CS181: Assignment 3

Ina (Weilin) Chen and Kathy Lin

March 15, 2013

1. (a) For each dimension:

$$P(|x_m - y_m| \leq \epsilon) = 2\epsilon$$

For M dimensions, multiply this probability M times:

$$\rho = P(max_m|x_m - y_m| \leq \epsilon) = (2\epsilon)^M$$

(b) Here we are picking a point x that is not in the center and picking another point y. The region of values such that y would be within ϵ of x in every dimension would be a hypersphere with radius ϵ. This "sphere" has the same volume ratio compared to the sample space which is ρ. However, if x is picked near the border, this region would be cut off, as y cannot be picked beyond the border, and the probability would be less than ρ. Thus,

- if x is within $(\epsilon, 1-\epsilon)$ for all dimensions, $P(max|x_m - y_m| \leq \epsilon) = \rho$
- - otherwise, $P(max|x_m - y_m| \leq \epsilon) < \rho$

(c) Euclidean distance is

$$\begin{aligned}
&||\vec{x} - \vec{y}|| \\
&= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + max_m(x_m - y_m)^2 + \cdots + (x_m - y_m)^2} \\
&= \sqrt{max_m(x_m - y_m)^2 + \text{(non-negative terms)}} \\
&\geq \sqrt{max_m(x_m - y_m)^2}
\end{aligned}$$

Because

$$\sqrt{max_m(x_m - y_m)^2} \geq max_m|x_m - y_m|$$

It follows that

$$||\vec{x} - \vec{y}|| \geq max_m|x_m - y_m|$$

In part b) we showed that the probability $max_m|x_m - y_m| \leq \epsilon$ is at most ρ. Since $||\vec{x} - \vec{y}|| \geq max_m|x_m - y_m|$, the probability $||\vec{x} - \vec{y}|| \leq \epsilon$ is also at most ρ.

(d) Let there be N points, $\vec{y_1}, \vec{y_2}, ..., \vec{y_N}$. The probability that one point outside the radius ϵ (in all dimensions) is:

$$P(||\vec{x_i} - \vec{y_i}|| \geq \epsilon) \geq 1 - \rho$$

The probability for all points:

$$P(||\vec{x_1} - \vec{y_1}|| \geq \epsilon, ..., ||\vec{x_N} - \vec{y_N}|| \geq \epsilon) \geq (1 - \rho)^N$$

The probability that at least one point is inside the radius ϵ is

$$\begin{aligned} P(\text{one point inside}) &\leq 1 - (1 - \rho)^N \\ 1 - \delta &\leq 1 - (1 - \rho)^N \\ \delta &\geq (1 - p)^N \\ ln(\delta) &\geq Nln(1 - p) \end{aligned}$$

Since $0 < 1 - p < 1$, $ln(1 - p) < 0$

$$\begin{aligned} N &\geq \frac{ln(\delta)}{ln(1 - \rho)} \\ &= ln_{1-\rho}\delta \\ &= ln_{1-(2\epsilon)^M}\delta \end{aligned}$$

(e) The hierarchical agglomerative clustering algorithm is not very effective in high dimensional spaces. This is because as the dimensions scale up, the distance between points becomes exponentially larger (as seen above). Since with hierarchical agglomerative clustering we want to cluster by closer distances, the increasing sparsity of the data points poses a huge problem. This is phenomenon is also know as the curse of dimensionality.

2. (a) Given D, θ, $Pr(\theta)$, $Pr(D|\theta)$, find $Pr(x|D)$.

Maximum Likelihood: The maximum likelihood parameter estimate θ_{ML} is

$$\theta_{ML} = \underset{\theta}{argmax} Pr(D|\theta)$$

Probability of the new point given the data is the probability of the new point given the maximum likelihood estimate parameter

$$Pr(X|D) \approx Pr(x|\theta_{ML}) = Pr(x|\underset{\theta}{argmax} Pr(D|\theta))$$

Maximum a posterior:The maximum a posterior parameter estimate θ_{MAP} is

$$\theta_{MAP} = \underset{\theta}{argmax} Pr(D|\theta)Pr(\theta)$$

As before,

$$Pr(X|D) \approx Pr(x|\theta_{MAP}) = Pr(x|\underset{\theta}{argmax} Pr(D|\theta)Pr(\theta))$$

