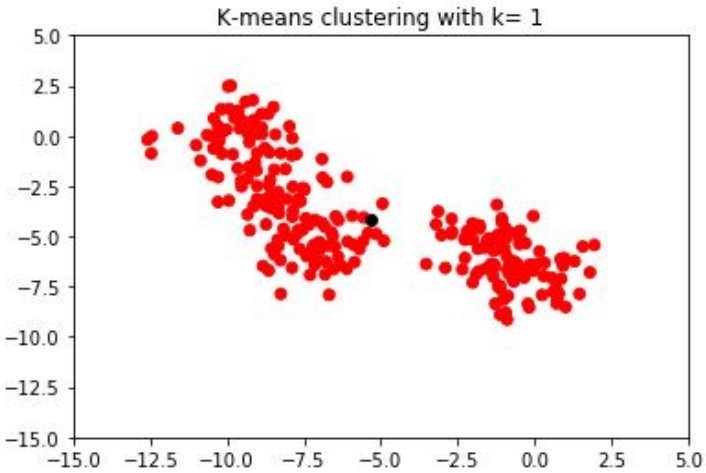
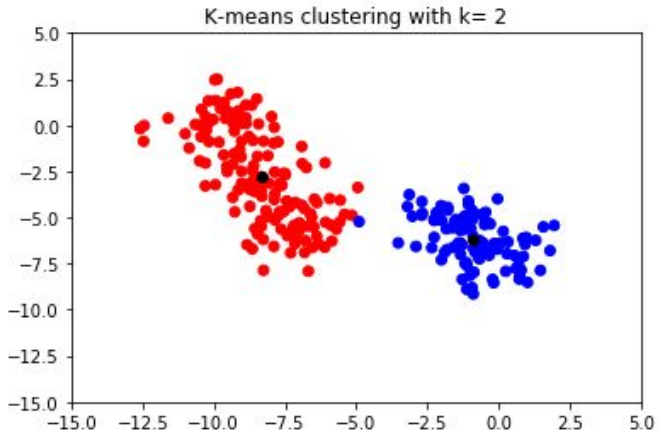
Abishkar Chhetri

