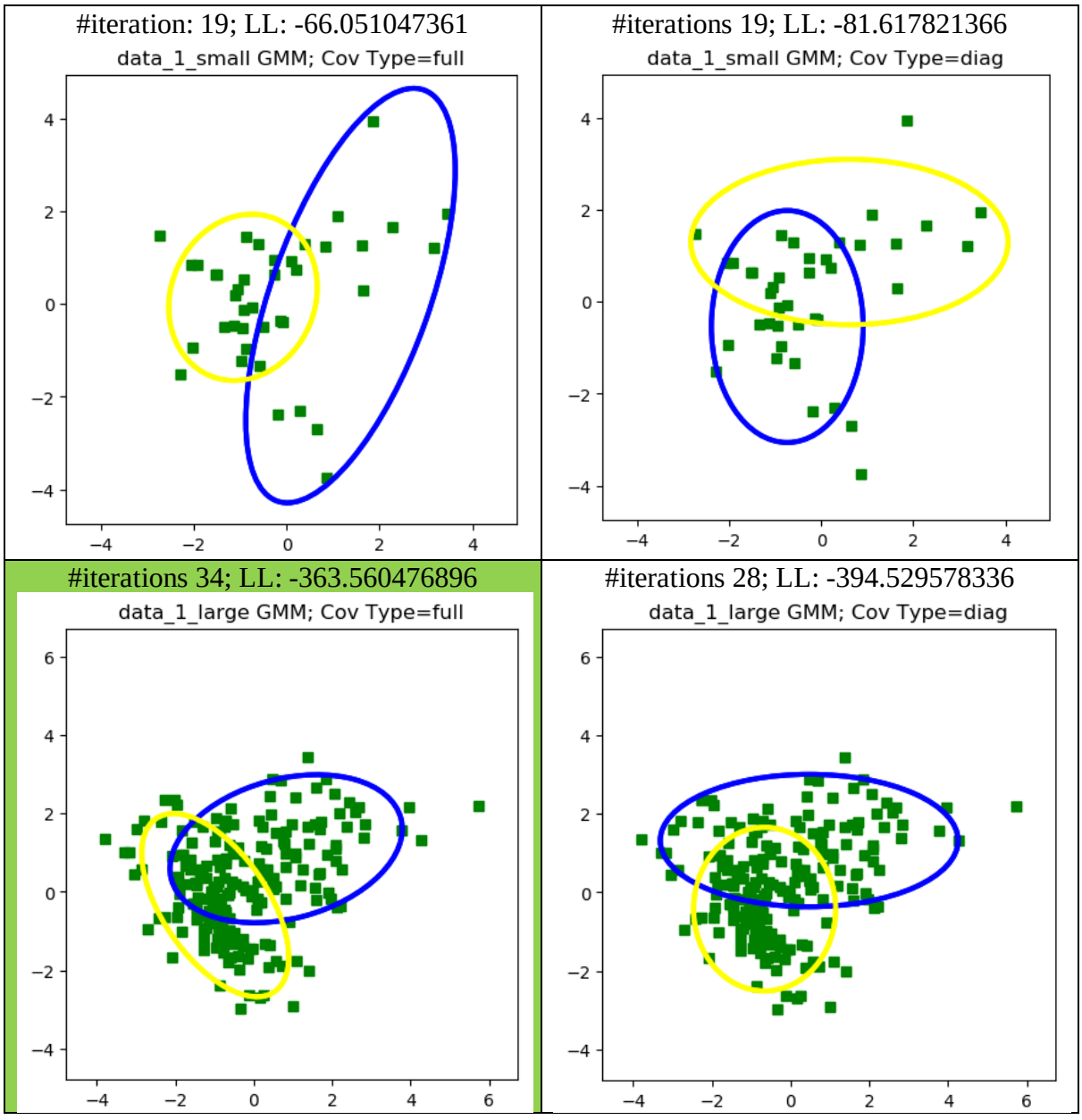
Sahil Gandhi
Programming Assignment – 03
CS 6140 – Christopher Amato

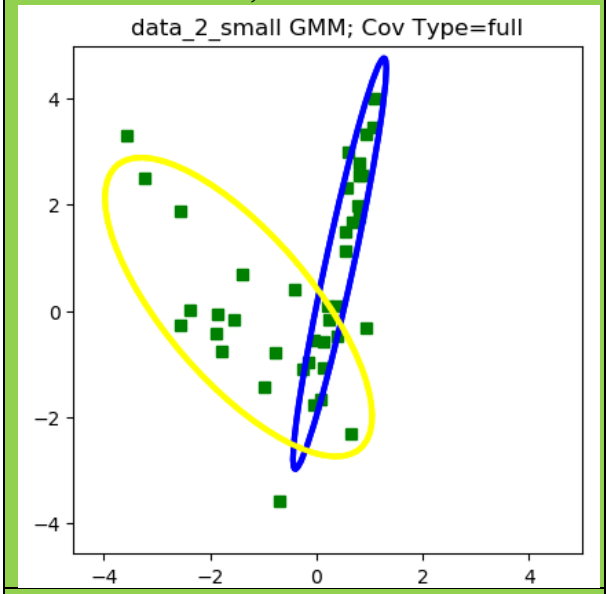
Q 1.3. Part a. Vary the number of components in the mixture

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.

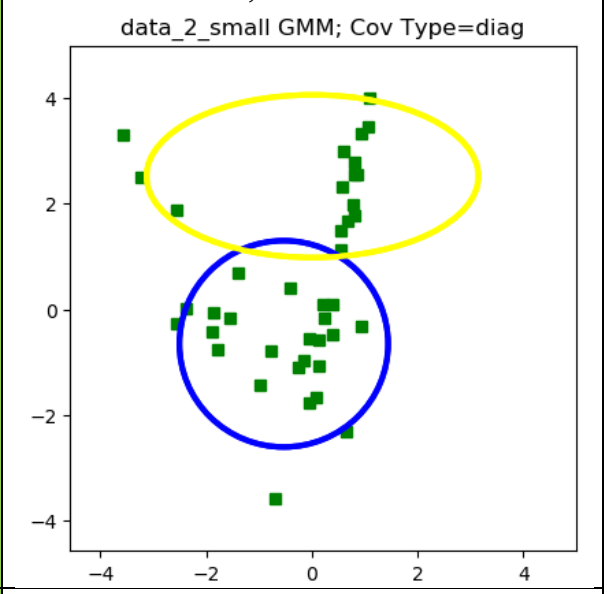
Plots for 2 components (Plots highlighted with green denote good fit and performance):