Full Bayesian: this approach provides that

$$Pr(x|D) = \int_{\Theta} Pr(x|\theta)Pr(\theta|D)d\theta$$

Using Bayes' Rules, we can substitute for $Pr(\theta|D)$

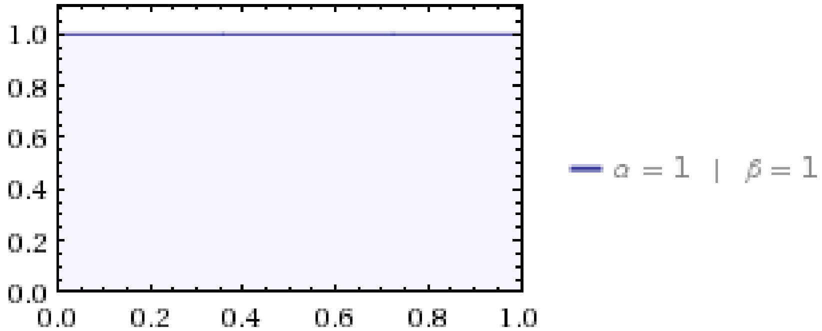
$$Pr(x|D) = \int_{\Theta} Pr(x|\theta)\frac{Pr(D|\theta)Pr(\theta)}{Pr(D)}$$

(b) The MAP method can be considered more Bayesian than the ML method because it takes into account the prior distribution of the estimate parameter: $Pr(\theta)$. Using the prior is part of the Bayesian statistics.

(c) MAP method is better than the ML method in that it introduces a prior for the estimate parameter. This lends to better estimates because the prior allows for regularization so that the effect of outlier estimate parameters is reduced. With the ML method, the extreme parameters are weighted equally but with the MAP method, the extreme parameters have lower probabilities of occuring in the prior distribution and thus would have lower weights.

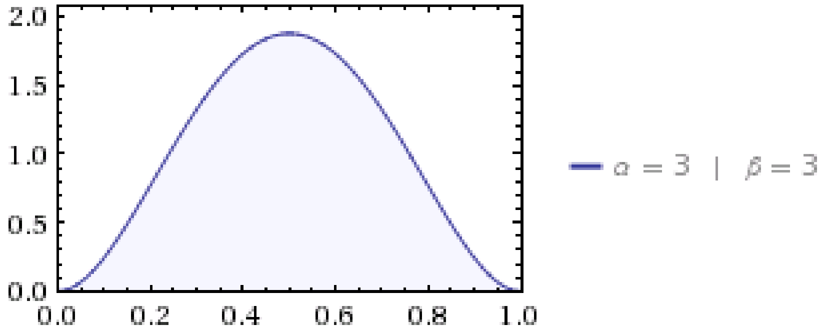
The MAP method is better than the FB method in that the Full Bayesian method involves complicated integrals that could be very messy to compute. The MAP method allows us to incorporate the prior without going through the computational challenge.

(d) Beta(1,1)



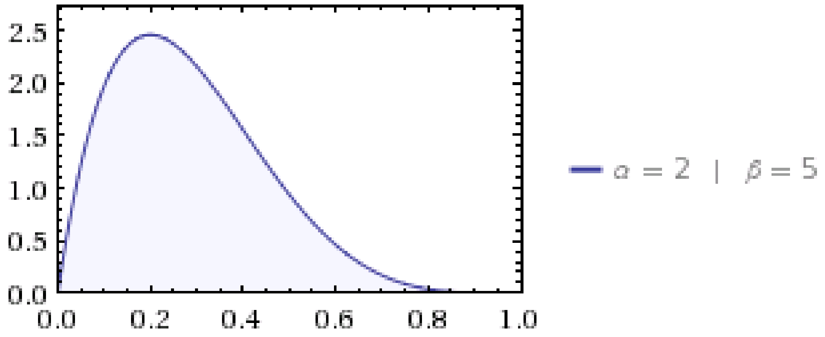
The intuition here is that we don't have any information about the probability of a win, thus all probabilities are equally likely (a uniform distribution).

Beta(3,3)



The intuition here is that the probability for a win is centered around 0.5. The distribution is symmetric, meaning that the soccer team is not more likely to win or lose but it is very unlikely that the team will win with probability 0 or 1

Beta(2,5)



The beta here is skewed towards the right, with the distribution centered around 0.2. The intuition behind using this Beta is that the team is more likely to lose than win, and that on average it has a probability of 0.2 of winning.

(e) The Beta distribution is useful because it is the conjugate prior to the Bernouilli distribution and the MAP method uses a prior distribution for the estimate parameters. The Beta distribution has two constants, to represent the binary variables used here. The posterior distribution is also Beta and can be obtained by simply updating the parameters.These features make the Beta a simple distribution to work with when carrying out the calculations for the MAP method. In addition, the Beta takes input from 0 to 1, which makes sense if we want to look at the values of the possible estimate parameters which are between 0 and 1.

