CS475 - Assignment 7

(K-Means)

REWARD: 160 points

DUE: Thu Apr 14 at 11pm PT

TEST DATA: testDataA7.tar or testDataA7.zip
LIBRARY: mat.tar, mat.zip
The 7.5 library includes an extra asserts that might catch attempting to take the mean or stddev of too few elements.

---

1 The Task

Write a program called kmeans to perform k-means analysis. On the command line they should take the number of expected groups called k and the number of times it will initialize the starting points and then run the algorithm to convergence. This second number I will call t, for trials. From standard input the program will read a matrix of input points in the form for the matrix library.

Your program should then run the k-means algorithm to determine the k center points that are the centers of each group of input points. It will do this t times. Each time it should begin by initializing the k cluster centers to unique random points in the data space. That is, pick k of the rows randomly without replacement to be the initial points for the space.

It will loop will cluster all the data points to the closest point from the k points. Then take the mean of the cluster to find the a new location for that center point for that group. (See the algorithm.) It is possible that this process will find that for some center point the process finds no points in its group. In that case, reinitialize that center point randomly as it did at the beginning. Continue the loop until the center points do not change.

When done, SORT the center points and then print them. Also print the the number of iterations it takes to converge (see sample output). It should also print the sum of (the average distance between all the points in each cluster):

$$\text{avg}(k, C) = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{\vec{x} \in C_i} \text{dist}(\vec{x}, \mu_i)$$

Distance should be the sum of squares of differences which is the square of the Euclidean distance.

Most importantly: after doing this t times, it should print the set of points with the smallest sum of average distance. See sample output below.

Remember: do the initialization/loop/print t times and sort the list of points before printing. Be sure to match the output format. In this example $k = 5, t = 2$:

```
kmeanst 5 2  < mix5.txt
Num Tries: 4
(size of Pts: 5 X 2)
  0.1941   0.6898
  0.1988   0.1034
  0.5011   0.5019
  0.8023   0.0989
  0.8038   0.6956
total average dist: 0.0586745
Num Tries: 9
(size of Pts: 5 X 2)
  0.1941   0.6898
  0.4156   0.1018
  0.4988   0.5009
  0.7927   0.6373
  0.8146   0.7591
total average dist: 0.111652
(size of Best Points: 5 X 2)
  0.1941   0.6898
  0.1988   0.1034
  0.5011   0.5019
  0.8023   0.0989
  0.8038   0.6956
best average dist: 0.0586745
```

2 Submission

Homework will be submitted as an **uncompressed** tar file to the homework submission page linked from the class web page. Be sure to include a make file to build your code and do NOT include the picture files. I will supply some. You can submit as many times as you like. **The LAST file you submit BEFORE the deadline will be the one graded.** For

all submissions you will receive email at your uidaho.edu mail address giving you some automated feedback on the unpacking and compiling and running of code and possibly some other things that can be autotested.

Have fun.