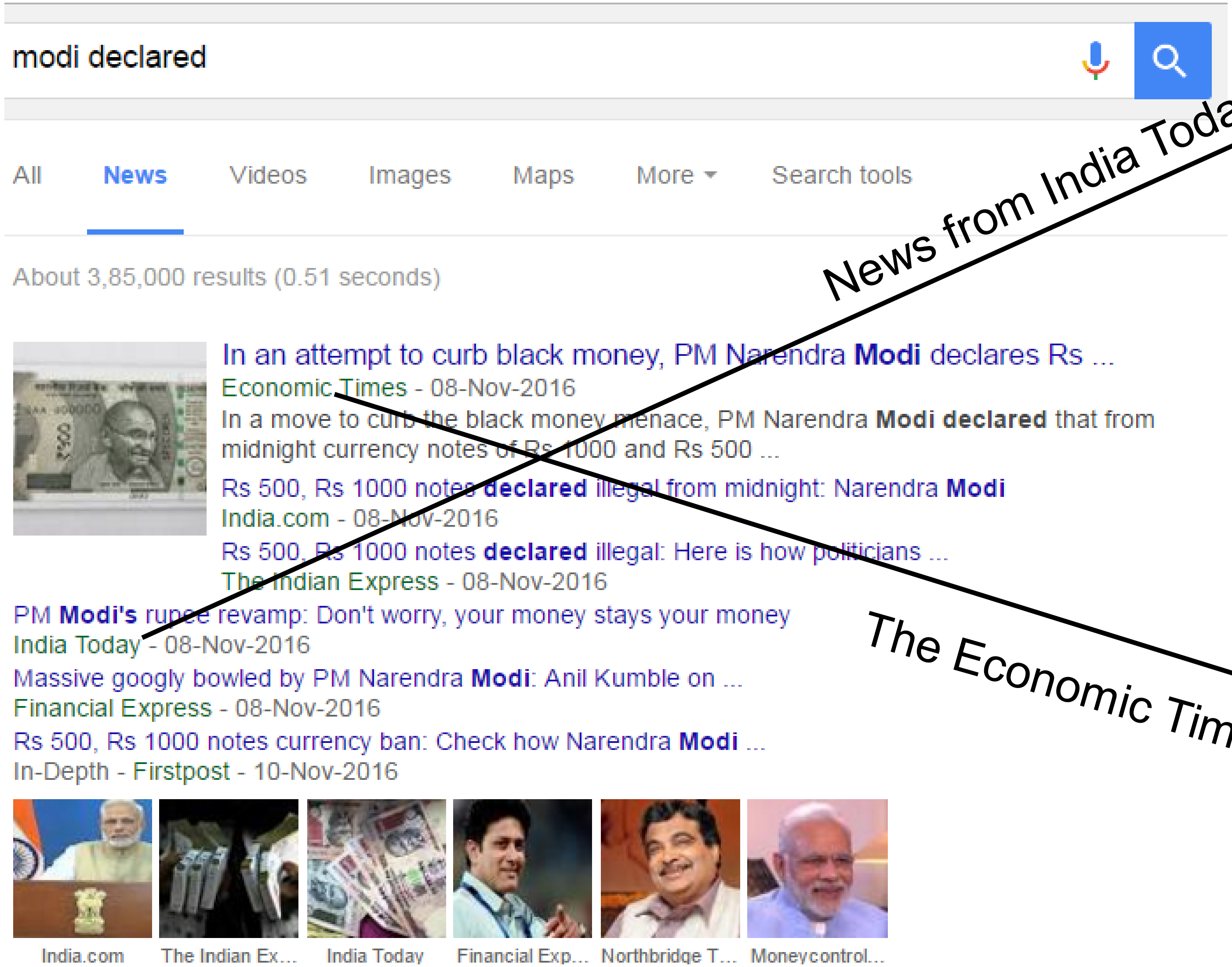
Practical machine learning

PG – DBDA, C-DAC BANGALORE

SESSION 2: CLUSTERING

CLUSTERING EXAMPLE – GROUPING OF NEWS ARTICLES

Clusters of News articles from Google



modi declared

All News Videos Images Maps More Search tools

About 3,85,000 results (0.51 seconds)

In an attempt to curb black money, PM Narendra **Modi** declares Rs ...
Economic Times - 08-Nov-2016
In a move to curb the black money menace, PM Narendra **Modi declared** that from midnight currency notes of Rs 1000 and Rs 500 ...

Rs 500, Rs 1000 notes **declared** illegal from midnight: Narendra **Modi**
India.com - 08-Nov-2016

Rs 500, Rs 1000 notes **declared** illegal: Here is how politicians ...
The Indian Express - 08-Nov-2016

PM **Modi's** rupee revamp: Don't worry, your money stays your money
India Today - 08-Nov-2016

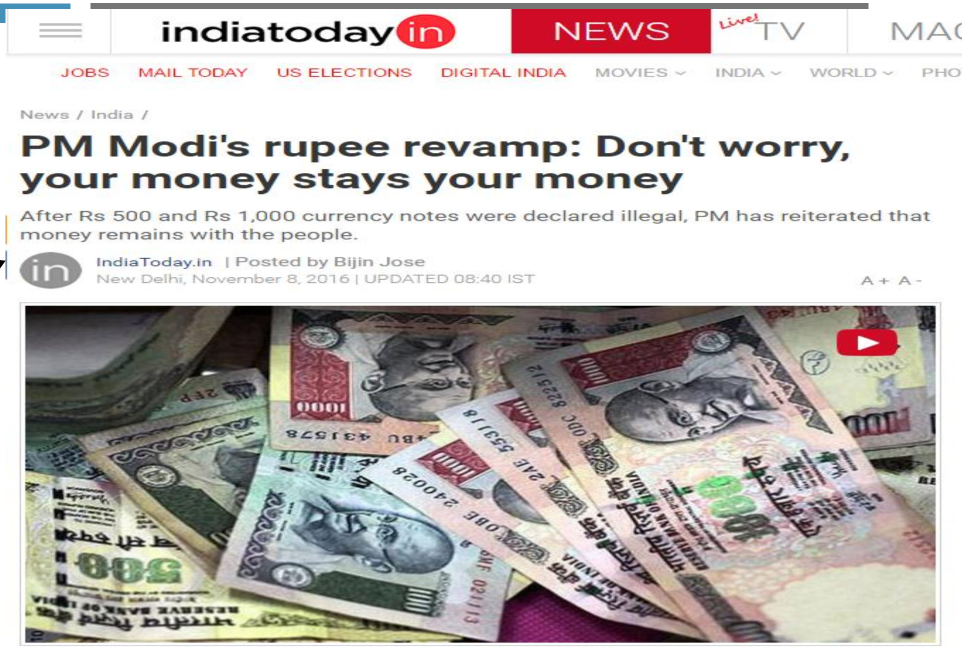
Massive googly bowled by PM Narendra **Modi**: Anil Kumble on ...
Financial Express - 08-Nov-2016

Rs 500, Rs 1000 notes currency ban: Check how Narendra **Modi** ...
In-Depth - Firstpost - 10-Nov-2016

India.com | The Indian Ex... | India Today | Financial Exp... | Northbridge T... | Moneycontrol...

News from India Today

The Economic Times



indiatoday.in NEWS Live! TV MAG

JOBS MAIL TODAY US ELECTIONS DIGITAL INDIA MOVIES INDIA WORLD PHO

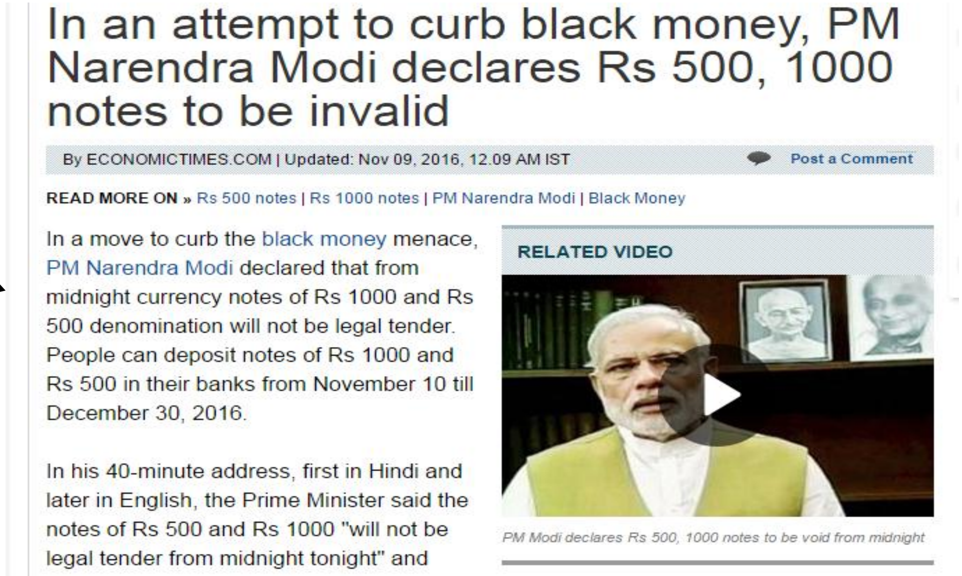
News / India /

PM Modi's rupee revamp: Don't worry, your money stays your money

After Rs 500 and Rs 1,000 currency notes were declared illegal, PM has reiterated that money remains with the people.

IndiaToday.in | Posted by Bijin Jose
New Delhi, November 8, 2016 | UPDATED 08:40 IST

A + A -



In an attempt to curb black money, PM Narendra Modi declares Rs 500, 1000 notes to be invalid

By ECONOMICTIMES.COM | Updated: Nov 09, 2016, 12.09 AM IST Post a Comment

READ MORE ON » Rs 500 notes | Rs 1000 notes | PM Narendra Modi | Black Money

In a move to curb the black money menace, PM Narendra Modi declared that from midnight currency notes of Rs 1000 and Rs 500 denomination will not be legal tender. People can deposit notes of Rs 1000 and Rs 500 in their banks from November 10 till December 30, 2016.

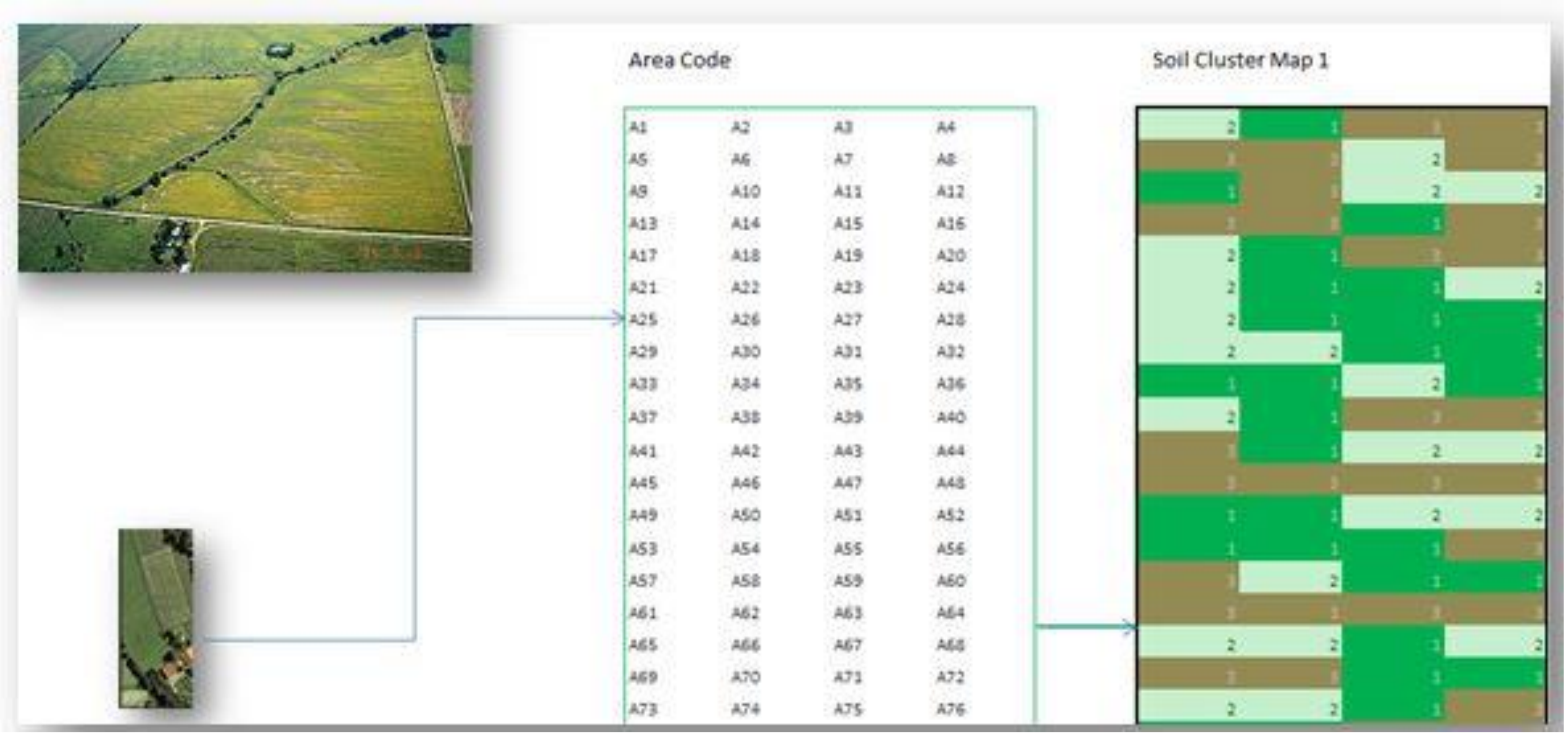
In his 40-minute address, first in Hindi and later in English, the Prime Minister said the notes of Rs 500 and Rs 1000 "will not be legal tender from midnight tonight" and

RELATED VIDEO

PM Modi declares Rs 500, 1000 notes to be void from midnight

CLUSTERING

- Finds natural grouping of objects/instances using unlabeled data
- A Cluster will have a subset of objects which are similar
- We may not even know what we are looking for, no predefined classes
- Clustering is used for knowledge discovery rather than prediction



Area Code

A1	A2	A3	A4
A5	A6	A7	A8
A9	A10	A11	A12
A13	A14	A15	A16
A17	A18	A19	A20
A21	A22	A23	A24
A25	A26	A27	A28
A29	A30	A31	A32
A33	A34	A35	A36
A37	A38	A39	A40
A41	A42	A43	A44
A45	A46	A47	A48
A49	A50	A51	A52
A53	A54	A55	A56
A57	A58	A59	A60
A61	A62	A63	A64
A65	A66	A67	A68
A69	A70	A71	A72
A73	A74	A75	A76

Soil Cluster Map 1

2	1	3	3
3	3	2	3
1	3	2	2
3	3	1	3
2	1	3	3
2	1	1	2
2	1	1	1
2	2	1	1
1	1	2	1
2	1	3	3
3	1	2	2
3	3	3	3
1	1	2	2
1	1	1	3
3	2	1	1
3	3	3	3
2	2	1	2
3	3	1	1
2	2	1	3

Soil Map - Agriculture

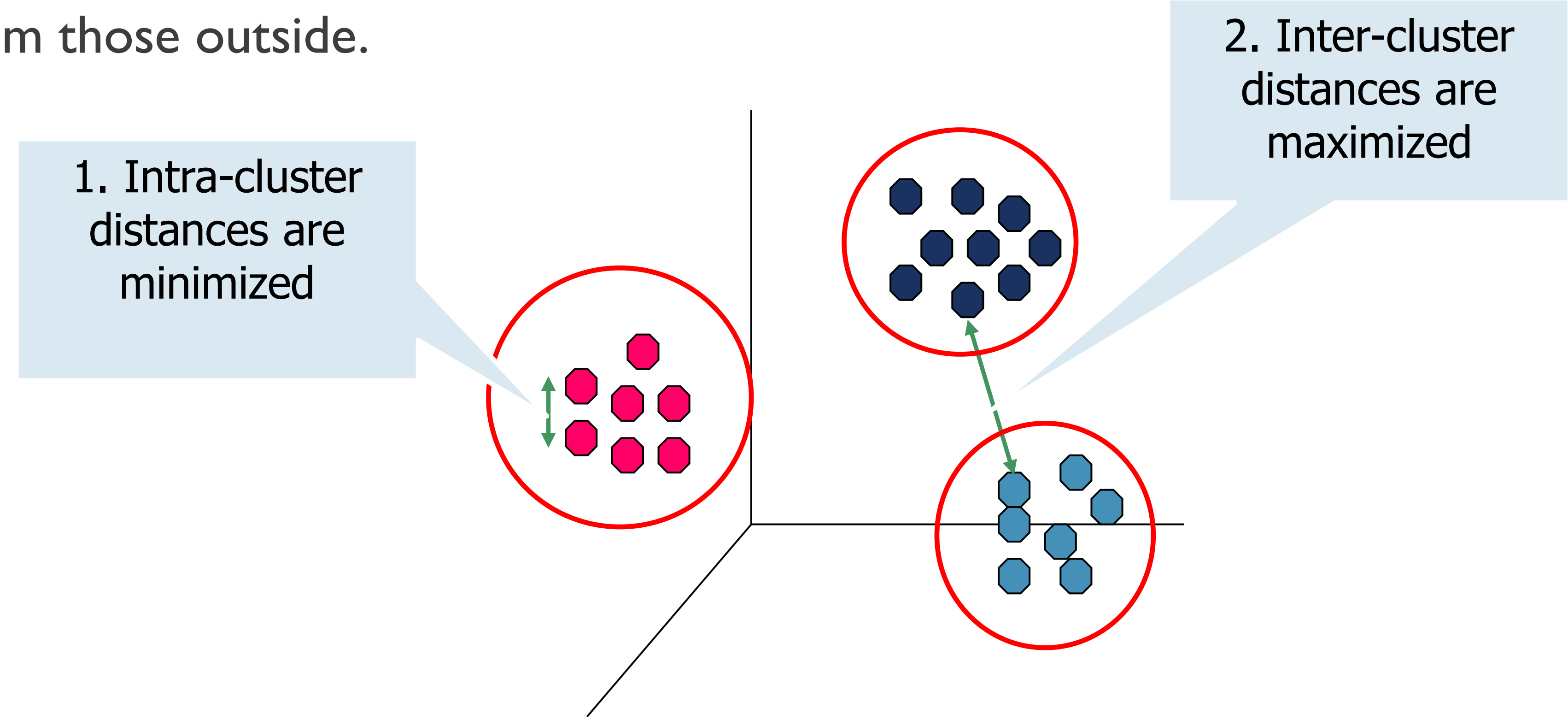
Image Credits: https://www.linkedin.com/pulse/k-means-clustering-field-maps-krishna-mohan?trk=sushi_topic_posts_guest

CLUSTERING…

Without advance knowledge of what comprises a cluster, how can a computer possibly know where one group ends and another begins?

Answer:

- Clustering is guided by the principle that items inside a cluster should be:
  - very similar to each other,
  - but very different from those outside.



1. Intra-cluster distances are minimized

2. Inter-cluster distances are maximized

CLUSTERING…

Clustering creates new data…

How?

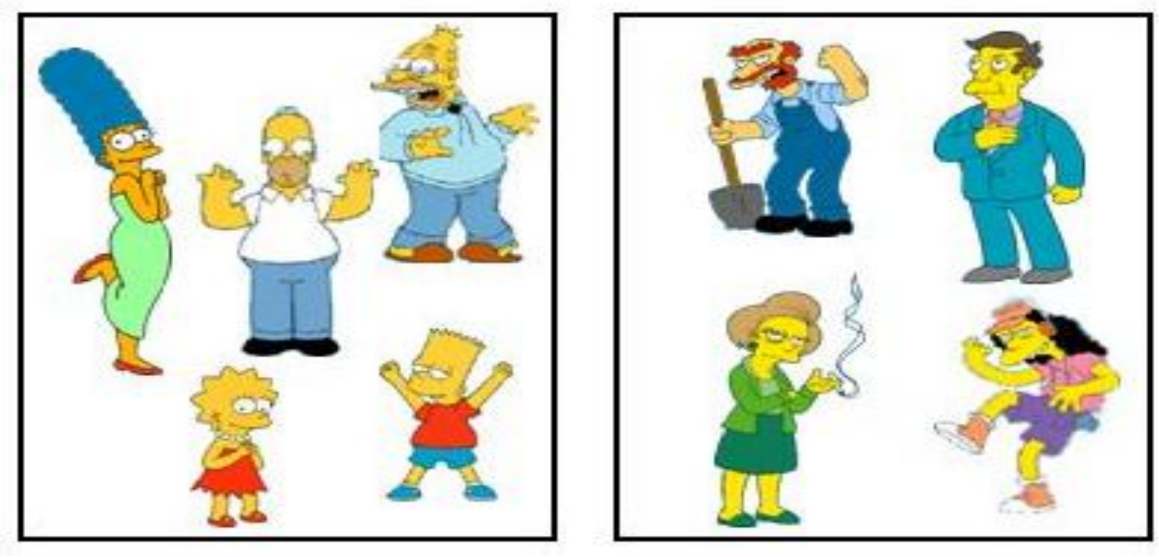
- Unlabeled examples are given a cluster label that has been inferred entirely from the relationships within the data.
- Sometimes, see the clustering task referred to as **unsupervised classification** because, in a sense, it classifies unlabeled examples.

CLUSTERING REQUIREMENTS

- Scalability
- Dealing with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- Handling the curse of dimensionality
- Interpretability and usability

TYPES OF CLUSTERING

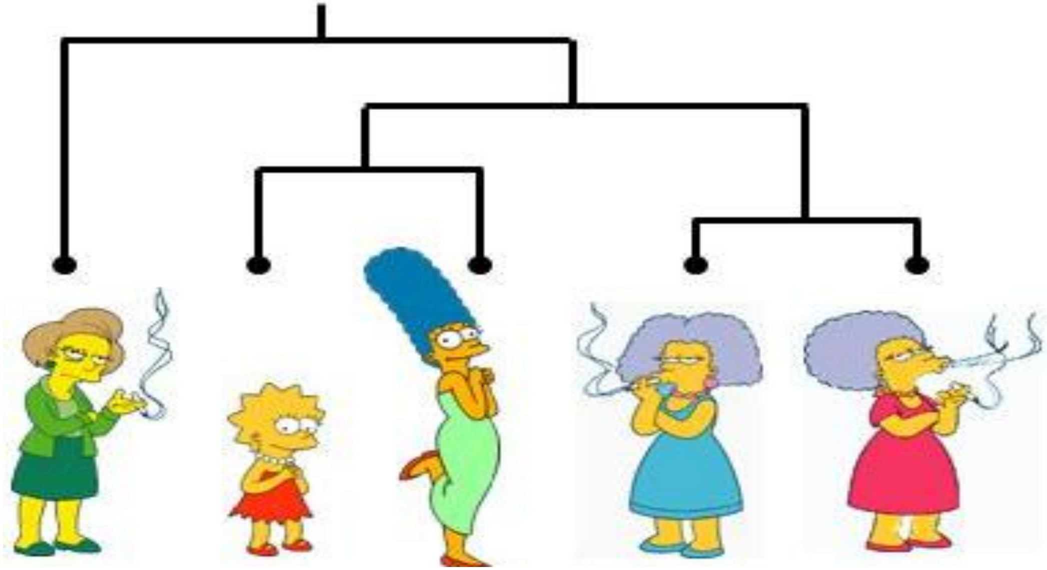
Partitional Clustering



Group the objects into different clusters.