(f) Data (2012-2013, first three games) = 3 wins, 0 loss. Prior (2011-2012) = 9 wins, 1 loss.

N_1 is the number of wins, N_0 number of losses, and $N = N_1 + N_0$ in the dataset.

ML:

$$\begin{aligned}
Pr(D|\theta) &= \underset{\theta}{argmax} Pr(D|\theta) \\
&= \theta^{N_1}(1-\theta)^{N_0} \\
lnPr(D|\theta) &= N_1 ln(\theta) + N_0 ln(1-\theta) \\
\frac{d}{d\theta} lnPr(D|\theta) &= \frac{N_1}{\theta} - \frac{N_0}{1-\theta} = 0 \\
\theta_{ML} &= \frac{N_1}{N_1 + N_0} = \frac{N_1}{N} \\
\theta_{ML} &= \frac{3}{3} = 1
\end{aligned}$$

MAP:
The prior distribution for θ is $Beta(9,1)$. The estimator θ_{MAP} can be derived as follows where x is the new data we are looking at.

$$\begin{aligned}
Pr(\theta|D) &\propto Pr(D|\theta)Pr(\theta) \\
&\propto \theta^{N_1}(1-\theta)^{N_0}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\
&= \theta^{N_1+\alpha-1}(1-\theta)^{N_0+\beta-1} \\
Pr(\theta|D) &= Beta(\Theta = \theta | N_1 + \alpha, N_0 + \beta)
\end{aligned}$$

Since $\theta_{MAP} = \underset{\theta}{argmax} Pr(\theta|D)$,

$$\theta_{MAP} = \frac{N_1 + \alpha - 1}{N + \beta - 2} = 1$$

FB:
The prior distribution is $Beta(9,1)$ and the posterior distribution is

$Beta(12, 1)$

$$\begin{aligned} Pr(x|D) &= \int_{\Theta} Pr(x = win|\theta) Pr(\theta|D) d\theta \\ &= \int_{0}^{1} \theta Pr(\theta|D) d\theta \\ &= \frac{\alpha + N_1}{\alpha + \beta + N} \\ &= \frac{12}{13} \end{aligned}$$

3. (a) The K-means clustering objective is to find K prototypes for the data (where the prototypes represent the clusters that they reside in) such that the sum of squared distances between prototypes and data is minimized.

Update steps:
We associate a binary indicator vector r_{nk} with each example x and cluster k such that r_{nk} indicates that the point x_n is associated with the cluster k and equals zero for the other clusters for the same point. Using this, we try to minimize the loss function

$$L(\mu_k{}_{k=1}^{k}, r_n{}_{n=1}^{N}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\vec{x}_n - \vec{\mu}_k||^2$$

We know that

$$||\vec{x}_n - \vec{\mu}_k||^2 = \sum_{m=1}^{M} (x_m - \mu_m)^2$$

Substituting we can find the gradient for the loss function. We try to move in the direction of gradient descent to minimize the loss function. This allows us to reach a local minimum where the error is the smallest in this region. This will then be used to update the prototypes so that error is minimized.

$$\frac{\partial L}{\partial \mu_k} = \sum_{n=1}^{N} -2 r_{nk} (\vec{x}_n - \vec{\mu}_k)$$

Setting this equal to zero we get:

$$0 = \sum_{n=1}^{N} -2 r_{nk} (\vec{x}_n - \vec{\mu}_k)$$

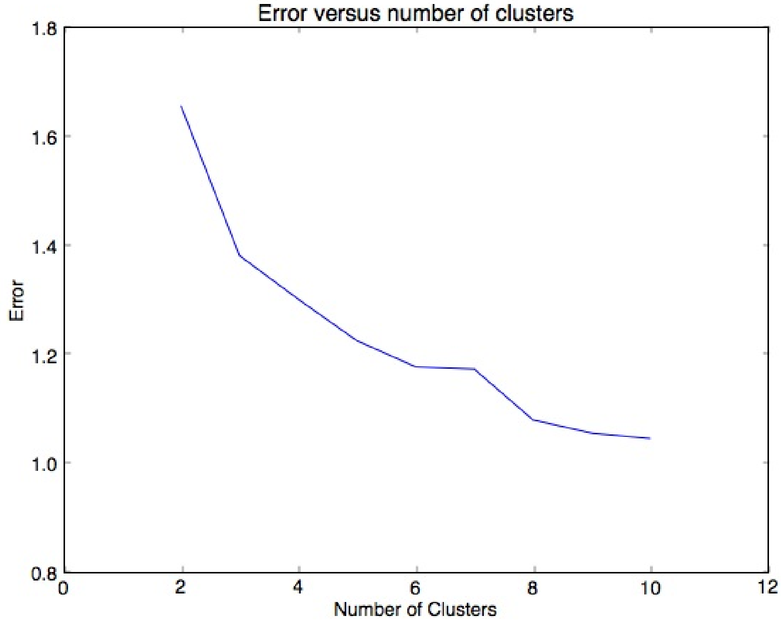
$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk} \vec{x}_n}{\sum_{n=1}^{N} r_{nk}}$$

