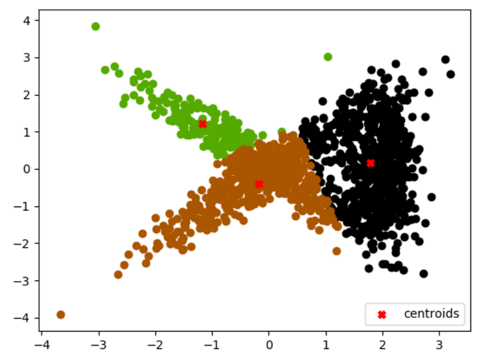
Dante Kiaunis

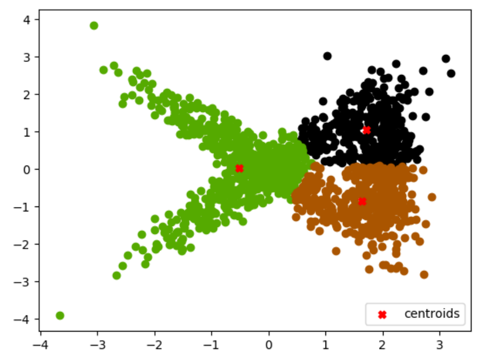
Programming 3 Report

k-Means

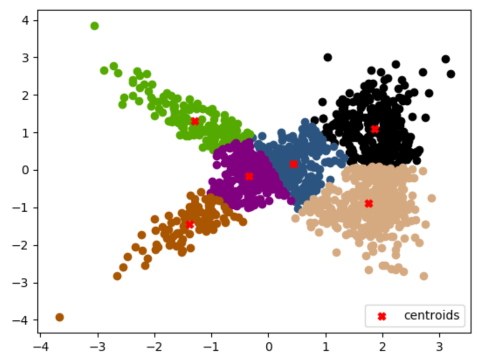
K = 3, R = 1, SSE = 1768.2



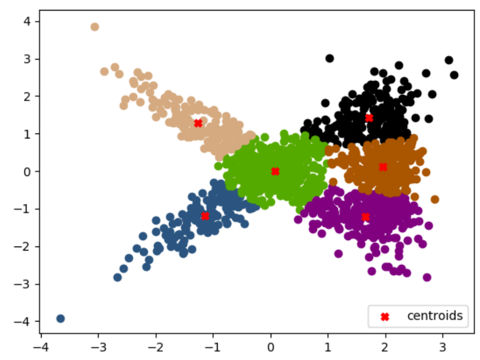
K = 3, R = 5, SSE = 1539.3



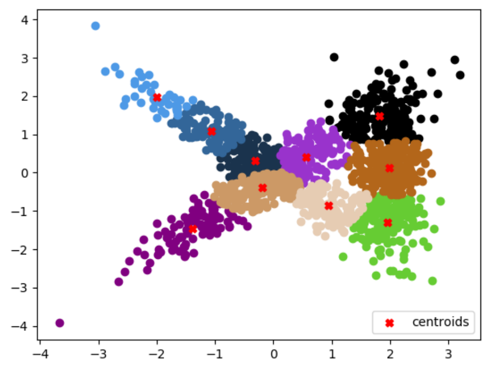
K = 6, R = 1, SSE = 698.9



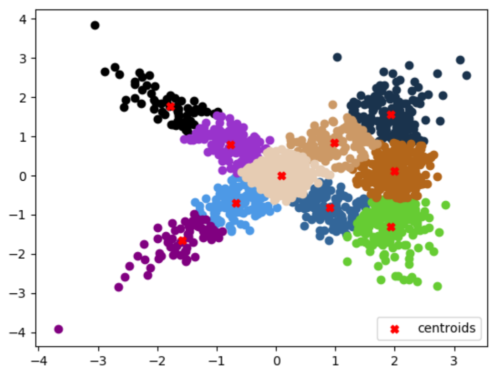
K = 6, R = 5, SSE = 627.0



K = 10, R = 1, SSE = 393.4



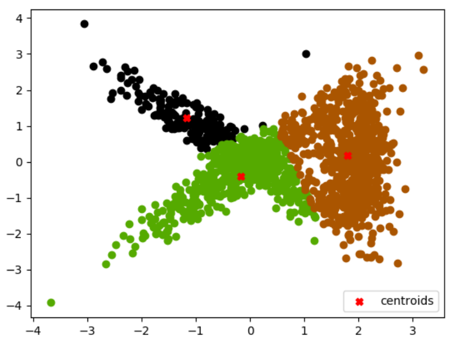
K = 10, R = 5, SSE = 367.7



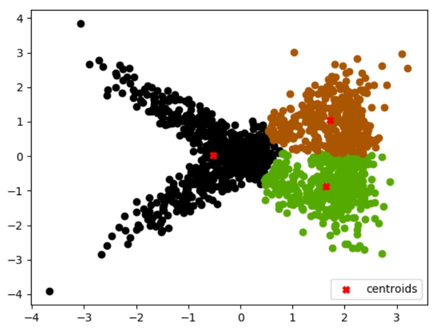
From the graphs we can see that with more clusters, the SSE goes down significantly. This makes sense because each cluster is more tightly packed together. Also, when the R value is higher the SSE value goes down because the best iteration is chosen out of R iterations. The graphs when R=5 have clusters that visually look more evenly distributed.

Fuzzy c-Means

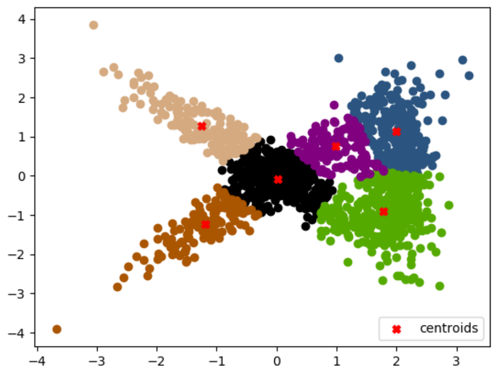
K = 3, R = 1, M = 1.1, SSE = 1762.0



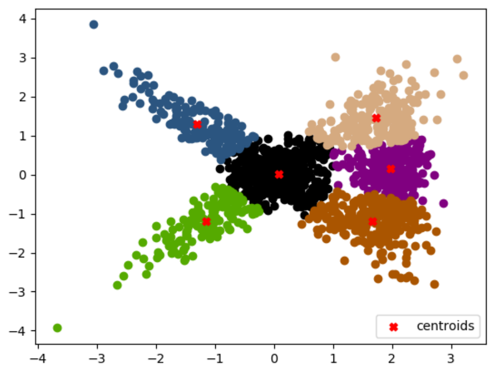
K = 3, R = 5, M = 1.1, SSE = 1536.7



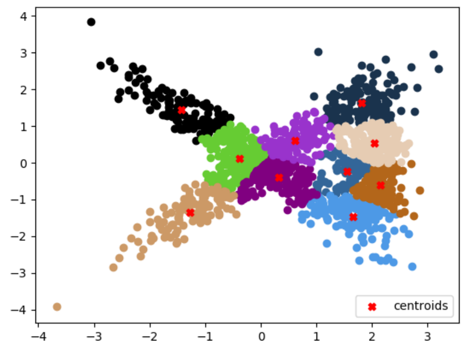
K = 6, R = 1, M = 1.1, SSE = 700.6



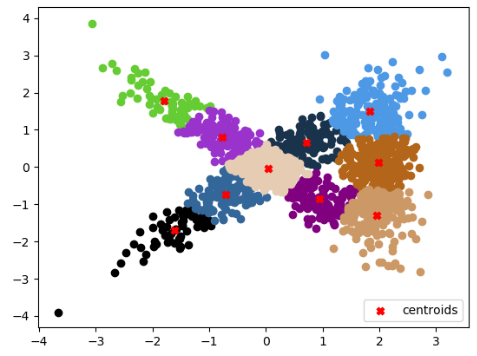
K = 6, R = 5, M = 1.1, SSE = 624.7



K = 10, R = 1, M = 1.1, SSE = 431.6



K = 10, R = 5, M = 1.1, SSE = 368.4



While running the fuzzy c means I plotted the data points with a hard threshold for ease of graphing. An interesting thing I noticed is that with small values of M, the graphs and SSE come out almost identical to the K means, which makes sense. With large values of M the hard threshold makes the clusters bad looking.

Another interesting thing is that the centroids for fuzzy c means start at the center and spread out from there. K means starts all over the place and then moves into the clusters. This is because of how the algorithms are initialized. K means sets the centroids to random points on the graph, while fuzzy c means sets the membership grades to random values. The random membership grades make almost no sense when trying to create centroids, so they all start towards 0,0. I sent along two gif animations in the folder with this report for a visual representation of the kmeans and fuzzy c means.

As far as both algorithms go, I just followed the algorithms that were written in the slides. To create the gifs, I save each step of an iteration in a list. When R > 1, if another iteration has a better sse than the previous best, I replace the steps of that iteration with the steps of the new iteration. At the end I save all the graphs as png files in a folder called outputs.