Image Clustering using K Means

Name: Manaff Khan
GMU ID: G01273277
Miner Username: dumbbot
Miner Score: HW3(Part1): 0.95; HW3(Part2)0.80

Approach and Findings:

Part 1:

For the IRIS dataset implemented the k means using cosine similarity as the similarity function to cluster the data points into 3 clusters. The outcome for this setup was 0.95 on Miner. Before Cosine Similarity, I used Euclidean Distance for clustering which resulted in a 0.65 on miner

Part 2:

For the Image Clustering I used the previous implementation with multiple distance/similarity metrics to cluster the data into 10 clusters. Following were my findings:

Distance/Similarity Metric	Sum of Squared Errors*	Miner Score
Manhattan Distance	86990883	0.42
Euclidean Distances	85558659	0.50
Cosine Similarity	88008053	0.51

*The Sum of Squared Errors here is the sum of all the distances between the data point and its centroid.

The next step was reducing the number of features. Since the each feature represents one pixel data, for a 28x28 image there are 784 feature.

For feature reduction I used PCA. Following are the results of clustering the data that was reduced to 50 principle components. These 50 principle components accounted for approximately 83 percent of variation in the data.

Distance/Similarity Metric	Miner Score
Manhattan Distance	0.49
Euclidean Distances	0.50
Cosine Similarity	0.46

The next step was fitting the PCA results to 2 dimensions using t-SNE and then clustering the final data. Following are the findings:

Distance/Similarity Metric	Sum of Squared Errors*	Miner Score
Manhattan Distance	86990883	0.75
Euclidean Distances	85558659	0.79
Cosine Similarity	88008053	0.67

I also changed the number of principle components to 60, 70 and 90 to increase the variation of my data before implementing t-SNE. I got the best results for n_components = 70 (**MINER = 0.80**)

Following are the scatter plots for clustering using just PCA at n_components = 70(fig.1) and PCA(n_components = 70) + tSNE(fig.2)