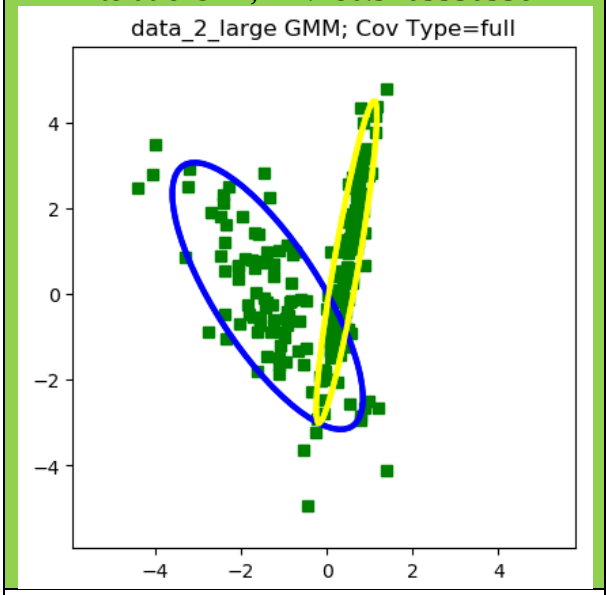
#iterations 15; LL: -14.4307119353



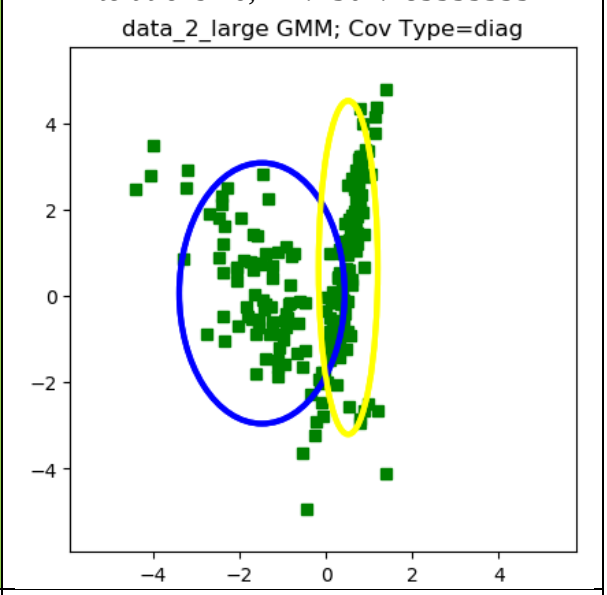
#iterations 14; LL: -71.4774039936



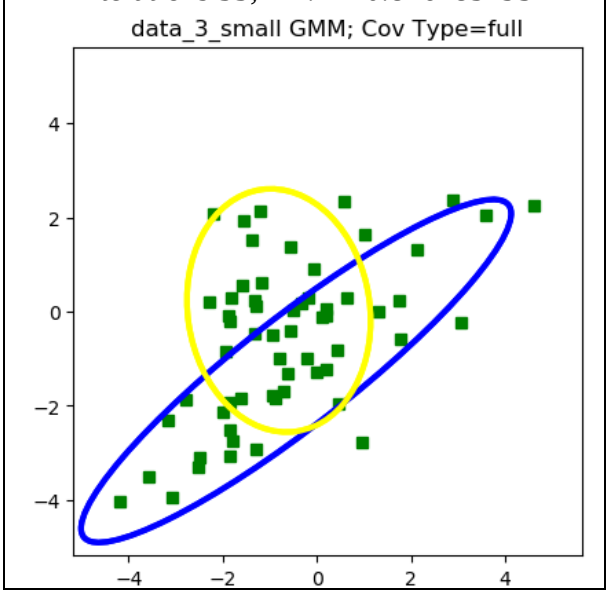
#iterations 11; LL: -80.9103350356



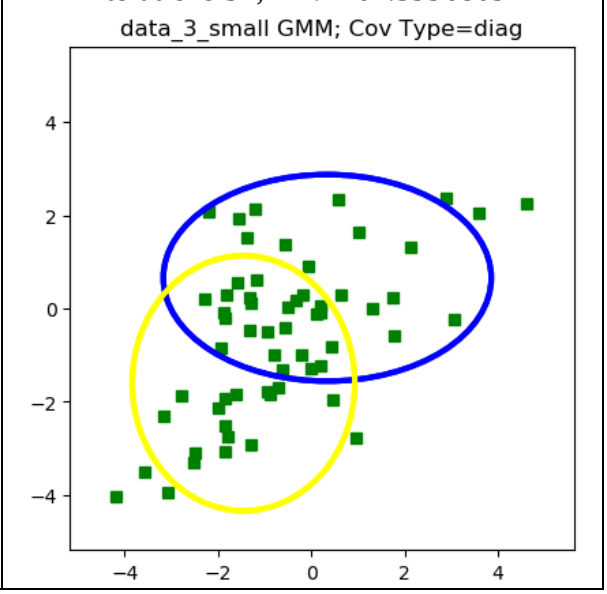
#iterations 26; LL: -307.483333353

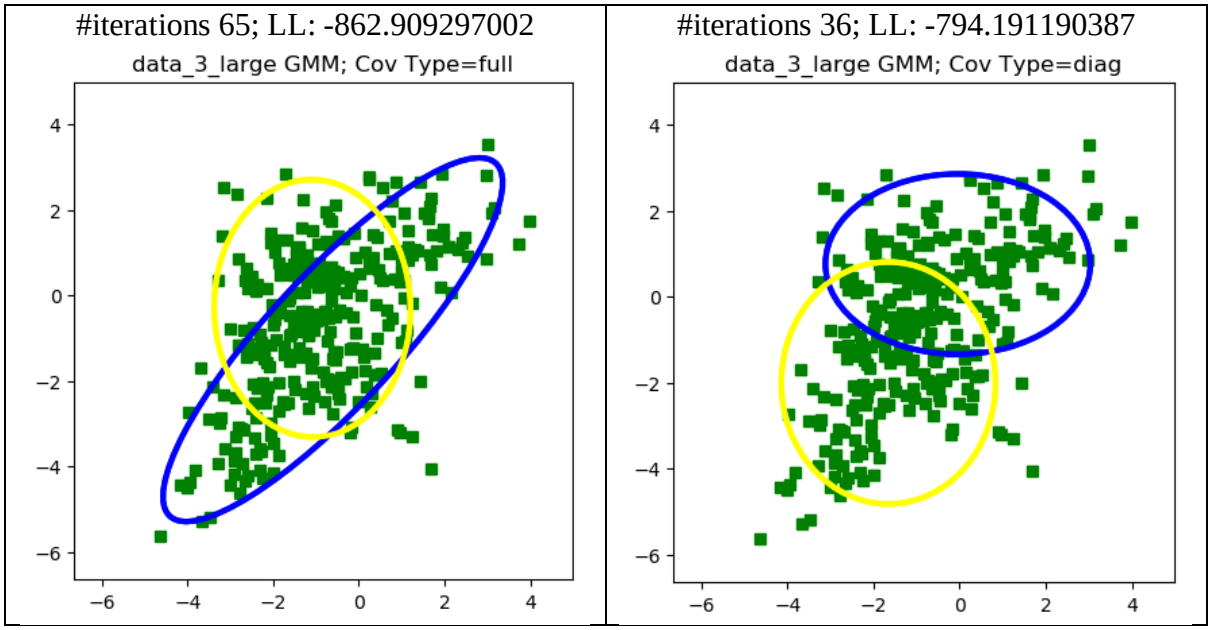


#iterations 33; LL: -140.876783795

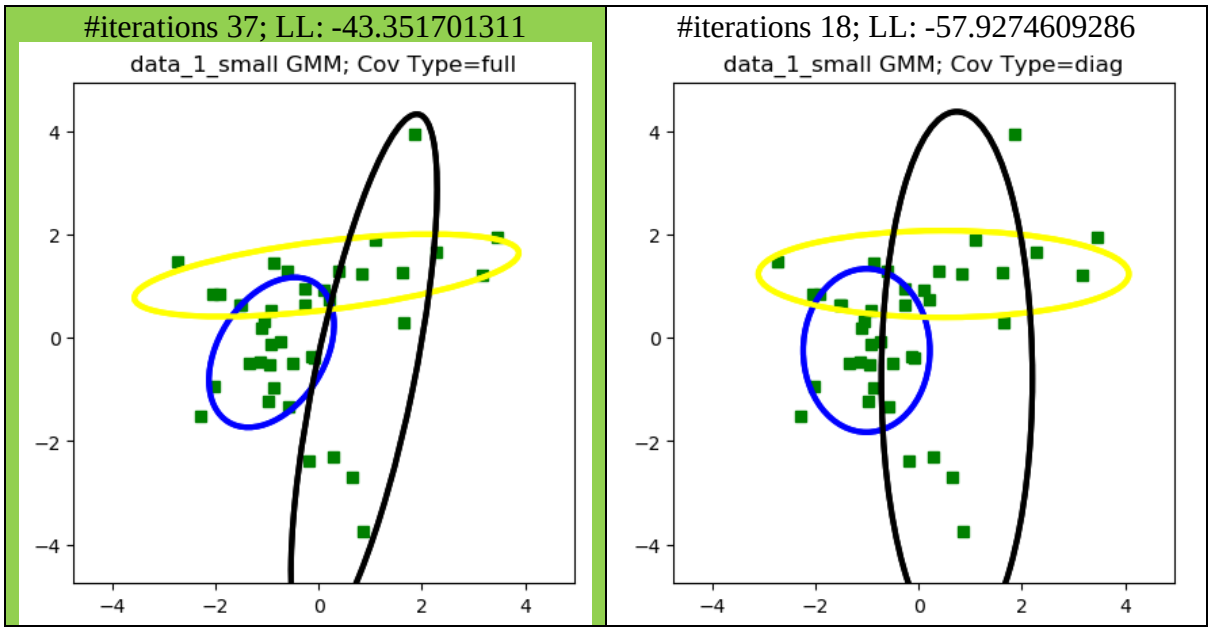


#iterations 34; LL: -164.39968094

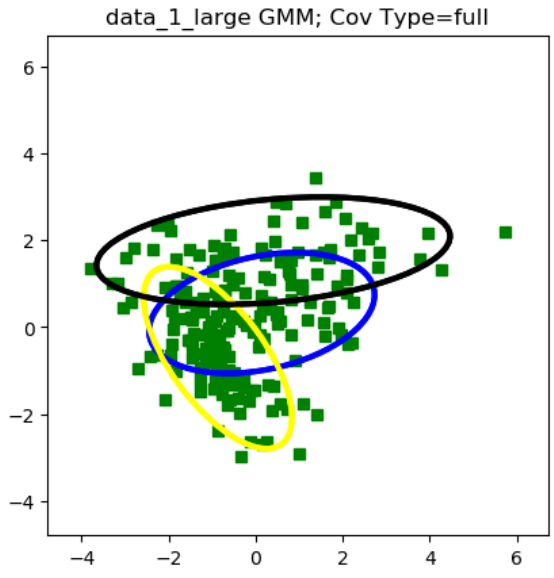




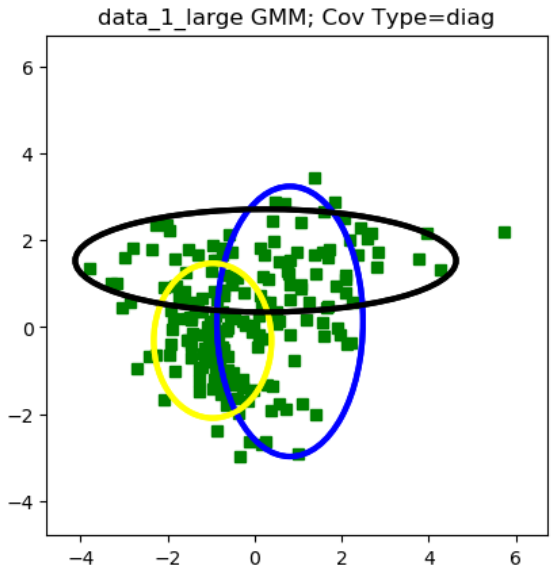
Plots for 3 components (Plots highlighted with green denote good fit and performance):



