ORACLE
Certified Professional

Oracle Certified Professional Java Programmer

Java

Microsoft Certified Professional

sohailimran@yahoo.com

Sohail IMRAN سهيل عمران

Clustering

Ref: Web

Introduction

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation*, …)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Example: Classes → Students can be classified as Excellent, Very good, Good, Average and Fail. Not necassary that any student actually fails!

k-Means clustering is nondeterministic

The basic k-means clustering is based on a non-deterministic algorithm.

This means that running the algorithm several times on the same data could give different results, i.e. it may be possible that the centroid value for a given class may come out to be different at different instances of time for the same dataset.

types

1. Centroid-based Clustering

Centroid-based clustering organizes the data into non-hierarchical clusters, in contrast to hierarchical clustering defined below. k-means is the most widely-used centroid-based clustering algorithm. Centroid-based algorithms are efficient but sensitive to initial conditions and outliers.

2. Density-based Clustering

Density-based clustering connects areas of high example density into clusters. This allows for arbitrary-shaped distributions as long as dense areas can be connected. These algorithms have difficulty with data of varying densities and high dimensions. Further, by design, these algorithms do not assign outliers to clusters.

3. Distribution-based Clustering

This clustering approach assumes data is composed of distributions, such as Gaussian distributions. The distribution-based algorithm clusters data into three Gaussian distributions. As the distance from the distribution's center increases, the probability that a point belongs to the distribution decreases. The bands show a decrease in probability. When you do not know the type of distribution in your data, you should use a different algorithm.

4. Hierarchical Clustering

Hierarchical clustering creates a tree of clusters. Hierarchical clustering, not surprisingly, is well suited to hierarchical data, such as taxonomies. See Comparison of 61 Sequenced Escherichia coli Genomes by Oksana Lukjancenko, Trudy Wassenaar & Dave Ussery for an example. In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level.

k-Means Clustering

k-means algorithm is an iterative algorithm that tries to partition the dataset into 'k' pre-defined distinct non-overlapping subgroups or clusters where each data point belongs to only one group. It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different or as far as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster’s centroid is at the minimum. The less variation we have within clusters, the more homogeneous the data points are within the same cluster.

Why k-Means Clustering?

K-means converges in a finite number of iterations. Since the algorithm iterates a function whose domain is a finite set, the iteration must eventually converge.

The computational cost of the k-means algorithm is O(k*n*d), where n is the number of data points, k the number of clusters, and d the number of attributes.

Compared to other clustering methods, the k-means clustering technique is fast and efficient in terms of its computational cost.

It’s difficult to predict the optimal number of clusters or the value of k. To find the number of clusters, we need to run the k-means clustering algorithm for a range of k values and compare the results.

