COV884
Assignment

Problem Statement -> Given a bayesian network and sample datasets (with a few missing or unknown values), predict the probability distribution or CPT for each node of the bayesian network.

Method Used -> We used Expectation maximisation to do the following task. That is given the distribution of the data by the input dataset, what probability distribution best fits them and also maximises the expectation of the whole distribution at the same time. This is why a few examples which we never saw got uniform probability values in their CPT table, because uniform distribution is the best distribution for expectation maximisation, given the only constraint is the sum of probability to be 1.

Conventions -> In implementation, we will only talk in terms of count (not probability). The probability can always be calculated from the counts by dividing it by total count.

Initialisation -> After reading the data, we initialised the count of each cell in CPT to be equal to 0.01 for Laplace smoothing. Also since the sample data also have some values missing, we distributed the probability of the missing values uniformly among all possible values thus giving some weights to each sentence. Later we used the weight of the sentence to increase any count(instead of just using ‘1’).

The E- step -> At this step, we visit each input sample and then find the new probabilities of the missing value of that sample being any one of the possible values using the CPT available to us. Thus we can replace the old sample weights with these new sample weights.

The M-step -> At this step, we visit each cell in our CPT and we update the value of the cell by iterating through the dataset and counting the number of times the given condition occurs. Here we can use the old sample weights to update our CPTs instead of recreating them again from the start.

Implementation -> While implementing, we kept only the count of the occurrence as the entry in the CPT table. This count was nothing but the sum of the weights of the sentences which satisfies the given condition. All the possible values of every variable as converted into indices, from 0 to n-1 where ‘n’ are total number of different values that variable can take. Eg -> for “low”, “average”, “high”- 0 mapped to “low”, 1 mapped to “average”, 2 mapped to “high”.
The code keeps of repeating the E-M steps for 590 seconds from the start after which the code stops running and the latest CPTs and stored in the output file.

Discussed With -> Saket Dingliwal, Rahul Agarwal, Ronak Agarwal, Sagar Goyal, Rohit Raj