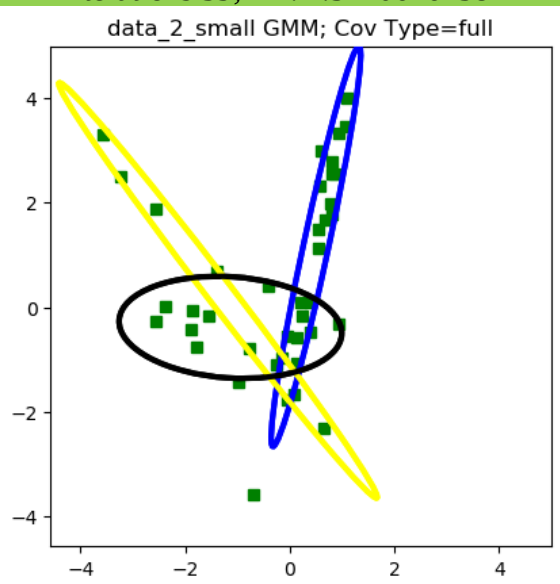
#iterations 308; LL: -358.869594119



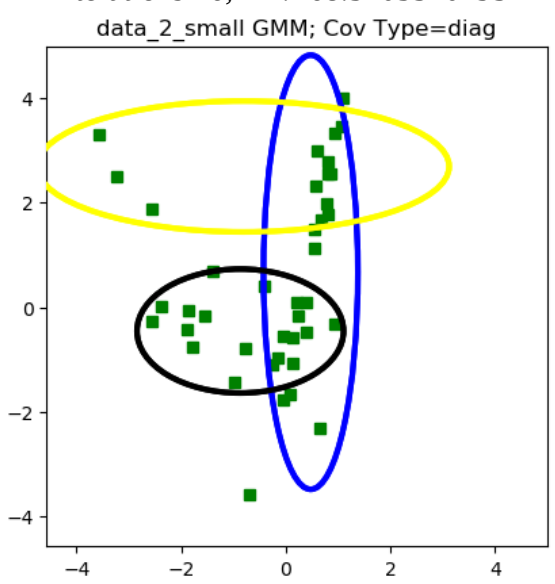
#iterations 37; LL: -377.968408162



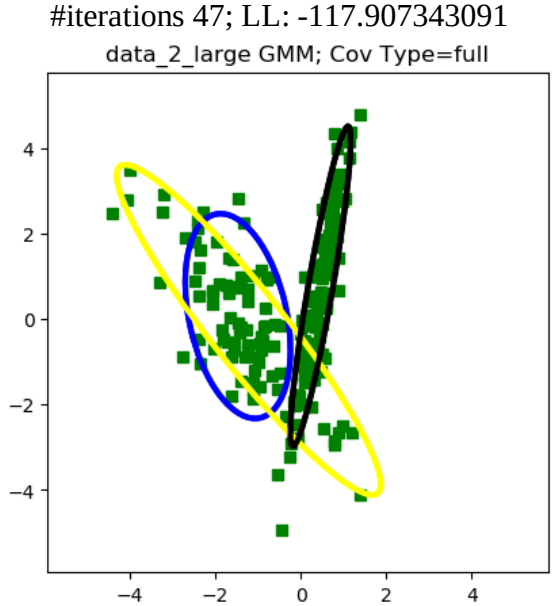
#iterations 39; LL: 2.52406264387



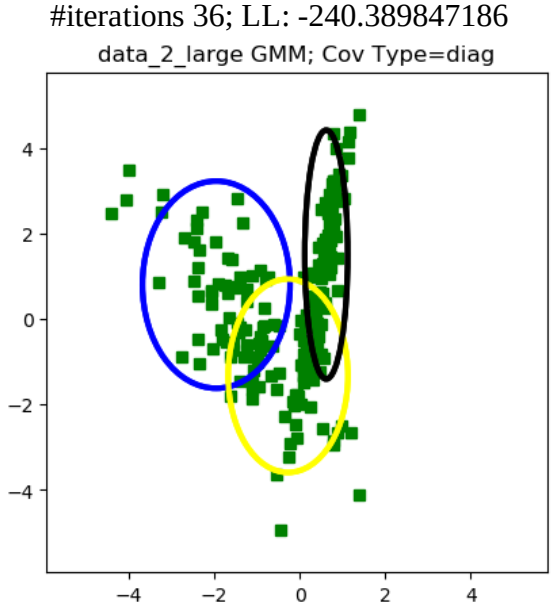
#iterations 16; LL: -68.3103520195



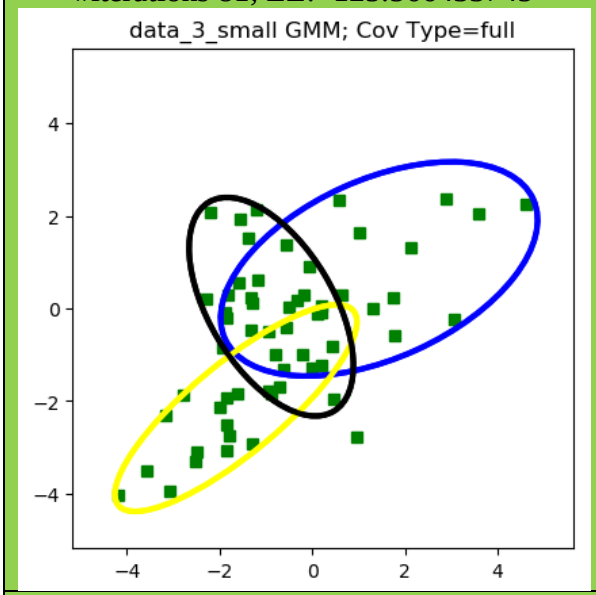
#iterations 47; LL: -117.907343091



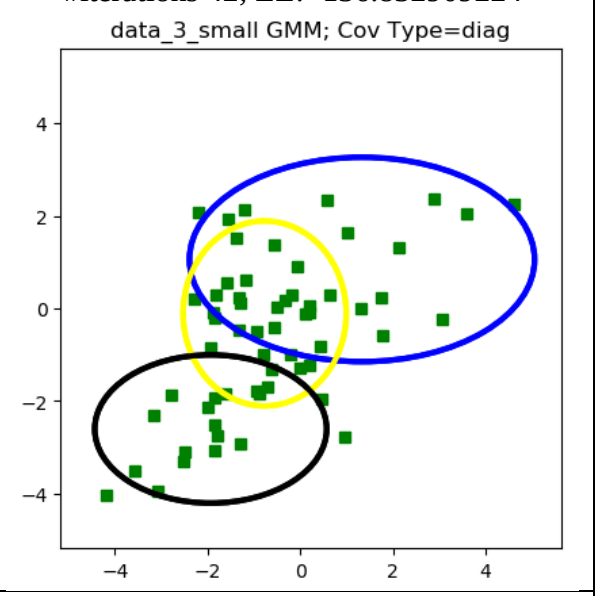
#iterations 36; LL: -240.389847186



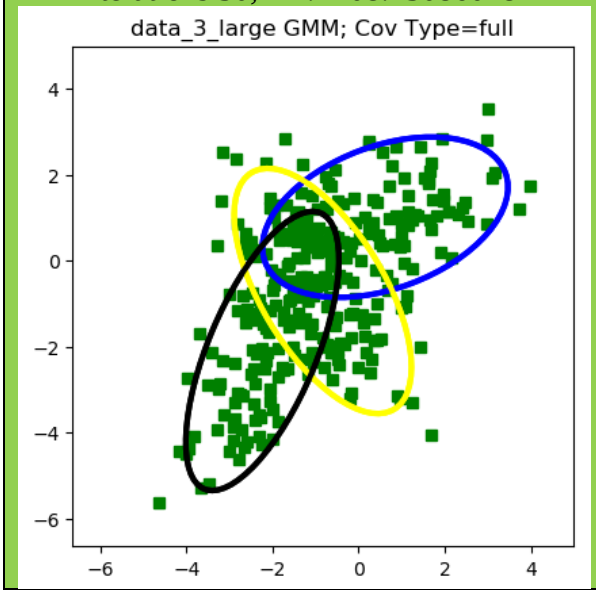
#iterations 81; LL: -125.566433745



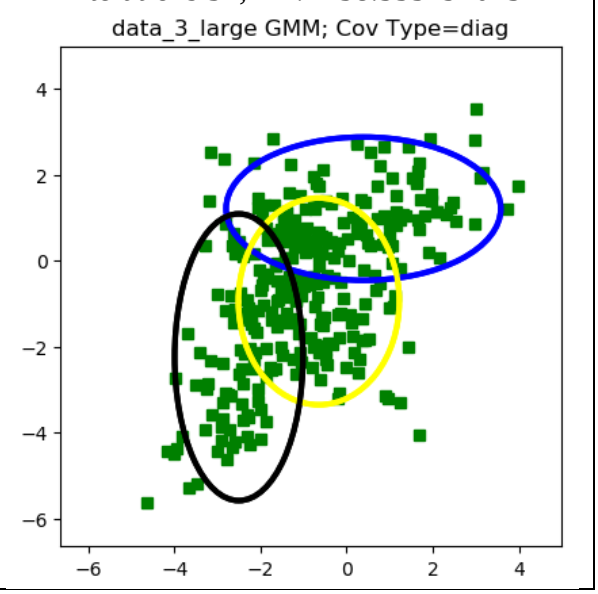
#iterations 42; LL: -136.832909224



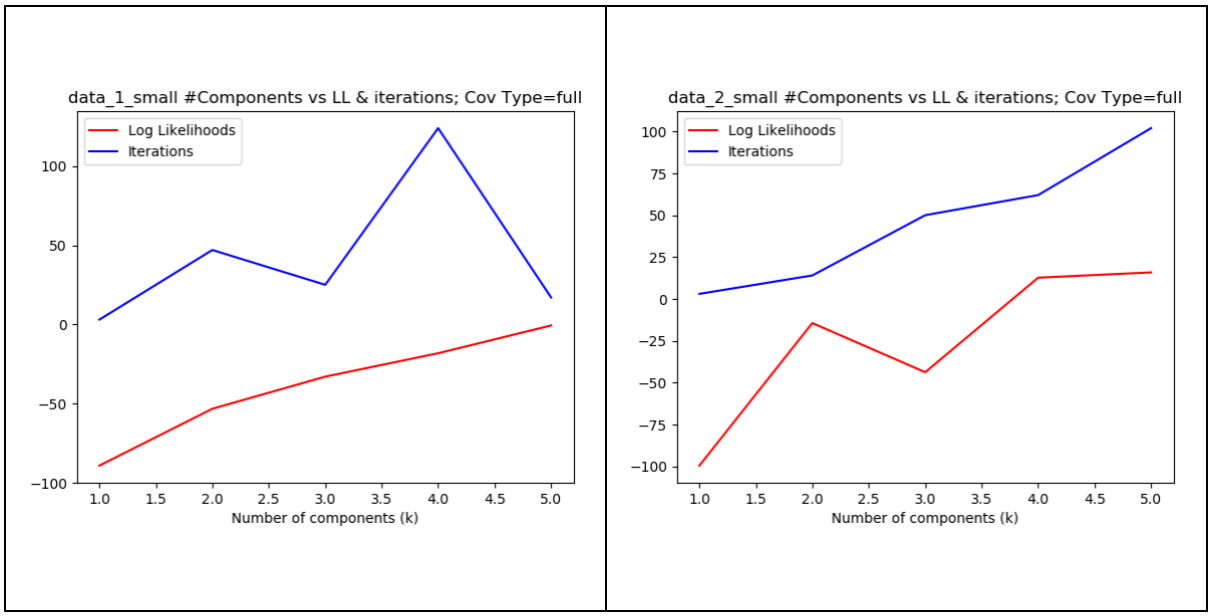
#iterations 50; LL: -708.450860187

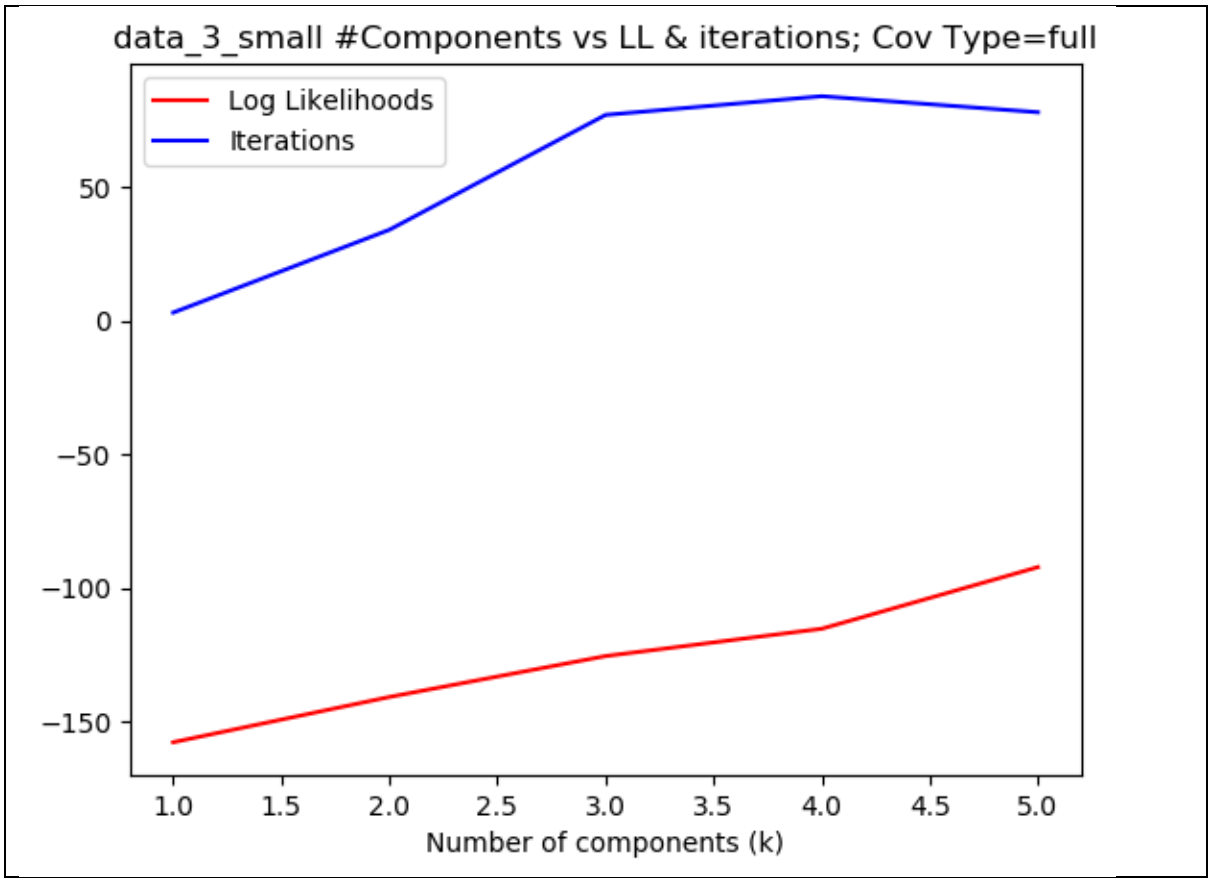


#iterations 34; LL: -738.533191075



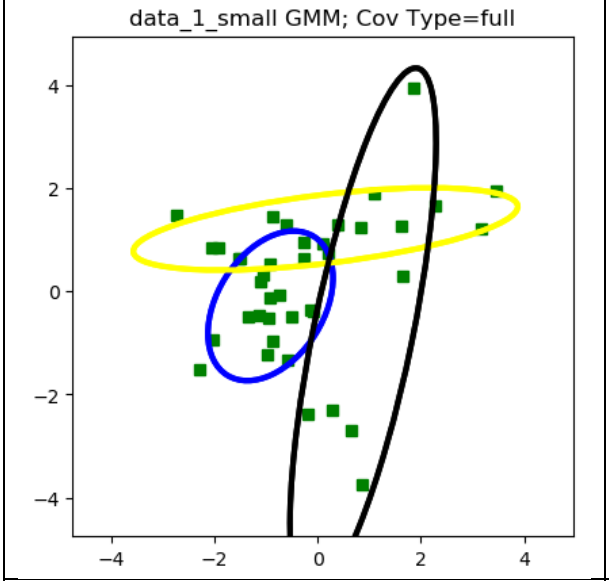
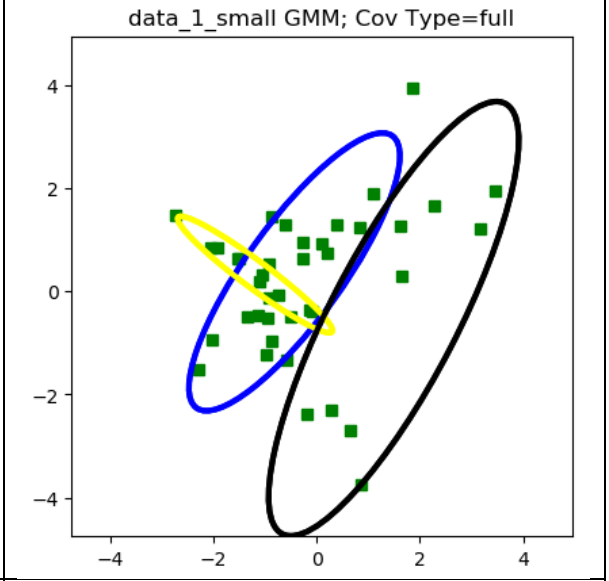
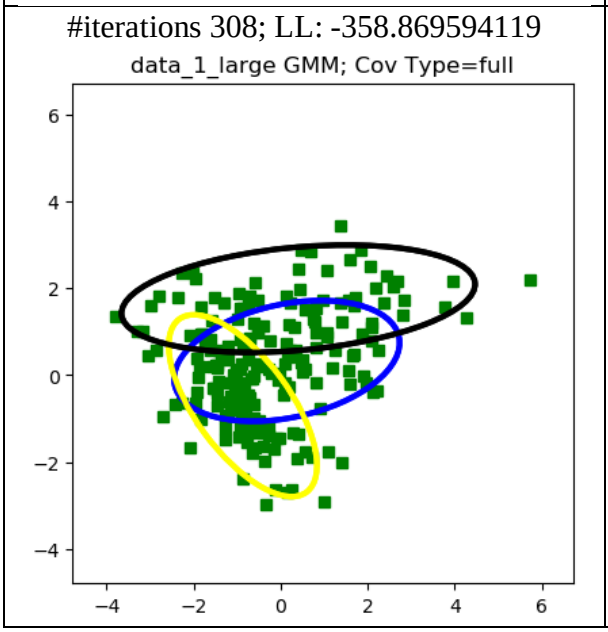
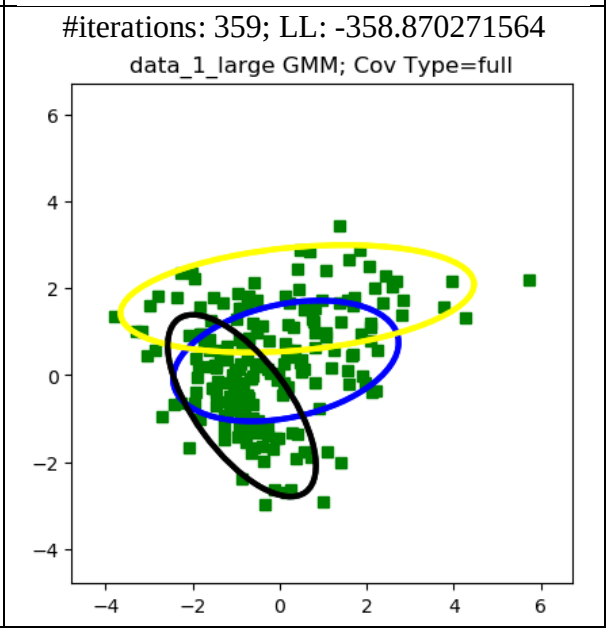
Below are 3 plots for number of components vs iterations and log-likelihoods. This bolsters our observation that increasing the number of components will give a better likelihood but is also likely to over-fit. More on this when we do cross-validation.



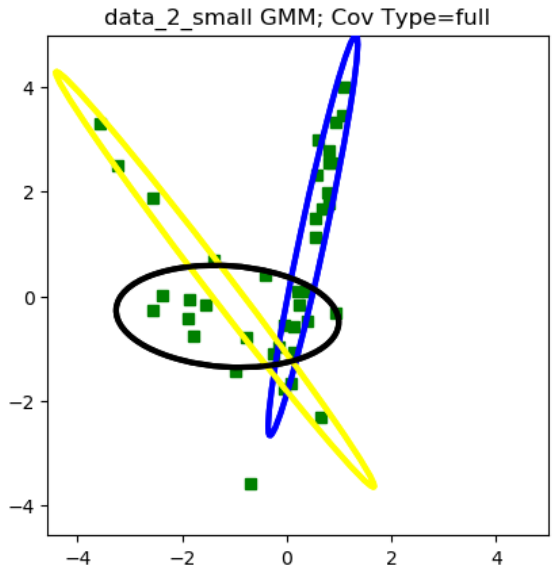


Q 1-3 Part b. Vary initial mixture parameters

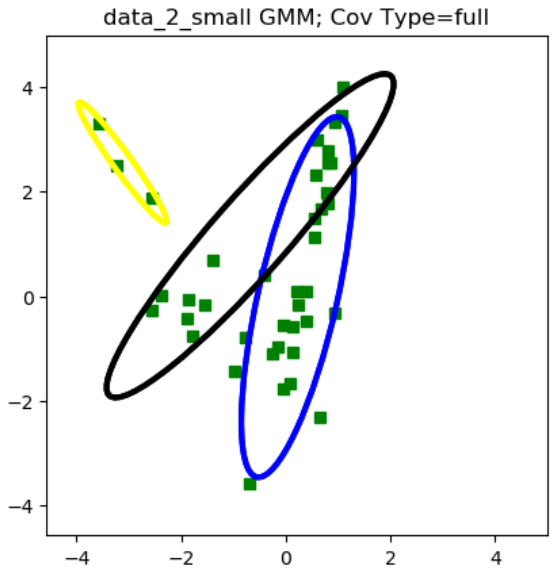
- The mean(s) is picked randomly at the beginning of the EM loop. Changing that will only affect the number of iterations the loop takes to converge.
- The mixture coefficients default to uniform distribution. They will adjust themselves even if we set them as something else. However, setting the coefficients 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
	