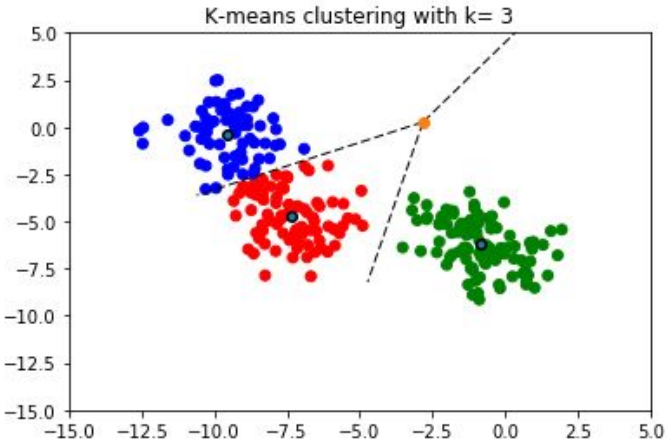
6.036 Writeup for Pset 3

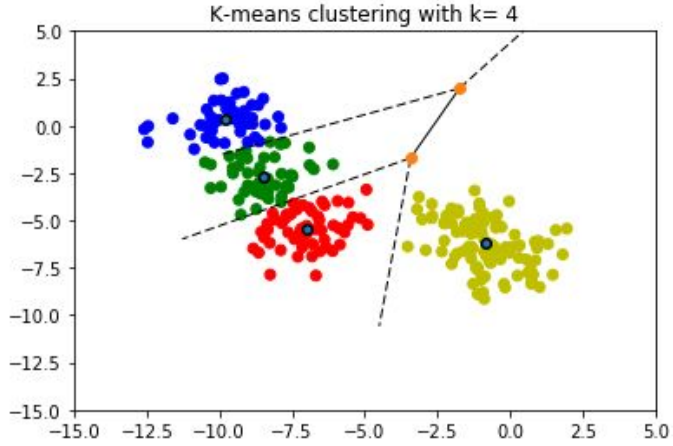
Part One

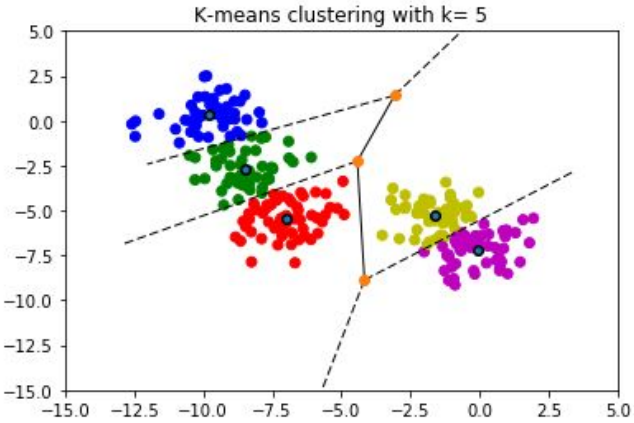
1.1 The following were the best K means solution plots for each K in [1,2,3,4,5].



