Homework Assignment 4
Due: March 9, 2017 at 11am

Total points: 100
Deliverables: `hw4.pdf` containing typeset solutions to Problems 1-6.

Guidelines: All files must be submitted by Dropbox. You can brainstorm with others, but please solve the problems and write up the answers and code by yourself. You may use textbooks (Koller & Friedman, Russell & Norvig, etc.), lecture notes. Please do NOT use any other resources or references (e.g., example code, online problem solutions, etc.) without asking.

1 Sampling-based Inference

1. In class we have shown that a distribution in detailed balance is stationary and that the Gibbs sampling transition probability is in detailed balance with the posterior probability $P(\boldsymbol{X}|\boldsymbol{e})$. (Thus, showing that the posterior is a stationary distribution of Gibbs sampling).

   Remember that a distribution $\pi(\boldsymbol{X})$ is a stationary distribution for a Markov chain defined by a transition probability $\mathcal{T}$ if it satisfies:

   $$\pi(\boldsymbol{X} = \boldsymbol{x'}) = \sum_{\boldsymbol{x} \in Val(\boldsymbol{X})} \pi(\boldsymbol{X} = \boldsymbol{x}) \mathcal{T}(\boldsymbol{x} \rightarrow \boldsymbol{x'})$$

   Show directly from the definition of a stationary distribution (without using the detailed balance equation) that the posterior distribution $P(\boldsymbol{X}|\boldsymbol{e})$ is a stationary distribution of Gibbs sampling.

2. The *Metropolis-Hastings* algorithm is a member of the MCMC family; as such, it is designed to generate samples $\boldsymbol{x}$ (eventually) according to target probabilities $\pi(\boldsymbol{X})$. (Typically we are interested in sampling from $\pi(\boldsymbol{x}) = P(\boldsymbol{x}|\boldsymbol{e})$).

   Metropolis-Hastings operates in two stages. First, it samples a new state $\boldsymbol{x'}$ from a *proposal distribution* $q(\boldsymbol{x} \rightarrow \boldsymbol{x'})$, given the current state $\boldsymbol{x}$. Then, it probabilistically accepts or rejects $\boldsymbol{x'}$ according to the *acceptance probability*

   $$\alpha(\boldsymbol{x} \rightarrow \boldsymbol{x'}) = min\left(1, \frac{\pi(\boldsymbol{x'})q(\boldsymbol{x'} \rightarrow \boldsymbol{x})}{\pi(\boldsymbol{x})q(\boldsymbol{x} \rightarrow \boldsymbol{x'})}\right)$$

   If the proposal is rejected, the state remains at $\boldsymbol{x}$.

   (a) Consider the case where $\pi(\boldsymbol{x}) = P(\boldsymbol{x}|\boldsymbol{e})$. Explain why the fact that we can't sample directly from $P(\boldsymbol{x}|\boldsymbol{e})$ (this is what we are trying to estimate using the MCMC in the first place) does not pose a problem when computing $\alpha(\boldsymbol{x} \rightarrow \boldsymbol{x'})$.

   (b) Consider an ordinary Gibbs sampling step for a specific variable X_i. Show that this step, considered as a proposal, is guaranteed to be accepted by Metropolis-Hastings. (Hence, Gibbs sampling is a special case of Metropolis-Hastings).

   (c) Show that the two-step process above, viewed as a transition probability is in detailed balance with π.

2 Partially Observed Data

3. Consider learning the following Bayesian network: $A \rightarrow B \rightarrow C$ with the following data table, where entries '?1' and '?2' are missing at random:

A	B	C
T	T	F
T	?2	F
T	F	F
?1	F	T
F	T	T
F	T	F
F	F	F

(a) Use the data to estimate initial parameters for this network, using maximum likelihood estimation over the observed data for simplicity.

(b) Perform two iterations of the EM algorithm (by hand): estimate the values of the missing data, re-estimate the parameters, re-estimate the values of the missing data and re-estimate the parameters once more. Show your calculations.

4. Consider the problem of applying EM to parameter estimation for a variable X whose local probabilistic model is a noisy-or. Assume that X has parents $Y_1, \ldots, Y_k$, so that our task for X is to estimate the noise parameters $\lambda_0, \ldots, \lambda_k$. Explain how we can use the EM algorithm to accomplish this task by utilizing the structural decomposition of the noisy-or node. That is, construct an equivalent network with hidden variables and derive the expectation and maximization steps.

3 Structure Learning

5. Consider learning the structure of a Bayesian network for some given ordering, $\prec$, of the variables $X_1, \ldots, X_n$. (This can be done efficiently as described in section 18.5.2.1 of Koller & Friedman). Now assume that we want to perform a search over the space of orderings; that is, we are searching for a network (with bounded in-degree k) that has the highest score. We do this by defining the score of an ordering as the score of the (bounded in-degree) network with the maximum score consistent with that ordering, and the we search for the ordering with the highest score. We bound the in-degree so that we have smaller and smoother search space.

   We will define our search operators o to be swapping two adjacent variables in the ordering. Starting from some given ordering, $\prec$, we evaluate a decomposable structure score of all successor orderings, $\prec'$, where a successor ordering is found by applying o to $\prec$ once (see figure 1).

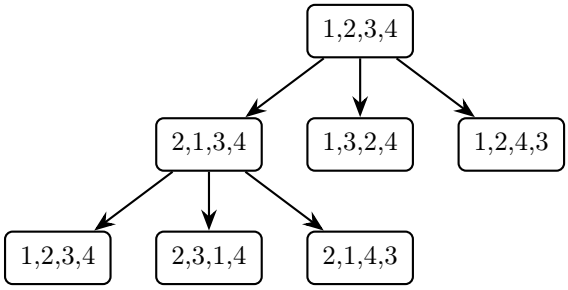

Figure 1: Partial search tree example for orderings over variables X_1, X_2, X_3, X_4. Successors to $\prec = (1, 2, 3, 4)$ and $\prec' = (2, 1, 3, 4)$ shown.

   We now choose a particular successor, $\prec'$. Provide an algorithm for computing *as efficiently as possible* the scores for all successors of the new ordering $\prec'$, given that we have already computed the scores for all successors of $\prec$.

Note: A structure score $score(\mathcal{G} : \mathcal{D})$ is *decomposable* if the score of the structure $\mathcal{G}$ can be written as

$$score(\mathcal{G} : \mathcal{D}) = \sum_i FamScore(X_i | Pa^{\mathcal{G}}_{X_i} : \mathcal{D})$$

where the *family score* $FamScore(X|\boldsymbol{U} : \mathcal{D})$ is a score measuring how well a set of variables $\boldsymbol{U}$ serves as parents of X in the dataset $\mathcal{D}$.

6. Show that adding edges to a Bayesian network increases the likelihood.