Data Clustering

Assoc. Prof. Kitsana Waiyamai, Ph. D.

Dept. Of Computer Engineering

Faculty of Engineering, Kasetsart University

Bangkok, Thailand

Unsupervised Learning

- Association Rules
  - What happens together?
- Data Clustering
  - What belongs together?

Data Clustering

Data classification organizes data in given classes based on attribute values

=> Supervised-learning !

Data Clustering divides data into meaningful or useful clusters

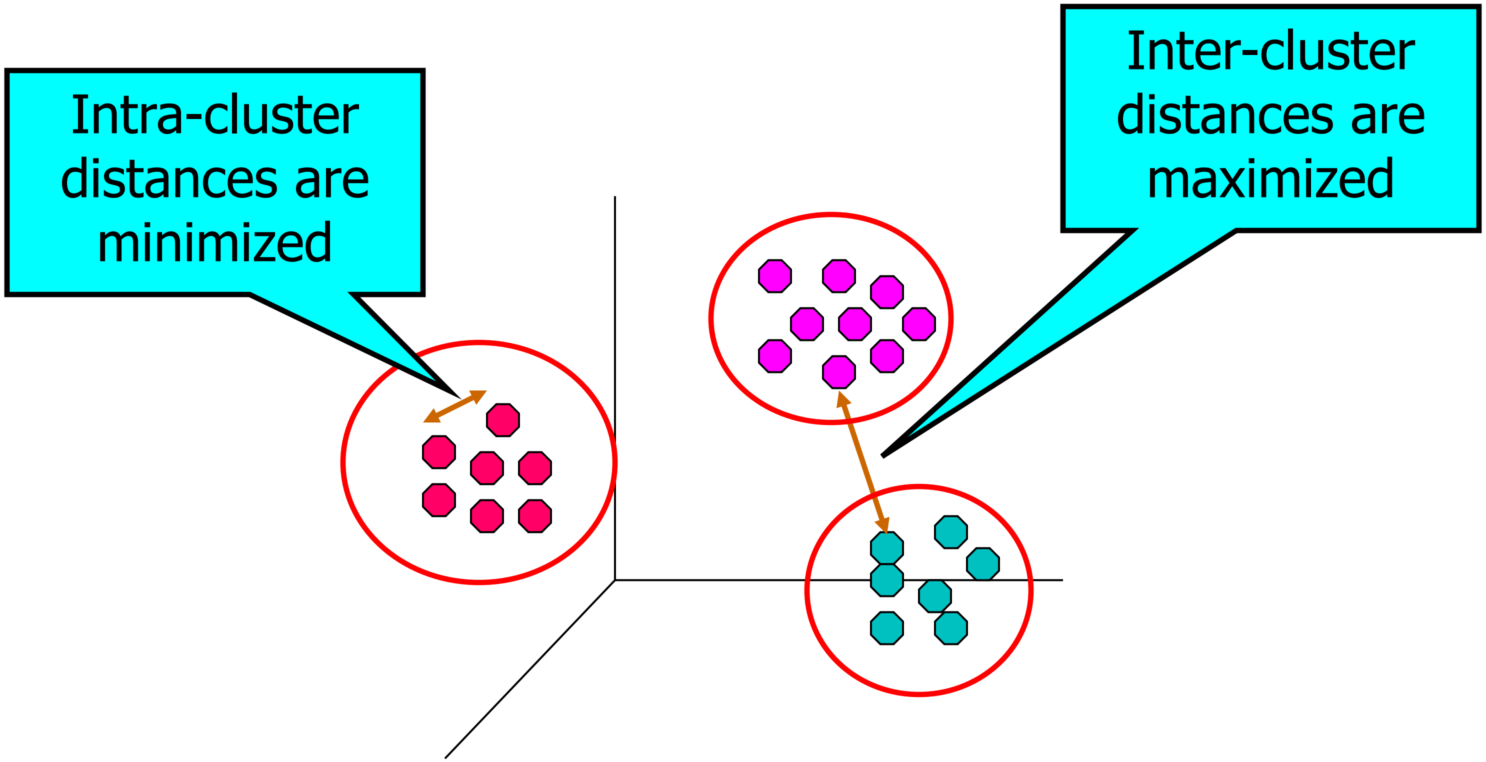
=> Un-Supervised-learning !

Example: 22 Public Utilities

Company	Fixed_charge	RoR	Cost	Load	Δ Demand	Sales	Nuclear	Fuel_Cost
Arizona	1.06	9.2	151	54.4	1.6	9077	0	0.628
Boston	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
Central	1.43	15.4	113	53	3.4	9212	0	1.058
Commonwealth	1.02	11.2	168	56	0.3	6423	34.3	0.7
Con Ed NY	1.49	8.8	192	51.2	1	3300	15.6	2.044
Florida	1.32	13.5	111	60	-2.2	11127	22.5	1.241
Hawaiian	1.22	12.2	175	67.6	2.2	7642	0	1.652
Idaho	1.1	9.2	245	57	3.3	13082	0	0.309
Kentucky	1.34	13	168	60.4	7.2	8406	0	0.862
Madison	1.12	12.4	197	53	2.7	6455	39.2	0.623
Nevada	0.75	7.5	173	51.5	6.5	17441	0	0.768
New England	1.13	10.9	178	62	3.7	6154	0	1.897
Northern	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
Oklahoma	1.09	12	96	49.8	1.4	9673	0	0.588
Pacific	0.96	7.6	164	62.2	-0.1	6468	0.9	1.4
Puget	1.16	9.9	252	56	9.2	15991	0	0.62
San Diego	0.76	6.4	136	61.9	9	5714	8.3	1.92
Southern	1.05	12.6	150	56.7	2.7	10140	0	1.108
Texas	1.16	11.7	104	54	-2.1	13507	0	0.636
Wisconsin	1.2	11.8	148	59.9	3.5	7287	41.1	0.702
United	1.04	8.6	204	61	3.5	6650	0	2.116
Virginia	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

What is Data Clustering?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Data Clustering

- 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
- Clustering = set of clusters
- Unsupervised learning: no predefined classes
- Typical applications
  - As a pre-processing step for other algorithms
  - As a stand-alone tool to get insight into data distribution

Clustering Characteristic

- Provides a way to learn about the structure of complex data
- Simply finds structure that exists in the data without regard to any target variable => un-supervised learning
- Finding clusters is not often an end in itself
- Once clusters have been detected, other methods must be applied in order to figure out Meaning of clusters

Example: Public Utilities

Goal: find clusters of similar utilities

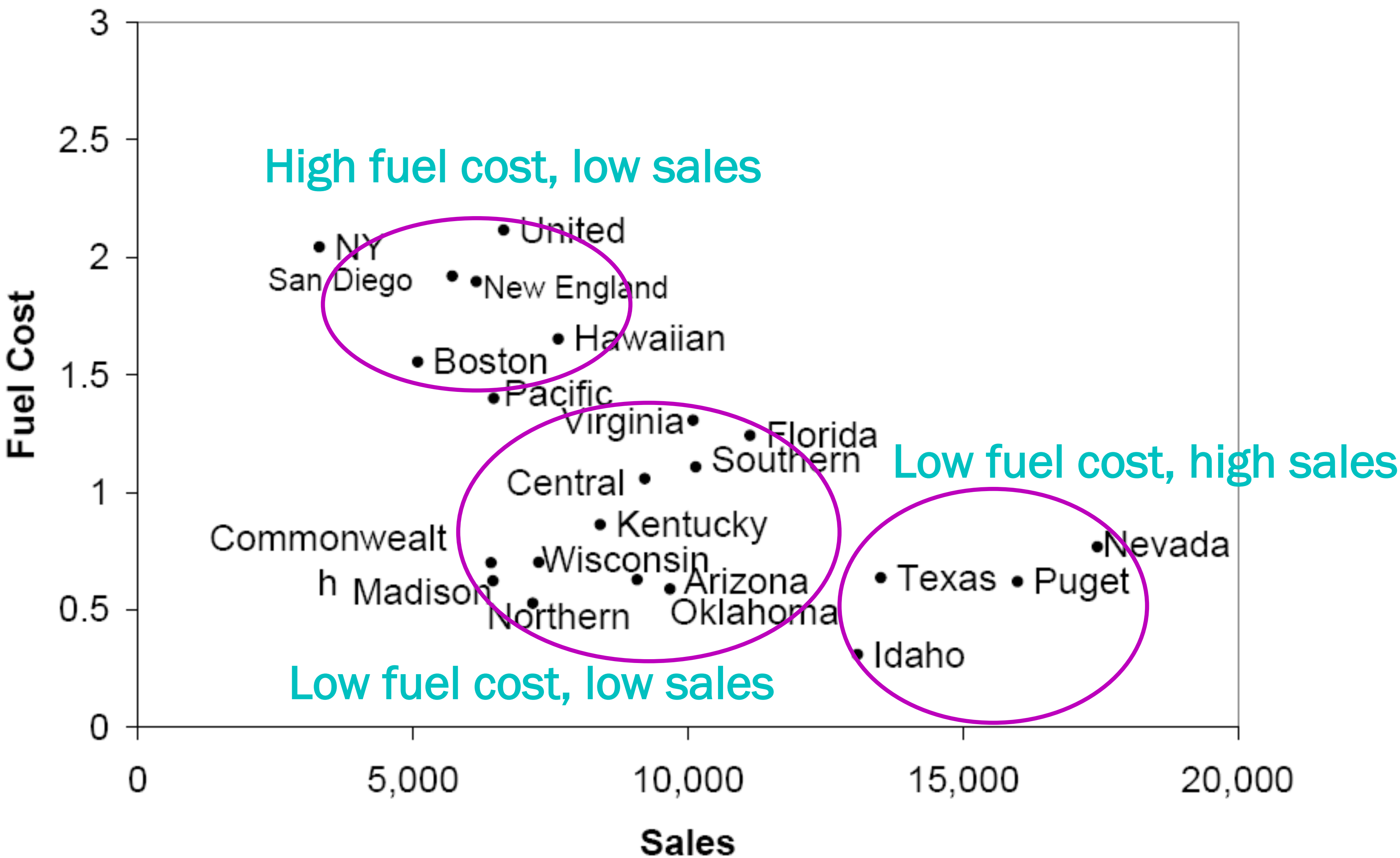
Data: 22 firms, 8 variables

- Fixed-charge covering ratio
- Rate of return on capital
- Cost per kilowatt capacity
- Annual load factor
- Growth in peak demand
- Sales
- % nuclear
- Fuel costs per kwh

Example: 22 Public Utilities

Company	Fixed_charge	RoR	Cost	Load	Δ Demand	Sales	Nuclear	Fuel_Cost
Arizona	1.06	9.2	151	54.4	1.6	9077	0	0.628
Boston	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
Central	1.43	15.4	113	53	3.4	9212	0	1.058
Commonwealth	1.02	11.2	168	56	0.3	6423	34.3	0.7
Con Ed NY	1.49	8.8	192	51.2	1	3300	15.6	2.044
Florida	1.32	13.5	111	60	-2.2	11127	22.5	1.241
Hawaiian	1.22	12.2	175	67.6	2.2	7642	0	1.652
Idaho	1.1	9.2	245	57	3.3	13082	0	0.309
Kentucky	1.34	13	168	60.4	7.2	8406	0	0.862
Madison	1.12	12.4	197	53	2.7	6455	39.2	0.623
Nevada	0.75	7.5	173	51.5	6.5	17441	0	0.768
New England	1.13	10.9	178	62	3.7	6154	0	1.897
Northern	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
Oklahoma	1.09	12	96	49.8	1.4	9673	0	0.588
Pacific	0.96	7.6	164	62.2	-0.1	6468	0.9	1.4
Puget	1.16	9.9	252	56	9.2	15991	0	0.62
San Diego	0.76	6.4	136	61.9	9	5714	8.3	1.92
Southern	1.05	12.6	150	56.7	2.7	10140	0	1.108
Texas	1.16	11.7	104	54	-2.1	13507	0	0.636
Wisconsin	1.2	11.8	148	59.9	3.5	7287	41.1	0.702
United	1.04	8.6	204	61	3.5	6650	0	2.116
Virginia	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

