CS 228, Winter 2008
Final

You have 24 hours to complete this exam. You must return the completed exam to Gates 120 (the Fishbowl) at either 12:00 pm or 6:00 pm the day after you receive the exam, depending on what time you chose to receive it.

This exam is long and difficult, and we do not expect everyone to finish all of the questions. Be sure to use good test taking skills and attack the easier problems first, spend more time on questions worth more points, and generally pay attention to how you spend your time. Also, you are welcome (and we expect you) to use or refer to algorithms from the reader when appropriate, without having to rederive or explain them. Furthermore, please use standard notation from the reader (when possible) and clearly define any terms you introduce. Algorithm answers should be provided in the form of pseudocode (with explanations where necessary), and your answers will be easier to grade (read: you will get higher grades) if you use proper spacing and layout of your answers on the page. We have provided approximate times and lengths for some of the problems to give you a rough estimate of how long we think it might take (in time and pages not including diagrams).

Short Questions

1. **[12 points] Clique Tree Calibration**

   Suppose that we have a clique tree over a set of factors $\mathcal{F}$ with cliques $C_1, ..., C_N$, which we have calibrated using sum-product message propagation so that we have all messages $\delta_{i \to j}$.

   (a) **[6 points]** If we modify a factor in some clique C_i, which message updates do we have to perform to recalibrate the tree?

   (b) **[6 points]** If we modify a factor in some clique C_i, but we just want the marginal over a single pre-specified variable X_k, which message updates do we have to perform?

2. **[7 points] Learning 2-TBNs**

   In this question we will analyze the problem of learning a 2-TBN model from data.

   Assume that our state is represented by a set of variables $X_1, \ldots, X_n$, and that our goal is to learn the 2-TBN structure. Assume also that we are considering only models where there are at most 3 parents per variable.

   Explain *briefly* why the problem of learning a 2-TBN structure is considerably easier (that is, in terms of the asymptotic running time) when we assume that there are no intra-time-slice edges in the 2-TBN.

   Estimated length: 2–3 sentences.

3. **[12 points] Multi-conditional Parameter Learning, Markov Networks** In this problem, we will consider the problem of learning parameters for a Markov network using a specific objective function. In particular assume that we have two sets of variables

$\boldsymbol{Y}$ and $\boldsymbol{X}$, and a dataset $\mathcal{D} = \{\langle \boldsymbol{x}[1], \boldsymbol{y}[1]\rangle, \ldots, \langle \boldsymbol{x}[m], \boldsymbol{y}[m]\rangle\}$ We will estimate the model parameters $\boldsymbol{\theta} = [\theta_1 \ldots \theta_n]$ by maximizing the following objective function:

$$f(\boldsymbol{\theta} : \mathcal{D}) = (1-\alpha)\ell_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{\theta} : \mathcal{D}) + \alpha\ell_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{\theta} : \mathcal{D})$$

where $\ell_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{\theta} : \mathcal{D})$ means the conditional log-likelihood of the dataset $\mathcal{D}$ using the distribution $P(\boldsymbol{X} \mid \boldsymbol{Y})$ defined by the Markov network with parameters $\boldsymbol{\theta}$ (similarly for $\ell_{\boldsymbol{Y}|\boldsymbol{X}}$). Thus, our objective is a mixture of two conditional log-likelihoods ($0 < \alpha < 1$). As usual, we consider a log-linear parameterization of a Markov network, using a set of n features $\phi_i[\boldsymbol{X}_i, \boldsymbol{Y}_i]$ where $\boldsymbol{X}_i$ and $\boldsymbol{Y}_i$ are some (possibly empty) subsets of the variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively.

(a) **[4 points]** Write down the full objective function $f(\boldsymbol{\theta} : \mathcal{D})$ in terms of the features ϕ_i and weights θ_i

(b) **[8 points]** Derive $\frac{\partial}{\partial \theta_i} f(\theta : \mathcal{D})$: the derivative of the objective with respect to a weight θ_i. Write your final answer in terms of feature expectations $\boldsymbol{E}_Q[\phi_i]$, where Q is either: the empirical distribution of our dataset $\hat{P}$; or a conditional distribution of the form $P_{\boldsymbol{\theta}}(\boldsymbol{W} \mid \boldsymbol{Z} = \boldsymbol{z})$ (for some sets of variables $\boldsymbol{W}, \boldsymbol{Z}$, and assignment $\boldsymbol{z}$.)

Estimated length: 1/2 page.

