Dylan Hutchison
CS 810B Causal Inference
Project Proposal due 29 October 2012

A Comparison between Dempster-Shafer and Bayesian Approaches to Soft Evidence

Dempster-Shafer theory offers an alternative approach to probability through the concept of belief functions which generalize probability functions. The probability of a proposition A is classically defined as $\frac{\#\text{ of instances where A is \textbf{possibly true}}}{\text{total }\#\text{ of instances}}$. The belief of proposition A is $\frac{\#\text{ of instances where A is \textbf{provably true}}}{\text{total }\#\text{ of instances}}$, offering a lower bound on classic probability. An upper bound can be calculated from the related concept of plausibility, $1 - \frac{\#\text{ of instances where A is \textbf{never true}}}{\text{total }\#\text{ of instances}}$.

The two functions are equal when we have a complete probability model, that is, we have enough information to generate the joint probability table (JPT) for all variables. Under the Bayesian assumption that a variable is independent of all non-descendants given its causes, we can generate the JPT if we have enough information to write the conditional probability table (CPT) for every variable conditioned on its parent causes. The situation gets more interesting when we do not have enough information to write the JPT. There are two instances where this occurs:

1. We don't know the prior distribution of a variable conditioned on its parents. Bayesian analysis tells us to assume a default uniform prior distribution as it is all that we can infer given no evidence (it's the most likely guess we can make given no information).
Dempster-Shafer analysis tells us instead to impart no belief to any single state of the variable because we have no epistemological reason to believe anything about the variable. Viewing belief as a lower bound on probability, the two approaches are compatible.

2. We have soft (uncertain) evidence on a variable. Bayesian analysis uses likelihood ratios and the notion of virtual children to update probabilities. Dempster-Shafer theory will also update its belief and plausibility values depending on the evidence. Examples of this kind tend to be complicated as we need special kinds of evidence or discrete variables that take on more than two values to differentiate between the two methods here.

For this project, I intend to accomplish the following goals:

1. Create a Java tool to compute belief, plausibility and other interesting values given a probability mass over the power set of the values of a discrete variable. I will also use the tool to combine different masses using Dempster's Rule of Combination.

2. Build new functionality into the probability package of the open source project aima-java[1]. Currently it can only evaluate queries on Bayesian networks with complete probability models. My extension will enable it to answer queries given partial evidence using likelihood ratios.

3. Use both tools to compute beliefs and probabilities in several case studies. These examples should provide insight into the where the two theories agree and diverge, and how they might complement each other when seeking to infer information given some partial evidence. Ideally, I will discover how the two approaches can be combined to obtain greater information.
Here are the examples I plan on using under circumstances of different types of evidence such as none, single certain, single uncertain, multiple certain, and multiple uncertain.

   - Boolean single variable – Cancer; states = true, false
   - 3-state single variable – Murderer; states = Paul, Peter, Mary
   - Alarm network – earthquake, burglary, alarm, news, different kinds of testimony on alarm

4. If possible, I will attempt to integrate Dempster-Shafer bounds into the aima-java project. This is a stretch goal; I will first have to determine whether it is feasible and advantageous to do so.

I would also like to comment on how the two theories might work in the continuous realm and on computational complexity issues when scaling to larger networks, but I doubt I will make it that far given the project's timeline.

---

[1]http://code.google.com/p/aima-java/