Project Proposal

for

Implementation of Junction Tree Inference Algorithm in Typescript

Version 1.0

Prepared by Jordan Emslie - 20600152

March 8, 2020

Statement of interest:

I am interested in this project to implement Junction tree inference algorithm in Typescript. The main reason being, I am passionate in the space of inference and prediction so being able to investigate the semantics of how the Junction Tree Inference Algorithm works seems very interesting to me. I choose to develop in Typescript as I've become very proficient of this language after developing with it over the summer and during my Capstone project. Also, it does not require you to be strictly typed.

Scope:

During this project what I plan to have finished is a working implementation of the Junction Tree inference Algorithm in Typescript. This will involve the creation of Bayesian networks with directed acyclic graphs (DAG). Furthermore, the secondary structures will be created. This being where the Junction Tree will be created from the DAG of the Bayesian networks. Next, the process of Initialization, Propagation, Marginalization will be completed in order to compute the probability of a single variable. This is the main goal of the project. However, if time permits then observations will be added allowing us to predict the probability of a variable of interest given observations. This will all be developed with the concept of Probability Propagation in Trees of Clusters (PPTC).

Milestones:

Milestone	Details	Estimated Completion
Implement Bayesian Networks and DAGs	Create data structures to create the DAGs and the Bayesian Network.	March 13th
Construct the Junction Tree	Implement the J-Tree structure and add functionality to convert the DAG of the Bayesian Network into a Junction Tree structure using the four steps below. 1. Construct an undirected graph (moral graph) from the DAG. 2. Selectively add arcs to the moral graph to form a triangulated graph. 3. From the triangulated graph, identify select subsets of nodes, called cliques. 4. Build the optimal junction tree	March 18th
Initialization / Propagation / Marginalization	Initialization process resulting in an inconsistent J-Tree. Propagation process resulting in a consistent J-Tree. Finally, add functionality for marginalization allowing you to use the consistent J-Tree to compute Pr(A) for each variable of interest A.	March 22nd
Initialization with Observations / Observation Entry / Normalization / Presentation Prep	Now we add observations to Initialization Incorporate observations into the J-tree Add Normalization to compute Pr(A|B) for a variable A of interest with observations B.	March 31st

References

Huang, Cecil, and Adnan Darwiche. “Inference in Belief Networks: A Procedural Guide.” *International Journal of Approximate Reasoning*, vol. 15, no. 3, Jan. 1996, pp. 225–263., doi:10.1016/s0888-613x(96)00069-2.