Programming Assignment 1: Advanced Machine Learning

Advanced Machine Unlearning
Tejas Sharma: 22B0909
Muskaan Jain: 22B1058
Ojas Maheshwari: 22B0965

Contents

1 Introduction

Given an undirected graph with cliques and their associated potentials, this assignment focuses on computing marginal probabilities and the Maximum A Posteriori (MAP) assignment using message-passing algorithms. The input includes predefined cliques, which are complete subgraphs, ensuring structured factorization. The process involves triangulation, junction tree construction, and message-passing algorithms for inference.

2 Triangulation

2.1 Overview

Triangulation transforms an undirected graph into a chordal graph by adding edges such that every cycle of length greater than 3 has a chord. This ensures the graph can be decomposed into cliques, which is essential for constructing a junction tree.

2.2 Steps

1. **Identify Simplicial Vertices**: A vertex is simplicial if its neighbors form a clique. Simplicial vertices are removed iteratively to determine the elimination order.
2. **Elimination Order**: If no simplicial vertices are found, the vertex with the minimum degree is selected for elimination. If multiple simplicial vertices exist, they are eliminated simultaneously in arbitrary order.
3. **Add Edges**: During elimination, edges are added between the neighbors of the eliminated vertex to maintain the chordal property.
4. **Maximal Cliques**: The cliques formed during elimination are identified as maximal cliques.

2.3 Pseudocode

Algorithm 1 Triangulation and Maximal Clique Extraction

```
Initialize adjacency list and empty list for maximal cliques.
while adjacency list is not empty do
    Find simplicial vertices.
    if simplicial vertices exist then
        Add simplicial vertices to the elimination order.
        Remove simplicial vertices and their edges.
        Add the clique formed by the simplicial vertex and its neighbors to maximal cliques.
    else
        Find the vertex with the minimum degree.
        Add missing edges between its neighbors to triangulate.
        Add the vertex and its neighbors to maximal cliques.
    end if
end while
Remove redundant cliques.
Return maximal cliques.
```

3 Junction Tree Construction

3.1 Overview

A junction tree is constructed from the maximal cliques of the triangulated graph. The tree must satisfy the running intersection property, ensuring that variables common to two cliques (separator variables) are present in all cliques on the path between them.

3.2 Steps

1. **Create Nodes**: Each maximal clique becomes a node in the junction tree.
2. **Connect Cliques**: Connect cliques that share common variables.
3. **Assign Potentials**: Assign potentials to each node as the product of all potentials in the original graph whose variables are subsumed by that node.

3.3 Pseudocode

Algorithm 2 Junction Tree Construction

```
Initialize an empty adjacency list for the junction tree.
for each pair of maximal cliques do
    if cliques share common variables then
        Connect the cliques in the junction tree.
    end if
end for
Assign potentials to cliques based on the original graph.
Return the junction tree.
```

4 Marginal Probability

4.1 Overview

Marginal probabilities are computed using the sum-product algorithm on the junction tree. The algorithm propagates messages between cliques to compute the marginal distribution for each variable.

4.2 Steps

1. **Initialize Potentials**: Assign initial potentials to the cliques.
2. **Message Passing**: Propagate messages from leaf cliques to the root and back.
3. **Marginalization**: Marginalize out all other variables to compute the final probabilities.

4.3 Pseudocode

Algorithm 3 Marginal Probability Computation

```
Initialize all factors with potentials from the junction tree.
for each variable in elimination order do
    Collect factors containing the variable.
    Multiply factors and marginalize out the variable.
    Update the remaining factors.
end for
Compute the partition function Z.
Normalize the marginals using Z.
Return the marginals.
```

5 MAP Assignment

5.1 Overview

The MAP assignment is the most probable assignment of variables in the graphical model. It is computed using the max-product algorithm, which maximizes the joint probability distribution.

5.2 Steps

1. **Initialize Potentials**: Assign initial potentials to the cliques.
2. **Message Passing**: Propagate messages using the max-product rule.
3. **Backtracking**: Determine the assignment that maximizes the joint probability.

6 Top k Assignments

6.1 Overview

The top k assignments are the k most probable assignments of variables. These are computed by extending the max-product, message-passing algorithm to keep track of the top k configurations at each step.

6.2 Steps

1. **Initialize Potentials**: Assign initial potentials to the cliques.
2. **Message Passing**: Propagate messages while maintaining the top k assignments.
3. **Normalization**: Normalize the probabilities using the partition function Z.

6.3 Pseudocode

Algorithm 4 Top k Assignments

```
Initialize all factors with potentials from the junction tree.
for each variable in elimination order do
    Collect factors containing the variable.
    Multiply factors and marginalize out the variable, keeping top k assignments.
    Update the remaining factors.
end for
Normalize the probabilities using Z.
Return the top k assignments.
```

7 References

- GitHub copilot, for code autocomplete
- Lecture slides, for the algorithm