**IS 604 – Homework #5**

**Include the programming code (R, MATLAB, or Python) you used for each problem.**

1. At the start of each week, the condition of a machine is determined by measuring the amount of electrical current it uses. According to its amperage reading, the machine is categorized as being in one of the following four states: low, medium, high and failed. A machine in the low state has a probability of 0.05, 0.03, and 0.02 of being in the medium, high, or failed state, respectively, at the start of the next week. A machine in the medium state has a probability of 0.09 and 0.06 of being in the high or failed state, respectively, at the start of the next week (it cannot, by itself, go to the low state). And, a machine in the high state has a probability of 0.1 of being in the failed state at the start of the next week (it cannot, by itself, go to the low or medium state). If a machine is in the failed state at the start of a week, repair is immediately begun on the machine so that it will (with probability 1) be in the low state at the start of the following week. Let $X$ be a Markov chain where $X_n$ is the state of the machine at the start of week $n$.

**a)** Give the Markov transition matrix for $X$.

**b)** A new machine always starts in the low state. What is the probability that the machine is in the failed state three weeks after it is new?

**c)** What is the probability that a machine has at least one failure three weeks after it is new?

**d)** What is the expected number of weeks after a new machine is installed until the first failure occurs?

**e)** On average, how many weeks per year is the machine working?

**f)** Each week that the machine is in the low state, a profit of $1000 is realized; each week that the machine is in the medium state, a profit of $500 is realized; each week that the machine is in the high state, a profit of $400 is realized; and the week in which a failure is fixed, a cost of $700 is incurred. What is the long-run average profit per week realized by the machine?

**g)** A suggestion has been made to change the maintenance policy for the machine. If at the start of a week the machine is in the high state, the machine will be taken out of service and repaired so that at the start of the next week it will again be in the low state. When a repair is made due to the machine being in the high state instead of a failed state, a cost of $600 is incurred. Is this new policy worthwhile?

2. Rao (1973) presented an example on genetic linkage of 197 animals in four categories. The group sizes are (125, 18, 20, 34). Assume that the probabilities of the corresponding multinomial distribution are $\left(\frac{1}{2}+\frac{\theta}{4}, \frac{1-\theta}{4}, \frac{1-\theta}{4}, \frac{\theta}{4}\right)$. Estimate the posterior distribution of $\theta$ given the observed sample, using the Metropolis-Hasting random walk sampler with a uniform proposal distribution. Note that the posterior distribution of $\theta$ given the observed frequencies $k = (k_1, k_2, k_3, k_4)$ is

$$f_{\theta|K}(\theta) \propto (2+\theta)^{k_1}(1-\theta)^{k_2+k_3}\theta^{k_4}$$

3. Let $y_1,\ldots,y_n$ be a sample from a Poisson distribution for which there is a suspicion of a change point $m$ along the observation process where the means change, $m = 1, \ldots, n$.

Given $m$, the observation distributions are:

$y_i \mid \lambda \sim \text{Poi}(\lambda)$, $i = 1,\ldots,m$
$y_i \mid \phi \sim \text{Poi}(\phi)$, $I = m+1,\ldots,n$.

The model is completed with independent prior distributions

$\lambda \sim \text{Gamma}(\alpha,\beta)$
$\phi \sim \text{Gamma}(\gamma, \delta)$,
$m \sim$ Uniform Discrete $(1,n)$.

Then the posterior joint density is:

$$\pi(\lambda,\phi,m \mid y_1,\ldots,y_n) \propto f(y_1,\ldots,y_n \mid \lambda,\phi,m)\,p(\lambda,\phi,m)$$
$$= \left[\prod_{i=1}^{m} f(y_i \mid \lambda) \prod_{i=m+1}^{n} f(y_i \mid \phi)\right] f(\lambda \mid \alpha,\beta)\, f(\phi \mid \gamma,\delta)\,\frac{1}{n}$$
$$\propto \left[\prod_{i=1}^{m} e^{-\lambda}\lambda^{y_i} \prod_{i=m+1}^{n} e^{-\phi}\phi^{y_i}\right]\left(\lambda^{\alpha-1}e^{-\beta\lambda}\right)\left(\phi^{\gamma-1}e^{-\delta\phi}\right)$$
$$\propto \lambda^{\alpha+s_m-1}e^{-(\beta+m)\lambda}\,\phi^{\gamma+s_n-s_m-1}e^{-(\delta+n-m)\phi}$$
$$s_m = \sum\nolimits_{i=1}^{m} y_i$$
$$s_n = \sum\nolimits_{i=1}^{n} y_i$$

Use the vector Y below, containing count data of coal mining accidents in the U.K. from 1851-1962, and perform Gibbs sampling on this data to perform inference on the joint posterior distribution of $(\lambda,\phi, m)$.