Sales & Fuel Cost: 3 rough clusters can be seen



Types of Clusterings

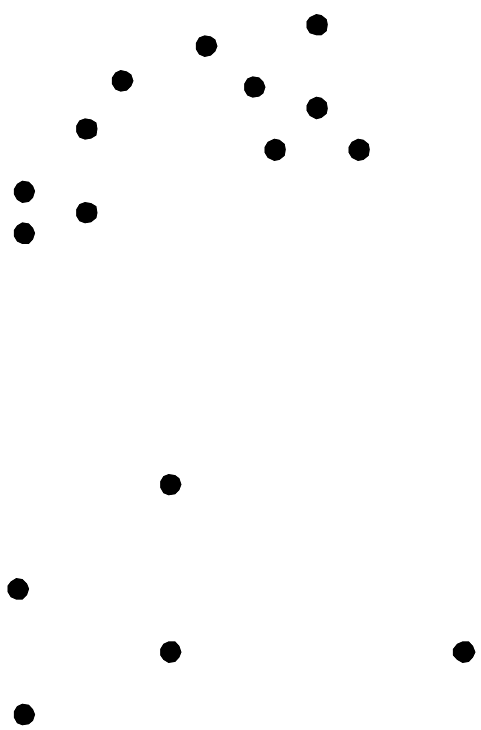
- A clustering is a set of clusters

- Important distinction between hierarchical and partitional sets of clusters

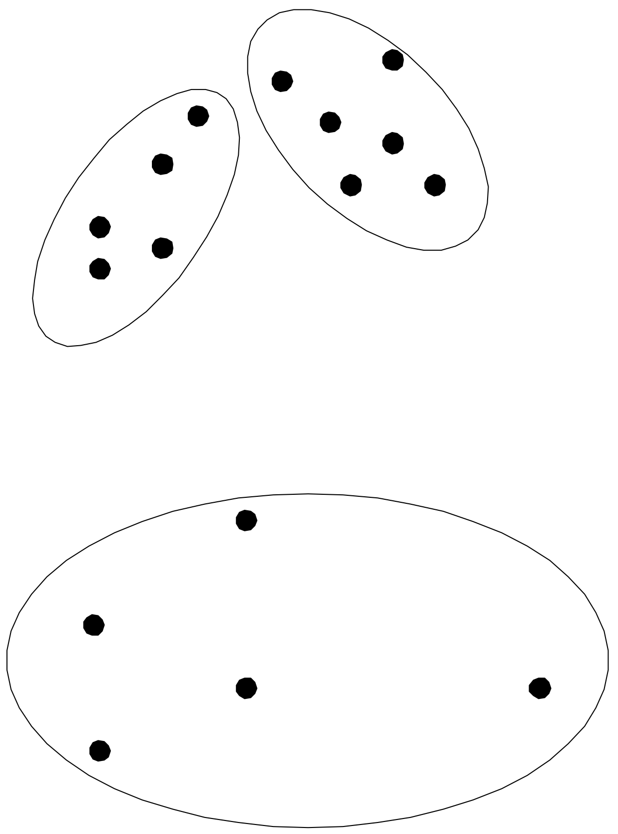
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

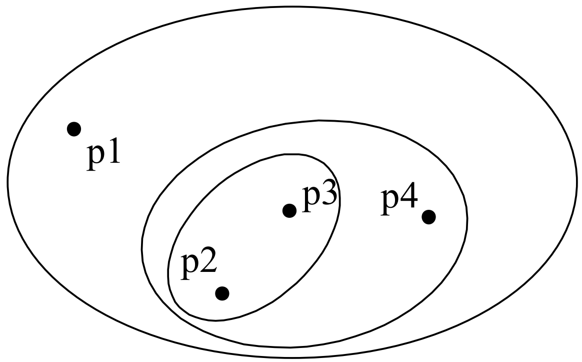


Original Points

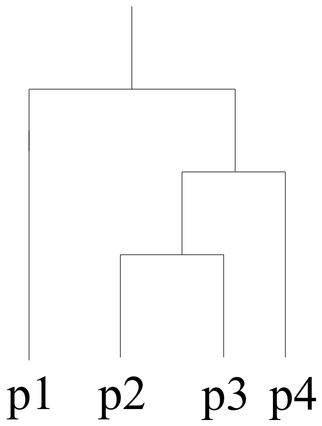


A Partitional Clustering

Hierarchical Clustering



Hierarchical Clustering



Dendrogram

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

Four Clusters

Partitioning Algorithms: Basic Concept

Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion

- Global optimal: exhaustively enumerate all partitions
- Heuristic methods: *k-means* and *k-medoids* algorithms
- *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
- *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

- K-Means: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters, s.t., min sum of squared distance

$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - m_i)^2$$

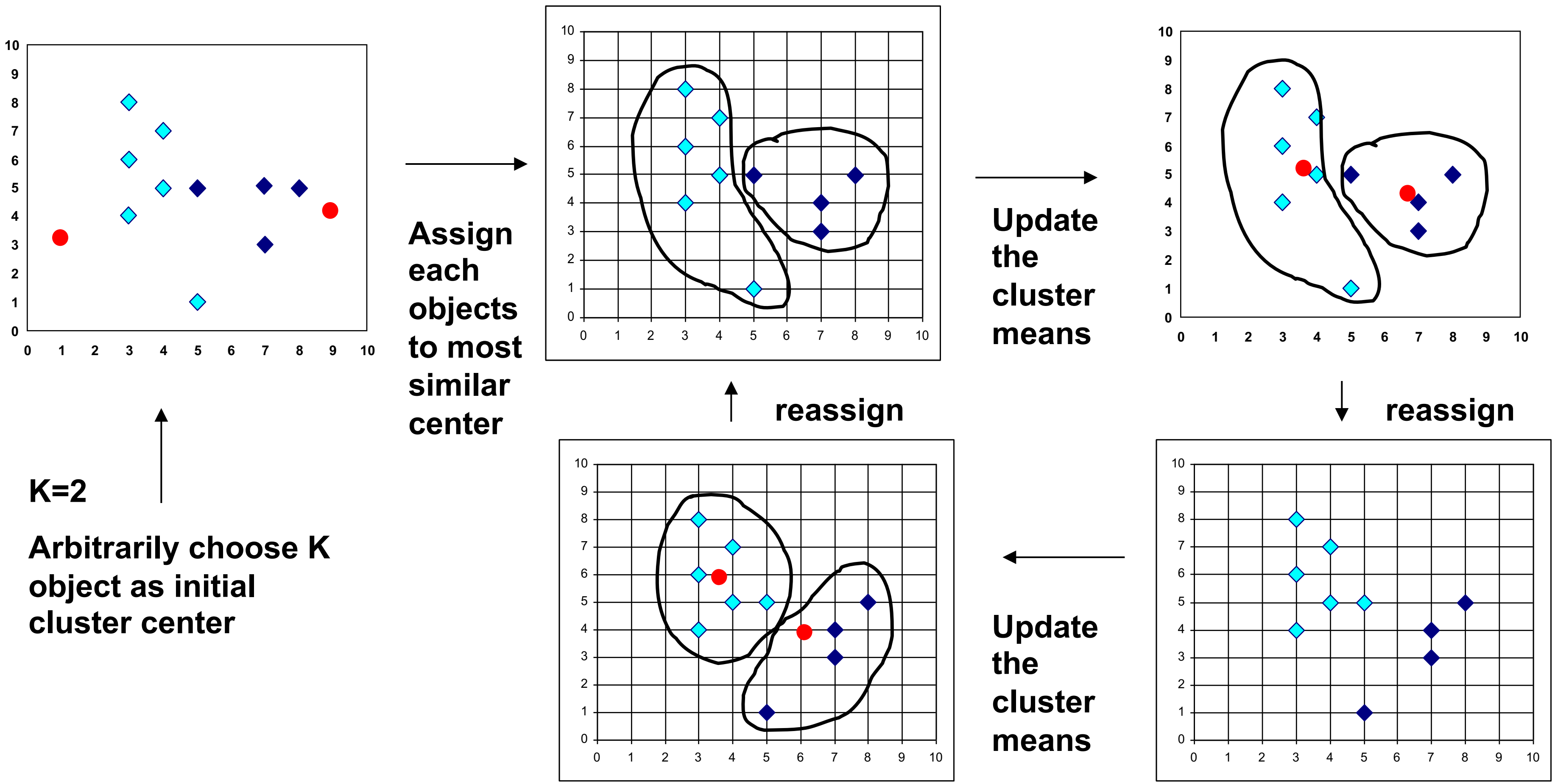
Simple Clustering: K-means

- Uses a set of clusters(K), then iteratively recalculates the center of each cluster.
- Works with numeric data only.

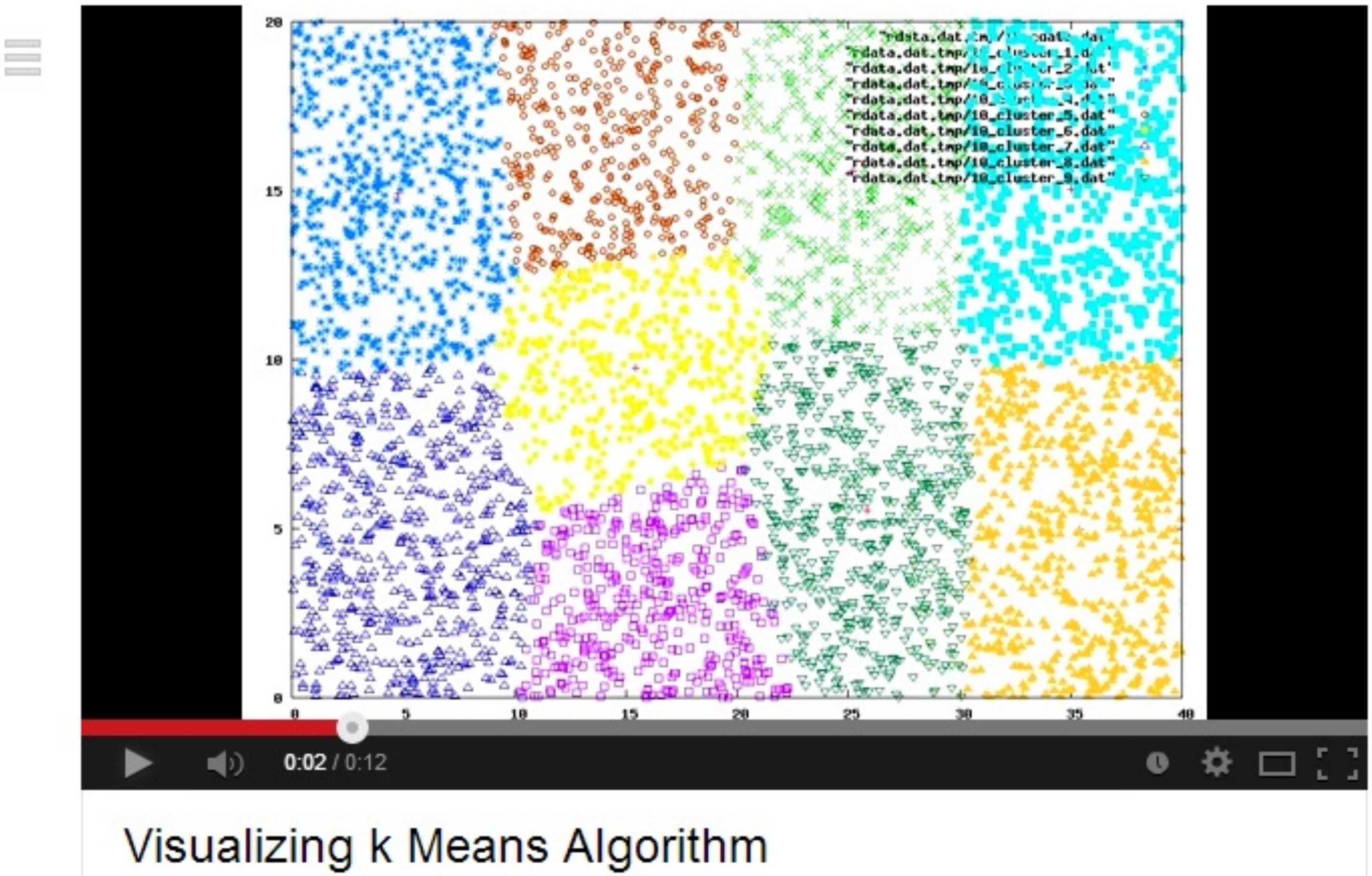
1) Pick a number (K) of cluster centers (at random)
2) Assign every item to its nearest cluster center (example, using Euclidean distance)
3) Move each cluster center to the mean of its assigned items
4) Repeat steps 2,3 until convergence (or change in cluster assignments less than a threshold)

The *K-Means* Clustering Method

- Example



K-mean clustering visualization



http://www.youtube.com/watch?v=gSt4_kcZPxE

Simple Clustering: K-means

Not changed

- Data points

Changed

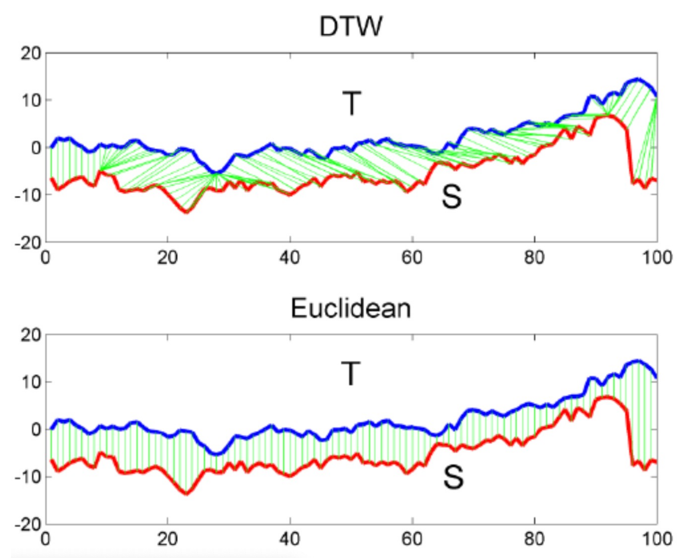
- Boundary
- Centroid
- Membership

Distance Between Two Records

Euclidean Distance is most popular:

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

- **Problem:** Raw distance measures are highly influenced by scale of measurements
- **Solution:** normalize (standardize) the data first
  - Subtract mean, divide by std. deviation
  - Also called **z-scores**
- Other Distances
  - Correlation-based similarity
  - [Continuous + categorical] Gower's similarity
  - Text-Analytics => Angle [lot of words]
  - Time series => DTW
  - Etc.



