Probabilistic Programming Homework 1

Submit your solutions to the TA in his office (room 3425 in the E3-1 building) by 2:00pm on 6 November 2017 (Monday). If you type up your solutions, you can email them to him (bskim90@kaist.ac.kr).

Question 1

This is a question about writing and reasoning about models in Anglican. The first two sub-questions come from McElreath's book "Statistical Rethinking" and the last from MacKay's book "Information Theory, Inference, and Learning Algorithms".

(a) Suppose that there are two species of panda bear. Both are equally common in the wild and live in the same place. They look exactly alike and eat the same food, and there is yet no genetic assay capable of telling them apart. They differ, however, in their family sizes. Species A gives birth to twins 10% of the time, otherwise birthing a single infant. Species B births twins 20% of the time, otherwise birthing a single infant. Assume that these numbers are known with certainty, from many years of field research.

Now suppose you are managing a captive panda breeding program. You have a new female panda of unknown species, and she has just given birth to twins. We would like to find out the probability that her next birth will also be twins. Write an Anglican program that expresses this situation. Then, compute the probability by performing inference on your program.

(b) Continuing on, suppose that the same panda mother has a second birth and that it is not twins, but a single infant. By modifying your Anglican program and performing inference on it, compute the posterior probability that this panda is species A.

(c) Suppose that seven scientists all go and perform the same experiment, each collecting a measurement $x_i \in \mathbb{R}$ for $i = 1, \ldots, 7$ for some unknown quantity $\mu \in \mathbb{R}$. These scientists are varyingly good at their job, and while we can assume each scientist would estimate μ correctly *on average*, some of them may have much larger error in their measurements than others. They come back with the following seven observations:

```
(def measurements [-27.020  3.570  8.191  9.898  9.603  9.945  10.056])
```

Note that scientist 1 does not know what he is doing, and that scientists 2 and 3 are probably not very good, either.

We can model this situation by assuming that the i-th scientist makes a noisy observation on μ with the noise level σ_i for $i = 1, \ldots, 7$. That is, the random variable x_i is distributed according to the Normal distribution with mean μ and standard deviation σ_i:

$$x_i \sim \text{Normal}(\mu, \sigma_i) \text{ for } i = 1, \ldots, 7.$$

These mean and standard deviation parameters are also random variables. We place uninformative prior distributions on these parameters:

$$\mu \sim \text{Normal}(0, 50), \qquad \sigma_i \sim \text{Uniform}(0, 25) \text{ for } i = 1, \ldots, 7.$$

Write an Anglican program that describes this model and the measurement. Then, answer the following questions by performing posterior inference on the program. (1) What is the posterior distribution of μ? (2) What distribution over noise level σ_i do we infer for each of these scientists' estimates?

Question 2

Let X be a (very large) finite set and f a function from X to $\mathbb{R}_+$, the set of non-negative real numbers. Assume that we are given an unnormalised probability r on X, and that we would like to estimate the expectation of f under the distribution r:[1]

$$\mathbb{E}_{r(x)/Z}[f(x)] \qquad \text{where } Z = \textstyle\sum_{x \in X} r(x). \tag{1}$$

In the lecture, we learnt how to do this estimation using the Metropolis-Hastings algorithm (in short MH algorithm). First, we set the (unnormalised) target probability of the algorithm to r, and pick some proposal distribution q. Next, we instantiate the MH algorithm for r and q, generate samples $x_1, \ldots, x_N$ using the algorithm, and compute

$$\sum_{i=1}^{N} \frac{f(x_i)}{N}. \tag{2}$$

This sum provides an estimate of the expectation in (1).

This question is about the MH algorithm. Throughout this question, we use $p(x) = r(x)/Z$, the (normalised) target distribution.

(a) The HM algorithm is correct largely because the target distribution $p(x)$ becomes a stationary distribution (also called invariant distribution) with respect to the random update of the loop body of the algorithm. That is, when we write $k(x'|x)$ for the probability of the body generating a next state x' from a given current state x, the stationarity of $p(x)$ means that

$$\left(\sum_x k(x'|x) \cdot p(x) \right) = p(x') \text{ for all } x' \in X. \tag{3}$$

Prove that this property indeed holds. Your proof should consist of the following two steps. First, show that if

$$(k(x'|x) \cdot p(x)) = (k(x|x') \cdot p(x')) \text{ for all } x, x' \in X, \tag{4}$$

then the property (3) holds. Second, prove the condition (4). The proof of this step will become easier if you consider two cases $x = x'$ and $x \neq x'$ separately. This two-step proof is more common than the one that we discussed in the lecture. Also, the condition (4) features in several other contexts and is called *detailed balance*.

