CS726 Programming Assignment 1 - Report

Anshu Arora
Kshitij Vaidya
Raunak Mukherjee

1 Approach

1.1 Triangulation

We desire to convert an undirected graph into a chordal graph. The steps taken are as follows

- We iteratively select the node with the minimum degree (has the fewest neighbors). This helps us minimize the number of extra edges to be added later
- The node chosen along with all of its neighbors are transformed into a clique.
- Once this is done, we remove the selected node from the working of the algorithm
- The algorithm then proceeds for the next iteration, and once all nodes are done processing, the algorithm identifies the maximal connected subgraphs- to extract the clique

1.2 Junction Tree Construction

Having triangulated the graph, we now proceed to construct the junction tree from the maximal cliques obtained. Our step-wise approach is as follows-

- First we ensure that triangulation has been performed and the maximal cliques are available
- We then find the common variables between the cliques, that is the variables shared between both the cliques.
- Next the directed separator set is defined
- To create the junction tree, we need to ensure that the cliques are connected (without forming cycles) and the nodes with largest shared variables are prioritized.

- The algorithm iterates through the sorted edges. If adding an edge between two cliques does not form a cycle, we include it in the junction tree. This guarantees that the resulting structure is a tree and not a general graph.
- The junction tree is then stored as a list of tuples

1.3 Computing Z value

We require the value of the partition function to normalize the probabilities obtained by factorization. It is calculated as follows

$$Z = \sum_{X} P(X) \tag{1}$$

where:

- X represents all possible assignments of the random variables.
- $P(X)$ is the unnormalized probability of X.

In terms of factor functions $\phi_i(X)$, the partition function can be rewritten as:

$$Z = \sum_{X} \prod_{i} \phi_i(X) \tag{2}$$

where $\phi_i(X)$ represents the factor values. As suggested and taught in class, we are using the message passing algorithm to calculate the same

1. We assign initial messages between variable and factor nodes
2. Each factor node sends messages to neighboring variable nodes based on incoming messages.
3. Each variable node sends messages to neighboring factor nodes based on received messages, and only sends these messages once it receives from all its neighboring nodes.
4. The message updates are repeated until messages stabilize or a fixed number of iterations is reached.
5. We then compute beliefs by estimating marginal distributions at each variable node using incoming messages.
6. Finally, we estimate the partition function Z by using computed marginals and factor potentials.

1.4 Marginal Probability

We calculate the marginal probability by using the message passing algorithm, the steps for which are highlighted as follows.

1. Each node initializes messages to its neighbors
$$m_{i \to j}(x_j) = 1, \quad \forall x_j$$
2. The message from node i to node j is computed as:
$$m_{i \to j}(x_j) = \sum_{x_i} \psi_{i,j}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus j} m_{k \to i}(x_i)$$
where $\psi_{i,j}(x_i, x_j)$ is the pairwise potential, and $\mathcal{N}(i)$ represents the neighbors of i.
3. We iterate the algorithm till convergence, and then proceed to calculate the 'belief' of the node as follows
$$b_i(x_i) = \prod_{j \in \mathcal{N}(i)} m_{j \to i}(x_i)$$
4. To obtain the marginal probability we then divide by the partition function Z
$$P(x_i) = \frac{b_i(x_i)}{Z}$$
5. These obtained marginals are then used for inference

1.5 MAP Assignment

1. Unlike the case of marginal inference, we do not sum, and instead take the max operation
$$m_{i \to j}(x_j) = 1, \quad \forall x_j$$
2. To update the message, we use the rule that from node i to node j
$$m_{i \to j}(x_j) = \max_{x_i} \left[\psi_{i,j}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus j} m_{k \to i}(x_i) \right]$$
3. We repeat the messages until convergence
4. We compute the belief at each node as follows
$$b_i(x_i) = \prod_{j \in \mathcal{N}(i)} m_{j \to i}(x_i)$$
5. We obtain the MAP estimate by selecting the value that maximizes the belief:
$$x_i^* = \arg\max_{x_i} b_i(x_i)$$

2 Results

1. We successfully created the triangulated graph from the undirected graph
2. Junction tree was correctly formed
3. We were able to calculate the value of Z correctly in the first test case, however while trying on the larger test cases we were facing issues
4. The marginals were also correctly calculated for the sample testcase provided. Further testcases could not validate the correctness of the code
5. All other functions in the inference class were defined, however the working has some issues

3 Contributions

- **Kshitij:** Extensively worked upon writing the code for the intricate functions and flagged many issues in the code
- **Raunak:** Created basic framework of the code, by filling in the functions and contributed in high-level debugging of the complicated functions
- **Anshu:** Created the second iteration of the basic framework of the code and created the assignment report