2012 FALL BIOSTAT 615/815 Homework #6

Due is Wednesday December 19th, 2012 11:59PM by google document (shared to hmkang@umich.edu and atks@umich.edu) containing the source code and answers to the questions. Also email of the compressed tar.gz file containing all the source codes. NO LATE SUBMISSION WILL BE ACCEPTED FOR THIS LAST HOMEWORK.

Problem 1. Traveling Salesman Problem (50 pts)

Implement the Traveling Salesman Problem program described in the class with the following extensions.

- When the number of input data points is less than 12, perform exhausitve search to report minumum, maximum, mean distance, with durations and the number of intervals.
- Regardless of the number of input data points, perform Simulated Annealing using the example code described in the lecture note. If possible, try to make the code more efficient. You will receive bonus points if your modification makes the algorithm.
- Try to evaluate 300,000 random points and take the minimum among the points evaluated.

Your example outputs are supposed to be as follows (* to be replaced to actual numbers).

```
$ ./hw-6-1 ~hmkang/Public/615/data/tsp.10.txt
-----------------------------------------------------------
Minimum distance = *.******
Maximum distance = *.*****
Mean distance = *.*****
Exhaustive search duration = **.** seconds
Number of trials = *******
-----------------------------------------------------------
SA distance = *.******
SA search Duration = *.** seconds
Number of trials = ******
-----------------------------------------------------------
Random distance = *.******
Random search Duration = *.** seconds
Number of trials = 300000
-----------------------------------------------------------

$ ./hw-6-1 ~hmkang/Public/615/data/tsp.100.txt
-----------------------------------------------------------
SA distance = *.*****
SA search Duration = **.** seconds
Number of trials = ******
-----------------------------------------------------------
Random distance = *.*****
Random search Duration = **.** seconds
Number of trials = 300000
-----------------------------------------------------------
```

Run your program and explain why you see such a different between exhaustive search, simulated annealing, and random distance between each cases. Based on your observations, briefly explain the advantages and disadvantages of each method.

Problem 2. Gibbs Sampler for Gaussian Mixture Models (50 pts)

1. Implement the Gibbs sampling methods in C++ and fit a mixture of two and three normal distributions to the data.
2. Using no burn in period and thin interval of 5,000, output the log-likelihood, mixing proportions, mean, and standard deviation of each of the parameters using 1 million iterations (you will have approximately 200 thinned observations to compute the parameters)

Below is an example output (using a different input data) when $k = 2$.

```
ITER  LLK    PI0    MU0    SD0    PI1    MU1    SD1
0  -22316.9 0.499351 1.00691  2.24057  0.50065  1.05183  2.26782
5000  -22317.4 0.493549 1.01449  2.24773  0.506451 1.0434   2.26029
10000 -22316.7 0.485749 1.00737  2.23362  0.514251 1.04937  2.27261
15000 -22315.1 0.483648 0.981594 2.2266   0.516352 1.07354  2.27857
20000 -22307.4 0.479448 0.940069 2.19487  0.520552 1.11208  2.30462
25000 -22301.2 0.480548 0.919209 2.17656  0.519452 1.13188  2.31912
30000 -22293.2 0.466747 0.886239 2.15929  0.533253 1.15504  2.32711
...
```

You can save the output by running

```
$ ./hw-6-2 ~hmkang/Public/615/data/normmix.txt 2 > out.k2.txt
$ ./hw-6-2 ~hmkang/Public/615/data/normmix.txt 3 > out.k3.txt
```

Run the following R code to plot the likelihoods, mixing proportions, means, and SDs for each component. Paste the pdf outputs produced to your submission for each $k = 2$ and $k = 3$. Between $k = 2$ and $k = 3$, which model fits to the data better? Explain why.

```
M <- as.data.frame(read.table('out.k2.txt',header=T))
pdf('out.k2.pdf')
par(mfrow=c(2,2))
par(cex=0.6)
par(mar=c(2,4,2,2), oma=c(1,1,1,1))
plot(M$ITER,M$LLK,xlab='Iterations',ylab='Log-likelihood',main='Log-likelihood')
plot(M$ITER,M$PI0,xlab='Iterations',ylab='Mixing Proportions',main='Mixing Proportions',ylim=c(0,1),col="red")
points(M$ITER,M$PI1,col="blue")
plot(M$ITER,M$MU0,xlab='Iterations',ylab='Means',main='Means',ylim=c(-0.5,5.5),col="red")
points(M$ITER,M$MU1,col="blue")
plot(M$ITER,M$SD0,xlab='Iterations',ylab='SD',main='SD',ylim=c(0,3),col="red")
points(M$ITER,M$SD1,col="blue")
dev.off()

M <- as.data.frame(read.table('out.k3.txt',header=T))
pdf('out.k3.pdf')
par(mfrow=c(2,2))
par(cex=0.6)
par(mar=c(2,4,2,2), oma=c(1,1,1,1))
plot(M$ITER,M$LLK,xlab='Iterations',ylab='Log-likelihood',main='Log-likelihood')
plot(M$ITER,M$PI0,xlab='Iterations',ylab='Mixing Proportions',main='Mixing Proportions',ylim=c(0,1),col="red")
points(M$ITER,M$PI1,col="blue")
points(M$ITER,M$PI2,col="green")
plot(M$ITER,M$MU0,xlab='Iterations',ylab='Means',main='Means',ylim=c(-0.5,5.5),col="red")
points(M$ITER,M$MU1,col="blue")
points(M$ITER,M$MU2,col="green")
plot(M$ITER,M$SD0,xlab='Iterations',ylab='SD',main='SD',ylim=c(0,3),col="red")
points(M$ITER,M$SD1,col="blue")
points(M$ITER,M$SD2,col="green")
dev.off()
```

Problem 3. Uniform Hidden Markov Model (50 pts)

Implement the uniform hidden Markov model described in the class by filling in the function defintion in the lecture note. You only need to implement the forward-backward algorithm (neither Viterbi nor Baum-Welch).

Write a program that runs both (1) regular hidden Markov model (from your previous homework), and (2) the efficient implementation of uniform hidden Markov model under the following transition and emission probabilities, and measure the time elapsed using clock() function, similar to the problem 1.

$$\Pr(q_t = j | q_{t-1} = i) = \begin{cases} \frac{1}{n}\theta & i \neq j \\ 1 - \frac{n-1}{n}\theta & i = j \end{cases}$$

$$\Pr(o_t = j | q_t = i) = \begin{cases} \frac{1}{n}\mu & i \neq j \\ 1 - \frac{n-1}{n}\mu & i = j \end{cases}$$

At each row, the program should print the sequence (time) of each observation, the observed value, and $\Pr(q_t = o_t | \lambda, \mathbf{o})$ for each result of HMM forward-backward algorithm as the following example of $\theta = 0.1$ and $\mu = 0.5$ with 100 states. Assume that the priors are uniform across all possible states.

```
$ ./hw-6-3 100 0.1 0.5 ~hmkang/Public/615/data/uhmm.obs.txt
...
990     27      0.000127432     0.000127432
991     68      0.999996        0.999996
992     68      0.999996        0.999996
993     63      0.000129007     0.000129007
994     68      0.999988        0.999988
995     1       0.000129009     0.000129009
996     68      0.999996        0.999996
997     68      0.999996        0.999996
998     85      0.000140803     0.000140803
999     68      0.999957        0.999957
1000    68      0.998875        0.998875
Uniform HMM: 0.09 seconds
Regular HMM: 4.88 seconds
```

Store the full output into a file from the example above and include in your submission. Briefly explain why you observe similarity or difference between the two methods, in terms of outcomes and computational efficiency.