# CLUSTERING

CSE 634 Data Mining
Prof. Anita Wasilewska

# REFERENCES


1. K-medoids:

https://www.coursera.org/learn/cluster-analysis/lecture/nJ0Sb/3-4-the-k-medoids-clustering-method

https://anuradhasrinivas.files.wordpress.com/2013/04/lesson8-clustering.pdf

2. K-means:

https://www.datascience.com/blog/k-means-clustering

https://en.wikipedia.org/wiki/Elbow_method_(clustering)

3. CLARA:

http://www.sthda.com/english/articles/27-partitioning-clustering-essentials/89-clara-clustering-large-applications/

4. Book:

Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber ,Morgan Kaufman ,2011
Chapter : 10, Page: 445-454

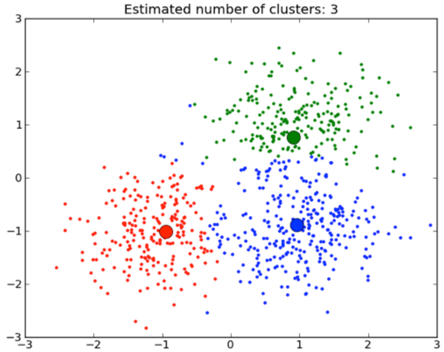
# PART 1

# OVERVIEW

- What is clustering?
- Similarity Measures
- Requirements of good clustering algorithm
- K-mean clustering
- K-medoids clustering - PAM
- K-medoids clustering - CLARA
- Applications of K-means and K-medoids

# 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***
- It can also be called a method of ***data exploration*** **.**



# SIMILARITY MEASURES

- A ***good clustering*** method will produce high quality clusters with

1.high intra-class similarity

2.low inter-class similarity

- The ***quality*** of a clustering result depends on:

1. similarity measure used by the method and its implementation.
2. its ability to discover some or all of the hidden patterns.

# REQUIREMENTS OF GOOD CLUSTERING ALGORITHM

- ❖ Scalability
- ❖ Discovery of clusters with arbitrary shape
- ❖ Able to deal with noise and outliers
- ❖ Insensitive to order of input records
- ❖ High dimensionality
- ❖ Incorporation of user-specified constraints

# CLUSTERING ALGORITHMS

1. K-Means
2. K-medoids
    - 2.1 Basic K-medoids
    - 2.2 PAM
    - 2.3 CLARA

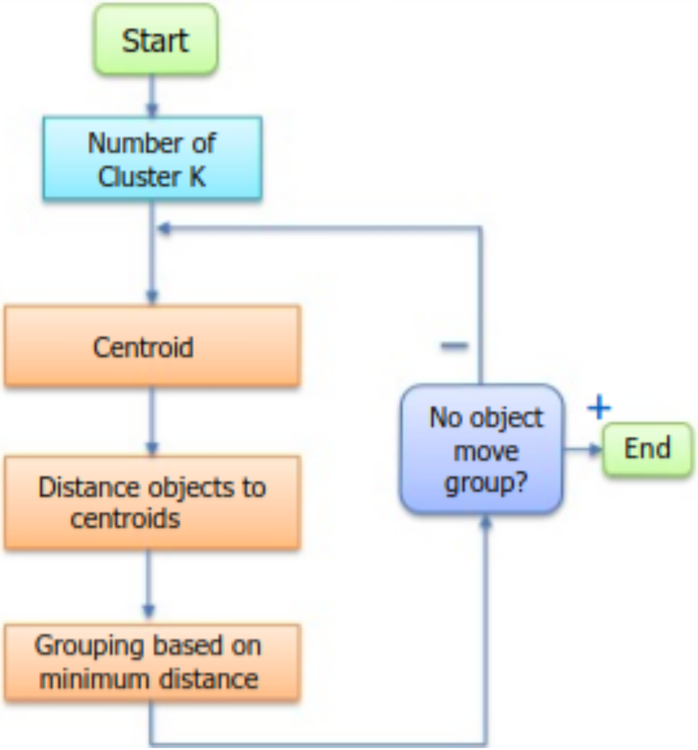
# K-MEANS CLUSTERING

- First used by James Mcqueen in 1967
- Unsupervised Learning
- Goal : Find the groups in the given data where no of groups is denoted by K
- Groups made on Feature similarity
- Results expected:
  - Centroids used to label data
  - Labels for training data
- Uses : Behavioral segmentation, Inventory categorization,Sorting sensor measurements