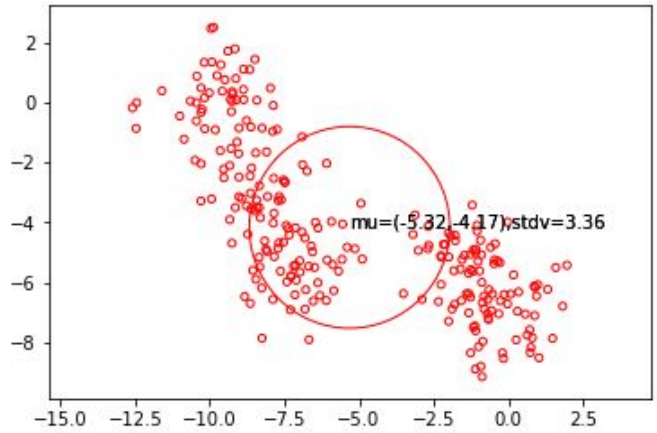


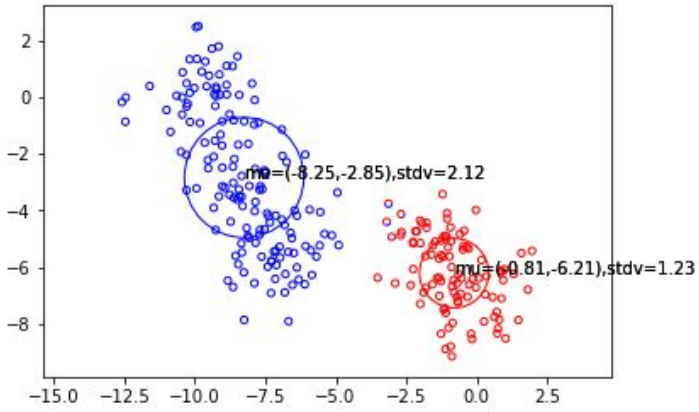


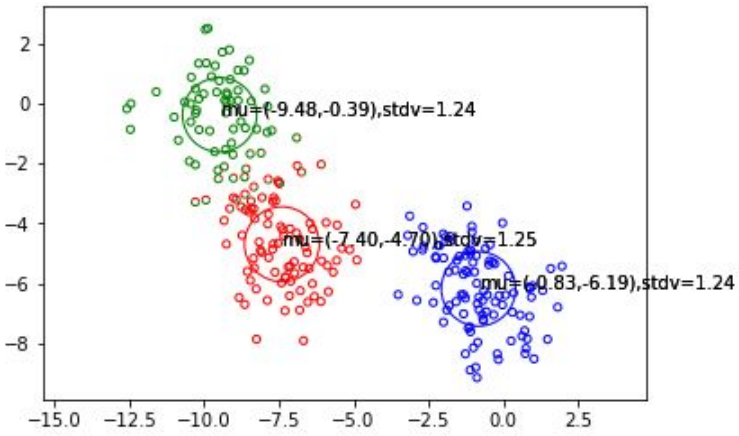


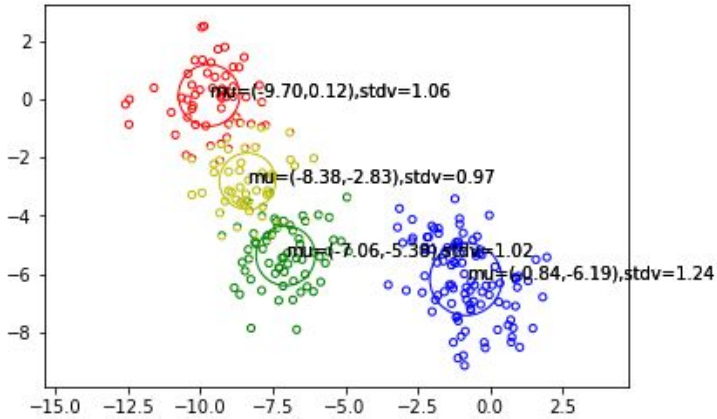


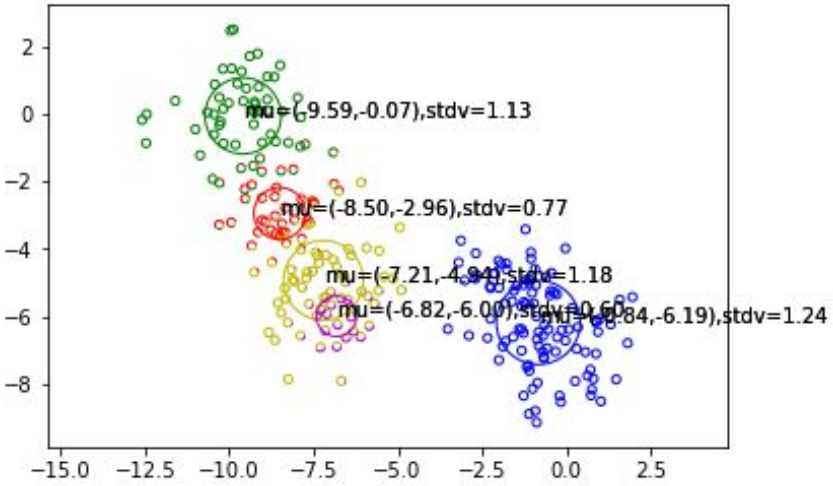
1.4. The following were the best EM solution plots for each K in [1,2,3,4,5].











1.5. Compare the best K-means and EM solutions for K = [1, 2, 3, 4, 5]. Briefly explain when, how, and why they differ.

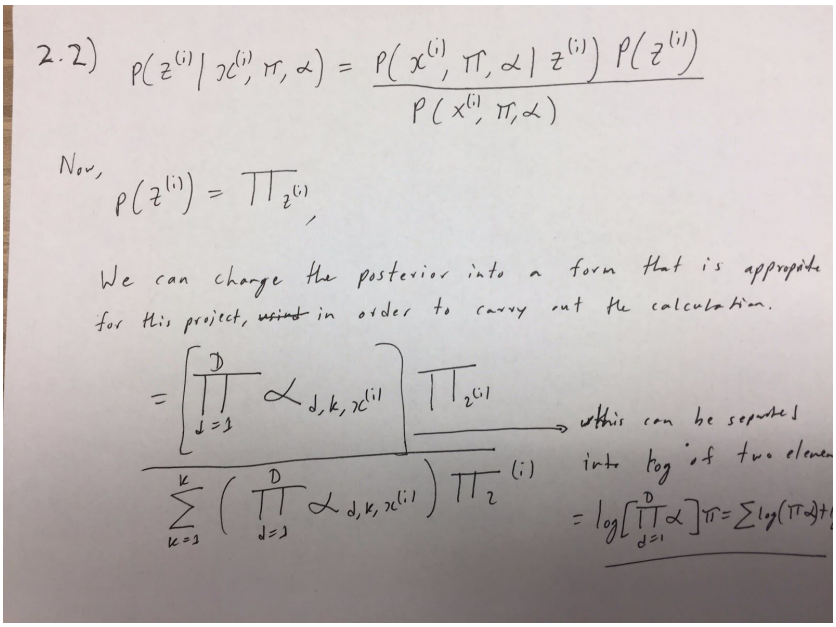
The K-means and the EM classification were fairly similar in the way they categorized the data except that the EM does a soft classification, rather than completely assigning a point to a cluster it assigns it a probability that it came from each of the clusters. Clearly for both, K = 4 produced the best results. The EM algorithm also gives more information about the clusters than the K means and uses more than just its relative position to a centroid to classify the point into a cluster.

