IE411 Term Project

Vlad Predovic

Monday, December 7th, 2015.

[Introduction 1](#_Toc436230733)

[K-Means 1](#_Toc436230734)

[K-Medoid 2](#_Toc436230735)

[Implementation Details 2](#_Toc436230736)

[Input and Verification 3](#_Toc436230737)

[Selecting Initial Centroids/Medoids 3](#_Toc436230738)

[Forming and Finalizing Clusters 4](#_Toc436230739)

[*K-Means* 4](#_Toc436230740)

[*K-Medoid* 4](#_Toc436230741)

[Assessing Cluster Variation 4](#_Toc436230742)

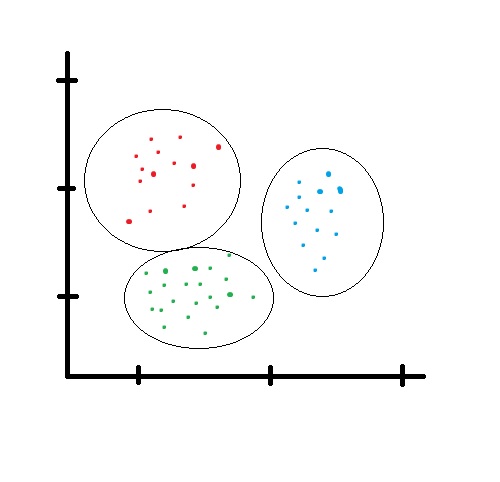
[User Guide 4](#_Toc436230743)

[Section I: Guide Introduction 4](#_Toc436230744)

[Section II: The program 5](#_Toc436230745)

[Section III: Exported Data 5](#_Toc436230746)

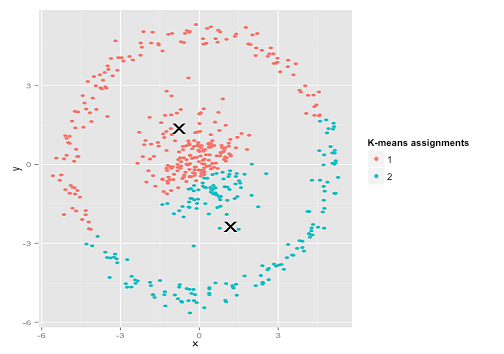
# Introduction

The Visual Basic program that goes along with this paper is designed to provide a quick and user friendly way to input data and perform cluster analysis. The program allows the user to pick between K-Means and K-Medoid, two different methods used for performing this type of analysis .

Cluster Analysis in a short statement is the grouping of data into separate categories based on a quantitative attribute. The grouping can be done through different methods but ultimately each data point becomes attributed to a distinct group. This results of this type of analysis are easier to understand graphically, but can be imagined to be somewhat like the picture on the right. A common use of this analysis is to try to differentiate data and see if it came from different sources.

*Fig. 1*

K-Means

http://www.purplemath.com/modules/xyplane/dist07b.gif K-Means is a common method employed to solve the clustering problem and a commonly associated mathematical implementation to solve it is labeled as Lloyd’s algorithm. It begins with defining an n amount of centroids within the range of your data set. There are many different ways to place these centroids but ideally you would want them spread throughout the data. After this you associate each data point to the centroid which it has the lowest distance from. Since this program only allows 2-dimensional objects, finding the distances can be done using the Pythagorean theorem as in figure 2. Afterwards new centroids are deduced from all the objects that currently make up each cluster. The process loops, attributing each data point to a given cluster and deducing the center point until these stop changing altogether.

*Fig. 2*

The K-Means method has some drawbacks. The easiest to represent of these is the idea that it tries to create circular groupings of data as in figure 1. If you had an example with a different type of pattern, the K-means method would fail such as the one shown in figure 3 to the right where it is obvious that all the data in the middle is one cluster and the large ring on the outside is a separate set of data.

*Fig. 3*

K-Medoid

The K-medoid method is similar to the k-means method described above in that it attempts to attribute data objects into clusters depending on the distance of these points to the designated cluster centroids. One of the more common realizations of this method is the Partitioning Around Medoids (PAM) method which begins by selecting random existing data points as your medoids. Each data point is then associated to the closest medoid. Each data point in a cluster is then taken as the medoid and the total distance cost of all data points to this point is calculated. The central point of the configuration with the lowest cost is selected and that point then becomes the medoid. The algorithm loops as all the data is then re-associated to the closest of the new medoids. Once the medoids do not change anymore the algorithm ends.

The main drawback of the K-Medoid method is that it is usually much more expensive to run than the K-means method. In this case you have to calculate every distance from every point to all others each time you run an iteration. The latter method only requires that you calculate the distance to the given centroid points. However, the K-Medoid method is considered more reliable since it is comparable to the median, and therefore less susceptible to being displaced by outliers. The K-means method is more comparable to the mean, and therefore an aggressive outlying point can severely

# Implementation Details

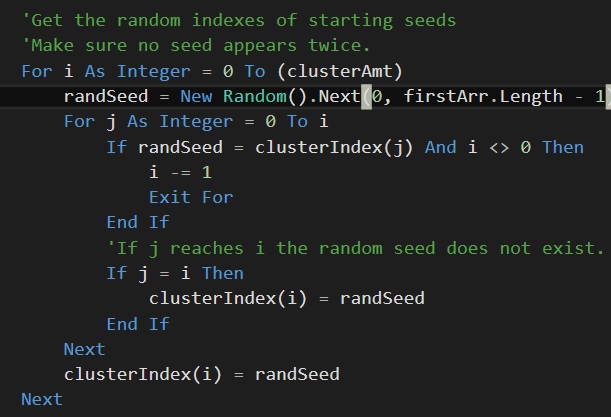
This next section will go into detail over the implementation of the algorithms used by the program. These are known as Lloyd’s algorithm and the Partitioning Around Medoids algorithm, used to solve the K-Means method and K-Medoids method respectively. They methods will be split into four subsections. Each one of these will outline the different paths taken for the respective section or clarify as to why similar code was utilized.

