Stats 598z: Homework 7

Due before midnight Monday, Apr 30

Important:

R code, tables and figures should be part of a single .pdf or .html files from `R Markdown` and `knitr`. See the class reading lists for a short tutorial.
Include R commands for all output unless explicitly told not to.
If you collaborated with anyone else, mention their names and the nature of the collaboration

1 The law of large numbers [20pts]

This assignment uses `ggvis` with reactive programming, see the last slide of lecture 20. Create a dataframe with three columns: `(indx,value,running_mean)`. Initially all are initialized to 0. Write code to

(a) Every 200ms, add a new row to the data-frame, with `indx` increasing by 1, `value` assigned a random value from the normal distribution, and `running_mean` giving the mean of all `value`s so far. Include a `ggvis` interface, so that you display a video of the evolution of the `running_mean` with time. [15pts]

(b) Change the `ggvis` part of the code, so it rather that updating the trajectory of `running_mean`, it plots the histogram of `value` every 200ms. [10pts]

For both parts, look at the documentation of ggvis to keep the x- and y- limits clamped over some suitable range (hint: see http://stackoverflow.com/questions/24491783/ggvis-density-plot-xlim-xlab)

2 Monte Carlo sampling [25pts]

- Consider two points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$. The coordinates of p_1 are distributed as Gaussian with mean 0, and of p_2, as Gaussian with mean 1. You want to calculate the average length of the line-segment connecting two such points. Get a Monte Carlo estimate of this using 5000 samples. Recall the procedure: sample 5000 instances of p_1 and p_2 from their distributions, calculate the length of the line joining them for each instance and then calculate the average. [15pts]
- Repeat the above procedure 1000 times, getting a random estimate each time. Plot a histogram of these values using `ggvis`. Include a tooltip that give the value in bin of the histogram you're pointing at, which is highlighted in red (see the slides from Lecture 20). [10pts]

3 Importance sampling [50pts]

(a) What is the mean and standard deviation of the sum of 100 fair dice? [5]

(b) Write a few lines of `R` to simulate the output of 100 fair die a thousand times. Plot the histogram of the sum, and show that the mean and standard deviation match the previous question. [5]

(c) Use the `pnorm` function to calculate the log-probability a Gaussian with this mean and standard deviation exceeds 450. [5]

(d) What fraction of your outcomes had a sum exceeding 450? This is your Monte Carlo estimate. [5]

(e) Now, simulate 100 biased dice, with each die having probability proportional to i of showing side i. Do this 1000 times. Plot the histogram of the sum of these values. How many times does the sum exceed 450? [5]

(f) Calculate the log-probability (under the biased dice) of each of the 1000 outcomes. Do not use for-loops. (note that the log-probability of a 100-dice outcome is the sum of the log-probabilities of the outputs of each of the 100 dice that constitute it). [10]

(g) What is the log-probability of any outcome under the fair dice? [5]

(h) Given the two 1000-vectors of log-probabilities of the 1000 outputs under the biased and fair dice, obtain an importance sampling estimate of the log-probability that the sum of 100 dice exceeds 450 [10]