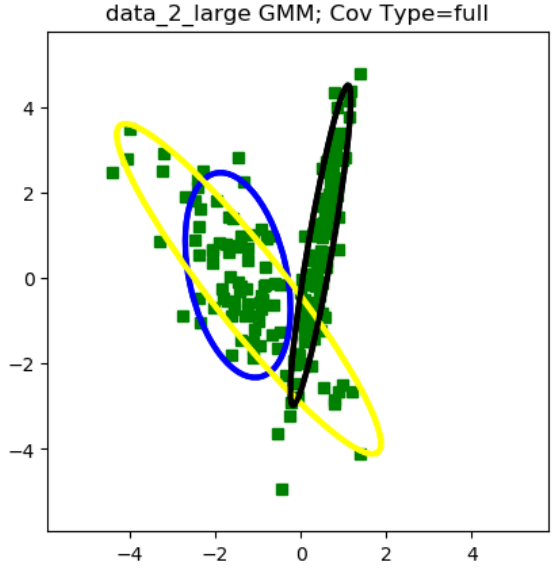
#iterations 39; LL: 2.52406264387



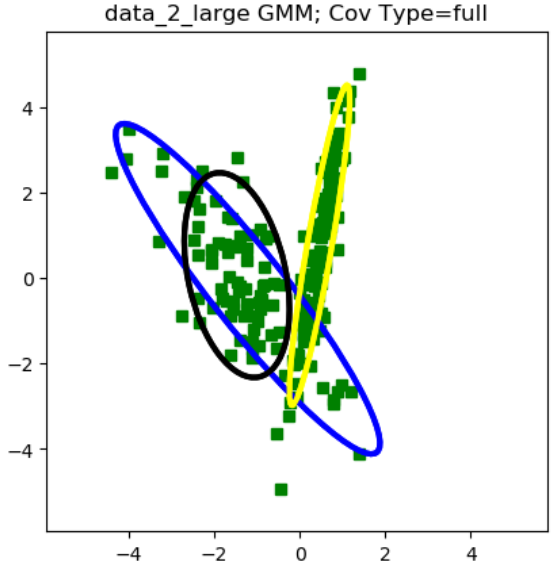
#iterations: 50; LL: -43.8076493039



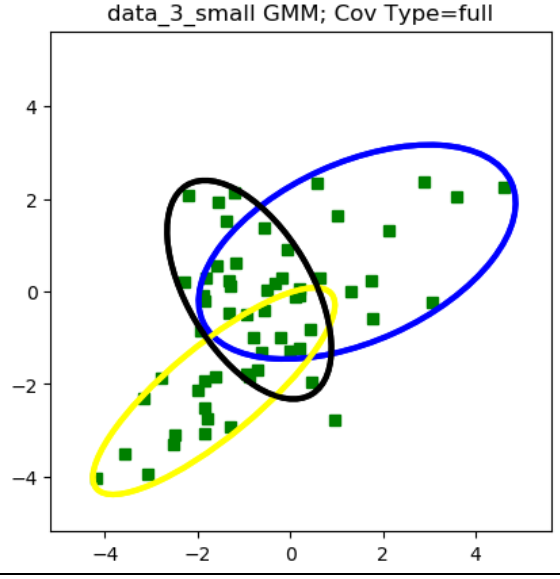
#iterations 47; LL: -117.907343091



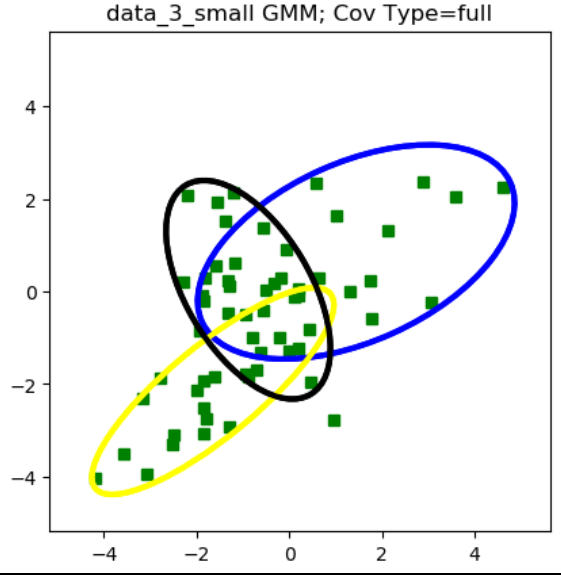
#iterations: 31; LL: -117.905186806

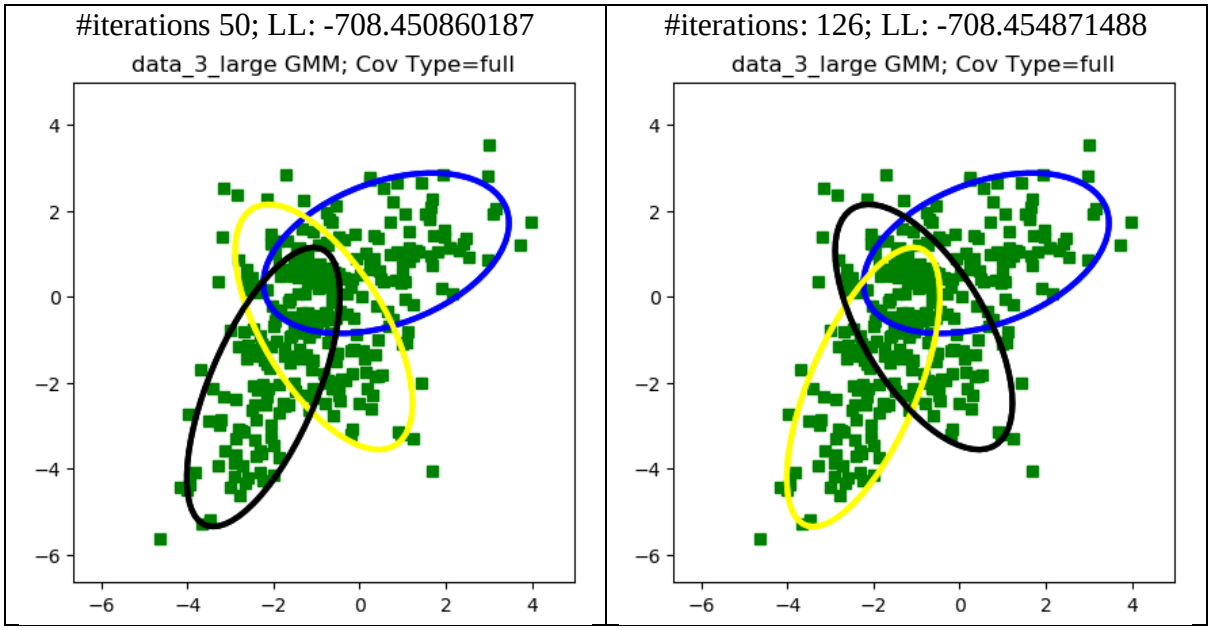


#iterations 81; LL: -125.566433745



#iterations: 77; LL: -125.566305874





Varying the convergence threshold has the obvious effect that the mixtures don't converge to the optima and are have a lower likelihood.

Q2-1. Explain the difference between the diagonal and full covariance matrices

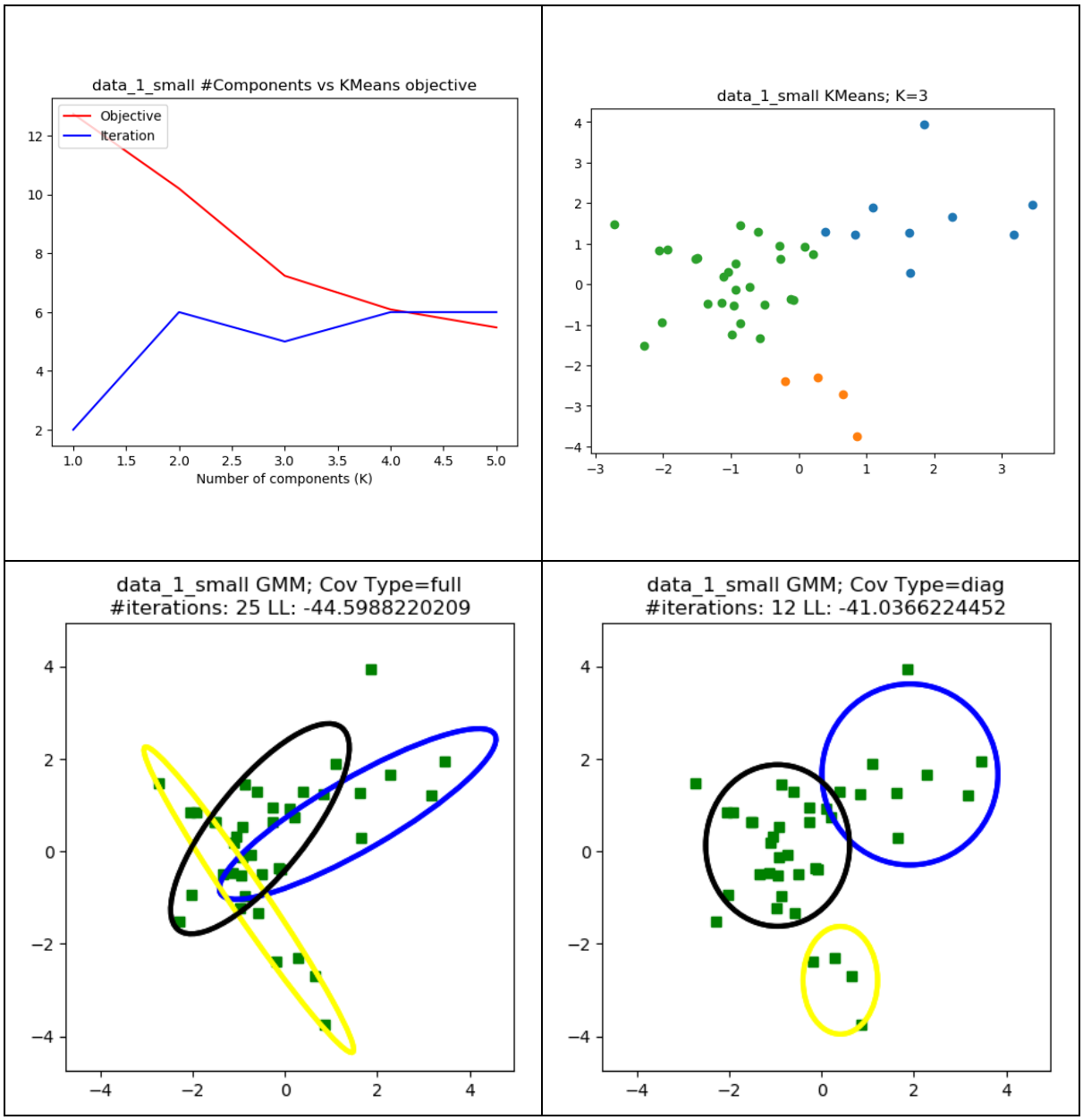
Covariance matrices, by definition signify the correlation between points X and Y. The elements on the diagonal signify the covariance of each variable with itself, i.e. variance of that variable. Hence, when used in conjunction with Gaussian Mixture Models, the version with full covariance matrix can be approximated that by using a diagonal matrix, when the variables become grow independent from each other. Using the diagonal covariance matrix has geometric implications that the Gaussians always align the axes, because they assume independence. But, the important point that because the component Gaussian are acting together to model the overall feature density, full covariance matrices are not necessary even if the features are not statistically independent. The linear combination of diagonal covariance basis Gaussians is capable of modelling the correlations between feature vector elements. The effect of using a set of M full covariance matrix Gaussians can be equally obtained by using a larger set of diagonal covariance Gaussians.

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

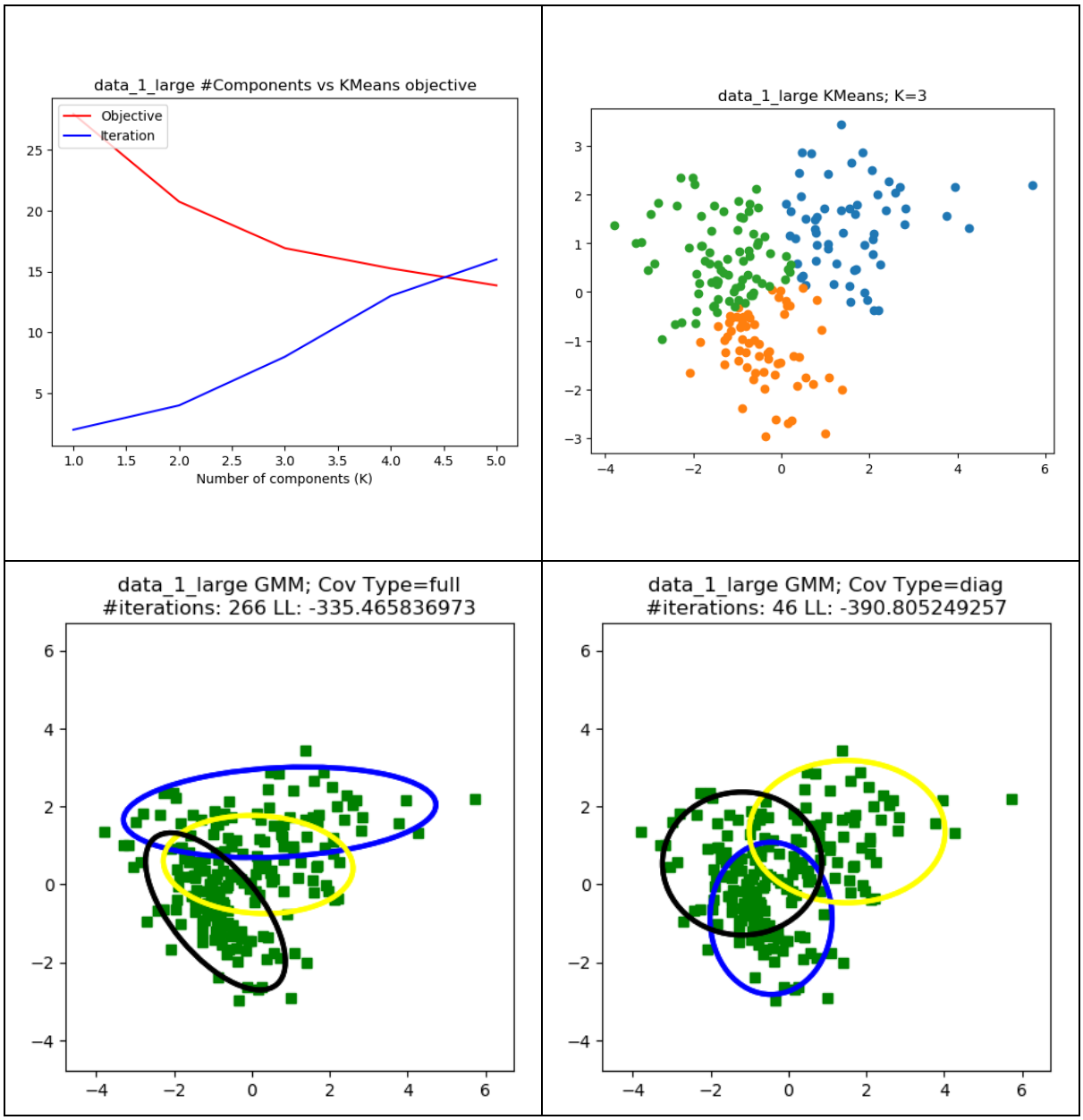
For all the below datasets, we do the following steps:

- 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.

