<u>CS 6601:</u> *Probabilistic Modeling and Inference*

Daniel Kohlsdorf, Titilayo Craig, Thad Starner

This homework consists of three assignments from your textbook. For the warmup you will model a simple diagnostic network for a power plant (Assignment 14.11). As a practice question we will look into another MCMC algorithm called metropolis hastings (Assignment 14.20). During the implementation you will model soccer game outcomes (Assignment 14.21) and implement three inference algorithms on your net. I also attached the sampling chapter from the Bishop book.

Warmup: Modeling (20 %)

14.11 In your local nuclear power station, there is an alarm that senses when a temperature gauge exceeds a given threshold. The gauge measures the temperature of the core. Consider the Boolean variables A (alarm sounds), F_A (alarm is faulty), and F_G (gauge is faulty) and the multivalued nodes G (gauge reading) and T (actual core temperature).

a. Draw a Bayesian network for this domain, given that the gauge is more likely to fail when the core temperature gets too high.

b. Is your network a polytree? Why or why not?

c. Suppose there are just two possible actual and measured temperatures, normal and high; the probability that the gauge gives the correct temperature is x when it is working, but y when it is faulty. Give the conditional probability table associated with G.

d. Suppose the alarm works correctly unless it is faulty, in which case it never sounds. Give the conditional probability table associated with A.

e. Suppose the alarm and gauge are working and the alarm sounds. Calculate an expression for the probability that the temperature of the core is too high, in terms of the various conditional probabilities in the network.

Practice: Inference (20 %)

14.20 The **Metropolis–Hastings** algorithm is a member of the MCMC family; as such, it is designed to generate samples $\mathbf{x}$ (eventually) according to target probabilities $\pi(\mathbf{x})$. (Typically we are interested in sampling from $\pi(\mathbf{x}) = P(\mathbf{x} \mid \mathbf{e})$.) Like simulated annealing, Metropolis–Hastings operates in two stages. First, it samples a new state $\mathbf{x}'$ from a **proposal distribution** $q(\mathbf{x}' \mid \mathbf{x})$, given the current state $\mathbf{x}$. Then, it probabilistically accepts or rejects $\mathbf{x}'$ according to the **acceptance probability**

$$\alpha(\mathbf{x}' \mid \mathbf{x}) = \min\left(1, \frac{\pi(\mathbf{x}')q(\mathbf{x} \mid \mathbf{x}')}{\pi(\mathbf{x})q(\mathbf{x}' \mid \mathbf{x})}\right) .$$

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

a. 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.)

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

Implementation (40%)

14.21 Three soccer teams A, B, and C, play each other once. Each match is between two teams, and can be won, drawn, or lost. Each team has a fixed, unknown degree of quality—an integer ranging from 0 to 3—and the outcome of a match depends probabilistically on the difference in quality between the two teams.

a. Construct a relational probability model to describe this domain, and suggest numerical values for all the necessary probability distributions.

b. Construct the equivalent Bayesian network for the three matches.

c. Suppose that in the first two matches A beats B and draws with C. Using an exact inference algorithm of your choice, compute the posterior distribution for the outcome of the third match.

d. Suppose there are n teams in the league and we have the results for all but the last match. How does the complexity of predicting the last game vary with n?

e. Investigate the application of MCMC to this problem. How quickly does it converge in practice and how well does it scale?

Answer a and d as above.

In b, c) use the Matlab library Bayesian Net Tookit:
https://code.google.com/p/bnt/
and implement your model in it. Then run extant inference in order to get the posterior for the third game. Once you defined the model this should be easy. If you do not have Matlab, use Gatech computers or try using octave instead of Matlab.

In e) use your probability tables and implement a Gibbs sampler in a language of your choice. Compute the posterior of the third match after 10, 1000, 10000, 100000 iterations and compare the results to the exact posterior. Now implement the metropolis hastings algorithm and compute the posterior of the third match after 10, 1000, 10000, 100000 iterations again. Compare the results to the exact posterior again.

Write one paragraph about which of the approximate algorithm performs best and argue why that might be.

Late

From the attached reading, read and understand the slice sampling method, too.
Write a paragraph about the advantages of slice sampling compared to Gibbs sampling.
Furthermore, write a paragraph about how to apply slice sampling to Bayesian networks and give pseudo code.