Pre-processing and Post-processing

- Pre-processing
  - Normalize the data
  - Dimension reduction
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters,
  - Merge clusters that are ‘close’
  - Etc.

K-means clustering summary

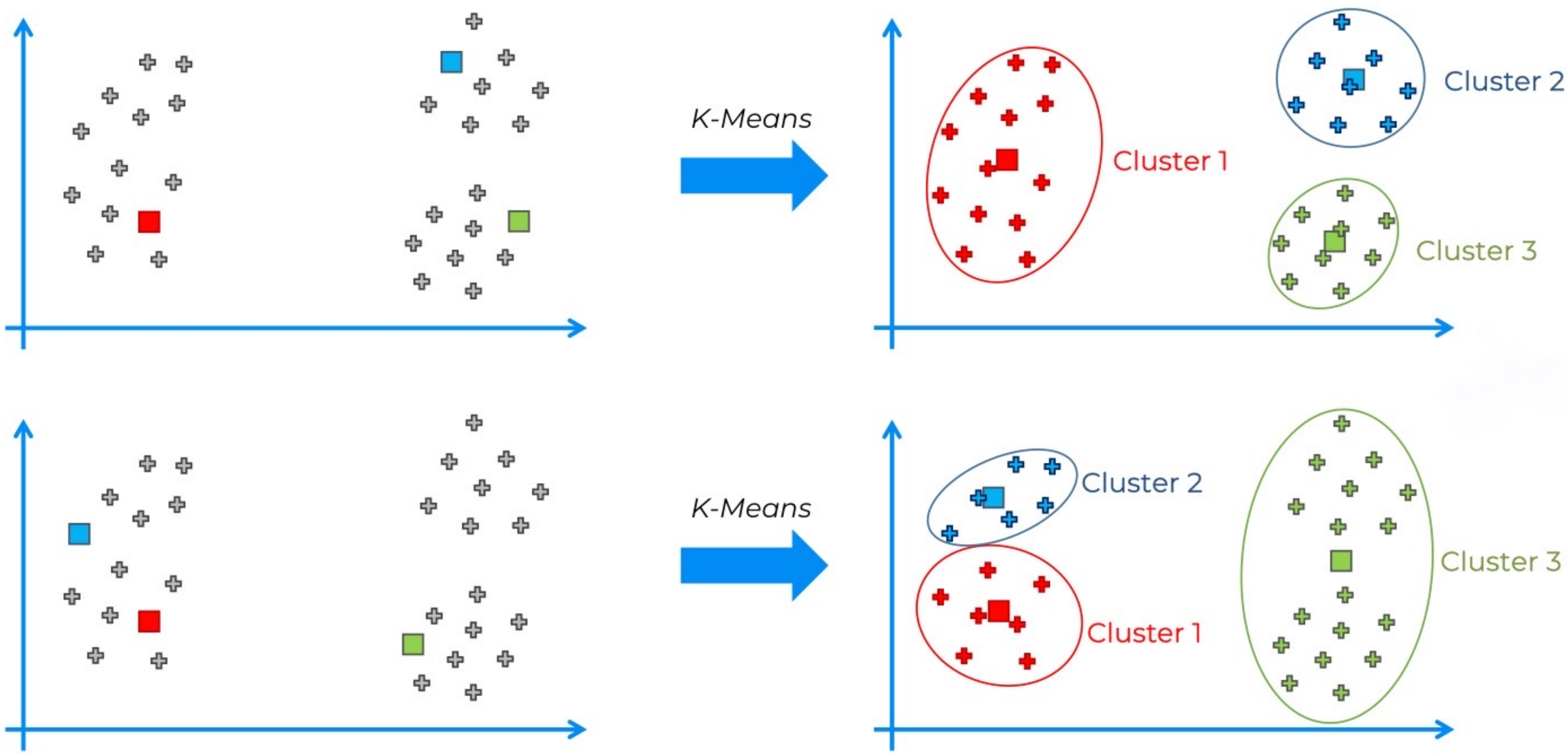
Advantages

- Items automatically assigned to clusters
- Simple, understandable
- *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, $t \ll n$

Disadvantages

- Must pick number of clusters before hand
- Result can vary significantly depending on initial choice of seeds (number and position)
- All items forced into a cluster
- Too sensitive to outliers
- Not suitable to discover <> clusters (size, density) and with *non-globular shapes*

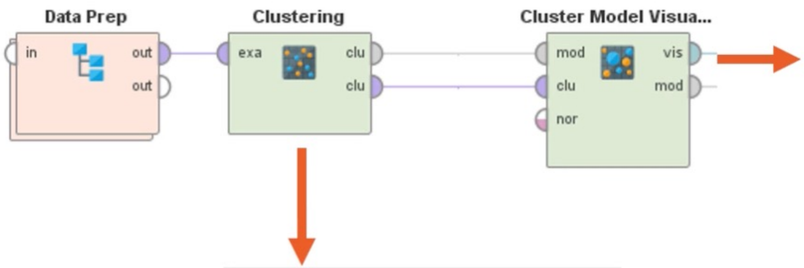
Some Problem(s) of K-Means



Result can vary significantly depending on initial choice of seeds

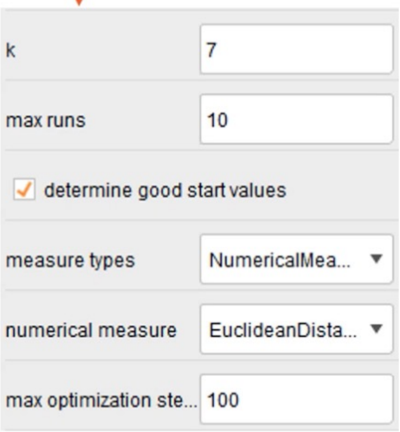
Use **K-Mean++** [David Arthur and Sergei Vassilvitskii 2007] to determine good start seeds.

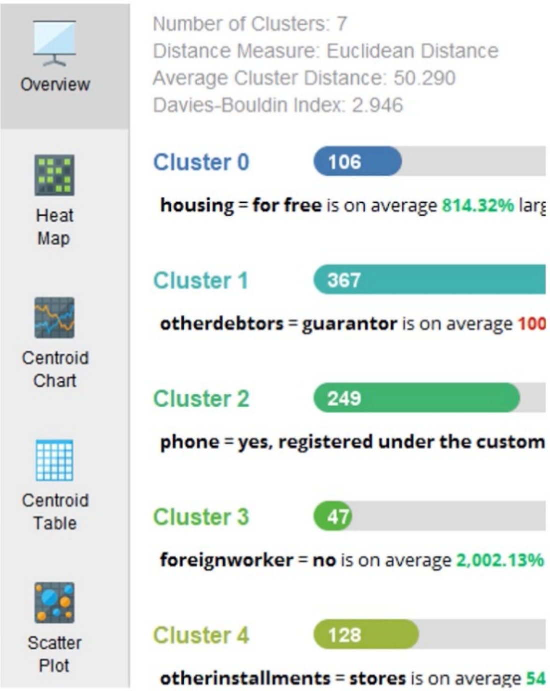
RapidMiner K-Mean Clustering



k sets number of clusters

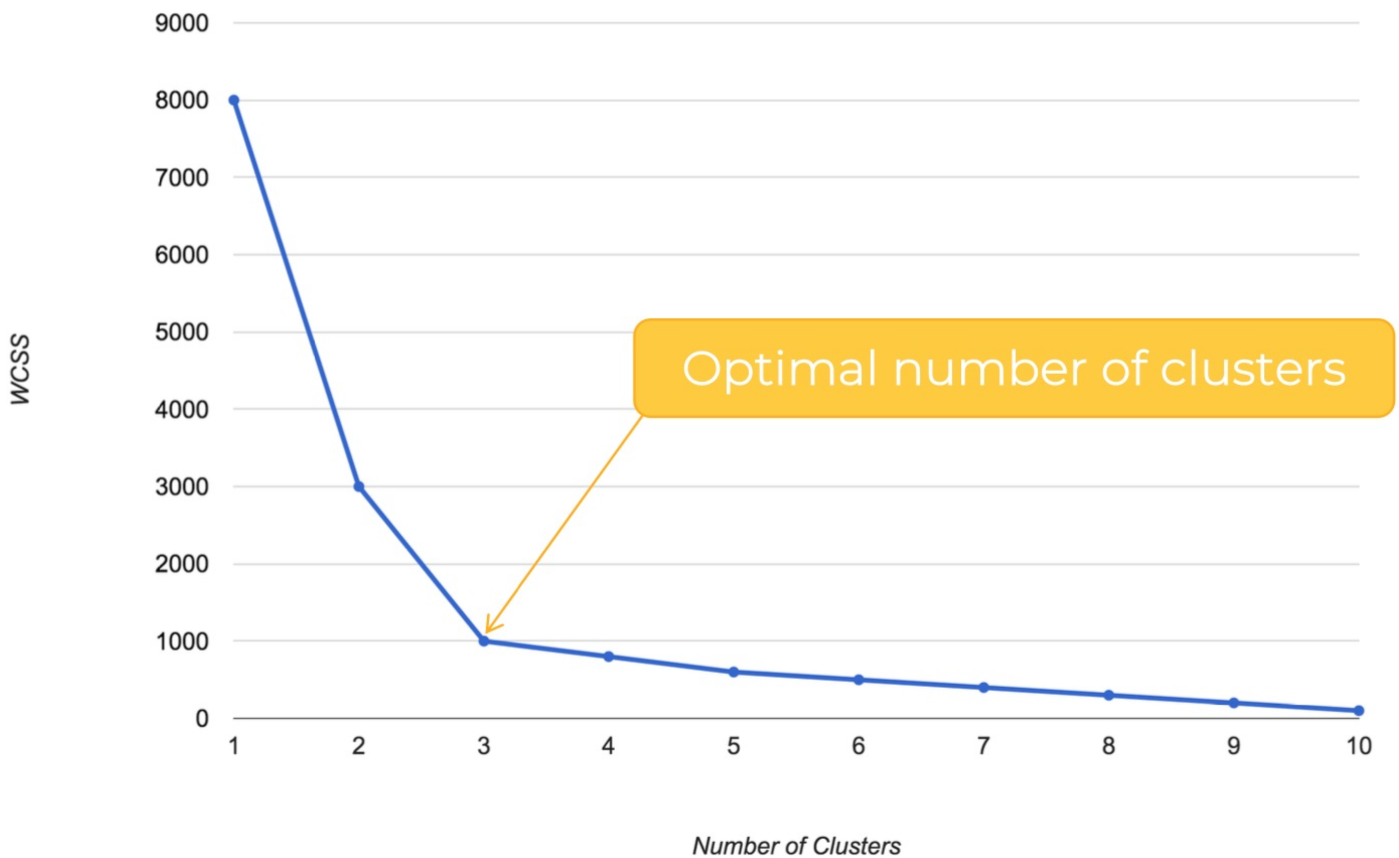
Measure types sets how distances are measured