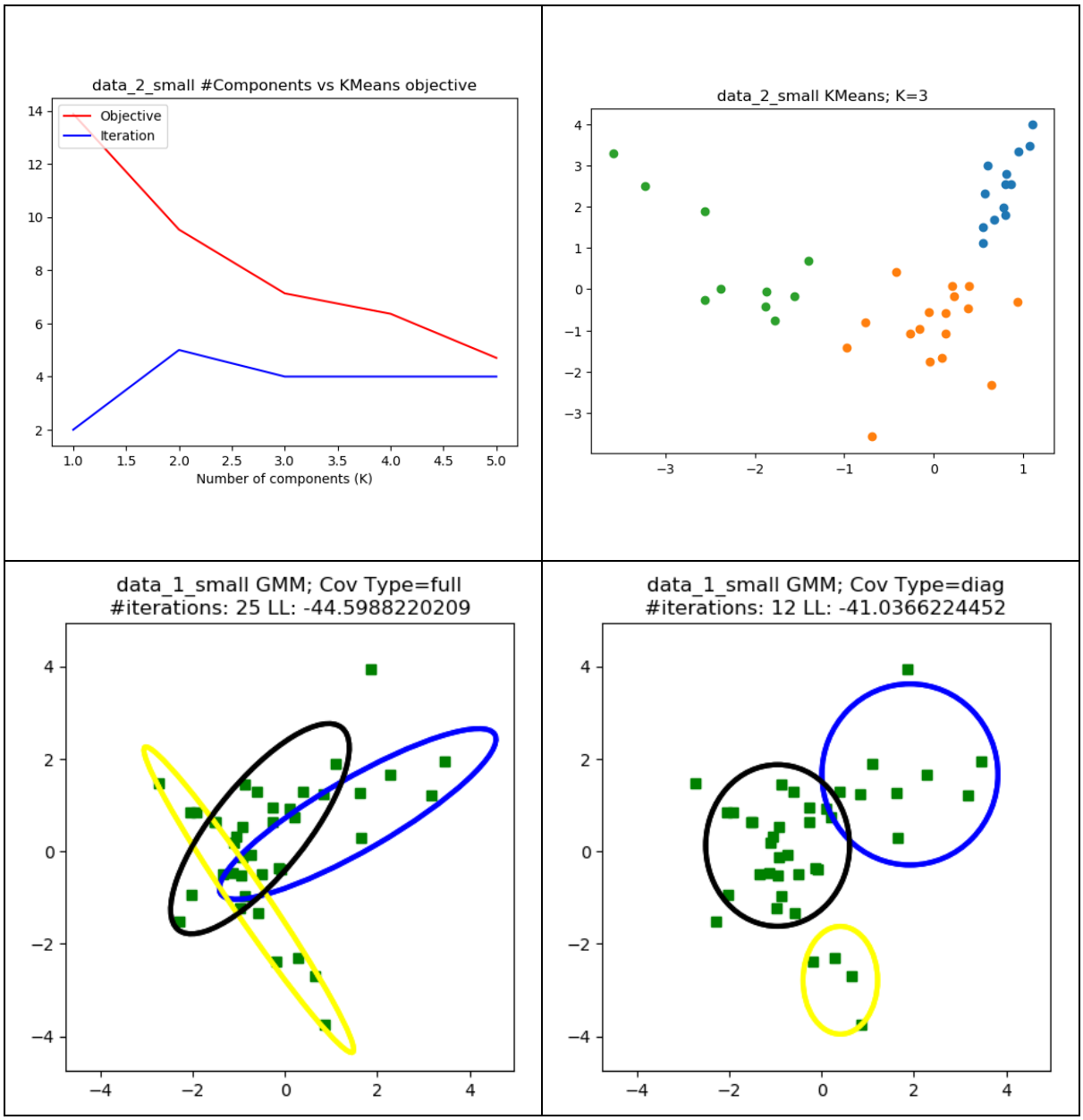
Data-1-small:


data_1_small #Components vs KMeans objective
Objective
Iteration
Number of components (K)
data_1_small KMeans; K=3
data_1_small GMM; Cov Type=full
#iterations: 25 LL: -44.5988220209
data_1_small GMM; Cov Type=diag
#iterations: 12 LL: -41.0366224452

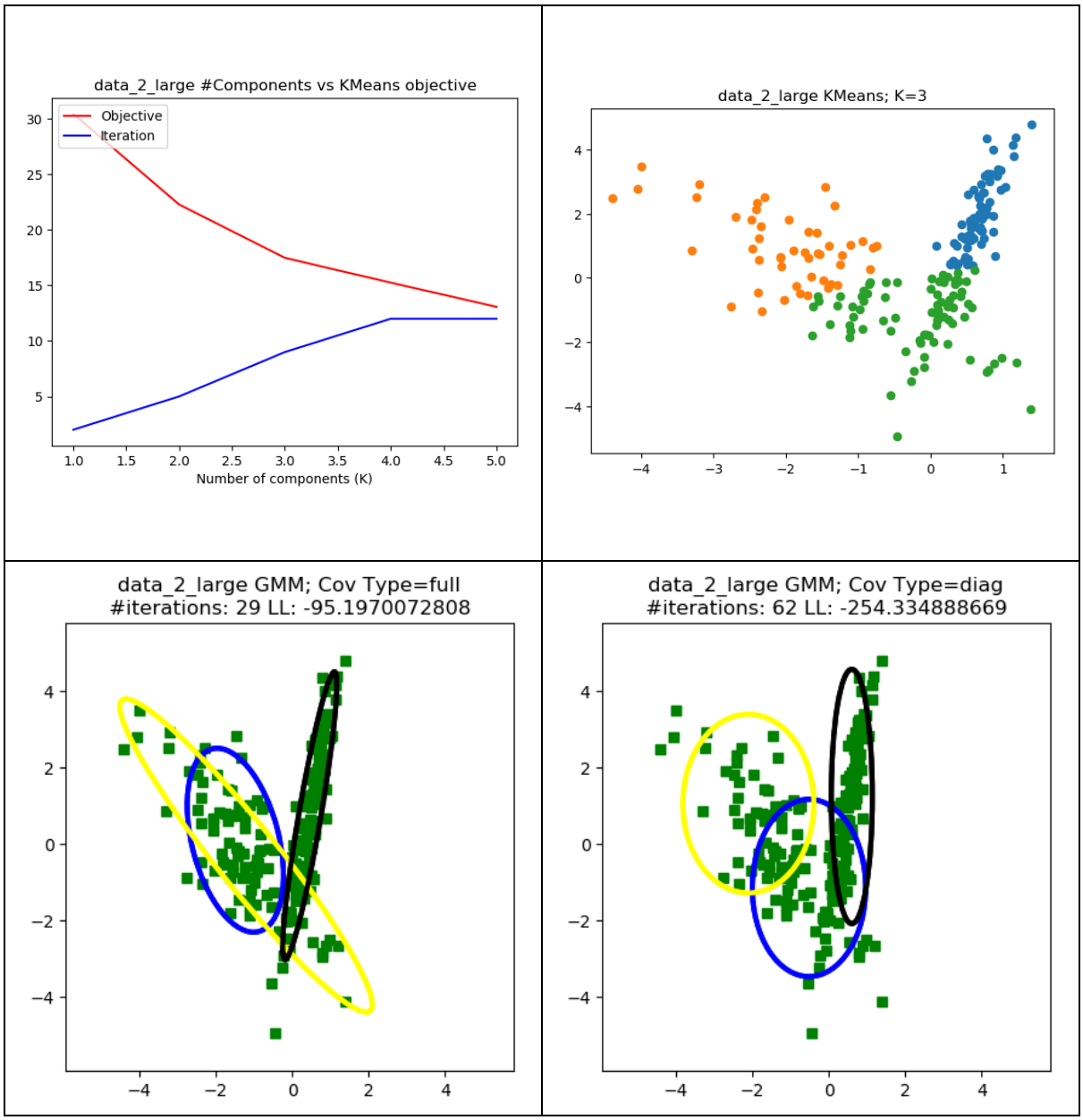
Data-1-large:


data_1_large #Components vs KMeans objective
Objective
Iteration
Number of components (K)
data_1_large KMeans; K=3
data_1_large GMM; Cov Type=full
#iterations: 266 LL: -335.465836973
data_1_large GMM; Cov Type=diag
#iterations: 46 LL: -390.805249257

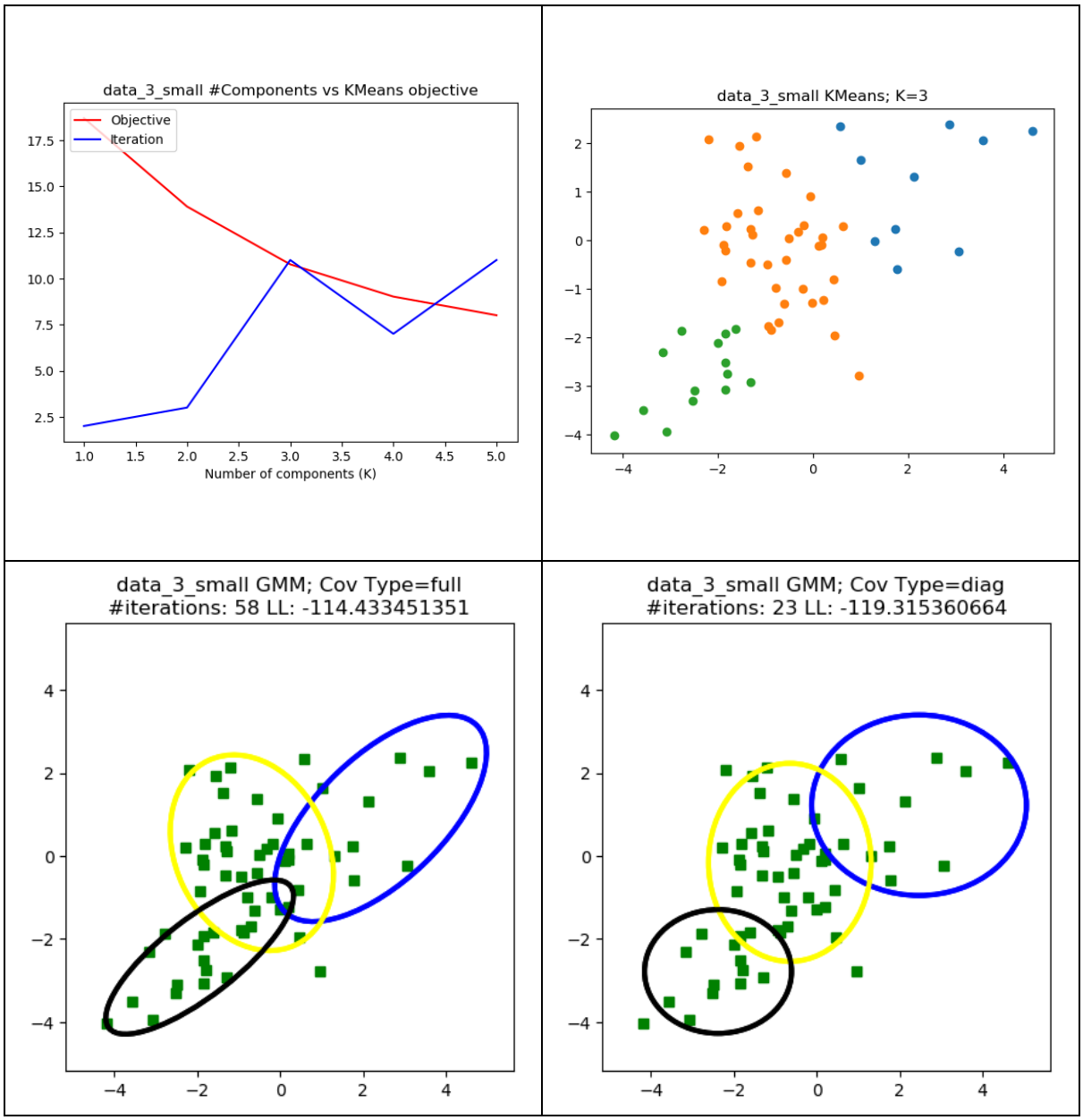
Data-2-small:


data_2_small #Components vs KMeans objective
Objective
Iteration
Number of components (K)
data_2_small KMeans; K=3
data_1_small GMM; Cov Type=full
#iterations: 25 LL: -44.5988220209
data_1_small GMM; Cov Type=diag
#iterations: 12 LL: -41.0366224452