- K-Means
- Fuzzy K-Means

Hierarchical Clustering



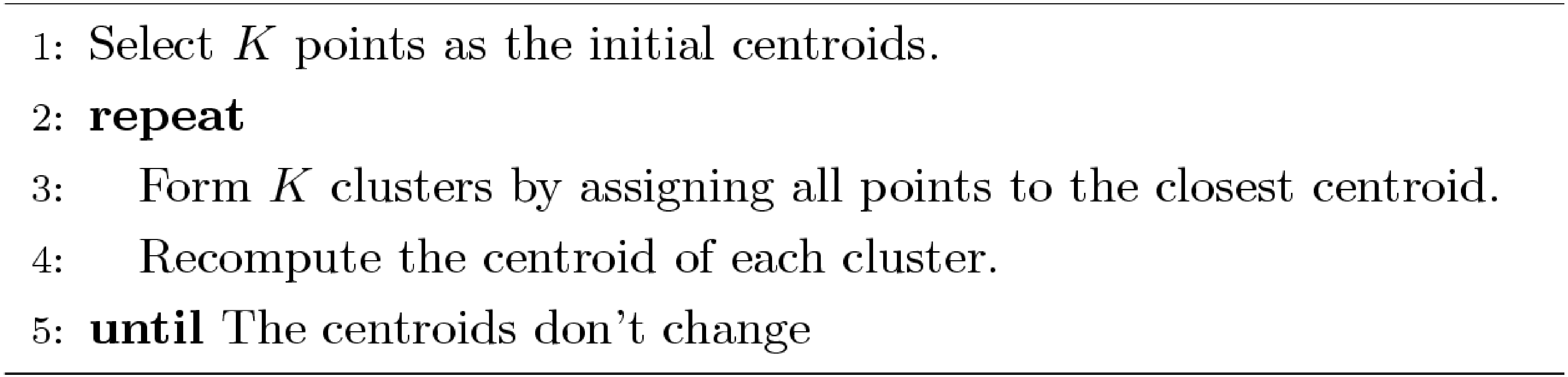
Set of nested clusters organized as a tree.

- Single Link
- Multi Link

Ref: http://www.cs.cmu.edu/afs/andrew/course/15/381-f08/www/lectures/clustering.pdf

K-MEANS CLUSTERING

- *K*-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms
- Aims to partition N observations into K clusters



