Sahil Gandhi

Programming Assignment – 03

CS 6140 – Christopher Amato

**Q 1.3. Part a. Vary the number of components in the mixture**

As we increase the number of components, it is likely that we over-fit (as a result, likelihood improves), but also that it takes longer (more iterations) to fit the data.

**Plots for 2 components (Plots highlighted with green denote good fit and performance):**

|  |  |
| --- | --- |
| #iteration: 19; LL: -66.051047361iter 19 ll -66.051047361 | #iterations 19; LL: -81.617821366 |
| #iterations 34; LL: -363.560476896 | #iterations 28; LL: -394.529578336 |
| #iterations 15; LL: -14.4307119353 | #iterations 14; LL: -71.4774039936 |
| #iterations 11; LL: -80.9103350356 | #iterations 26; LL: -307.483333353 |
| #iterations 33; LL: -140.876783795 | #iterations 34; LL: -164.39968094 |
| #iterations 65; LL: -862.909297002 | #iterations 36; LL: -794.191190387 |

**Plots for 3 components (Plots highlighted with green denote good fit and performance):**

|  |  |
| --- | --- |
| #iterations 37; LL: -43.351701311 | #iterations 18; LL: -57.9274609286 |
| #iterations 308; LL: -358.869594119 | #iterations 37; LL: -377.968408162 |
| #iterations 39; LL: 2.52406264387 | #iterations 16; LL: -68.3103520195 |
| #iterations 47; LL: -117.907343091 | #iterations 36; LL: -240.389847186 |
| #iterations 81; LL: -125.566433745 | #iterations 42; LL: -136.832909224 |
| #iterations 50; LL: -708.450860187 | #iterations 34; LL: -738.533191075 |

Below are 3 plots for number of components vs iterations and log-likelihoods. This bolsters our observation that increasing the number of components will give a better likelihood but is also likely to over-fit. More on this when we do cross-validation.

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |

**Q 1-3 Part b. Vary initial mixture parameters**

* The mean(s) is picked randomly at the beginning of the EM loop. Changing that will only affect the number of iterations the loop takes to converge.
* The mixture coefficients default to uniform distribution. They will adjust themselves even if we set them as something else. However, setting the coefficients of a component (say c1) to 0 will essentially mean that none of the data points belong to c1 and we will not see c1 on the plot.
* Below are plots of varying the covariance matrix between an identity matrix and a matrix containing all 1’s.
  + Changing the covariance matrix, results in a change for some datasets, but most stay relatively the same.

**Covariance = Identity matrix Covariance = Matrix with all 1’s**

|  |  |
| --- | --- |
| #iterations 37; LL: -43.351701311 | #iterations: 25; LL: -33.0258043445 |
| #iterations 308; LL: -358.869594119 | #iterations: 359; LL: -358.870271564 |
| #iterations 39; LL: 2.52406264387 | #iterations: 50; LL: -43.8076493039 |
| #iterations 47; LL: -117.907343091 | #iterations: 31; LL: -117.905186806 |
| #iterations 81; LL: -125.566433745 | #iterations: 77; LL: -125.566305874 |
| #iterations 50; LL: -708.450860187 | #iterations: 126; LL: -708.454871488 |

**Q2-3. Use K-Means to set the initial mixture parameters**

For all the below datasets, we do the following steps:

* Find the best K (number of clusters)
  + For this we plot k vs the K-Means objective and pick the k from the position of the ‘kink’ in the curve
* The K-Means cluster plot (next to the K-Means objective curve) is to visualize how the K-Means clusters are formed
* Using the centroids as mixture mean and setting the mixture coefficients as the distribution of K-Means predicted labels, we run the GMM on the same dataset both, full covariance matrix and diagonal covariance matrix.

**Data-1-small:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**Data-1-large:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**Data-2-small:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**Data-2-large:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**Data-3-small:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**Data-3-large:**

|  |  |
| --- | --- |
|  |  |
|  |  |

**Q3-1 Construct candidate models on \*\_small datasets and compare ranking of models on \*\_large datasets**

By choosing [1-5] number of clusters and 2 types of covariance matrices, we have 10 models to run each dataset on. The models are described as follows:

* Model 1; #Clusters: 1; Covariance Type: full
* Model 2; #Clusters: 1; Covariance Type: diag
* Model 3; #Clusters: 2; Covariance Type: full
* Model 4; #Clusters: 2; Covariance Type: diag
* Model 5; #Clusters: 3; Covariance Type: full
* Model 6; #Clusters: 3; Covariance Type: diag
* Model 7; #Clusters: 4; Covariance Type: full
* Model 8; #Clusters: 4; Covariance Type: diag
* Model 9; #Clusters: 5; Covariance Type: full
* Model 10; #Clusters: 5; Covariance Type: diag

Analysis of each dataset is split into 4 plots.

1. Represents computing Gaussian mixtures and their respective log-likelihoods for both small & large datasets and plotted together for comparison
2. Represents a training-testing approach, where the mixture parameters are computed using the small dataset, and then prediction membership probabilities for the large dataset are computed. The log-likelihood for both is plotted for comparison
3. Plot A and B are analyzed and the best model is run on the small dataset
4. Plot A and B are analyzed and the best model is run on the large dataset

**Data 1:** Best model is model #8

|  |  |
| --- | --- |
| Plot A | Plot B |
| Plot C | Plot D |

**Data-2:** Best model is model #8

|  |  |
| --- | --- |
| Plot A | Plot B |
| Plot C | Plot D |

**Data-3:** Best model is model #9

|  |  |
| --- | --- |
| Plot A | Plot B |
| Plot C | Plot D |