4. **[5 points] Learning Causal Models**

Intervention	$x^0y^0z^0$	$x^0y^0z^1$	$x^0y^1z^0$	$x^0y^1z^1$	$x^1y^0z^0$	$x^1y^0z^1$	$x^1y^1z^0$	$x^1y^1z^1$
None	4	2	1	0	3	2	1	4
$do(X = x^0)$	3	1	2	1	0	0	0	0
$do(Y = y^0)$	7	1	0	0	2	1	0	0
$do(Z = z^0)$	1	0	1	0	1	0	1	0

Table 1: Counts for Interventional Data

Calculate $M[x^0; y^0z^0]$, $M[y^0; x^0z^0]$, $M[x^0]$.

5. **[7 points] Value of Im-perfect Information**

In a decision problem, we know how to calculate the value of perfect information of X at decision D. Now imagine that we cannot observe the exact value of X, but we can instead observe a noisy estimate of X.

For this problem, assume X is binary. Also assume the noisy observation has a false positive rate of p and a false negative rate of q. (That is when $X = 0$ we observe 1 with probability p, and when $X = 1$ we observe 0 with probability q.)

Give a simple method by which we can calculate the improvement in MEU from observing this imperfect information. (Your answer should be just a couple lines long, but you should explain exactly how p and q are used.)

Long Questions

6. **[23 points] Context-specific d-separation**

   Consider a Bayesian network $\mathcal{B}$ parameterized by a set of tree-CPDs. Recall that in such cases, the network also exhibits context-specific independencies (CSI). In this exercise, we define a simple graph-based procedure for testing for these independencies; your task will be to prove that this procedure is sound.

   Consider a particular assignment of evidence $\boldsymbol{Z} = \boldsymbol{z}$. We define an edge $X \to Y$ to be *spurious* in the context $\boldsymbol{z}$ if, in the tree CPD for Y, all paths down that tree that are consistent with the context $\boldsymbol{z}$ do not involve X. (Example 4.3.6 in the notes provides two examples.) We define $\boldsymbol{X}$ and $\boldsymbol{Y}$ to be *CSI-separated* given $\boldsymbol{z}$ if they are d-separated in the graph where we remove all edges that are spurious in the context $\boldsymbol{z}$.

   You will now show that CSI-separation is a sound procedure for detecting independencies. That is: If P is the distribution defined by $\mathcal{B}$, and $\boldsymbol{X}$ and $\boldsymbol{Y}$ are CSI-separated given $\boldsymbol{z}$ in $\mathcal{B}$ (written $\textit{CSI-sep}_{\mathcal{B}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{z})$), then $P \models ((\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{z}))$.

   This proof will roughly follow the analysis in Sections 5.4.1.2-3 in the book, and specifically will mirror Theorem 5.4.9. However, the proof of Theorem 5.4.9 does not provide enough details, so we will provide you with an outline of a proof of the above statement, which you must complete.

   (Note that "CSI: Separation" would probably be a good name for a new television series.)

   (a) **[4 points]** Let $\boldsymbol{U} = \boldsymbol{X} \cup \boldsymbol{Y} \cup \boldsymbol{Z}$, let $G' = G^+[\boldsymbol{U}]$ be the induced Bayesian network over $\boldsymbol{U} \cup \textit{Ancestors}_{\boldsymbol{U}}$, and let $\mathcal{B}'$ be the Bayesian network defined over G' as follows: the CPD for any variable in $\mathcal{B}'$ is the same as in $\mathcal{B}$. You may assume without proof for the rest of this problem that $P_{\mathcal{B}'}(\boldsymbol{U}) = P_{\mathcal{B}}(\boldsymbol{U})$.

   Define a Markov network $\mathcal{H}$ over the variables $\boldsymbol{U} \cup \textit{Ancestors}_{\boldsymbol{U}}$ such that $P_{\mathcal{H}} = P_{\mathcal{B}'}$.

   (b) **[4 points]** Define *spurious edges* and therefore *CSI-separation* for Markov networks. (Hint: use Problem 3 from Problem Set 1. Your definition should be a natural one, but in order to receive credit, 6c and 6d must hold.)

   (c) **[12 points]** Show that if $\textit{CSI-sep}_{\mathcal{B}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{z})$ then $\textit{CSI-sep}_{\mathcal{H}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{z})$. (Hint: as one part of this step, use Proposition 5.4.7.)

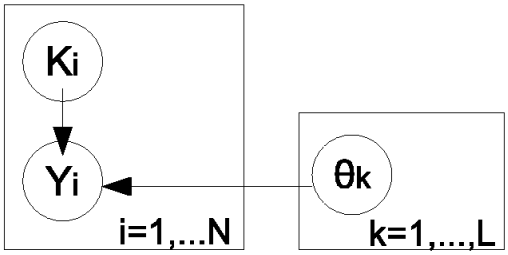
   (d) **[2 points]** Show that if $\textit{CSI-sep}_{\mathcal{H}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{z})$ then $P_{\mathcal{H}} \models (\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{z})$.