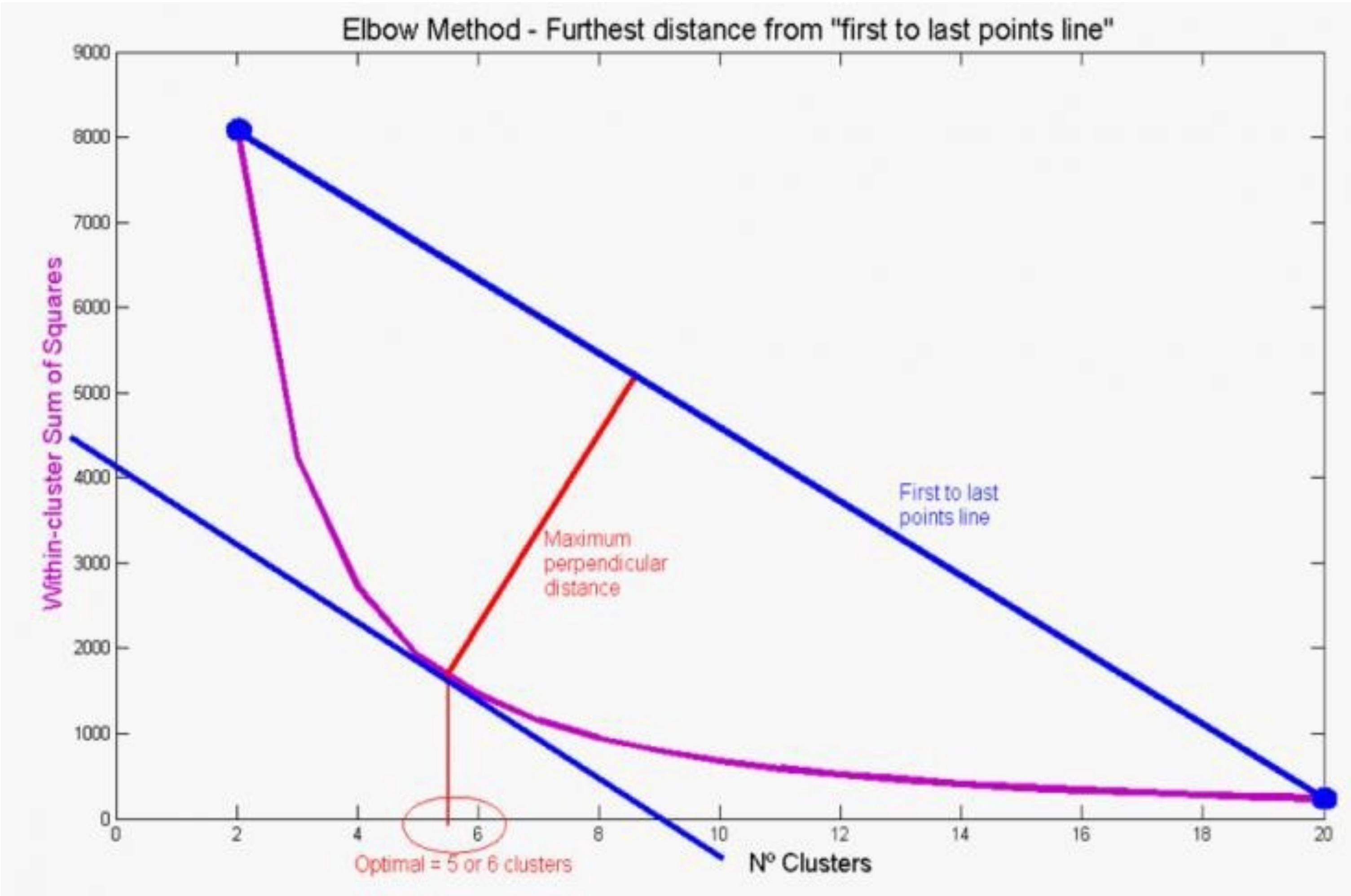


How to know K: Elbow method

- First, draw relationship between the number of clusters and Within Cluster Sum of Squares (WCSS),
- Then we select the number of clusters where the change in WCSS begins to level off.

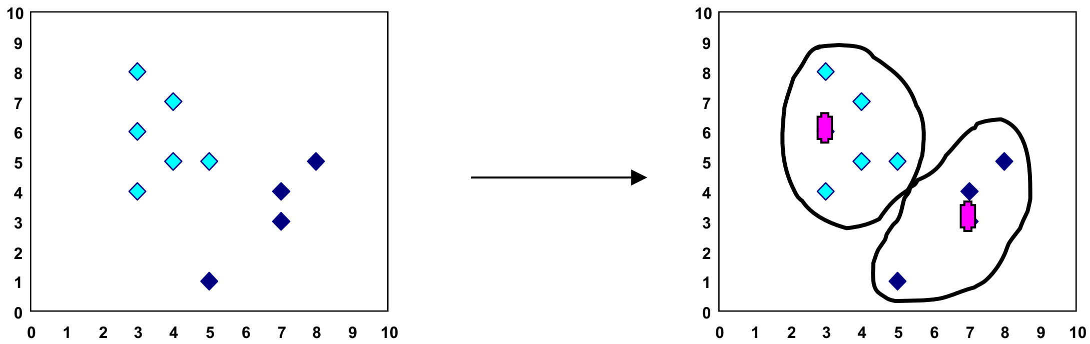


How to know K: Elbow method

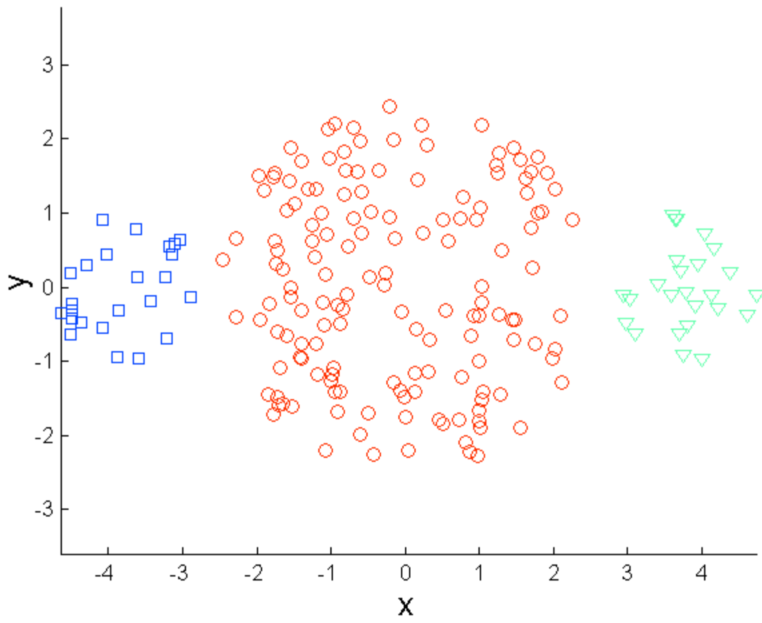


K-means variations

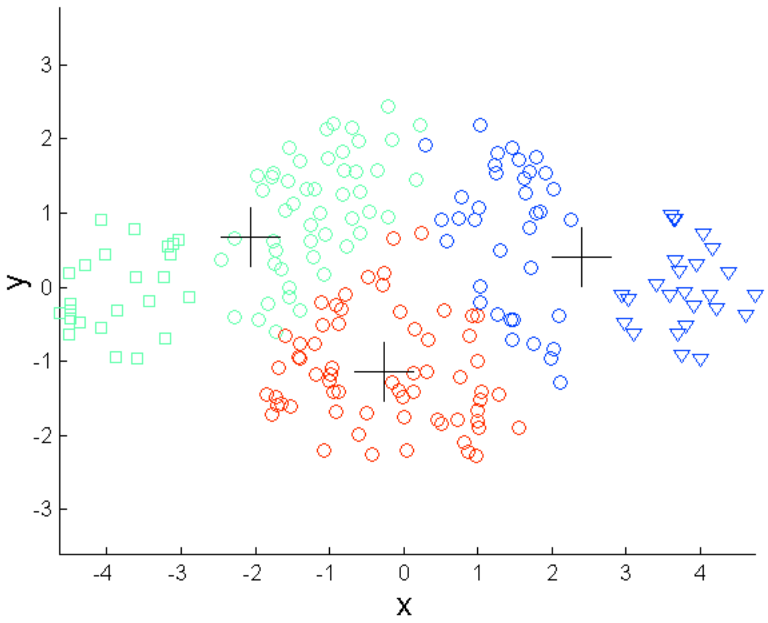
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
  - Mean of 1, 3, 5, 7, 1009 is
  - Median of 1, 3, 5, 7, 1009 is
  - Median advantage: not affected by extreme values
- K-Medoids or PAM: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.
- Etc.



Limitations of K-means: Differing Sizes

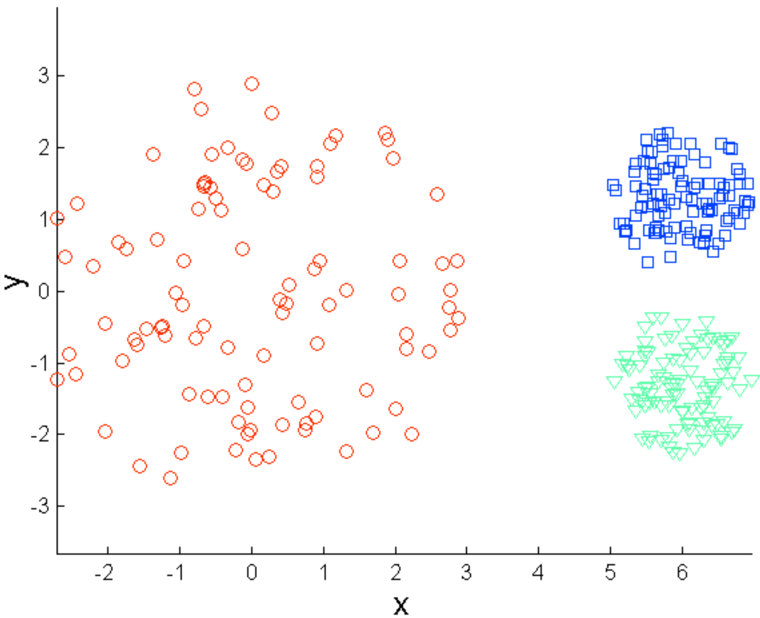


Original Points

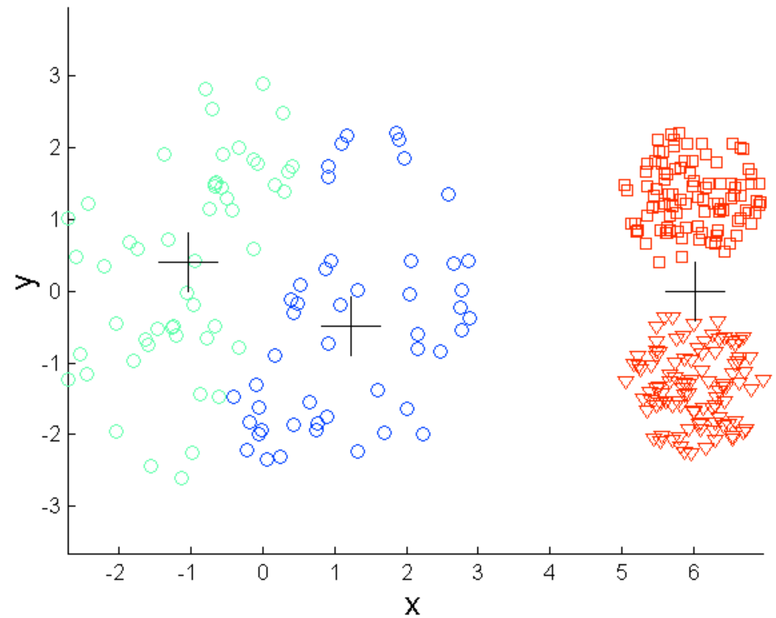


K-means (3 Clusters)