Y =
[4,5,4,1,0,4,3,4,0,6,3,3,4,0,2,6,3,3,5,4,5,3,1,4,4,1,5,5,3,4,2,5,2,2,3,4,2,1,3,2,2,1,1,1,1,3,0,0,1,0,1,
1,0,0,3,1,0,3,2,2,0,1,1,1,0,1,0,1,0,0,0,2,1,0,0,0,1,1,0,2,3,3,1,1,2,1,1,1,1,2,4,2,0,0,0,1,4,0,0,0,1,0,0,
0,0,0,1,0,0,1,0,1];

<u>Guide to Implementation:</u>

It turns out to work even better if you add two more distributions to the above conditional. Assume that the scale parameters $\beta$ and $\delta$ for the distributions of $\lambda$ and $\phi$, respectively are themselves distributed as Inverse Gamma

$$\beta \sim InvGamma(\alpha, \lambda + 1)$$
$$\delta \sim InvGamma(\gamma, \phi + 1)$$

Then implement the Gibbs sampling as a Markov chain of 5 values for each iteration: ($\lambda$, $\phi$, $m$, $\beta$, $\delta$).

**<u>Pseudo-code for Implementation</u>**

Initialize # of iterations, hyperparameters, initial guesses of the five parameters

Create a vector Mprob to contain the discrete probability distribution over all possible values of *m* (size *N*, the number of observations in *Y*).

Create a matrix Chain with 5 columns and as many rows as iterations

```
For i = 1 to number of iterations
        % Conditional on m, b, and d, update samples of l and f
        Calculate sum1 = sum(Y(1) to Y(m))
        Calculate aum2 = sum(Y(m+1) to Y(N))
        Sample lambda from Gamma(sum1+alpha, beta / (beta*m + 1))
        Sample phi from Gamma(sum2+gamma, delta / (delta*(n-m) + 1))

        % Update probability distribution of m, using function fullcondm() defined below
        For j = 1 to n
                Mprob(j) = fullcondm(j, lambda, phi, Y, n, alpha, beta, gamma, delta)
        end

        sample new m from discrete distribution in mprob

        % update samples of beta and delta conditional on lambda, phi, and m

        Beta = 1 / (sample from Gamma(alpha, lambda+1))
        Delta = 1 / (sample from Gamma(gamma, phi+1))
```

HINT: I found that occasionally, this sampling would give crazy huge numbers for beta, and delta which would break the MCMC, so to be safe, embed the above two statements in loop that only exits when new beta and delta samples are <= 99.

```
        Store new samples for (lambda, phi, m, beta, delta)

End (for loop)
```

Define function `fullcondm( m,lambda,phi,y,n,alpha0,beta0,gamma0,delta0 )` – the conditional probability that the true change point is the parameter value passed in m, given the other values.

Lamexp = sum(Y(1:m)) if m > 1, 0 otherwise
Phiexp = sum(Y(m+1:n) if m < n, 0 otherwise

Return prob = `lambda^(alpha0-1+lamexp)*exp(-(beta0+m)*lambda)*`
    `phi^(gamma0-1+phiexp)*exp(-(delta0+n-m)*phi);`

## Questions

a) Run your code for 5000 iterations. Create and save the following figures:

   Histogram of lambda
   Histogram of phi
   Histogram of m
   Plot of lambda vs. phi
   Plot of lambda vs. m
   Plot of beta vs. delta

b) When do you think the change point occurred, based on your results? Can you put a 95% confidence bound on this result? What were the average rates of coal mining accidents before and after the change? Are these results consistent with the time series of the observations? Why or why not?

c) How is Gibbs sampling different from the Metropolis-Hastings approach?

4. Implement a Metropolis-Hastings simulated annealing optimization to find a least cost solution to the traveling salesman problem (TSP). A sample problem is provided below, which is a 17x17 matrix of cost (C) between every pair of cities in a 17-city TSP problem.