   (e) **[1 points]** Conclude that if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are CSI-separated given $\boldsymbol{z}$ in $\mathcal{B}$, then $P_{\mathcal{B}} \models (\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{z})$.

7. **[18 points] Plate Models**

   In this question we are going to cluster a set of variables using Gibbs sampling.

   Let $\boldsymbol{Y} = \{Y_1, Y_2, \ldots, Y_N\}$ be our observed set of variables. Each variable Y_i is an integer value between 1 and M. Our variables are generated from L different classes: For each class k there is a multinomial $\boldsymbol{\theta}_k$ that generates the variables of that class. We denote by $\boldsymbol{K} = \{K_1, K_2, \ldots, K_N\}$ the class assignments of the variables so that K_i is the class of variable Y_i. Hence, we can model our scenario as follows:



$$\begin{aligned} Y_i|K_i, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_L &\sim \textit{multinomial}(\boldsymbol{\theta}_{K_i}) \\ K_i &\sim \textit{multinomial}(\beta) \\ \boldsymbol{\theta}_k &\sim \textit{Dirichlet}(\alpha_1, \ldots, \alpha_M) \end{aligned}$$

(a) **[2 points]** Draw the full graph for $N = 3$ and $L = 2$.

(b) **[5 points]** Given the context $\boldsymbol{K} = \boldsymbol{k}$ find a minimal set $D \subseteq \boldsymbol{Y}_{-i}$ such that the following independency holds:

$$(Y_i \perp (Y_{-i} - D) \mid (\boldsymbol{K} = \boldsymbol{k}, D))$$

where $\boldsymbol{Y}_{-i} = \boldsymbol{Y} - \{Y_i\}$.

(c) **[11 points]** Now we are ready to do Gibbs sampling: $\boldsymbol{Y} = \boldsymbol{y}$ are fixed and always observed, and we want to draw samples from the posterior $P(\boldsymbol{K}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L | \boldsymbol{Y} = \boldsymbol{y})$. For each sample, only the class assignments $\boldsymbol{K}$ are sampled, and we integrate over the $\boldsymbol{\theta}_k$'s (this is effectively distributional Gibbs sampling).

We define the local transition model in the standard way:

$$T_i((\boldsymbol{k}_{-i}, k_i) \rightarrow (\boldsymbol{k}_{-i}, k_i')) = P(k_i' | \boldsymbol{y}, \boldsymbol{k}_{-i})$$

where $\boldsymbol{k}_{-i}$ are all the values of $\boldsymbol{k}$ except k_i.

Show how you can efficiently sample k_i' from $P(K_i | \boldsymbol{y}, \boldsymbol{k}_{-i})$.

Hints:

- Start with Bayes' rule: $P(K_i = k_i | \boldsymbol{y}, \boldsymbol{k}_{-i}) \propto P(y_i | \boldsymbol{k}, \boldsymbol{y}_{-i}) P(k_i | \boldsymbol{k}_{-i}, \boldsymbol{y}_{-i})$
- Use the independence properies of the graph to simplify the probability terms
- Use what you know about the Dirichlet conjugate prior to avoid computing integrals

8. **[16 points] Ideal Children and Parents**

Consider two procedures for learning the structure of a Bayes Net. We begin with a graph G with no edges and are given an ordering $X_1, X_2, ..., X_N$ of the variables. In the first

procedure, at iteration i, we connect X_i to its "ideal parents." That is, we consider all possible sets of parents for X_i that we can add to G_{i-1} that do not induce a cycle, and choose G_i to be the choice that yields the highest BIC score. In the second procedure, at iteration i, we similarly connect X_i to its "ideal children:" we add to G_i the set of children for X_i that does not induce a cycle and yields the highest BIC score.

(a) **[9 points]** Show how the "ideal children" procedure may be performed efficiently. That is, describe how we can implement this procedure so that finding the ideal children at iteration i is efficient, given that we have performed the necessary computations for iterations 1 through $i-1$. You should not store more information than necessary.

(Note: Though it may seem that there is no criterion for when your procedure is "efficient enough," if you have the correct answer then it should be clear that you are done. You should be able to provide a simple description that uses tools you have seen in the course, and don't need to worry about more detailed considerations such as optimizations related to specific data structures.)

Estimated length: 1/2 page

(b) **[7 points]** Explain why the "ideal parents" procedure cannot be done as efficiently as the "ideal children" procedure. Provide an analysis of the efficiency of your ideal children procedure, and explain why this efficiency cannot be achieved for the case of ideal parents.

Estimated length: One or two lines to analyze the efficiency, and a few brief sentences on why it cannot be achieved for the case of ideal parents.