Problem 1

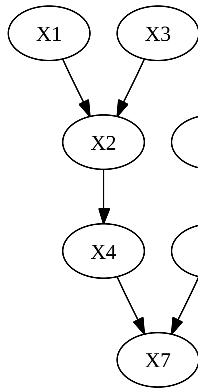

Figure 1: A probabilistic model (Problem 1a).

(a) Illustrate your use of the Bayes ball algorithm to determine whether the following conditional independencies hold in the probabilistic model depicted in Figure 1.

i.) $(X3 \perp\!\!\!\perp X7)$

ii.) $(X3 \perp\!\!\!\perp X7) \,|\, X4$

iii.) $(X3 \perp\!\!\!\perp X5)$

iv.) $(X3 \perp\!\!\!\perp X5) \,|\, X7$

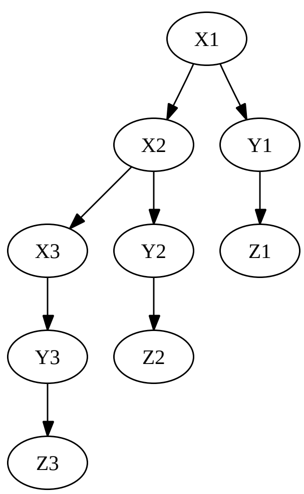

Figure 2: Another probabilistic model (Problem 1b).

(b) Suppose we are interested in inferring $P(X1, X2, X3 \,|\, Y1, Y2, Y3)$ in Figure 2. Use the Bayes Ball algorithm and properties of conditional independence to formally determine whether the Z nodes affect the inference. If we knew *a priori* that the Z nodes are never going to be observed, would it make sense to replace the above probabilistic model with a simpler one? How does this idea generalize to arbitrary probabilistic models?

(c) Let G be a probabilistic model. Let X_n be a node in G. Use the Bayes Ball algorithm to prove that $P(X_n|X_{-n}) = P(X_n|\mathrm{MB}(X_n))$ where X_{-n} denotes all nodes of G except X_n and $\mathrm{MB}(X_n)$ denotes the Markov blanket of X_n.

Why is this equality significant for Gibbs sampling?

Problem 2

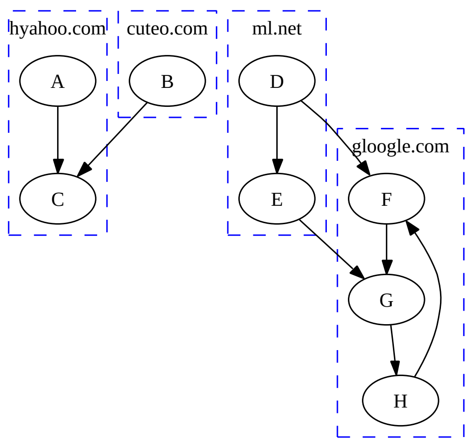

Figure 3: Query link structure (Problem 2).

Once upon a time, two guys came up with an algorithm for ranking the results of a web search query. The figure above shows a set of webpages related to a particular query. Each node represents a webpage and every edge represents a link. As mentioned in class, the idea behind this algorithm is that webpages are ranked based on the stationary distribution of the Markov chain defined by a random websurfer walking on the graph and randomly jumping to a new page.

(a) Suppose that the probability of of transitioning from a webpage X to another Y is:

$$q(X \to Y) = \begin{cases} \frac{p}{|X|_{\text{out}}} + \epsilon & \text{if there is a link between } X \text{ and } Y \\ \epsilon & \text{otherwise} \end{cases} \tag{1}$$

where $|X|_{\text{out}}$ denotes the out-degree of X and p and ϵ are constants and p is between 0 and 1.

Determine ϵ as a function of p and the number of webpages, n, keeping in mind that every column of the transition matrix must sum to 1.

(b) Assuming that $p = 0.9$, determine the transition probabilities from webpage D using what you derived in (a).

(c) Now's your chance to be a billionaire. The smart people at gloogle.com have noticed that each person seems to have a set of website preferences π_X. These preferences satisfy

$$\sum_X \pi_X = 1 \tag{2}$$

You've come up with the following equation for transition probabilities incorporating preferences.

$$q(X \to Y) = \begin{cases} \frac{p}{|X|_{\text{out}}} + \epsilon\pi_Y & \text{if there is a link between } X \text{ and } Y \\ \epsilon\pi_Y & \text{otherwise} \end{cases} \tag{3}$$

Now what is the expression for ϵ?

(d) Now recompute the transition probabilities from webpage D using part (c). The preference for each page will be assigned according to the domain in which it belongs:

Domain	Ranking
hyahoo.com	.05
cuteo.com	.1
ml.net	.1
gloogle.com	.2

(e) (Extra Credit) Find the stationary distribution of the Markov chain defined by the transition matrix, $\mathbf{Q}$. The stationary distribution can be determined by simply computing $\mathbf{Q}^\infty$ (or a reasonable approximation thereof).

Problem 3

Consider the noisy observation model pictured in Figure 4. Recall that the forward messages are

$$f_{1:t} := p(x_t \mid e_{1:t}), \tag{4}$$

and backward messages are

$$b_{k+1:t} := p(e_{k+1:t} \mid x_k). \tag{5}$$

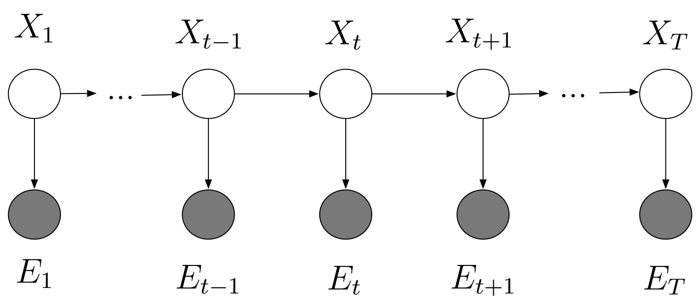

Figure 4: The noisy observation model (Problem 3).

The smoothing problem is to compute

$$g_k := p(x_k \mid e_{1:T}). \tag{6}$$

In class, we described how to solve the filtering problem by using the forward messages to k and the backward messages from k. In this problem, we are interested in solving the smoothing problem g_k for all k from 1 to T.

(a) Derive a recursion for computing g_k in terms of g_{k+1} and the forward messages. (Hint: include and marginalize x_{k+1}.)

(b) What is the time and space complexity of this algorithm for computing g_k for all k.

(c) What are the advantages, if any, of this algorithm versus naïvely running the forward/backward algorithm for each g_k?