Part Two

2.1) In our model, we have assumed that the features are independent, which is not realistic. **Explain why this will cause the model to produce overconfident cluster assignments.** Having made the independence assumption, show that the simple $(\ldots)_d$ expression can be obtained from the fully-observed model by marginalizing out unknown values.

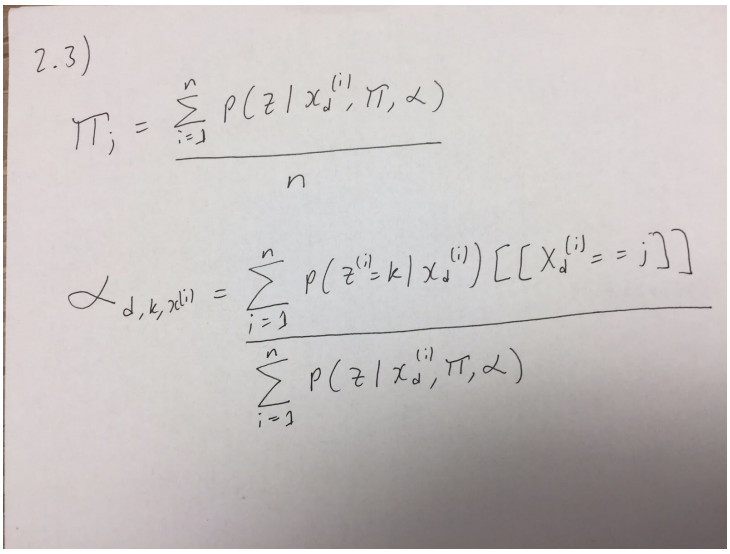
The model will produce overconfidence cluster assignments because it will group data points that don't necessarily take into account the relationship between the features. As we saw in the graph above in the EM K=2 case, the ideal shape of the gaussian is elliptical rather than circular which was produced. This is because we weren't using a covariance matrix which is equivalent to treating the feature as independent. By marginalizing out unknown values using an indicator variable matrix and then later taking a log, it helps us weigh the features that we know about more heavily and produce an overall better result.

2.2) Derive the posterior distribution for the latent variable associated to a particular data point, $p(z_{(i)} \mid x_{(i)}, \pi, \alpha)$. Your answer should be in terms of π, α, and $x_{(i)}$.



2.3 a) Write down the ML (maximum likelihood) estimate of π from the weighted data. You should find that it is analogous to the GMM case.

(b) Perform a similar computation for the ML estimate of α. The update will be different from Part 1 since we're using categorical rather than Gaussian mixture components.



