Ryan Armstrong
Dr. Phillips
CSCI 4350/5350
December 4, 2018

Open Lab 4: Unsupervised Learning Using K-Means Clustering Report

The lab implemented the K-means clustering algorithm to demonstrate its performance in classifying a four-dimensional set of data given 140 training examples and 10 testing examples, with K, the number of clusters, falling in the range [1, 140]. The K-means clustering algorithm takes a set of N training examples each with a D-dimensional vector of attributes and treats them as D-dimensional points in space. It is initialized by selecting K unique random training examples and creating K cluster centroids at their locations. Following this, we iterate over each training example and assign it to the closest cluster centroid, after which we update the location of the centroid to be the average location of each assigned training example. This step is repeated until all training examples are assigned to the same cluster as on the previous iteration. We then assign each cluster centroid a classification corresponding to the majority classification of its assigned training examples and the algorithm is considered complete. Classification of a testing example occurs by finding the closest cluster centroid to the example's vector of attributes and assigning it the cluster centroid's classification.

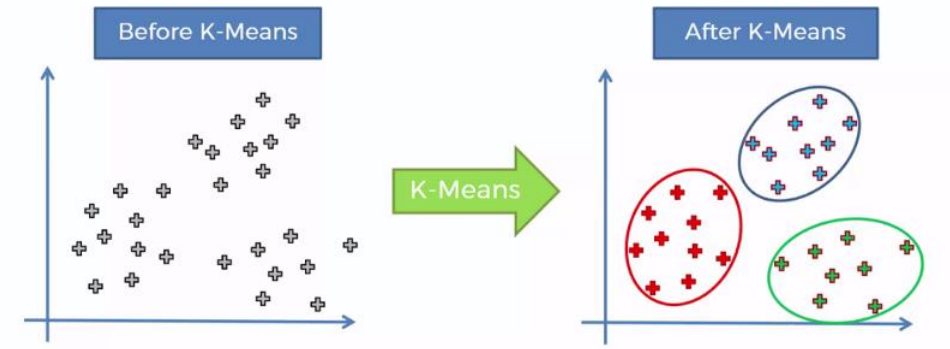


Fig 1: Example result of applying K-means clustering to a set of training data.

The program utilizes two user-defined structs and a user-defined class to organize data and operations.

The Dataset struct is a representation of training and testing examples and is primarily used for organizing and retrieving data. It contains the following:

- Static Member Data
  - int nFeatures – the number of attributes expected for each set of data
  - int nClasses – the number of unique classifications found amongst all sets of data
  - set<int>* classifications – a pointer to a set of all unique integer classifications discovered amongst all datasets
- Member Data
  - vector<double> features – contains the attributes of an example
  - int classification – the classification of an example

The Cluster struct is a representation of K-Means cluster centroids. It contains the following:

- Member Data

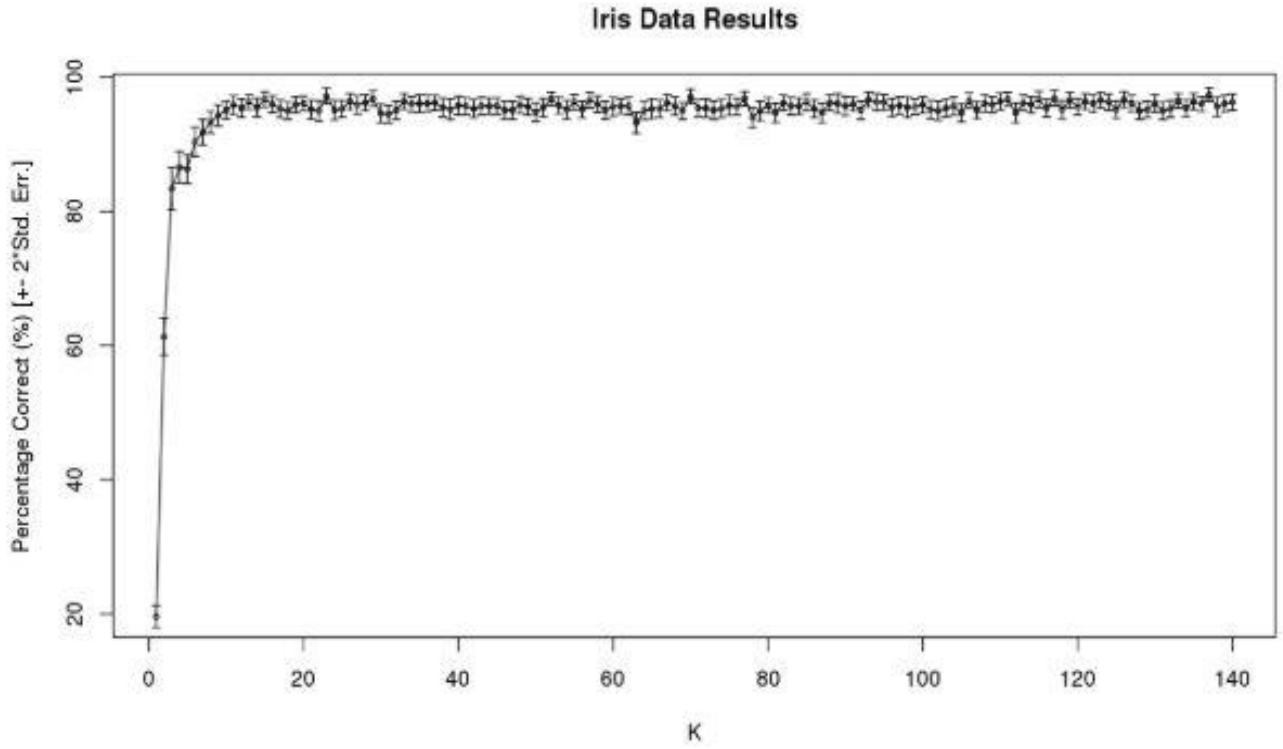
    - vector<double> center – the D-dimensional location of the center of the Cluster
    - set<Dataset*> assigned – a set of Datasets that are closer to the centroid than to others
    - int classification – the majority classification of all assigned Datasets
- Methods
    - Cluster(vector<double> p) - initializes a Cluster with a given D-dimensional point p
    - void calculateClassification() - finds the majority classification of all assigned Datasets and assigns it as the Cluster's classification
    - double distanceTo(vector<double> p) - returns the distance from the Cluster to a given D-dimensional point p
    - vector<double> generateCenter() - returns the location obtained by averaging the locations of all assigned Datasets
    - bool assign(Dataset* ds) - attempts to assign a Dataset to the cluster; returns true if the operation was successful, false otherwise

