Assignment 1: Probabilistic Inference with Bayesian Networks and Markov Networks

CS786 – Spring 2012

Out: Thursday, May 24, 2012
Due: Wednesday, June 6, 2012, at midnight

This assignment consists entirely of programming questions to be done in the Java programming language. You will implement some basic functions for inference with Bayesian networks and Markov networks. Submission is done electronically via Marmoset. The Marmoset server will automatically compile your code and run various tests to verify its correctness. Instructions will be posted shortly on the course website about how to interact with Marmoset.

Download from the course website the definitions of several Java classes to implement a basic library for probabilistic inference with Bayesian networks and Markov networks. Your job is to fill in the core methods of those classes.

1. **Factor class (F):** This class implements a factor (which may correspond to a conditional probability table in a Bayesian network or a potential in a Markov network. Implement the following methods to perform basic operations on factors:

   (a) **restrictedFactor = F.restrict(factor,variable, value):** restrict a factor to the entries consistent with the value of a certain variable.

   (b) **productFactor = F.product(factor1,factor2):** multiply two factors.

   (c) **resultFactor = F.sumout(factor,variable):** sum out a variable in a given factor.

   (d) **normalizedFactor = F.normalize(factor):** normalize a factor by dividing each entry by the sum of all the entries. This is useful when the factor is a distribution (i.e. sum of the probabilities must be 1).

2. **Probabilistic Graphic Model class (PGM):** This is an abstract class that can be instantiated to Bayesian network (BN) or Markov network (MN). Implement the following methods:

   (a) **factor = pgm.variableElimination(queryVariables, evidence):** compute the answer to an inference query. This method should be instantiated for Bayesian networks and Markov networks. Use the greedyOrdering and relevant methods defined below and the operations on factors as subroutines inside variable elimination.

   (b) **variableOrdering = pgm.greedyOrdering(queryVariables, evidence):** return an ordered list of variables indicating the order in which they should be eliminated for inference by variable elimination. This order should greedily minimize the size of the intermediate factor generated at each step. This method should be instantiated for Bayesian networks and Markov networks.

   (c) **relevantBayesNet = BN.relevant(bayesNet, queryVariables, evidence):** find the relevant part of a Bayesian network for a specific query. This method should produce a new Bayesian network that is a subset of the original network that is sufficient to answer the query.

   (d) **trueOrFalse = pgm.independent(variable1, variable2, evidenceVariables):** indicate wether variables 1 and 2 are independent given a set of evidence variables. This methods should be instantiated for Bayesian networks and Markov networks.

Evaluation of your code will be done automatically by Marmoset. Your grade will be equal to the fraction of tests that your code successfully passes.