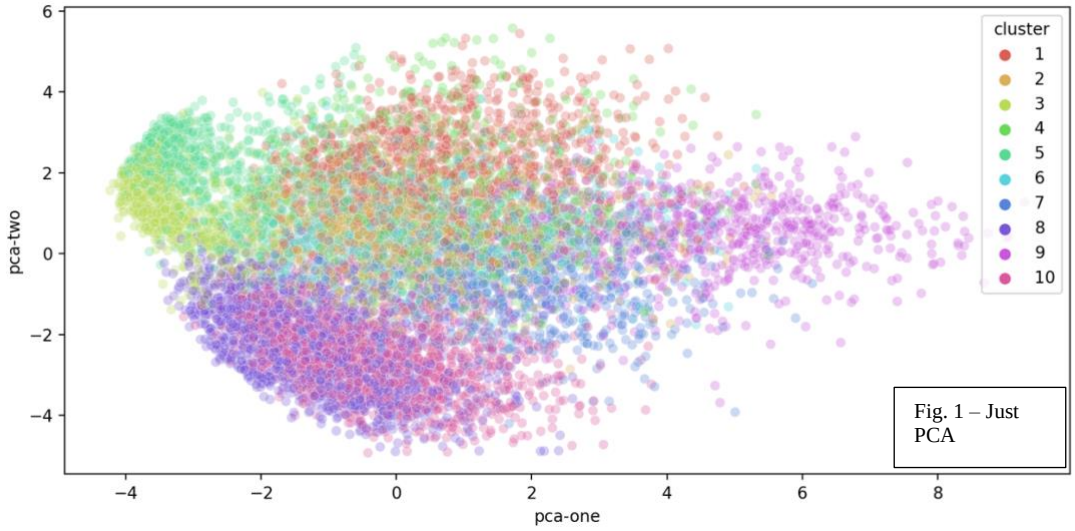


Fig. 1 – Just PCA

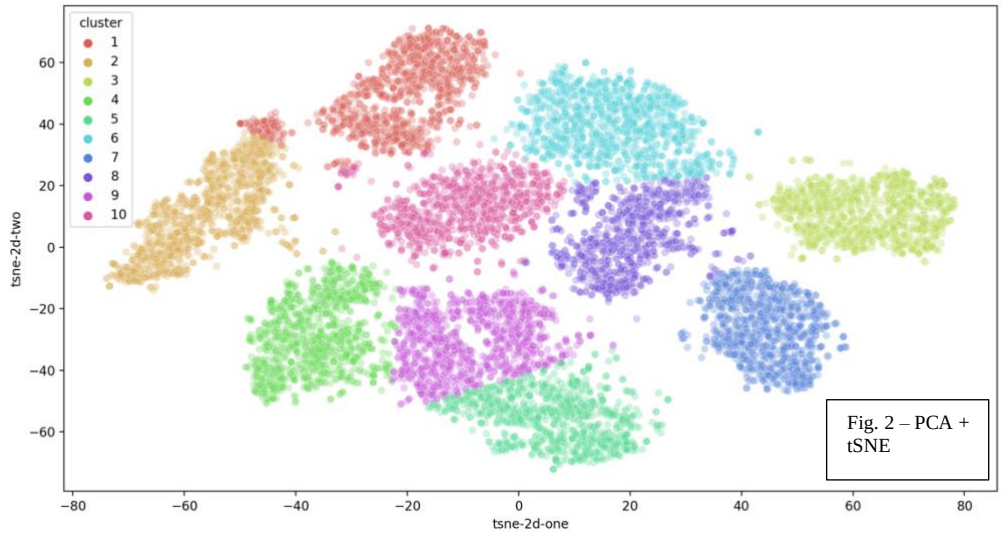


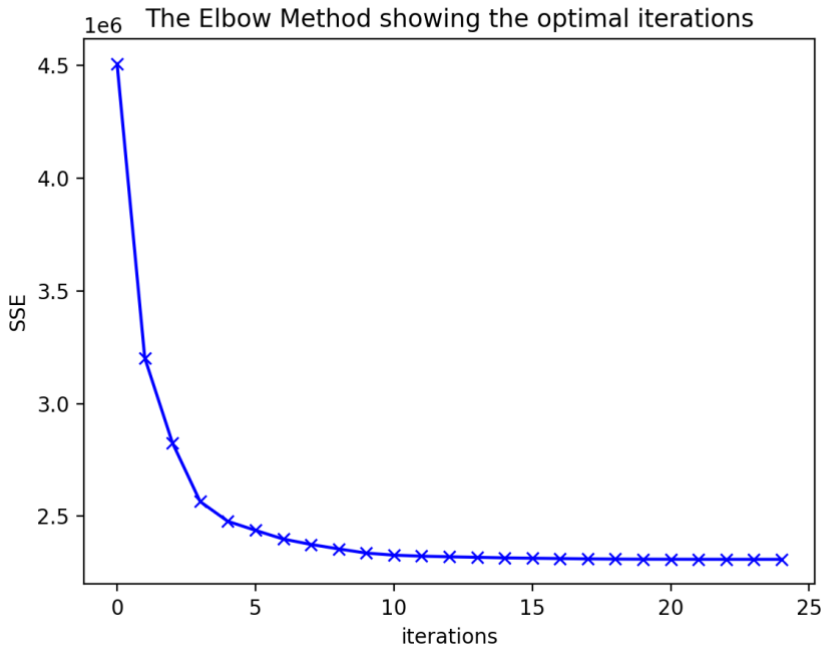
Fig. 2 – PCA + tSNE

Pseudocode :

```
Input data into a Dataframe <df>;
pca_result = PCA(df, n_components = 70);
tsne_result = t-SNE(pca_results, n_components = 2);
centroids = []
repeat 25 times:
    if (first iteration) then:                  //this step initializes the centroids
        centroids[10] = random(tsne_results, 10) //select random 10 data points as initial centroids
                for every element in tsne_result, repeat:
                    assign the element to a cluster based on its minimum distance to a centroid[i = range(0,10)]
        else:                                   //this step updates centroids
            for i from 0 to 9 repeat :
                centroids[i] = mean of i'th cluster
              for every element in tsne_result, repeat:
```

assign the element to a cluster based on its minimum distance to a centroid[i = range(0,10)]

Value of SSE as the iterations increase for PCA(n_component = 70) + tSNE(n_component = 2):



Value of K from 2 to 20 VS Sum of Squared Error: