
DAT 520 Module Five Overview

Decision trees come in two main types, as you saw in last module’s assignment. The most well-known are the bottom-up diagrams of recursively partitioned data that you can produce with Rattle. The other variety of decision tree is a top-down graphical model for sequential decisions, which you were exposed to in the von Winterfeldt paper and can create with TreePlan.



Rattle 2013-Jun-19 15:43:30 Joe.Rickert

TreePlan Decision Tree Probability-Weighted Rollback Values



We are calling the first variety **bottom-up decision models** because they are made from raw data, a.k.a. the “bottom.” They are a 100% pure form of data mining. **Top-down tree models** are constructed from known results and are worked backwards to find the most effective, lucrative, or lowest risk solution to the research question. Bottom-up modeling requires you to have raw data in hand before you embark. Top-down modeling requires you to have costs, probabilities, utilities, and/or payoffs in hand before you embark.

These two types of models are sometimes interchangeable to answer a research question and sometimes not. It depends on what you are trying to do and the data that you have. Each style has its uses, and each requires a particular form of data prep. As of this writing, there is no literature that directly relates these two types of decision trees, but we may do so on our own because everything we have done so far is a type of Bayesian network.

Bayesian networks are collections of conditions that produce observable results, a finite math concept. Sometimes the transitions from one state to another are not known, just the probability or value of the overall outcomes, and we have to guess the internal process. This mimicry of reality is a universal principle of science. Things have cause and effect. Sometimes, we may not know which is the cause or the effect, but we can see the correlation and try to make sense of it probabilistically. They are called Bayesian networks because whenever you combine two or more related probabilities into a conditional system, you need Bayes’ theorem. Thus, a Bayesian network. There are many ways to model these networks. For this course, we concentrate on just two ways: bottom-up and top-down tree models. We will look at the stats and terminology behind them, and starting this module, the software needed to visually represent, calculate, and optimize them.

Your final project needs to be one of these two types of decision tree models, i.e., you have a *decision* to make about your chosen data set and research question. Do you use the data to produce probabilities and then construct a top-down model with TreePlan? Or do you use the data as the basis for data mining with Rattle and produce a bottom-up decision analysis?

The thing to keep in mind is what type of data you have. If you have aggregate data, and you know the specific end points but not which one(s) to pick as the best, then most likely you have a top-down decision tree modeling situation. If you have raw data that has not been aggregated, and you do not know the end points, then most likely you have a bottom-up decision tree modeling situation. If you know the endpoints as a list of options, but you do not necessarily know the path of decisions that need to be made to choose the best outcome, then you are again in a recursive partitioning, bottom-up decision tree modeling situation. If you have aggregated data but no way to construct a probability for occurrences within that data, then most likely you have a problem: you would need another data source to assist you with determining the frequency of events.

Image source: Wikimedia Commons

Thomas Bayes, 1701–1761

You may encounter a situation in which the only information you have is the frequency of an event. How would you construct a decision model from that type of scant knowledge? It is possible, using conditional probability rollbacks.

Conditional rollbacks use prior states and their eventual outcomes to determine mathematically which is most dominant. By following the exercise in the videos, you will see not only how to construct a top-down probabilistic tree, but also how to construct the calculations for the partial probabilities, feeding into the tree structure for the conditional probability rollback.

If you followed closely in the past few weeks, the items on the spreadsheet that Professor Wu uses for calculating his probabilities will look familiar. Sometimes, the terminology is a little bit different. Beneath the terminology is the same Bayes' theorem with the same inputs: it is the same logical flow of determining the percentage contribution (its proportion) and the probability associated with each element's contribution.

Refer again to the visual representation of conditional probability on this data visualization website. In this example, you saw the proportions and the probability associated with each proportion; then put them together with Bayes' theorem to see the conditional probability of the balls hitting both platforms to become purple.

