6.036 Project 3

Dimitris Koutentakis

05 May, 2017

Part I - K-Means versus EM

Question 1

[ht!] The K-Means algorithm results in:

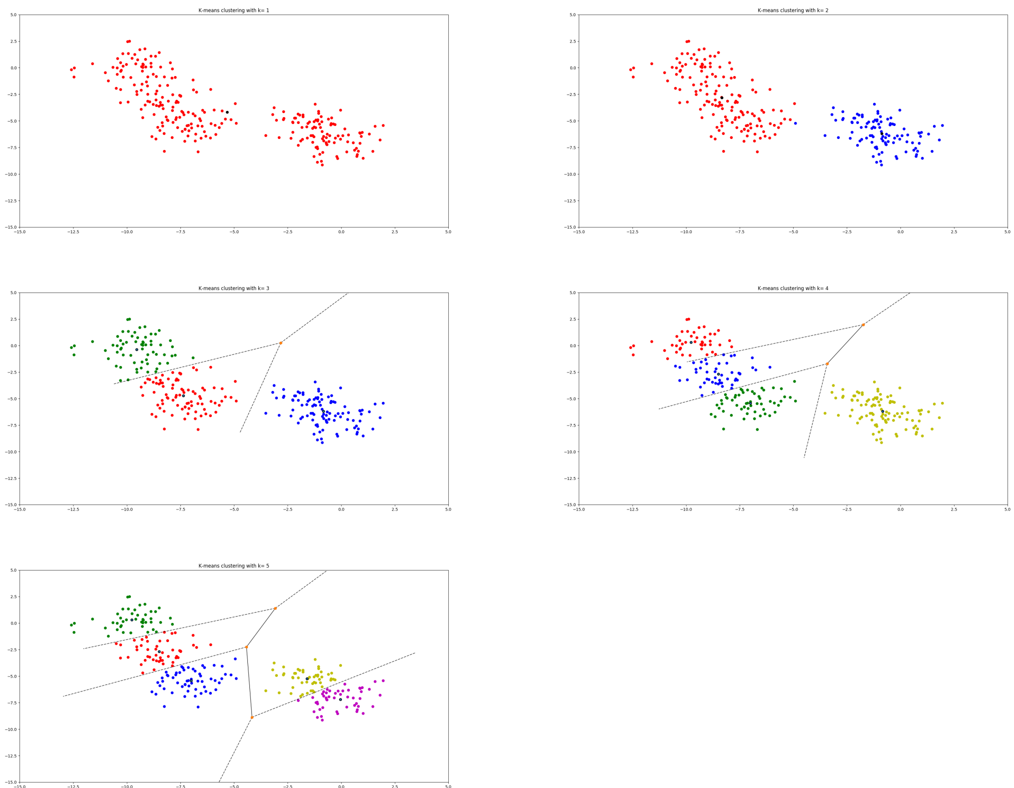

Figure 1: plots for k=[1,2,3,4,5]

Question 4

After running the EM algorithm for the Gaussian mixture model for cluster number of $K = [1, 2, 3, 4, 5]$, several times, I chose the plots for the following log-likelihood values:

```
Fitting k = 1: max ll = -1315.31768  (0.00 min, 3 iters)
Fitting k = 2: max ll = -1139.72995  (0.00 min, 10 iters)
Fitting k = 3: max ll = -1072.60383  (0.01 min, 26 iters)
Fitting k = 4: max ll = -1059.10908  (0.03 min, 40 iters)
Fitting k = 5: max ll = -1045.38417  (0.05 min, 62 iters)
```

Figure 2: log-likelihood values for EM algorithm

The plots I got are in the next page:

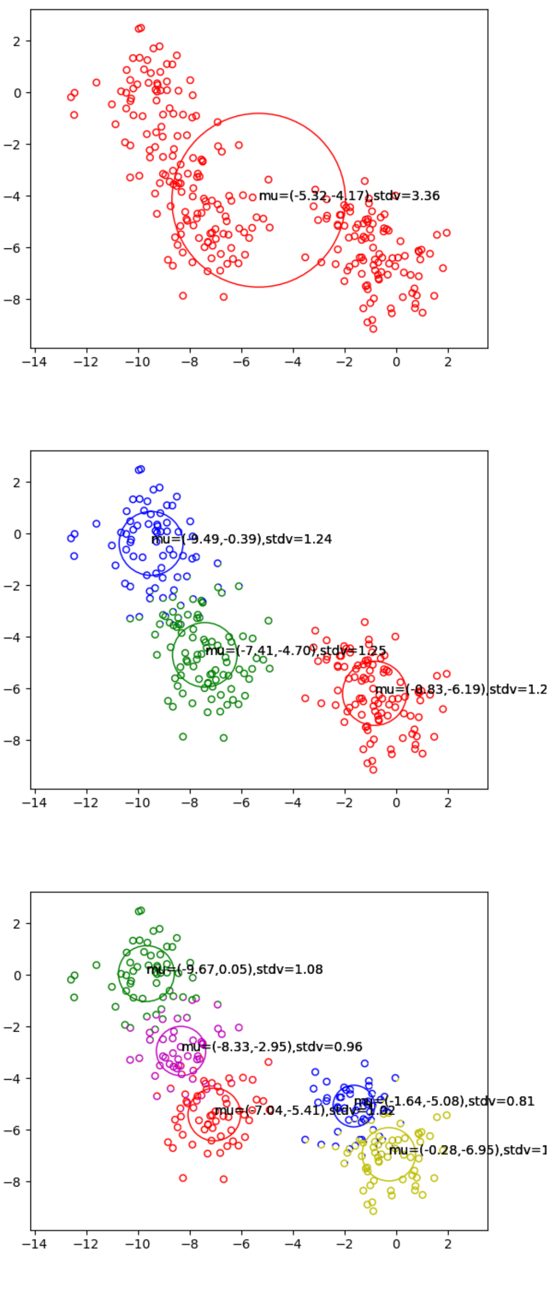

Figure 3: plots for k=[1,2,3,4,5]

Question 4

It is easily seen that the larger the K value, the more spread out the clusters will be and the smaller σ^2 will be. As we include more clusters, the algorithm sort the points into more specific groups instead of one larger and general one.

Part II - Clustering Census Data

1 Question 1

By assuming that the features are independent, we will have overconfident cluster assignments, as their distance will increase. As the distance increases, so does the weight of the points that are far away. Since the weight of those points increases, the clusters will be more strictly separated.

2 E-Step

From Bayes rule, we get that:

$$p(z^{(i)}|x^{(i)}, \pi, \alpha) = \frac{p(z^{(i)})p(x^{(i)}|z^{(i)}, \pi, \alpha)}{p(x^{(i)})}$$

By marginalizing over π, α, we get the following:

$$p(z^{(i)}|x^{(i)}, \pi, \alpha) = \frac{\pi_k \cdot \prod_{d=1}^{D} \alpha_{k,d}[x_d^{(i)}]^{[\![x_d^{(i)} \text{ is not missing}]\!]}}{\sum_{j=1}^{k} \pi_j \cdot \prod_{d=1}^{D} \alpha_{j,d}[x_d^{(i)}]^{[\![x_d^{(i)} \text{ is not missing}]\!]}}$$

3 M-Step

3.a) ML of π

The maximum likelihood estimate of π will be:

$$\pi_j = \frac{\sum_{i=1}^{N} p(z^{(i)=j}|x^{(i)}, \pi, \alpha)}{N}$$

3.b) ML of α

The maximum likelihood estimate of α will be:

$$\alpha_{k,d}[c] = \frac{1}{N} \cdot \sum_{i=1}^{N} p(z^{(i)} = j | x^{(i)}, \pi, \alpha) \cdot [\![x_d^{(i)} == c]\!]$$