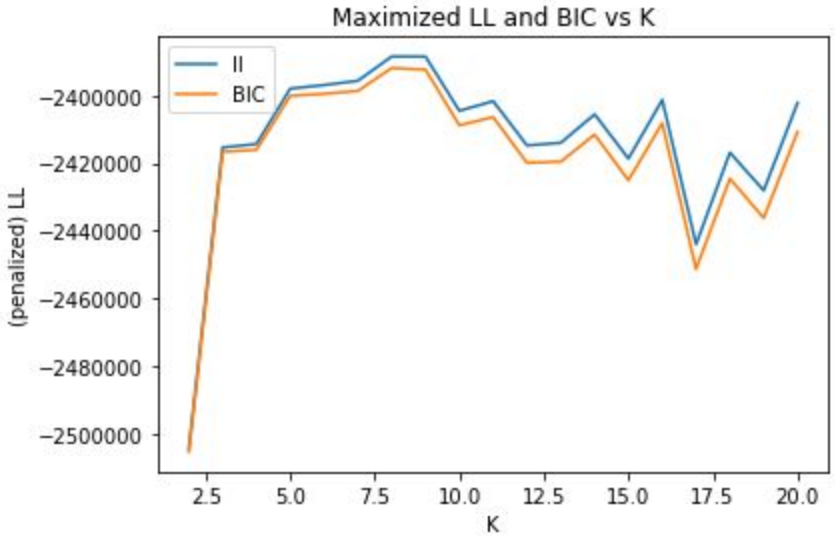
2.5) After verifying your E and M steps by running test_em_cmm, uncomment the lines associated with this part and run your EM algorithm on the full census dataset. Given what you know about the EM algorithm, comment on the overall trend of the log-likelihood over the iterations.

When I ran the tests, this is the report that lists the max log likelihood:

```
In [75]: runfile('/Users/abishkarchhetri/Dropbox (MIT)/AAA Subjects/AA Sophmore
Spring/Machine Learning/Projects/Project 3/release 2/main.py', wdir='/Users/
abishkarchhetri/Dropbox (MIT)/AAA Subjects/AA Sophmore Spring/Machine Learning/
Projects/Project 3/release 2')
Reloaded modules: project3, utils
Fitting k = 2: max ll = -2504509.96780  (0.27 min, 22 iters)
Fitting k = 3: max ll = -2415356.47804  (0.51 min, 32 iters)
Fitting k = 4: max ll = -2414338.66944  (0.26 min, 15 iters)
Fitting k = 5: max ll = -2397949.22262  (0.53 min, 28 iters)
Fitting k = 6: max ll = -2396893.17503  (0.82 min, 41 iters)
Fitting k = 7: max ll = -2395637.74136  (1.66 min, 76 iters)
Fitting k = 8: max ll = -2388441.38208  (0.42 min, 18 iters)
Fitting k = 9: max ll = -2388470.86838  (1.09 min, 43 iters)
Fitting k = 10: max ll = -2404511.96199  (0.90 min, 33 iters)
Fitting k = 11: max ll = -2401663.02075  (0.80 min, 28 iters)
Fitting k = 12: max ll = -2414750.15277  (1.18 min, 36 iters)
Fitting k = 13: max ll = -2413969.62694  (1.43 min, 40 iters)
Fitting k = 14: max ll = -2405538.15243  (1.92 min, 54 iters)
Fitting k = 15: max ll = -2418625.18292  (2.52 min, 65 iters)
Fitting k = 16: max ll = -2401280.68828  (1.36 min, 34 iters)
Fitting k = 17: max ll = -2443997.16958  (1.20 min, 28 iters)
Fitting k = 18: max ll = -2416862.17577  (1.39 min, 31 iters)
Fitting k = 19: max ll = -2428069.87235  (1.59 min, 34 iters)
Fitting k = 20: max ll = -2402162.10971  (3.14 min, 66 iters)
```

The max log likelihood increases steadily until K reaches a value of 9. As K increases from that point, the log likelihood steadily decreases. This is because beyond the certain optimal number of clusters, the data starts to overfit to the clusters.

2.6b) Uncomment and run the code in main.py that plots the maximized LLs and BICs for the trained models. Based on the graph of the LL, what value of K should you choose? What about for the BIC? Do both results agree?



Both results agree that the value of K should be near 8 or 9, we can just pick 9 as the ideal K value.

2.7. (10 points) The model defined in Equation 1, trained on census data, may be interpreted as identifying K demographics that describe the population. Using the model you just learned, you can answer interesting questions about the data:

(a) Call print clusters with the model trained with the best choice of K, as determined by the BIC. Qualitatively describe the clusters that your model finds; include several of the key features and their values.

