What is Clustering?

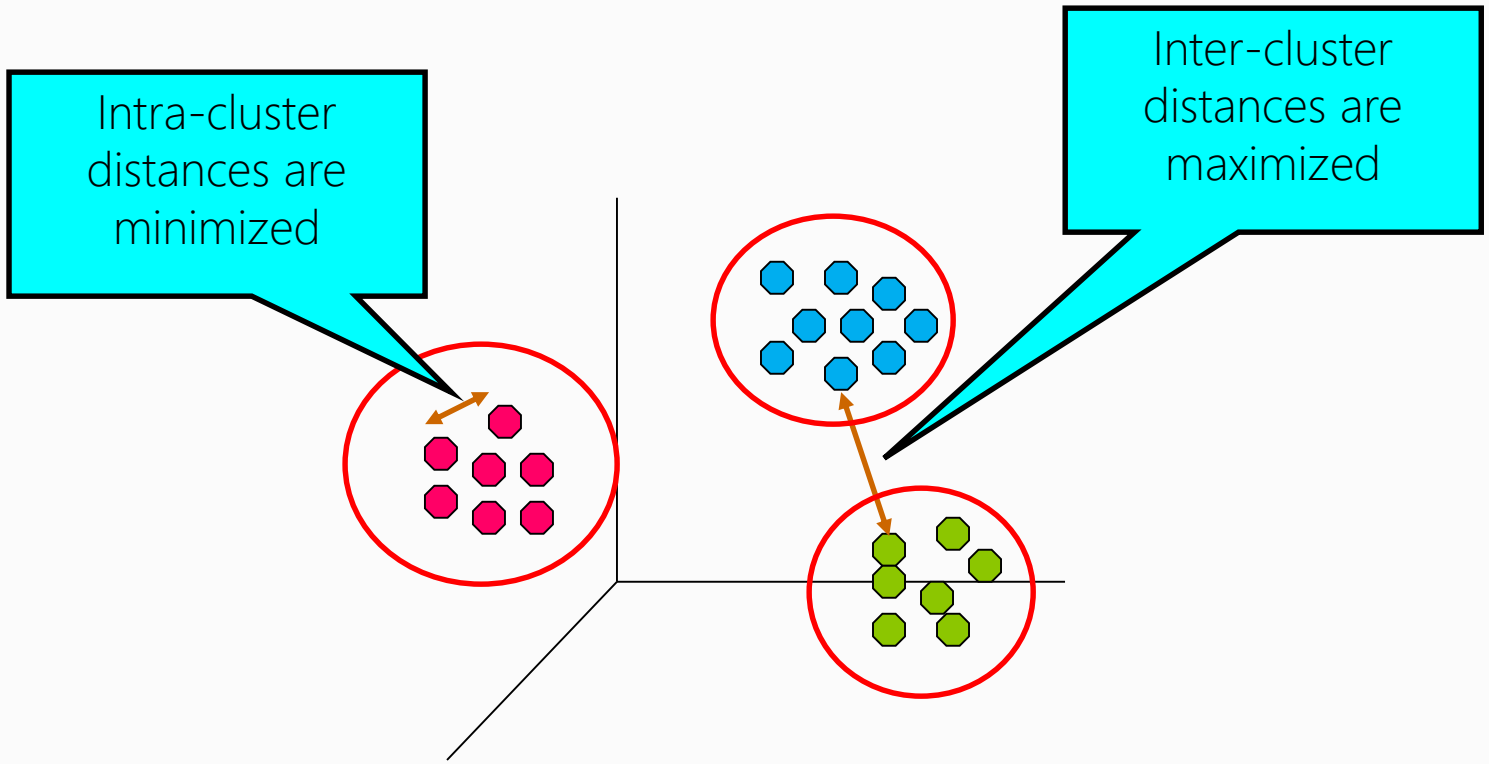
A way of grouping together data samples that are ***similar*** in some way - according to some criteria that you pick

A form of ***unsupervised learning*** – you generally don't have examples demonstrating how the data *should* be grouped together. *The most important unsupervised learning problem, imho…*

So, it's a method of ***data exploration*** – a way of looking for patterns or structure in the data that are of interest

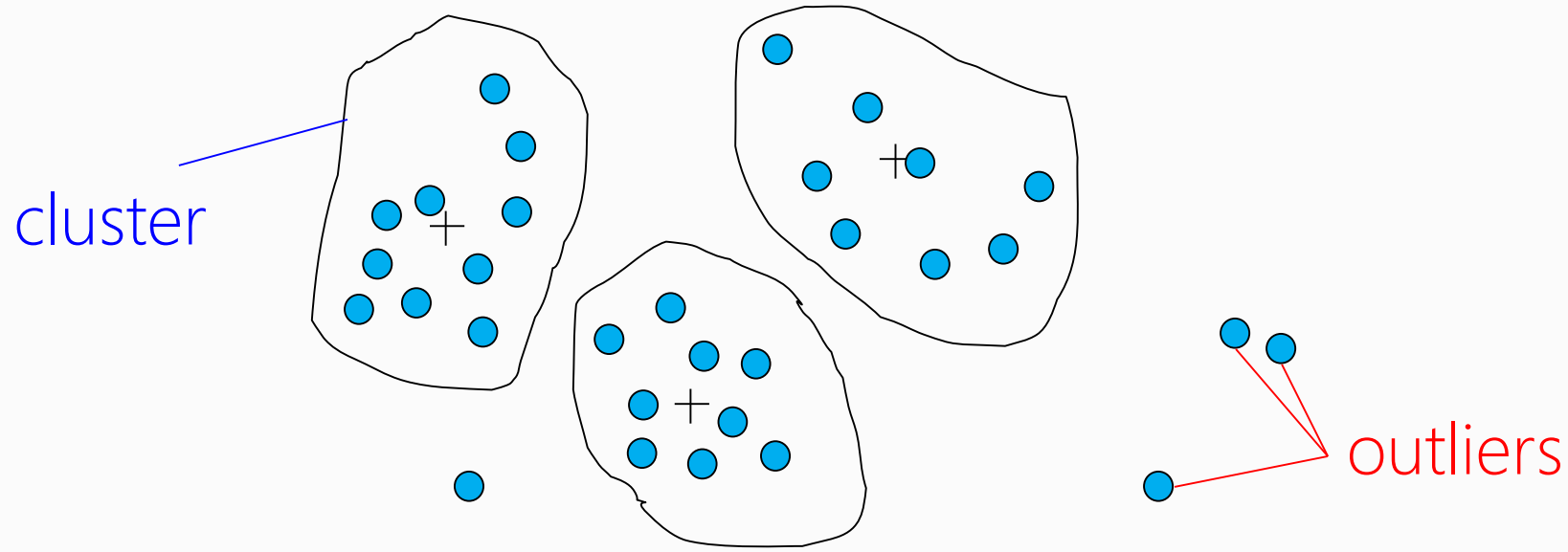
What is clustering?

A grouping of data objects such that the objects within a group are similar (or related) to one another and different from (or unrelated to) the objects in other groups



Outliers

- **Outliers** are **objects that do not belong to any cluster** or form clusters of very small cardinality



In some applications we are interested in discovering outliers, not clusters (**outlier analysis**)

How do we define "similarity"?

- Goal is to group together "similar" data – but what does this mean?
- No single answer, it depends on what we want to find or emphasize in the data; this is one reason why clustering is an "art"
- The similarity measure is often more important than the clustering algorithm used – don't overlook this choice!
- There is no golden standard, depends on goal: data reduction, "natural clusters", "useful" clusters, outlier detection, etc.

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

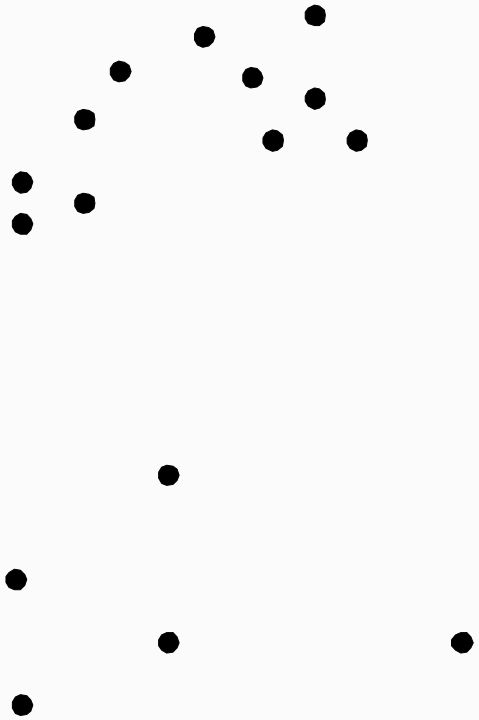
Two Clusters

Four Clusters

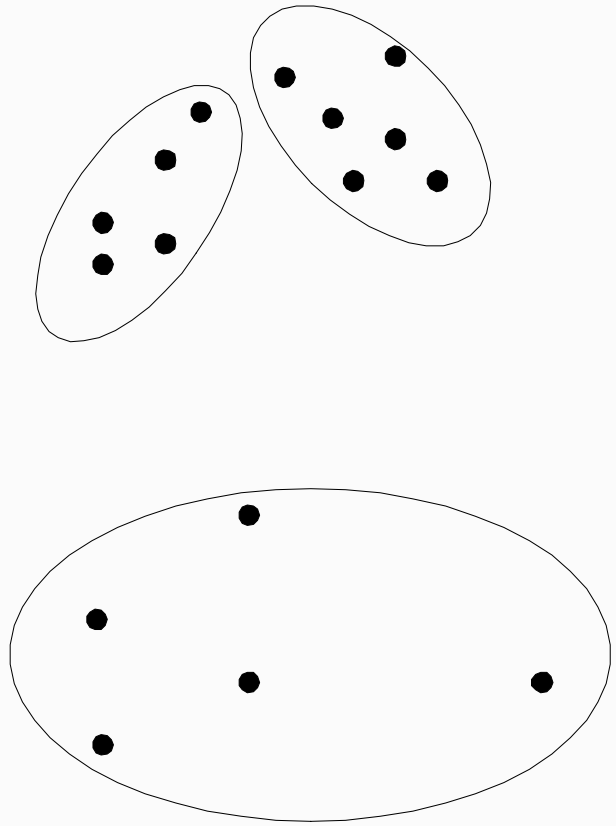
Types of Clusterings

- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

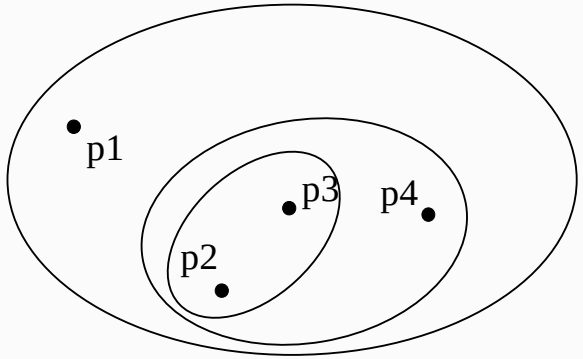


Original Points

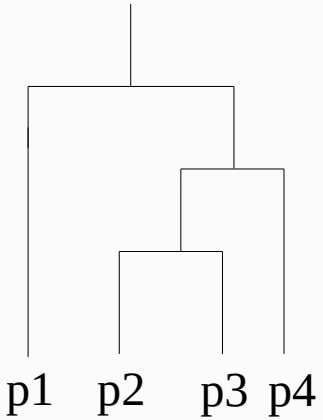


A Partitional Clustering

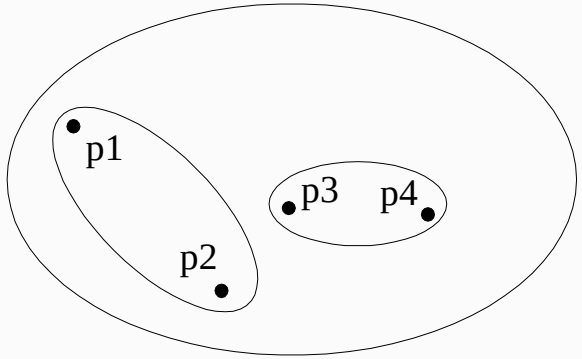
Hierarchical Clustering



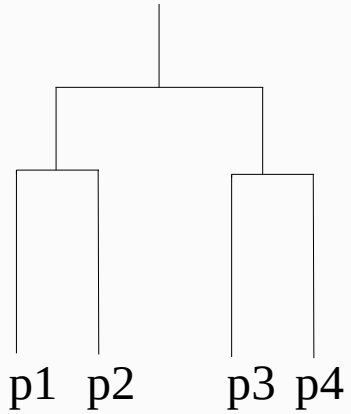
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



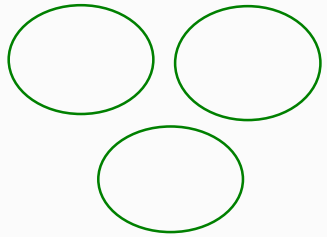
Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

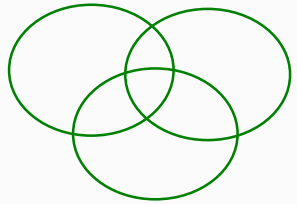
- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or 'border' points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Cluster of widely different sizes, shapes, and densities

Clustering Design Space

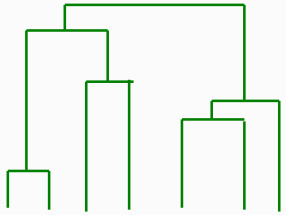
Non overlapping



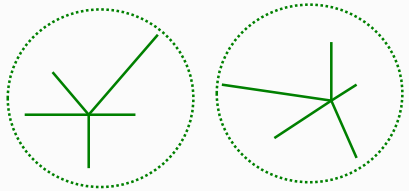
Overlapping



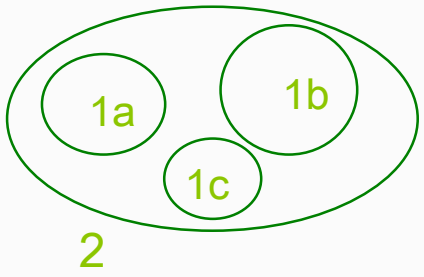
Hierarchical



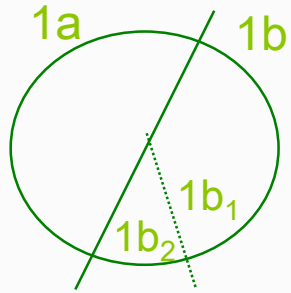
Non-hierarchical

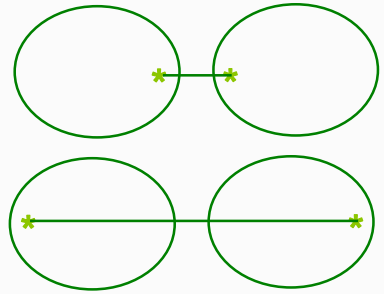


Agglomerative



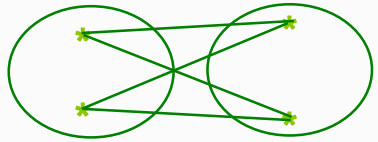
Divisive



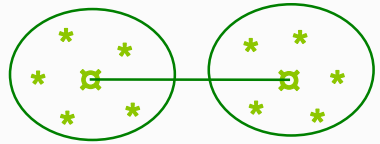


Single Linkage:
Minimum distance

Complete Linkage:
Maximum distance



Average Linkage:
Average distance



Centroid method:
Distance between centres

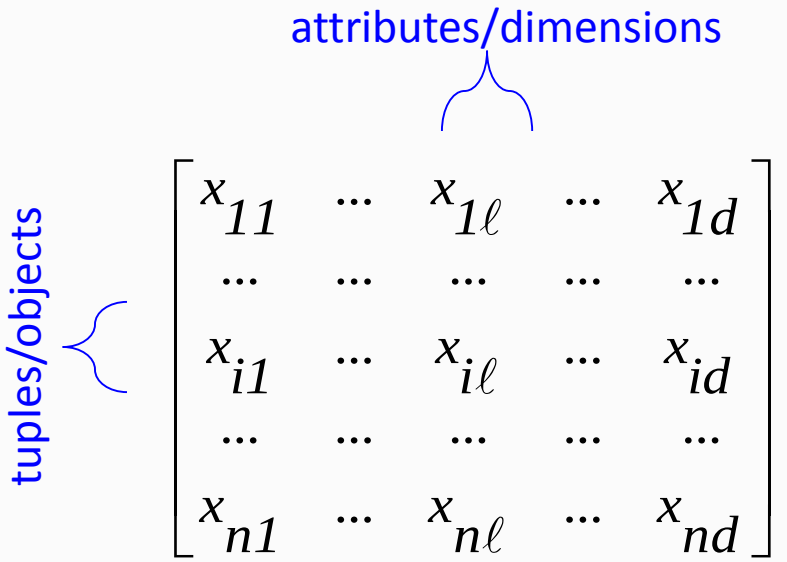
Wards method:
Minimization of within-cluster variance

Distance Functions

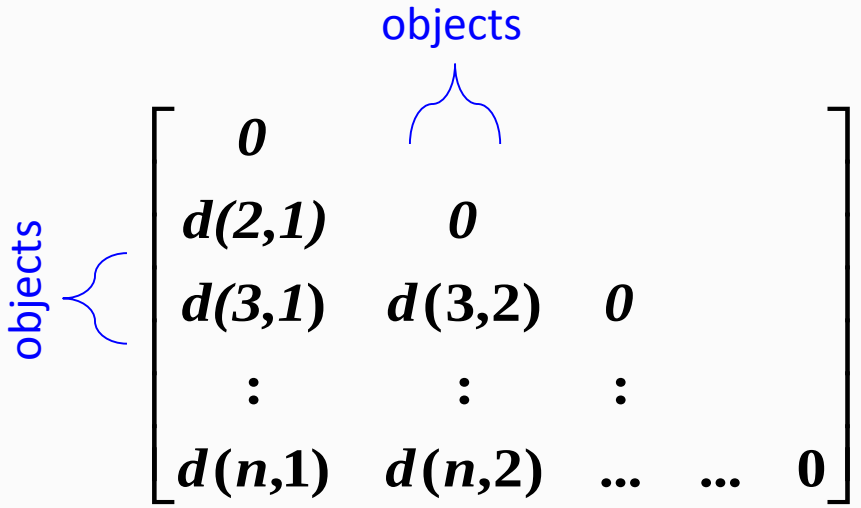
- The distance $d(x, y)$ between two objects x and y is a metric if
  - $d(i, j) \geq 0$ (non-negativity)
  - $d(i, i) = 0$ (isolation)
  - $d(i, j) = d(j, i)$ (symmetry)
  - $d(i, j) \leq d(i, h)+d(h, j)$ (triangular inequality)
- The definitions of distance functions are usually different for real, boolean, categorical, and ordinal variables.
- Weights may be associated with different variables based on applications and data semantics.

Data Structures

- *data* matrix



- *Distance* matrix



Distance functions for real-valued vectors

- If p = 2, L_2 is the **Euclidean distance**:

$$d(x,y)=\sqrt{(|x_1-y_1|^2+|x_2-y_2|^2+...+|x_d-y_d|^2)}$$

- Also one can use **weighted distance**:

$$d(x,y)=\sqrt{(w_1|x_1-x_1|^2+w_2|x_2-x_2|^2+...+w_d|x_d-y_d|^2)}$$

$$d(x,y)=w_1|x_1-y_1|+w_2|x_2-y_2|+...+w_d|x_d-y_d|$$

Distance functions for binary vectors

- Jaccard similarity between binary vectors X and Y

$$JSim(X,Y) = \frac{X \cap Y}{X \cup Y}$$

- Jaccard distance between binary vectors X and Y

  Jdist(X,Y) = 1- JSim(X,Y)

- Example:
  - JSim = 1/6
  - Jdist = 5/6

	Q1	Q2	Q3	Q4	Q5	Q6
X	1	0	0	1	1	1
Y	0	1	1	0	1	0

Distance functions for real-valued vectors

- L_p norms or *Minkowski distance*:

$$L_p(x,y) = \left(|x_1 - y_1|^p + |x_2 - y_2|^p + \ldots + |x_d - x_d|^p \right)^{1/p} = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

  where *p* is a positive integer

- If p = 1, L_1 is the *Manhattan (or city block)* distance:

$$L_1(x,y) = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_d - y_d| = \sum_{i=1}^{d} |x_i - y_i|$$

K-means Clustering

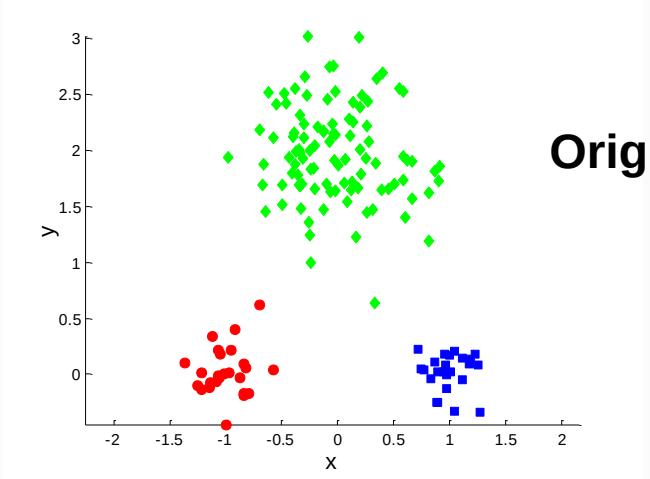
- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

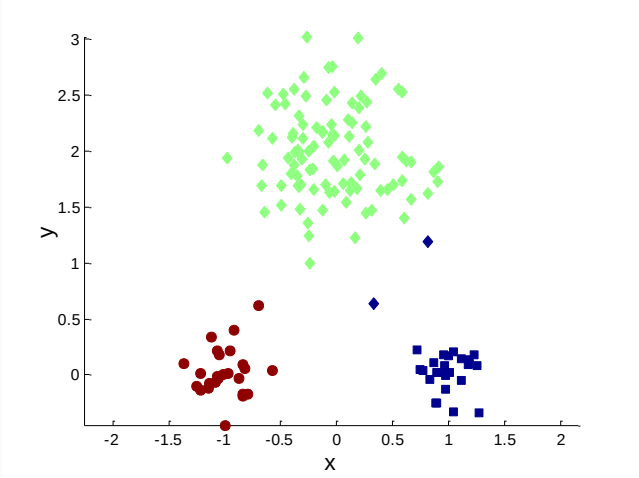
K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- 'Closeness' is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to 'Until relatively few points change clusters'
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

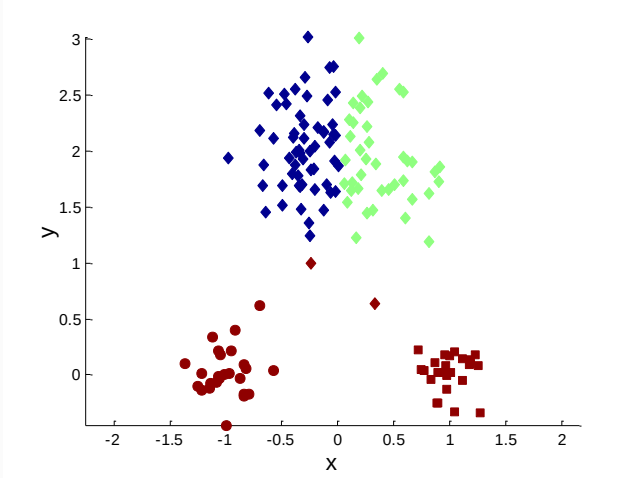
Two different K-means Clusterings



Original Points

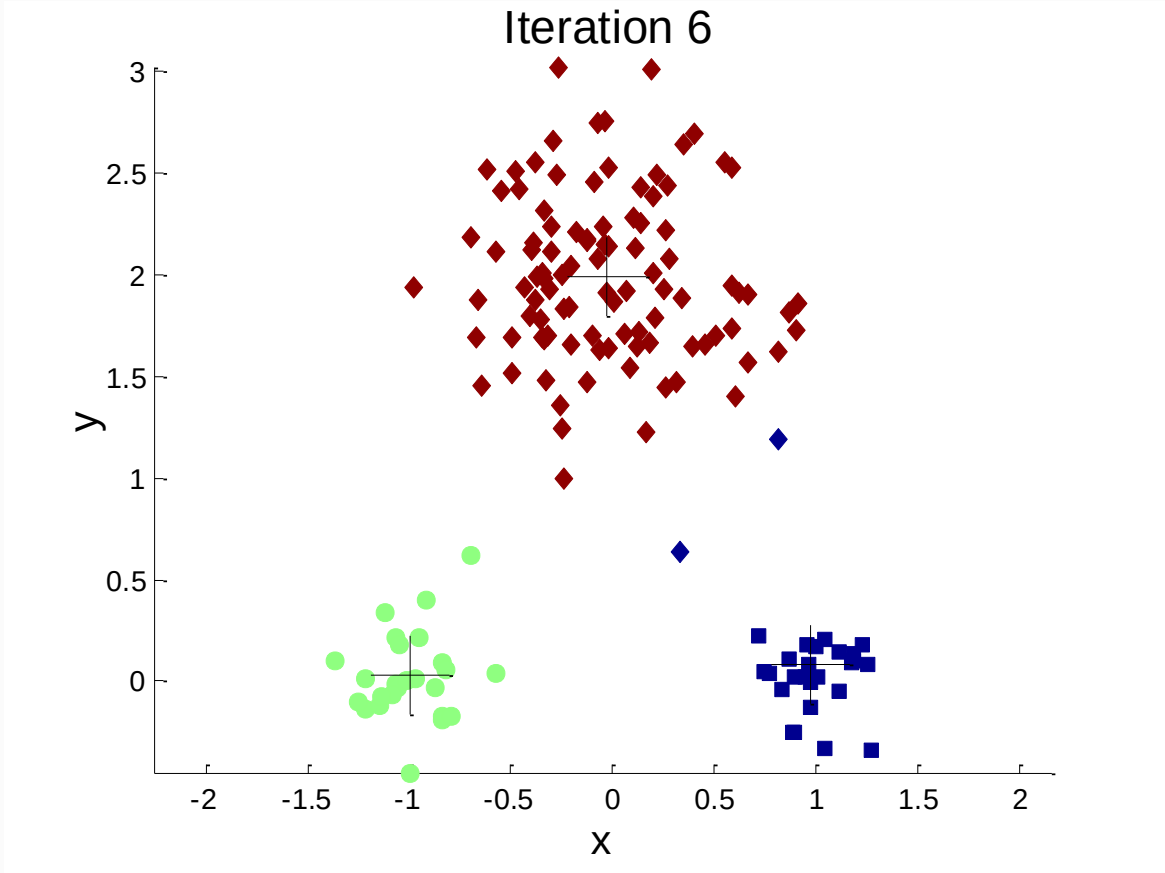


Optimal Clustering

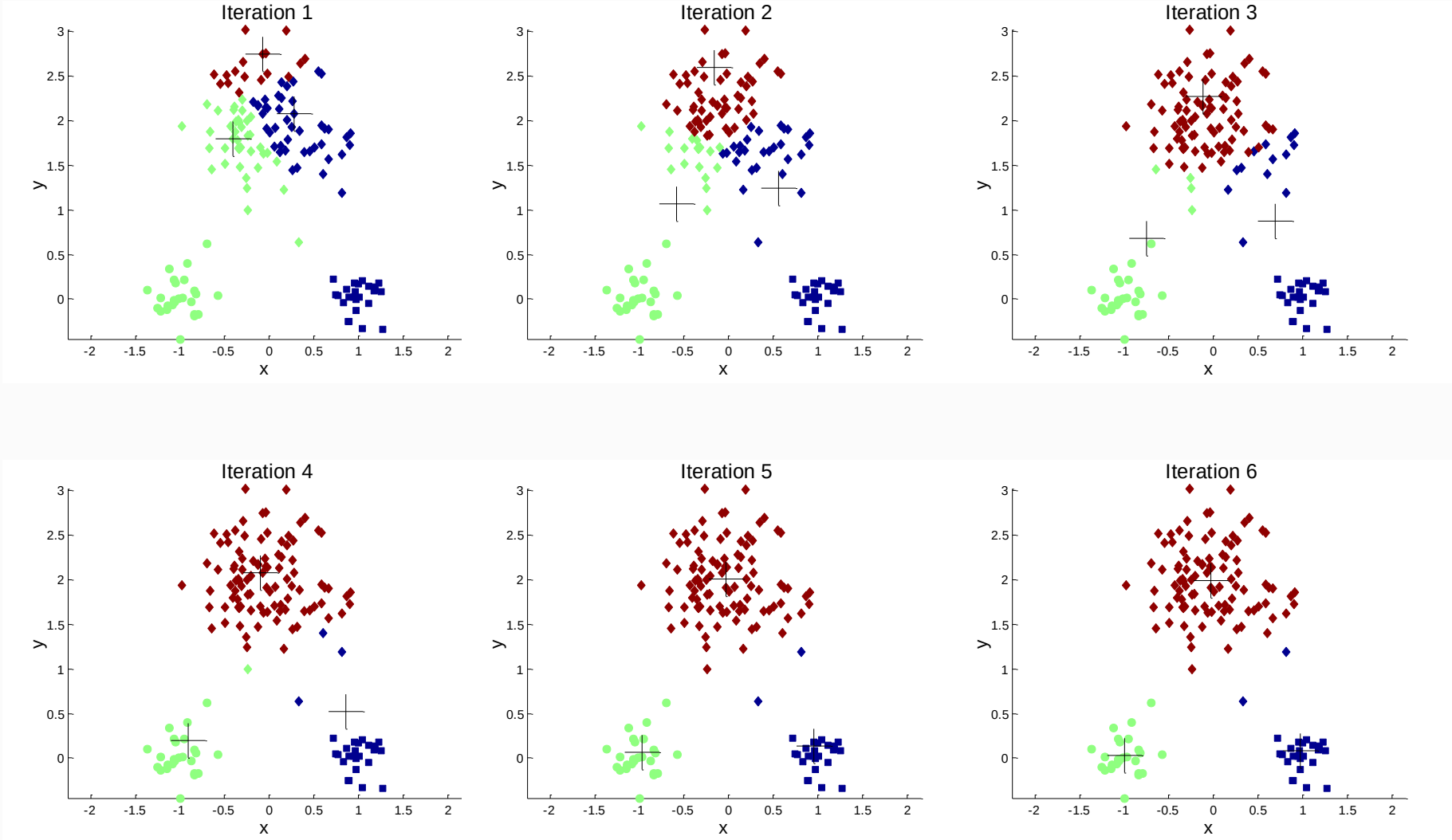


Sub-optimal Clustering

Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



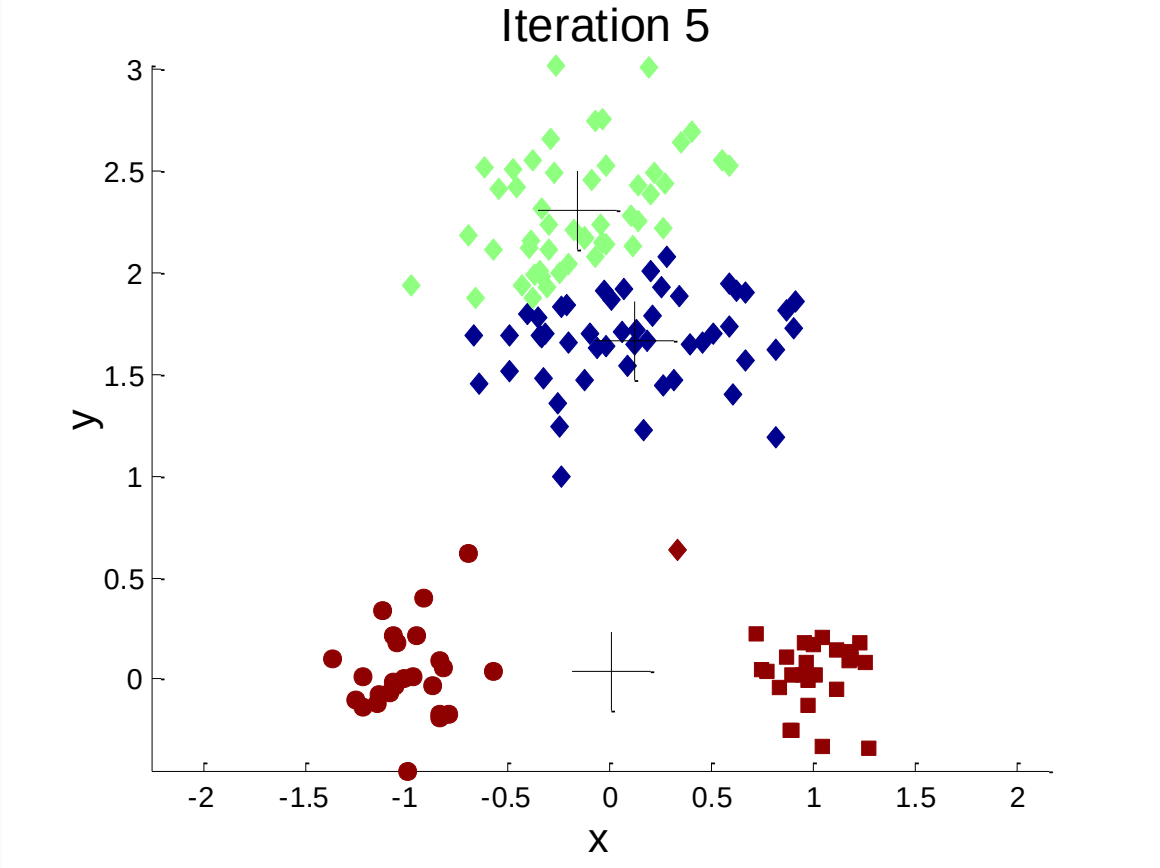
Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

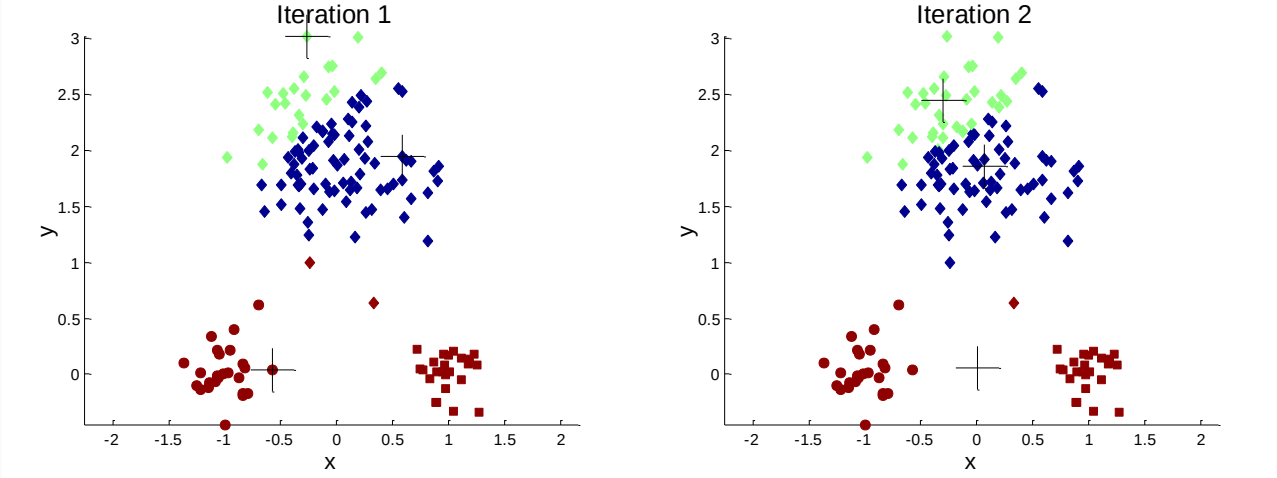
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

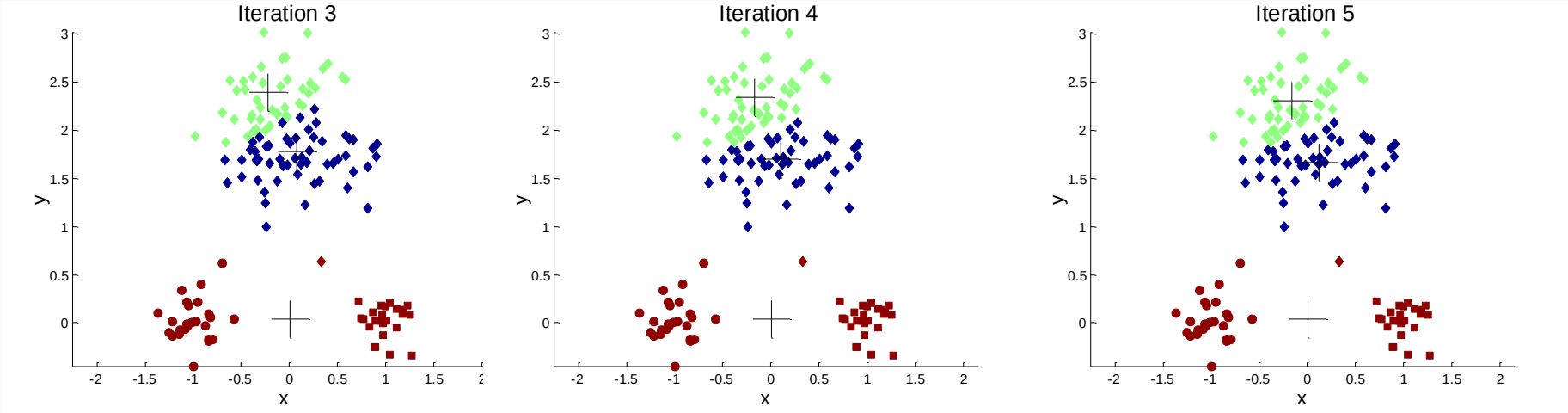
  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …





Problems with Selecting Initial Points

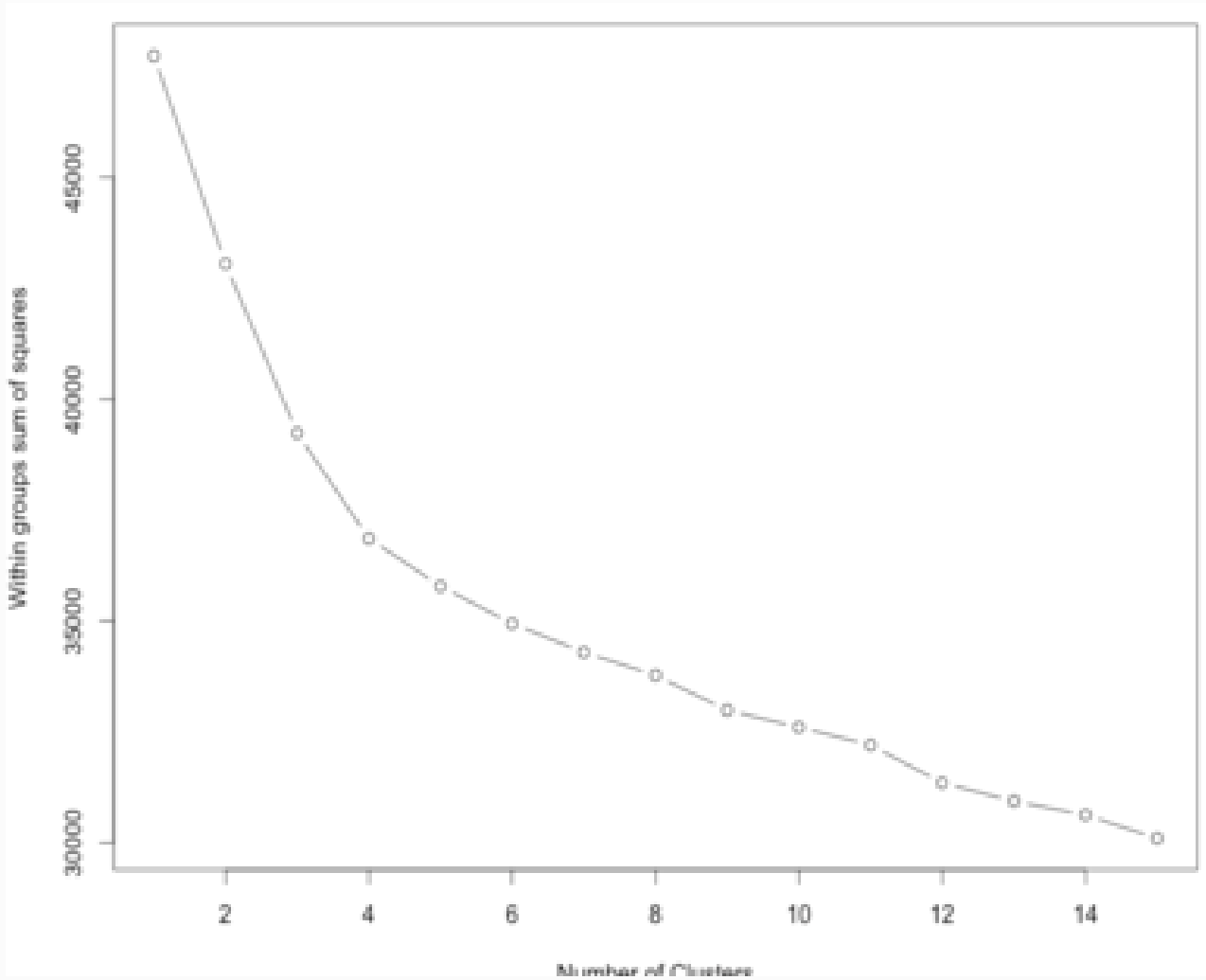
- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when K is large
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

  - For example, if K = 10, then probability = $10!/10^{10} = 0.00036$
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- *Do **Scree plots** with different seeds!!!*
  - *Pick the best K at the elbows*



K-Means Clustering Discussion
Non-numeric data

Feature values are not always numbers

- Example
  - Boolean Values: Yes or no, presence or absence of an attribute
  - Categories: Colors, educational attainment, gender

How do these values factor into the computation of distance?

K-Means Clustering Discussion

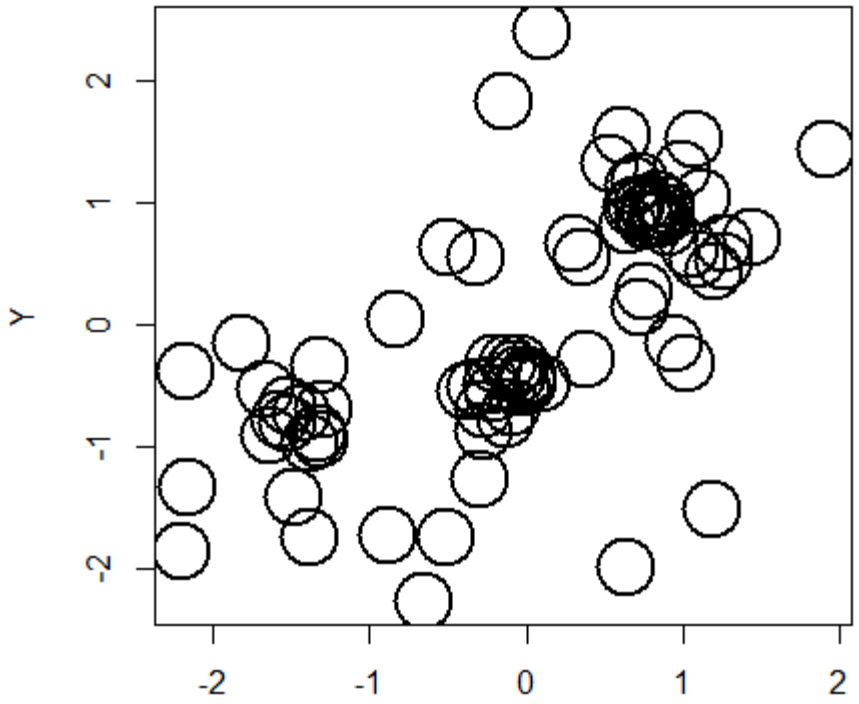
Dealing with non-numeric data

- Boolean values => convert to 0 or 1
  - Applies to yes-no/presence-absence attributes
- Non-binary characterizations
  - Use natural progression when applicable; e.g., educational attainment: GS, HS, College, MS, PHD => 1,2,3,4,5
  - Assign arbitrary numbers but be careful about distances; e.g., color: red, yellow, blue => 1,2,3
- How about unavailable data?
  (0 value not always the answer)

K-Means Clustering Discussion
Preprocessing your dataset

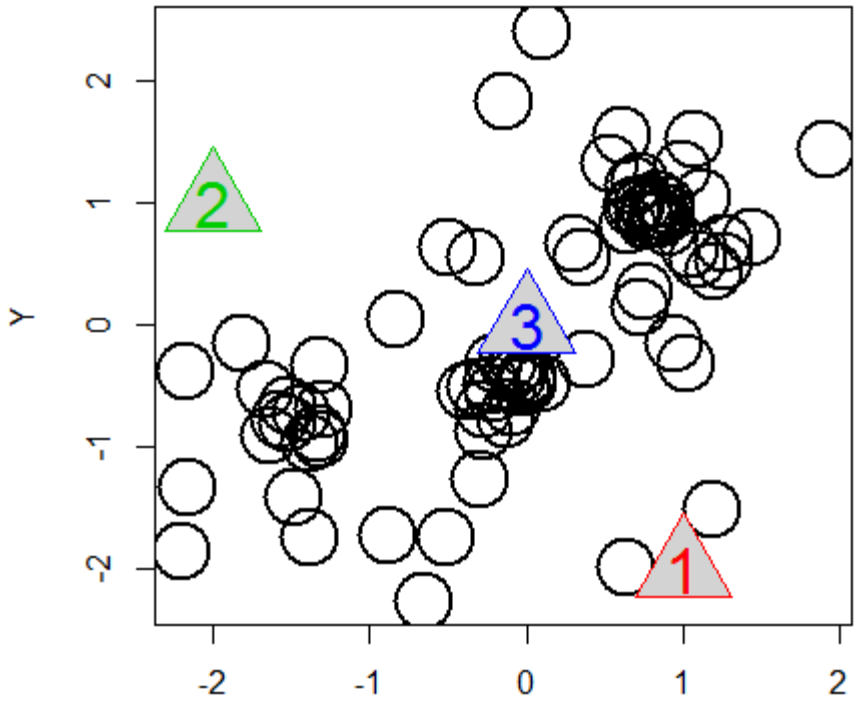
- Dataset may need to be preprocessed to ensure more reliable data mining results
- Conversion of non-numeric data to numeric data
- Calibration of numeric data to reduce effects of disparate ranges
  - Particularly when using the Euclidean distance metric

K-Means Clustering (0)



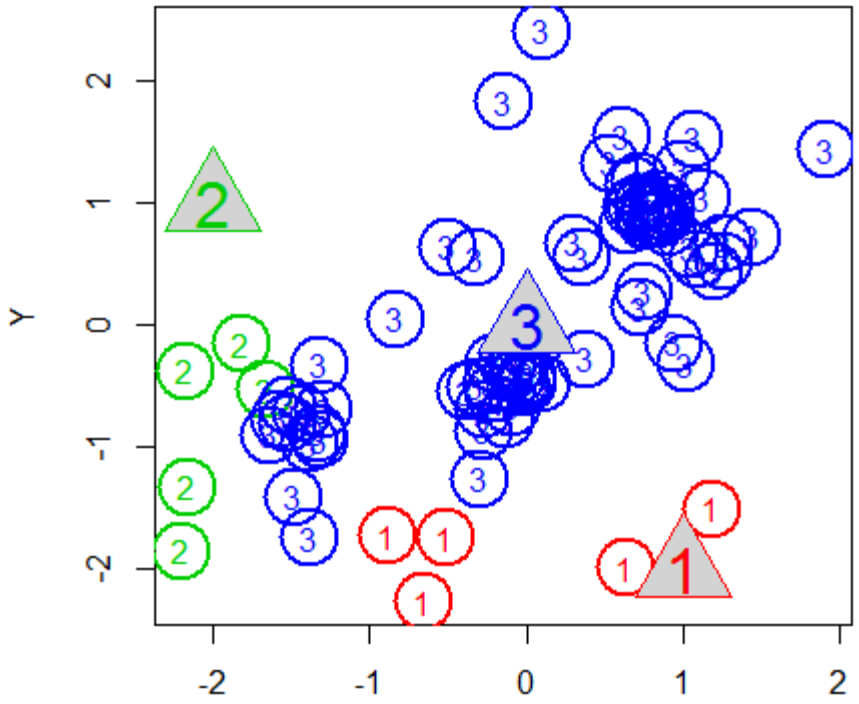
- Clustering starts by getting the data and representing the data as points in space. In this example the space is 2-dimensional.
- Each point describes an observation. An observation is an individual item.
- The dimensions are attributes that describe the item.

K-Means Clustering (1)



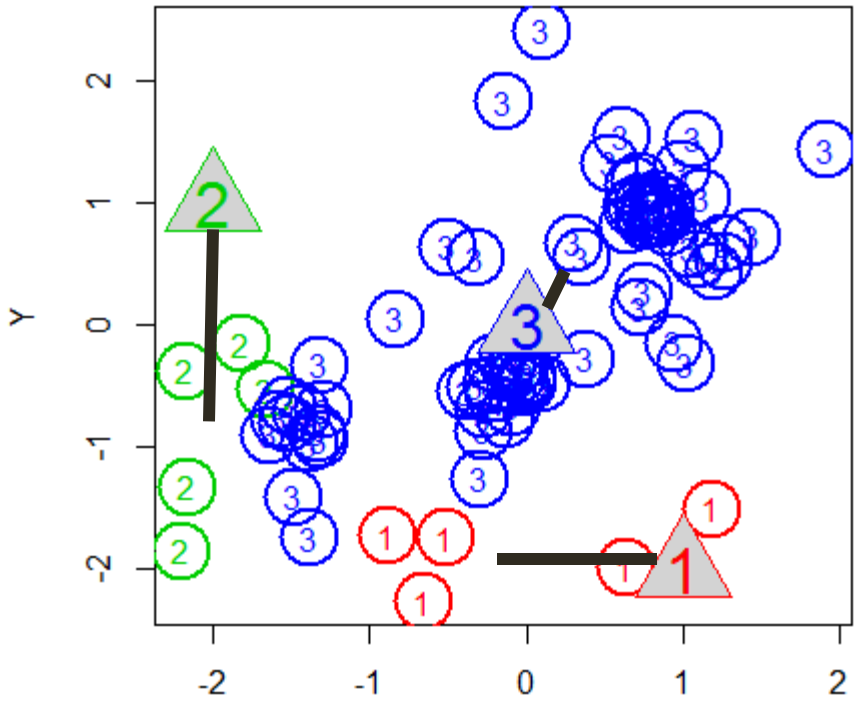
- Clustering continues by guessing, presuming, or specifying a number of clusters.
- Each centroid represents a cluster.
- The centroid positions are determined randomly. The centroids should be within the bounds of the points.

K-Means Clustering (2)



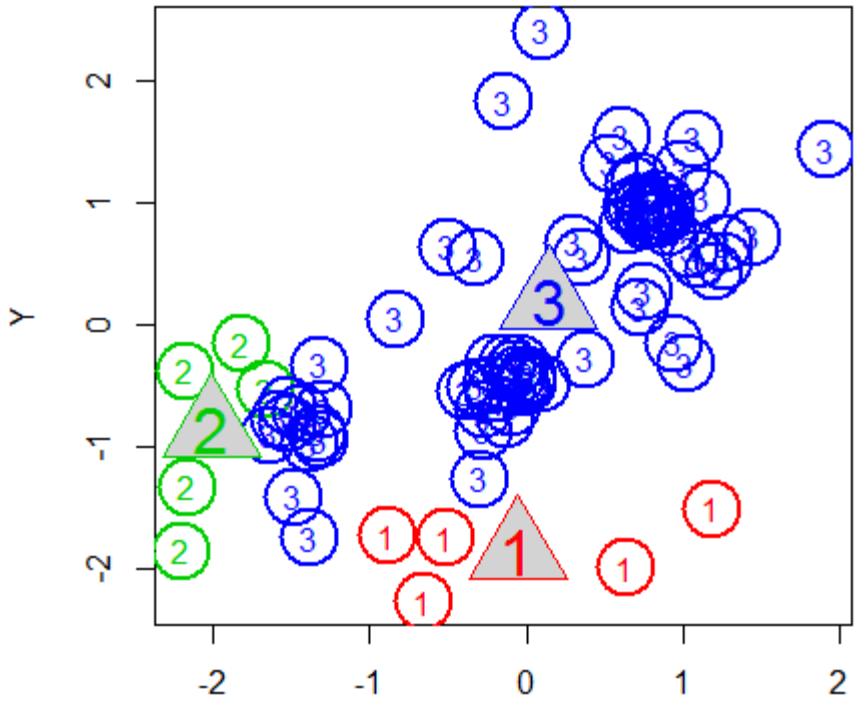
- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

K-Means Clustering (2)



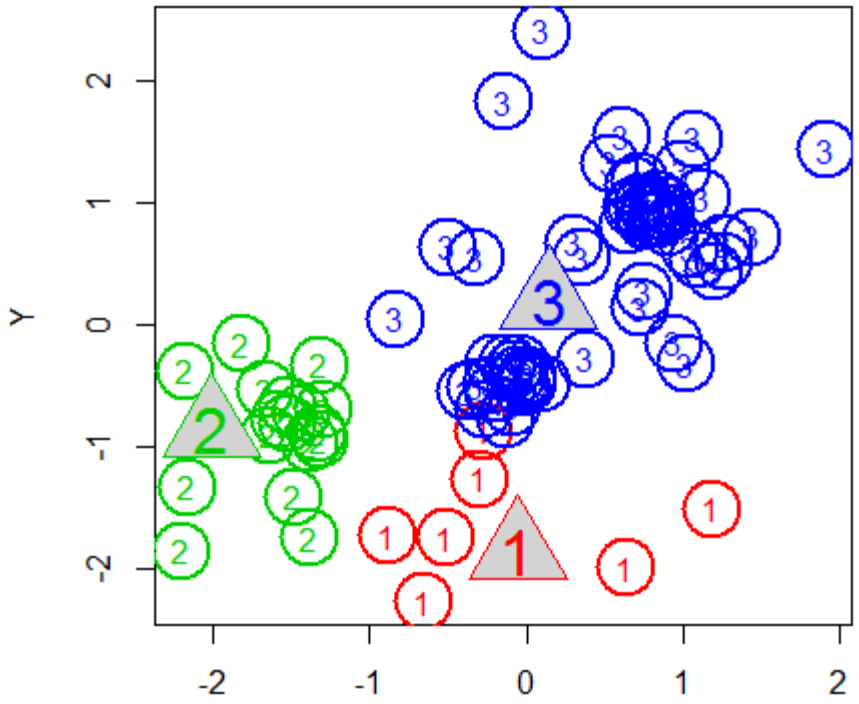
- Clustering continues by moving each centroid to the center of its cluster.

K-Means Clustering (3)



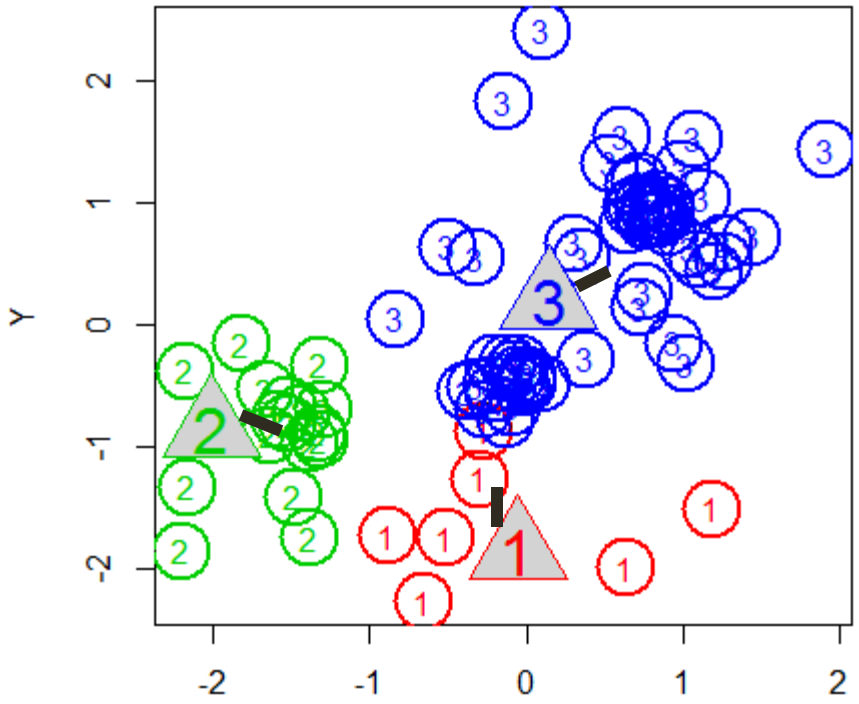
- Clustering continues by moving each centroid to the center of its cluster.

K-Means Clustering (4)



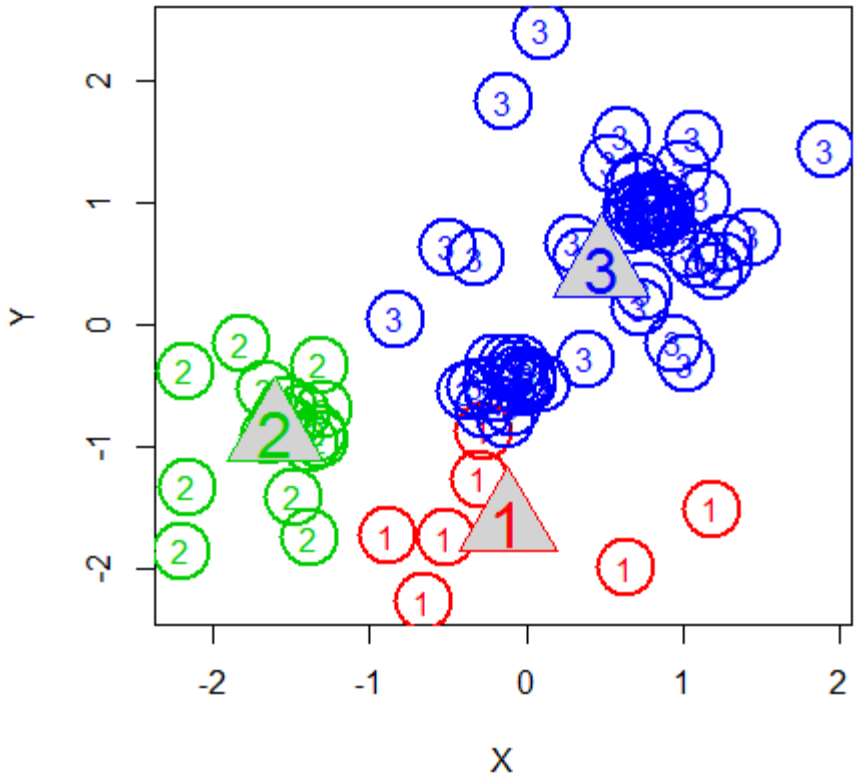
- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

K-Means Clustering (4)

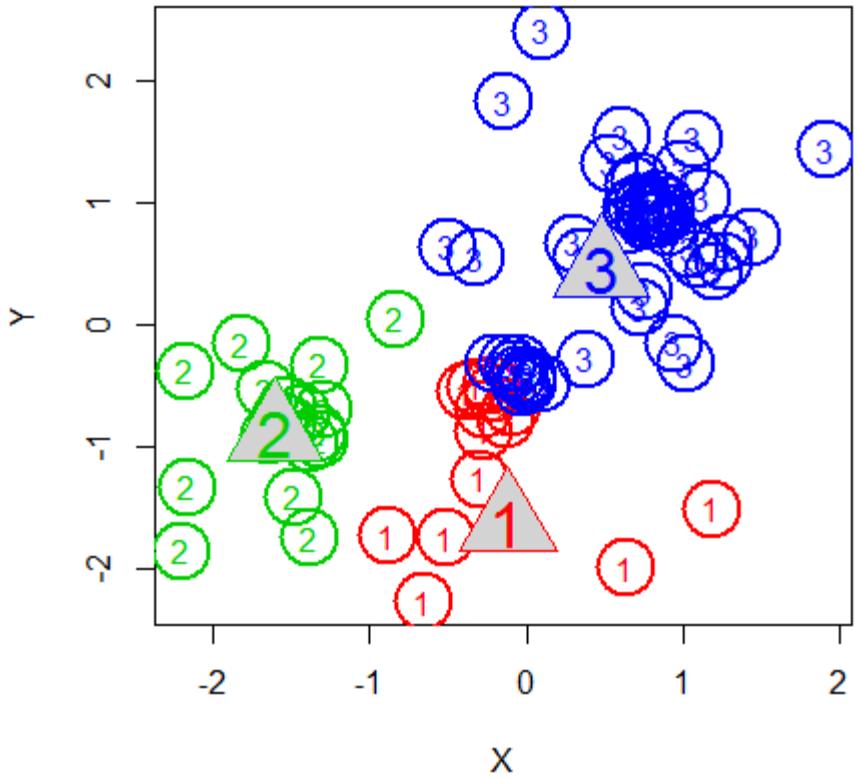


- Clustering continues by assigning each point to a cluster.
- For each point, the algorithm measures the distance to each centroid.
- For each point, the smallest distance to a centroid indicates the assignment.

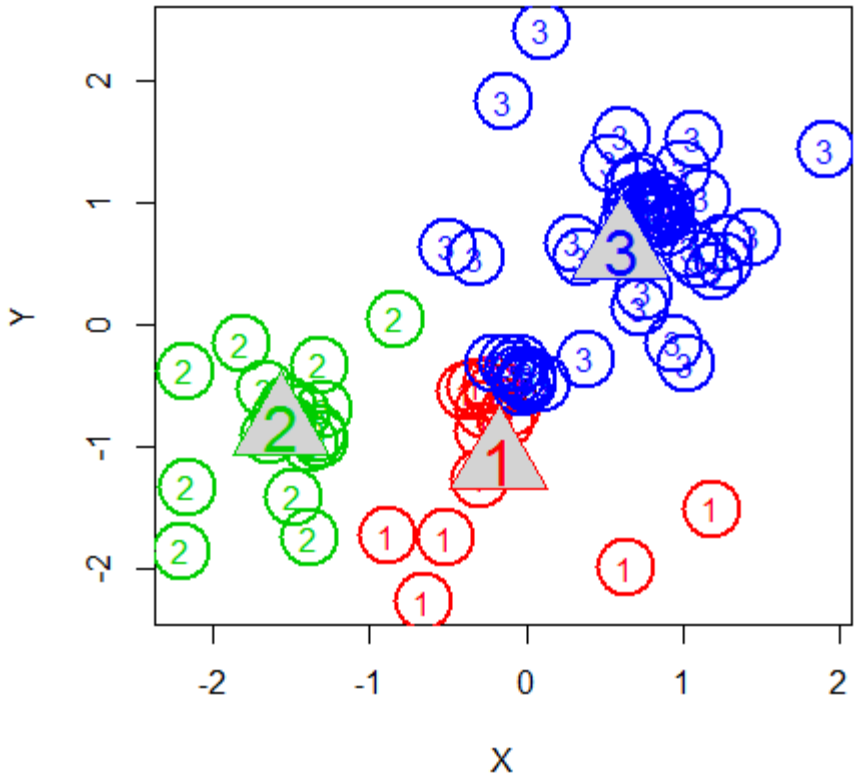
K-Means Clustering (5)



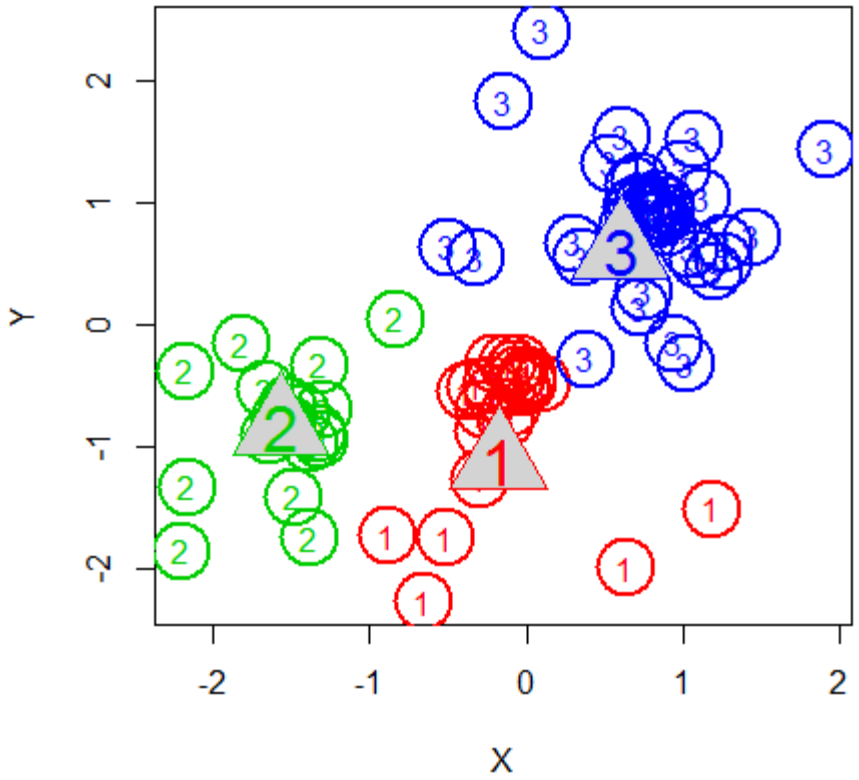
K-Means Clustering (6)



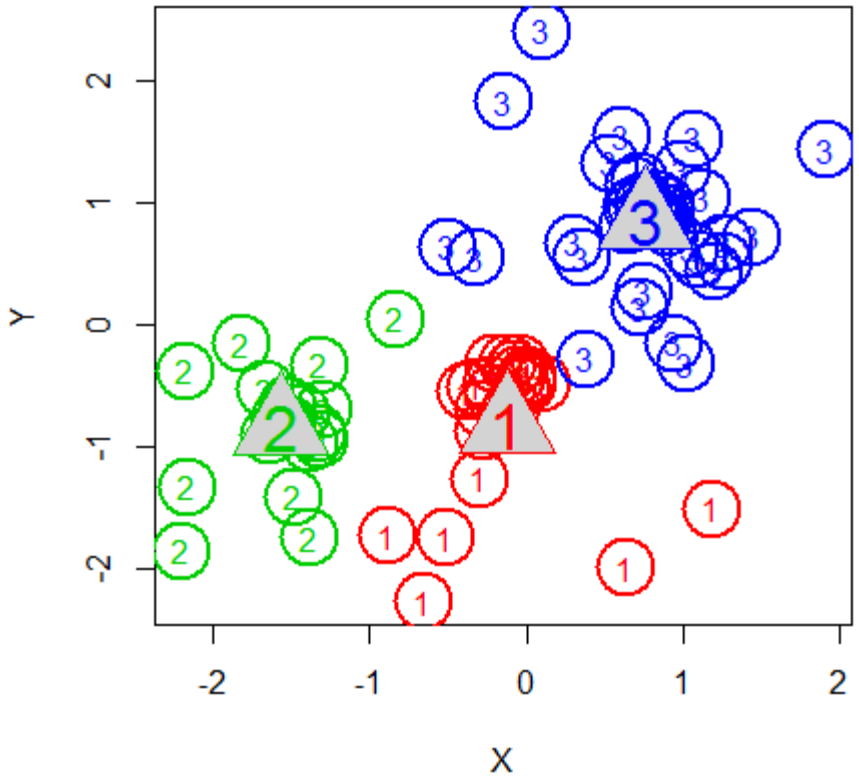
K-Means Clustering (7)



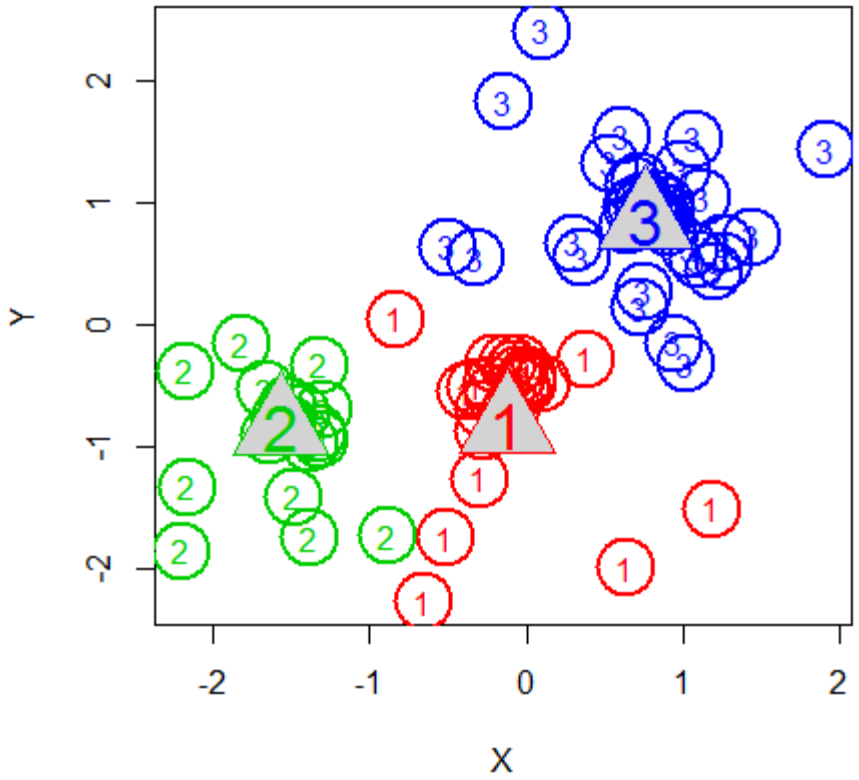
K-Means Clustering (8)



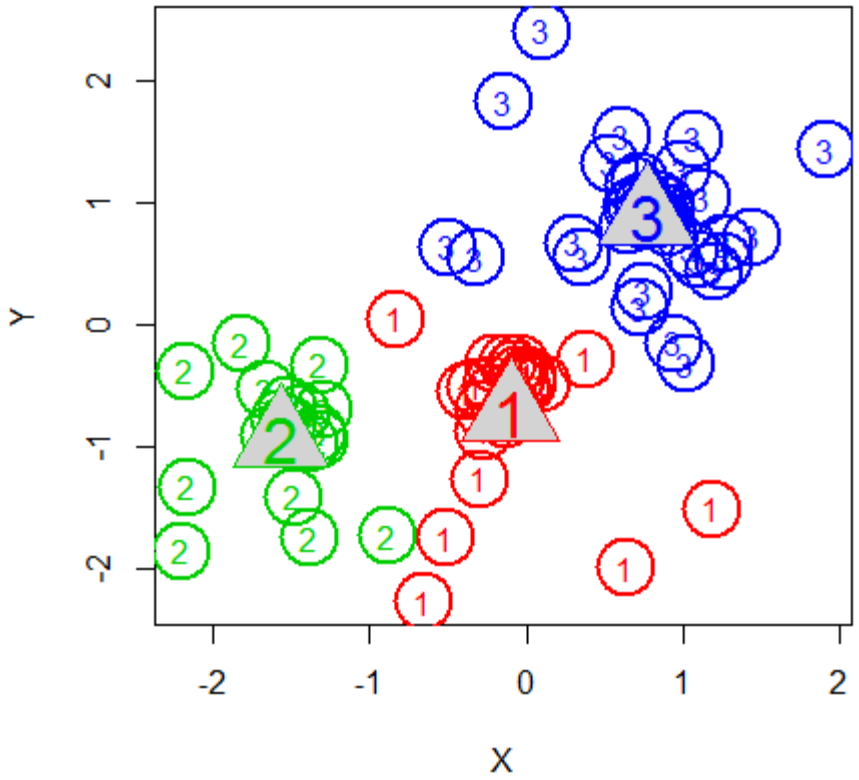
K-Means Clustering (9)



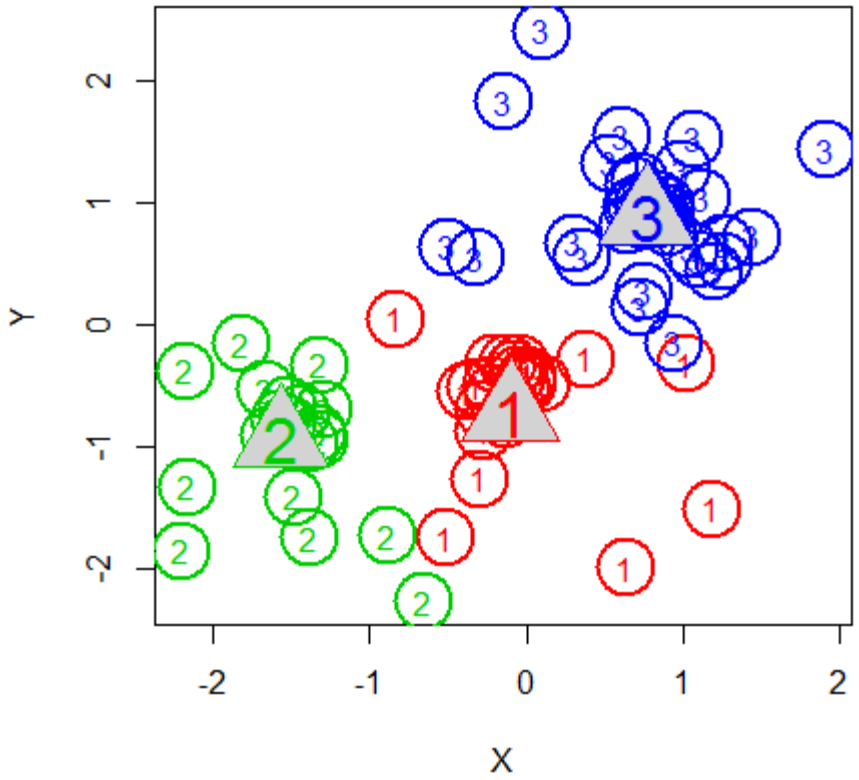
K-Means Clustering (10)



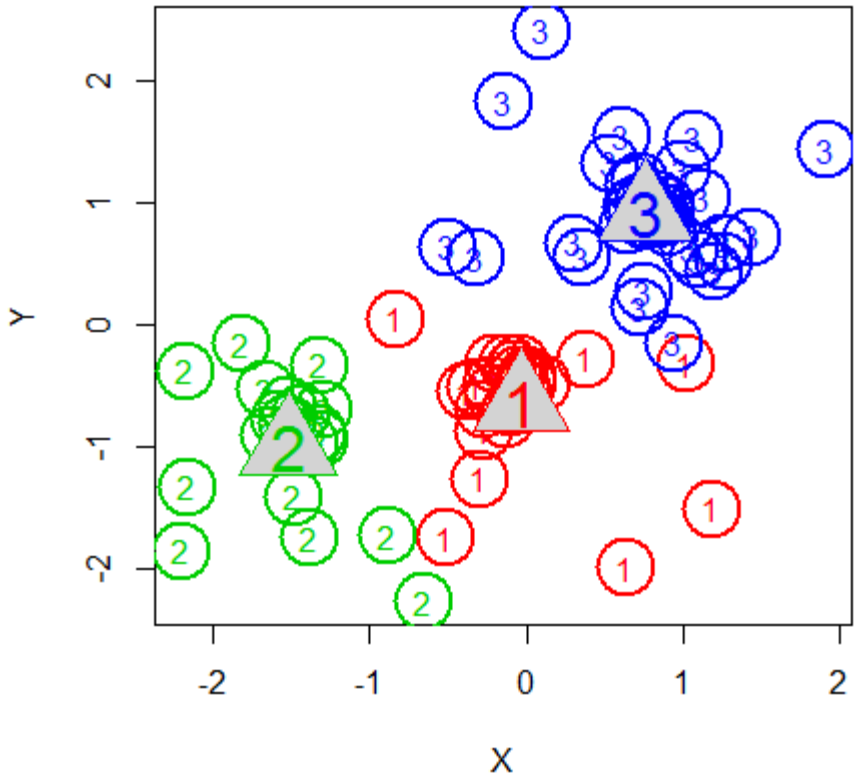
K-Means Clustering (11)



K-Means Clustering (12)



K-Means Clustering (13)

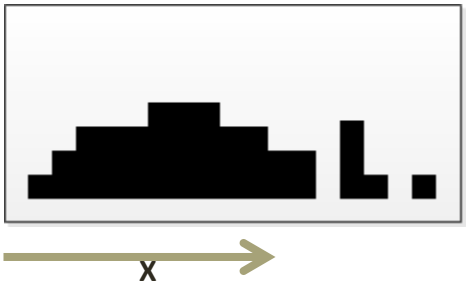


K-means Demo

- KMeansDemo

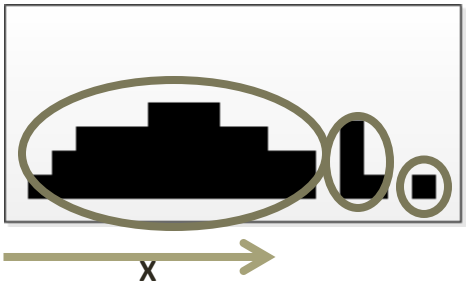
Dimensions in Clustering

Clustering: Dimensions (1)



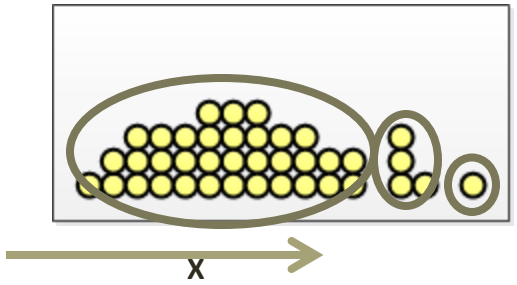
Where are the three clusters?

Clustering: Dimensions (2)



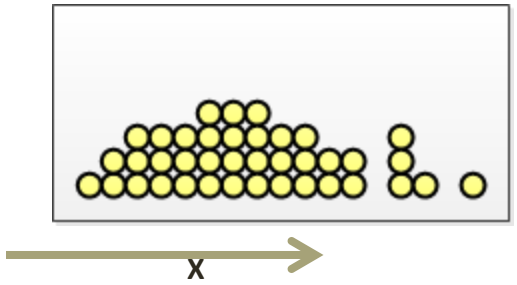
Simple assignment based on a 1D distribution

Clustering: Dimensions (3)



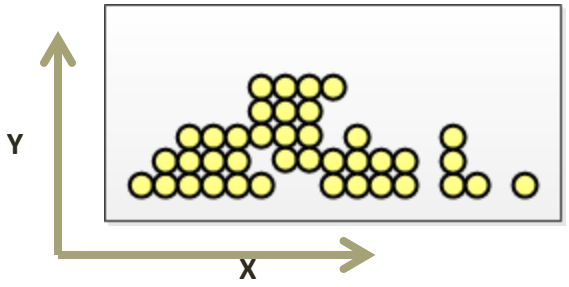
Simple assignment based on a 1D distribution

Clustering: Dimensions (4)



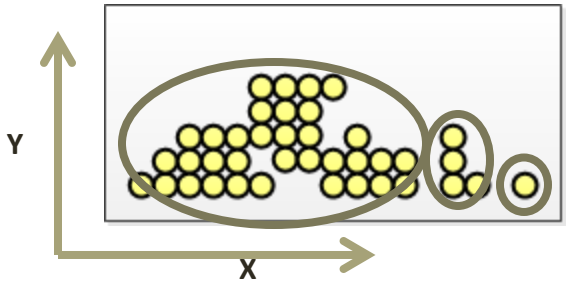
What if this was not a 1D distribution?

Clustering: Dimensions (5)



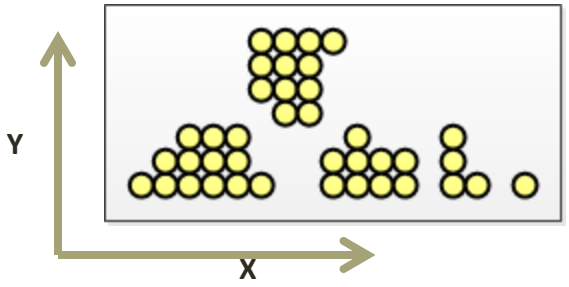
The distribution is in 2D. Some points differ in the 2nd D

Clustering: Dimensions (6)



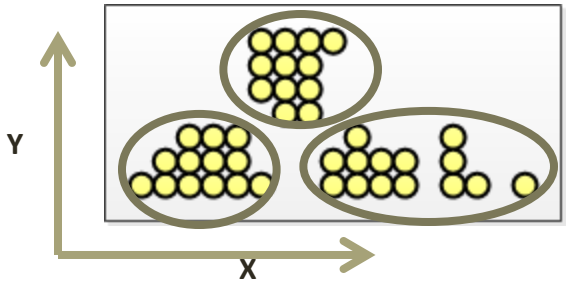
If the difference is minor, we still get the same clusters

Clustering: Dimensions (7)



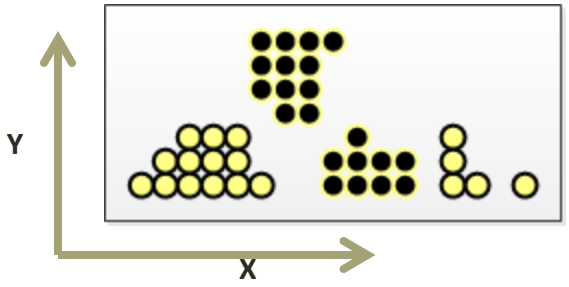
The difference could be significant

Clustering: Dimensions (8)



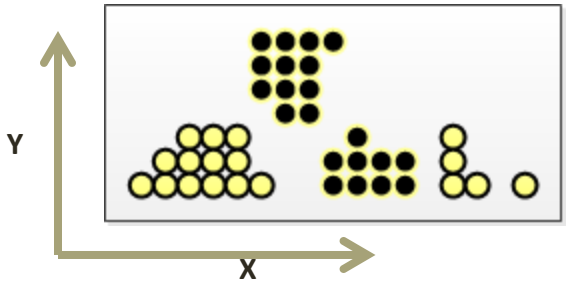
A big difference in the 2nd D can lead to different clusters

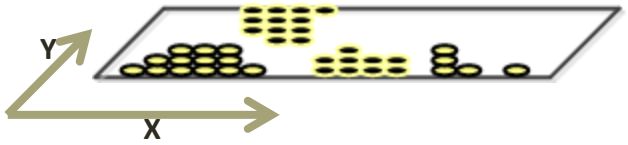
Clustering: Dimensions (9)



We can introduce another D by color coding. This is a Boolean Dimension

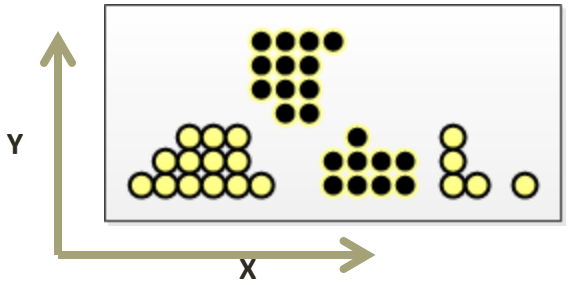
Clustering: Dimensions (10)

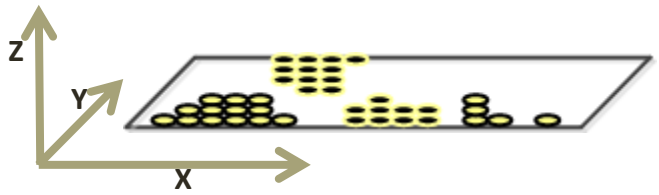




Create a 3rd Dimansion

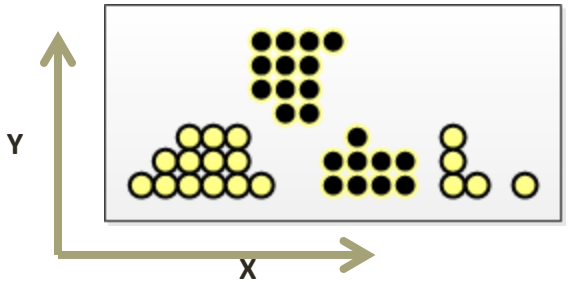
Clustering: Dimensions (11)

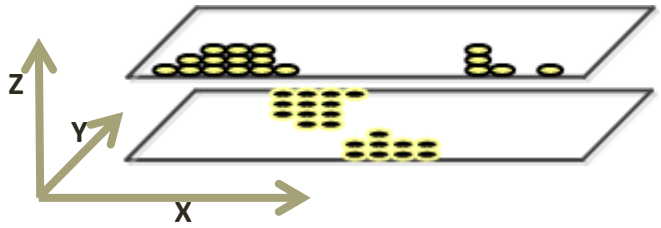




Create a 3rd Dimansion

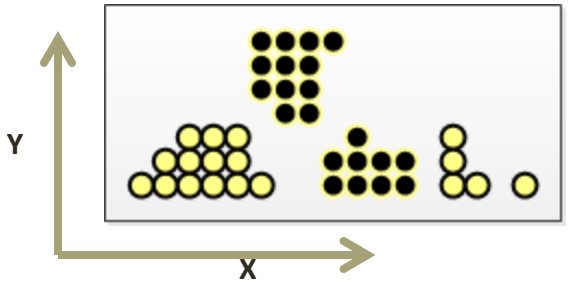
Clustering: Dimensions (12)

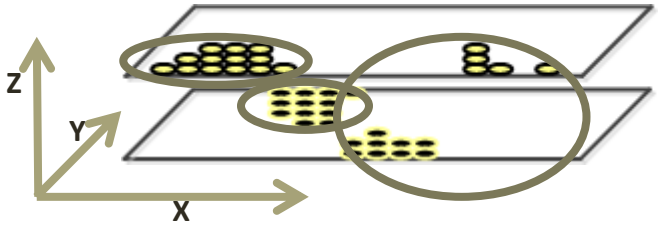


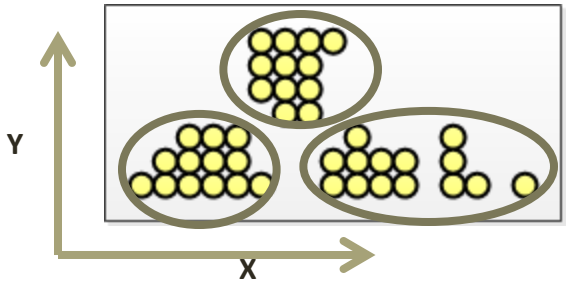


Where are the 3 clusters now?

Clustering: Dimensions (13)

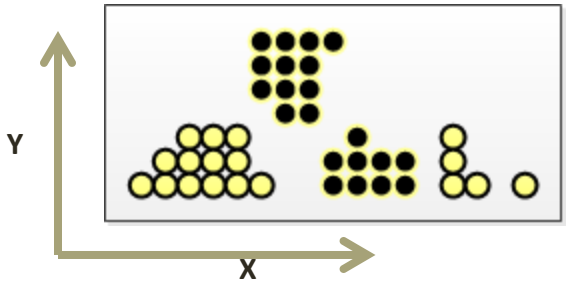


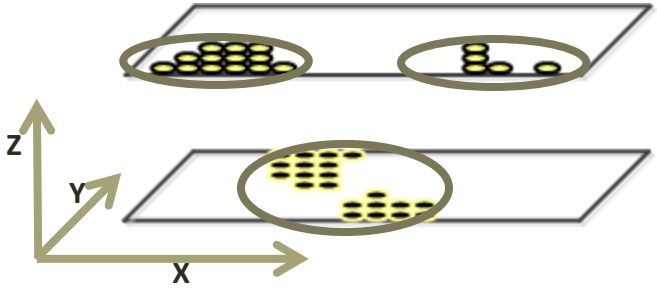


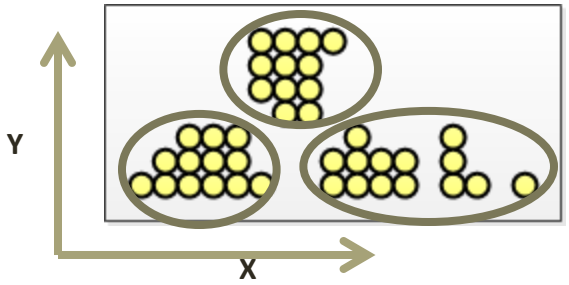


If the 3rd is small, then the clustering is the same as in 2D

Clustering: Dimensions (14)







If the 3rd is big, then the clustering differs from 2D

Dimensions in Clustering

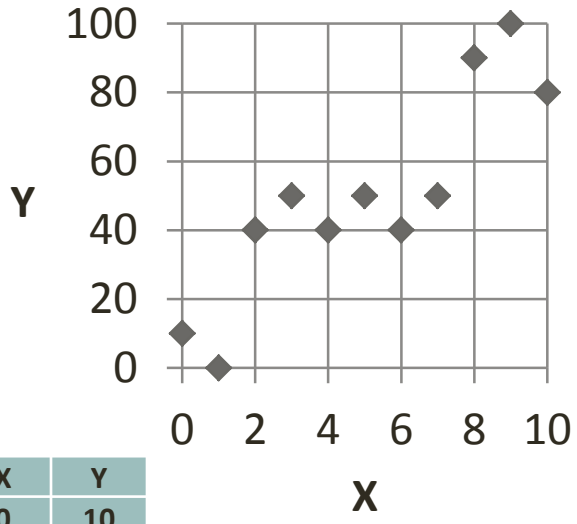
Break?

Normalization in Clustering

Normalization of a linear relationship (1)

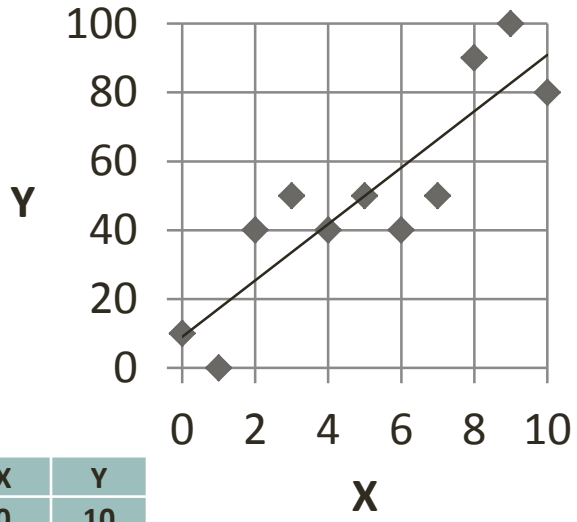
X	Y
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100
10	80

Normalization of a linear relationship (2)



X	Y
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100
10	80

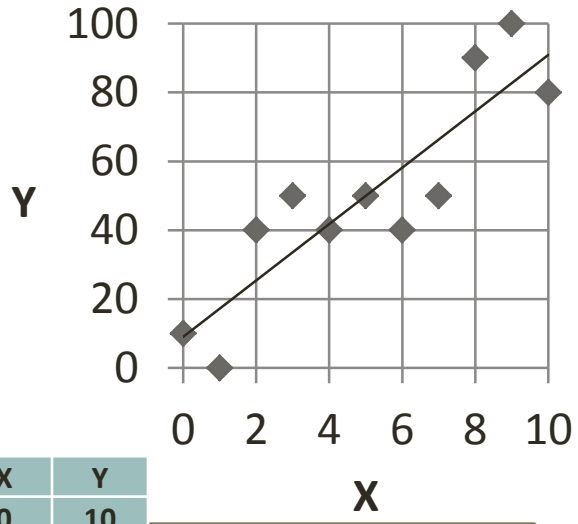
Normalization of a linear relationship (3)



X	Y
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100
10	80

$Y = 10 + 8*X$

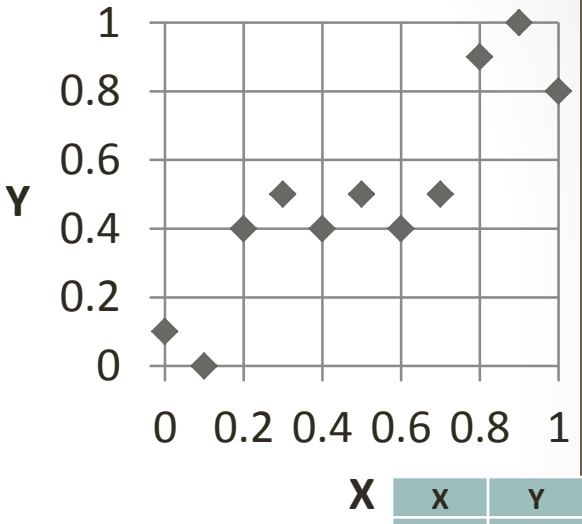
Normalization of a linear relationship (4)



X	Y
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100
10	80

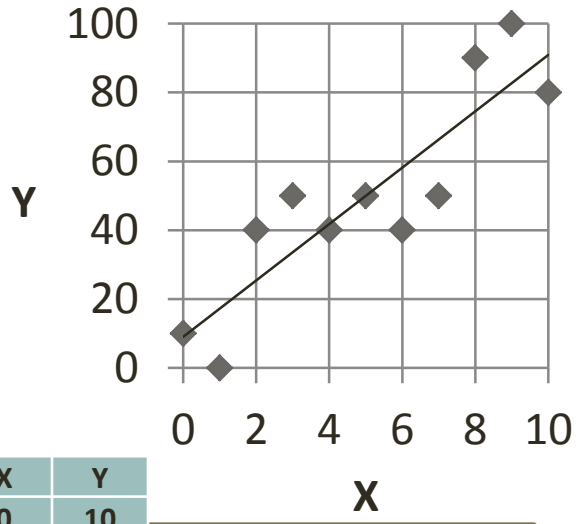
Y = 10 + 8*X

Normalize



X	Y
0	0.1
0.1	0
0.2	0.4
0.3	0.5
0.4	0.4
0.5	0.5
0.6	0.4
0.7	0.5
0.8	0.9
0.9	1
1	0.8

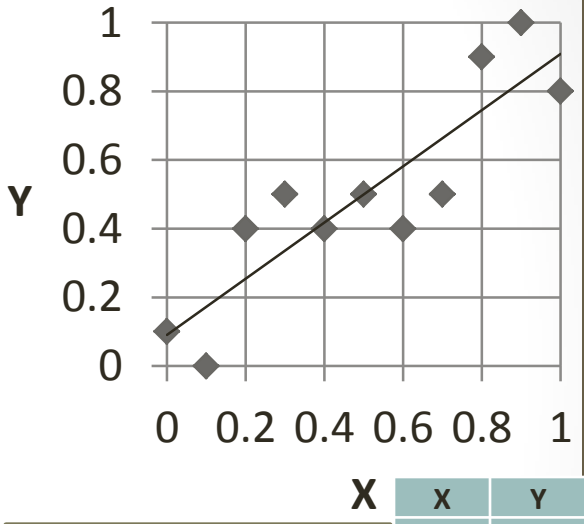
Normalization of a linear relationship (5)



X	Y
0	10
1	0
2	40
3	50
4	40
5	50
6	40
7	50
8	90
9	100
10	80

Y = 10 + 8*X

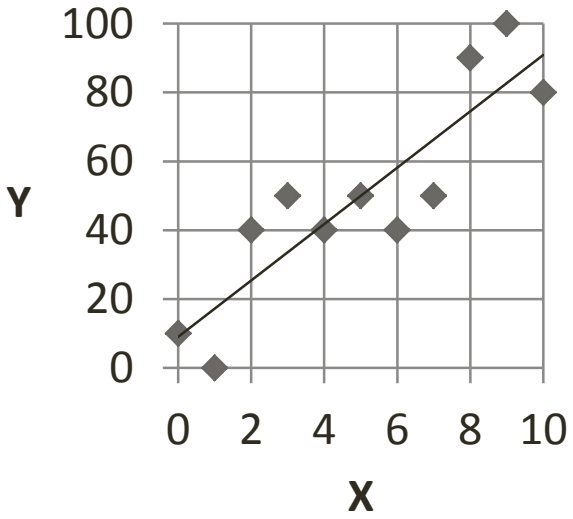
Normalize



Y = 0.1 + 0.8*X

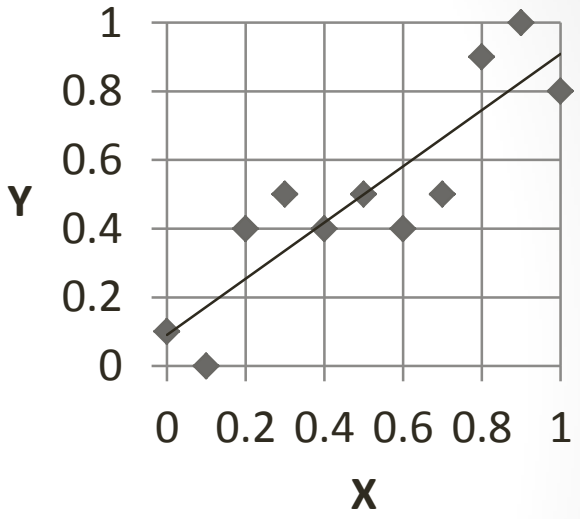
X	Y
0	0.1
0.1	0
0.2	0.4
0.3	0.5
0.4	0.4
0.5	0.5
0.6	0.4
0.7	0.5
0.8	0.9
0.9	1
1	0.8

Normalization of a linear relationship (6)



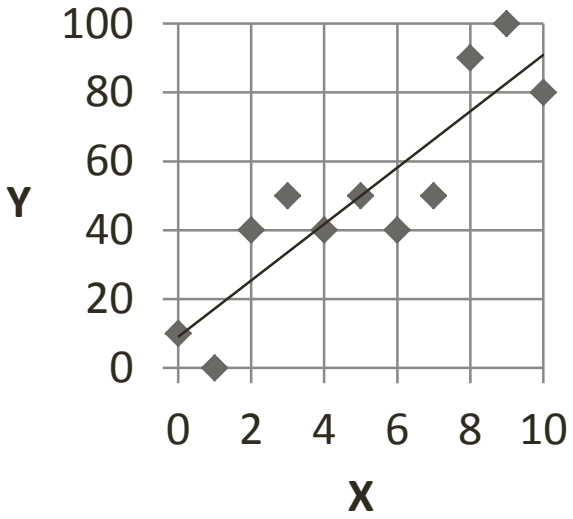
Y = 10 + 8*X

Normalize



Y = 0.1 + 0.8*X

Normalization of a linear relationship (7)



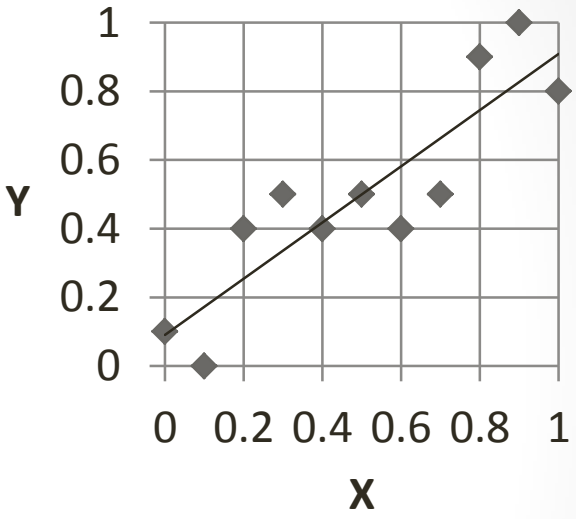
Y = 10 + 8*X

Normalize

Normalize Input
X = 2 -> X' = 0.2

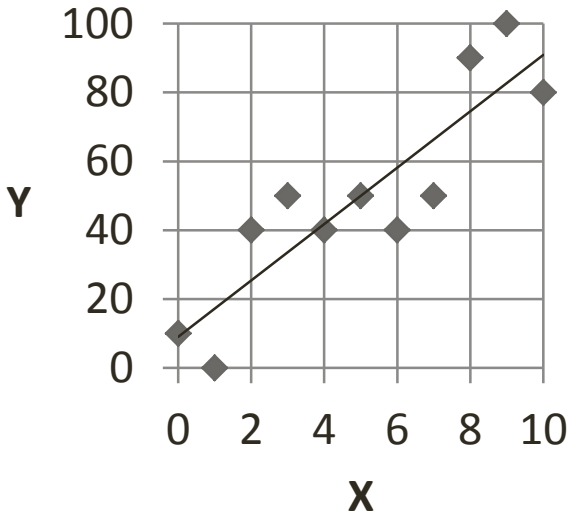
Predict Output
X' = 0.2 -> Y'= 0.26

Denormalize Output
Y'= 0.26 -> Y = 26



Y = 0.1 + 0.8*X

Normalization of a linear relationship (8)



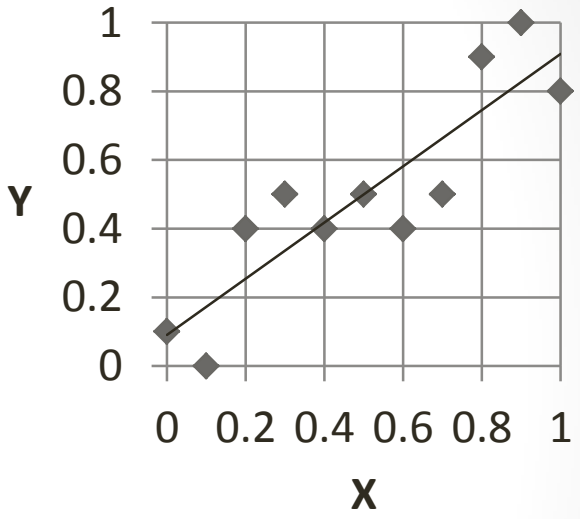
Y = 10 + 8*X

Normalize

Normalize Input
X = 2 -> X' = 0.2

Predict Output
X' = 0.2 -> Y'= 0.26

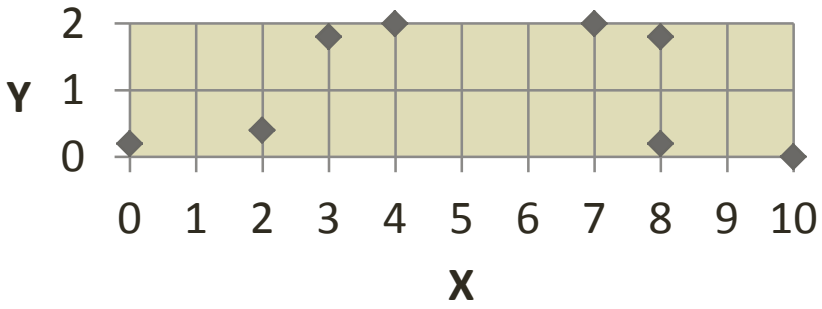
Denormalize Output
Y'= 0.26 -> Y = 26



Y = 0.1 + 0.8*X

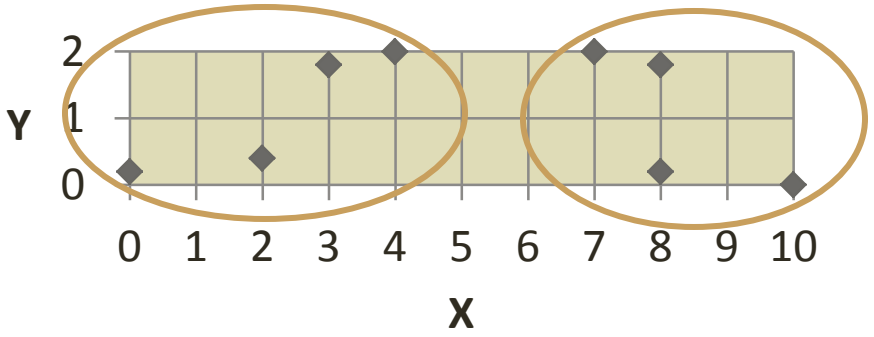
Prediction in Original Space:
X = 2 -> Y = 26

Normalization of a non-linear relationship (1)



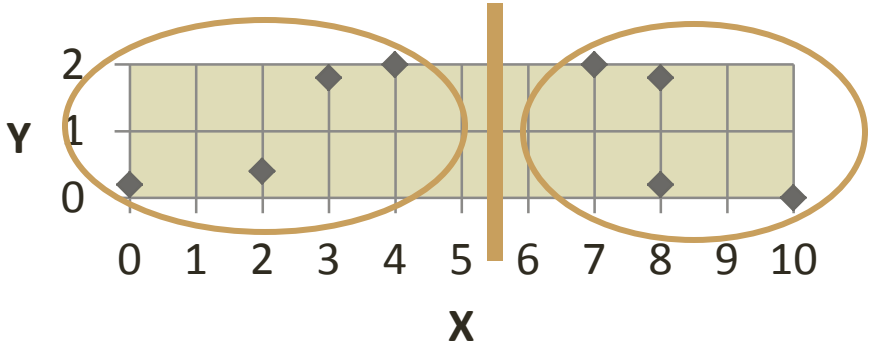
Original data in 2D:
Find 2 clusters

Normalization of a non-linear relationship (2)



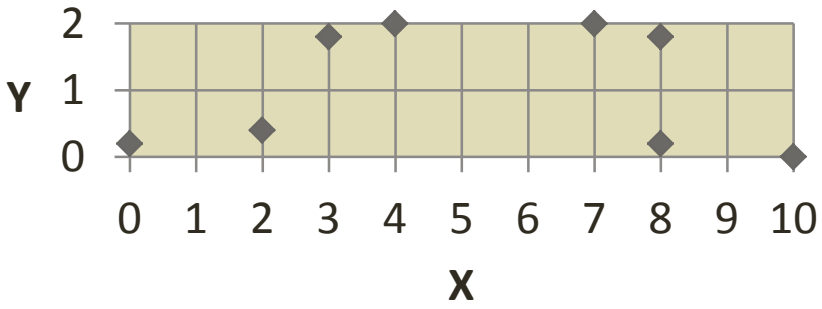
Found 2 Clusters

Normalization of a non-linear relationship (3)



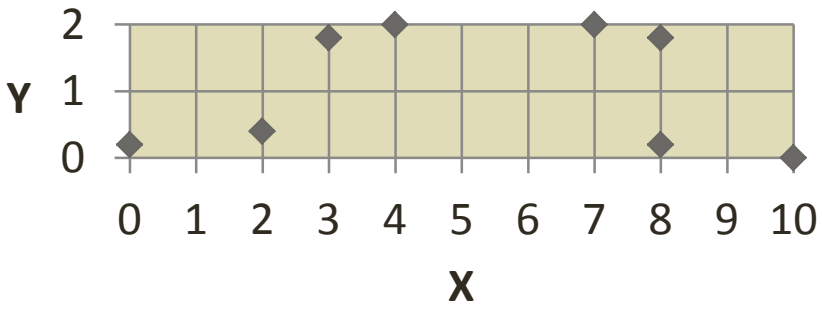
Clusters segment the image

Normalization of a non-linear relationship (4)

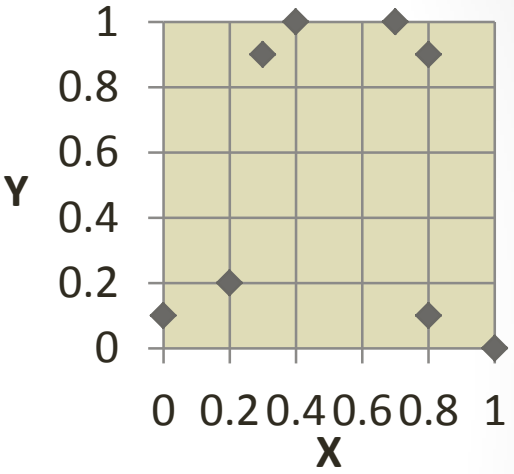


Non-normalized 2D data

Normalization of a non-linear relationship (5)



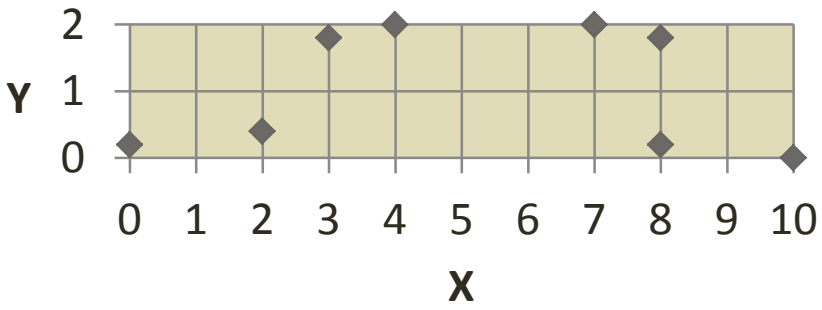
Normalize



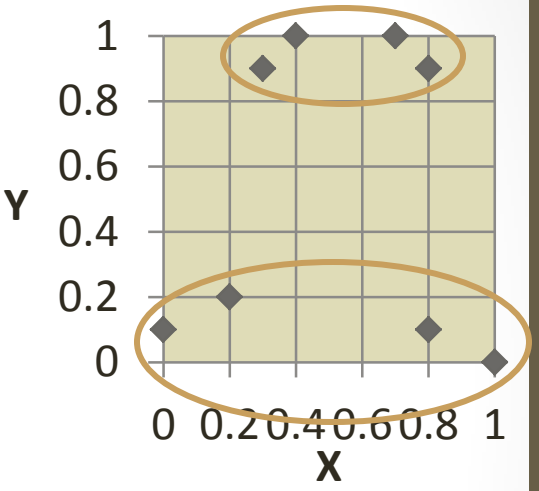
Non-normalized 2D data

Normalize the data:
Search for 2 Clusters

Normalization of a non-linear relationship (6)



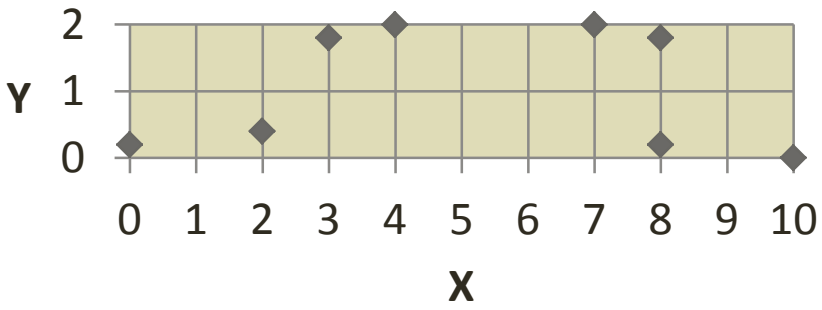
Normalize



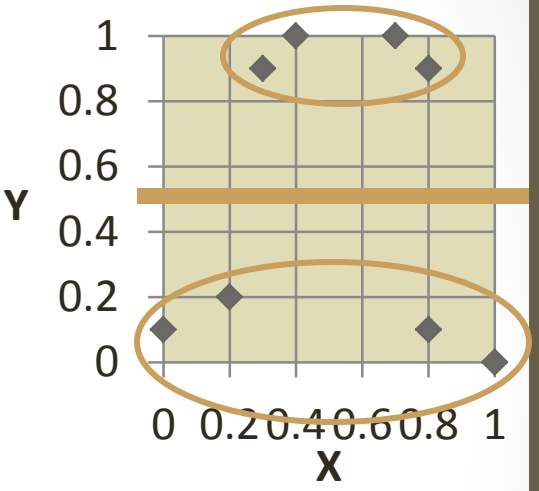
Non-normalized 2D data

Found 2 Clusters in the normalized data

Normalization of a non-linear relationship (6)



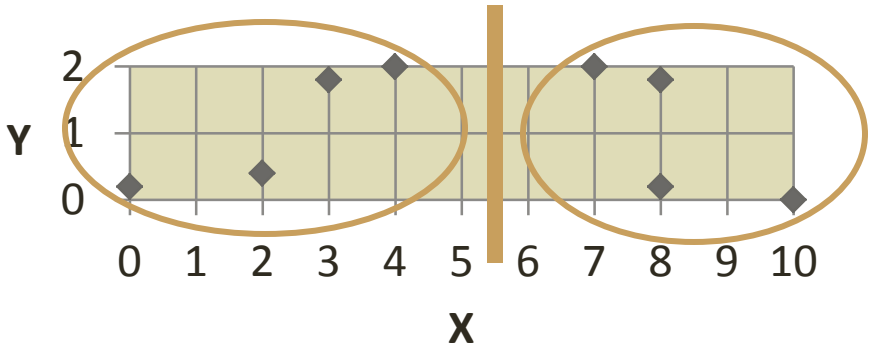
Normalize



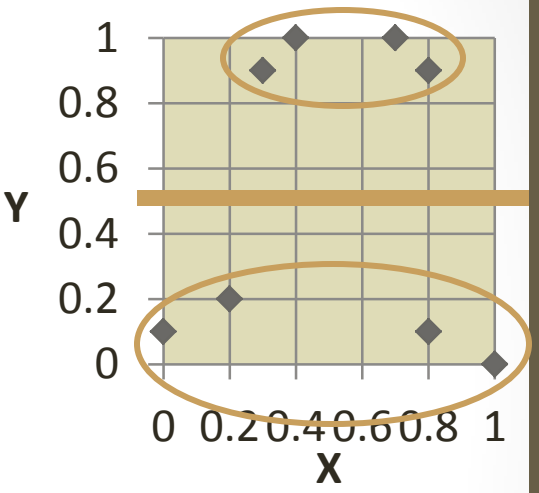
Non-normalized 2D data

Clusters Segment the Image

Normalization of a non-linear relationship (7)



Normalize



Clustering before normalization

Clustering after normalization

Normalization of Linear and Non-Linear Outcomes

- Non-linear (Normalization can change outcome):
- K-Means
- Neural Net

- Linear (Normalization should not change outcome):
- Logistic Regression
- Linear Regression
- Mixture of Gaussians