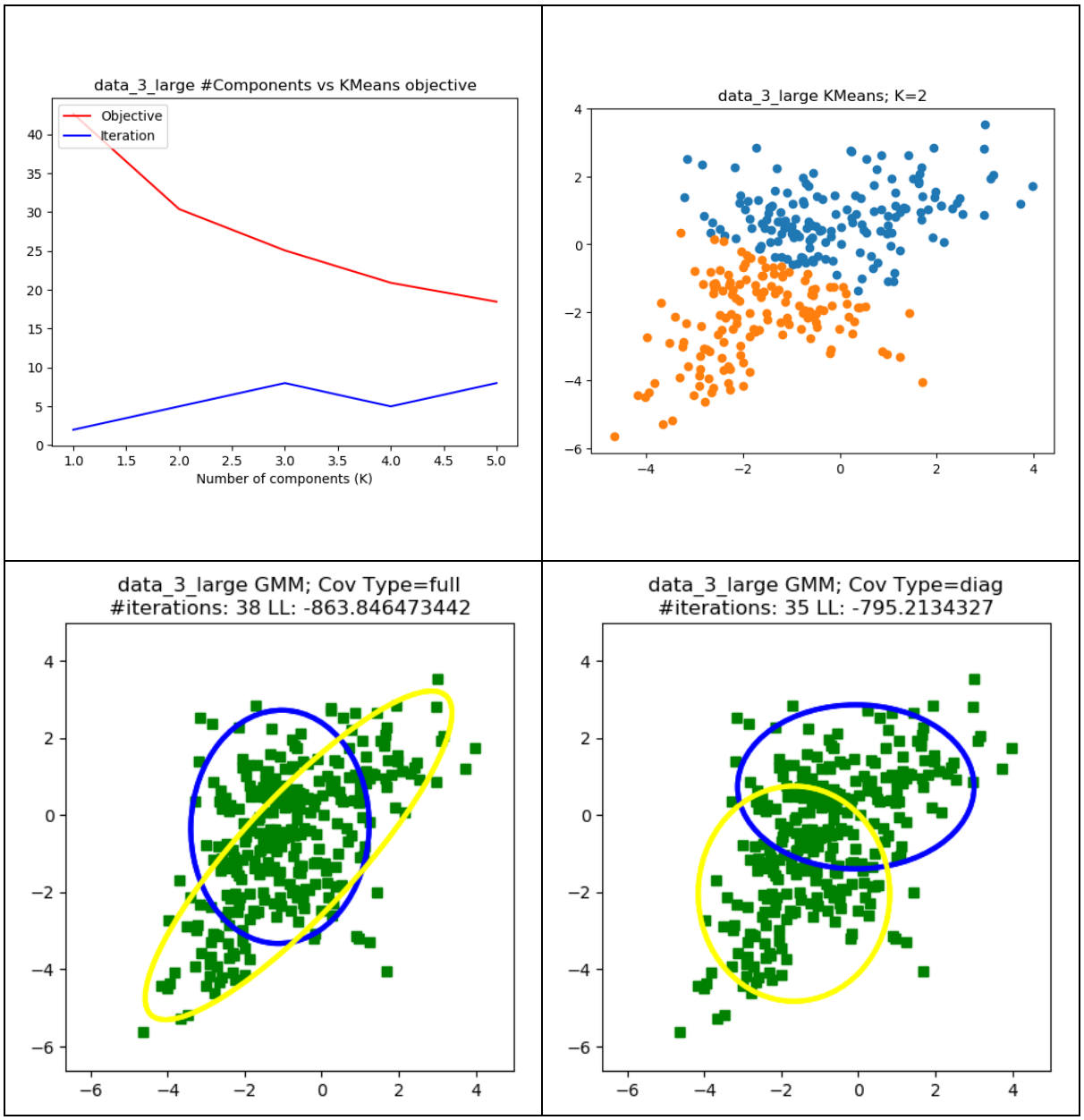
Data-2-large:


data_2_large #Components vs KMeans objective
Objective
Iteration
Number of components (K)
data_2_large KMeans; K=3
data_2_large GMM; Cov Type=full
#iterations: 29 LL: -95.1970072808
data_2_large GMM; Cov Type=diag
#iterations: 62 LL: -254.334888669

Data-3-small:


data_3_small #Components vs KMeans objective
Objective
Iteration
Number of components (K)
data_3_small KMeans; K=3
data_3_small GMM; Cov Type=full
#iterations: 58 LL: -114.433451351
data_3_small GMM; Cov Type=diag
#iterations: 23 LL: -119.315360664

Data-3-large:



The above plots make sense,

- Setting parameters using K-Means works better than a random selection, as K-Means (inherently) works towards finding dense clusters in the data.
- Also, the plots signify that setting parameters from K-Means works better on larger datasets than small datasets
    - This again, makes sense as GMM works better on larger datasets and struggles to find a good fit on small datasets.

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

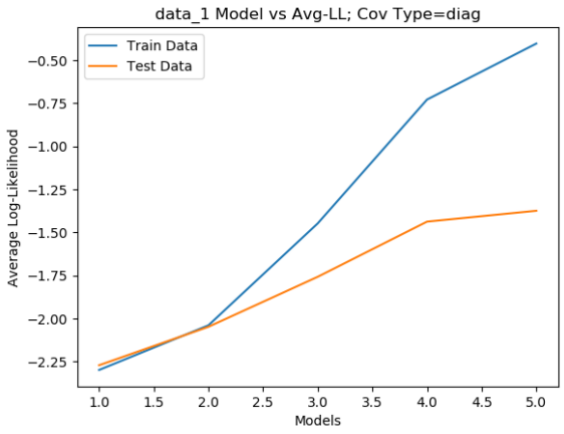
By choosing [1-5] number of clusters and 2 types of covariance matrices, we have 10 models to run each dataset on.

Analysis of each dataset is split into 4 plots.

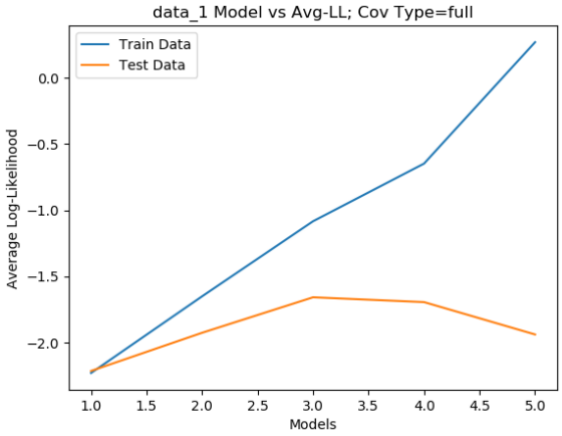
A. Represents training the small data using diagonal covariance matrix model for various number of components and testing the model on the large data set
B. Represents training the small data using full covariance matrix model for various number of components and testing the model on the large data set
C. Plot A and B are analyzed, and the best model is run on the small dataset
D. Plot A and B are analyzed, and the best model is run on the large dataset