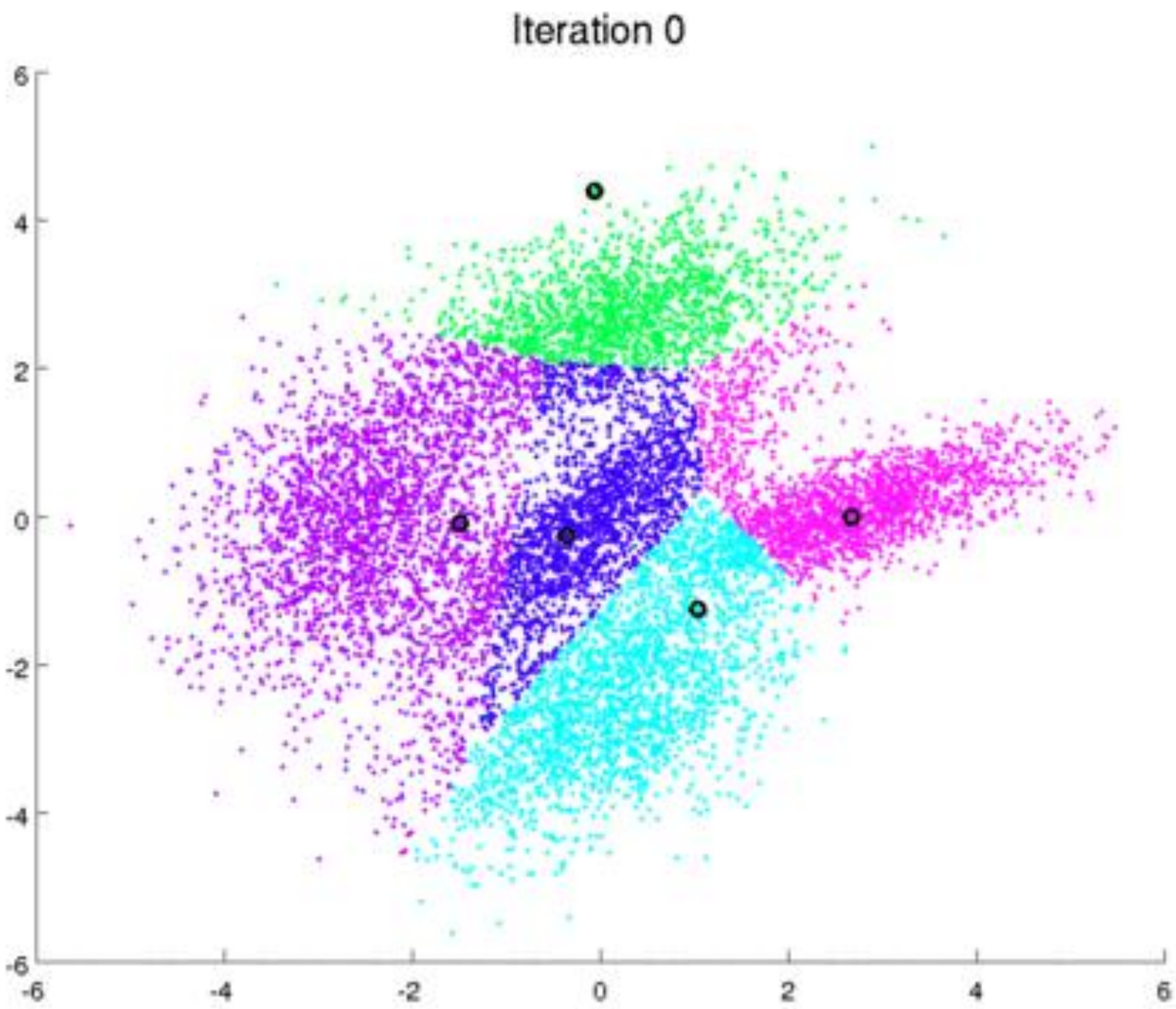
```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

K-MEANS CLUSTERING – STEPS

Algorithm:

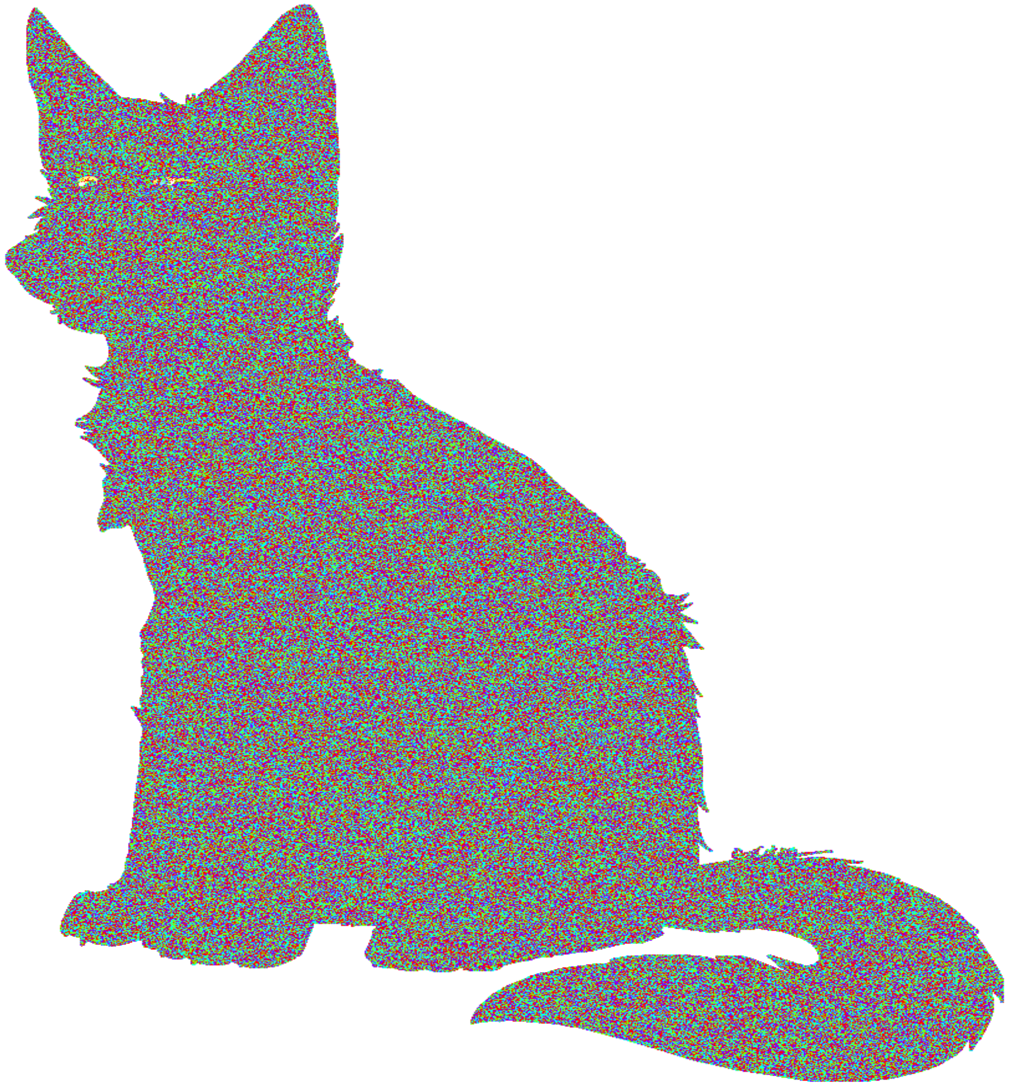
1. **Partition** the objects into “k” non-empty subsets.
2. **Identify** the cluster **centroids** (mean values) of the current partition.
3. **Assign** each point to a specific cluster.
4. **Compute the distance** from each point, **allot points to the cluster** from where the distance from the centroid is minimum.
5. After re-allotting the points, **find the new centroid** of the new cluster formed.
6. **Repeat** the above steps until we get the best solution.

K-MEANS CLUSTERING – HOW IT WORKS?



K-MEANS CLUSTERING – IMAGES - HOW IT WORKS?

- Divides the image into K clusters based on color.
- In this case it was 5 colors.
- Here the colors that represent each cluster are random instead of being the original colors they were in the image.



STOPPING/CONVERGENCE CRITERION – WHEN TO STOP CLUSTERING??

- NO or Minimum re-assignments of data points to different clusters **OR**
- NO or Minimum change in centroid positions, **OR**
- Minimum decrease in the Sum of Squared Error (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

  - Ci is the jth cluster, mj is the centroid of cluster Cj (the mean vector of all the data points in Cj), and
  - dist(x, mj) is the distance between data point x and centroid mj.

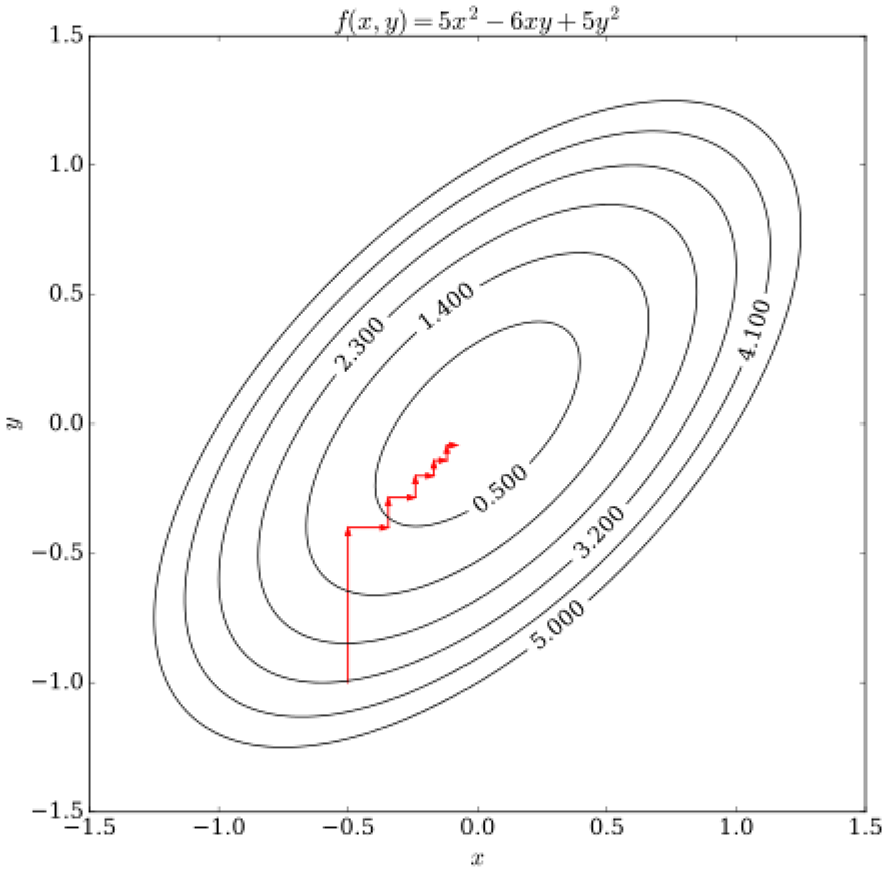
THE COST FUNCTION OF K-MEANS

- A strength of the K-Means clustering algorithm efficiency (Big O Notation) with O(nkt),

  where n, k, t equal the number of iterations, clus

$$\mathcal{L}(\Delta) = \sum_{i=1}^{n} ||x_i - \mu_{k(i)}||^2 = \sum_{k=1}^{K}$$

- K-Means is fundamentally a coordinate descent algor
- Coordinate descent serves to minimize a multivariat time. The inner-loop of k-means repeatedly minimi while holding μ fixed, and then minimizes with respe
- This distortion function is a non-convex function, not guaranteed to converge to the global minimum,



- This is why it is optimal to run k-means many times using random initialization values for the clusters, then selecting the run with lowest distortion or cost.

STANDARDIZATION

- In general you need to have all of the features in the same scale.
- The reason for this is because otherwise the feature with the highest range will have more weight on the clustering process.
- For example, if you have a feature with range (0,100) and another with range (0,1), the last will have no effect on the clustering.
- Since clustering relies on distances you can see how the feature with the smallest range contributes almost nothing when a distance is calculated.

STANDARDIZATION

- Example:
  - Your data set has:
    - 3 colors: blue, brown, green and
    - 2 continuous variables: age and weight
  - Data Before standardization:

blue	brown	green	age	weight
0	1	0	25	150
1	0	0	26	140
0	0	1	26	130

  - Data After Standardization:

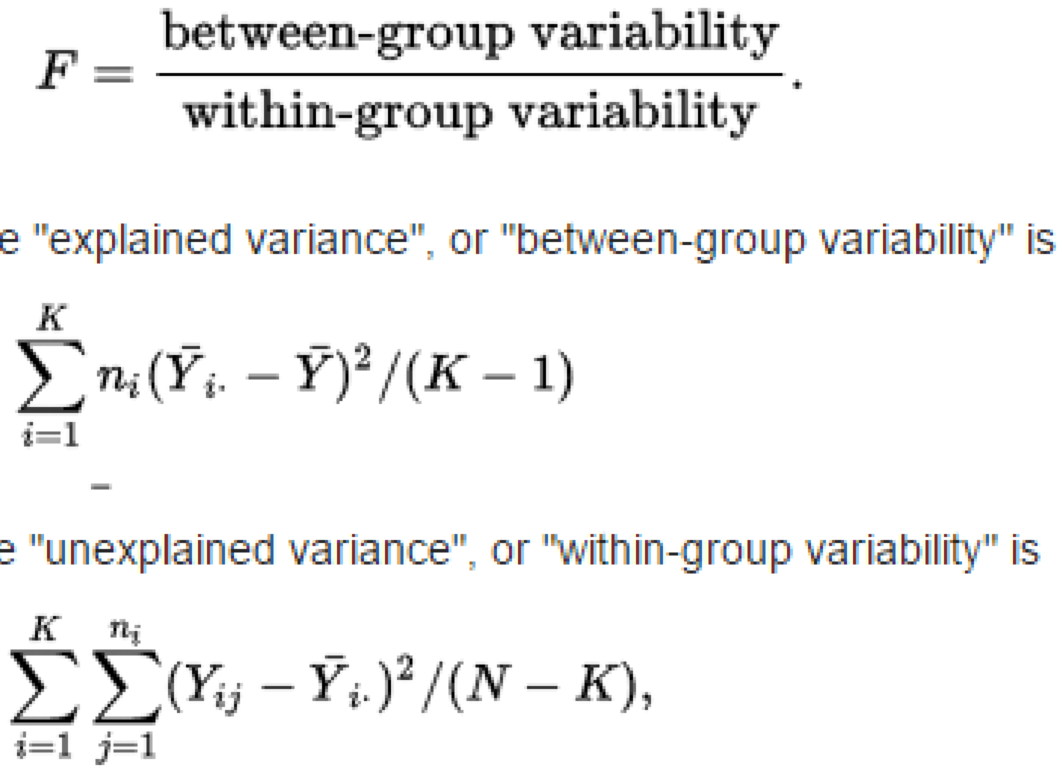
blue	brown	green	age	weight
0	1	0	0.8	1
1	0	0	1	0.8
0	0	1	1	0.6

HOW TO CHOOSE K?

- There isn't a general theoretical approach to find the optimal number of k for a given data set.
  - Too small k – we may not get accurate results
  - Too large k – It may over fit the data
- Solution:
  - Simple Method: Sqrt(N/2), where N is the number of feature vectors
    - If N=32, then k = sqrt(32/2) = 4
    - It fails if the N is too large, k becomes too large
  - **Statistical Methods:** Compare the results of multiple runs with different k classes and choose the best one according to a given criterion **((Elbow, BIC, Schwarz Criterion, etc.).**

HOW TO CHOOSE K – ELBOW METHOD

- Based on Homogeneity and Heterogeneity of the data within the cluster
  - Homogeneity increases as k increases
  - Heterogeneity decreases as K increase
- Use F-Test to measure the homogeneity or heterogeneity



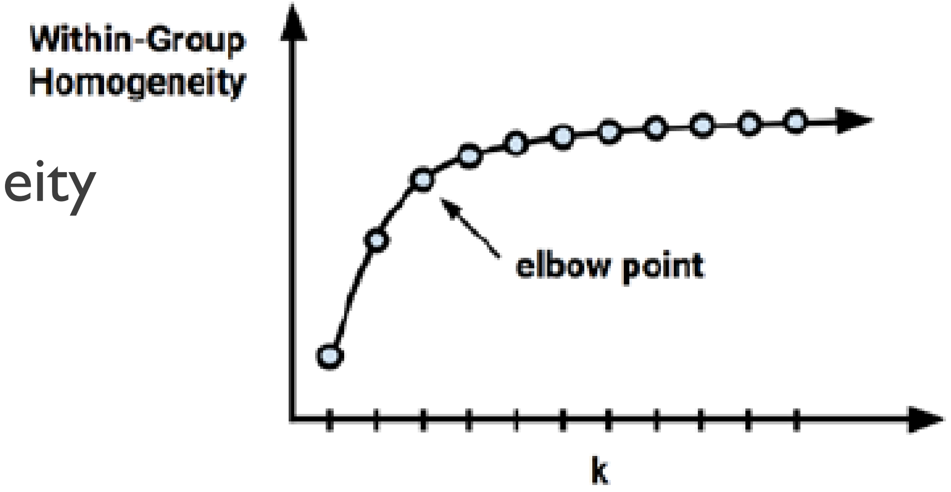
$$F = \frac{\text{between-group variability}}{\text{within-group variability}}.$$

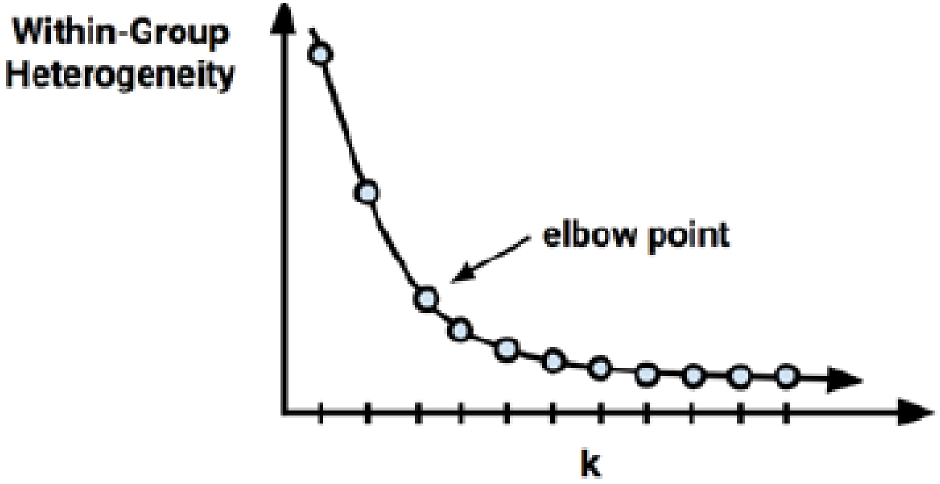
The "explained variance", or "between-group variability" is

$$\sum_{i=1}^{K} n_i(\bar{Y}_{i\cdot} - \bar{Y})^2/(K-1)$$

The "unexplained variance", or "within-group variability" is

$$\sum_{i=1}^{K}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_{i\cdot})^2/(N-K),$$





HOW TO CHOOSE K – ELBOW METHOD

K – Number of clusters

ni – Number of elements in ith cluster

Ybari – Mean distance of ith clsuter

Ybar – Mean distance of all the points of all the clusters

N – Total data elements

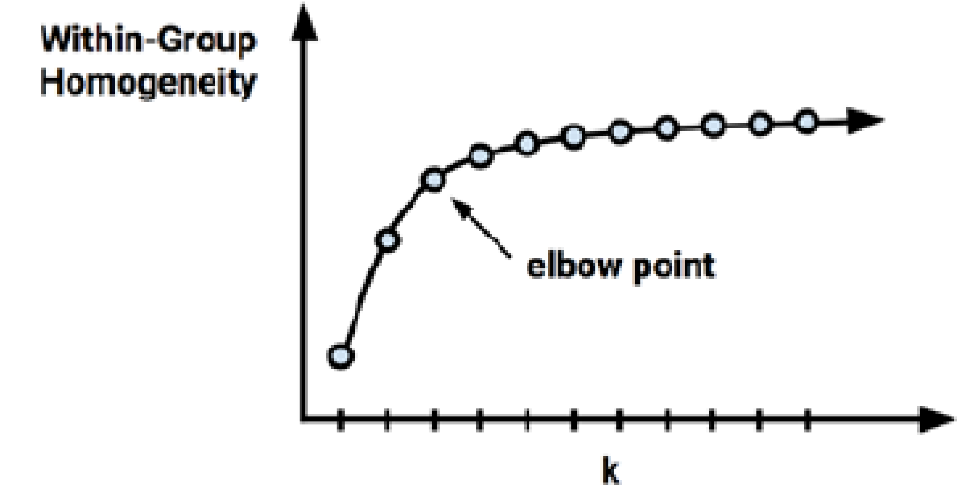
$$F = \frac{\text{between-group variability}}{\text{within-group variability}}.$$

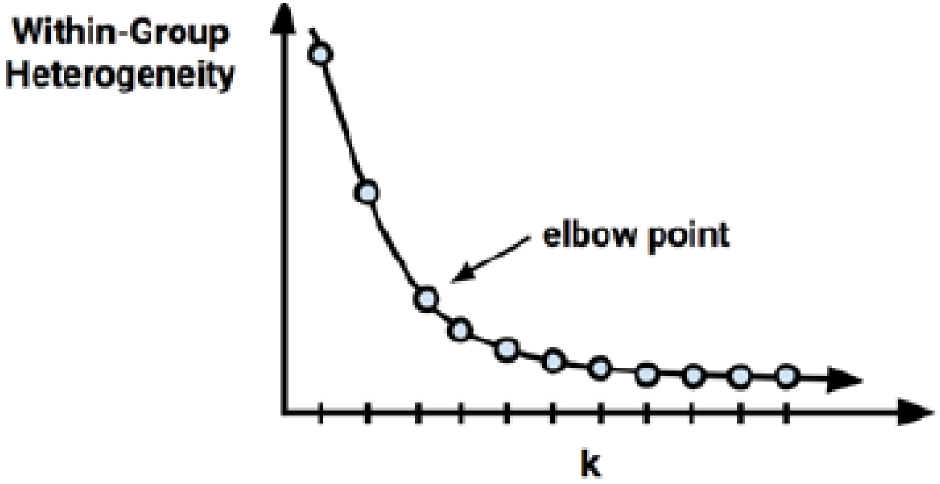
The "explained variance", or "between-group variability" is

$$\sum_{i=1}^{K} n_i(\bar{Y}_{i\cdot} - \bar{Y})^2/(K-1)$$

The "unexplained variance", or "within-group variability" is

