UIUC, CS498, Section EA - Autumn 04 - Homework #3

Due Date: Tuesday, November 18, 2004, 12pm

November 9, 2004

1 Graphical Models (25%)

1. Consider a Bayesian network that describes relationship between genes of parents and children. The genes of every child are determined by the genes of its parents, with some noise. Pat and Sue are parents to Marge, Pat and Mary are parents to Fay, Mary and Bob are parents to Homer, Homer and Marge are parents to Bart, Lisa, and Maggie. Fay and Homer are parents to Sue, and Bart and Sue are parents to John.
   (a) Draw the Bayesian network that describes these relationships.
   (b) Draw a triangulated version of this network. Is it moralized?
   (c) In this graph, are are $Homer, Marge$, d-separated by $Bart$?
   (d) Are $Homer, Marge$, d-separated by $Bart, Lisa, Maggie$?
   (e) Are $Homer, Marge$, d-separated by $Pat, Bart, Lisa, Maggie$?
   (f) Are $Homer, Marge$ independent?
   (g) Are $Homer, Marge$ independent given $John$?
2. Prove that whenever X, Y are vertex-separated by Z, in an undirected graphical model, then X, Y are independent given Z.
3. Give an example of an undirected graphical model that encodes conditional independence assumptions that cannot be captured by a directed graphical model (Bayesian network) on the same variables.
4. Give an example of a directed graphical model that encodes conditional independence assumptions that cannot be captured by an undirected graphical model on the same variables.

2 Exact Probabilistic Inference (25%)

1. Consider the Bayesian network in Figure 1. All variables have four states, besides A that has three.
   (a) Calculate the size of the table $P(A, B, C, D = d_1)$.
   (b) In the calculation $P(A|E = e_1)$ the variables have been marginalized in the following order: B, C, D, F, G, H. Calculate the size of each table produced in the process, and sum the sizes up.
   (c) Find an elimination order that yields a smaller sum of table sizes than the one you achieved in 1b.
   (d) Construct a junction tree for this network.
2. BN has the potentials in Table **??**.
   (a) Calculate $P(A|D = y)$, $P(B|D = y)$, $P(C|D = y)$.

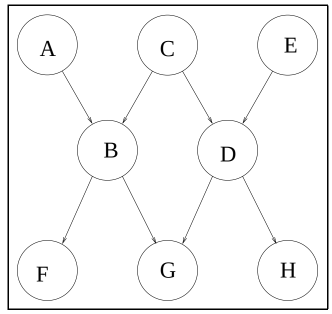

Figure 1: Bayesian network.

	A			A			B,C			
B	y	n	C	y	n	D	y,y	y,n	n,y	n,n
y	0.2	0.6	y	0.1	0.5	y	0.3	0.9	0.2	0.6
n	0.8	0.4	n	0.9	0.5	n	0.7	0.1	0.8	0.4

$P(B|A)$ $\quad$ $P(C|A)$ $\quad$ $P(D|B,C)$

Figure 2: BN Potentials

(b) Calculate $P(B|C = y)$.

3. One reason why variable elimination can be done in time that is exponential only in the treewidth of the graph, is the commutativity of the operations of summation and product when these do not refer to the same terms (e.g., $\sum_{A,B,C} f(A,B)f(B,C) = \sum_{A,B}(f(A,B)\sum_C f(B,C)))$, provided the factors are non-negative (which holds for potentials representing probability distributions). A similar commutativity holds for the *max* function, namely, $\max_{A,B,C} f(A,B)f(B,C) = \max_{A,B}(f(A,B)\max_C f(B,C)))$.

   Create an algorithm for finding the most likely configuration of a set of random variables, C, given some evidence, E, and marginalizing over the rest of the random variables, R. This is called MPE (Most Probable Explanation), and is defined formally as $argmax_C Pr(C,E,R)$.

   Prove that your algorithm is correct.

3 Sampling (25%)

1. We are given an image of 100x100 pixels (256 values each gray scale), and are looking for an object in that image. We know that the object is particularly dark (with some variation in darkness across it) in comparison with the background, but we do not know where it is or what that darkness means in pixel values. However, we do know that if a pixel belongs to the object then the adjacent pixel belongs to the object w.h.p., and that significant changes in intensity across pixels increase the probability of the darker pixel being part of an object. Close-by dark pixels also increase the probability of those pixels belonging to the object.

   Our task is to find a pixel that belongs to the object with the highest probability.

   (a) Represent this problem with a generative model Bayesian network (i.e., an object is at a position and size with some prior, and generates pixel values with some distribution as a result).

   (b) Download the Bayes Net Toolbox for Matlab (from Kevin Murphy: http://www.ai.mit.edu/ murphyk/Software/BNT/bnt.html) and use any sampling algorithm from those in the package (e.g., Gibbs, likelihood weighting) to detect an object in an image of your choice (can be something you find online, but can also be hand-made).

Hand-tune the values for the CPTs. Please provide both the image, the CPT values, the algorithm (that uses the toolbox as a subroutine), the output of the program, and the time taken.

4 Variational Approximation (25%)

Consider the multiple cause model discussed in class. This a with binary latent (hidden) variables, s_i, real-valued observed vector y and parameters $\phi = \{\{\mu_i, \pi_i\}_{i=1}^K, \sigma^2\}$.

$$p(s_1, ..., s_K|\pi) = \prod_{t=1}^{K} p(s_i) = \prod_{i=1}^{K} \pi_i^{s_i}(1-\pi_i)^{(1-s_i)}$$

$$p(y|s_1, ..., s_K, \mu\sigma^2) = \mathcal{N}(\sum_{t=1}^{K} s_i\mu_i, \sigma^2 I)$$

Assume you have a data set of N i.i.d. observations of y, i.e. $Y = \{y^{(1)}, ..., y^{(N)}\}$.

General Matlab hint: wherever possible, avoid looping over the data points. Many (but not all) of these functions can be written using matrix operations.

1. Implement the fully factored (a.k.a. mean-field) variational approximation described in class. That is, for each data point $y^{(n)}$, approximate the posterior distribution over the hidden variables by a distribution:

   $$q_n(s) = \prod_{i=1}^{K} \lambda_{in}^{s_i}(1-\lambda_{in})^{(1-s_i)}$$

   and find the λ's that maximize $\mathcal{F}$ holding ϕ fixed. Specifically, write a Matlab function:

   $$[lambda, F] = MeanField(Y, mu, sigma, pie, lambda0, maxsteps)$$

   where lambda is $N \times K$, F is the lower bound on the likelihood, Y is the $N \times D$ data matrix, mu is the $D \times K$ matrix of means, *pie* is the $1 \times K$ vector of priors on s, $lambda0$ are initial values for $lambda$ and maxsteps are maximum number of steps of the fixed point equations. You might also want to set a convergence criterion so that, if F changes by less than ϵ, the iterations halt. **Hand in:** code.

2. Apply your algorithm to derive an answer to the same problem as in question 3. **Hand in:** same as question 3.2.