Qiestion 5

For different K values, the maximum Log Likelihood does not change much. This is expected since the algorithm should perform about the same for low numbers of cluster we need to classify such a large amount of points in.

```
PS C:\Users\dkout\OneDrive\MIT\Junior Spring\6.036\project3> python main.py
Fitting k = 2: max ll = -2504434.91216  (0.28 min, 36 iters)
Fitting k = 3: max ll = -2500562.71668  (0.12 min, 13 iters)
Fitting k = 4: max ll = -2464155.19074  (0.18 min, 18 iters)
Fitting k = 5: max ll = -2380741.43433  (0.20 min, 19 iters)
Fitting k = 6: max ll = -2375977.30513  (0.81 min, 73 iters)
Fitting k = 7: max ll = -2371041.77476  (0.44 min, 35 iters)
Fitting k = 8: max ll = -2379920.00965  (0.60 min, 40 iters)
Fitting k = 9: max ll = -2387483.04171  (0.33 min, 23 iters)
Fitting k = 10: max ll = -2412975.97575  (0.56 min, 37 iters)
Fitting k = 11: max ll = -2396652.81084  (0.47 min, 29 iters)
Fitting k = 12: max ll = -2440695.01630  (0.91 min, 50 iters)
Fitting k = 13: max ll = -2407733.30134  (0.50 min, 31 iters)
Fitting k = 14: max ll = -2403693.57161  (0.28 min, 18 iters)
Fitting k = 15: max ll = -2386536.78079  (0.96 min, 58 iters)
Fitting k = 16: max ll = -2391742.33436  (0.50 min, 28 iters)
Fitting k = 17: max ll = -2406574.93282  (1.21 min, 67 iters)
Fitting k = 18: max ll = -2402315.23946  (1.12 min, 55 iters)
Fitting k = 19: max ll = -2405017.43905  (1.18 min, 63 iters)
Fitting k = 20: max ll = -2444785.15912  (1.81 min, 92 iters)
```

Figure 4: Maximum Log Likelihood for different K values

However, when we see the trend of the log likelihood for only one k, we see that it increases significantly. For example, for k=4, we can see that the log-likelihood increases from -499415 to 2393848. This can be seen in the following figure:

```
Fitting k = 4: Log likelihood =  -4991495.50085
Log likelihood =  -2929421.47089
Log likelihood =  -2762254.94477
Log likelihood =  -2668765.13394
Log likelihood =  -2633926.98304
Log likelihood =  -2610274.75486
Log likelihood =  -2592286.1571
Log likelihood =  -2579483.57373
Log likelihood =  -2570496.41905
Log likelihood =  -2563529.2233
Log likelihood =  -2557537.80112
Log likelihood =  -2552208.13999
Log likelihood =  -2547552.10211
Log likelihood =  -2543546.31188
Log likelihood =  -2540015.50553
Log likelihood =  -2536730.19992
Log likelihood =  -2533283.18336
Log likelihood =  -2529039.19417
Log likelihood =  -2524756.17378
Log likelihood =  -2521357.74575
Log likelihood =  -2518863.87752
Log likelihood =  -2517049.68753
Log likelihood =  -2515712.79763
Log likelihood =  -2514704.88398
Log likelihood =  -2513928.67226
Log likelihood =  -2513325.01695
Log likelihood =  -2512849.44538
Log likelihood =  -2512454.82068
Log likelihood =  -2512104.26079
Log likelihood =  -2511776.81971
Log likelihood =  -2511445.50961
Log likelihood =  -2511066.75778
Log likelihood =  -2510577.45387
Log likelihood =  -2509889.48608
Log likelihood =  -2508879.53343
Log likelihood =  -2507357.74246
Log likelihood =  -2504985.47395
Log likelihood =  -2501130.63723
Log likelihood =  -2494632.72175
Log likelihood =  -2483637.83209
Log likelihood =  -2466392.11393
Log likelihood =  -2443903.23675
Log likelihood =  -2421815.59497
Log likelihood =  -2405292.15162
Log likelihood =  -2396849.40363
Log likelihood =  -2394469.53324
Log likelihood =  -2394015.4017
Log likelihood =  -2393848.80582
max ll = -2393848.80582  (0.55 min, 48 iters)
```

Figure 5: Log-Likelihood progression for k=4

Question 6

Based on the figure shown below, the optimal value of K is 7 when judging based on the Log-Likelihood (LL) as well as when judging based on the Bayesian Information Criterion (BIC). Both results agree.