$$\sum_{i=1}^{K}\sum_{j=1}^{n_i}(Y_{ij} - \bar{Y}_{i\cdot})^2/(N-K),$$





HOW TO CHOOSE K – BIC METHOD

- Bayesian Information Criterion (BIC, Schwarz, 1978);
- the optimal clustering is the one with the lowest value.
- Based on log-likelihood
- The BIC is defined as:

$$\text{BIC} = -2L_m + m \ln(n)$$

Where, n is the sample size,
L_m is the maximized *log-likelihood* of the model
m is the number of parameters in the model.

K-MEANS CLUSTERING ……

Clustering has mainly three components:

- *An algorithm* — This is the method used to group the items together.
- *A notion of both similarity and dissimilarity* — we rely on the assessment of which item belonged in an existing cluster/group and which should start a new one.
- *A stopping condition* — This might be the point beyond which an item can't be stacked anymore, or when the clusters are already quite dissimilar.

DISTANCE MEASURE - MEASURING SIMILARITY OF ITEMS

The goal of clustering is to group together **“similar”** data – but what does this mean?

It depends on what we want to find or emphasize in the data;

Measuring Distance

- In order to group similar items, we need a way to measure the distance between objects (e.g., records)
- **Note:**

  Distance (D) = inverse of similarity (S) = 1/S

- Often based on the representation of objects as **“feature vectors”**

DISTANCE MEASURE - MEASURING SIMILARITY OF ITEMS…

Distance Measures/Similarity Metrics

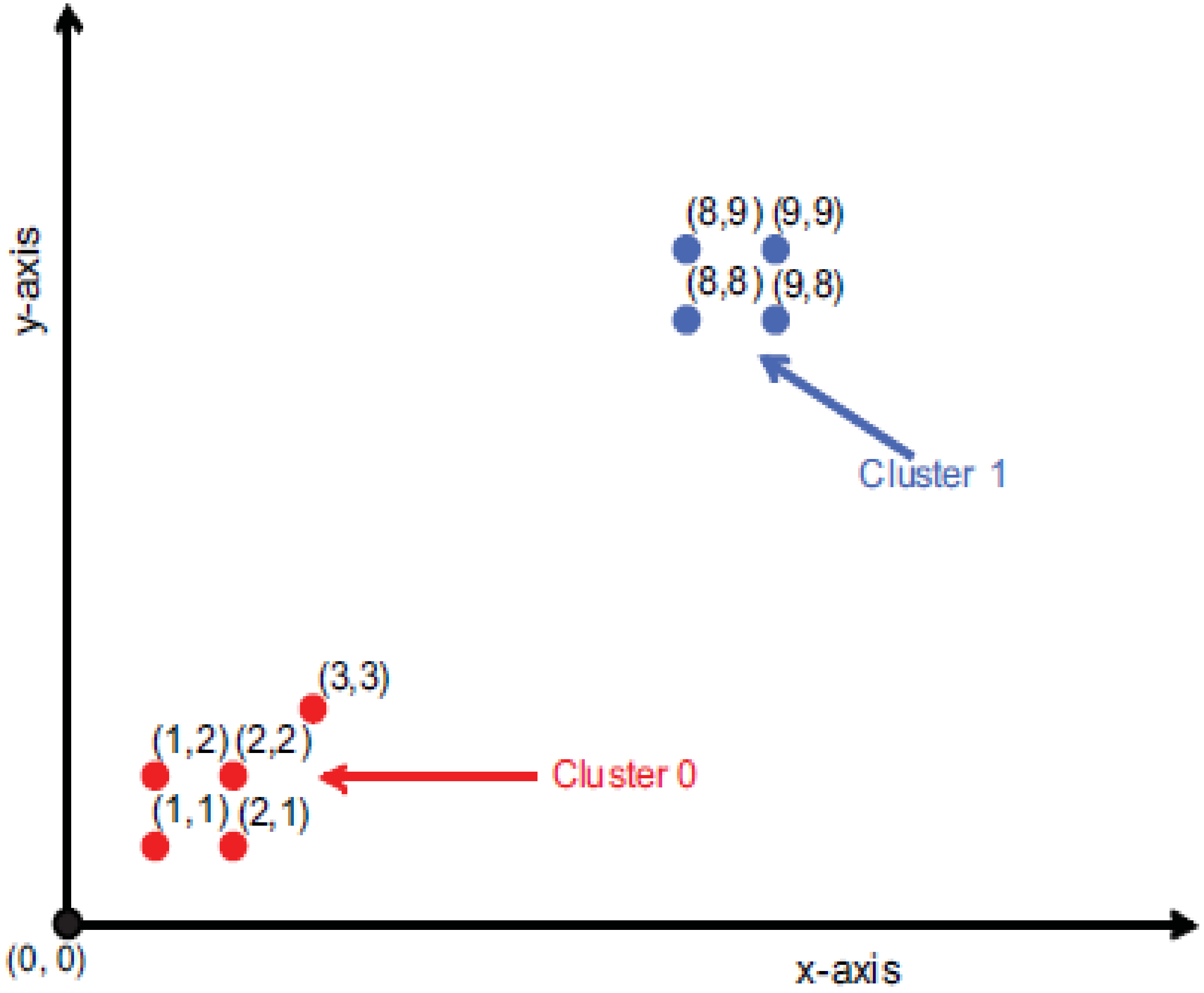
- Euclidean distance measure
- Squared Euclidean distance measure
- Manhattan distance measure
- Cosine distance measure
- Tanimoto distance measure
- Weighted distance measure

DISTANCE MEASURE - EUCLIDEAN DISTANCE

Euclidean distance measure

- Given two points on a plane, the Euclidean distance measure could be calculated by using a ruler/scale to measure the distance between them.
- Mathematically, Euclidean distance between two *n*-dimensional vectors (x1, x2, ... , xn) and (y1, y2, ... , yn) is:

$$dist(X,Y)=\sqrt{(x_1-y_1)^2+\cdots+(x_n-y_n)^2}$$

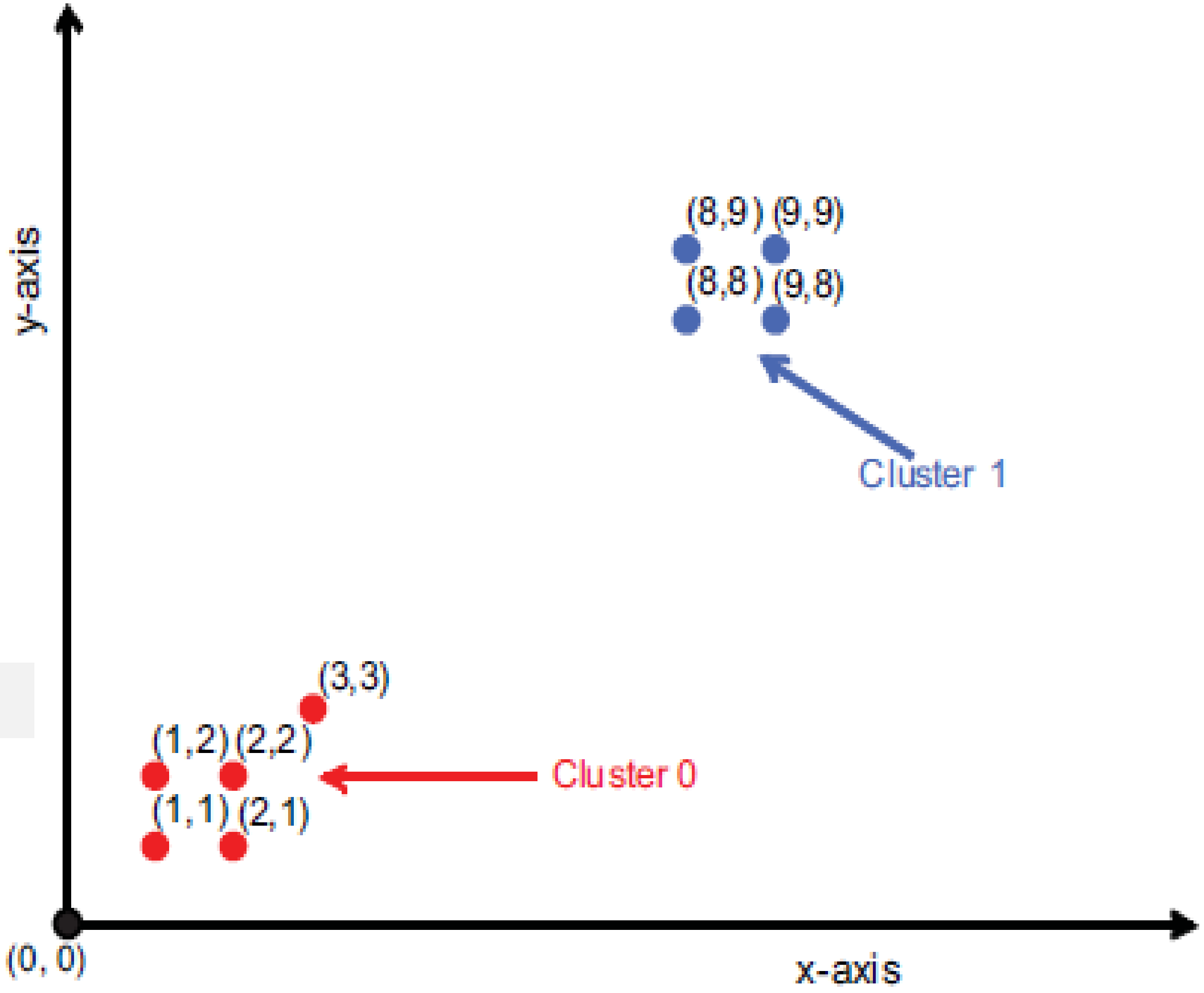


DISTANCE MEASURE - SQUARED EUCLIDEAN DISTANCE

Squared Euclidean distance measure

- This distance measure's value is the square of the value returned by the Euclidean distance measure.
- Mathematically, Squared Euclidean distance between two *n*-dimensional vectors (x1, x2, ... , xn) and (y1, y2, ... , yn) is:

$$\boldsymbol{dist(X,Y) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots\ldots + (x_n - y_n)^2}$$

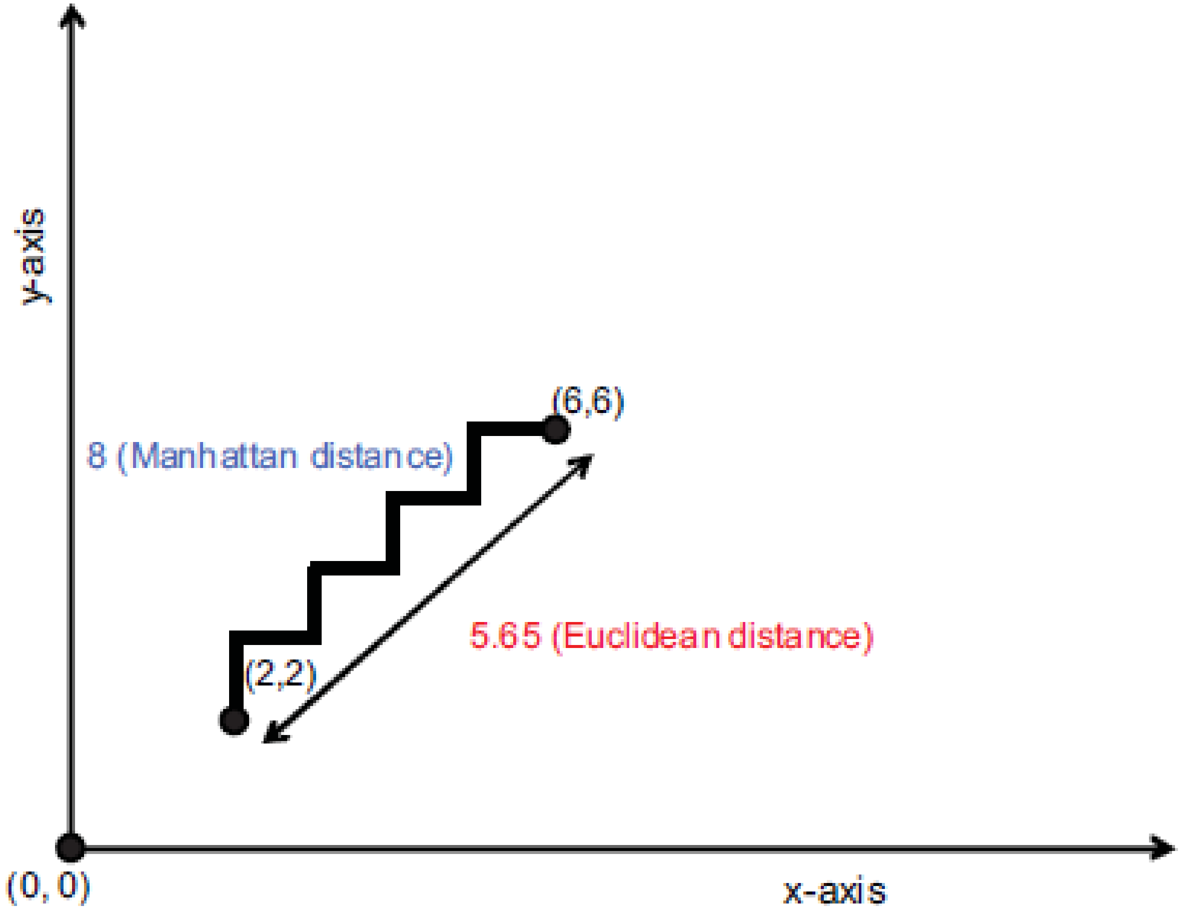


DISTANCE MEASURE - MANHATTAN DISTANCE

Manhattan distance measure

- The distance between any two points is the sum of the absolute differences of their coordinates.
    - This distance measure takes its name from the grid-like layout of streets in Manhattan.
    - As any New Yorker knows, you can't walk from 2nd Avenue and 2nd Street to 6th Avenue and 6th Street by walking straight through buildings.
    - The real distance walked is four blocks up and four blocks over.
- Mathematically, Manhattan distance between two *n*-dimensional vectors (x1, x2, ... , xn) and (y1, y2, ... , yn) is:

$$\boldsymbol{dist}(\boldsymbol{X},\boldsymbol{Y}) = |\, \boldsymbol{x_1} - \boldsymbol{y_1}| + |\, \boldsymbol{x_2} - \boldsymbol{y_2}\, | + \ldots\ldots + |\, \mathbf{x_n} - \mathbf{y_n}\, |$$



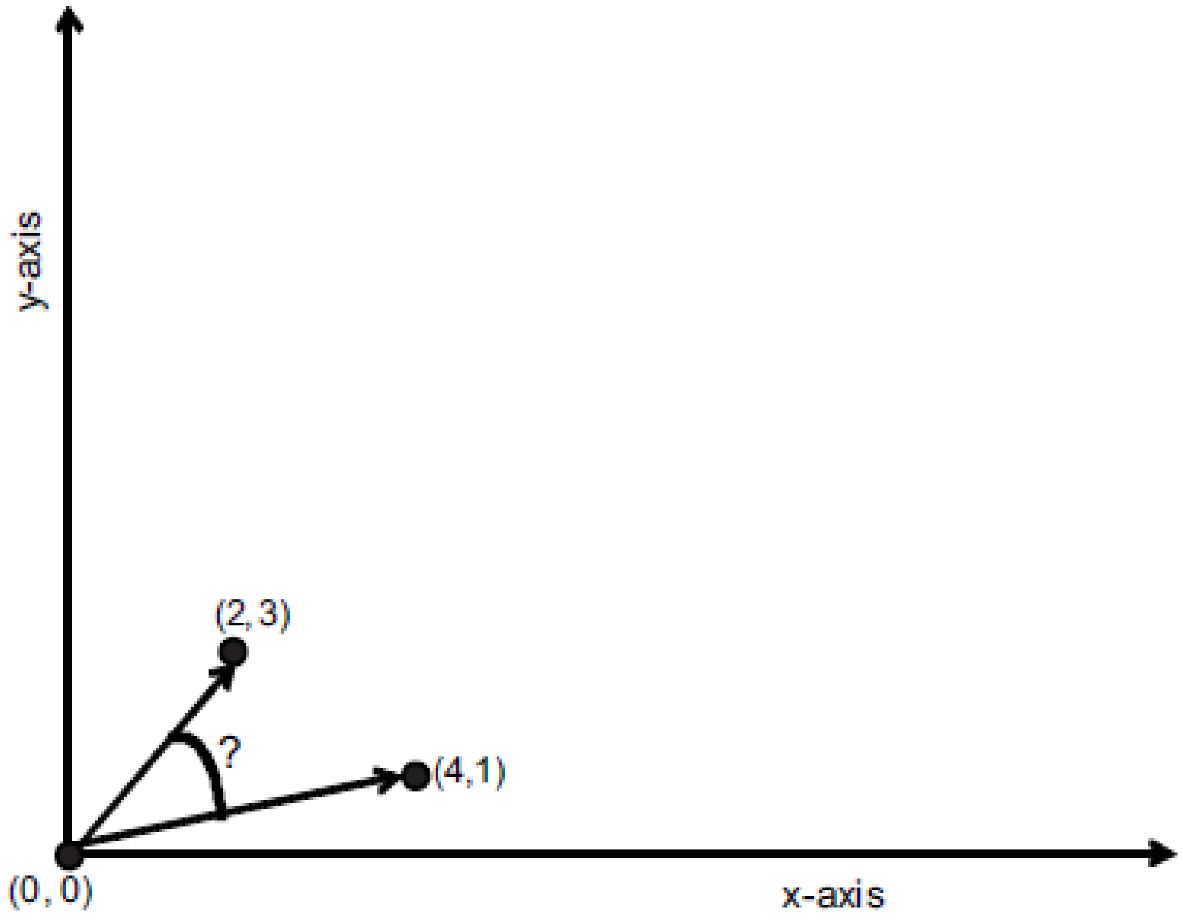
DISTANCE MEASURE - COSINE DISTANCE

Cosine distance measure

- Think points as vectors.
- When the angle is small, then the vectors must be pointing in the same direction.
- Mathematically, Squared Euclidean distance between two *n*-dimensional vectors (x1, x2, ... , xn) and (y1, y2, ... , yn) is:

$$dist(X,Y) = 1 - sim(X,Y)$$

$$sim(X,Y) = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2 \times \sum_i y_i^2}}$$



DISTANCE MEASURE - PROBLEMS WITH COSINE DISTANCE

Problems with cosine distance measure

- The cosine distance measure do not consider the lengths of the vectors, considers only angle.
- It may not work well for some vectors (Feature vectors) which contain valuable information in relative lengths.
- **Example:**
    - consider three vectors: A (1.0, 1.0), B (3.0, 3.0), and C (3.5, 3.5)
    - Cosine Similarity of A & B = [(1.0 x 3.0) + (1.0 x 3.0)] / [sqrt(1^2 + 1^2) x sqrt(3^2 + 3^2)]

      = [6] / [sqrt(2) x sqrt(18)]

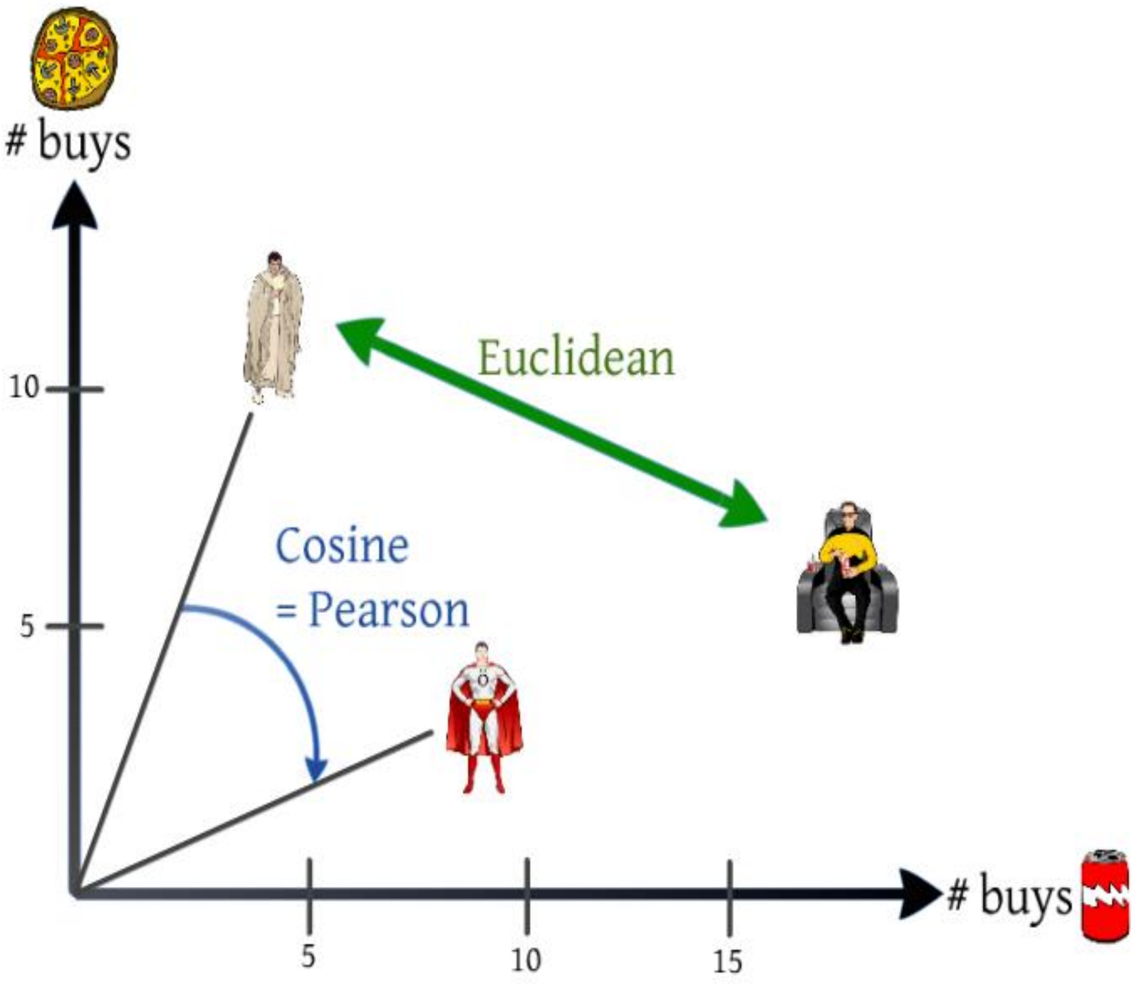
      = 6 / sqrt(36) = 6/6

      = 1
    - Cosine Distance = 1 – Similariy = 1 – 1 = 0

DISTANCE MEASURE - PROBLEMS WITH COSINE DISTANCE…

Problems with cosine distance measure….

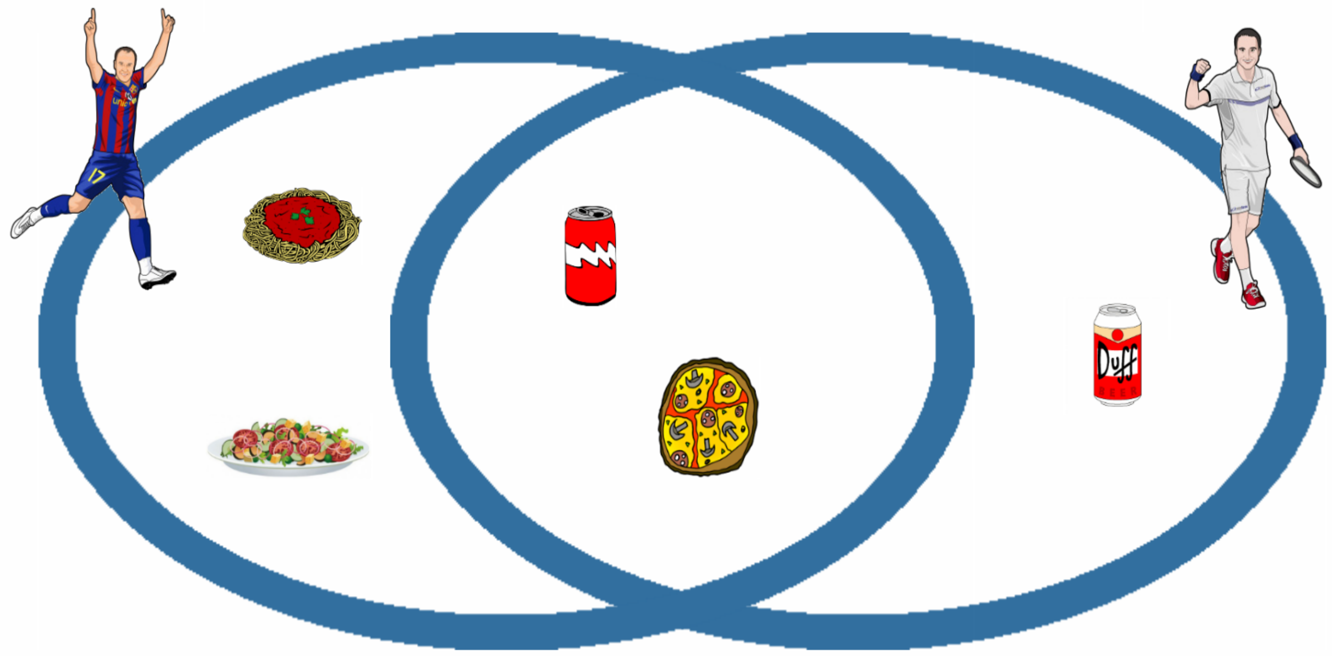
- Cosine distance doesn't capture the fact that B and C are in a sense closer.
- The Euclidean distance measure would reflect this, but it doesn't take account of the angle between the vectors.
- **We need a distance measure that captures both angle and distance.**



