COMP4680/8650: Advanced Machine Learning: Project 1

August 25, 2014

1 Logistics

1.1 Overview

In this project, you will implement a variety of probabilistic inference methods on a variety of Bayesian networks and run these algorithms on some simple networks specified here. You will need to submit your code and a report describing how you implemented each algorithm.

1.2 Teams

For this project you can work individually, or in teams of 2. If you work as a team, either member of the team can submit your report, please include both team members' names. All reports and code should be submitted through the wattle system.

1.3 Due Date

23:55 on 5 September, 2014. You are allowed to resubmit (upload a new version) as long as the site is open, and your grade will be based on the last version. So if you find a mistake after the 5 September deadline, you may wish to consider the tradeoff between the marks earned and the late penalty (raw score multiplied by 0.75 for up to 1 day late, and multiplied by 0.5 up to 2 days late). This assignment contributes 20% to your final score.

1.4 What to Submit

Your report should have five sections (one for each of the five models above) each of which contains three subsections (one for each of the three algorithms). In each subsection, you should either explain how you implemented that algorithm for that model and roughly how long it took to run (in 1-2 paragraphs), or argue why it is not possible to use that algorithm in that situation in a reasonable amount of time. Please also provide a table for each section as below with the estimated probabilities. Give a comment on how fast you found each

Along with your report, submit your code, along with a short `readme.txt` file that explains how to run it. Your code can be in any language, but should not use any extra libraries beyond basic arrays and methods to print results.

The .latex and .lyx source of the assignment is provided, if you would like to use it to start your report.

1.5 Expectations

It is expected that implementing each algorithm on each model will take on the order of an 1-2 hours, assuming you already understand the relevant material. There are 15 model/algorithm pairs. In addition, you will need to spend some time reviewing the material, and writing the final report.

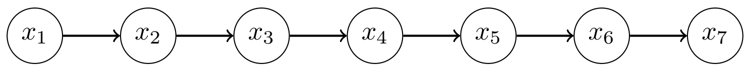
2 Models

2.1 Chain

The first graph to be considered is a simple "chain", of the form

$$p(x_1, x_2, ..., x_n) = p(x_1)p(x_2|x_1)p(x_3|x_2)...p(x_n|x_{n-1}).$$

An example graph, with $n = 7$ is shown as follows.



For this problem, you will use the following probabilities:

$$\begin{aligned} p(x_1 = 1) &= .95 \\ p(x_i = 1|x_{i-1} = 1) &= .95 \\ p(x_i = 0|x_{i-1} = 0) &= .95 \end{aligned}$$

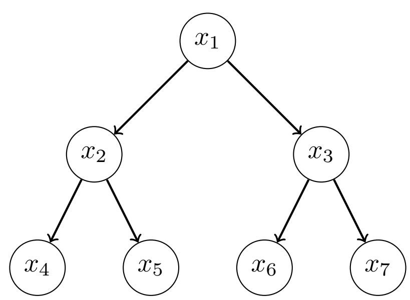
In this project, you will use a chain with $n = 15$.

2.2 Binary Tree

The second graph to be considered is a fully-branching tree. If the tree has L layers, the total number of nodes will be $n = \sum_{i=1}^{L} 2^{(i-1)} = 2^L - 1$. This can be written as

$$p(x_1, ..., x_n) = p(x_1) \prod_{i=2}^{n} p(x_i|x_{\lfloor i/2 \rfloor}).$$

An example graph with $L = 3$ and $n = 7$ is shown as follows.



For this problem, use probabilities similar to those above, namely:

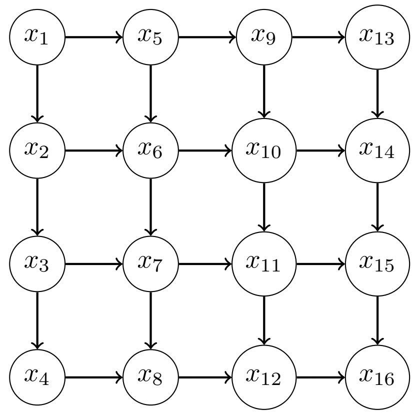
$$\begin{aligned} p(x_1 = 1) &= .95 \\ p(x_i = 1|x_{\lfloor i/2 \rfloor} = 1) &= .95 \\ p(x_i = 0|x_{\lfloor i/2 \rfloor} = 0) &= .95 \end{aligned}$$

In this project, you will deal with a "small" tree with $L = 4$ and $n = 15$ and a "large" tree with $L = 6$ and $n = 63$.

2.3 Grid Graph

The third type of graph to be considered is a $L \times L$ "grid", in which each variable is dependent on those to the left and above it.

An example graph with $L = 4$ and $n = 16$ is shown as follows.