Data 1: Best model is k=3 and covariance type = full

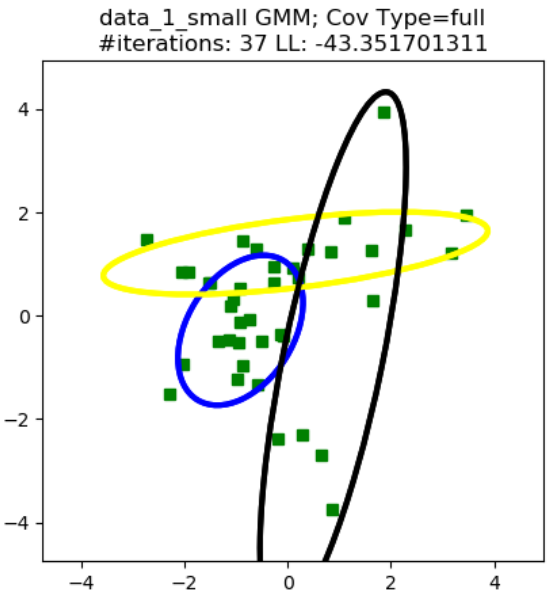
Plot A



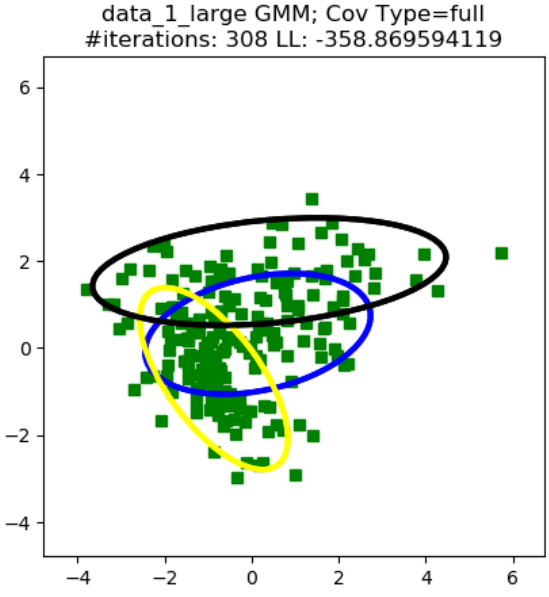
Plot B



Plot C

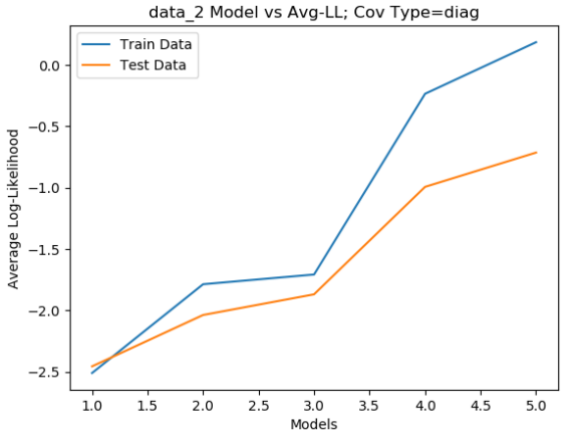


Plot D

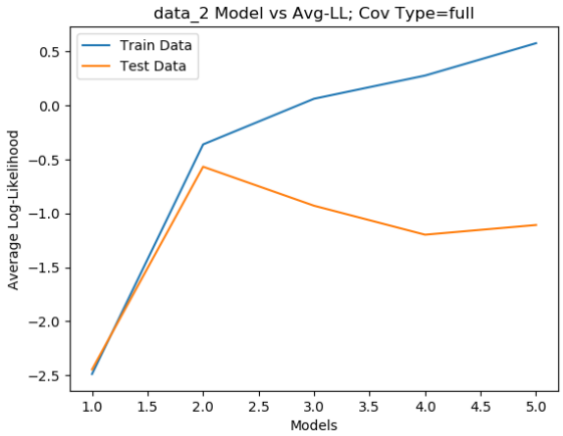


Data-2: Best model is k=2 and covariance type = full

Plot A

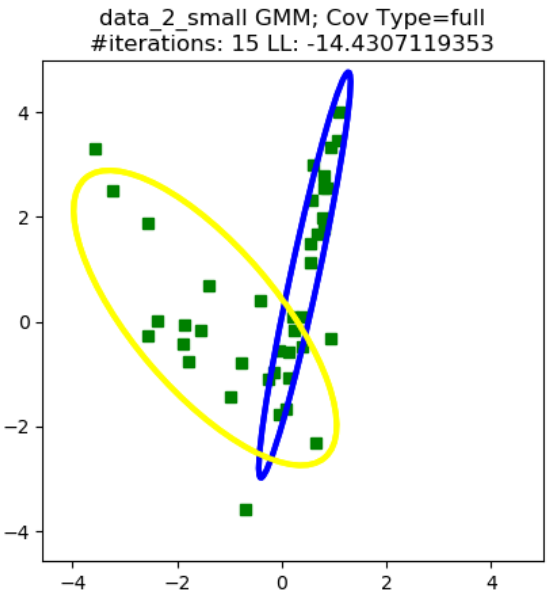


Plot

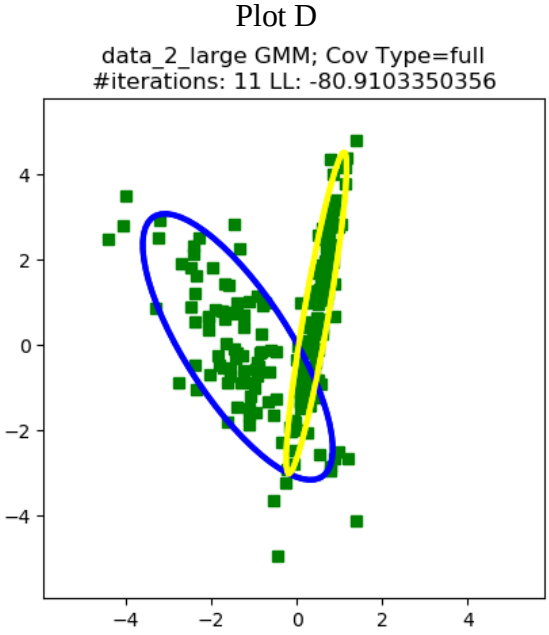


B

Plot C

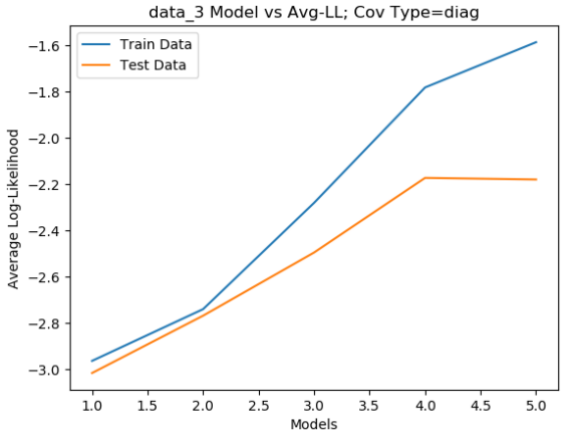


Plot D

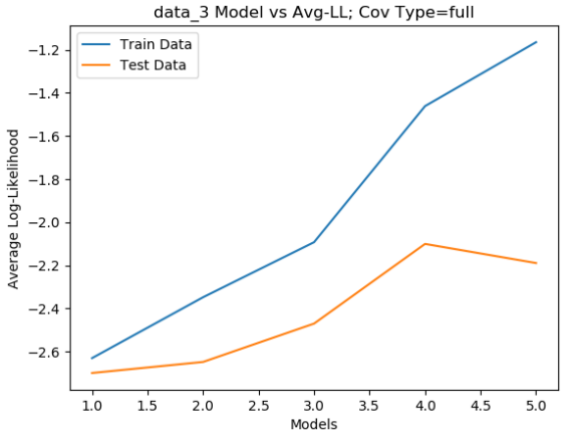


Data-3: Best model is k=3 and covariance type = full

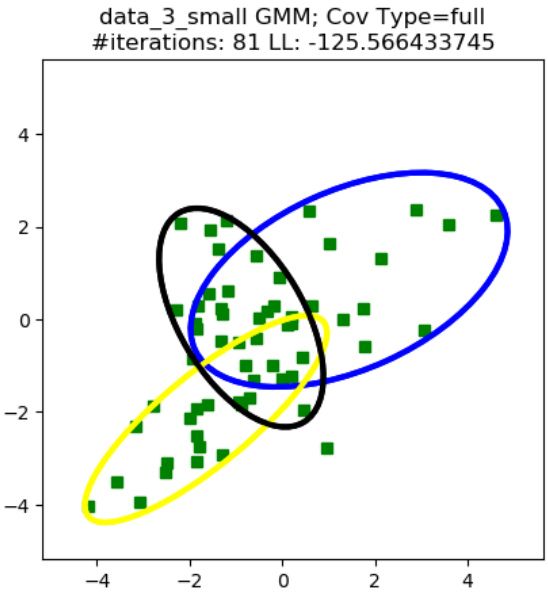
Plot A



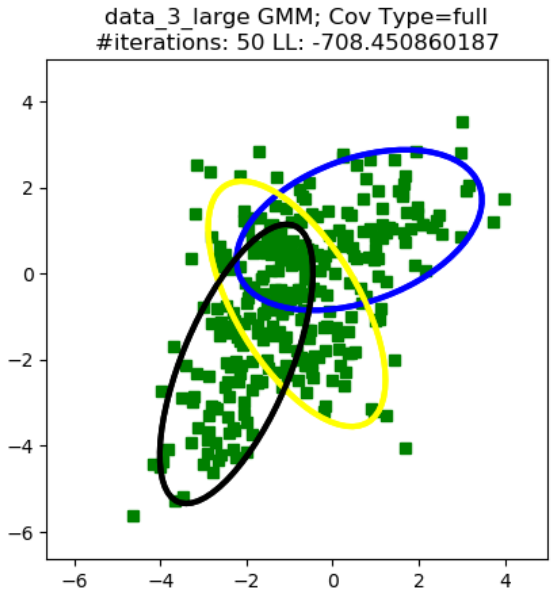
Plot B



Plot C



Plot D



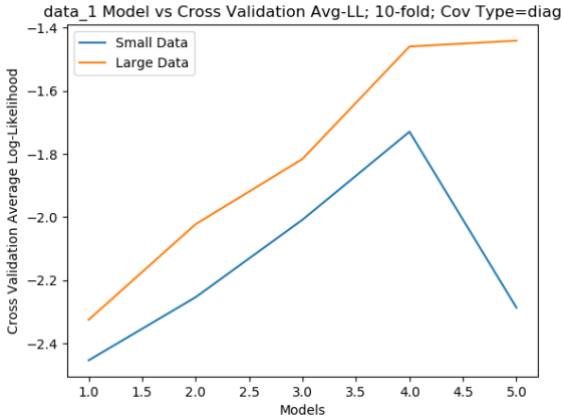
Q3-3 Test the cross-validation procedure. (a) Compare small and large datasets, (b) Compare results with that of Q3-1

We use the same set of models (as used in Q3-1) to test the cross-validation procedure:
By choosing [1-5] number of clusters and 2 types of covariance matrices, we have 10 models to run each dataset on.

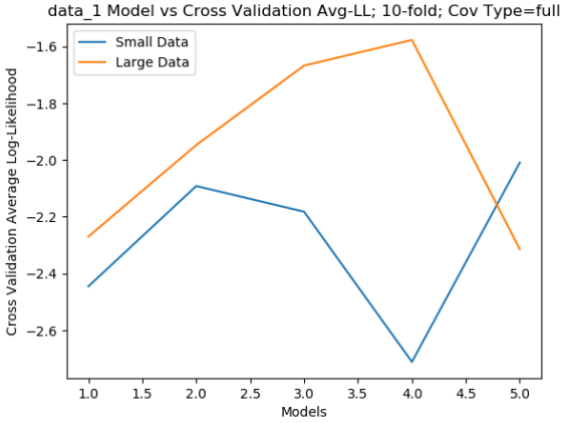
Part (a)
Again, we split the analysis of each dataset into 4 plots.
Analysis of each dataset is split into 4 plots.
Plots A, B, C, D represents training the cross validated *_small data using diagonal/full covariance matrix model for various number of components and testing the model on the large data set.
Plots E and F are the GMM output for the best model chosen based on the above plots

Data 1: Best model is k=2 and full

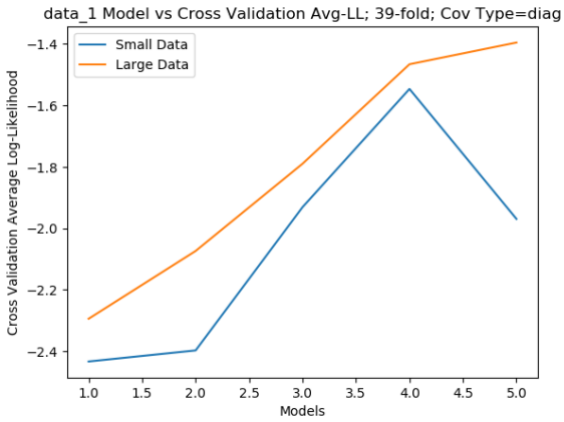
Plot A



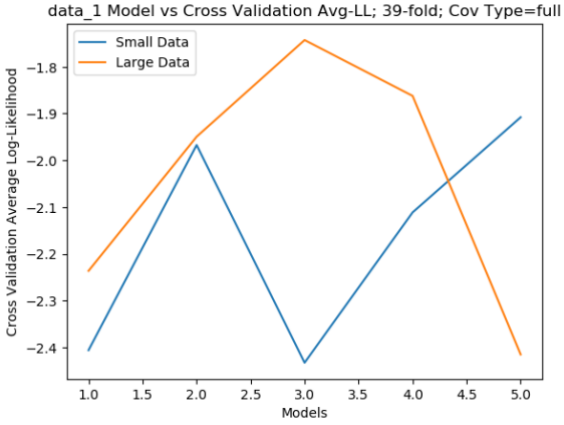
Plot B



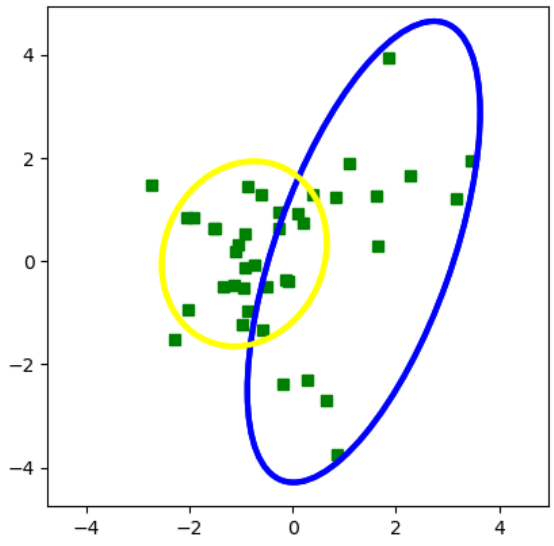
Plot C



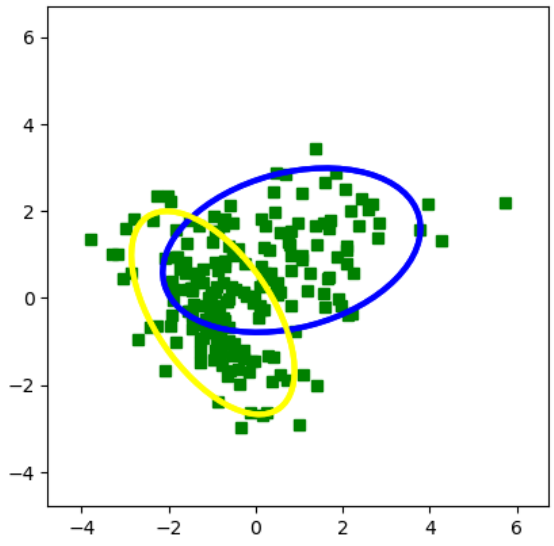
Plot D



Plot E

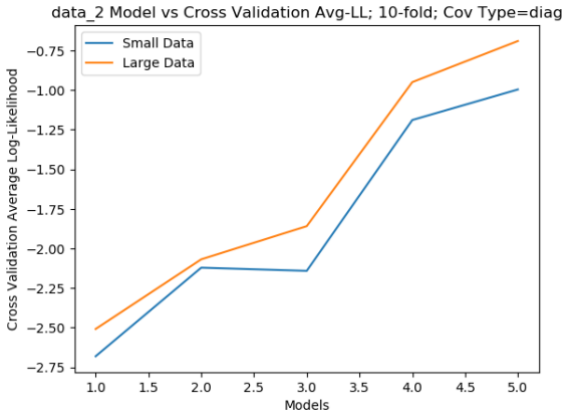

data_1_small GMM; Cov Type=full
#iterations: 19 LL: -66.051047361
4
2
0
−2
−4
−4
−2
0
2
4

Plot F

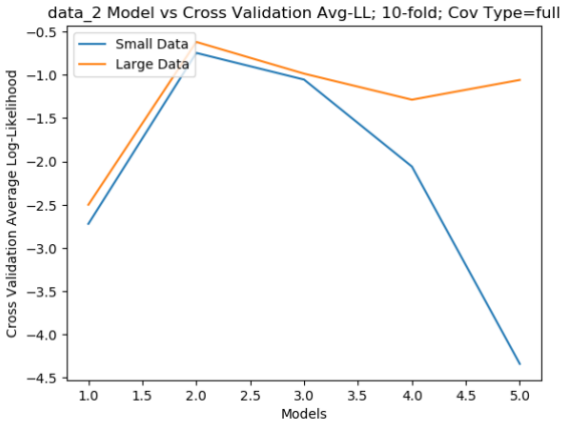

data_1_large GMM; Cov Type=full
#iterations: 34 LL: -363.560476896
6
4
2
0
−2
−4
−4
−2
0
2
4
6