DISTANCE MEASURE - TANIMOTO DISTANCE

Tanimoto distance measure

- Captures the information about the angle and the relative distance between the points.
- The Tanimoto similarity between 2 users is computed as the number of products the 2 users have in common divided by the total number of products they bought (respectively clicked or viewed) overall.



$$d = 1 - \frac{(a_1b_1 + a_2b_2 + \ldots + a_nb_n)}{\sqrt{(a_1{}^2 + a_2{}^2 + \ldots + a_n{}^2)} + \sqrt{(b_1{}^2 + b_2{}^2 + \ldots + b_n{}^2)} - (a_1b_1 + a_2b_2 + \ldots + a_nb_n)}$$

$$d = \frac{A\,.B}{|A|+|B|-A.\ B}$$

Image Credits: https://comsysto.files.wordpress.com/2013/03/tanimoto_koeffizient.png

WEIGHTING AND NORMALIZATION

- **Weighting Attributes**
  - in some cases we want some attributes to count more than others
  - **associate a weight with each of the attributes** in calculating distance, e.g.,

$$\boxed{dist(X,Y) = \sqrt{w_1(x_1 - y_1)^2 + \cdots + w_n(x_n - y_n)^2}}$$

- **Nominal (categorical) Attributes**
  - can use simple matching: distance=1 if values match, 0 otherwise
  - or convert each nominal attribute to a set of binary attribute, then use the usual distance measure
  - if all attributes are nominal, we can normalize by dividing the number of matches by the total number of attributes

- **Normalization:**
  - want values to fall between 0 an 1:
  - other variations possible

$$\boxed{x'_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}}$$

DISTANCE OR SIMILARITY MEASURES - EXAMPLE

Example

- max distance for salary: 100000 -19000 = 79000
- max distance for age: 52-27 = 25

$$x'_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}$$

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

ID	Gender	Age	Salary
1	1	0.00	0.00
2	0	0.96	0.56
3	0	1.00	1.00
4	1	0.24	0.44
5	0	0.72	0.32

- Gender is categorized (F – 1, M – 0), Age and Salary are normalized.
- dist(ID2, ID3) = SQRT(0 + $(0.04)^2$ + $(0.44)^2$) = 0.44
- dist(ID2, ID4) = SQRT(1 + $(0.72)^2$ + $(0.12)^2$) = 1.24

DOMAIN SPECIFIC DISTANCE FUNCTIONS

For some data sets, we may need to use specialized functions

- we may want a single or a selected group of attributes to be used in the computation of distance - same problem as "feature selection"
- may want to use special properties of one or more attribute in the data

Example: Zip Codes

$dist_{zip}(A, B) = 0$, if zip codes are identical
$dist_{zip}(A, B) = 0.1$, if first 3 digits are identical
$dist_{zip}(A, B) = 0.5$, if first digits are identical
$dist_{zip}(A, B) = 1$, if first digits are different

Natural distance functions may exist in the data

Example: Customer Solicitation

$dist_{solicit}(A, B) = 0$, if both A and B responded
$dist_{solicit}(A, B) = 0.1$, both A and B were chosen but did not respond
$dist_{solicit}(A, B) = 0.5$, both A and B were chosen, but only one responded
$dist_{solicit}(A, B) = 1$, one was chosen, but the other was not

DISTANCE (SIMILARITY) MATRIX

Similarity (Distance) Matrix

- based on the distance or similarity measure we can construct a symmetric matrix of distance (or similarity values)
- (*i*, *j*) entry in the matrix is the distance (similarity) between items *i* and *j*

$$\begin{array}{c|cccc|}
 & I_1 & I_2 & \cdots & I_n \\
\hline
I_1 & \bullet & d_{12} & \cdots & d_{1n} \\
I_2 & d_{21} & \bullet & \cdots & d_{2n} \\
\vdots & \vdots & \vdots & \bullet & \vdots \\
I_n & d_{n1} & d_{n2} & \cdots & \bullet \\
\hline
\end{array}$$

$$d_{ij} = \text{similarity (or distance) of } D_i \text{ to } D_j$$

Note that dij = dji (i.e., the matrix is symmetric. So, we only need the lower triangle part of the matrix.)

The diagonal is all 1's (similarity) or all 0's (distance)

K-MEANS CLUSTERING – STRENGTHS & WEAKNESSES

Strengths	Weaknesses
▪ Uses simple principles that can be explained in non-statistical terms ▪ Highly flexible, and can be adapted with simple adjustments to address nearly all of its shortcomings ▪ Performs well enough under many real-world use cases	▪ Not as sophisticated as more modern clustering algorithms ▪ Because it uses an element of random chance, it is not guaranteed to find the optimal set of clusters ▪ Requires a reasonable **guess** as to how many clusters naturally exist in the data ▪ **Not ideal for non-spherical clusters** or **clusters of widely varying density**

K-MEANS CLUSTERING – STRENGTHS

- Strengths:
  - Simple: easy to understand and to implement
  - Efficient: Time complexity: $O(tkn)$,

    where, n is the number of data points,

    k is the number of clusters, and

    t is the number of iterations.
  - Since both k and t are small, *k*-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.
- Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

WEAKNESSES OF K-MEANS

- The result may change if we change the initial centroids.
- The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- The user needs to specify *k*.