Limitations of K-means: Differing Density

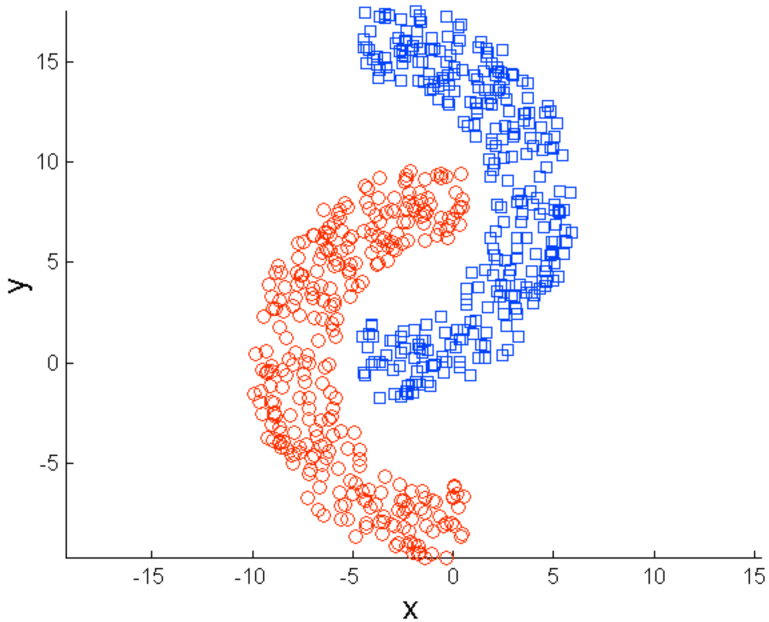


Original Points

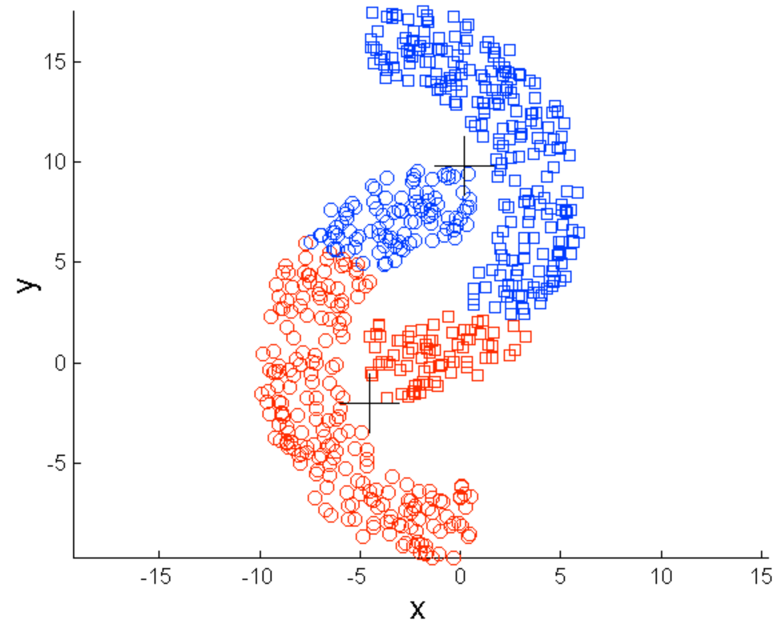


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

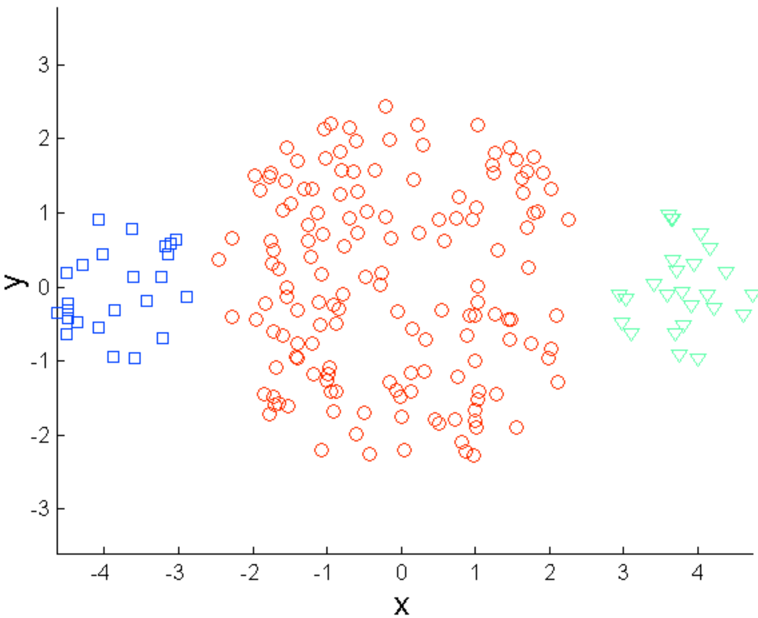


Original Points

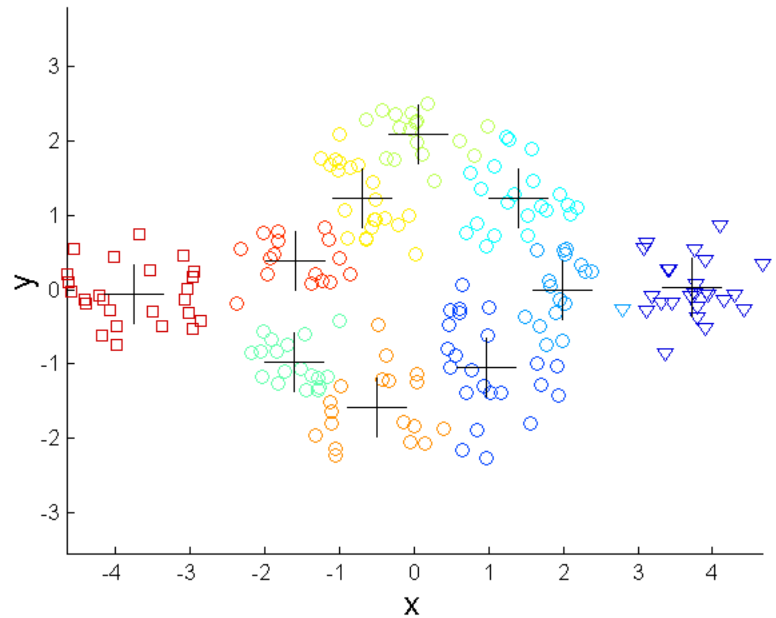


K-means (2 Clusters)

Overcoming K-means Limitations



Original Points

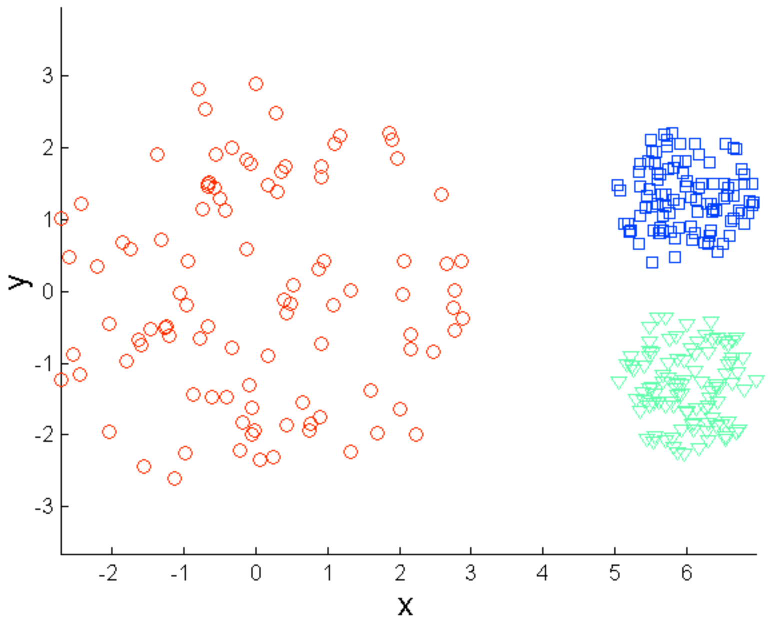


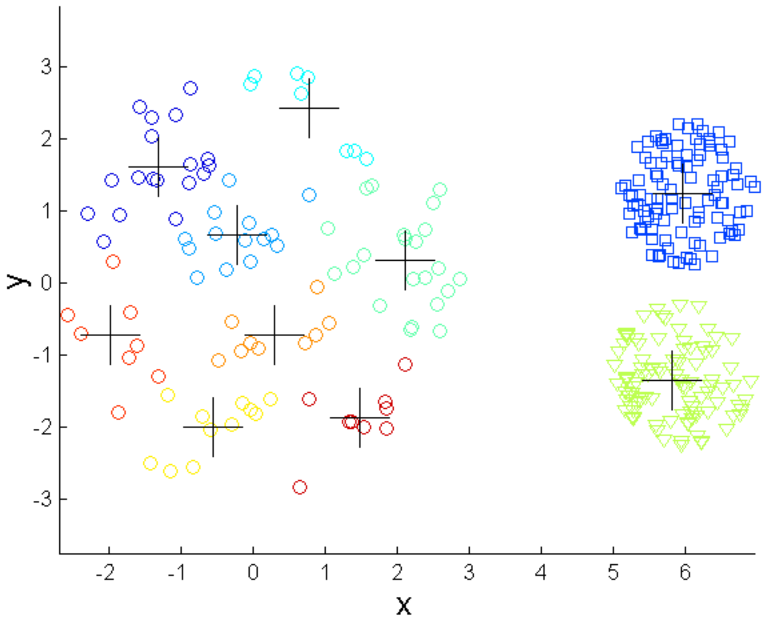
K-means Clusters

One solution is to use many clusters.

Find parts of clusters, but need to put together.

