Problem Sheet 2

CMPE 58K, Bayesian Statistics and Machine Learning

Instructor: A. T. Cemgil

Due: 22 Oct 2008, 10:00.

Exercises are labelled with greek characters α, β, γ. Each label denotes the type of the question and roughly corresponds to its difficulty with α the hardest. I **don't** expect you to solve all the questions but you should solve **at least one new** question of each type **that you haven't handed in before**. A π denotes questions that have a programming component; these are optional. Don't send any executables or mfiles, just printout the source code and a few example run outputs. However, write your programs clearly as some of these will be used as subroutines in later exercises.

You can still hand in the following questions from the previous assignment sheet, if you haven't handed in them yet:

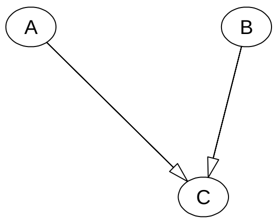
A1.6, A1.7, A1.8, A1.11, A1.12, A1.15

However, these don't count towards the minimum requirement for handing in Assignment 2.

Questions A1.16, A1.17, ..., A1.27 (all questions after and including A1.16) can be handed in and count towards the minimum requirement. If a question does not appear in this list, it means it has expired.

Note that handing in three questions is the minimum requirement; you can always solve and submit more – needless to say the more, the better.

A2.1 (γ, π) (**Explaining away**) Consider the following graphical model:



(a) Here, all variables are binary. $p(A = 1) = 0.9$, $p(B = 1) = 0.3$, $C = A \oplus B$ where $\oplus$ is the xor (exclusive or) operation.

(b) Find the following quantities:

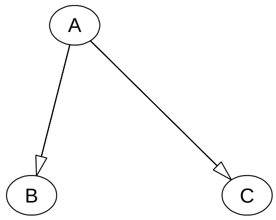
- $p(C)$
- $p(A, B|C)$

(c) Write a program that will compute above quantities for arbitrary $p(A)$, $p(B)$ and $p(C|A, B)$

(d) Write a program that will generate random probability tables $p(A)$, $p(B)$ and $p(C|A, B)$. [Hint: Use the Beta distribution as a prior.]

(e) Using the `randgen` subroutine you developed in the previous assignment sheet, write a program that will generate random instances from the above model.

A2.2 (γ, π) (**Sensor Fusion**) Consider the following graphical model:



(a) How does the associate probability distribution factorise?

(b) Write a program that will generate random probability tables (i.e. parameters) compatible with this graph.

(c) Using the `randgen` subroutine you developed in the previous assignment sheet, write a program that will generate random instances from the above model.

(d) Write a program that will compute the following quantities

- $p(C)$
- $p(A|B, C)$
- $p(C|B)$

A2.3 (β) (**One sample source separation**) Consider the following model

$$\begin{aligned} s_1 &\sim p(s_1) = \mathcal{N}(s_1; \mu_1, P_1) \\ s_2 &\sim p(s_2) = \mathcal{N}(s_2; \mu_2, P_2) \\ x|s_1, s_2 &\sim p(x|s_1, s_2) = \mathcal{N}(x; s_1 + s_2, R) \end{aligned}$$

We will use the following parameters: $\mu_1 = 3$, $\mu_2 = 5$, $P_1, P_2 = 0.5$ and $R = 0.3$.

(a) Draw the graphical model

(b) Find $p(x)$, $p(s_1|x)$, $p(s_1, s_2|x)$, $p(s_1|s_2, x)$ and $p(s_2|s_1, x)$

(c) Suppose we observe $x = 9$. Find $p(s_1, s_2|x = 9)$ analytically. Plot the posterior.

A2.4 (α) (**AR(1) Model**) Consider the following model:

$$\begin{aligned} A &\sim \mathcal{N}(A; 0, 1.2) \\ R &\sim \mathcal{IG}(R; 0.4, 250) \\ x_k|x_{k-1}, A, R &\sim \mathcal{N}(x_k; A x_{k-1}, R) \\ x_0 &= 1 \qquad\qquad x_1 = -6 \end{aligned}$$