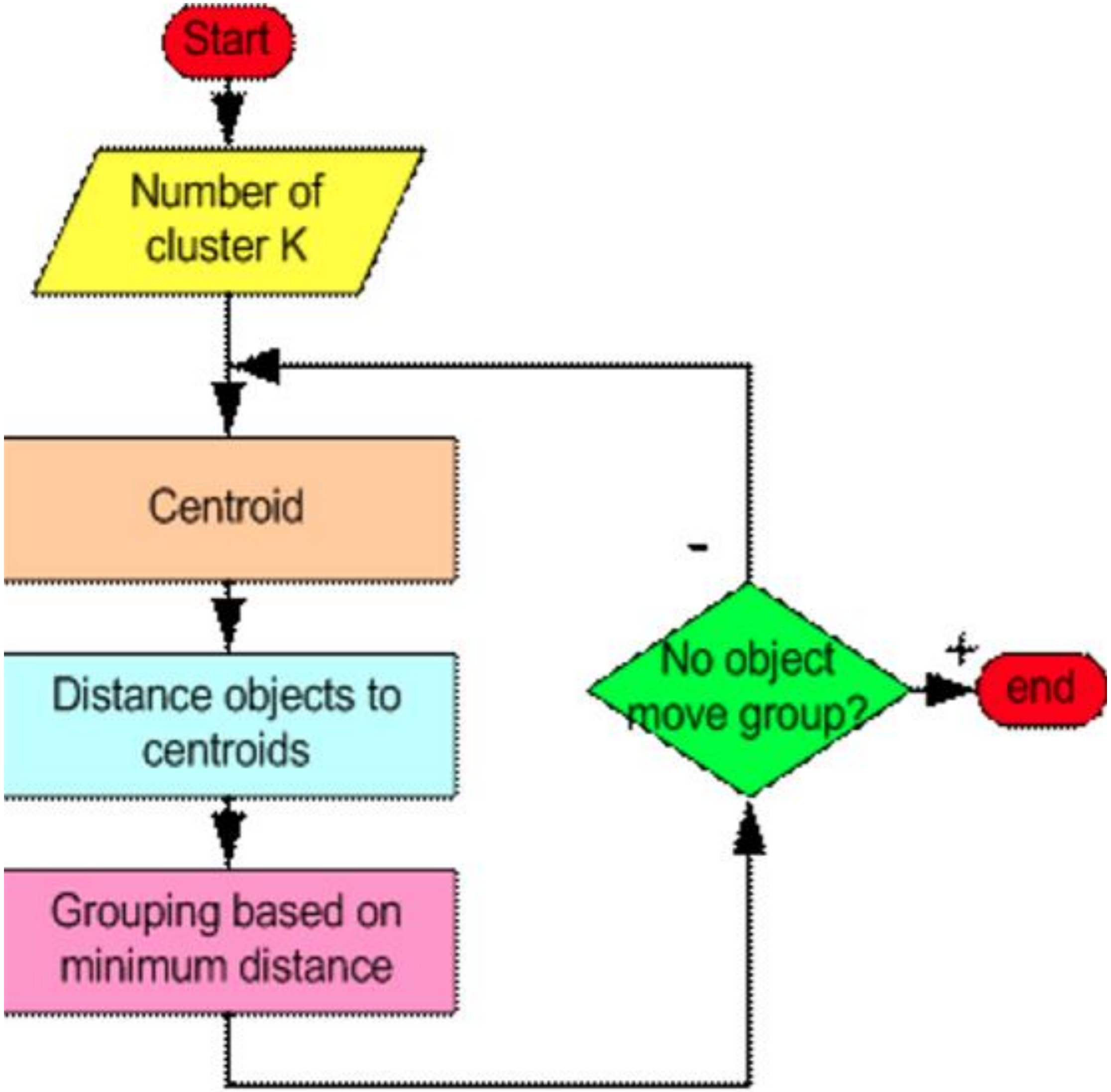
Centroids

In mathematics and physics, the centroid or geometric center of a plane figure is the arithmetic mean position of all the points in the figure. Informally, it is the point at which a cutout of the shape could be perfectly balanced on the tip of a pin.

What are Centroids in k-Means?

The k-means clustering algorithm attempts to split a given anonymous data set (a set containing no information as to class identity) into a fixed number (k) of clusters. Initially, the k number of so-called centroids are chosen. A centroid is a data point (imaginary or real) at the center of a cluster. Each centroid is an existing data point in the given input data set, picked at random, such that all centroids are unique (that is, for all centroids ci and cj, ci ≠ cj). These centroids are used to train a classifier. The resulting classifier is used to classify (using k = 1) the data and thereby produces an initial randomized set of clusters. Each centroid is thereafter set to the arithmetic mean of the cluster it defines. The process of classification and centroid adjustment is repeated until the values of the centroids stabilize. The final centroids will be used to produce the final classification/clustering of the input data, effectively turning the set of initially anonymous data points into a set of data points, each with a class identity.

How does k-Means clustering work?


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

steps

Step 1
Begin with a decision on the value of k = number of clusters.

