Probabilistic Programming Homework 1

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 30 March 2018 (Friday). If you type up your solutions, you can email them to him (kwonsoo.chae@gmail.com).

Question 1

Let F_n be the n-th Fibonacci number, and L_n the leading digit of F_n. For instance, the 7-th Fibonacci number is $F_7 = 13$, and its leading digit is $L_7 = 1$. For $1 \leq i \leq 9$ and $m \geq 1$, define $C_{m,i}$ to be the count of Fibonacci numbers from 1 up to and including m whose leading digit is i. Formally,

$$C_{m,i} = |\{n \mid 1 \leq n \leq m \wedge L_n = i\}|.$$

Write a Clojure program that takes $m \geq 2$ and returns the list

$$\frac{C_{m,1}}{m}, \quad \frac{C_{m,2}}{m}, \quad \ldots, \quad \frac{C_{m,9}}{m}.$$

Benford's law says that in many real-world datasets, the leading digit d occurs with probability close to $p(d) = log_{10}((d+1)/d)$. Is it also the case for Fibonacci numbers?

When you answer this question, you can use any programs that we discussed in the lectures, such as the following function for convering a number to a list of its digits:

```
(defn get-digits [n]
  (map (fn [c] (- (int c) (int \0))) (str n)))
```

Question 2

This question is about Bayes' rule. It is taken from MacKay's book "Information Theory, Inference and Learning Algorithms."

The inhabitants of an island tell the truth one third of the time. They lie with probability 2/3. On an occasion, after one of them made a statement, you ask another 'was that statement true?' and he says 'yes'. What is the probability that the statement was indeed true?

Question 3

This is a question about the importance-sampling algorithm.

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]

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

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

The importance-sampling algorithm (in short IS algorithm) is a 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:

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

In this question, we look at a few properties of this algorithm. 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 the estimate of the IS algorithm

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

does not converges to the expectation $\mathbb{E}_{p(x)}[f(x)]$ when we increase N to ∞ in (2) (the number of samples).

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

$$\sum_{i=1}^{N} \frac{w_i}{N} f(x_i). \tag{3}$$

Show that

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

The subscript $q(x_1), \ldots, q(x_N)$ expresses that the sequence $x_1, \ldots, x_N$ is generated with the probability $q(x_1) \times \ldots \times q(x_N)$, i.e., the x_i are independent. The above equation means that on average, the estimate in (3) 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. Assume that $f(x) \geq 0$ for all $x \in X$, and that f is not the constant-zero function. Define a proposal q_{opt} that has the minimum variance. This proposal is called *optimal proposal*.