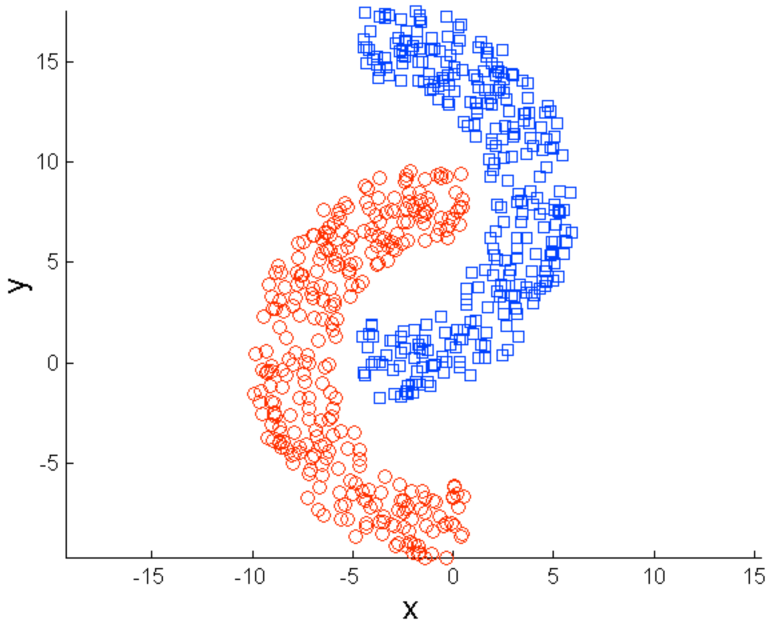
Overcoming K-means Limitations



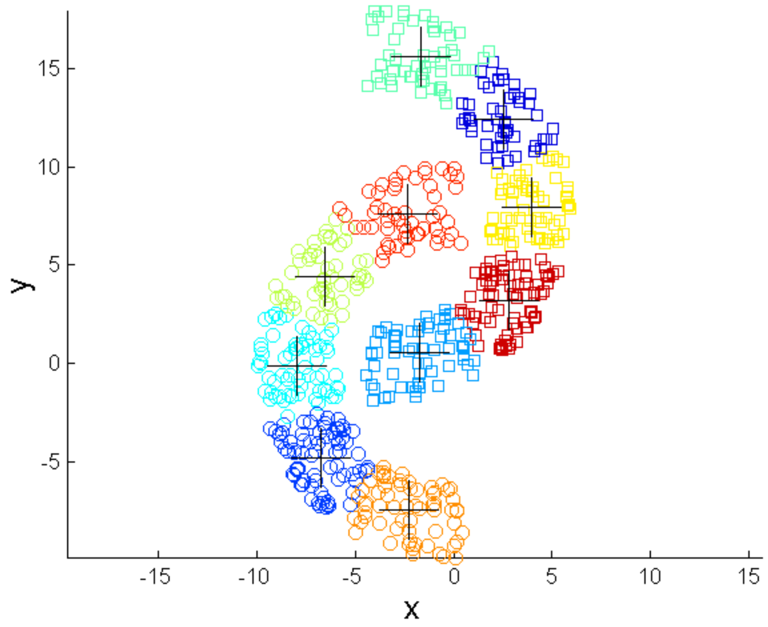


Original Points **K-means Clusters**

Overcoming K-means Limitations



Original Points



K-means Clusters

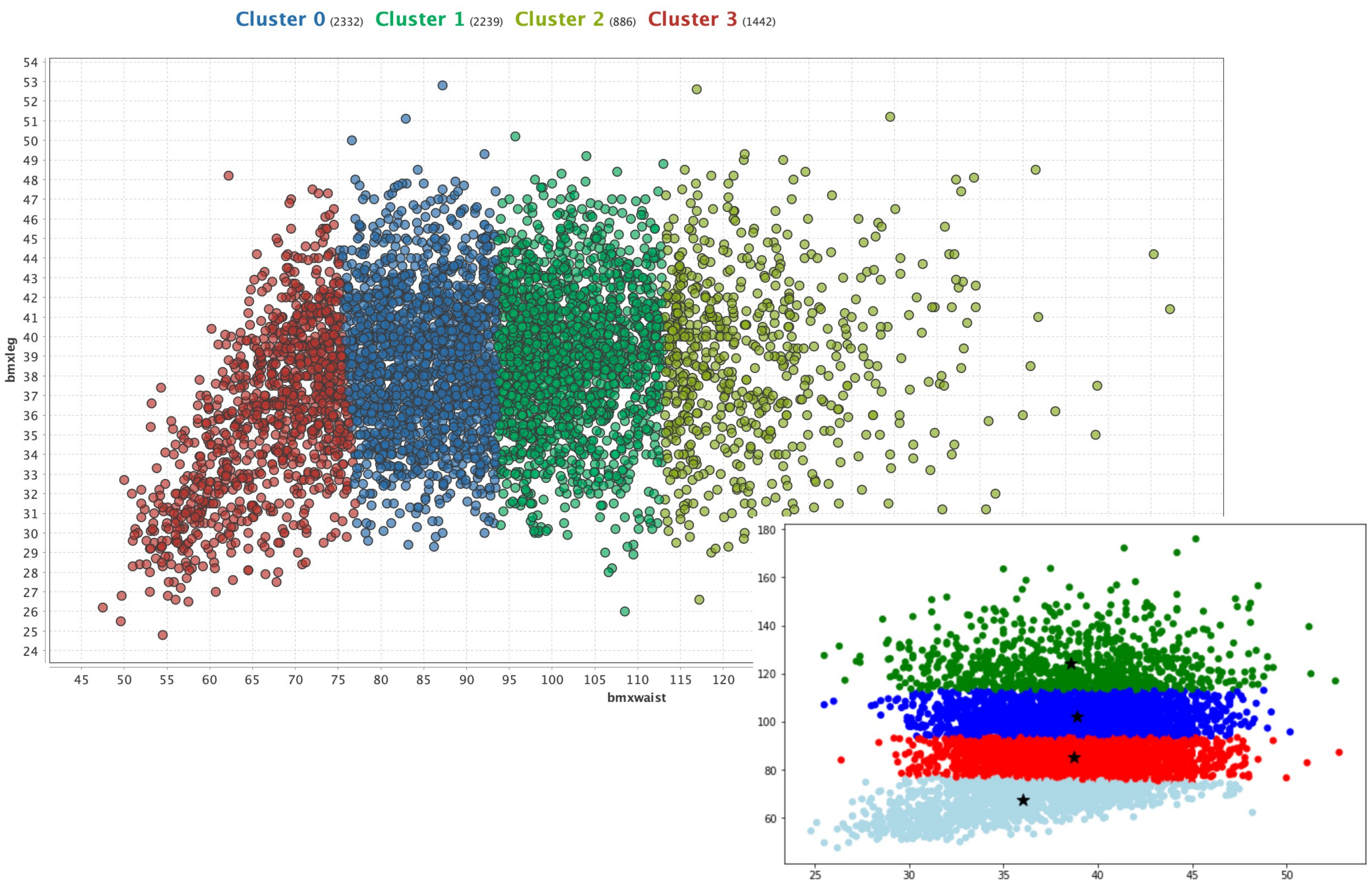
Cluster Validation

- Cluster Interpretability
- Cluster accuracy
- Cluster separation
- Etc.

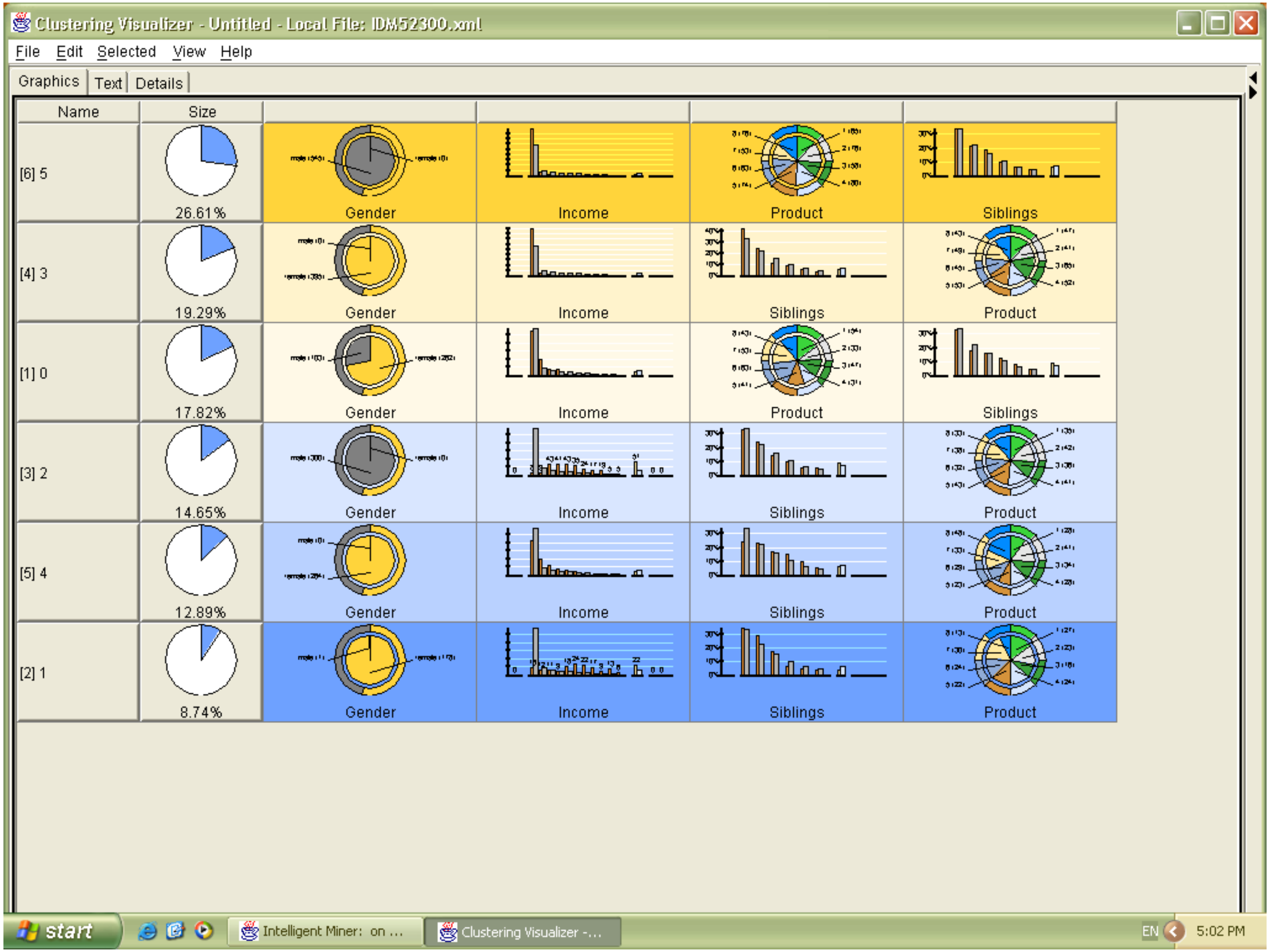
Cluster Interpretability

- **Goal:** Obtain meaningful and useful clusters
- **Possible solutions**
  - Summary statistics of each cluster
  - Use visualization
  - Distribution compared to the whole population
    - for example,
    - Mean (of each attribute in a cluster) compared to Mean of the population
  - Use dimension reduction
  - Etc.