(b) The Principle Component Analysis attempts to make a projection from the space of the data to a space of a lower dimension that preserves as much of the variance in the data as possible. Like the K Means algorithm, the PCA algorithm attempts to reduce the dimensionality of the data set. Whereas K Means does so by clustering the data into groups and giving the data a label (prototype), PCA does so by projecting the data from a higher dimension onto a lower dimension. In both cases, some sort of distance minimization is needed. For the K Means method, the distance of data in a cluster to the prototype of the cluster needs to be minimized. For the PCA method, the distance of the data to the projected line or plane needs to be minimized.

A situation where K means would be appropriate would have data that can be easily divided into groups (data in a group are close to one another but the groups are somewhat separated). An example would be pixel maps of images where there would be regions of roughly the same color being clustered together. K means would not be appropriate in situations where we need to separate data structures where the distance between data is not insightful in describing the features of the data. A dataset that looks like the pinwheel would be an example dataset where the arms of the spiral are close together that the points on different arms may be seen as close together by the K Means formula. More generally, K means would not be good for data where linear decision boundaries are not present.

A situtation where PCA would be appropriate would be a situation that gives data with a strong linear pattern. An example of such a data would be a data with linear features such as weight or height. A situation where PCA would not be appropriate would in scenarios where the data is uncorrelated or the direction of the largest variances is not the most interesting feature of the data. As before, an example data set for this scenario would be the pinwheel data structure where the linear projection would give any insight into the rotating spiral.

4. (a) i. Graph of mean squared error for K = 2,3,4,5,6,7,8,9,10

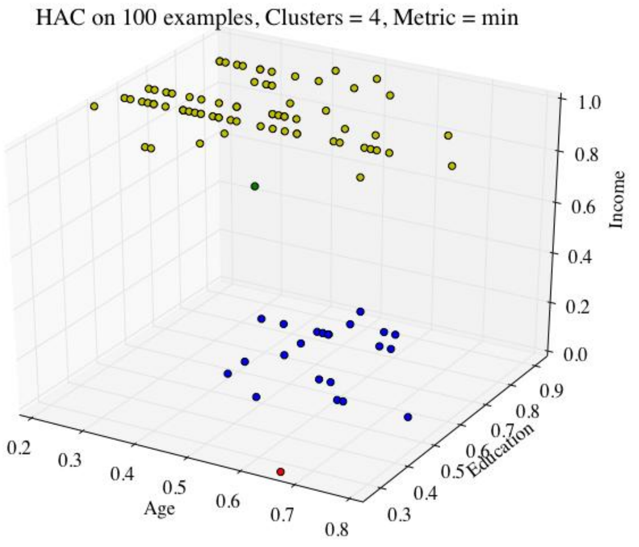


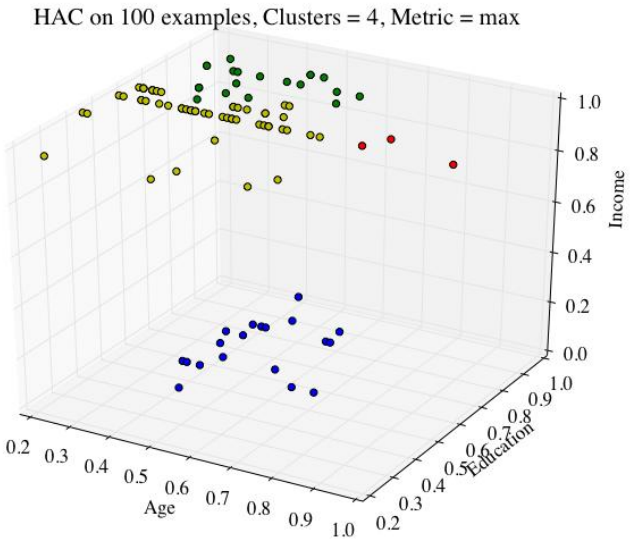
ii. From the graph, it is clear that the mean square error decreases as the number of clusters increasing, which makes sense because more clusters means that each individual point has a better chance of being closer to a prototype. However, we need to be aware of overfitting in this situation (in the extreme case, K = N would result in zero error, but have massive overfitting). Thus we should pick a K that is large enough to have a relatively low error while not so large that additional increases in K do not help the error improve very much. Thus, we would probably pick a K to be around 8 clusters.