This has the distribution

$$p(x_1, ..., x_n) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3)p(x_5|x_1)p(x_6|x_2, x_5)...p(x_{16}|x_{12}, x_{15}).$$

You can use the following conditional probabilities:

$$\begin{aligned} p(x_1 = 1) &= .5 \\ p(x_2 = 1|x_1 = 1) = p(x_3 = 1|x_2 = 1) &= .95 \\ p(x_2 = 0|x_1 = 0) = p(x_3 = 0|x_2 = 0) &= .95 \end{aligned}$$

For a node i having ancestors j and k, use the following probabilities

$$\begin{aligned} p(x_i = 1|x_j = 1, x_k = 1) &= .99 \\ p(x_i = 1|x_j = 1, x_k = 0) &= .5 \\ p(x_i = 1|x_j = 0, x_k = 1) &= .5 \\ p(x_i = 1|x_j = 0, x_k = 0) &= .01 \end{aligned}$$

Here, you will deal with a "small" grid with $L = 4$ and $n = 16$ and a "large" grid with $L = 8$ and $n = 64$.

3 Algorithms

For each scenario, there are thee inference methods to consider. In some cases, the inference method may not be possible, because it would take an unacceptable amount of time. You should either implement the method, describe how you have done it, and give your inferred results, or you should explain why that method does not work for that particular model.

Report: For the report, you should describe in a few paragraphs how you implemented each of the algorithms for each of the models. If the algorithm cannot work in a given setting, you should explain why not. Note that an algorithm may work on some queries for a given algorithm/model pair, but not others.

3.1 Brute - Force

The simplest sampling method is brute-force summation. Here, you implement a function that takes a full configuration $x_1, ..., x_n$ and returns the probability $p(x_1, ..., x_n)$. Next, you will loop over all the (2^n) possible states and sum the appropriate joint probabilities to obtain conditional probabilities.

3.2 Belief Propagation / Sum-Product

For treelike graphs, you can exactly compute probabilities by running the belief-propagation or sum-product algorithm. In this project, you do not use loopy belief propagation or any other approximate message-passing scheme, and do not use the junction tree algorithm or any variable grouping scheme. The sum-product algorithm should only be used when it can return exact marginals on the original graph.

3.3 Directed Sampling

In this method, you should use ancestor sampling to draw a set of 10^5 samples. Then, do evaluate a conditional probability, you count the number of samples consistent with that probability. For example, if you wanted to estimate $p(x_2 = 1|x_7 = 0)$ and you observed 34,023 samples with $x_2 = 1$ and $x_7 = 0$ and $96,707$ samples with $x_7 = 0$ would would estimate the probability as $\approx .3518$.

This is a randomized method. So please run your algorithm five times for each setting, and show the five different values obtained.

4 Results

Along with your report, you should provide a table of the estimated probabilities for each algorithm on each of the queries listed below for each of the models. If an algorithm cannot work for a given query, place an "×" or "N/A" in that entry of the table.

4.1 Chain

	Chain			
	$p(x_5 = 1)$	$p(x_5 = 1\|x_1 = 1)$	$p(x_5 = 1\|x_1 = 1, x_{10} = 1)$	$p(x_5 = 1\|x_1 = 1, x_{10} = 1, x_{15} = 0)$
Brute Force Inference				
Belief Propagation				
Directed Sampling				

Question: Compare for directed sampling, the mean error of the estimated probability from the true one for the last two queries. Does one of these yield higher errors on average than the other? Why is this happening? (Understanding this will be important later on.)

Implementation note: For reference, in your instructor's implementation, implementing all these algorithms and answering all the above queries in Matlab took about 170 lines, and ran for a few seconds. Make sure you are very comfortable, and have a very clean implementation of these algorithms before proceeding, because the following will build on your implementation here.

4.2 Small Tree

	Chain			
	$p(x_8 = 1)$	$p(x_8 = 1\|x_{12} = 1)$	$p(x_8 = 1\|x_{12} = 1, x_7 = 1)$	$p(x_8 = 1\|x_{12} = 1, x_7 = 1, x_{15} = 0)$
Brute Force Inference				
Belief Propagation				
Directed Sampling				

4.3 Large Tree

Chain				
	$p(x_{32} = 1)$	$p(x_{32} = 1 \mid x_{45} = 1)$	$p(x_{32} = 1 \mid x_{45} = 1, x_{31} = 1)$	$p(x_{32} = 1 \mid x_{45} = 1, x_{31} = 1, x_{63} = 0)$
Brute Force Inference				
Belief Propagation				
Directed Sampling				

4.4 Small Grid

Chain				
	$p(x_6 = 1)$	$p(x_6 = 1 \mid x_{16} = 0)$	$p(x_6 = 1 \mid x_{16} = 0, x_1 = 0)$	$p(x_6 = 1 \mid x_{16} = 0, x_1 = 0, x_{15} = 0)$
Brute Force Inference				
Belief Propagation				
Directed Sampling				

4.5 Large Grid

Chain				
	$p(x_6 = 1)$	$p(x_6 = 1 \mid x_{64} = 0)$	$p(x_6 = 1 \mid x_{64} = 0, x_1 = 0)$	$p(x_6 = 1 \mid x_{57} = x_{58} = ... = x_{64} = 0)$
Brute Force Inference				
Belief Propagation				
Directed Sampling				