Vishwanath Dattatreya Doddamani A0286188L e1237250@u.nus.edu

# CS5340 ASSIGNMENT 4 – Monte Carlo Inference

1. **Part 1: Importance Sampling**
   a. **_sample_step() :** This function takes input of nodes in topological order along with proposal factors. Iteratively it goes through each node, and performs factor_evidence of the proposal factor of corresponding node and evidence being the sample dictionary. Now we get the distribution of the node, which is used to generate sample using np.random.choice(). Then its value is added to the dictionary and sample dictionary is returned.
   b. **_get_conditional_probability():** Once target factors, proposal factors, evidence, and number of iterations is received, this function performs factor_evidence of both target and proposal factors. Then using helper function, it generates graph for factors. Helper function generate graphs based on keys in the proposal factors for nodes and adds edges for the key with all the random variables existing in the proposal factor. Using networkx topological_sort method, the topological order of nodes is identified for the graph. Then for each iteration, sample is generated for each node. Then corresponding value of proposal distribution ($\boldsymbol{\delta}$) and target distribution (qX) is looked up. Weight (w) is then calculated and stored. Finally, after the number of iterations, conditional probability is calculated using the formula:

$$p(x_F \mid x_E) = \frac{\sum_m w_m \delta(x^{(m)})}{\sum_m w_m}$$

Source: Lecture slides

2. **Part 2: Gibbs Sampling**
   a. **_sample_step():** This is done similar to part 1. In factor_evidence, the node for which the sample value is being generated is removed from the samples dictionary and then factor_evidence is performed. Once the distribution of the node is received, sample for that node is generated using np.random.choice().
   b. **_get_conditional_probability():** The inputs for this function is different from part 1. Here number of burn in period is also given along with number of iterations. Since the proposal factors consist of all the random variables, it is important to calculate Markov blanket of the node. In this function, the graph is created similar to part 1. Then using helper function, Markov blanket is calculated. Markov blanket helper function looks for parents, children and parents of children of a node and then returns all these nodes as Markov blanket of the requested node. Here samples are generated until burn in period is over. Then using assignment_to_index function, the index of the sample value generated is calculated. That particular index counter is increases by one, every time that sample is generated. Finally, the conditional probability is calculated: number_of_samples / number_of_iterations.

3. **Challenges faced:**
   Figuring the sample logic takes some time. In Gibbs sampling, it has to be taken care that while sampling, topological order of nodes is sent, however while assigning the final values, the sorted order of nodes has to be considered.

4. **Conclusion:**
   This assignment was fairly easy compared to other assignments. It gives holistic picture of sampling making equations in lecture slides much clear.