Cluster 1:
age: 13 - 19, sex: female, birthplace: US, ancestry1: Western Europe (except Spain), citizen: born in US, income: none, edlevel: 5th - 8th grade, employer: n/a, under 16

This cluster had middle/high school teenagers with ancestry from western europe. This is essentially what we can call the “Generation Z”

Cluster 2:
age: 30 - 39, sex: female, birthplace: US, ancestry1: Western Europe (except Spain), citizen: born in US, income: $15k - $29999, edlevel: high school or ged, employer: private, for profit

This cluster represents the lower/lower middle class caucasian americans aged 30 to 40 who are part of the, “Generation X”

Cluster 3:
age: 0 - 12, sex: male, birthplace: US, ancestry1: Western Europe (except Spain), citizen: born in US, income: none, edlevel: 1st - 4th grade, employer: n/a, under 16

This cluster, another member of the “Generation Z”, were the youngest member of the population who have a lot of similar demographics as they’re still going to school.

Cluster 4:
age: 20 - 29, sex: male, birthplace: America (non US), ancestry1: Hispanic (including Spain) citizen: not a US citizen, income: $1 - $14999, edlevel: 5th - 8th grade, employer: private, for profit

This cluster is mostly consists of the Hispanic Millennials who are starting to join the workforce and are making up to 15k. Their education level is shown as 5 to 8th grade.

Cluster 5:
age: 30 - 39, sex: female, birthplace: Asia, ancestry1: Western Europe (except Spain), citizen: naturlized US citizen, income: $1 - $14999, edlevel: high school or ged, employer: private, for profit

This cluster represents the naturalized US citizens who were born in other countries but immigrated to the US. This demographic’s education level is generally high school/GED and they make around 15k.

Cluster 6:
age: 30 - 39, sex: male, birthplace: US, ancestry1: Western Europe (except Spain), citizen: born in US, income: $30k - $59999, edlevel: high school or ged, employer: private, for profit

This cluster represents middle class caucasian americans with a decent income level. They are also the

Cluster 7:
age: 65 and above, sex: female, birthplace: Asia, ancestry1: Western Europe (except Spain), citizen: not a US citizen, income: none, edlevel: 5th - 8th grade, employer: n/a, under 16

This cluster is the group of older non citizens who have a relatively low education level and no income.

Cluster 8:
age: 20 - 29, sex: female, birthplace: US, ancestry1: Western Europe (except Spain), citizen: born in US, income: $1 - $14999, ed level: high school or ged,employer: private, for profit

This cluster represents the Millennials born in the US who are currently in college or just joining the workforce.

Cluster 9:
age: 65 and above, sex: female, birthplace: US, ancestry1: Western Europe (except Spain), citizen: born in US, income: $1 - $14999, edlevel: high school or ged, employer: n/a, under 16

This cluster represents the older (65+) americans who have an income and a low educational level.

Next, call print clusters with a model trained with a larger or smaller value of K. How do the clusters found by the second model compare to the first? You may want to discuss whether the cluster identities are stable and, if not, what concepts has the second model added or removed?

Printing clusters with a larger value of K created greater divisions between the existing cluster. For example, the 30-40 age range in my K=9 clusters, broke down into people with a high school degree and people with a bachelor's degree with a higher amount of income. Decreasing the value of K, I assume, has the opposite effect as it might merge certain different clusters into one which can remove certain important distinctions between the clusters.

(b) Now we will explore the qualitative effects of different initializations. Run your best model several times and print the clusters. How stable are the discovered clusters? Are some found more often than others?

The discovered clusters are fairly stable, but there are some clusters that are more common or are found more commonly than others. For example the age range 0-12, 13-19 and 65+ were fairly stable, however, age 30-39 had categories of stable and unstable classifications.

(c) Imagine that you trained your model on the census data for each state. How could you use your model to determine how similar two states' populations are?

One could analyze the clusters and calculate how many clusters of similar characteristics the two states have. Another way you can compare the similarity of the two states by generating likely estimates from the two models and calculating the similarity between those generated examples.