COMPSCI 546: Assignment 6

ppachpute@umass.edu

December 2019

1 Questions

1. Description of the system, design trade offs, questions you had and how you resolved them, etc. List the software libraries you used, and for what purpose.

   - One of the decision that had to be taken was around the placement of class PriorNode in the inference class hierarchy. It resembled some of the features of BeliefNode but as such it behaves independently of all the nodes in the hierarchy. Hence, I created PriorNode as direct subclass of QueryNode.
   - When calculating the probability for random distribution, I generated a random number between 0 to 1 using Math.random function and then divided that with the sum of all the probabilities so that sum of the probabilities of all the priors would be 1.

2. What is the difference between your two query runs? Why would it be that way? Be specific.

   - The output of uniform probability prior is exactly same as that of BELIEF_AND node (ordering of document is same relative scores have been changed) while output of random probability prior is slightly different.
   - When score of the two documents is near each other at that time prior probability plays a crucial role and it can decide the ordering of the documents in the final result.
   - For example, in uniform probability distribution, 2nd ranked document is henry_v:4.1 it has a score of -29.052760150338614. Prior uniform probability of that document is 0.001336898395721925. While in random probability distribution, henry_v:4.1 is 7th ranked document with the score of -30.091194461623953. If we look at the prior random probability of that document, it is 0.0013133366645229048. Here, we can see that the random prior probability is less than uniform probability and hence the document is ranked lower .

3. How should the priors be stored in the index? Raw probabilities? Log probabilities? Some other value? What should drive your choice? Be specific.

    - The file from which the priors are obtained contains raw probabilities
    - In the Prior node, I take the log of the probability returned from the index. I have chosen this way of doing it because it is more consistent with rest of my implementation of inference network.
    - I chose to take log in the Prior node because all of the other nodes in the inference network work with log probabilities. They expect their input to be log probabilities and they output the result in log probabilities. Hence it made sense to be consistent in that implementation.
    - These priors could be arbitrary values in the domain of 0 & 1 hence it is possible that some of the values could be very small. When dealing with such values we might face issues such as Numeric Underflow. To avoid such issues it is better to work in the log space.