Step 2
Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly, or systematically as the following:

1. Take the first k training sample as single- element clusters
2. Assign each of the remaining (N-k) training samples to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

Step 3
Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Step 4
Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

Advantages/disadvantages

Advantages/Features of k-Means

- Easy to implement
- With a large number of variables, K--Means may be computationally faster than hierarchical clustering (if K is small).
- k-Means may produce tighter clusters than hierarchical clustering
- An instance can change cluster (move to another cluster) when the centroids are re--computed

Disadvantages/Shortcomings of k-Means

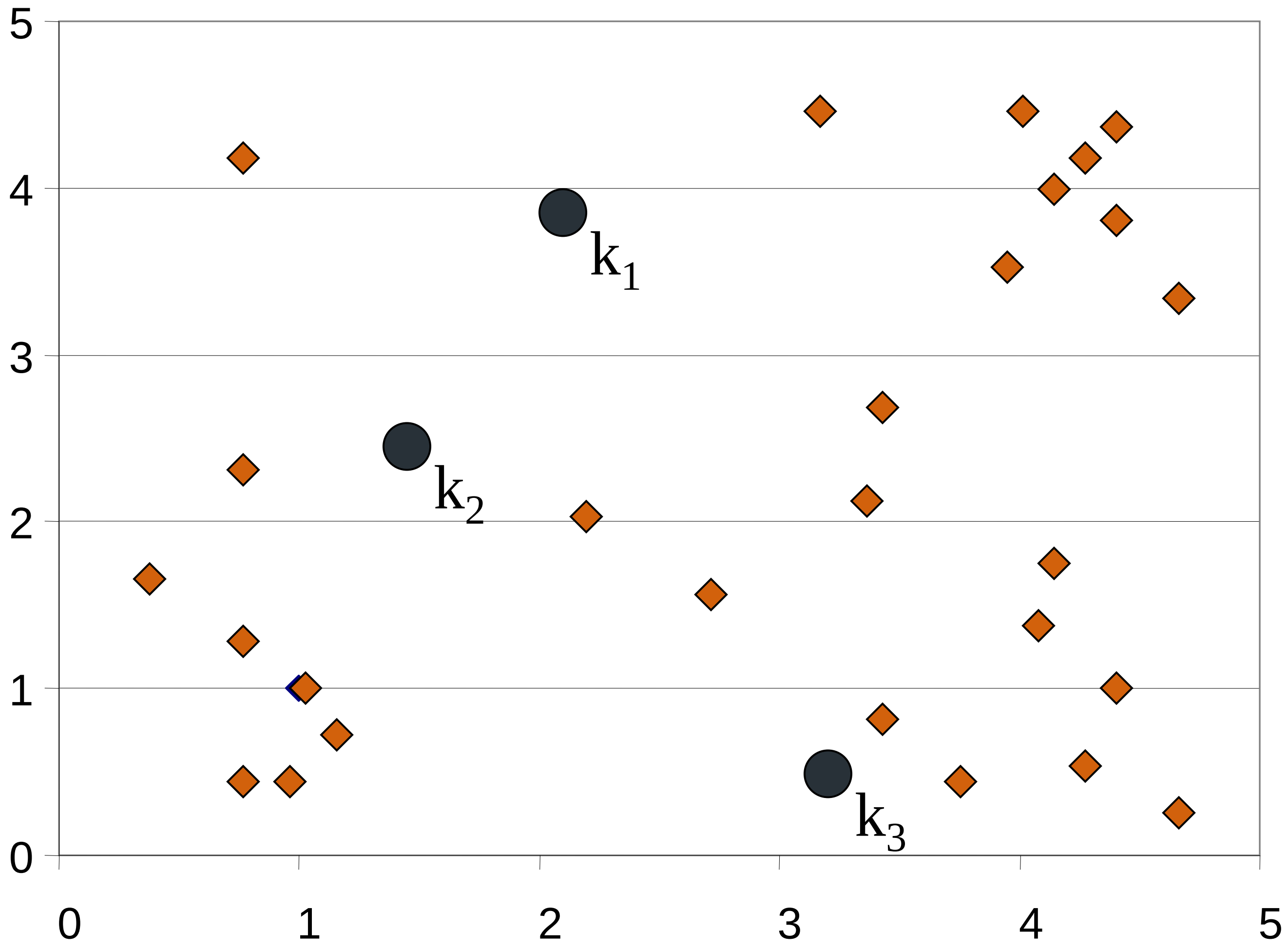
- Difficult to predict the number of clusters (K--Value)
- Initial seeds have a strong impact on the final results
- The order of the data has an impact on the final results
- Sensitive to scale: rescaling your datasets (normalization or standardization) will completely change results. While this itself is not bad, not realizing that you have to spend extra a4ention to scaling your data might be bad.