C =

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 633 | 257 | 91 | 412 | 150 | 80 | 134 | 259 | 505 | 353 | 324 | 70 | 211 | 268 | 246 | 121 |
| 633 | 0 | 390 | 661 | 227 | 488 | 572 | 530 | 555 | 289 | 282 | 638 | 567 | 466 | 420 | 745 | 518 |
| 257 | 390 | 0 | 228 | 169 | 112 | 196 | 154 | 372 | 262 | 110 | 437 | 191 | 74 | 53 | 472 | 142 |
| 91 | 661 | 228 | 0 | 383 | 120 | 77 | 105 | 175 | 476 | 324 | 240 | 27 | 182 | 239 | 237 | 84 |
| 412 | 227 | 169 | 383 | 0 | 267 | 351 | 309 | 338 | 196 | 61 | 421 | 346 | 243 | 199 | 528 | 297 |
| 150 | 488 | 112 | 120 | 267 | 0 | 63 | 34 | 264 | 360 | 208 | 329 | 83 | 105 | 123 | 364 | 35 |
| 80 | 572 | 196 | 77 | 351 | 63 | 0 | 29 | 232 | 444 | 292 | 297 | 47 | 150 | 207 | 332 | 29 |
| 134 | 530 | 154 | 105 | 309 | 34 | 29 | 0 | 249 | 402 | 250 | 314 | 68 | 108 | 165 | 349 | 36 |
| 259 | 555 | 372 | 175 | 338 | 264 | 232 | 249 | 0 | 495 | 352 | 95 | 189 | 326 | 383 | 202 | 236 |
| 505 | 289 | 262 | 476 | 196 | 360 | 444 | 402 | 495 | 0 | 154 | 578 | 439 | 336 | 240 | 685 | 390 |
| 353 | 282 | 110 | 324 | 61 | 208 | 292 | 250 | 352 | 154 | 0 | 435 | 287 | 184 | 140 | 542 | 238 |
| 324 | 638 | 437 | 240 | 421 | 329 | 297 | 314 | 95 | 578 | 435 | 0 | 254 | 391 | 448 | 157 | 301 |
| 70 | 567 | 191 | 27 | 346 | 83 | 47 | 68 | 189 | 439 | 287 | 254 | 0 | 145 | 202 | 289 | 55 |
| 211 | 466 | 74 | 182 | 243 | 105 | 150 | 108 | 326 | 336 | 184 | 391 | 145 | 0 | 57 | 426 | 96 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 268 | 420 | 53 | 239 | 199 | 123 | 207 | 165 | 383 | 240 | 140 | 448 | 202 | 57 | 0 | 483 | 153 |
| 246 | 745 | 472 | 237 | 528 | 364 | 332 | 349 | 202 | 685 | 542 | 157 | 289 | 426 | 483 | 0 | 336 |
| 121 | 518 | 142 | 84 | 297 | 35 | 29 | 36 | 236 | 390 | 238 | 301 | 55 | 96 | 153 | 336 | 0 |

To implement:

- Write a function that returns the total cost of a candidate solution for TSP, given the proposed solution and the cost matrix.

- Write a function to perform simulated annealing that takes three input parameters: the initial temperature T0, beta (determines annealing schedule), and number of iterations, and returns two results: the solution vector x and the cost of the solution s.

The pseudo-code for this algorithm is:

```
Initialize: T=T0, C, n = length(C),
        initial guess x = random permutation of (1:17), Sx = cost(x)
        xbest = x, sbest = s
        set up vector to save the cost of current solution at each iteration

Loop from 1 to number of iterations
        Choose another random permutation of the integers 1-17
        Sort the first 2 in this list -> call this vector of two indices I
        Create a candidate solution by reordering current x between I(1) and I(2):
                Y = [x(1:I(1)-1),x(I(2):-1:I(1)),x(I(2)+1:end)];
                % y is now new proposed solution

        Set Sy = cost(y)
        If Sy < Sx then set alpha=1 otherwise set alpha = exp(-(Sy-Sx)/T);
        Draw u ~ U(0,1) and if u < alpha, set x=y, Sx=Sy
        Set T = beta * T
        Set xbest = x, sbest = Sx
        Save sbest in results(i)
End loop
```

Test this code with initial parameters T0=1, beta=0.9999, iterations = 10000. How does the cost of the minimum solution compare with the cost of a random tour? Plot the cost of the current best solution as a function of iteration number. What do you notice about the performance of simulated annealing? Repeat the simulated annealing procedure 3 or 4 times and comment on the variability of the minimum cost and the optimal solution from repeated attempts.