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- The algorithm is sensitive to **outliers.**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

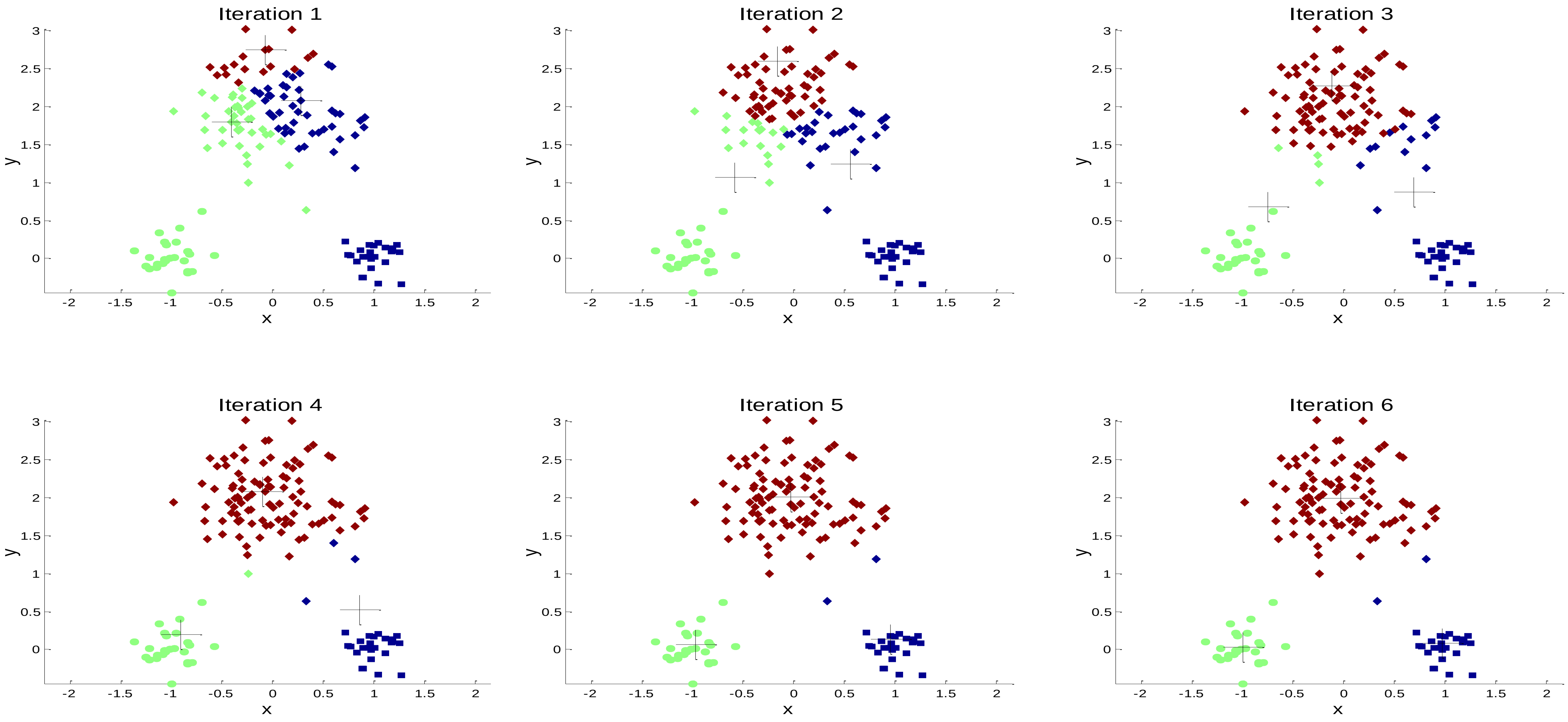
PROBLEMS WITH SELECTING INITIAL POINTS

- 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

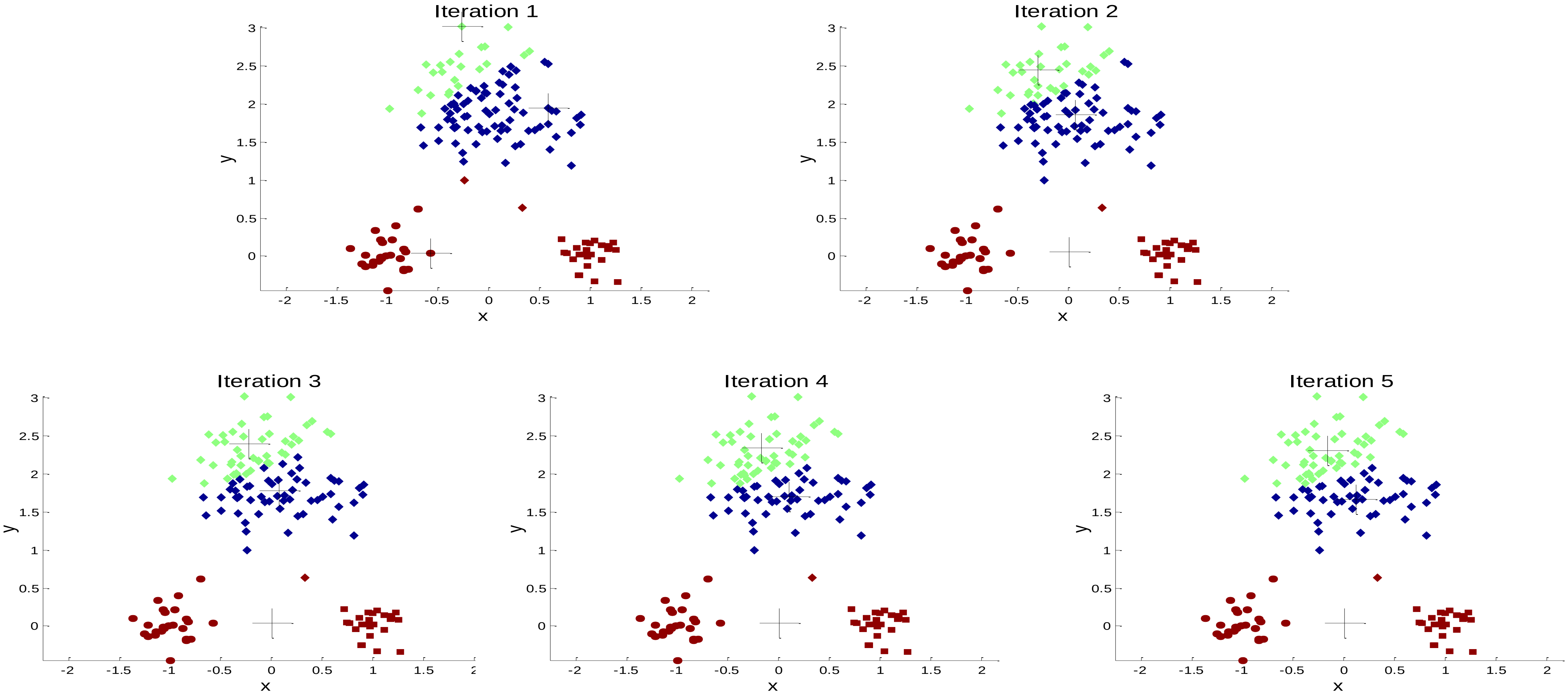
$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

  - 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**

IMPORTANCE OF CHOOSING INITIAL CENTROIDS



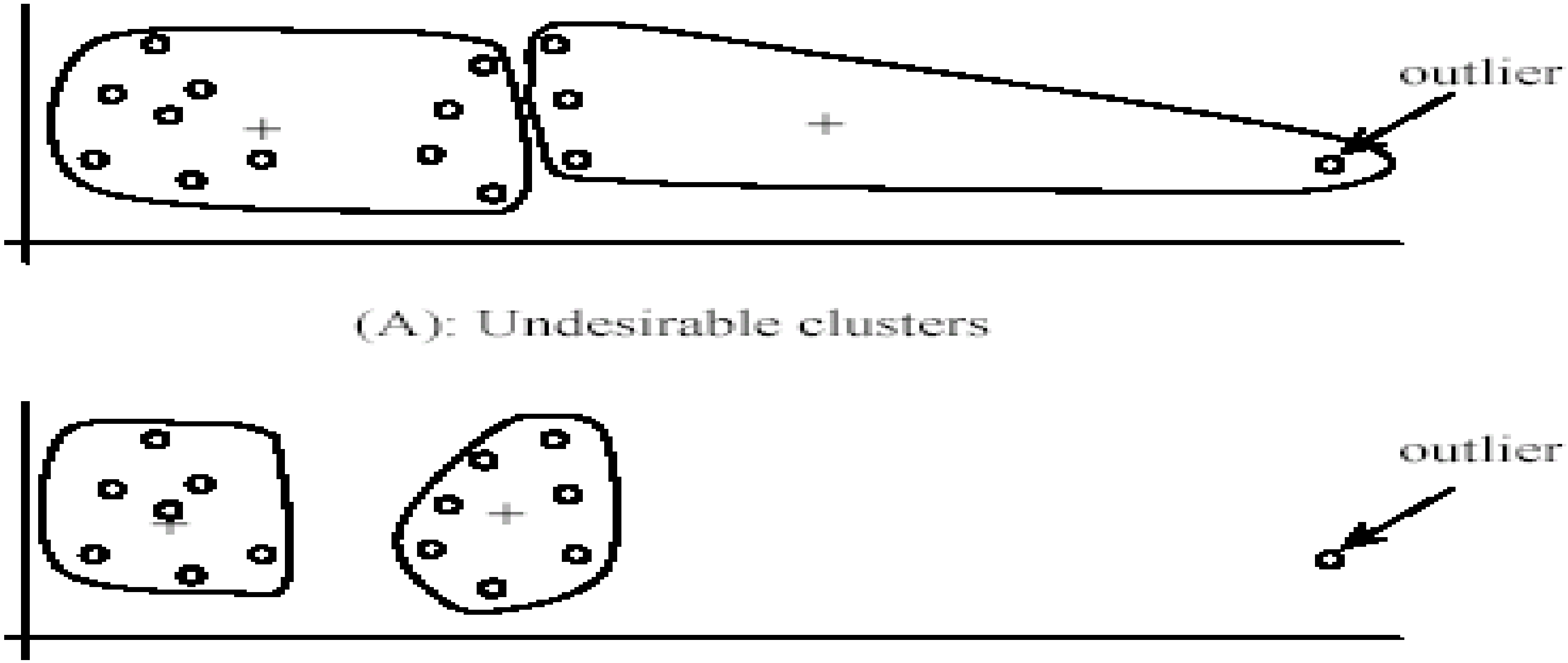
IMPORTANCE OF CHOOSING INITIAL CENTROIDS



SOLUTIONS TO INITIAL CENTROIDS PROBLEM

- Multiple runs
  - Helps, but probability is not on your side always
- 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
- Post-processing
- Bisecting K-means

PROBLEMS WITH OUTLIERS

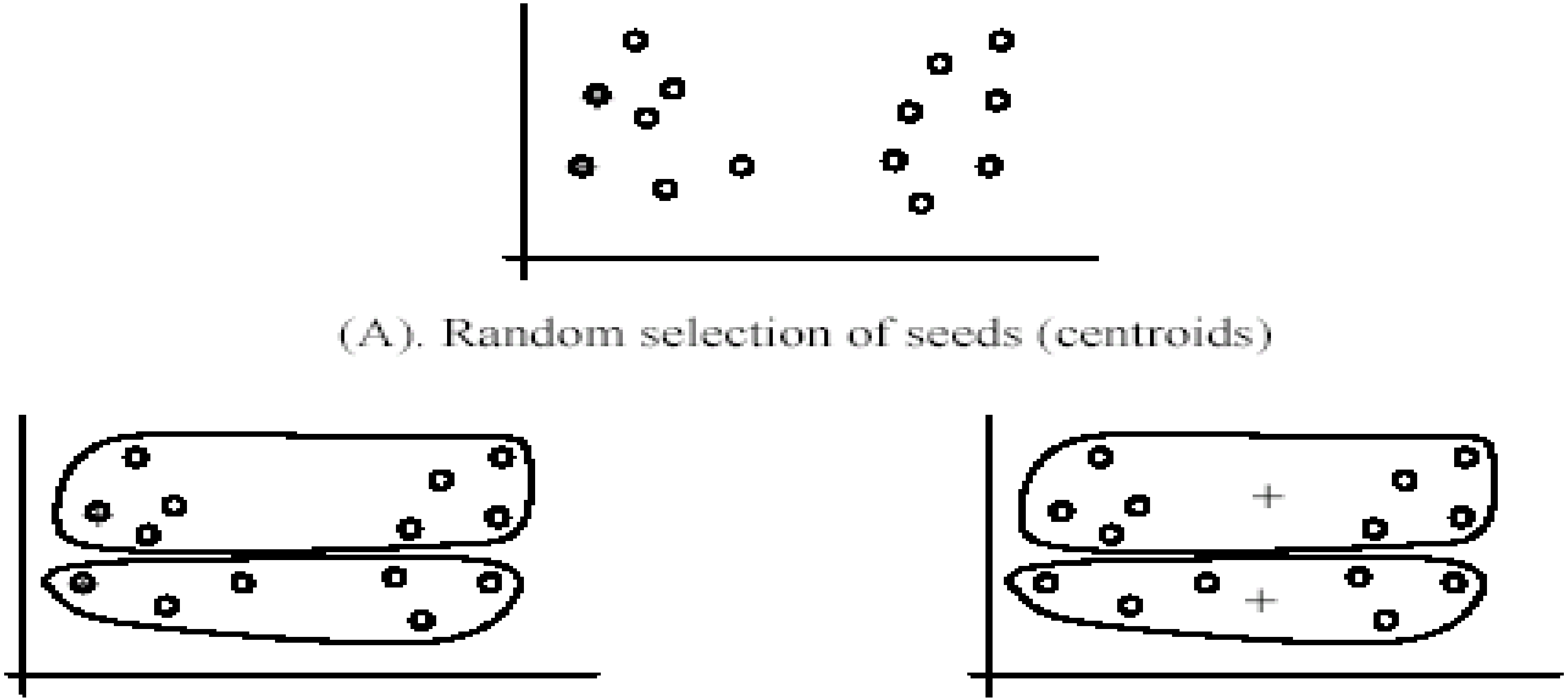


(A): Undesirable clusters

(B): Ideal clusters

HOW TO DEAL WITH OUTLIERS??

- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)

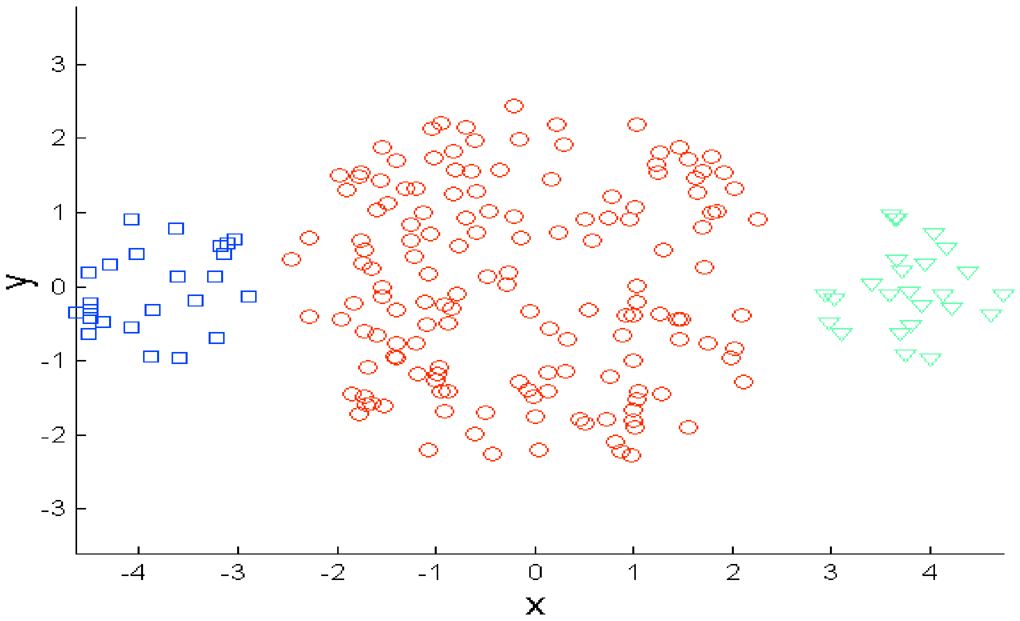
(B). Iteration 1

(C). Iteration 2

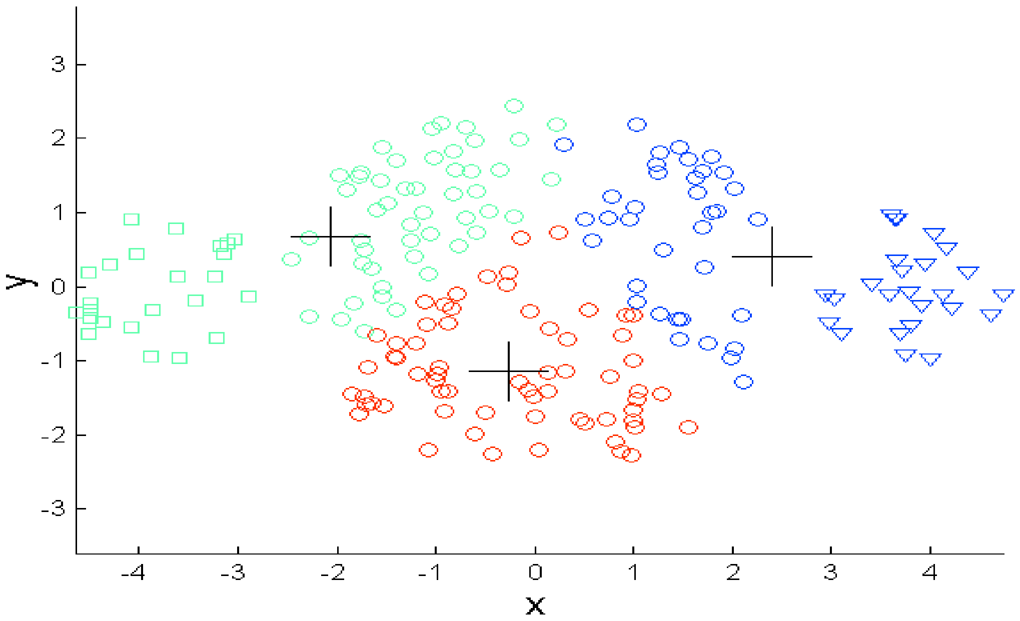
HOW TO DEAL WITH OUTLIERS ??

- One method is to **remove some data points** in the clustering process that are **much further away from the centroids** than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Another method is to **perform random sampling**. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

PROBLEMS WITH DIFFERING SIZES

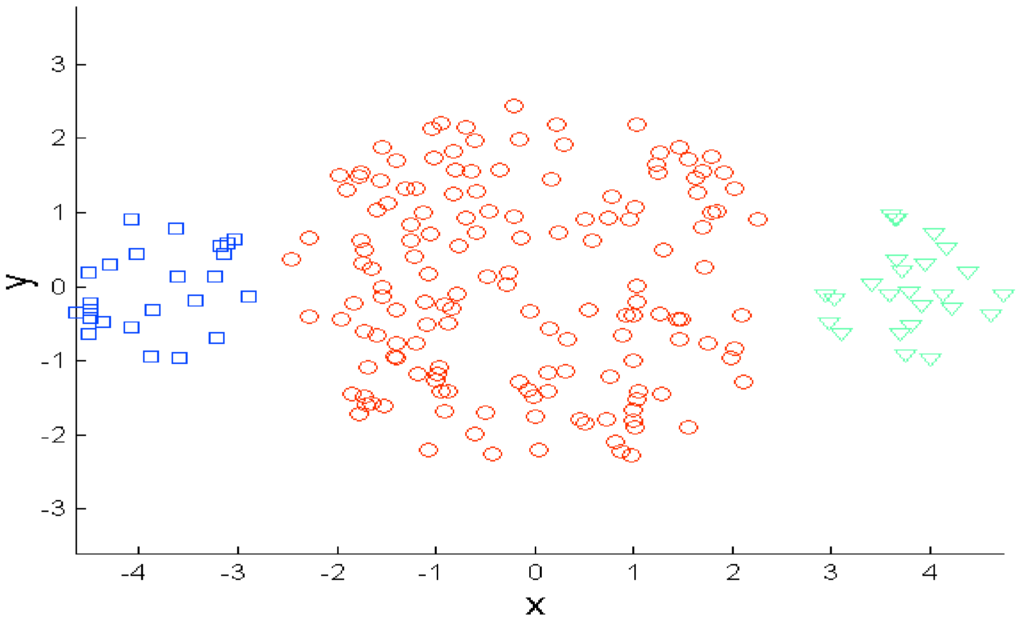


Original Points

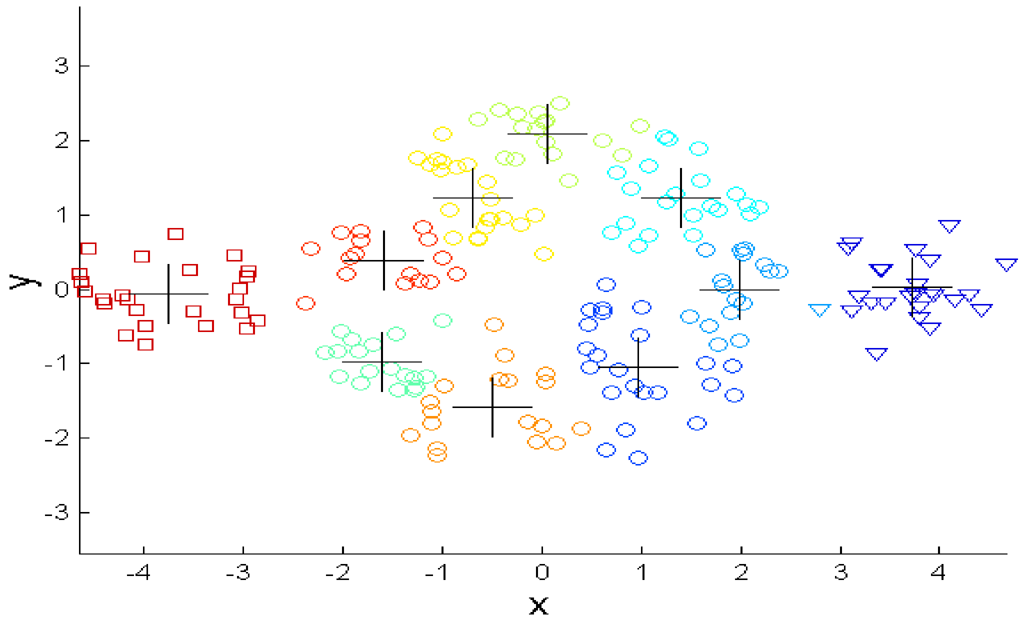


K-means (3 Clusters)

OVERCOMING SIZE PROBLEM



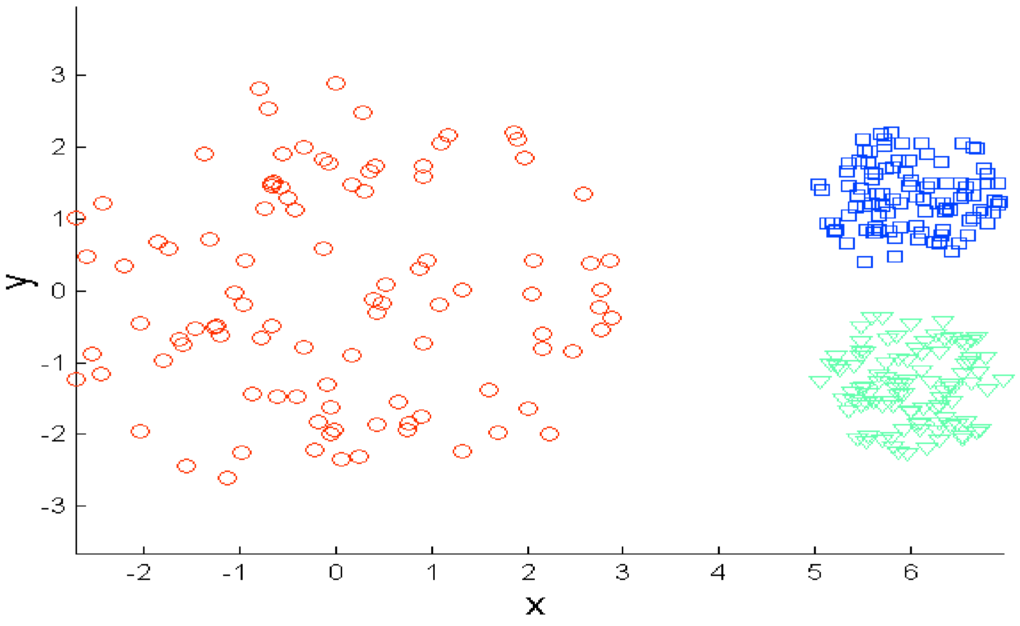
Original Points



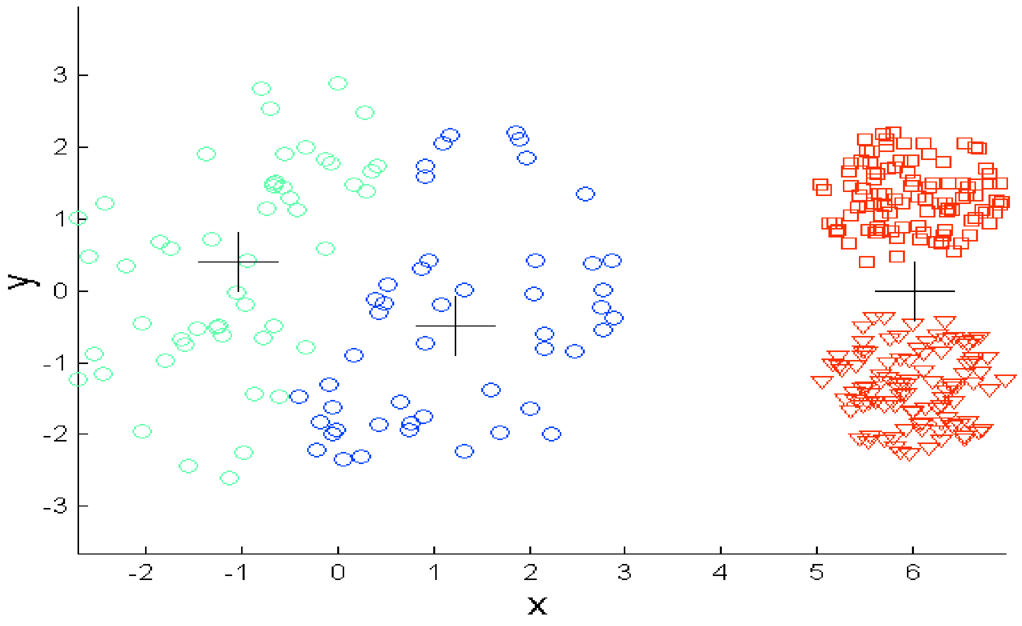
K-means Clusters

- One solution is to use many clusters.
- Find parts of clusters, but need to put together.

PROBLEMS WITH DIFFERING DENSITY

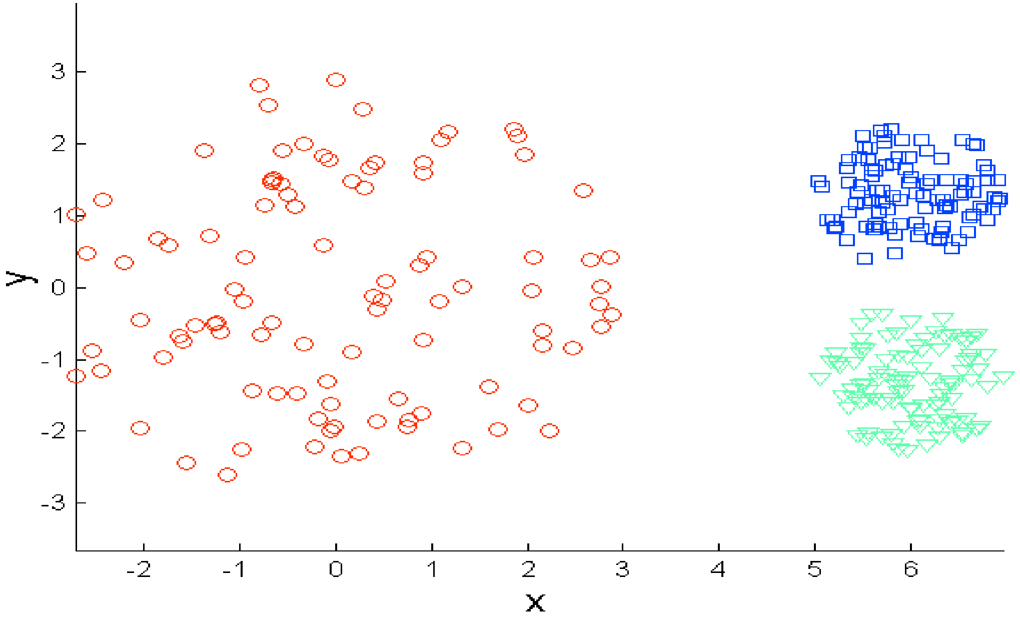


Original Points (3 clusters)

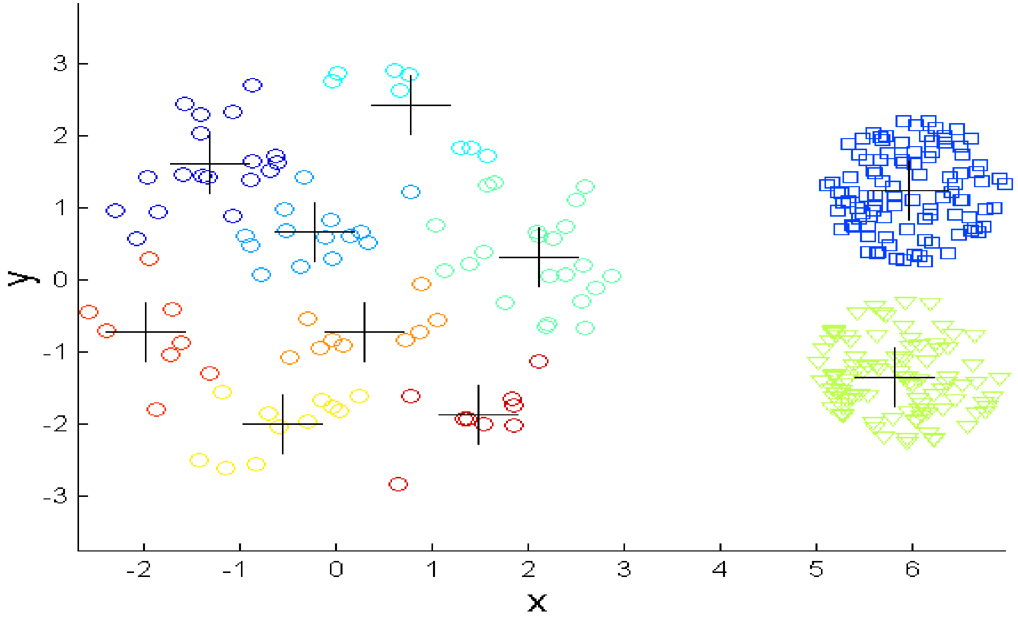


K-means (Got 4 Clusters)

OVERCOMING DENSITY PROBLEM



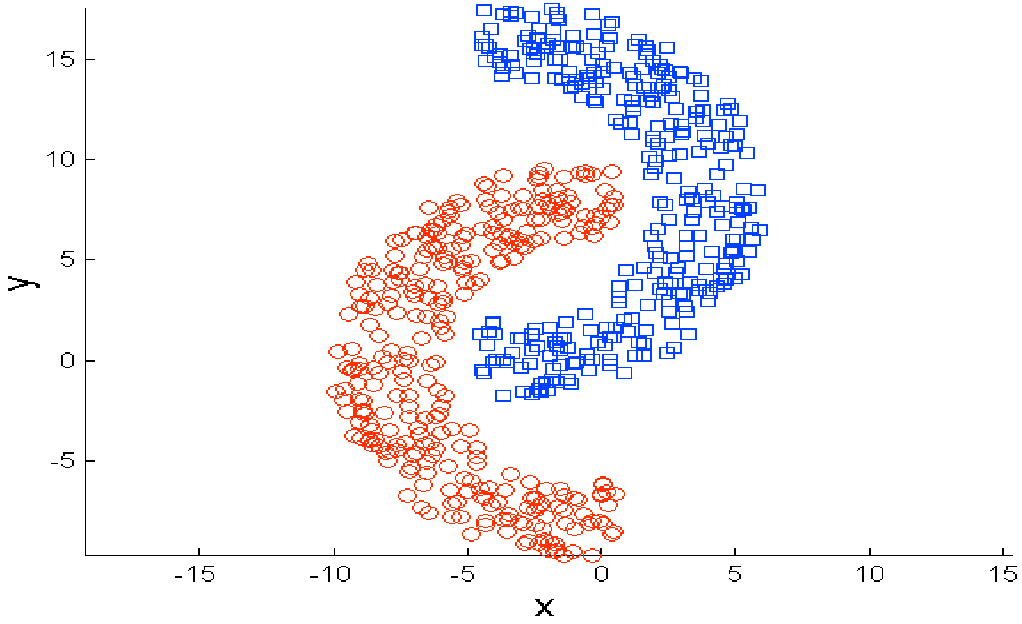
Original Points



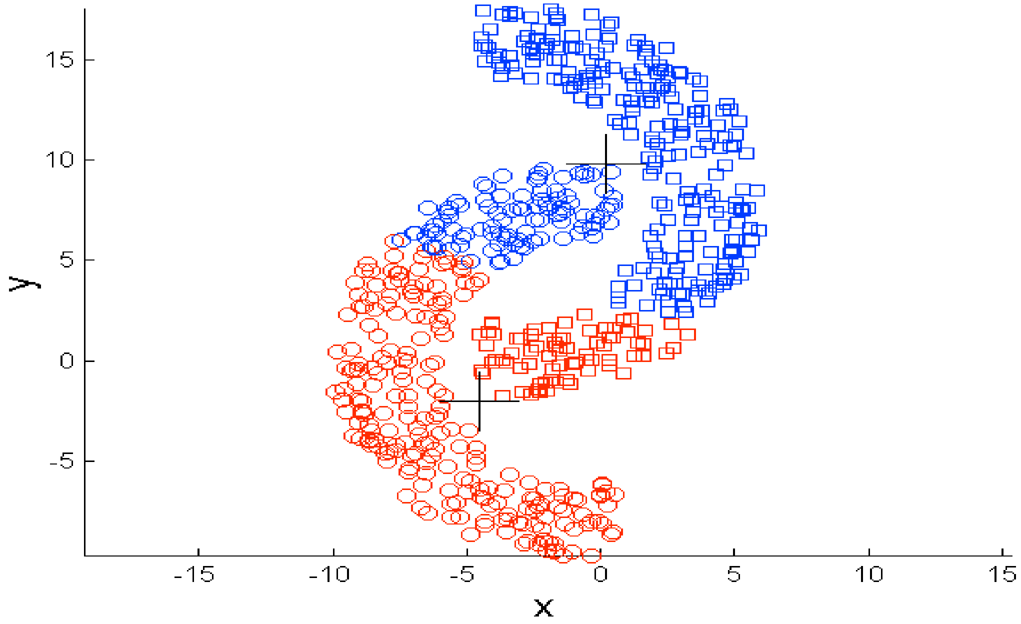
K-means Clusters

- One solution is to use many clusters.
- Find parts of clusters, but need to put together.

PROBLEMS WITH NON-GLOBULAR SHAPES



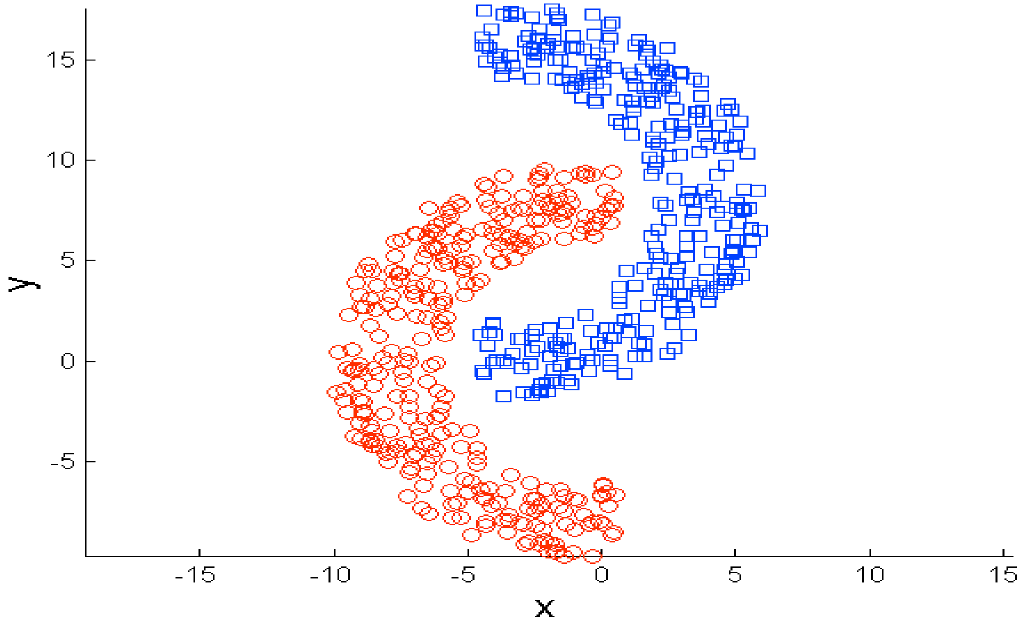
Original Points



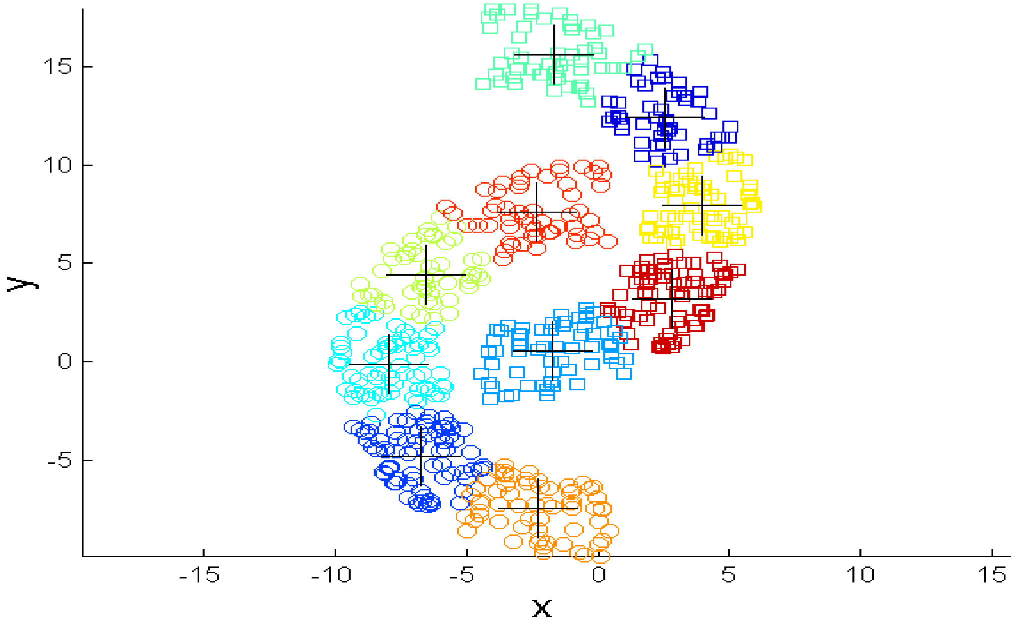
K-means (2 Clusters)

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).

OVERCOMING NON-GLOBULAR SHAPES PROBLEM



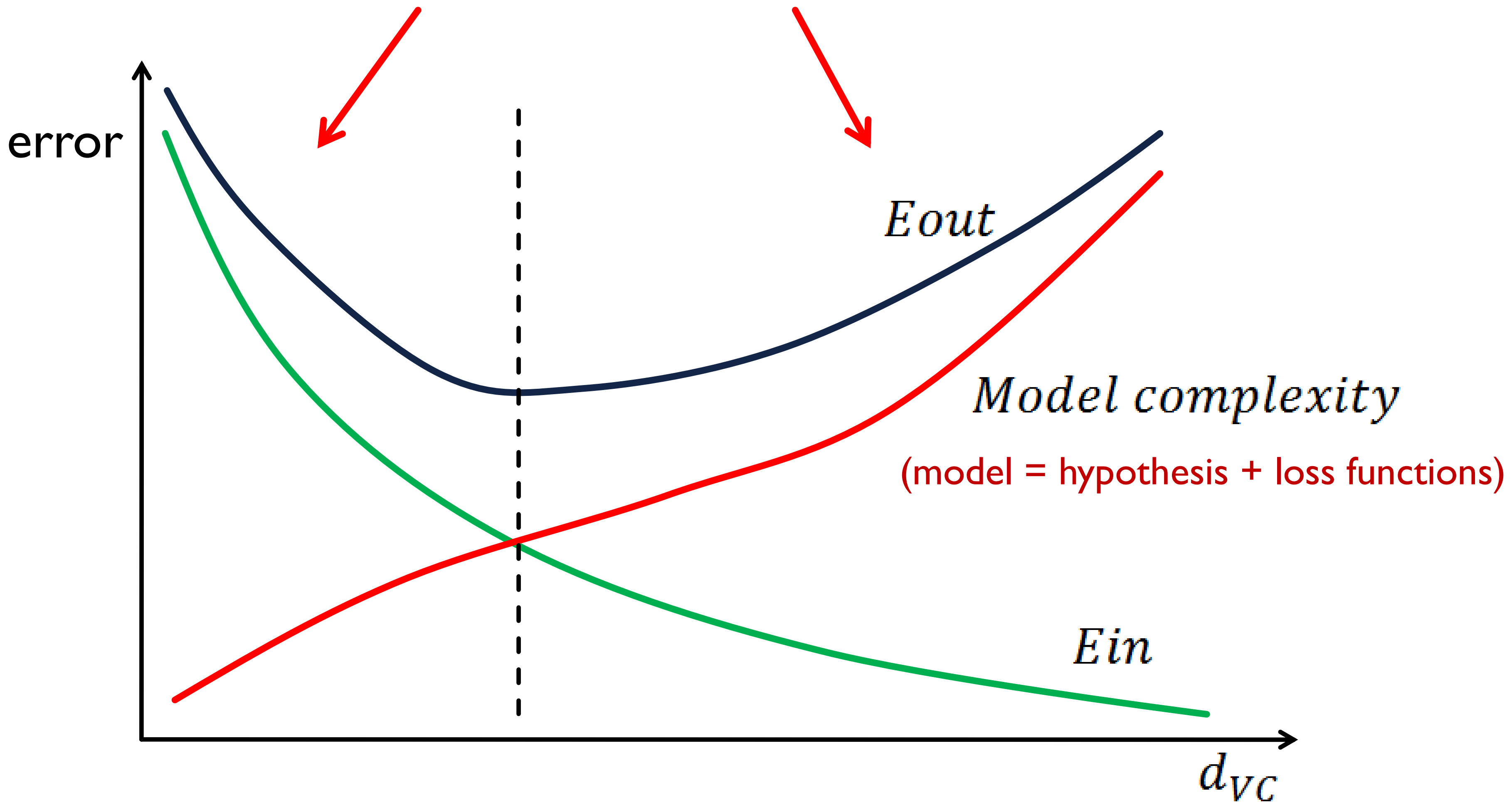
Original Points



K-means Clusters

- One solution is to use many clusters.
- Find parts of clusters, but need to put together.

K-MEANS CLUSTERING – UNDER FITTING .VS. OVERFITTING



DATA STANDARDIZATION

- In the Euclidean space, standardization of attributes is recommended so that all attributes can have equal impact on the computation of distances.
- Consider the following pair of data points
  - $\mathbf{x}_i$: (0.1, 20) and $\mathbf{x}_j$: (0.9, 720).

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(0.9-0.1)^2 + (720-20)^2} = 700.000457,$$

- The distance is almost completely dominated by (720-20) = 700.
- Standardize attributes: to force the attributes to have a common value range

INTERVAL - SCALED ATTRIBUTES

- Their values are real numbers following a linear scale.
  - The difference in Age between 10 and 20 is the same as that between 40 and 50.
  - The key idea is that intervals keep the same importance through out the scale
- Two main approaches to standardize interval scaled attributes, **range** and **z-score**.
- *f* is an attribute

$$range(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)},$$

INTERVAL-SCALED ATTRIBUTES (CONT …)

- Z-score: Transforms the attribute values so that they have a mean of zero and a **mean absolute deviation** of 1. The mean absolute deviation of attribute *f*, denoted by s_f, is computed as follows

$$s_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|\right),$$

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \ldots + x_{nf}\right),$$

Z-score: $$z(x_{if}) = \frac{x_{if} - m_f}{s_f}.$$

RATIO - SCALED ATTRIBUTES

- Numeric attributes, but unlike interval-scaled attributes, their scales are exponential,
- For example, the total amount of microorganisms that evolve in a time t is approximately given by

  Ae^{Bt},

  - where A and B are some positive constants.
- Do log transform:

$$\log(x_{if})$$

  - Then treat it as an interval-scaled attributes.

CASE STUDY: FINDING TEEN MARKET SEGMENTS

INTRODUCTION

Problem Statement

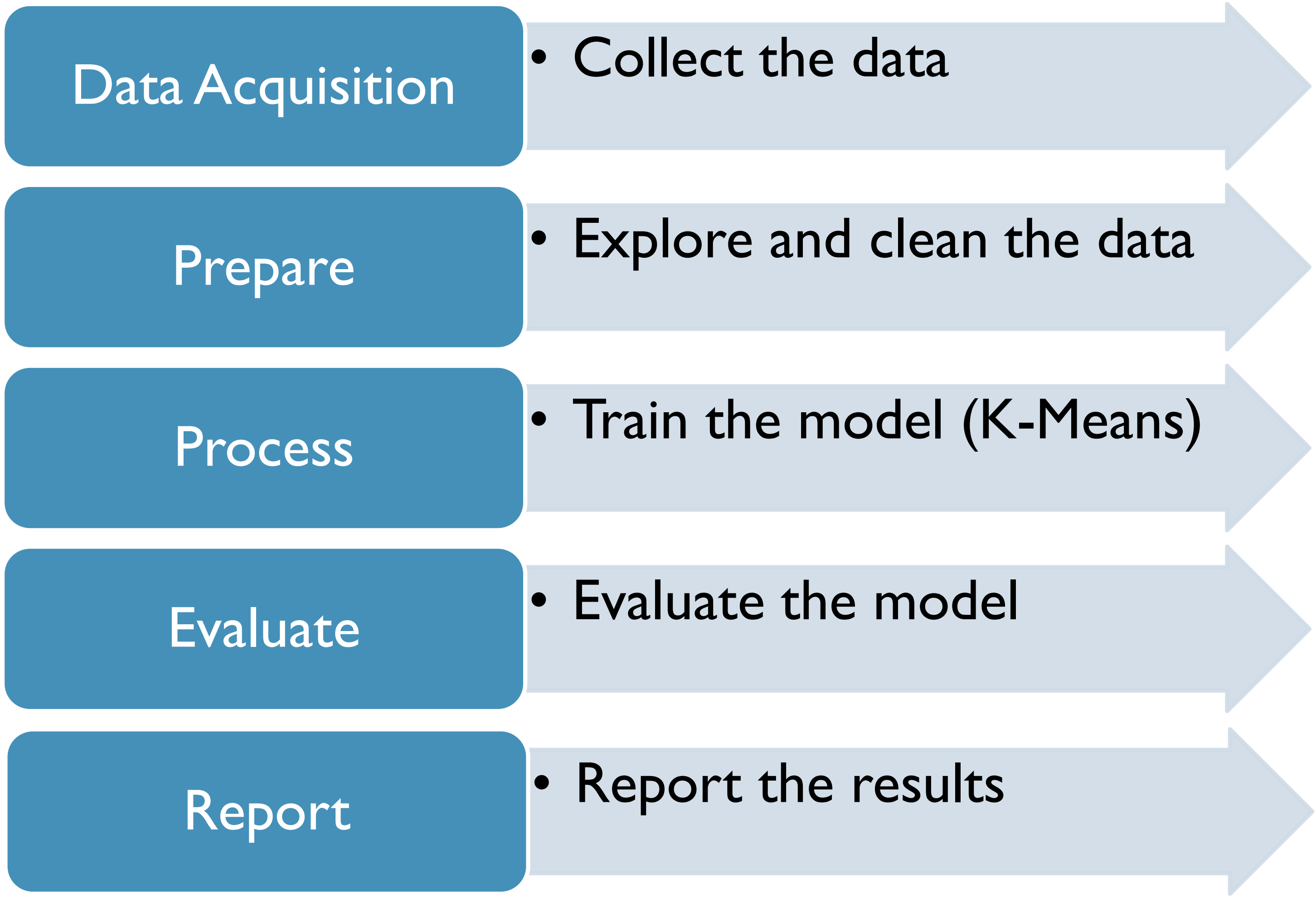
- Identify segments of teenagers who share similar tastes using K-Means clustering

Details

- It helps clients can avoid targeting advertisements to teens with no interest in the product being sold other variations possible
- For instance, sporting apparel is likely to be a difficult sell to teens with no interest in sports.