Cluster Interpretability: 2-Dimension

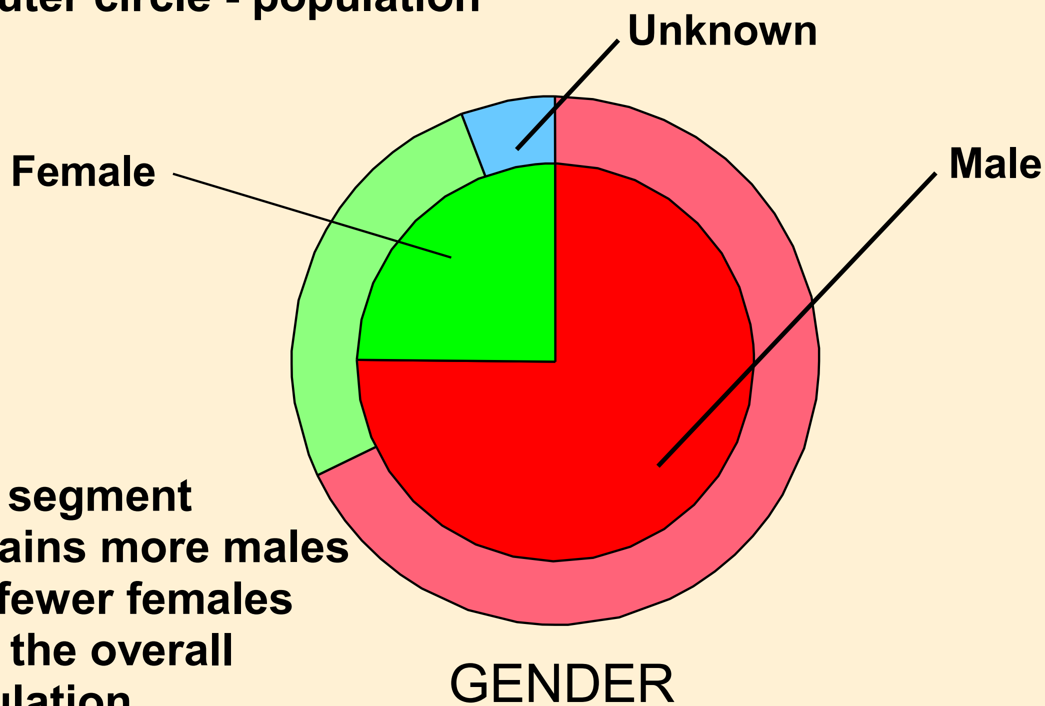


Cluster Interpretability: N-Dimensions



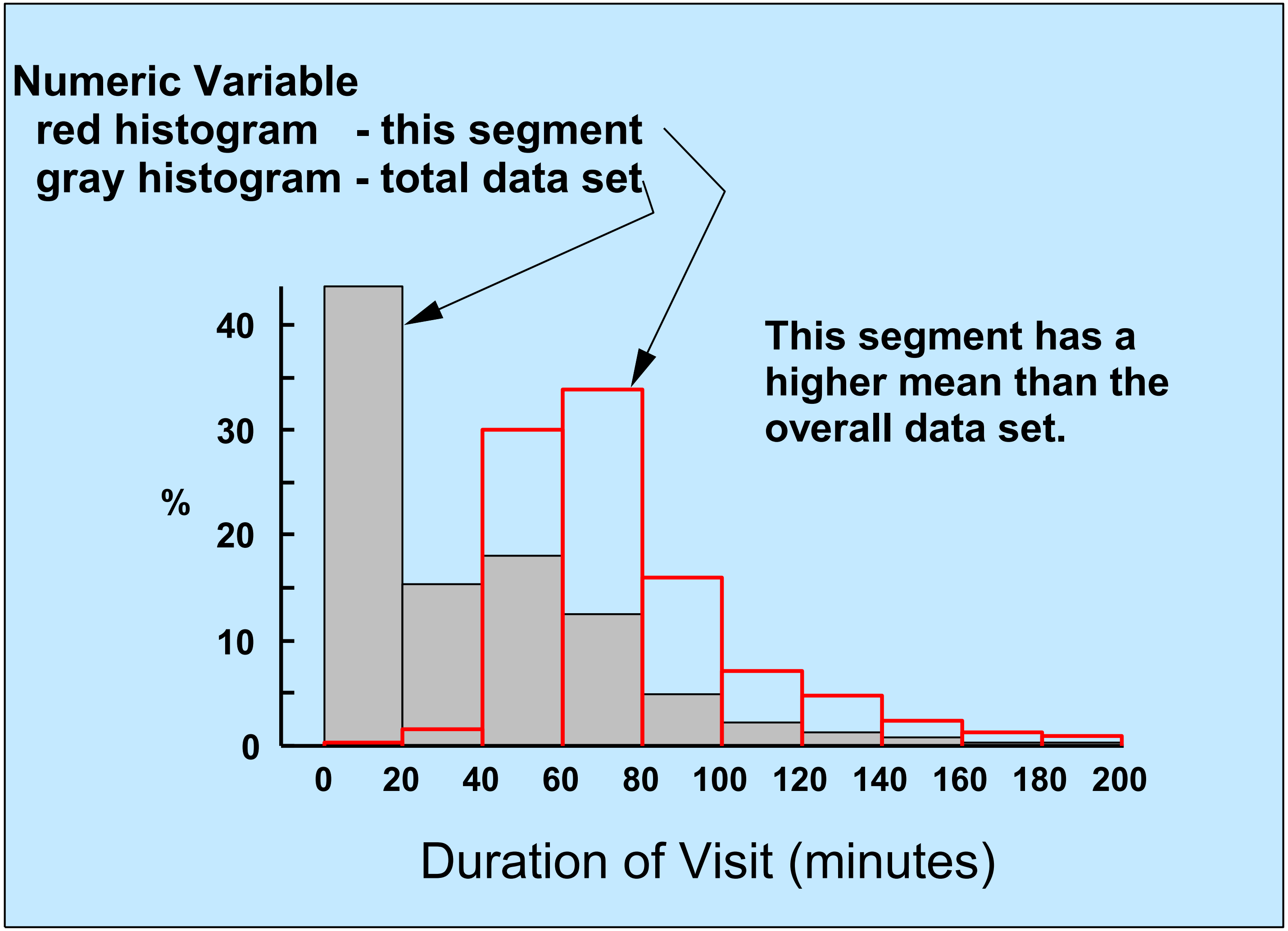
Cluster Interpretability : Categorical variables

Categorical Variable
inner circle - this segment
outer circle - population

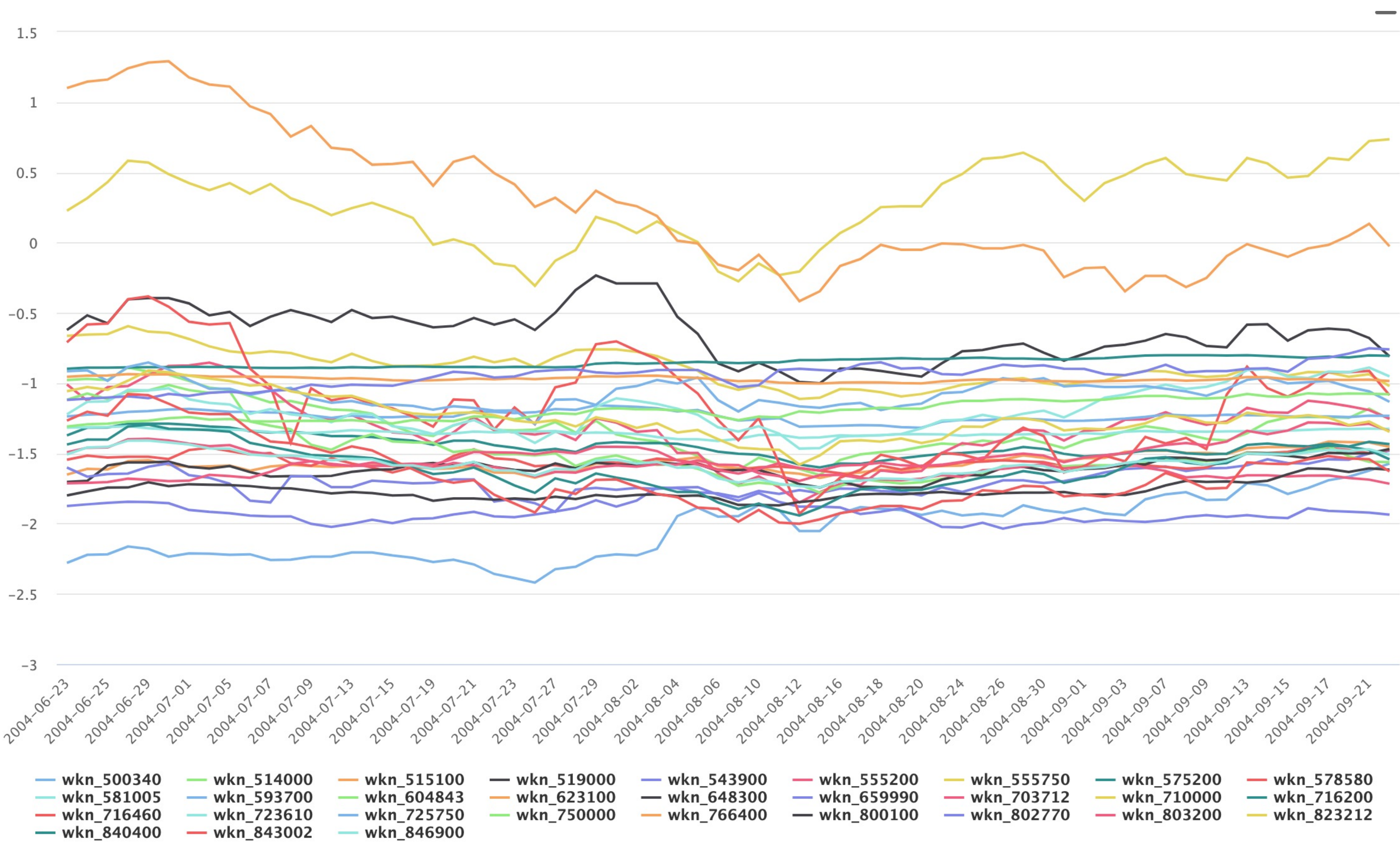


This segment contains more males and fewer females than the overall population.

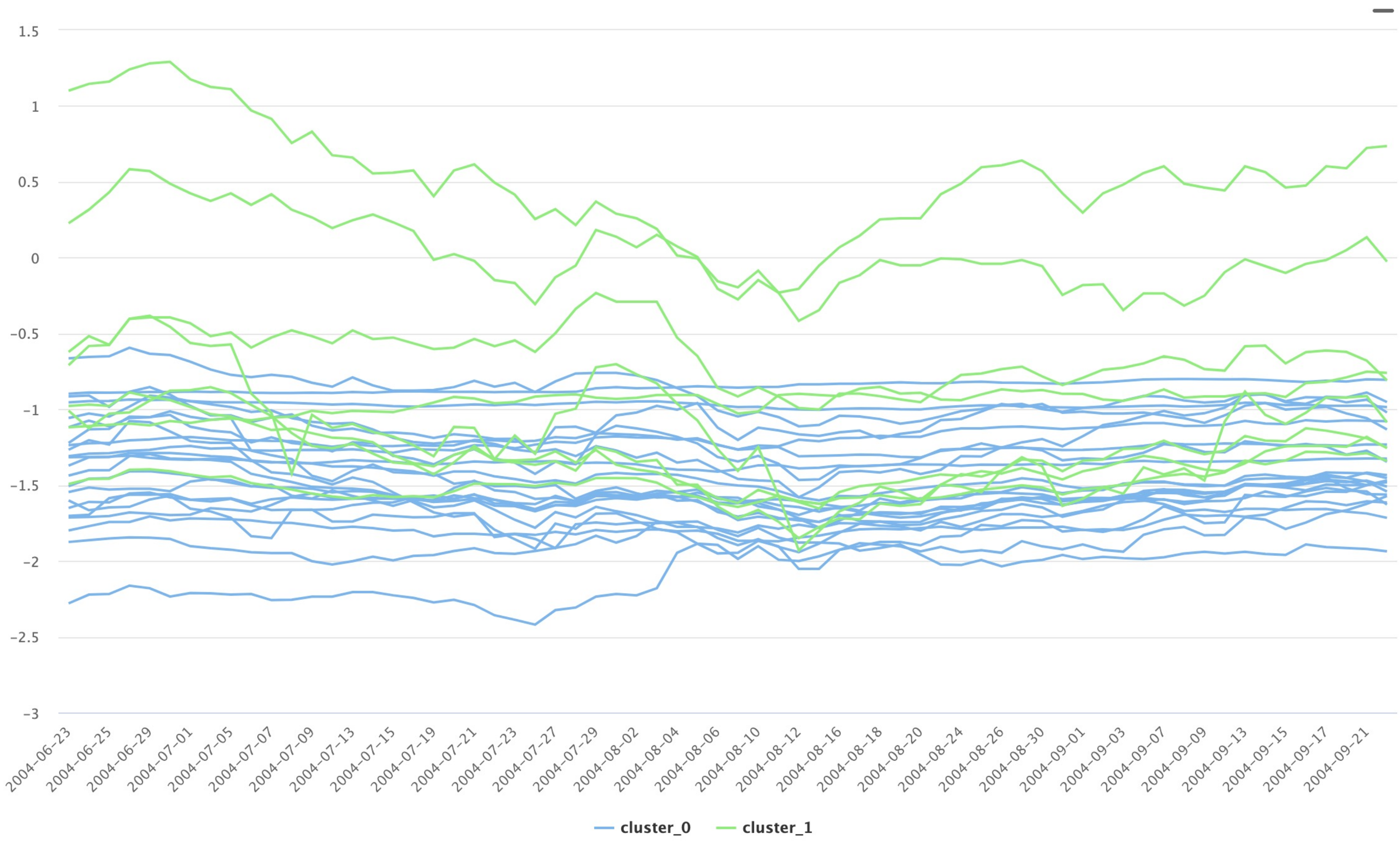
Cluster Interpretability : Numerical variables



Time series data clustering



Cluster Visualization: Time series data



Cluster Separation

Separation – Check ratio of between-cluster variation to within-cluster variation

Company	Fixed_charge	RoR	Cost	Load	Δ Demand	Sales	Nuclear	Fuel_Cost
Arizona	1.06	9.2	151	54.4	1.6	9077	0	0.628
Boston	0.89	10.3	202	57.9	2.2	5088	25.3	1.555
Central	1.43	15.4	113	53	3.4	9212	0	1.058
Commonwealth	1.02	11.2	168	56	0.3	6423	34.3	0.7
Con Ed NY	1.49	8.8	192	51.2	1	3300	15.6	2.044
Florida	1.32	13.5	111	60	-2.2	11127	22.5	1.241
Hawaiian	1.22	12.2	175	67.6	2.2	7642	0	1.652
Idaho	1.1	9.2	245	57	3.3	13082	0	0.309
Kentucky	1.34	13	168	60.4	7.2	8406	0	0.862
Madison	1.12	12.4	197	53	2.7	6455	39.2	0.623
Nevada	0.75	7.5	173	51.5	6.5	17441	0	0.768
New England	1.13	10.9	178	62	3.7	6154	0	1.897
Northern	1.15	12.7	199	53.7	6.4	7179	50.2	0.527
Oklahoma	1.09	12	96	49.8	1.4	9673	0	0.588
Pacific	0.96	7.6	164	62.2	-0.1	6468	0.9	1.4
Puget	1.16	9.9	252	56	9.2	15991	0	0.62
San Diego	0.76	6.4	136	61.9	9	5714	8.3	1.92
Southern	1.05	12.6	150	56.7	2.7	10140	0	1.108
Texas	1.16	11.7	104	54	-2.1	13507	0	0.636
Wisconsin	1.2	11.8	148	59.9	3.5	7287	41.1	0.702
United	1.04	8.6	204	61	3.5	6650	0	2.116
Virginia	1.07	9.3	174	54.3	5.9	10093	26.6	1.306

Cluster Centroids

Cluster	Fixed_charge	RoR	Cost	Load_factor
Cluster-1	0.89	10.3	202	57.9
Cluster-2	1.43	15.4	113	53
Cluster-3	1.06	9.2	151	54.4