Applications

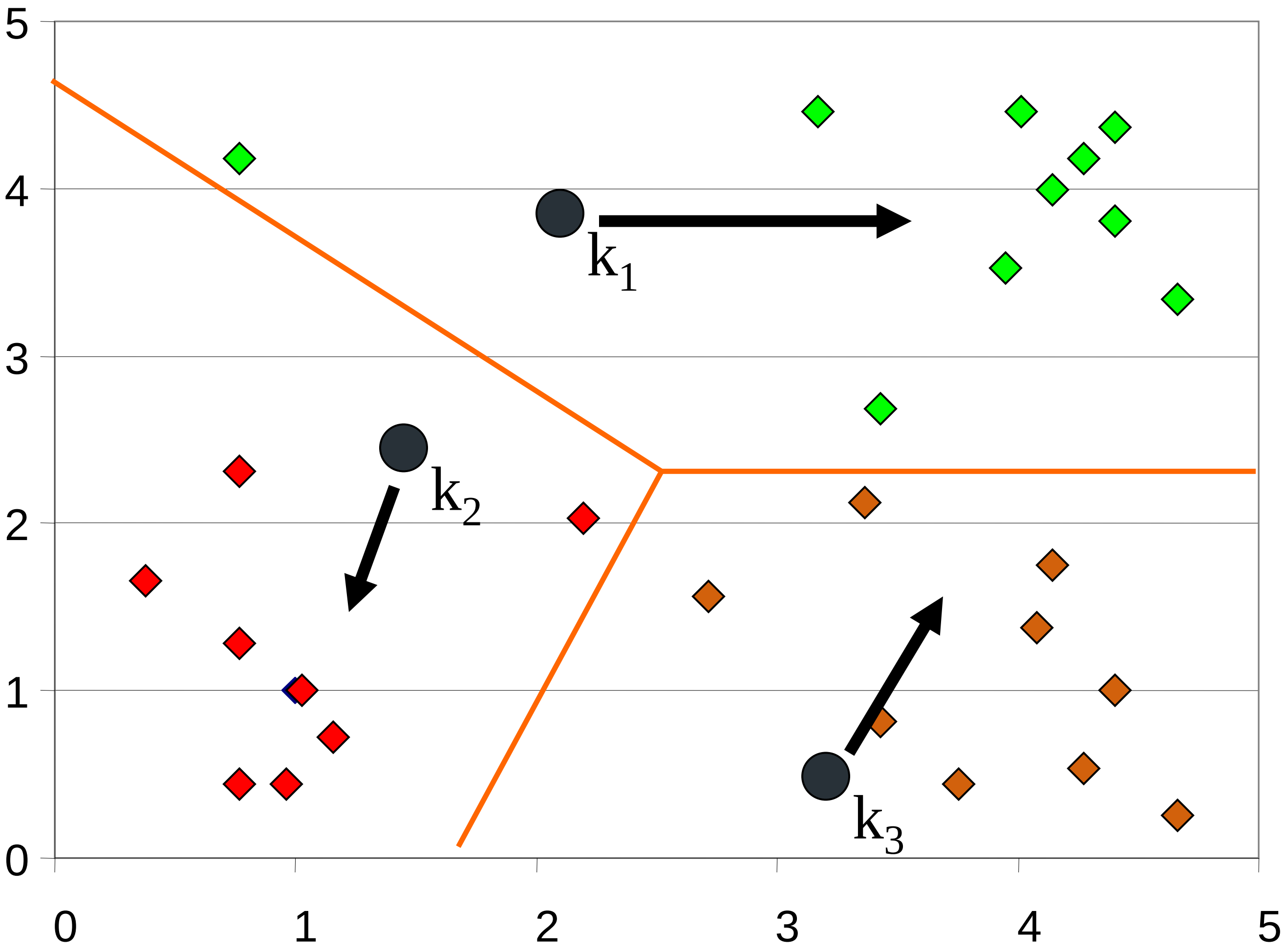
k-means can be applied to data that has a smaller number of dimensions, is numeric, and is continuous.

