Naomi Igbinovia
Joshua Phillips
CSCI 4350
6 December 2023

Unsupervised Learning and K-Means Clustering

**Introduction**

In machine learning, unsupervised learning provides an alternative predictive model to work with instead of decision trees in terms of supervised learning. Unlike supervised learning and the use of labeled data, algorithms that fit in the unsupervised learning arena work with unlabeled data and take in observed patterns. One of the many techniques of unsupervised learning is k-means clustering. This algorithm works like the decision-tree algorithm, as a tree-like structure is also used here. But instead of the decision tree, a clustering tree is used instead. Each node of a clustering tree represents a centroid and each lead represents a final cluster assignment. The goal of a clustering tree is to refine sweeps of data points into the most accurate groupings possible, based on feature similarities.

**K-Means Clustering Method**

To solve this algorithm, the kmeans.py program was developed. This program takes training data as a text file to use as an algorithm input. Then, a set of validation data, also in the form of a text file, is used to check the current clusters made by algorithm to see if they could be improved any further. The k-means clustering algorithm is used by sectioning the training data into clusters based on similarities, setting data points to clusters, and ending at convergence. With the validation data, data points are assigned to their closest centroid, then the data compares the assigned cluster labels with true labels (if they are available) to assess the clustering performance.


Figure 1. K-means clustering was performed on multiple data points until five clusters were created to make the most accurate groupings possible.

**Code Development**

The kmeans.py program utilizes a couple of methods prior to the main function. The ‘read_data(filename)’ method takes in the given file and reads in the data from it. The 'euclidean_distance(a, b)’ method takes in data points a and b, calculates the Euclidean distance between these two points, and returns the result. The 'update_centroids(clusters)’ method takes in a list of clusters, updates the centroid vectors based on assigned data points in each cluster, and returns the updated centroid vectors based on each of the clusters’ mean of data points. The 'classify_validation_data(validation_data, centroids, cluster_labels)’ method takes in the validation data set, centroid vectors, and labels assigned to each cluster, classifies the validation data points using the nearest centroid and compares the cluster labels to the true labels, and returns the total correct classified validation data points. The 'kmeans(training_file, validation_file, k)’ method takes in the training and validation datafiles and returns the total correct classified validation data points.

**Performance**

To test the performance of the k-means clustering algorithm, cross validation was used. The program ran twice. Each time, 100 rounds of randomly shuffled data were used with a training set size of n - 10 and a validation set size of 10. n serves as the total number of examples in the data set.

The mean of the correct classification percentages v. number of clusters was gathered for both datasets as plots displayed below.

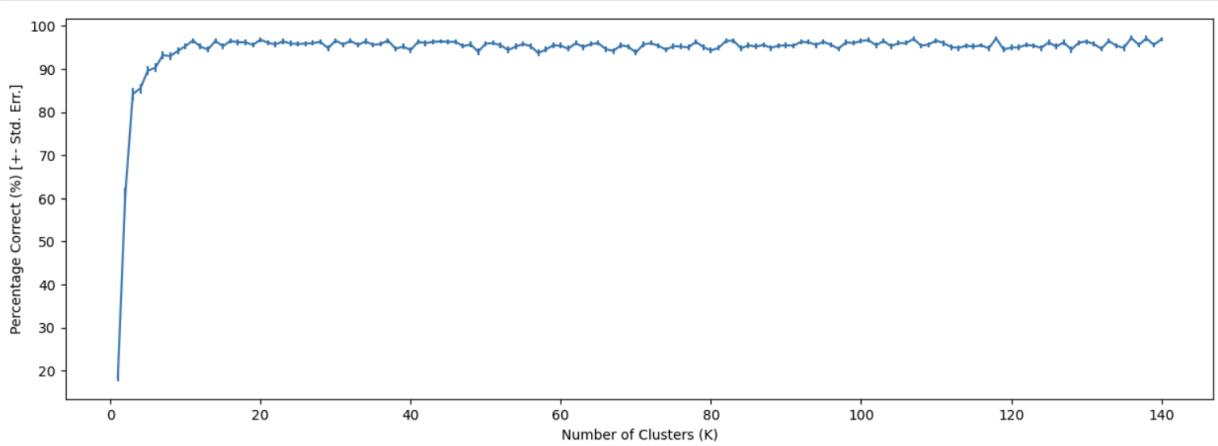


Figure 2a. The accuracy of the  K-means clustering on the iris dataset.

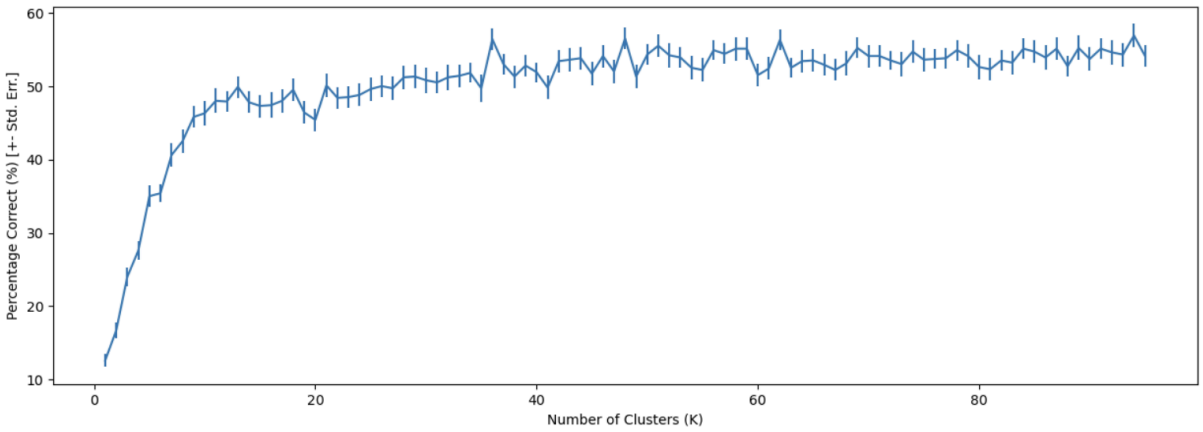


Figure 2b. The accuracy of the  K-means clustering on the cancer dataset.

**Analysis & Conclusion**

Based on the plots shown above, it can be inferred that an increasing number of clusters improved the classifications as the program continued. The point where the classification could not improve any further is where the performance plateaued. This notes that there is an importance in selecting an optimal number of clusters.

The implemented program does an excellent job of assessing K-means clustering performance. With cross-validation, insights were obtained of how the algorithm behaves with varied numbers of clusters.

**Sources**

(LEDU), Education Ecosystem. “Understanding K-Means Clustering in Machine Learning.” Medium, Towards Data Science, 12 Sept. 2018, towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1.
“K Means Clustering - Introduction.” GeeksforGeeks, GeeksforGeeks, 11 Mar. 2024, www.geeksforgeeks.org/k-means-clustering-introduction/.
Sharma, Pulkit. “The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications.” Analytics Vidhya, 20 Feb. 2024, www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/.