CS5340 Lab 4: Importance and Gibbs Sampling

Name: Niharika Shrivastava, Email: e0954756@u.nus.edu, Student ID: A0254355A

1. Importance Sampling: `sample_step()`

- Go through each node in topological order. - For each node, get the sample distribution and update it by considering previously computed samples as evidence. - Choose a random sample value for the node given the probability values of all other nodes.

2. Importance Sampling: `get_conditional_probability()`

- Update each proposal factor with given evidence. - Nodes are all the keys of the updated proposal factors. - Compute `num_iterations (N)` samples and store the frequency of each sample in a counter. - For each sample (i), compute the importance weight numerator as `target_factor(i) / updated_proposal_factor(i)`. - Normalize importance weights by dividing by sum of all computed weights. - Fill each entry in the output table as the importance weight of the sample multiplied by the frequency of the sample. Normalize all probabilities in the end.

---

1. Gibbs Sampling: `sample_step()`

- Go through each node in topological order. For each node, get the local factor. - Remove the current node value from the previous sample and treat it as evidence for the node in the current iteration. - Update the local factor of each node with this evidence and normalize this updated factor's probabilities. - Choose a random sample value for the node given the probability values of all other nodes.

2. Gibbs Sampling: `get_conditional_probability()`

- Update each node's conditional probability by marginalizing every node which is not in it's markov blanket. - Update each conditional probability with given evidence. - Nodes are all the keys of the updated conditional probability. - Compute `num_burn_in` samples and ignore them. - Compute `num_iterations (N)` samples and store the frequency of each sample in a counter. - Fill each entry in the output table as the frequency of the sample. Normalize all probabilities in the end.