- Document Classification
- Delivery Store Optimization
- Identifying Crime Localities
- Customer Segmentation
- Fantasy League Stat Analysis
- Insurance Fraud Detection
- Rideshare Data Analysis
- Cyber-Profiling Criminals
- Call Record Detail Analysis
- Automatic Clustering of IT alerts

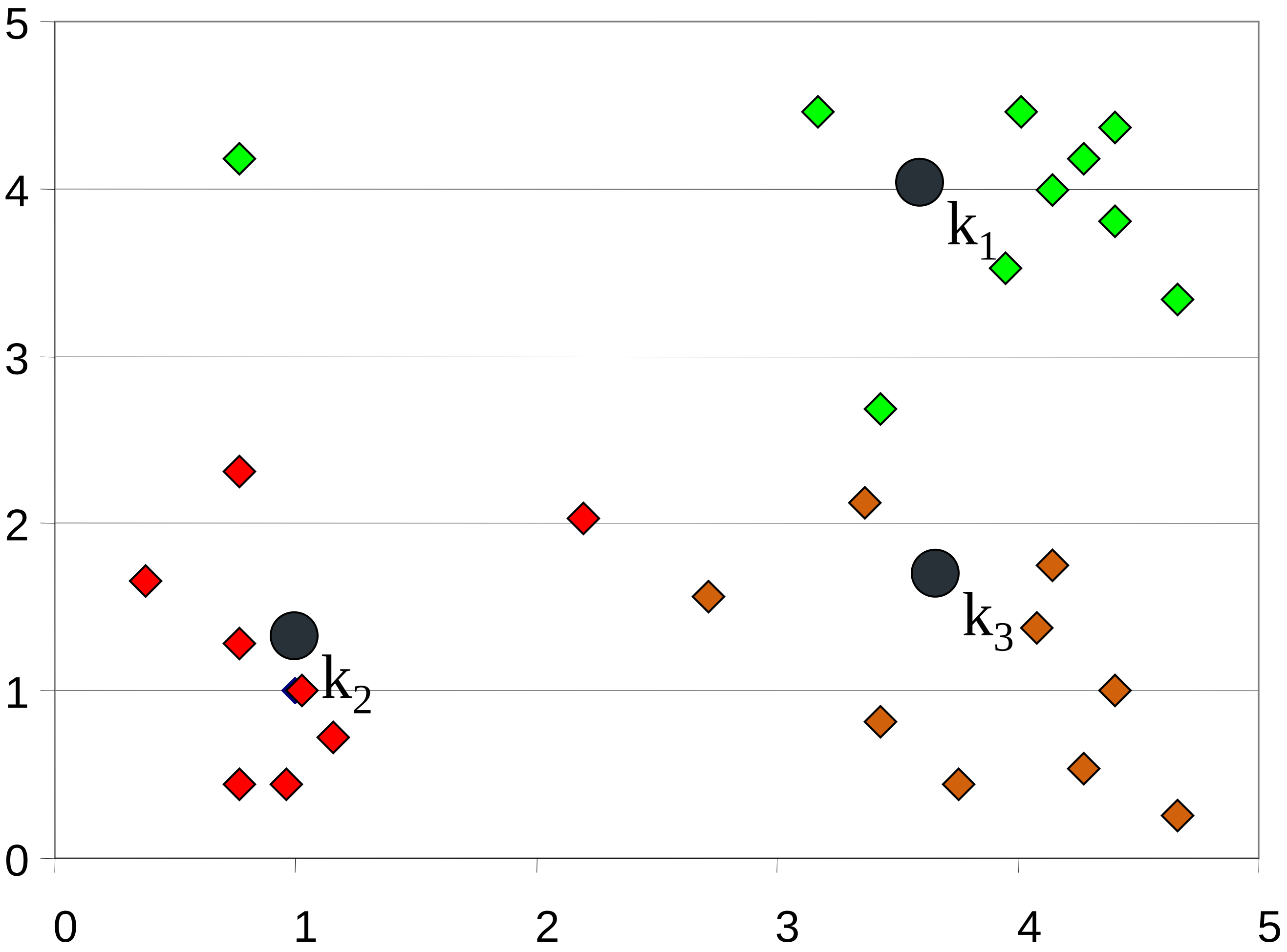
Step 1



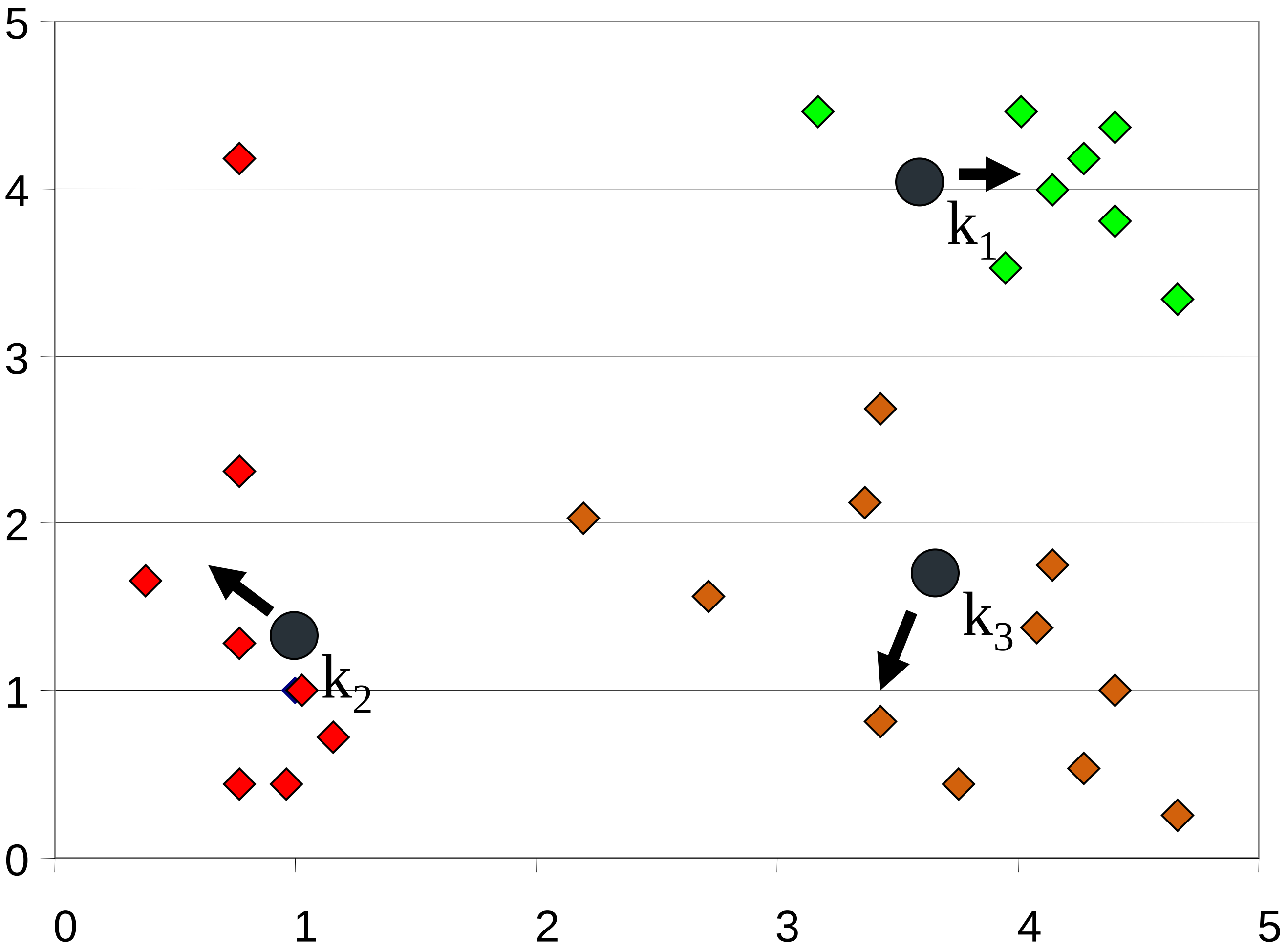
Step 2



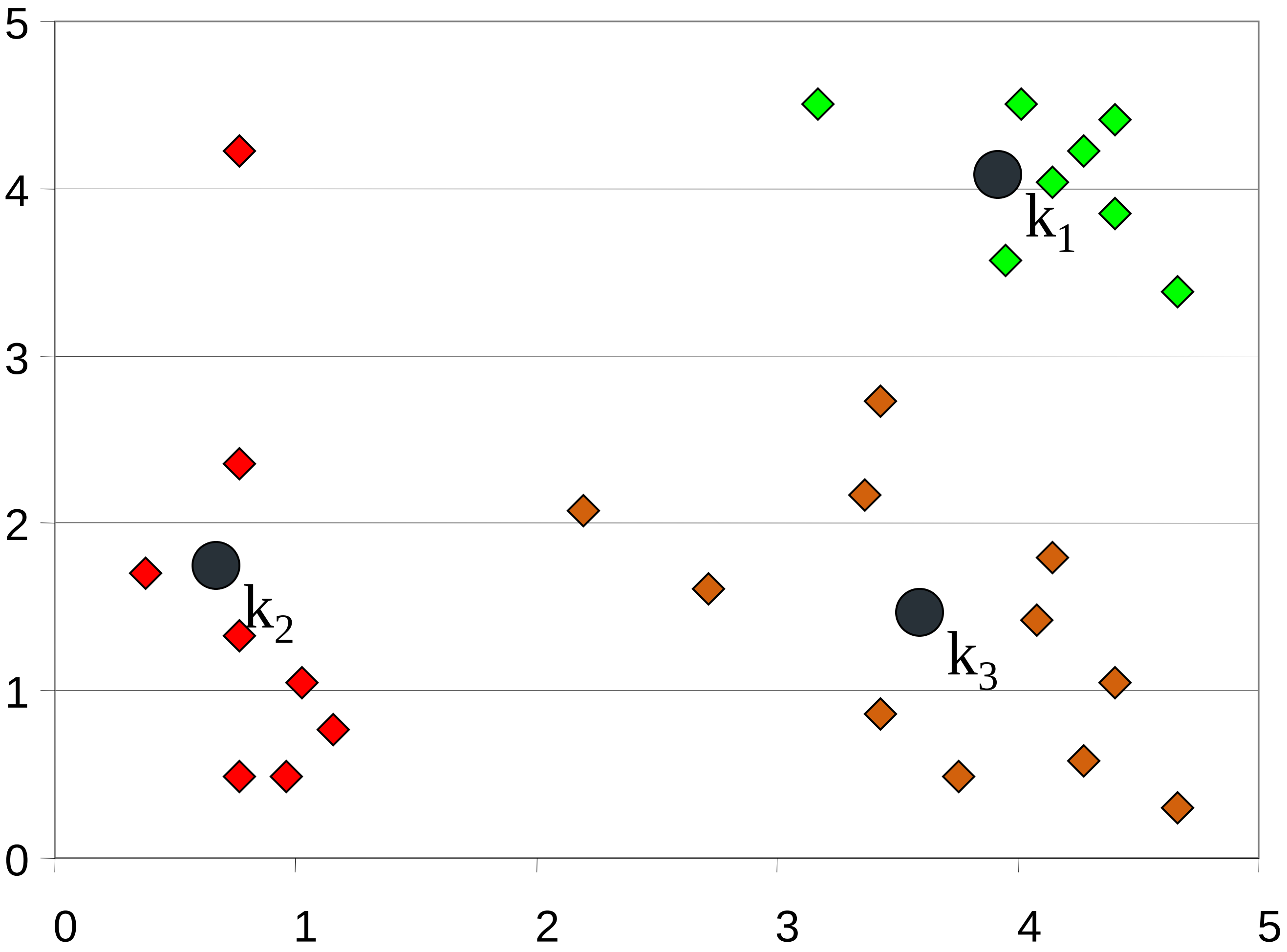
Step 3



Step 4



Step 5



1

Given the following dummy dataset with two clusters A and B.

Iteration 1

Data Points	X axis	Y axis
D1	2	0
D2	1	3
D3	3	5
D4	2	2
D5	4	6

Let D2 and D4 are the two Centroids.
C1 = D2 and **C2 = D4**

Let's start by finding the Euclidean Distance given by the formula: $\surd((x_2-x_1)^2 + (y_2-y_1)^2)$
Distance between D1 and C1 : $\surd((2-1)^2 + (0-3)^2$
$= \surd((1)^2 + (3)^2 \; = \surd(10) \; = 3.17$
Distance between D1 and C2 : $\surd((2-2)^2 + (0-2)^2$
$= \surd((0)^2 + (-2)^2 \; = \surd(4) \; = 2$

Similarly, let us find the distance between other points.
Seeing the above data, we can form the following two groups
Cluster 1: D1, D4
Cluster 2: D2, D3, D5

Data Points	Distance between C1 and other data points	Distance between C2 and other data points
D1	3.17	2.0
D3	2.83	3.17
D5	4.25	4.48

2

Now our next step is to find the **mean** values and the new centroid values with respect to the corresponding to the centroid values

Hence the centroid values are
Cluster 1: New C1 = (2.0, 1.0)
Cluster 2: New C2 = (2.67, 4.67)

Data Points	Mean value of data points along the x-axis	Mean value of data points along the y-axis
D1, D4	2.0 = (2+2)/2	1.0 = (2+0)/2
D2, D3, D5	2.67 = (1+3+4)/3	4.67 = (3+5+6)/3

Iteration 2
Based on the above data, we again calculate the euclidean distance for the new centroids.

Data Points	Distance between C1 and other data points	Distance between C2 and other data points
D1	1.0	4.72
D2	2.24	2.37
D3	4.13	0.47
D4	1	2.76
D5	5.39	1.89

From the above data, we infer the following
Cluster 1: (D1, D2, D4)
Cluster 2: (D3, D5)

n

Since the clusters have changed so we will now calculate the mean values again and the corresponding centroid values

Data Points	Mean value of data points along the x-axis	Mean value of data points along the y-axis
D1, D2, D4	1.67	1.67
D3, D5	3.5	5.5

So now the centroid values are:
Cluster 1: New C1 = (1.67, 1.67) =
Cluster 2: New C2 = (3.5, 5,5)

The above process has to be repeated until we find a constant value for centroids and the latest cluster will be considered as the final cluster solution.