(b) Consider the case that $X = A \times B = \{(a, b) \mid a \in A, b \in B\}$, the product of two finite sets A and B. Write a state x in terms of two random variables a and b, so that $x = (a, b)$. Let $q(a', b'|a, b)$ be the following conditional distribution on $A \times B$:

$$q(a', b'|a, b) = p(a'|b) \cdot p(b'|a').$$

[1] The expectation $\mathbb{E}_{r(x)/Z}[f(x)]$ is defined by $\mathbb{E}_{r(x)/Z}[f(x)] = \sum_{x \in X}((r(x)/Z) \cdot f(x))$.

Here $p(a'|b)$ and $p(b'|a')$ are conditional distributions derived from our target distribution $p(a,b) = r(a,b)/Z$ in the standard way:

$$p(a'|b) = \frac{p(a',b)}{\sum_{a''\in A} p(a'',b)}, \quad p(b'|a') = \frac{p(a',b')}{\sum_{b''\in B} p(a',b'')}.$$

Prove that if we use q as a proposal in the MH algorithm, the acceptance ratio

$$\alpha((a,b),(a',b')) = \min\left\{1,\ \frac{r(a',b')\cdot q(a,b|a',b')}{r(a,b)\cdot q(a',b'|a,b)}\right\}$$

is always 1. This means that all proposed samples get accepted. This instantiation of the MH algorithm is called *Gibbs sampling*.

(c) The Gibbs sampling in Part (b) sometimes fails to produce a correct answer. Find such a case. That is, find X, r and f such that the estimate in (2) with samples $x_1, \ldots, x_N$ from the Gibbs sampler never converges to the target expectation in (1).

Question 3

This is a question about the importance-sampling algorithm.

Consider the same setting as Question 2, and assume the same goal: we would like to estimate the expectation in (1). The importance-sampling algorithm (in short IS algorithm) is another method for computing this estimate. If we fix a proposal distribution q on X, the IS algorithm generates samples $x_1, \ldots, x_N$ from q together with their weights

$$w_1 = \frac{r(x_1)}{q(x_1)},\ w_2 = \frac{r(x_2)}{q(x_2)},\ \ldots,\ w_N = \frac{r(x_N)}{q(x_N)},$$

and estimates the expectation by the following weighted sum:

$$\sum_{i=1}^{N} \frac{w_i}{\sum_{j=1}^{N} w_j} f(x_i). \tag{5}$$

In this question, we look at a few properties of this algorithm. As before, we write $p(x) = \frac{r(x)}{Z}$ where $r(x)$ is an unnormalised target density and Z is its normalising constant $\sum_x r(x)$.

(a) One important condition on the proposal q is that

$$r(x) > 0 \implies q(x) > 0 \text{ for all } x \in X.$$

Find an example which shows that the IS algorithm may fail if this condition is violated. More concretely, find X, r, q, f such that no matter how much we increase N in (5) (the number of samples), the estimate of the IS algorithm

$$\sum_{i=1}^{N} \frac{w_i}{\sum_{j=1}^{N} w_j} f(x_i)$$

never converges to the expectation $\mathbb{E}_{p(x)}[f(x)]$.

(b) When r is normalised (i.e. $Z = \sum_x r(x) = 1$), we usually use a variant of the IS algorithm that computes the following sum:

$$\sum_{i=1}^{N} \frac{w_i}{N} f(x_i). \tag{6}$$

Show that

$$\mathbb{E}_{q(x_1),\ldots,q(x_N)} \left[\sum_{i=1}^{N} \frac{w_i}{N} f(x_i) \right] = \mathbb{E}_{p(x)}[f(x)].$$

This equation means that on average, the estimate in (6) equals our target expectation. When an estimate satisfies this property, it is called *unbiased*.

(c) We continue the discussion about the variant of the IS algorithm in Part (b). Note that the variant is parameterised by a proposal distribution q. One way to compare two proposals q and q' is to compare their variances, which are defined as follows:

$$\mathbb{E}_{q(x_1),\ldots,q(x_N)} \left[\left(\left(\sum_{i=1}^{N} \frac{w_i}{N} f(x_i) \right) - F \right)^2 \right], \qquad \mathbb{E}_{q'(x_1),\ldots,q'(x_N)} \left[\left(\left(\sum_{i=1}^{N} \frac{w_i}{N} f(x_i) \right) - F \right)^2 \right]$$

where $F = \mathbb{E}_{p(x)}[f(x)]$. Intuitively, the above expectations compute average errors of this (simplified) IS algorithm with q and q', respectively. Define a proposal q_{opt} that has the minimum variance. This proposal is called *optimal proposal*.