The KMeans class performs the K-Means clustering algorithm and classifies testing examples. It contains the following:

- Static Member Data
    - int nClusters – the total number of Clusters to create
- Member Data
    - vector<Cluster*> clusters – all the clusters we're working with in the algorithm
    - vector<Dataset*> training – all training examples used to perform K-Means clustering
    - map<Dataset*, Cluster*> - maps each training example to the Cluster to which it was assigned on a previous iteration of the algorithm
- Methods
    - KMeans(vector<Dataset*> t) - constructor, performs the K-Means clustering algorithm after initializing Clusters
    - int classify(Dataset* ds) - classifies a testing example by returning the classification of the closest Cluster
    - void initClusters() - initializes clusters by using a set<int> to keep track of random indices into the vector of training examples and selecting nClusters unique Datasets
    - Cluster* clusterClosestTo(vector<double> p) - finds the cluster closest to a given point p
    - bool clustersSettled() - iterates through each Dataset* in the vector of training examples and checks if the entry in dataClusterMap for the given Dataset* is the same as the Cluster returned by the clusterClosestTo method. If true, clusters are considered settled and new centers for centroids can stop being generated.

The program was trained with a set of 140 training examples and tested against a set of 10 testing examples, all of which contained four attributes and a unique integer classification in the range [1, 3]. The program was run 100 times for each K in the range [1, 140], with training and testing sets obtained by shuffling a set of 150 examples and splitting the first 140 lines from

the final 10 for each run. The Accuracy vs K is plotted below with error bars:



Beyond seven clusters, the accuracy of the algorithm does not improve with an increasing value of K; however, there is a noticeably significant improvement from K = 1 to K = 2 and from K = 2 to K = 3. From K = 3 to K = 7, small (but still noticeable) improvements in accuracy are made, ranging from about 85% accuracy to 95% accuracy. For the case of K = 1, only one cluster is initialized, and thus only one classification can be identified, resulting in an extremely low accuracy. This is like the case of K = 2, where two integer classifications can be classified. With K = 3, all integer classifications can be classified. For the cases of K = 4 to K = 7, imagine that we have two groups of examples with classification 4: one in the range ($\{x \mid 0 \leq x \leq 1\}$, $\{y \mid 0 \leq y \leq 1\}$) and the other in the range ($\{x \mid 2 \leq x \leq 3\}$, $\{y \mid 2 \leq y \leq 3\}$). With extra K, since the K-Means clustering algorithm in non-parametric, we would theoretically be able to have clusters with classification 4 at both locations. Thus, in a sense, we are "catching outlier groups" or "getting more detailed" with the values K in the range [4, 7].

Works Cited

Patil, Prasad. Example of K-Means Clustering. *Towards Data Science*, May 20, 2018, https://towardsdatascience.com/k-means-clustering-identifying-f-r-i-e-n-d-s-in-the-world-of-strangers-695537505d