Statistical Programming — Worksheet 6

1. Numbers. Dr. Winkel has 200 square tiles with which to decorate a wall of the kitchen in the Department of Statistics. 20 of the tiles are red, 30 blue, and the rest are white. Write down a formula for the number of distinct patterns he can create.

How many digits does this number have?

How many digits does 1000! have?

2. Metropolis Hastings. Suppose that $X_1, \dots, X_n \overset{\text{i.i.d.}}{\sim} \text{Gamma}(\alpha, \beta)$, and let α and β have independent Exponential(1) priors.

(a) Write a function to evaluate the log-posterior of α and β given a vector of data $\boldsymbol{x}$. The function should have arguments `x`, `alpha` and `beta`.

(b) Write a function to perform a single Metropolis-Hastings step to explore the posterior above. Use a proposal

$$\alpha' = \alpha + \sigma Z_1 \qquad \beta' = \beta + \sigma Z_2$$

for Z_1, Z_2 independent standard normals (i.e. $q(\alpha' \mid \alpha) \sim N(\alpha, \sigma^2)$.) It should take as arguments `x`, `alpha`, `beta` and `sigma`.

(c) Write a function to run the Metropolis-Hastings algorithm for N steps and return an $N \times 2$ matrix of the parameter values. It should take as input the data `x`, number of steps `N`, starting values `alpha` and `beta`, and proposal standard deviation `sigma`.

(d) The file `airpol.txt` (on the class website) contains daily PM2.5 readings taken from various measuring stations around Seattle during 2015. Read in the data as a vector and plot it in a histogram.

```
x <- scan("airpol.txt")  # note use of scan(), not read.table()
hist(x, breaks = 100, freq = FALSE)
```

Model the data as i.i.d. Gamma distributed observations using the model above. Run your Metropolis-Hastings algorithm for 5,000 steps with starting point $\alpha = 1$, $\beta = 1$. Plot your output with `plot()` and investigate different values of $\sigma \in \{0.01, 0.02, 0.05\}$.

(e) Find the posterior means for α and β. Plot the density of the corresponding Gamma distribution over the histogram of the data.

3. Image Reconstruction. Let the $n \times n$ matrix $Y = (y_{ij})$ of ± 1s follow the distribution of the Ising model with parameter θ, so that

$$\pi(Y) \propto \exp\left\{\theta \sum_{(i,j)\sim(i'j')} y_{ij} y_{i'j'}\right\}$$

where $(i,j) \sim (i'j')$ if either $i = i' \pm 1$ and $j = j'$, or vice versa (i.e. they differ by exactly one column or one row, but not both).

(a) Let $\widetilde{Y} = Y$ except that $\widetilde{y}_{ij} = 1 - y_{ij}$ (so they are equal except for a single entry). Show that

$$\log \pi(\widetilde{Y}) - \log \pi(Y) = \theta(d_{i,j} - 2a_{i,j})$$

where $d_{i,j}$ is the number of pixels adjacent to i, j, and $a_{i,j}$ is the number of adjacent pixels which have the same value as y_{ij}.

We will construct a Metropolis-Hasting algorithm to target π.

(b) First, look at the function `mh_step()` in the file `MHcode.R` on the website. The function performs one M-H step by proposing to flip `Y[r,c]`.
Complete the function by replacing the questions marks with code to calculate $\log \alpha$. Comment the code to show you understand what the rest of the function is doing.

(c) Now create a function with arguments `n`, `N` and `theta` which creates an $n \times n$ matrix with random entries 0 or 1, and then performs N M-H steps by calling `mh_step()`. When finished, it should return the state of the chain.

(d) Run the function for $n = 50$ and values $\theta = 0.2, 0.5, 0.8$ (you'll probably need $N > 10^5$ to get reasonable convergence). You can plot your solution using the `image()` function:

```
> out <- mh_ising(50, theta=0.5, N=1e5)
> image(out)
```

(e) Consider an $n \times n$ matrix $X = (x_{ij})$ of independent Bernoulli random variables, where

$$P(x_{ij} = 1) = \begin{cases} 1-p & \text{if } y_{ij} = 0 \\ p & \text{if } y_{ij} = 1 \end{cases}$$

for an unknown matrix of numbers $Y = (y_{ij})$. Defining $\widetilde{Y}$ as in (a), show that

$$\log L(\widetilde{Y}; X) - \log L(Y; X) = \begin{cases} +\log \frac{p}{1-p} & \text{if } y_{ij} \neq x_{ij} \\ -\log \frac{p}{1-p} & \text{if } y_{ij} = x_{ij} \end{cases},$$

where $L(Y; X)$ is the likelihood for the unknown parameter Y given X.

(f) Read in the data and look at it:

```
X <- as.matrix(read.table("image_noisy.txt"))
image(X)
```

Modify your previous M-H functions to accept a matrix X of data as an argument, and to include the change in the likelihood in your acceptance ratio α. Have the function return the estimated posterior mean of the chain (i.e. the average position of each pixel over the iterations).
Run the chain for a million iterations, setting $p = \frac{2}{3}$ and $\theta = 0.8$, and plot the results.