AI/ML - CS 337 Lab - Lab Assignment 4

B.Nikhil 170050099

October 6, 2019

Problem 1: Principal Component Analysis

Task 2

Execution time for `pca_small` for $d = 50, n = 3000$ is 0.028 sec where as for $d = 3000, n = 100$ is 18.753 sec
Bottleneck for our code is the value of d. As the covariance matrix for which we are calculating eigen values is of dimension $d \times d$. Calculation of eigen values takes more time and is the bottleneck.
For the case $d >> n$, We realize top k eigen vectors of $\hat{X}\hat{X}^T$ is same as eigen vectors of $\hat{X}^T\hat{X}$ premultiplied with X. Calculating eigen vectors of $\hat{X}^T\hat{X}$ will take much less time in this case as the dimension of $\hat{X}^T\hat{X}$ is $n \times n$.

Problem 2: k-means clustering

Task 1: Implementing k -means clustering

k-means clustering result for `flower.csv` using random seed and 8 clusters is shown in Figure 1. Figure 2 shows SSE for each iteration.

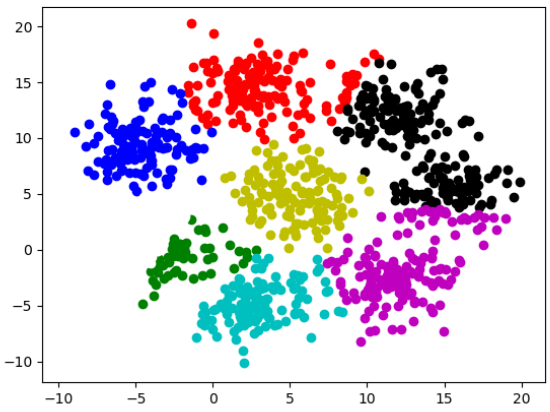

Figure 1: `flower.csv` clustering using k = 8.

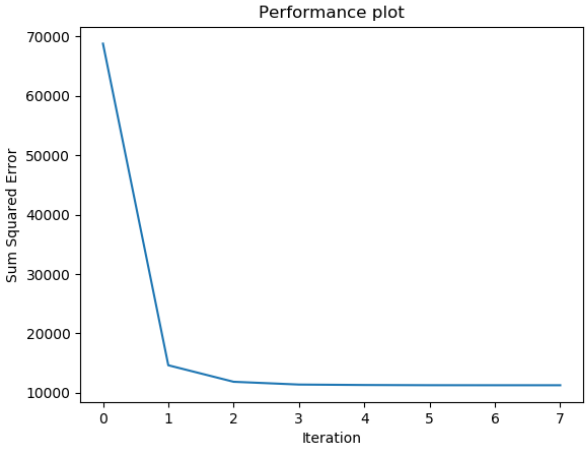

Figure 2: SSE vs iterations for `flower.csv`.

Task 2: Testing and Performance

1

While proving that,k-means clustering algorithm converges in finite number of iterations, we have shown that at each iteration, the K-Means algorithm reduces the objective. Assigning new centroids as mean of each cluster reduces SSE and assigning label for each point as its nearest cluster also reduces SSE. Therefore, SSE never increases with iterations.

2

If the initial values of cluster centroids is initialized towards end of lines for one cluster and towards other end for the next cluster, then the points on that line become close to the centroid of next line and the points on the same end of next line will be closer to this centroid. This makes clustering bad for `3lines.csv` but for `mouse.csv` clustering is as expected. If initial centroids as initialized as the midpoints on each line then we get the clustering as expected.
k-means clustering result for `3lines.csv` and `mouse.csv` using k = 3 is shown in Figure 3 and 4 respectively.

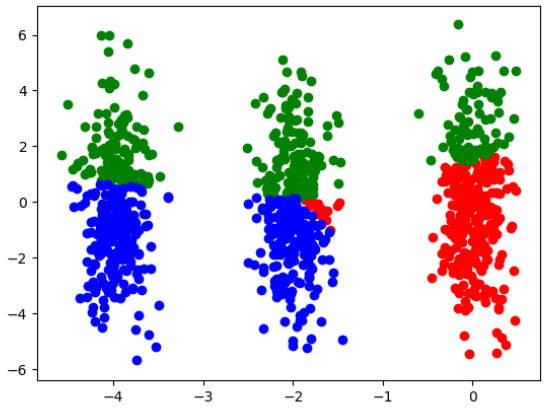

Figure 3: `3lines.csv` clustering using k = 3.

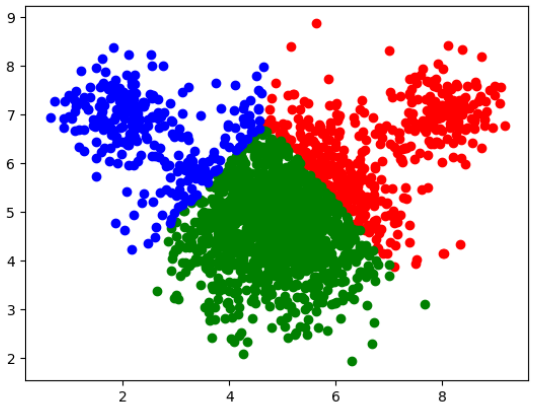

Figure 4: `mouse.csv` clustering using k = 3.

Task 3: Kernel k-means clustering

Final plots for the clusters `3lines.csv` and `mouse.csv` are shown below(and also placed in `kernel_plots` folder

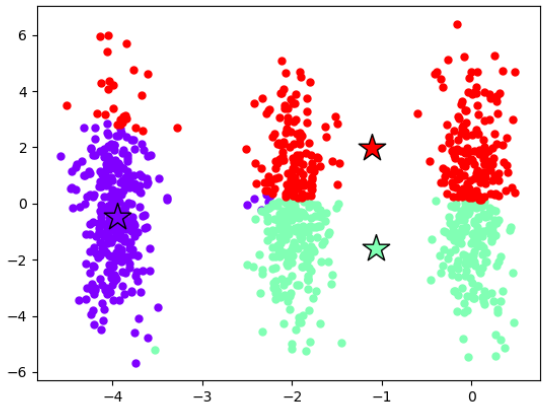

Figure 5: kernel clustering for `3lines.csv`.

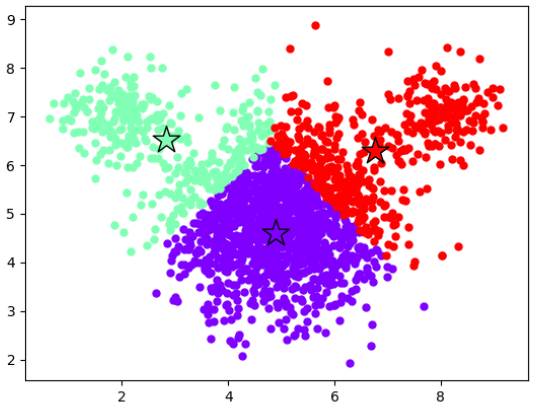

Figure 6: kernel clustering for `mouse.csv` .