CS158 Final Project: K-Means

Zach Hinz, Devin Guinney, Unity Tambellini Smith, Keya Gupta
Computer Science Department, Pomona College

1. Introduction

Unsupervised learning algorithms attempt to find patterns or groups within data without labels, with being the most common algorithm. In K-Means, we randomly instantiate k different groups, and then assign all points to the nearest centroid. We then recalculate the centroid to be the mean of all its associated points, repeating for a set number of iterations. For our final project, we implemented K-Means in Java and conducted several experiments to understand the best hyperparameters and overall performance. In our model, we implemented three hyperparameters: the number of centroids k, the distance measurement (choice between Cosine and Euclidean) and initialization (choice between random and the furthest centers heuristic).

2. Experimental Setup

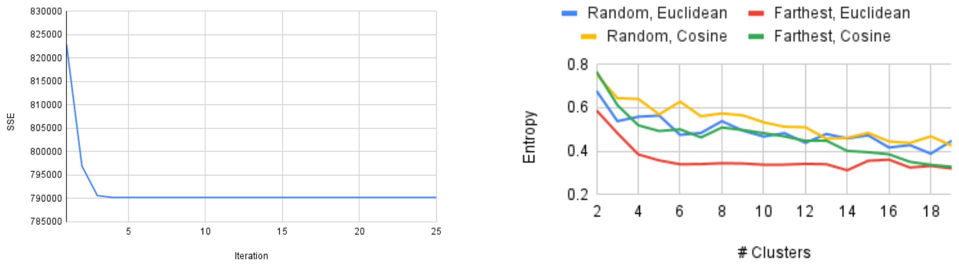
To evaluate our model, we used a dataset that classified 3 different wines based on 14 different continuous attributes (Magnesium, Ash, Alcohol, etc) (Garcia 2020). We went with this dataset over the text based wine dataset because we wanted to look at just a limited number possible attributes. To find optimal hyper-parameters, we looked at 4 evaluation metrics: Sum of Squared Errors, Entropy, Purity, and the Silhouette Score.

To ensure our model is working as expected, we looked at the Average Sum of Square Errors for each iteration. Because K-Means is trying to find the minimum of this loss function, we wanted to ensure this is occurring. Entropy is a measure of how much randomness there is within the cluster (the less the better), and it requires labelled data. Purity is the proportion of the most common label in the cluster, and it also requires labelled data. Silhouette Score is used on unlabelled data, and it measures both cohesion and separation of clusters by measuring average intra-cluster distance and average inter-cluster distance for each example. It varies from -1 to 1, with 1 being best.

For each of these three different evaluation metrics, we looked at 4 different combinations, using either Random Initialization or Farthest Centers Initialization, and either Euclidean Distance or Cosine Distance, varying our k value. For each evaluation metric and hyper parameter choice, instead of using 10-fold cross validation, we took an average of 10 different trials to find an overall trend.

3. Results

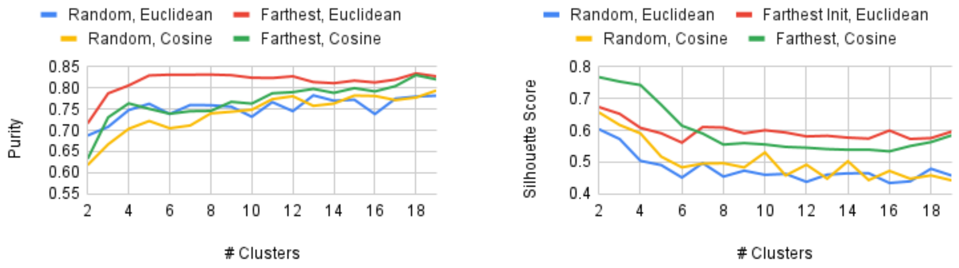
Firstly, we see a sharp drop in SSE within the first several iterations, as expected, before the model quickly converging to a minimum in Figure 1a. Moreover, in looking at all of our graphs, we see that the farthest centers heuristic is generally more effective compared to random initialization, as across all k, farthest centers produces



(a) SSE over Iterations

(b) Entropy across K

Figure 1: Evaluation I



(a) Purity Score Across K

(b) Silhouette Score across K

Figure 2: Evaluation II

lower average entropy in Figure 1b, higher average purity in Figure 2a, and higher average silhouette scores in Figure 2b. However, for this dataset, Euclidean and Cosine distance perform similarly well when using Farthest Centers. Euclidean distance seems to achieve better purity and entropy, while Cosine achieving a better Silhouette score. In all of our measurements, we see that 3 to 4 clusters is optimal, with Silhouette being highest at 3 clusters in Figure 2b, and purity and entropy reaching equilibrium at just 4 clusters in Figures 2a and 1b. This indicates a good model, as this dataset is known to contain 3 clusters.

4. Conclusion

Through our experiments, we see that K-Means is a relatively effective model for extracting trends or patterns in unlabelled data, while also being easy to implement. Through out experimentation, we found that the most effective model (based on Silhouette Score), was using Farthest Centers initialization with either Cosine Distance or Euclidean distance. As stated in lecture, distance measure depends on dataset, but Cosine is generally preferred for most real world datasets. Experimentation indicated that 3-4 centroids was recommended for this dataset.

References

Garcia, Xavier Vivancos. 2020. Clustering tutorial with wine dataset. https://www.kaggle.com/code/xvivancos/tutorial-clustering-wines-with-k-means. Accessed: 2023-12-4.