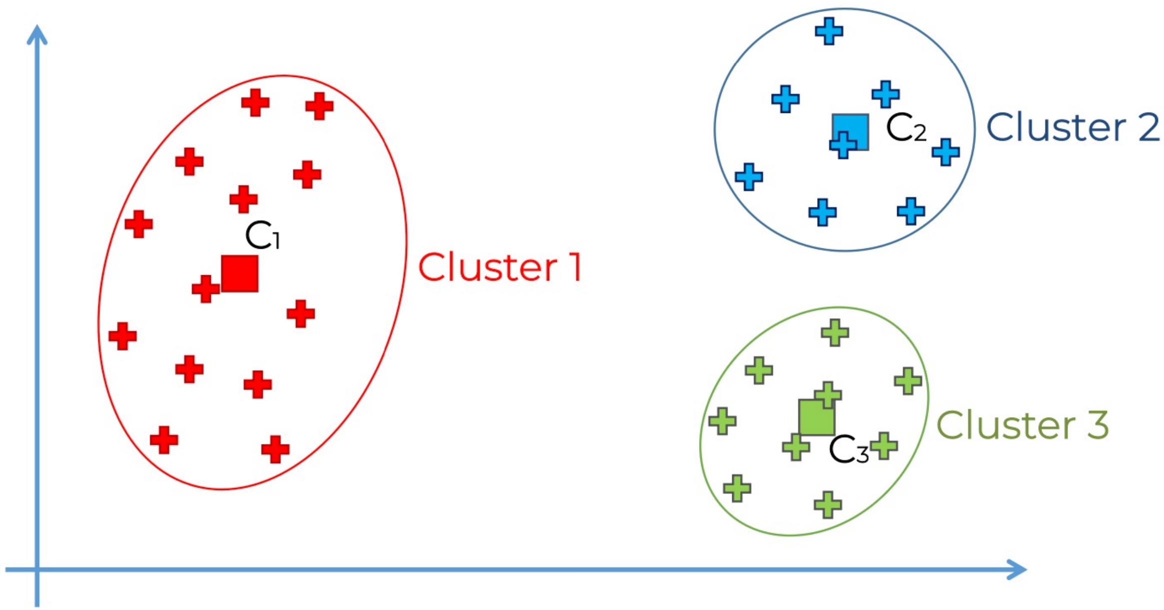
We chose $k = 3$

4 of the 8 variables are shown

Within-Cluster Squared Distance: WCSS

Within Cluster Sum of Squares (WCSS) is the sum of the squared distance between each member of the cluster and its centroid

$$\mathrm{WSS} = \sum_{i=1}^{m} (x_i - c_i)^2$$

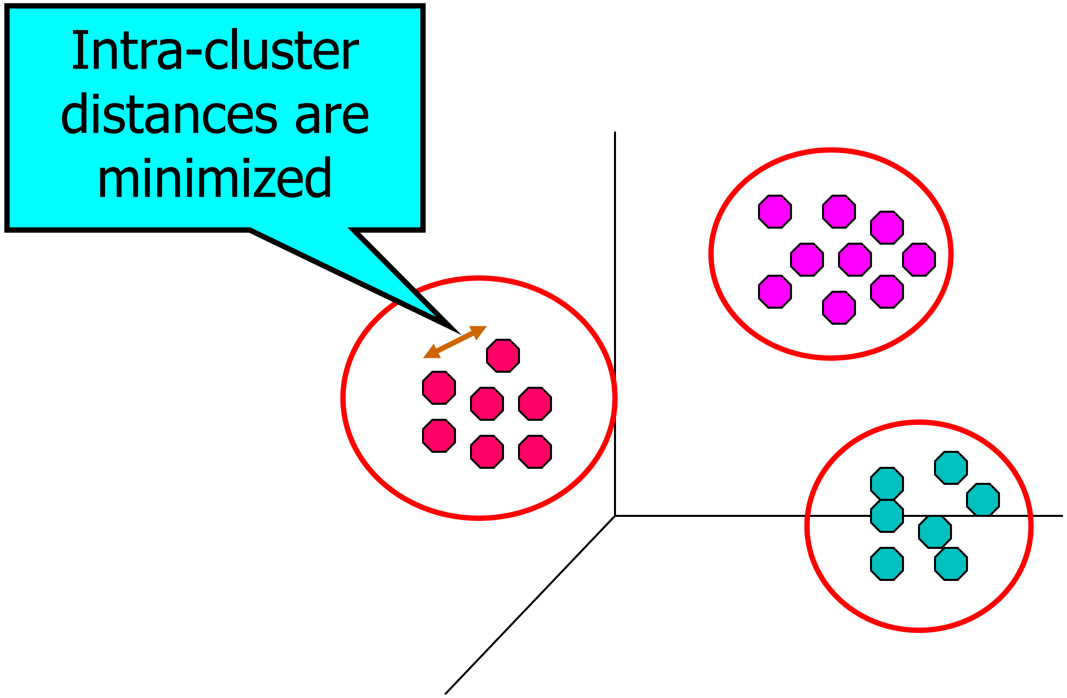


$$\sum_{\mathrm{P_i\ in\ Cluster\ 1}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_1})^2 + \sum_{\mathrm{P_i\ in\ Cluster\ 2}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_2})^2 + \sum_{\mathrm{P_i\ in\ Cluster\ 3}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_3})^2$$

Cluster Distance Performance in RapidMiner

Avg._within_centroid_distance

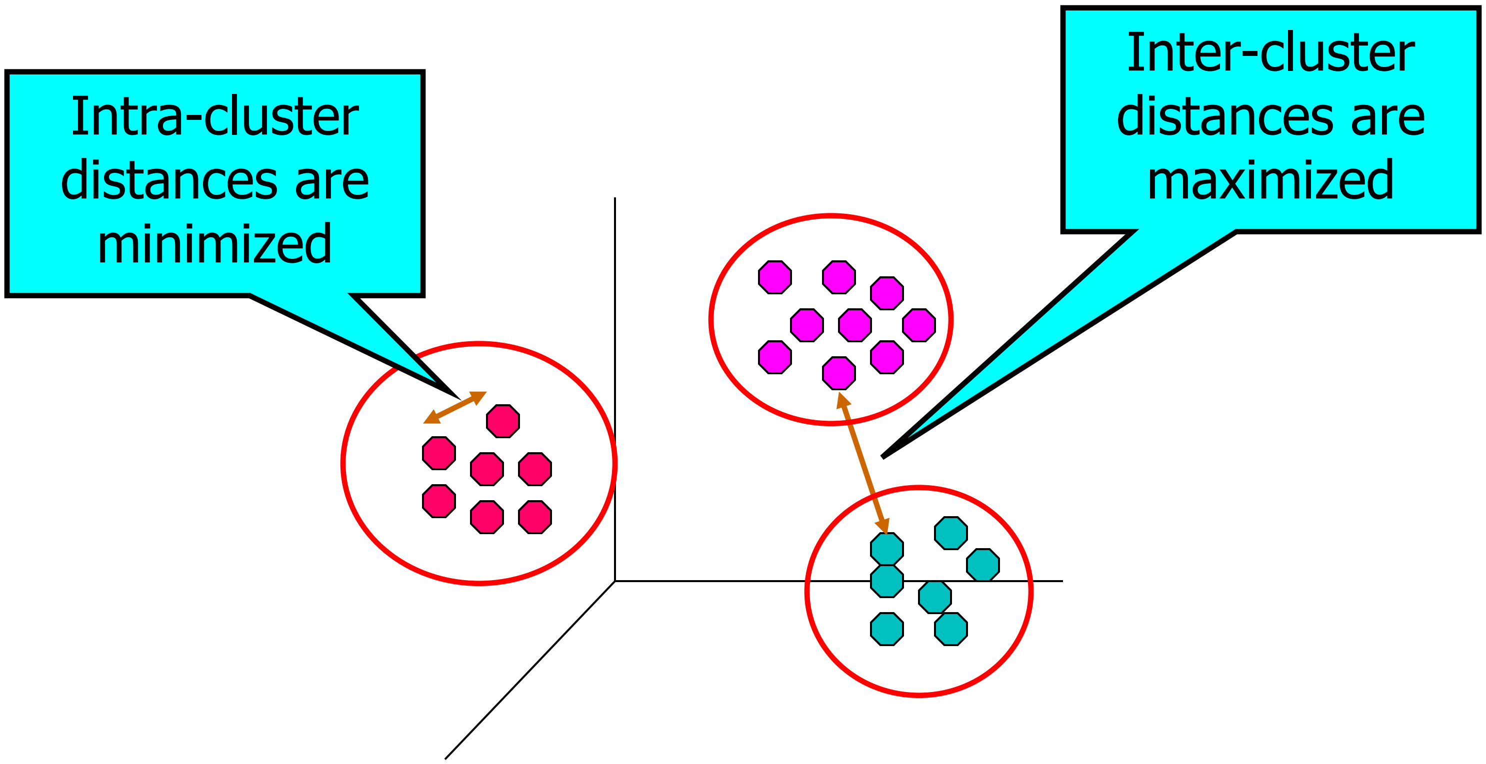
- Average within cluster distance
- Calculated by averaging the distance between the centroid and all examples of a cluster.



Cluster Distance Performance in RapidMiner

Davies–Bouldin index

- The algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a **low Davies–Bouldin** index.
- The algorithm that produces a collection of clusters with the **smallest Davies–Bouldin index** is considered the best algorithm based on this criterion.



Within-Cluster Dispersion

Data summary (In Original coordinates)

Cluster	#Obs	Average distance in cluster
Cluster-1	12	1748.348058
Cluster-2	3	907.6919822
Cluster-3	7	3625.242085
Overall	22	2230.906692

Clusters 1 and 2 are relatively tight, cluster 3 very loose

Conclusion: Clusters 1 & 2 well defined, not so for cluster 3

Next step: try again with k=2 or k=4

Distance Between Clusters

Distance between	Cluster-1	Cluster-2	Cluster-3
Cluster-1	0	5.03216253	3.16901457
Cluster-2	5.03216253	0	3.76581196
Cluster-3	3.16901457	3.76581196	0

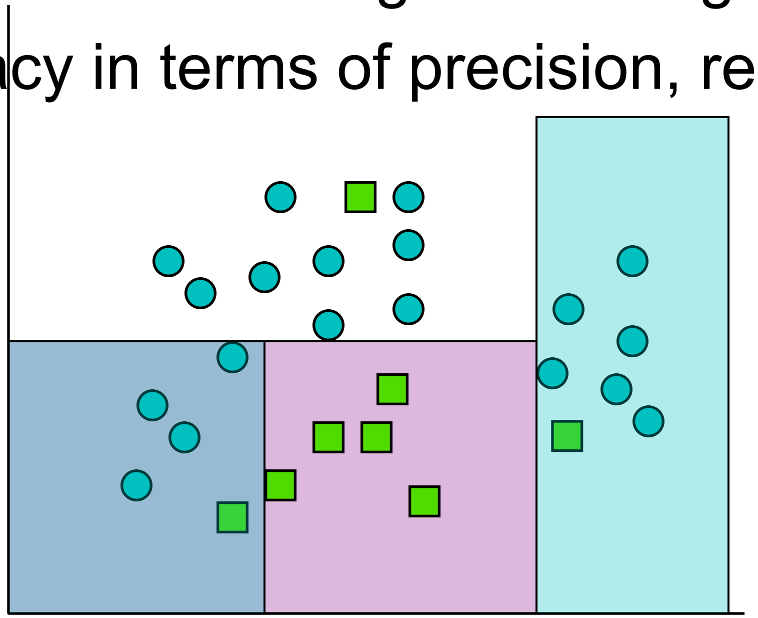
Clusters 1 and 2 are relatively well-separated from each other, while cluster 3 not as much

Use of Labelled Data for Evaluating Cluster Accuracy

Use of ground truth labels

Accuracy– Benchmarking on existing labels

– Accuracy in terms of precision, recall



Summary

- Cluster analysis is an exploratory tool, very useful only when it produces meaningful clusters
- 3 main types of clustering
  - Partitioning clustering is computationally cheap and more stable; requires user to set *k*
  - Hierarchical clustering gives visual representation of different levels of clustering
  - Density-based clustering is able to discover clusters of arbitrary shape
- Other clustering approaches
  - Grid-based approach
  - Model-based
  - Frequent pattern-based
  - Constraint-based
  - Link-based clustering
  - Etc.

Python: K-Mean

```
# K-Means Clustering
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values
# y = dataset.iloc[:, 3].values

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(2, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(2, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# Fitting K-Means to the dataset
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 0)
y_kmeans = kmeans.fit_predict(X)

# Visualising the clusters
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroids')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```