(b) i. Comparing the the metric functions min and max

Number of instances in the clusters in the figures below:

Cluster Color	Metric = min	Metric = max
Red	1	3
Green	1	21
Blue	24	19
Yellow	74	57



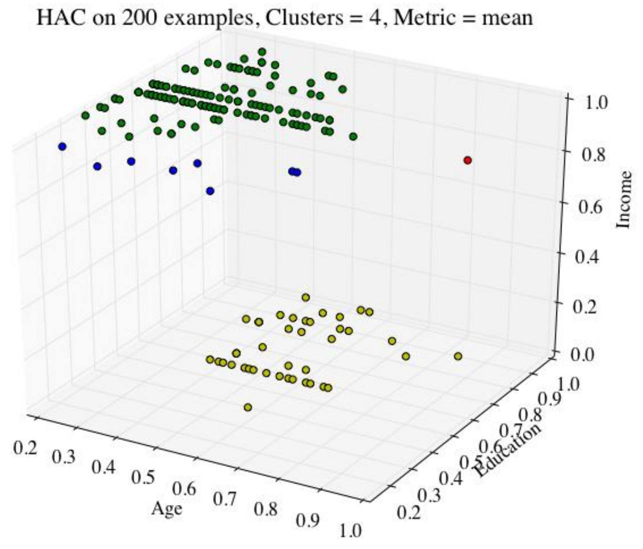


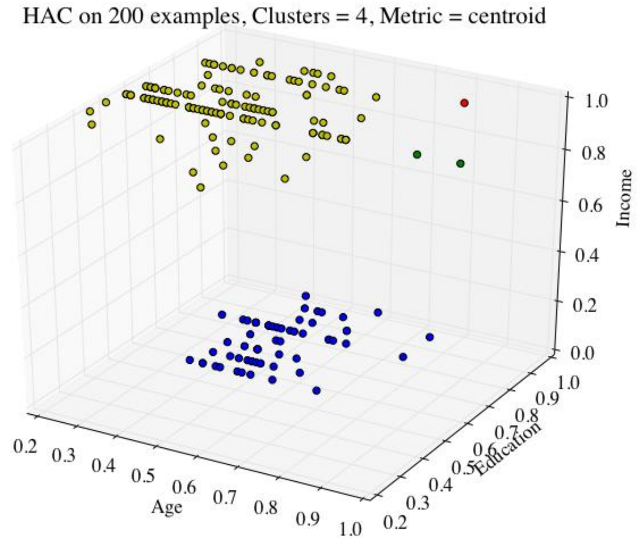
We noticed that the clusters produced using min are more elongated and sensitive to outliers (the two outliers are put into clusters by themselves) where as the clusters produced using max are more compact (the long top cluster seen in the figure for min is broken up into three smaller, more compact cluster here) and there are no clusters devoted to outliers so this metric care less about the outliers.

ii. Comparing the metric functions mean and centroid

Number of instances in the clusters in the figures below:

Cluster Color	Metric = mean	Metric = centroid
Red	1	1
Green	147	2
Blue	8	62
Yellow	44	135

