Probabilistic Programming Homework 2

Submit your solutions to the TA by putting them in the homework submission box in the third floor of the E3-1 building by 2:00pm on 23 April 2018 (Monday). If you type up your solutions, you can email them to him (kwonsoo.chae@gmail.com).

Question 1

This is a question about writing and reasoning about models in Anglican. The first two subquestions 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:

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

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

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

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

This example is borrowed from the Chapter 1 of the book "Probabilistic Programming and Bayesian Methods for Hackers".

Assume that we are given a series of daily text-message counts from a user of our system over 74 days, and that we want to find out whether the user's text-messaging habit has changed suddenly at some point. Our goal is to answer this question by formulating an appropriate probabilistic model in Anglican, conditioning it with the given series of text-message counts, and performing posterior inference.

Our model consists of three kinds of random variables. The first is a random variable $\tau \in \{1, \ldots, 75\}$ that indicates the day that the user has changed her text-messaging habit. Our model assumes a priori that all days are equally possible:

$$\tau \sim \text{UniformDiscrete}(1, 76).$$

The second are random variables λ_1 and λ_2 having values in $[0, \infty)$. They denote the average counts of text messages before and after the habit change. The model assumes that λ_1 and λ_2 are sampled from the Exponential distribution with the rate 0.05[1]:

$$\lambda_1 \sim \text{Exponential}(0.05), \qquad \lambda_2 \sim \text{Exponential}(0.05).$$

The third are random variables corresponding to the message counts. We have 74 such random variables $y_1, \ldots, y_{74}$, and each y_i is sampled from the Poisson distribution with mean λ_1 or λ_2 depending on whether $i < \tau$ holds or not:

$$\text{for } i \in \{1, \ldots, 74\}, \qquad y_i \sim \begin{cases} \text{Poisson}(\lambda_1) & \text{if } i < \tau \\ \text{Poisson}(\lambda_2) & \text{if } i \geq \tau \end{cases}$$

The random variables $y_1, \ldots, y_{74}$ are observed. Their values are the given text-message counts.

In this question, you are required to do two tasks. First, write an Anglican query that expresses the model just described. I strongly advise you to use a Gorilla worksheet for this first task. Second, estimate the posterior means of τ, λ_1 and λ_2. For this second task, I used the `:lmh` algorithm of Anglican to generate posterior samples of the random variables τ, λ_1 and λ_2. Then, I computed the averages of these samples for each of these variables.

You can download the text-message dataset for this question from the course webpage:

`https://github.com/hongseok-yang/probprog18/tree/master/Homework/Homework2/txtdata.csv`

Create the `anglican-user/data` directory in your machine, and store the downloaded csv file in that directory. Change the begining of your Gorilla worksheet as follows:

[1] The magic number 0.05 comes from averaging all 74 message counts in our dataset. The approach of using such estimates to set hyperparameters of a model is called empirical Bayes.

```
(use 'nstools.ns)
(ns+ message-example
  (:like anglican-user.worksheet)
  (:require [clojure-csv.core :as csv]
            [clojure.java.io :as io]))

(defn take-csv
  [fname]
  (with-open [file (io/reader fname)]
    (csv/parse-csv (slurp file))))

(def data
  (map (comp read-string first) (take-csv "data/txtdata.csv")))
```

Running this Clojure code lets the variable `data` store a list of 74 numbers.

Question 3

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:[2]

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

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

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

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

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 $q_1(a', b'|a, b)$ and $q_2(a', b'|a, b)$ be the following conditional distribution ons $A \times B$:

$$q_1(a', b'|a, b) = p(a'|b) \cdot [b = b'], \qquad q_2(a', b'|a, b) = p(b'|a) \cdot [a = a'].$$

Here $[\phi]$ is an indicator function that has value 1 if ϕ is true and 0 otherwise. Also, $p(a'|b)$ 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 for all $i \in \{1, 2\}$, if we use q_i as a proposal in the MH algorithm, the acceptance ratio

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

is always 1. This means that all proposed samples get accepted for each $i \in \{1, 2\}$. A variant of the MH algorithm that proposes a candidate using q_1 and q_2 alternatively is an example of *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).