Stats 545: Homework 6

Due before midnight on Sunday, Nov 25.
All plots should have labelled axes and titles.

Important:Rcode, 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.Any derivations can also be in Markdown, in Latex or neatly written on paper which you can give to me.

1 Problem 1: Generating gamma random variables via rejection sampling [60]

We will first generate an exponential random variable.

1. Let U be a Uniform$(0, 1)$ random variable. How will you transform this to generate an exponential with rate λ? Use the inverse-cdf method. [5 pts]

2. Let Y be a random variable on $[0, 1)$, with density $p(Y = y) \propto 1/y^a$, where $a \in (0, 1)$. This is the truncated Pareto distribution. What is it's normalization constant? Describe the inverse-cdf method to generate Y by transforming U. [10 pts]

3. Write down the density of a Gamma random variable with shape parameter α and rate parameter β. Call this $p_{\text{Gamma}}(x|\alpha, \beta)$. Note that calculating its cdf is not easy. [5pts]

4. Note that the sum of two exponentials with rate 1 is distributed as Gamma$(2, 1)$. How would you use previous results to sample from a Gamma distribution with parameters (`alpha_int`$, 1)$, where `alpha_int` is an integer. [10pts]

Next, we will generate a Gamma random variable, with `alpha` < 1

5. For $\alpha < 1$, $p_{\text{Gamma}}(x|\alpha, 1) \leq C_1 x^{\alpha-1}$ for $x < 1$, and $p_{\text{Gamma}}(x) \leq C_2 \exp(-x)$ for $x \geq 1$. What are C_1 and C_2? [5pts]

6. Use this information to write down a probability density $q(x|\alpha)$ and a scalar M such that $p_{\text{Gamma}}(x|\alpha, 1) \leq M \cdot q(x|\alpha)$. [5pts]

7. Now provide pseudocode for a rejection sampler to sample from $p_{\text{Gamma}}(x|\alpha, 1)$. Write down the acceptance probability. [10pts]

8. How would you use this to generate a Gamma with arbitrary shape parameter, and rate equal to one. How about arbitrary shape and rate parameters? [10pts]

2 Problem 1: Exploring the space of permutations with Metropolis-Hastings [100]

The file message.txt on the course webpage gives a paragraph of English encoded by a permutation code, where every symbol is mapped to a (usually) different one. The encryption key σ might thus be:

$$\begin{aligned} a &\to v \\ b &\to l \\ c &\to n \\ d &\to . \\ &\dots \end{aligned}$$

and a message like "`beware of dog.`" might read as "`lgavfgp wp. c`". To make life simple, assume there are only 30 unique symbols, `Symbol` = ('a', 'b',...,'z', '.', ',', ' ' and ':'). Thus the encription is a bijective function σ : `Symbol` $\to$ `Symbol`. Your job is to recover the original message (or equivalently find σ).

1. How many possible functions σ are there? [5 pts]

Clearly, brute force search is impossible. Instead, we will fit and use a model of the English language.

Download a large corpus of English (I used 'War and Peace' from Project Gutenberg). Use your favourite text-editor to covert all text to lowercase. You might also want to delete some of the clutter at the beginning and end of the file.

2. Read the file into `R` using the `readChar()` command. This will return a long string, split it up into a list of characters using the `strsplit()` command. The `table()` command will then give you the frequency of each unique character. This text contains more characters than we are interested in (we only care about `Symbol`). Extract the appropriate symbols from the table, and calculate and plot a histogram of the frequency of each character in `Symbol`. What is the entropy of this distribution? (This is a very crude estimate of the entropy of the English language, if you're interested, see http://www.math.ucsd.edu/~crypto/java/ENTROPY/ for how Shannon got a better (model-free) estimate using human subjects) [10 pts]

We will actually model English text as a first-order Markov process, characterized by a 30×30 transition matrix T. We will write this as $P_{Eng}(\cdot|T)$ (assume the distribution of the first symbol is uniform).

3. Given some data D, what are its sufficient statistics? These don't have to be minimal. What is the joint probability? What is the maximum likelihood estimate of T give D? [10 pts]

4. Now estimate T for the English text you downloaded. The simplest way to do this is by understanding what `table()` with two arguments does and then running `table(txt[1:length_txt-1], txt[2:length_txt])`. Plot the estimated T using a heatmap. For which symbols does the distribution of the following symbol have the largest and smallest entropy? [10 pts]

We can now write a Bayesian model to estimate the permutation σ. We place a uniform prior over σ:

$$p(\sigma) \propto 1$$

Given a message X encoded with σ, we expect $\sigma^{-1}(X)$ to be English text. We can evaluate the probability it is English using our Markov model of English:

$$p(X|\sigma) = P_{Eng}(\sigma^{-1}(X)|T)$$

5. Write a function for this likelihood. It should take a transition matrix `T`, a message `X` (or the sufficient statistics of `X`) and a permutation `perm`, and calculate the log-probability. (I implemented the permutation as a named-list, so that e.g. `perm['a']` gives `'v'`). [10 pts]

We are interested in the posterior $p(\sigma|X) \propto p(X|\sigma)p(\sigma)$.

6. Explain briefly why directly sampling from this is not easy. [2 pts]

Instead, we will use a simple Metropolis-Hastings algorithm. Start with an arbitrary permutation (I used the identity map). Randomly pick two symbols and propose a new permutation that swaps these values of the previous permutation

7. Write down the proposal distribution and the acceptance probability? [5 pts]

8. Finally implement this MH algorithm and run it on message.txt. I recovered the (almost perfect) answer after about 3000 iterations. Plot the evolution of the likelihood over your MCMC run. Also print the first 20 characters of the decoded message every 100 iterations. It might help to start by using your own encoded message where you know the true answer. Specify how many iterations your ran it for and what your burn-in period was. [25 pts]

9. Under the posterior distribution, which of the observed symbols had the highest and lowest uncertainty about their true value? Which ones did it seem to get wrong? [5 pts]

10. Consider an i.i.d. model of English text where we assume the next symbol is drawn from the solution to part 2? Does this work? All you have to do is change the function from part 5. Again, plot the decoded output every 100 iterations. [15 pts]

11. Comment on the usefulness and feasibility of using higher-order Markov models for decoding (where the next symbol depends e.g. on the previous two or five symbols). [3 pts]