- **Data**: Text of teenagers' social networking pages.
- we can identify groups that share common interests such as sports, religion, or music.
- These clusters helps us in advertising

STEPS

Regardless of whether the learner is a human or machine, the basic learning process is similar. It can be divided into the following components:



Data Acquisition
- Collect the data

Prepare
- Explore and clean the data

Process
- Train the model (K-Means)

Evaluate
- Evaluate the model

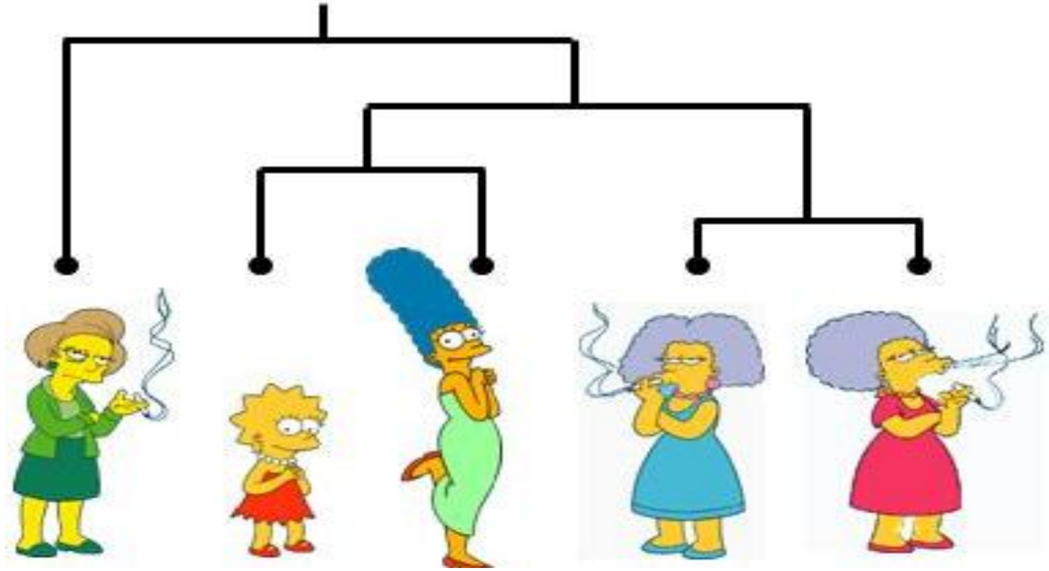
Report
- Report the results

DATA COLLECTION

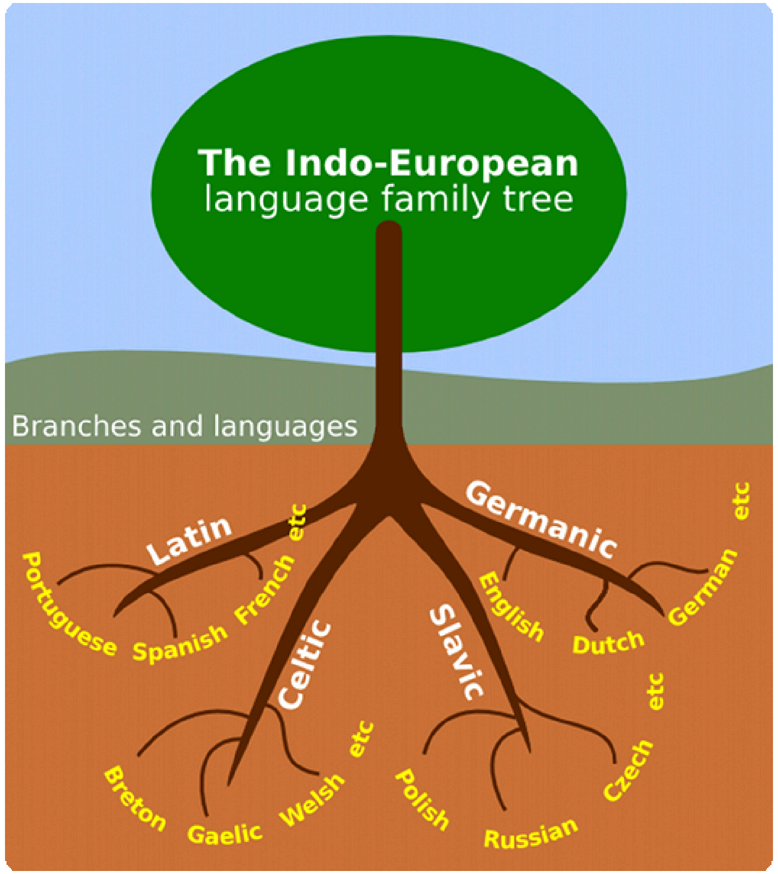
- 30,000 U.S. high school students (snsdata.csv)
- The data was sampled evenly across four high school graduation years (2006 through 2009) representing four classes:
  - the senior
  - junior
  - sophomore
  - freshman
- profiles were downloaded, and each teen's:
  - gender,
  - age,
  - number of SNS friends was recorded.
- Convert the SNS pages into words using text mining tools.

DATA COLLECTION…

- From the top 500 words appearing across all the pages, 36 words were chosen to represent five categories of interests: namely,
  - extracurricular activities,
  - fashion,
  - religion,
  - romance, and
  - antisocial behaviour.
- The 36 words include terms such as football, sexy, kissed, bible, shopping, death, and drugs.
- The final dataset indicates, for each person, how many times each word appeared in the person's SNS profile.



HIERARCHICAL CLUSTERING



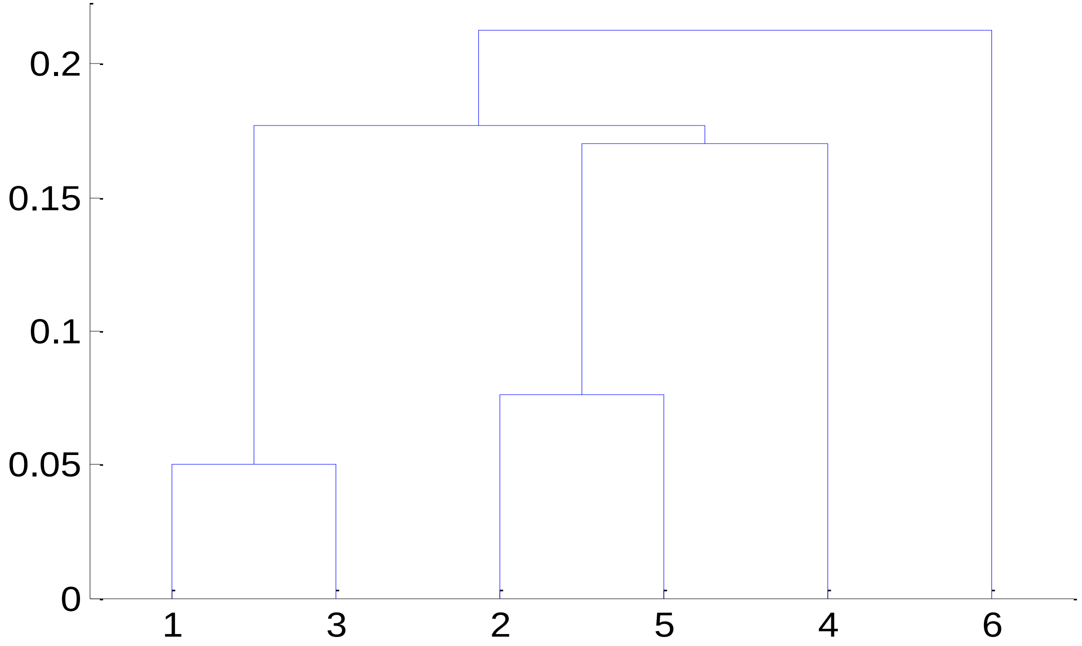
INTRODUCTION

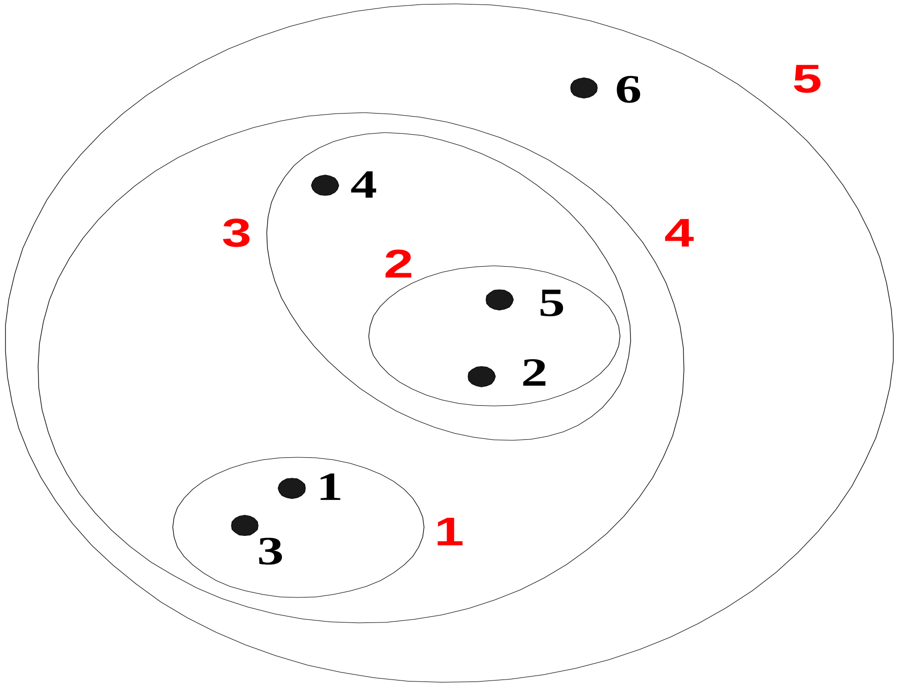
K-means clustering requires us to specify the number of clusters - k, and finding the optimal number of clusters can often be hard.

Hierarchical clustering is an alternative approach which builds a hierarchy from the bottom-up/top-down, and doesn’t require us to specify the number of clusters beforehand.

HIERARCHICAL CLUSTERING

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a **dendrogram**
  - A tree like diagram that records the sequences of merges or splits





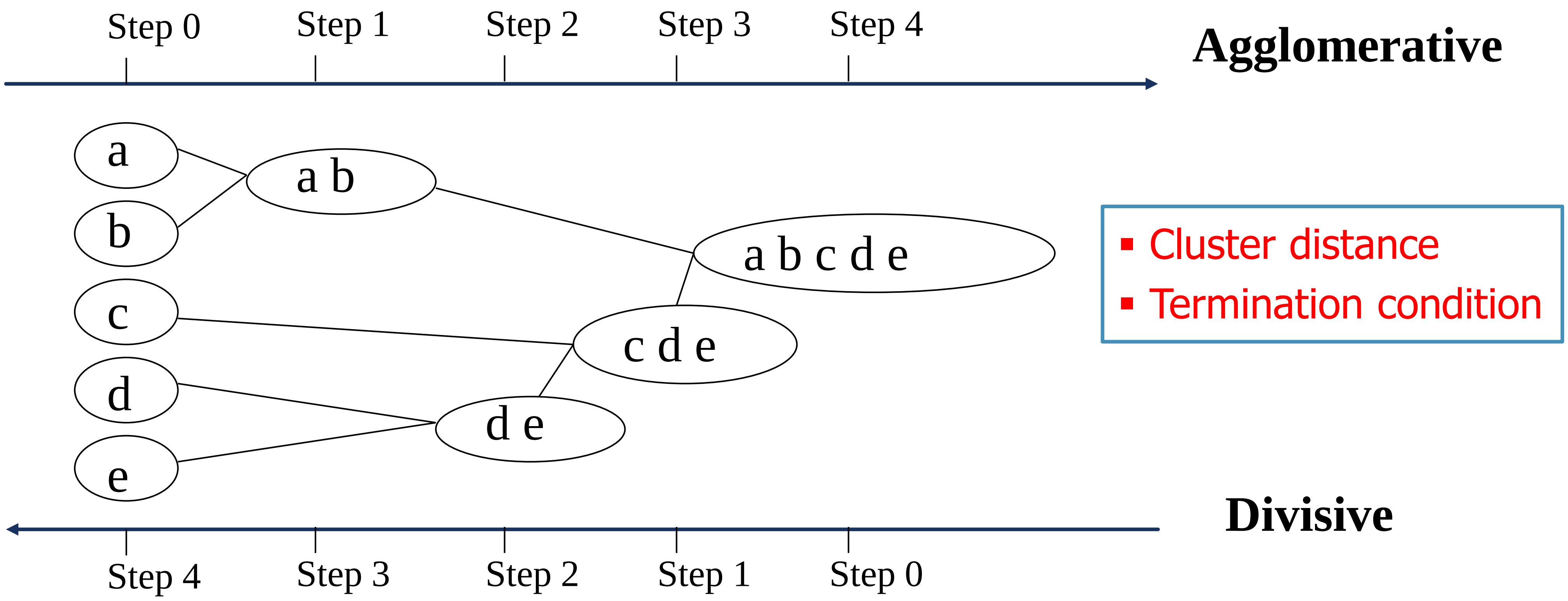
TYPES OF HIERARCHICAL CLUSTERING

- Agglomerative (bottom up) clustering: It **builds** the dendrogram (tree) **from the bottom level,** and
  - merges the most similar (or nearest) pair of clusters
  - stops when all the data points are merged into a single cluster (i.e., the root cluster).
- Divisive (top down) clustering: It **starts** with all data points in one cluster, **the root-top**.
  - Splits the root into a set of child clusters.
  - Each child cluster is recursively divided further
  - stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point

EXAMPLE

Illustrative Example

Agglomerative and divisive clustering on the data set {a, b, c, d, e }



- Cluster distance
- Termination condition

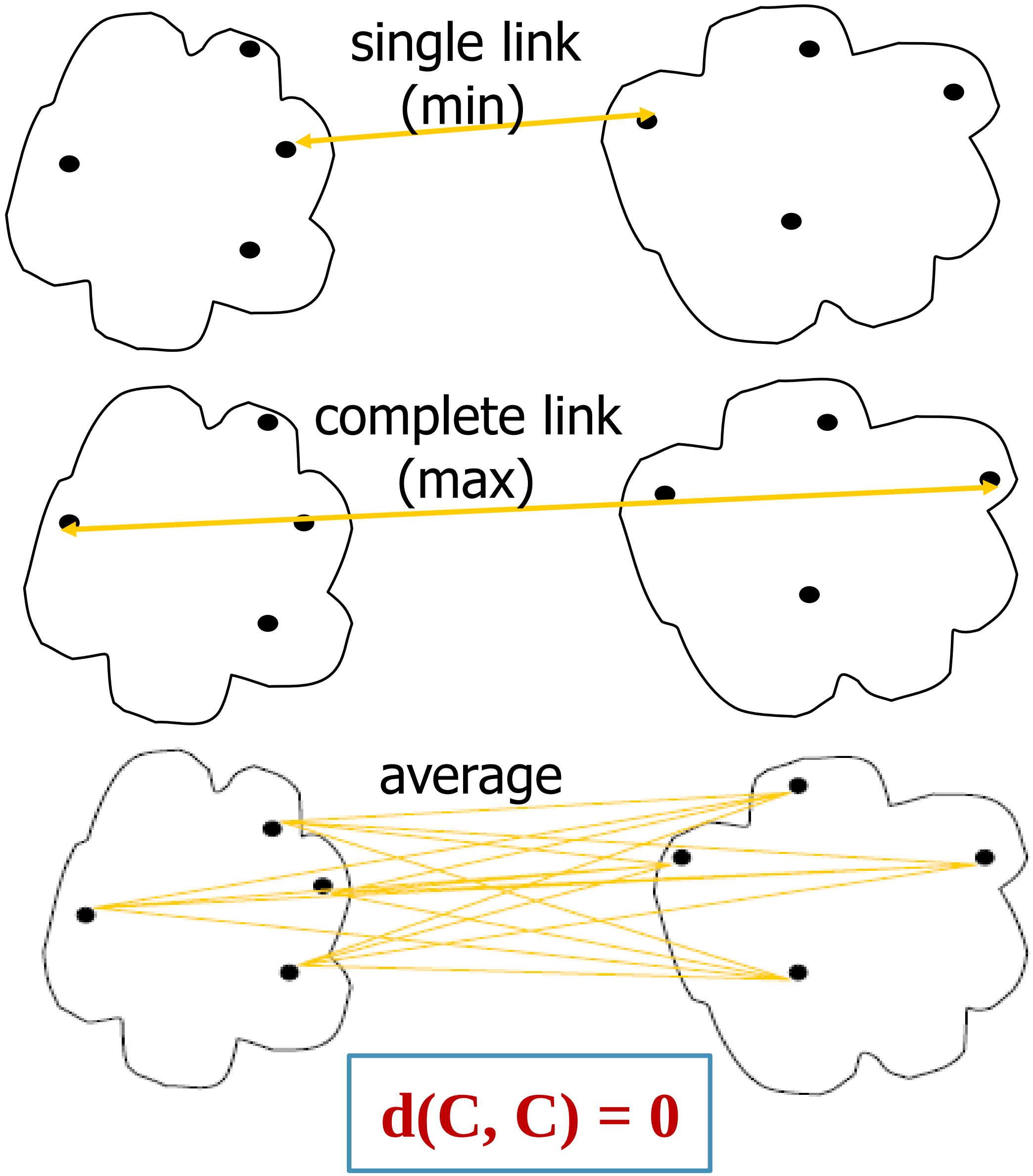
CLUSTER DISTANCE MEASURES

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \min\{d(x_{ip}, x_{jq})\}$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e.,

  $$d(C_i, C_j) = \max\{d(x_{ip}, x_{jq})\}$$

- Average: avg distance between elements in one cluster and elements in the other, i.e.,

  $$d(C_i, C_j) = \text{avg}\{d(x_{ip}, x_{jq})\}$$



$\boxed{d(C, C) = 0}$

CLUSTER DISTANCE MEASURES - EXAMPLES

	a	b	c	d	e
Feature	**1**	**2**	**4**	**5**	**6**

Example: Given a data set of five objects characterized by a single continuous feature, assume that there are two clusters: C1: {a, b} and C2: {c, d, e}. Calculate three cluster distances between C1 and C2.

	a	b	c	d	e
a	0	1	**3**	**4**	**5**
b	1	0	**2**	**3**	**4**
c	**3**	**2**	0	1	2
d	**4**	**3**	1	0	1
e	**5**	**4**	2	1	0

Single link:

$$\begin{aligned}\mathrm{dist}(\mathrm{C}_1,\mathrm{C}_2) &= \min\{d(\mathrm{a,c}), d(\mathrm{a,d}), d(\mathrm{a,e}), d(\mathrm{b,c}), d(\mathrm{b,d}), d(\mathrm{b,e})\}\\ &= \min\{3, 4, 5, 2, 3, 4\} = 2\end{aligned}$$

Complete link:

$$\begin{aligned}\mathrm{dist}(\mathrm{C}_1,\mathrm{C}_2) &= \max\{d(\mathrm{a,c}), d(\mathrm{a,d}), d(\mathrm{a,e}), d(\mathrm{b,c}), d(\mathrm{b,d}), d(\mathrm{b,e})\}\\ &= \max\{3, 4, 5, 2, 3, 4\} = 5\end{aligned}$$

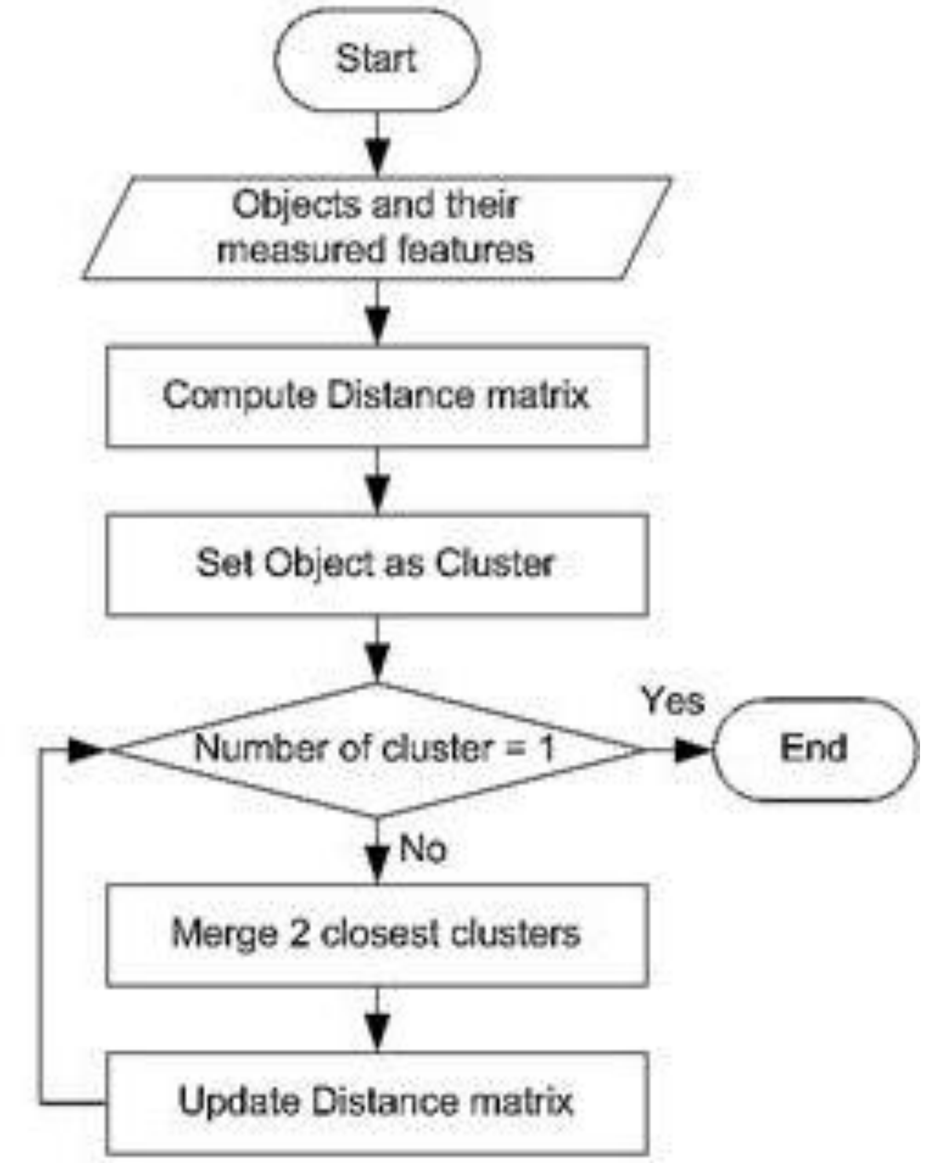
Average link:

$$\begin{aligned}\mathrm{dist}(\mathrm{C}_1,\mathrm{C}_2) &= \frac{d(\mathrm{a,c}) + d(\mathrm{a,d}) + d(\mathrm{a,e}) + d(\mathrm{b,c}) + d(\mathrm{b,d}) + d(\mathrm{b,e})}{6}\\ &= \frac{3+4+5+2+3+4}{6} = \frac{21}{6} = 3.5\end{aligned}$$

AGGLOMERATIVE ALGORITHM

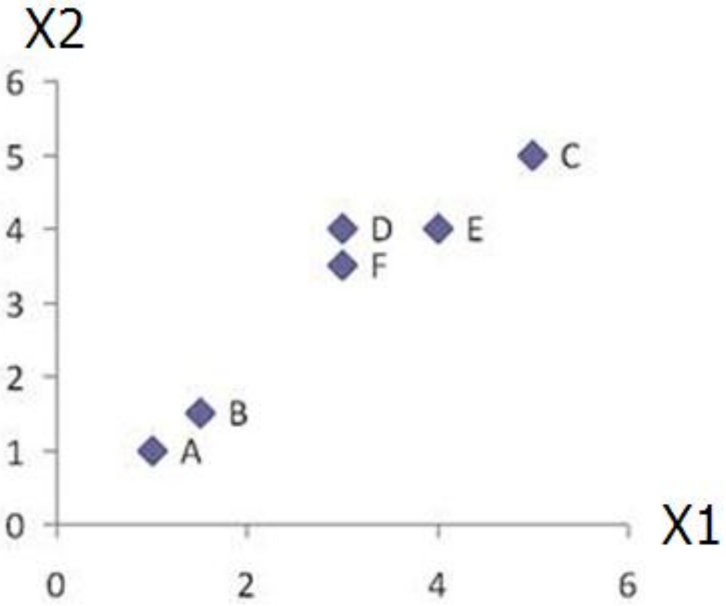
The *Agglomerative* algorithm (Bottom-up) is carried out in three steps:

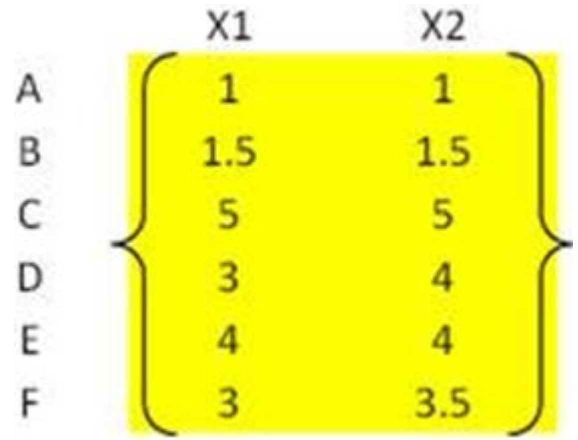
1. Convert all object features into a distance matrix.
2. Set each object as a cluster (thus if we have N objects, we will have N clusters at the beginning).
3. Repeat until number of cluster is one (or known # of clusters)
   - Merge two closest clusters