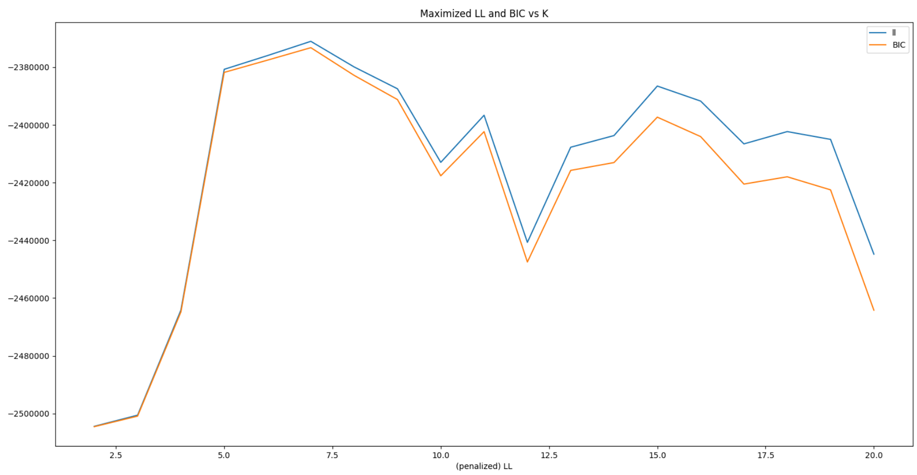

Figure 6: Graph of BIC and LL vs K

Question 7

(7.a)

When calling `print_clusters` with $K = 7$, I get 7 clusters that are grouped by the following features:

- age
- sex
- birthplace
- ancestry
- citizenship
- income
- education level
- employer

The clusters can be seen in the figure in the following page.
If we run the model for a $K = 10$, we can see that the clusters change a bit and become more specific. However, most of the clusters don't even change. The fact that many clusters stay the same, means that they are quite stable. One of the very few concepts added was "11th grade" for the educational level. It is clear that this is a bit too specific as can be easily inferred from the fact that we have more clusters.

```
Cluster 1:
  age: 65 and above
  sex: female
  birthplace: Europe
  ancestry1: Western Europe (except Spain)
  citizen: naturlized US citizen
  income: $1 - $14999
  edlevel: high school or ged
  employer: private, for profit

Cluster 2:
  age: 20 - 29
  sex: female
  birthplace: US
  ancestry1: Western Europe (except Spain)
  citizen: born in US
  income: $1 - $14999
  edlevel: high school or ged
  employer: private, for profit

Cluster 3:
  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

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

Cluster 5:
  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

Cluster 6:
  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

Cluster 7:
  age: 0 - 12
  sex: male
  birthplace: US
  ancestry1: Western Europe (except Spain)
  citizen: born in US
```

Figure 7: example cluster

(7.b)

When running the model several times for $K = 7$, we get a bit different clusters. More specific values such as age change quite a bit. However the clusters are not generally unstable. This change in the clusters (especially in age) is easily justifiable by the fact that there are more age groups than groups for other features.

(7.c)

In order to judge how similar two states are, we can just train our model on two different states and make decisions based on the outputted features, number of clusters etc. By comparing how siimlar the features and the cluster separations are, we can gauge how similar the two states are. Two states that have the same clusters would be very similar, but two stats that have neither the same number of clusters nor the same cluster features would be very dissimilar.