Data-2: Best model is is k=2 and full

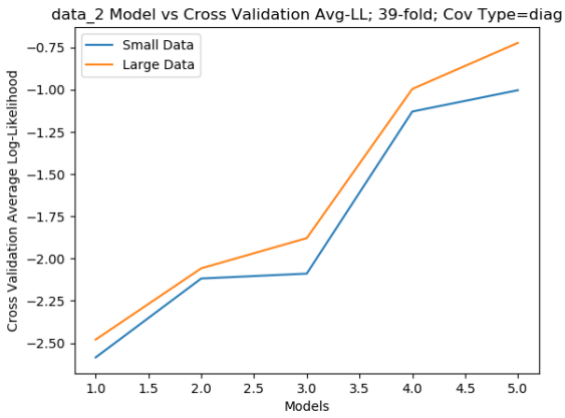
Plot A



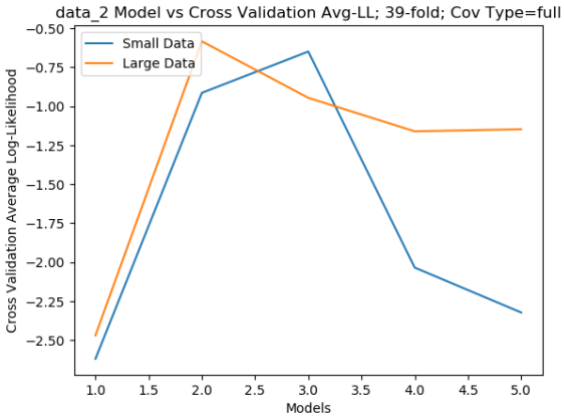
Plot B



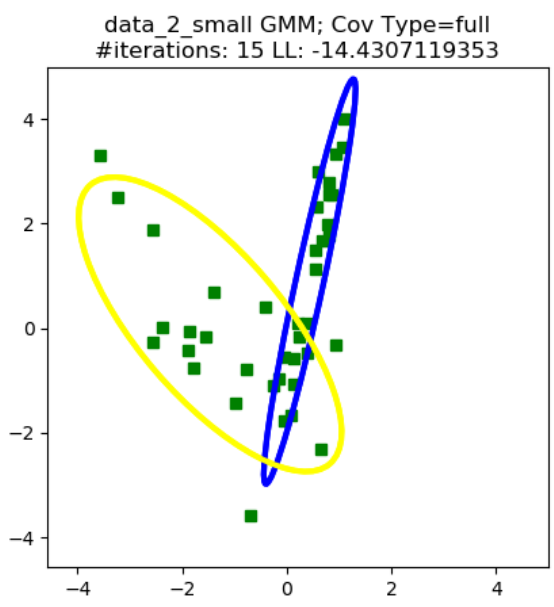
Plot C



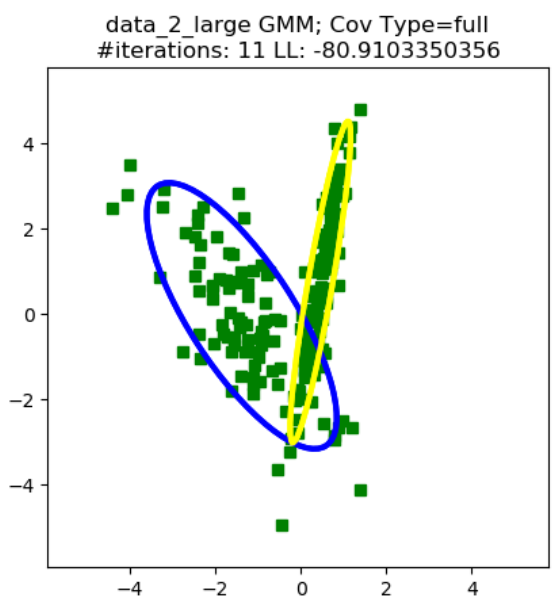
Plot D



Plot E

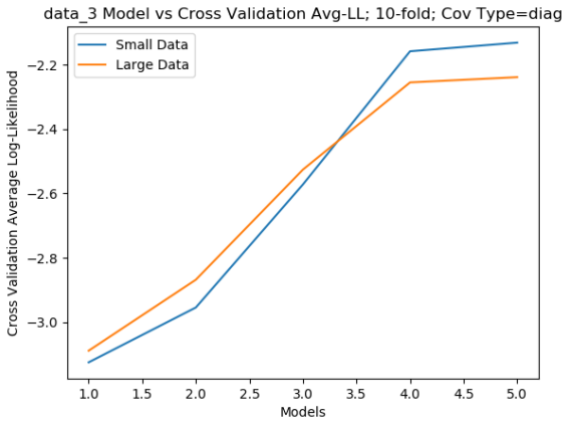

data_2_small GMM; Cov Type=full
#iterations: 15 LL: -14.4307119353

Plot F

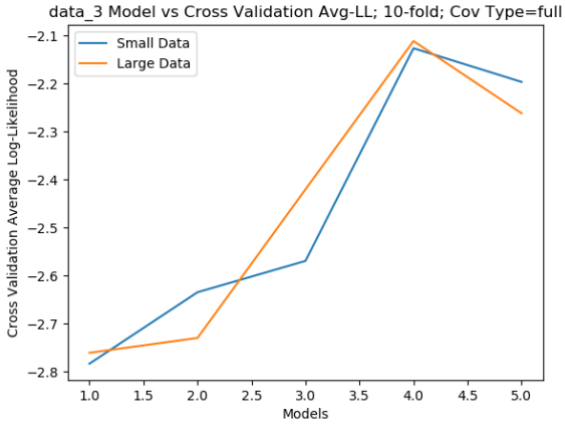

data_2_large GMM; Cov Type=full
#iterations: 11 LL: -80.9103350356

Data-3: Best model is is k=3 and full

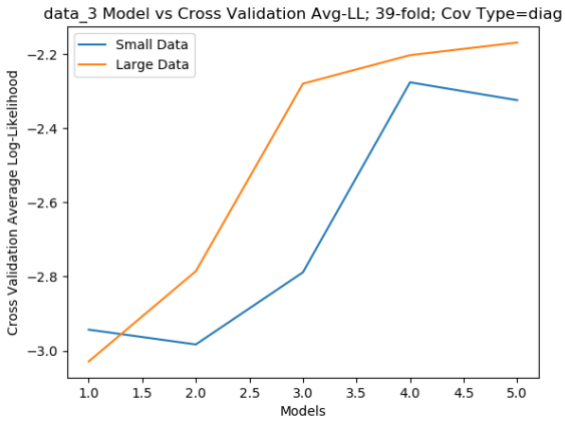
Plot A



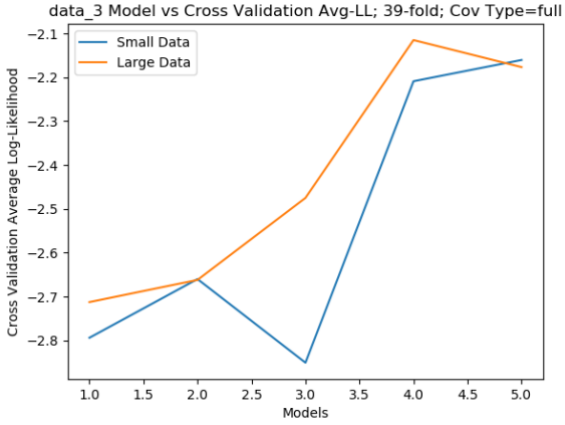
Plot B



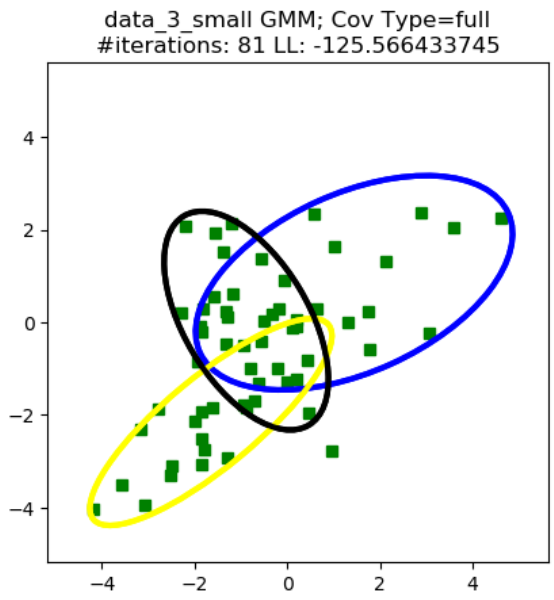
Plot C



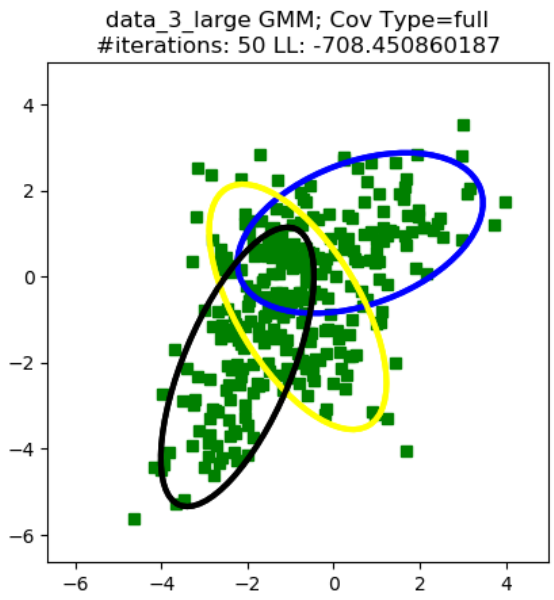
Plot D



Plot E

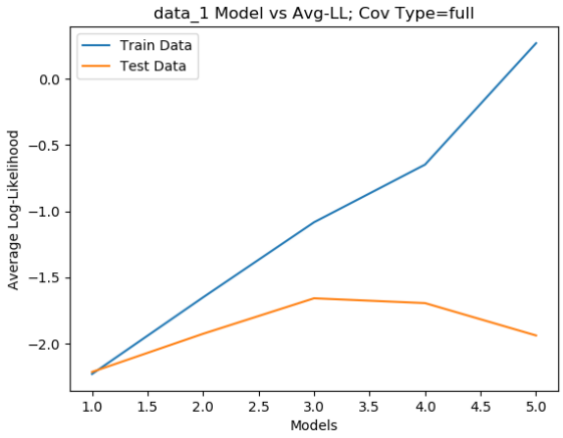
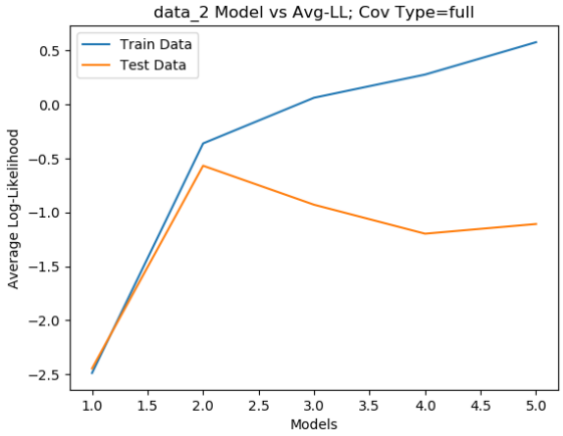
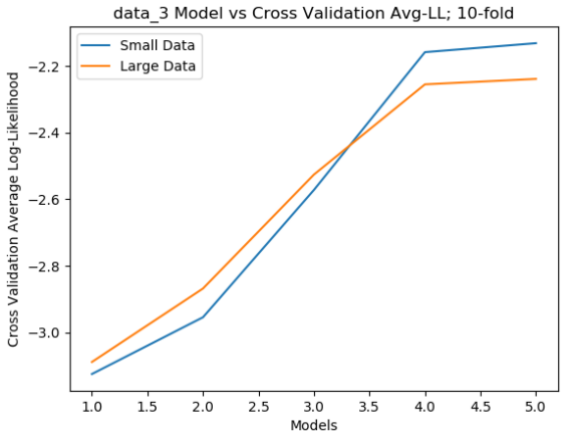
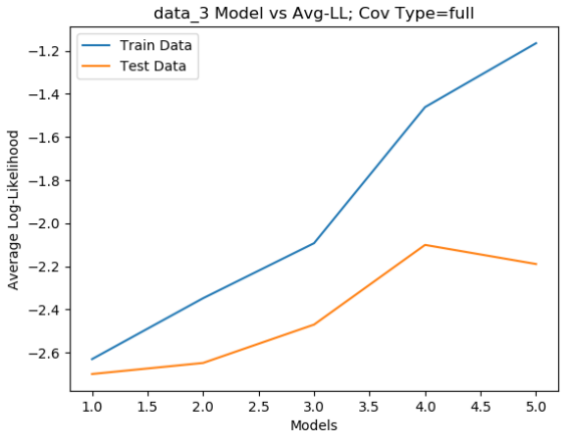


Plot F



Part (b) Comparison between Q3-3 and Q3-1

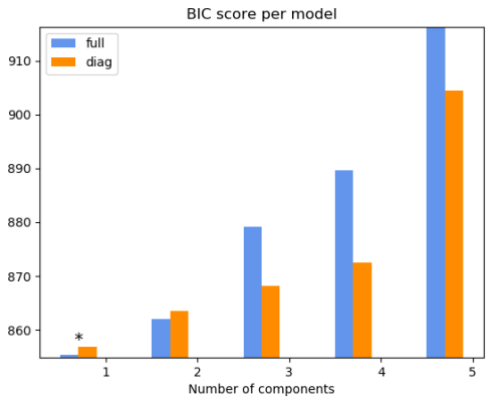
- Below are the comparison plots between that made via cross-validation and others made via simple training and testing using average likelihood. We see that the 2 methods don't display much difference.
- Even though the plots of Q3-1 are supposed to be biased, the plots don't show much difference
- My best guess is that, since this data is really small, the bias does not affect the working of the mixtures
- Another reason could be that, taking the average log-likelihood, reduces that little bias even further and hence, we don't see much difference.

Plots from Q3-3 (Cross Validation)	Plots from Q3-1 (Average LL)
	
	
	

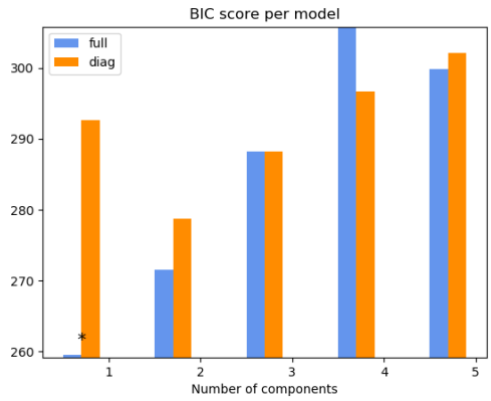
Q4 Best prediction on mystery dataset

In the previous questions, we used various ways to do model selection. We tried randomly setting the initial parameters, initialized the parameters from a K-Means clustering, and/or also did cross-validation to for setting the best parameters based on the available data.

However, we can use Bayesian Information Criteria (BIC) to perform model selection. When fitting models, we can increase the likelihood by adding more components, but doing so we may result in overfitting.



Mystery 1 Data

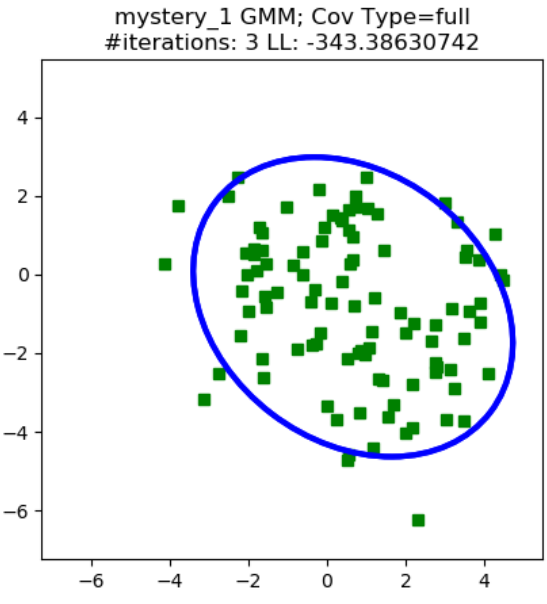


Mystery 2 Data

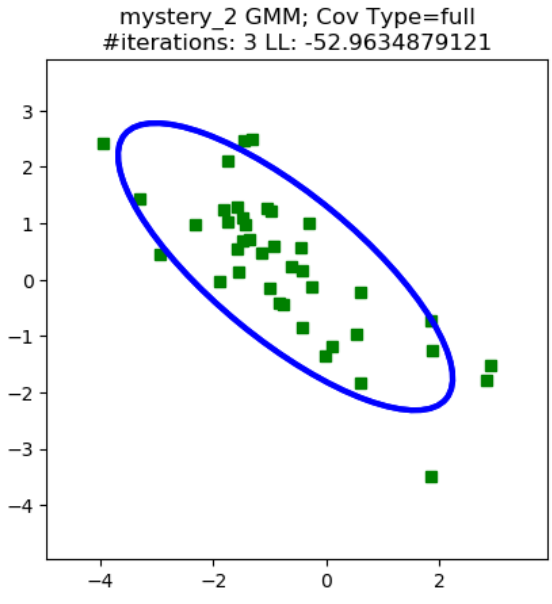
References: Link

Based on the above 2 BIC plots, it is clear that we both mystery plots work best for 1 component and full covariance matrix. Below are the plots for each.

Mystery 1 Data



Mystery 2 Data



Comparing my results (above) with that generated from Sklearn's Gaussian Mixture library (below), we can see that they match perfectly.