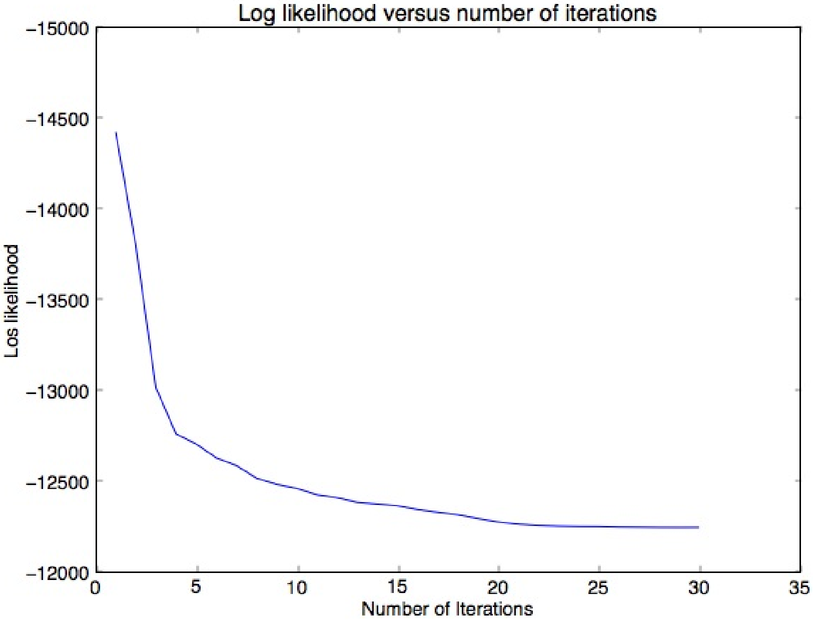




Here we see that the centroid and mean metrics are just as sensitive to outliers and produce roughly the same cluster shape. This makes sense as centroid and mean metrics are supposed to be compromises between the min and max metrics. However, we do see that the clusters produced by the mean metric are more likely to have the same variance, hence why the top portion of the data is divided in half for the mean such that the upper part with data that are closer together makes one cluster and the lower part with data that are more sparsely distributed making another cluster.

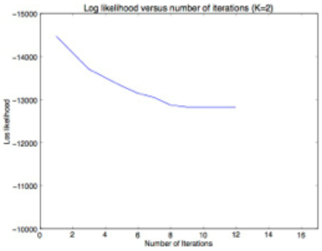
(c) i. It takes about 30 iterations for the parameters to converge.

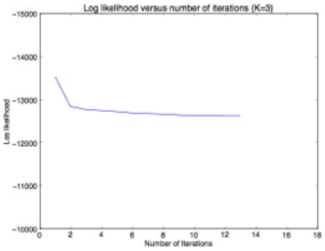
ii. Graph of log likelihood versus number of iterations:

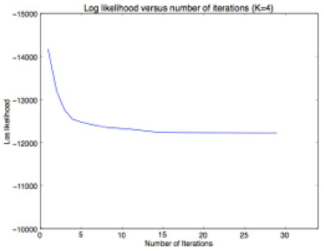


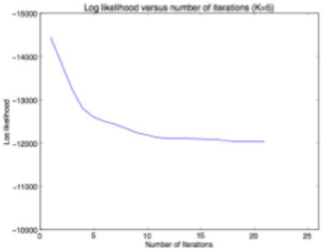
iii. Autoclass runs slower than K means because it has so many parameters to update and many large sums. One iteration of Autoclass takes about half a second (for a total of about 15 seconds) while the entire K means algorithm only takes about 3 or 4 seconds.

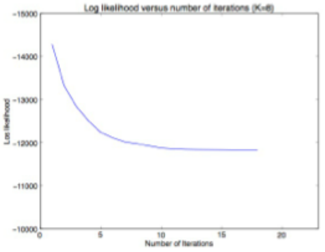
iv. Graphs for all values of K from 2 to 10:

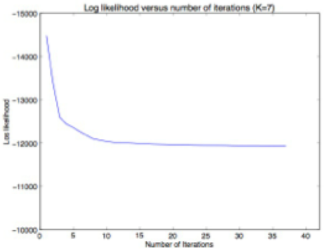


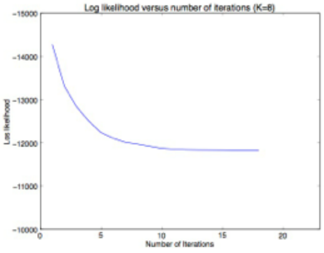


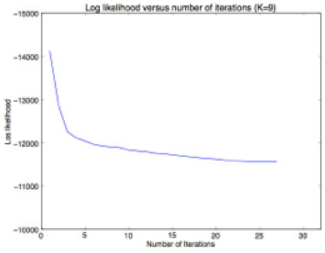


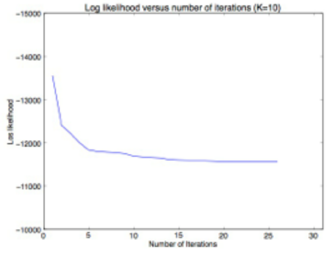












By K = 6, the log likelihoods stop increasing by significant amounts. We want to maximize the likelihood while also keeping the number of clusters down to avoid overfitting, so K = 6 is probably the best choice