(a) Draw the directed graphical model and the factor graph

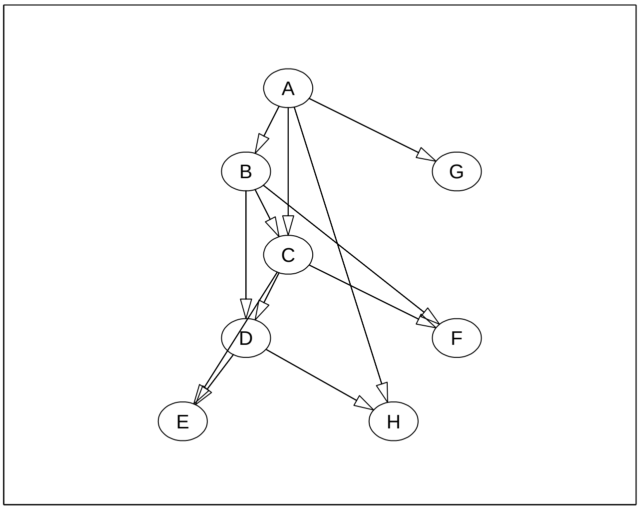
(b) Write the expression for the full joint distribution and assign terms to the individual factors on the factor graph

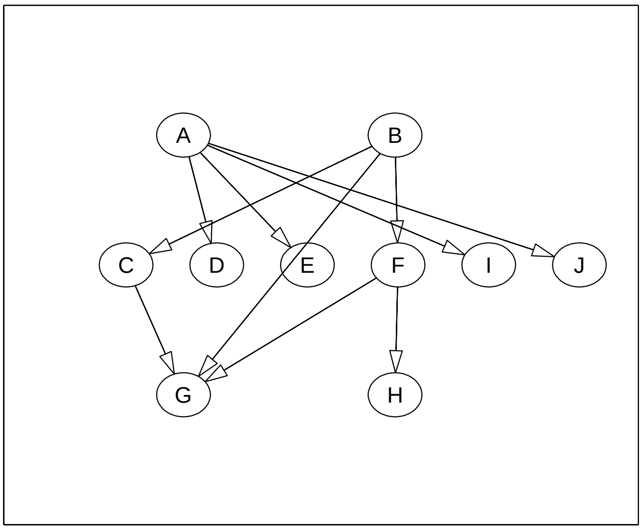
(c) Derive the full conditional distributions $p(A|R, x_0, x_1)$ and $p(R|A, x_0, x_1)$

(d) Derive the joint distribution $p(A, R, x_0 = 1, x_1 = -6)$ and create a contour plot.

A2.5 (π) (**Some Basic Graph operations**) The adjacency matrix of a graph with N nodes is a $N \times N$ matrix with entries 0 or 1, where $a_{i,j} = 0$ denotes a missing directed edge from i to j. Here, we represents an undirected edge when $a_{i,j} = a_{j,i}$.

(a) Find a topological ordering for the following graphs:





(b) Write a program for topological sort with the following specification:

```
TOPOSORT A Topological ordering of nodes in a directed graph

[SEQ] = TOPOSORT(ADJ)
```

```
Inputs :
   ADJ : Adjacency Matrix.
   ADJ(i,j)==1 ==> there exists a directed edge
   from i to j

Outputs :
   SEQ : A topological ordered sequence of nodes.
         empty matrix if graph contains cycles.

Usage Example :
N=5;
[l,u] = lu(rand(N));
adj = ~diag(ones(1,N)) & u>0.5;
seq = toposort(adj);
```

(c) Assuming the graphs encode a Bayesian network with discrete random variables, write a program that counts the number of free parameters.

```
 COUNT_BNET Count the number of free parameter probability tables
 compatible with a graph

 [CNT] = COUNT_BNET(ADJ, SIZES)

Inputs :
   ADJ : N_by N Adjacency Matrix.
   ADJ(i,j)==1 ==> there exists a directed edge
   from x_i to x_j
 SIZES : 1 by N Array. SIZES(i) gives
              the number of states of random variable x_i

Outputs :
   CNT : 1 by N Array of number of free parameters
            for each probability table $p(x_i| parents(x_i))$

Usage Example :
N=5;
[l,u] = lu(rand(N));
adj = ~diag(ones(1,N)) & u>0.5;
      sizes = [2 2 3 2 5];
cnt = count_bnet(adj, sizes);
```

[Hint: You may find the following matlab package useful for visualisation of your graphs: `http://www-sigproc.eng.cam.ac.uk/~atc27/matlab/layout.html`]