With that visualization in mind, let's extend the situation. Why would we use a conditional probability tree instead of doing hand calculations? Imagine that you have a whole chain of events, each with a known count from a known pool (a numerator and a denominator), but those events occur in specific chains that are, say, 10 events long. This could represent an industrial process. It could be an economic theory. It could be a medical situation. It could be a biological pathway. Or they could occur in psychology or sociology or geology or cosmology or linguistics contexts. These chains of events are common and 10 elements is miniscule.

So, imagine that you have all these events that fan out into chains from a single common original event and end in a terminal probability. If you know the individual probabilities along the way and the starting state, you could calculate all of the terminal probabilities. In the balls example, it would be a set of 10 platforms of various sizes according to their proportional contributions to the system, and overlapping only as much as their probabilities, and then however many sets of platforms for how many tree branches there are in the decision scenario.

Would you do this by hand? Only a masochist would do that. You would need to use Excel or write the code yourself, or better yet, use a program specifically designed for these kinds of complex conditional probability rollbacks, such as TreePlan or TreeAge. There are not very many software packages for this out there in the wild, but they do have their niches.

This module, you will gain experience with constructing a conditional probability tree, performing the rollback and discussing the meaning of the results. It is not an easy module, but if you understand the materials in this module, you will be able to tackle most decision scenarios because you will know the conditional probability math behind them.

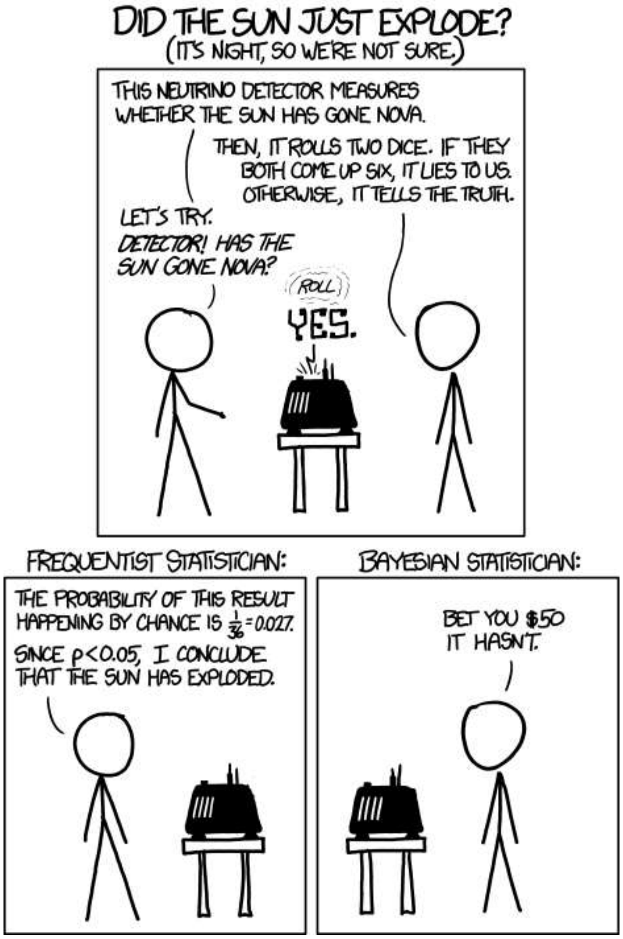

DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)
THIS NEUTRINO DETECTOR MEASURES WHETHER THE SUN HAS GONE NOVA.
THEN, IT ROLLS TWO DICE. IF THEY BOTH COME UP SIX, IT LIES TO US. OTHERWISE, IT TELLS THE TRUTH.
LET'S TRY. DETECTOR! HAS THE SUN GONE NOVA?
(ROLL)
YES.
FREQUENTIST STATISTICIAN:
THE PROBABILITY OF THIS RESULT HAPPENING BY CHANCE IS $\frac{1}{36}$=0.027. SINCE $p<0.05$, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.

Image source: xkcd.com