Probabilistic Lattice Learning and Backward Chaining

Internal UROP Report

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology

Tristan Thrush

Supervisor: Patrick Winston

December 16, 2016

I developed a model, within Genesis (Winston, 2012), for the human ability to learn general rules from specific observations (with a notion of probability based on how likely the generalization is to be correct), and chain them, keeping in mind how the probabilities of individual rules aggregate. To do this, I created a lattice learning system to learn probabilistic rules, and a backward chaining system to use them.

I define my probabilistic backward chaining model as follows. Fuzzy rules are of the form *if x then y with probability P*. Fuzzy assertions are of the form *z with probability P*. During the backward chaining process, fuzzy rules are added to the probabilistic goal tree as normal rules are in a normal goal tree. The probability of a probabilistic goal tree's hypothesis is defined as the probability that any of the possible paths to the hypothesis are valid. This is $P(\text{goal tree hypothesis}) = P(\text{path}_1 \cup \text{path}_2 \cup ...)$. It is possible to get a lower and upper bound of this probability by considering the cases of mutual exclusivity and the case complete inclusivity. The probability of any given path is defined as the probability that all of its necessary hypotheses are valid. This is $P(\text{path}_n) = P(\text{hypothesis}_{n,1} \cap \text{hypothesis}_{n,2} \cap ...)$. Luckily, such a joint probability can be recursively computed with conditional probabilities from the rules in the goal tree, if we assume hypotheses are only dependent on their children hypotheses. For example, if a certain path to the goal tree hypothesis, H_G, relies on a child hypothesis, H_0, and so on, then $P(\text{path}) = P(H_G \cap H_0 \cap ...) = P(H_G|H_0)P(H_0|.)...P(H_k)$, where H_k is an assertion (or several joint assertions) that the chainer grounds out on. And we have access to probabilities such as $P(H_G|H_0)$, because they are given by probabilistic rules that are contained in the goal tree. Note that the implementation of my model makes another simplifying assumption to find probabilities: It assumes that paths are independent. Defining the probabilistic backward chaining model is only part of the solution; I also address the issue of learning probabilistic rules from stories.

I present a model to learn probabilistic rules from generalizations about cause and effect relationships (or even just correlations) that Genesis identifies. My method utilizes a data structure that I call the experience tree. I have not coded a way for Genesis to identify causal relationships (they are necessary inputs to the experience tree, which serves as a probabilistic rule generator) but I understand that this problem is under consideration by other members of the Genesis group. One can use an experience tree to collect statistical information about more general and more specific forms of an entity in a way such that if a form of an entity is observed, the statistics of a more general form in the data structure reflect this change, but the statistics of a more specific form do not. It is then possible to query the data structure to find the most relevant statistical information about a desired form of the entity. The experience tree idea can be built on by keeping multiple trees for entities that are not of a related form (an experience graph). The experience tree model can generate many more rules than examples given to it, because it uses lattice information from WordNet (Miller, 1995) to deduce the probabilities of situations that it has not explicitly observed. For example, say that we have an example in a story: a cat is picked up and then it scratches its owner. Suppose we have another example: a golden retriever is picked up and then it does not scratch its owner. Given just these two examples, the experience tree has learned the probabilistic rules for several more situations (albeit they will not be very robust if there are just two examples); here are a few: if a cat is picked up, it scratches its owner with probability 1; if a dog is picked up, it scratches its owner with probability 0; if a golden retriever is picked up, it scratches its owner with probability 0; if an animal is picked up, it scratches its owner with probability 0.5.

So far, I have implemented the experience graph and its support for queries and insertions. I have also implemented a version of the probabilistic backward chainer that handles special cases of backward chaining problems. In my UROP proposal, I present a story about two potential murderers and some evidence regarding a crime; Genesis can now read the story and explain which person is more likely to be the murderer. It is worth noting that my backward chainer implementation cannot process arbitrary nested logic statements within rules and hypotheses. I have an idea for enabling the backward chainer to handle arbitrary logic statements, but it would require comparing arbitrary logic statements for equivalency, which I think may be harder that CSAT (an NP-hard problem).

References

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Winston, P. (2012). The next 50 years: a personal view. *Biologically Inspired Cognitive Architectures*, 1:92–99.