   - Update "distance matrix"



AGGLOMERATIVE ALGORITHM - EXAMPLE

Clustering analysis with agglomerative algorithm





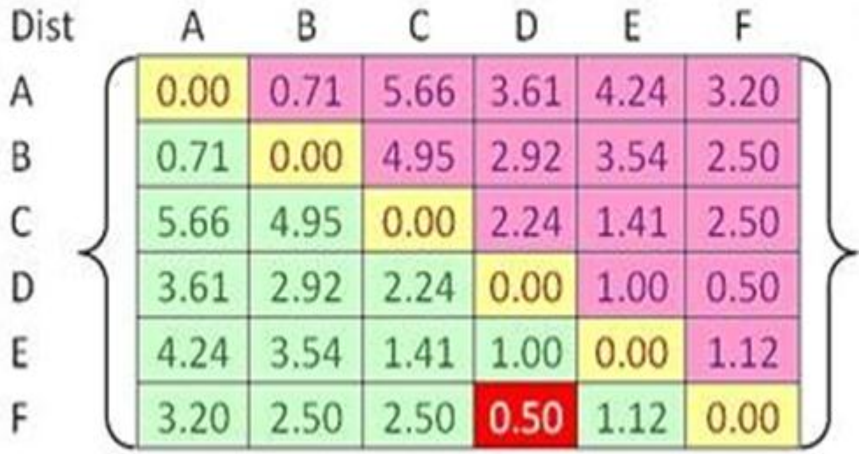
	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

data matrix

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

$$d_{DF} = \left((3-3)^2 + (4-3.5)^2\right)^{\frac{1}{2}} = 0.5$$

Euclidean distance

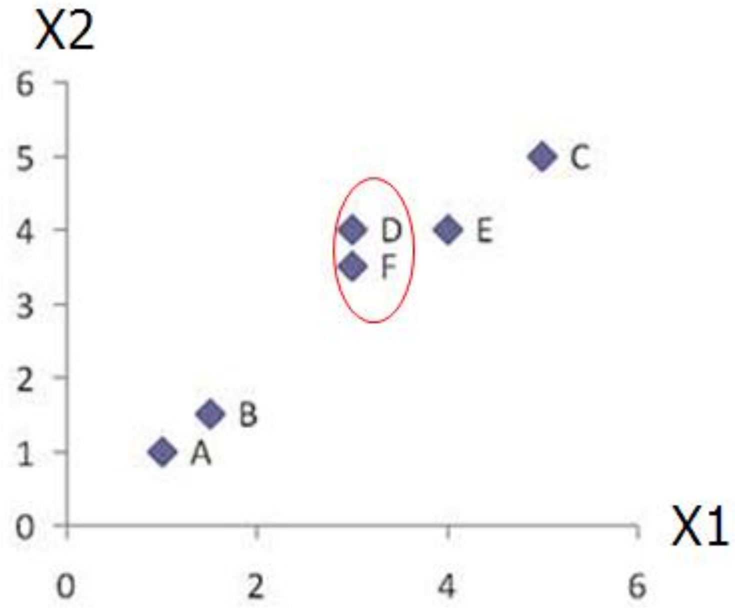


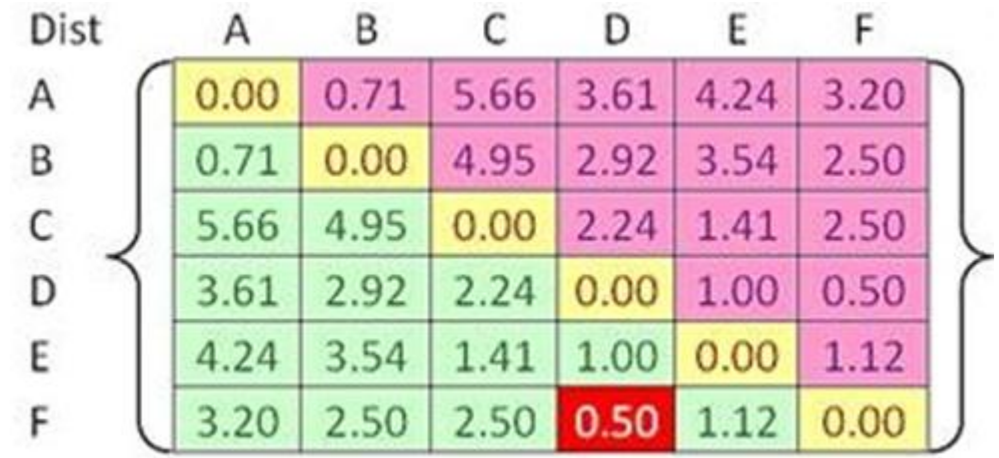
Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

distance matrix

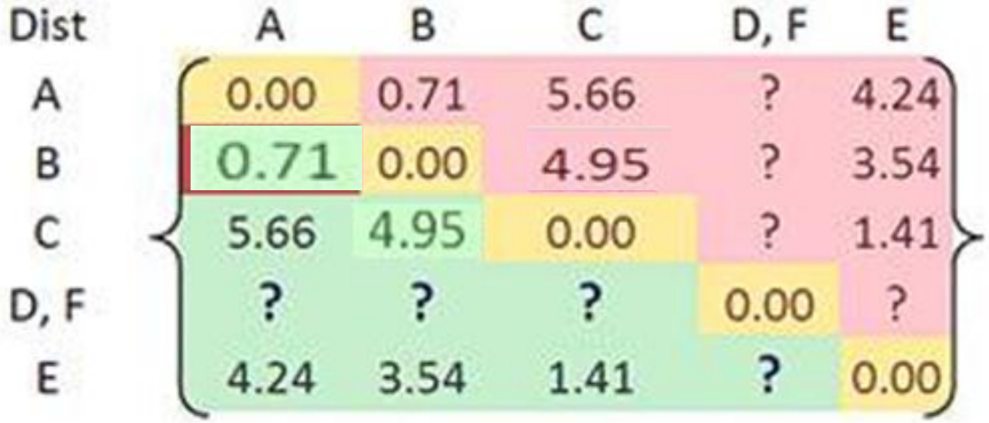
AGGLOMERATIVE ALGORITHM - EXAMPLE

Merge two closest clusters (iteration 1)





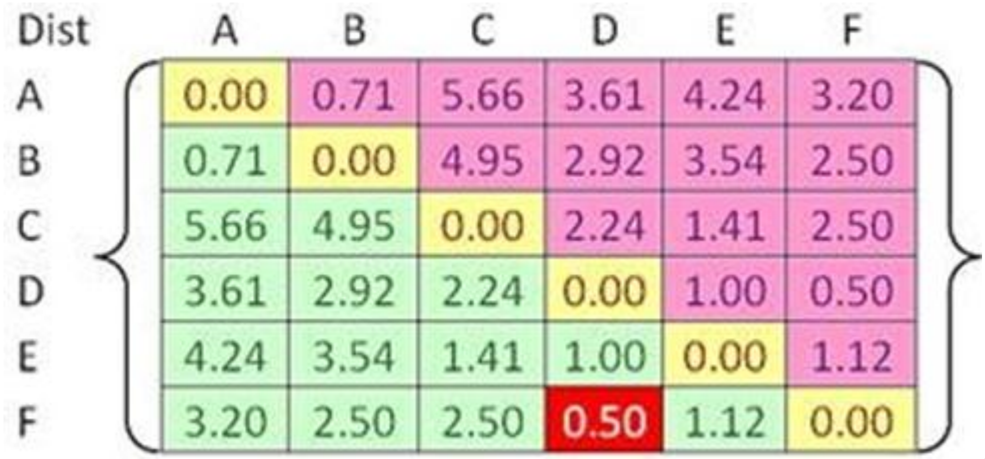
Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00



Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

AGGLOMERATIVE ALGORITHM - EXAMPLE

Update Distance Matrix (iteration 1)



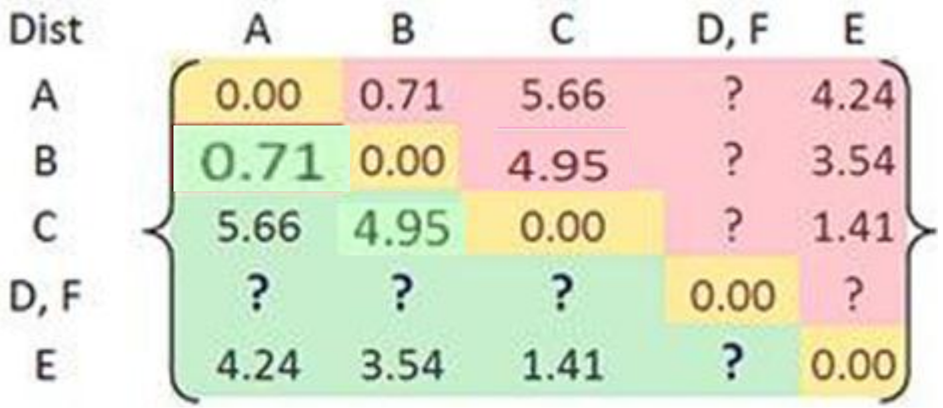
Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

$$d_{(D,F)\mapsto A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

$$d_{(D,F)\mapsto B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

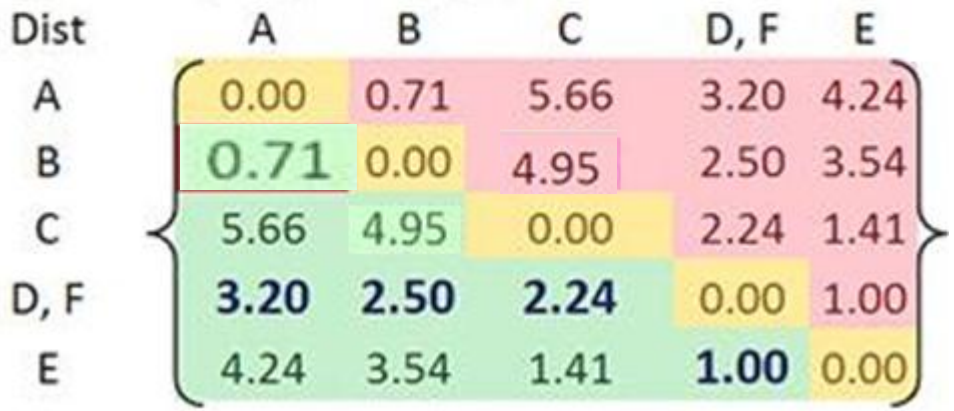
$$d_{(D,F)\mapsto C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$



Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

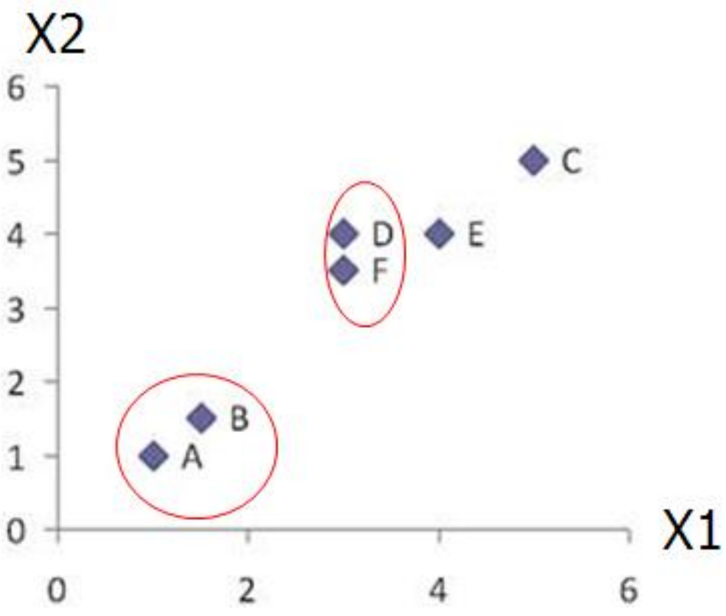
Min Distance (Single Linkage)

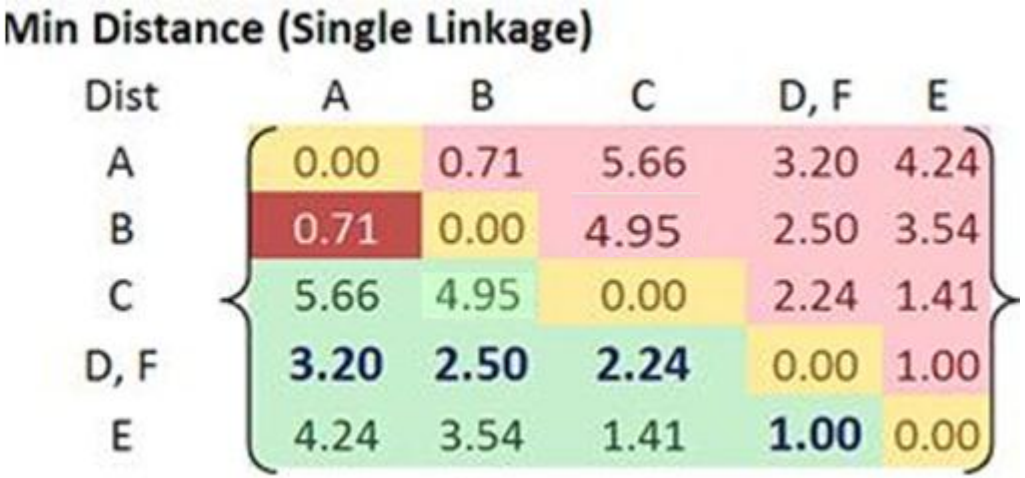


Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

AGGLOMERATIVE ALGORITHM - EXAMPLE

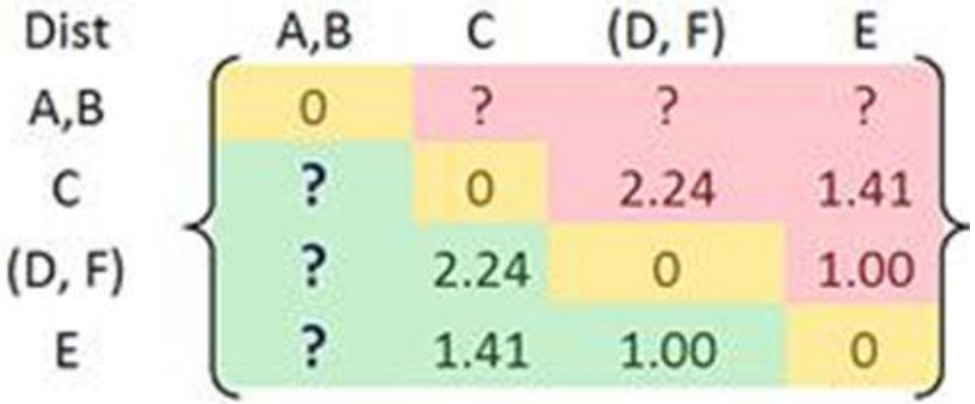
Merge two closest clusters (Iteration 2)





Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

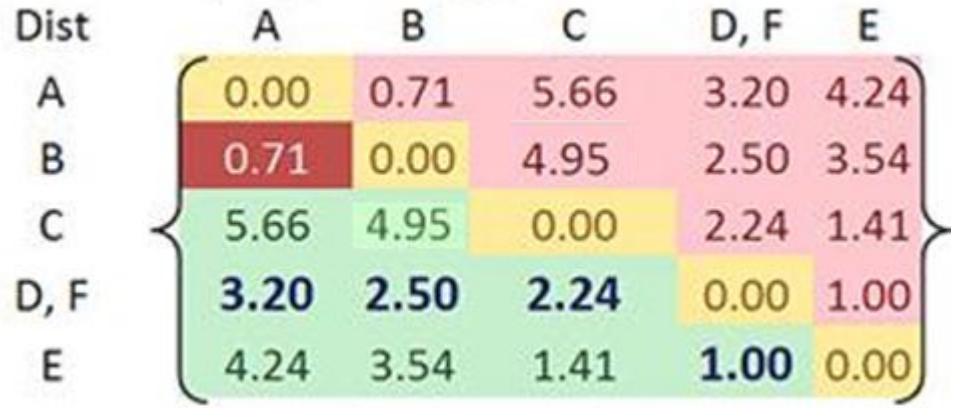


Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

AGGLOMERATIVE ALGORITHM - EXAMPLE

Update Distance Matrix (iteration 2)

Min Distance (Single Linkage)

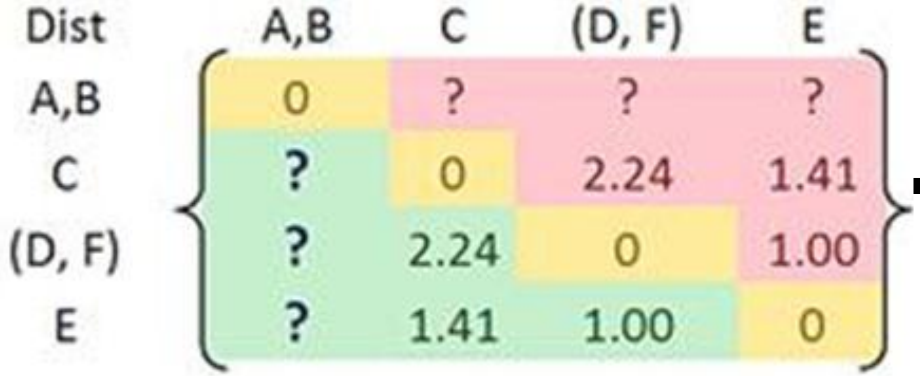


Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

$$d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

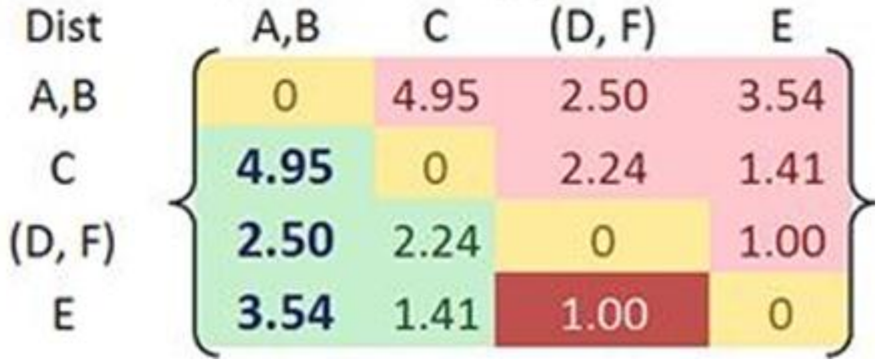
$$d_{(D,F)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}\right)$$
$$= \min\left(3.61, 2.92, 3.20, 2.50\right) = 2.50$$

$$d_{E\to(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$



Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	**?**	0	2.24	1.41
(D, F)	**?**	2.24	0	1.00
E	**?**	1.41	1.00	0

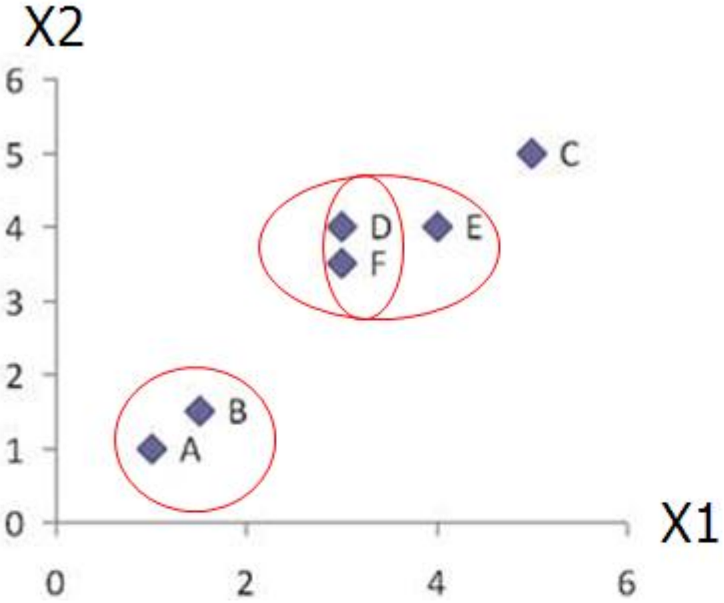
Min Distance (Single Linkage)

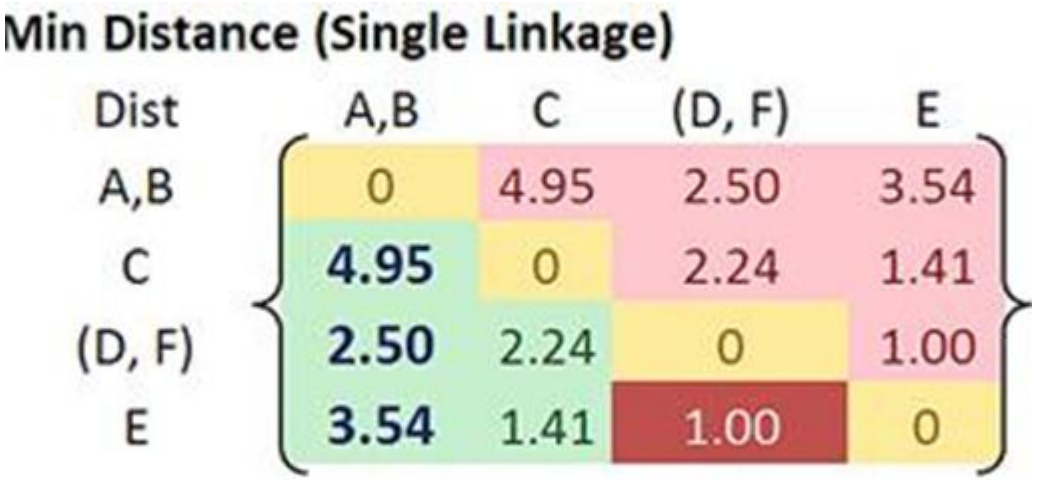


Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

AGGLOMERATIVE ALGORITHM - EXAMPLE

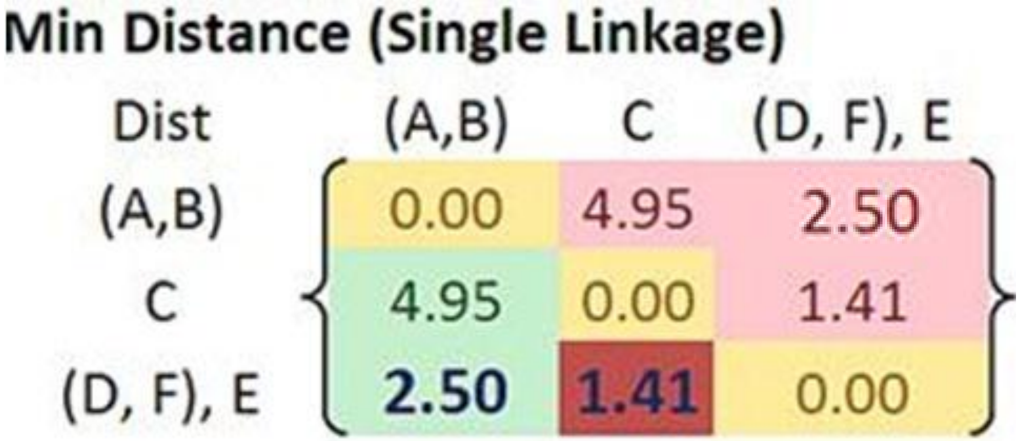
Merge two closest clusters (Iteration 3)





Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

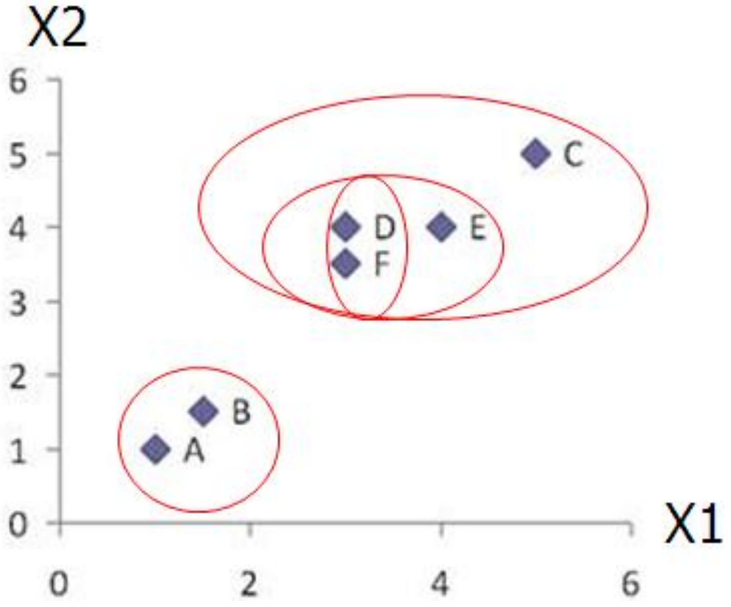


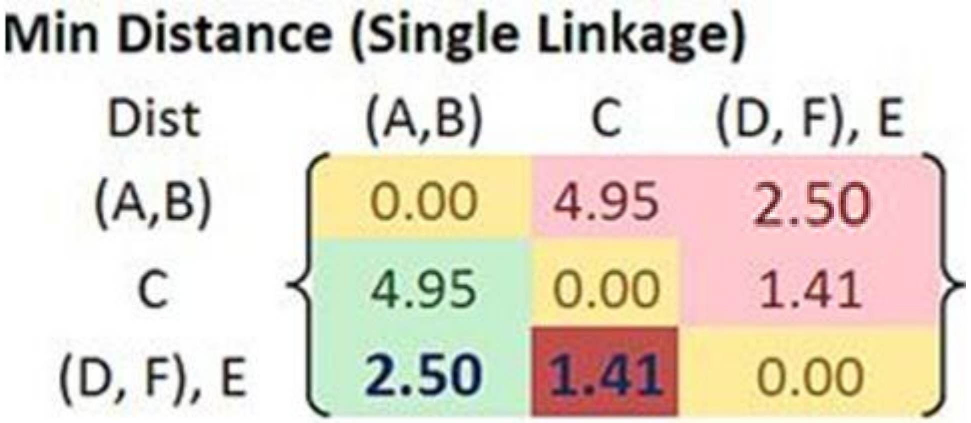
Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

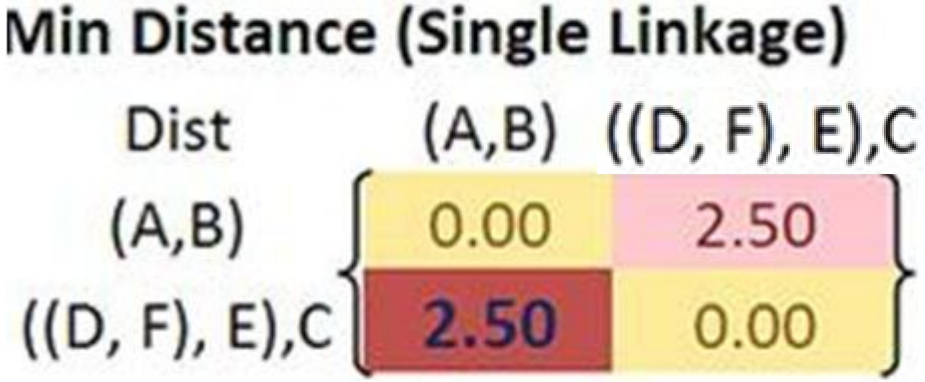
AGGLOMERATIVE ALGORITHM - EXAMPLE

Merge two closest clusters and update distance matrix (Iteration 4)




Min Distance (Single Linkage)

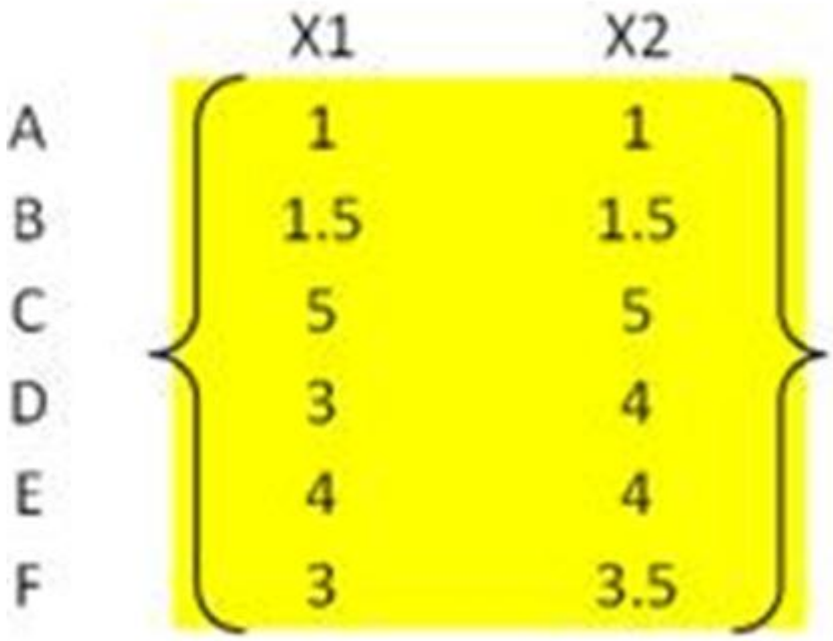
Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00


Min Distance (Single Linkage)

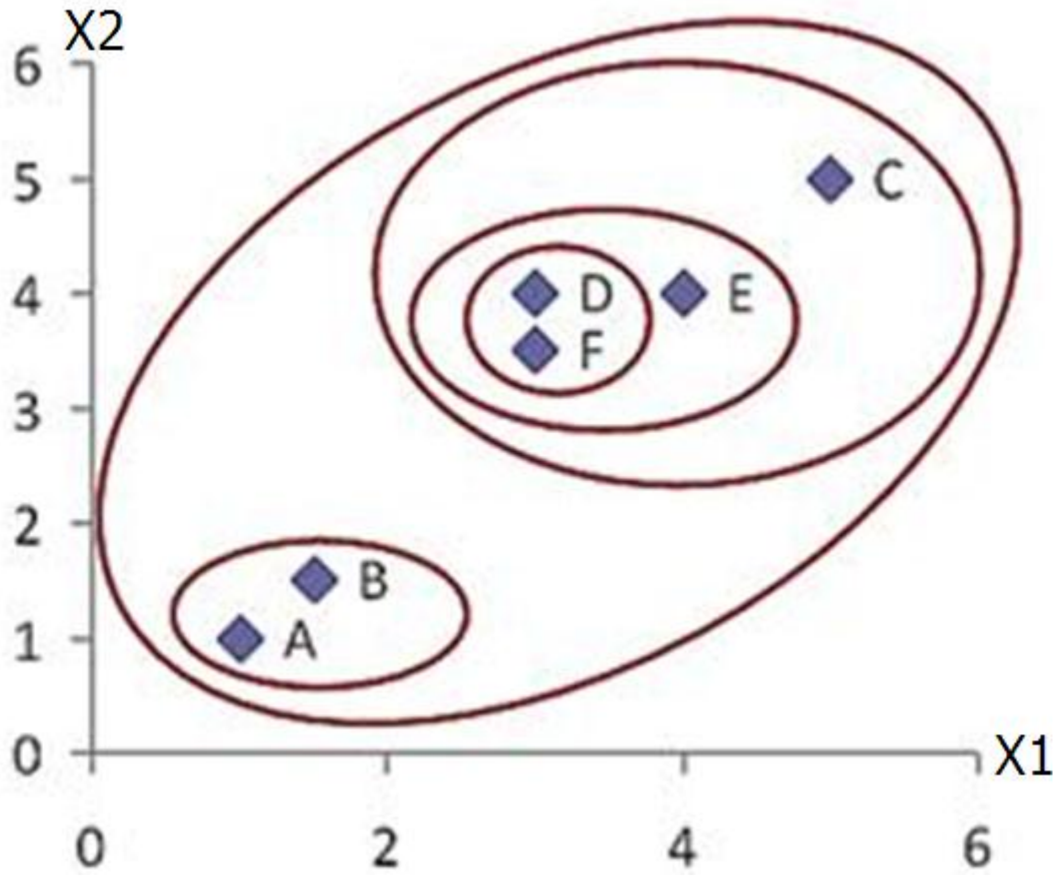
Dist	(A,B)	((D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

AGGLOMERATIVE ALGORITHM - EXAMPLE

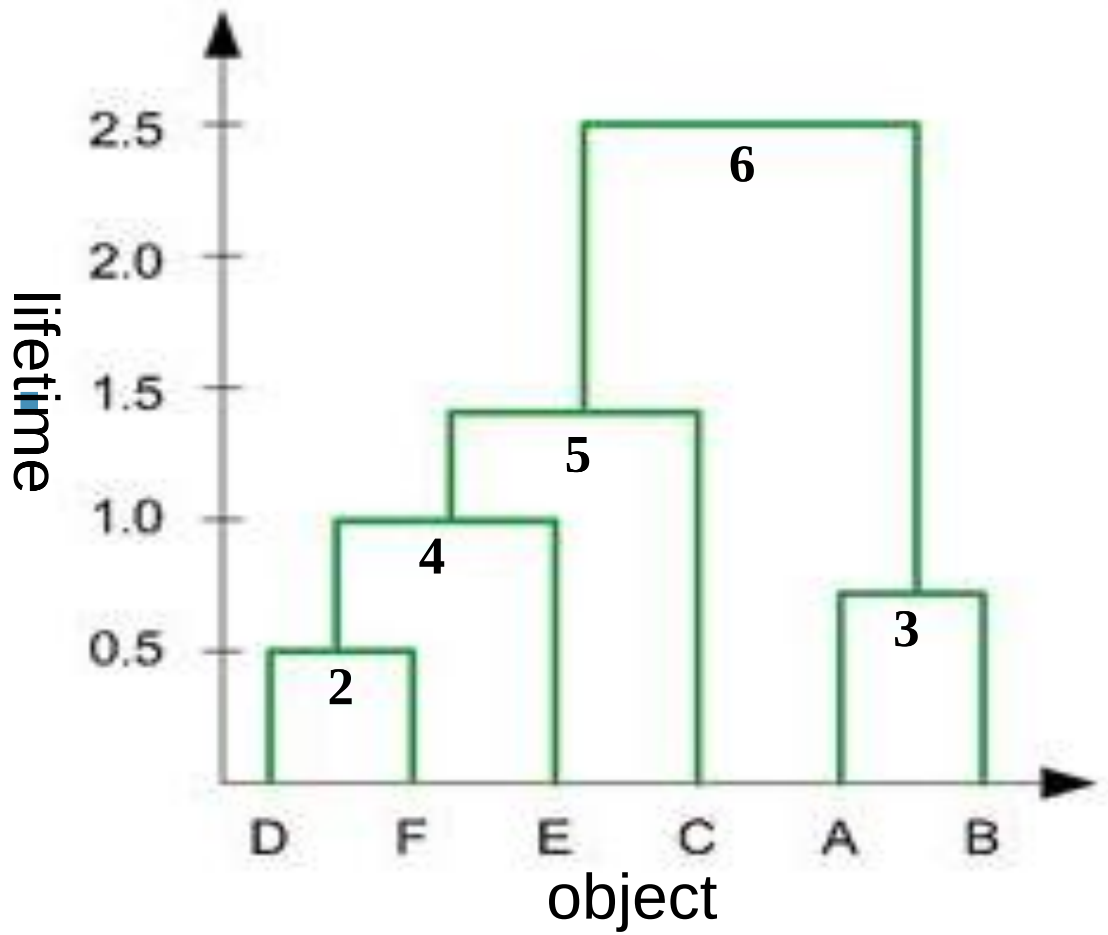
Final Result (Meeting Termination Condition)



	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



DENDROGRAM TREE REPRESENTATION



1. In the beginning we have 6 clusters: A, B, C, D, E and F
2. We merge clusters D and F into cluster (D, F) at distance 0.50
3. We merge cluster A and cluster B into (A, B) at distance 0.71
4. We merge clusters E and (D, F) into ((D, F), E) at distance 1.00
5. We merge clusters ((D, F), E) and C into (((D, F), E), C) at distance 1.41
6. We merge clusters (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50
7. The last cluster contain all the objects, thus conclude the computation

CASE STUDY: I CLUSTERING OF VEHICLES

CASE STUDY 2: CLUSTERING OF US VOTERS

SUMMARY

- Hierarchical algorithm is a sequential clustering algorithm
  - Use distance matrix to construct a tree of clusters (dendrogram)
  - Hierarchical representation without the need of knowing # of clusters (can set termination condition with known # of clusters)
- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Sensitive to cluster distance measures and noise/outliers
  - Less efficient: $O\ (n^2\ logn)$, where n is the number of total objects
- There are several variants to overcome its weaknesses
  - BIRCH: scalable to a large data set
  - ROCK: clustering categorical data
  - CHAMELEON: hierarchical clustering using dynamic modelling

REFERENCES

- **Presentation Material - Hierarchical Clustering - Ke Chen**

THANK YOU ☺ !

QUESTIONS?

THANK YOU!

Want to learn more about Big Data and Machine Learning ☺ ??