Complex Predictive Networks

# DECISION MODELS

Start by determining the choices that we have available. Determine the set of information, , that could affect those decisions. Often will not be best determined directly, but will need to be inferred probabilistically from a set of potentially acquirable information, , and other information, , that is possibly from a potentially changing universal data source, plus any knowledge that is always assumed.

# PREDICTIVE MODELS

The probability that a particular story, is true given that a particular case, , and other information, , is known is:

Determine for every defined and for every potentially relevant and .

# IN PRACTICE

1. Creating a separate function for every possible situation and case makes understanding and updating changes easier. Combining functions usually creates more consistency and lessens the upfront work.
2. Consistency checks should usually be made across predictions and by comparing results to direct human analysis of .
3. Use functions that take any and for one or more , and return the probability of for one or more , along with any other useful information for later computational efficiency (e.g., parameters of a Bayesian conjugate distribution) .
4. Where possible, break functions into reusable pieces. For example, cases of probabilistic independence where:

where are all cases in that were not in , and are all stories in that were not in .

1. Break complex models into multiple models where possible. This will usually also reduce the number of stories and cases to be considered. Don’t create superfluous stories. The minimum number is usually equal to the number of final actions or desired inferences of a model.
2. Create fast evaluation functions for computationally intensive functions.
3. Favor predictive models that accept missing values and can be easily updated with new information and variables.
4. Probability of stories must sum to one, so it can be useful to have an “other story.” Thus, the definition of this story changes when any of the stories changes or when a new story is added.

When there are many cases that share common facts, it can be helpful to manually build a Bayesian Network to find the conditional probabilities around a given story. The root is the story. The leaves are the facts. The nodes are sub-stories. The sub-stories can often be re-used for other stories. The structures of Bayesian Networks are relatively to update when a new fact type is present. Within a Bayesian Network, a Dirichlet Distribution (or Beta when applicable) is an easily-updatable Bayesian conjugate distribution. Ideally, the root and each node of a Bayesian Network would be broken down sub-stories with 4 facts if each fact and sub-story has 2 outcomes, or 3 facts if each fact and sub-story has 3 outcomes. Try to use verifiable sub-stories so that the relationships between the facts and story (and sub-stories) can be auto-updated with data. If sub-stories are not verifiable, it may make sense to not include the sub-story and instead put more work into estimation. To create a Bayesian Network of this type, ensure that each level (A,B,C…) has the following property:

Etc.

-ME (first upload 3/12/16, last update 4/9/16)