Input and Verification

Although this is done prior to running either algorithm, outlining the methods used to collect and verify the data is still important to understanding how it works. There are three separate ways to input data. Fifty numbers can be generated, fifty numbers can be inserted from a file, or the user can manually input data. If done manually, the program is set up so that you cannot input invalid data by disabling the specific buttons until conditions are met.

When inserting data, two separate global arrays will become filled with a maximum of 50 attributes each and a counter . Any amount of numbers up to 50 can be added, as the arrays will automatically resize to the amount of variables that were inserted. The quantity of clusters that will be seeded is also decided by the users, with a default value of two. When the Calculate button is pressed for either method, the amount of clusters chosen is immediately saved to a local variable in the associated sub-procedure.

Selecting Initial Centroids/Medoids

Selection the initial Centroids or Medoids is done in the same fashion, as the code is essentially reused with minor tweaks. After trying multiple methods including trying to standardize the data and seed it at a local level, the seeding was done using a randomizing algorithm as this appeared to be the most fair method.

The algorithm begins by picking a starting index between 0 and the amount of data points the user set. It then checks to make sure that this index was not previously picked. If it was, another random number is then picked. This repeats until the correct amount of clusters is reached as shown in figure 4. The data existing in the global arrays at the selected indexes then become the initial seeds. These values are transferred to a local 2-dimensional array titled ‘Medoids’ or ‘Centroids’ depending on which button was clicked.

*Fig. 4*

In the case that the first random number picked is the 0th index, this value is accepted. Otherwise an error occurs as the counter drops below 0, since an empty array has it’s values all initialized to 0.

Forming and Finalizing Clusters

### K-Means (Lloyd’s Algorithm)

The entire process of calculating the optimum centroids exists inside a Do-Until loop. This allows the algorithm to terminate smoothly if the initial seeds happen to be the optimal solution points. The Boolean variable Switchvar is set to True as soon as the loop is entered. This will be important later.

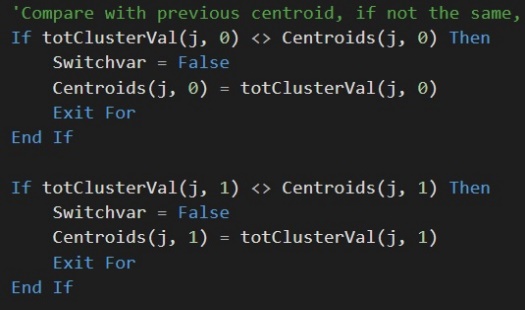
The first steps are all conducted for each individual data point separately by way of an encompassing outer-for loop. The steps taken are for each point are shown below:

* A loop calculates the distance between the given data point and each chosen centroid. The values are saved in a temporary one-dimensional array who’s size is equal to the amount of centroids selected. This inner loop ends.
* A function ‘minFinder’ returns the index of the temporary array in which the lowest distance calculated is found. The index value is add to a third global array the size of the amount of data points which represents the cluster that the given data point is connected to. The outer loop ends here.

A second outer loop begins, this time iterating through each cluster index. Now that the data points are all linked to a given centroid, they can be acted on categorically.

* An inner loop that runs through every data point checks if that point belongs to the given cluster by comparing the cluster index the outer-loop is going to with the value saved to the global array representing the cluster that the data point is associated to. This is done by comparing indexes. If it is, the attributes of that data point are added to a total, and a counter is incremented. Both inner and outer loop end.

A third outer loop begins once again iterating through each cluster index. We now have the total sum of the 2 dimensional data points for each cluster. We do not necessarily need a third loop, but it was put in as it helps separate the program into sections.

* These totals are divided by the quantity of elements in a given cluster to get the new centroid. If the centroid is different from the previous value help by that centroid with the same index, the Boolean variable ‘Switchvar’ is set to false.

The Do-Until loop iterates until the Boolean variable returns true which would signify that the centroids are no longer changing as the switch was never triggered.

### K-Medoid (Partitioning Around Medoids)

Assessing Cluster Variation

# User Guide

This section provides a comprehensive guide to running the visual basic program. It assumes the user is beginning on the opening page that appears when the solution is ran and details the many effects that different combinations of inputs will have on the solution. It also is aimed to walk the user through how to correctly set up and run the program, as well as how to interpret the results.

Section I: Guide Introduction

Upon starting up the program you should be taken to an initial form that introduces the program. The first page of the solution presents the user with the concept of data clustering. Afterwards, it gives a brief numbered explanation on how to run the program and how to manage your results.

Clicking next takes the user to a page which explains in a few sentences basics of the K-Means method. It then gives a quick listing of the pros and cons of this method meant to give the user a better idea of whether or not they should employ the K-Means method. Finally the page provides a sentence on what the data that program displays represents.

Clicking next a second time will take them to the K-Medoids page. The page is in the same format as the prior, with a quick introduction to the method and a listing of pros and cons, ending with an explanation of how the data is presented. In-depth descriptions of both these methods can be found in the Implementation Details section of this write-up.

Clicking next again will initiate the main part of the program.

Section II: The program

{{ INSERT PICTURE OF THE FINISHED PRGORAM HERE }}

Section III: Exported Data

How to export yo motha fukin data nergerrhhh