# K-MEANS PROCESS

- **<u>Input</u>**
  - Data set
  - K i.e no of clusters
- **<u>Data Assignment Step</u>**
  - Each centroid represents one cluster
  - Each data point is assigned to its nearest cluster based on squared **Euclidean distance**
- **<u>Centroid Update Step</u>**
  - Recompute the centroids by taking the mean of the data points assigned to that particular centroid
- The algorithms repeats the two steps until end condition is met:
  - No change in clusters

# K-MEANS PROCESS



# K-MEANS ALGORITHM

**Input :** K (No of Clusters to form) and Input Data Set

**Initialize:** Randomly assign K cluster centroids $\mu_1$ , $\mu_2$, ……………………..$\mu_{K,}$є $R^n$

**Repeat{**

for i = 1 to m

$c^{(i)}$:=index(1 to K) of cluster centroid closest to $x^{(i)}$(datapoint)

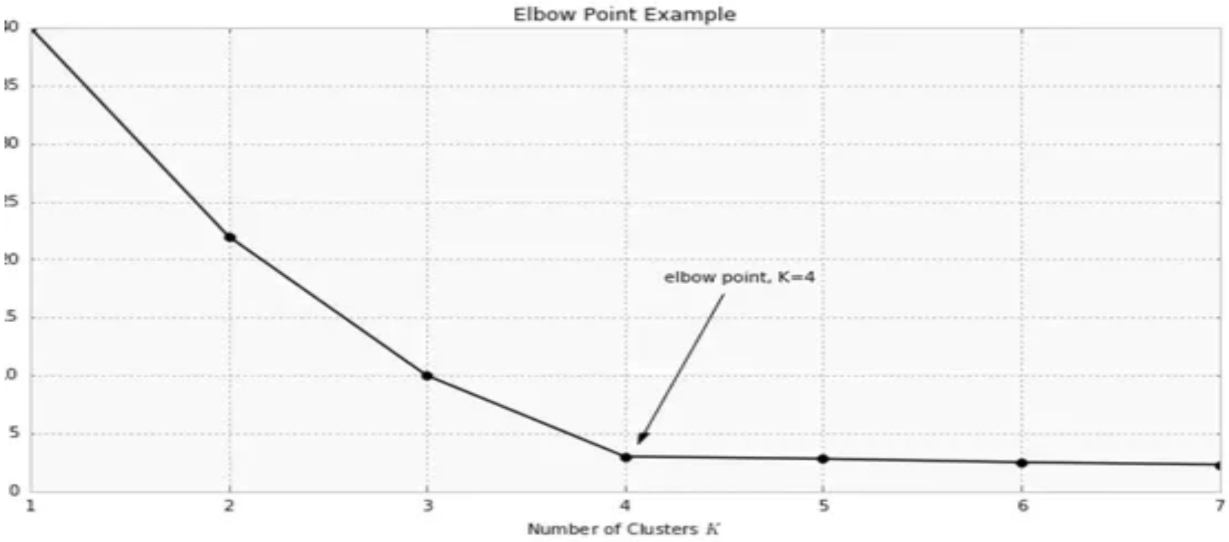
for k = 1 to K

$\mu_K$:=average mean of points assigned to cluster K

**}** Stop when convergence criteria is meet.

# CHOOSING NUMBER OF CLUSTERS K

- **<u>Elbow- Join method</u>**
  - Metric used is mean distance between data points and their cluster centroid.



# K-MEANS EXAMPLE

| Subject | A | B |
|---|---|---|
| 1 | 1.0 | 1.0 |
| 2 | 1.5 | 2.0 |
| 3 | 3.0 | 4.0 |
| 4 | 5.0 | 7.0 |
| 5 | 3.5 | 5.0 |
| 6 | 4.5 | 5.0 |
| 7 | 3.5 | 4.5 |

| | Individual | Mean Vector (centroid) |
|---|---|---|
| Group 1 | 1 | (1.0, 1.0) |
| Group 2 | 4 | (5.0, 7.0) |

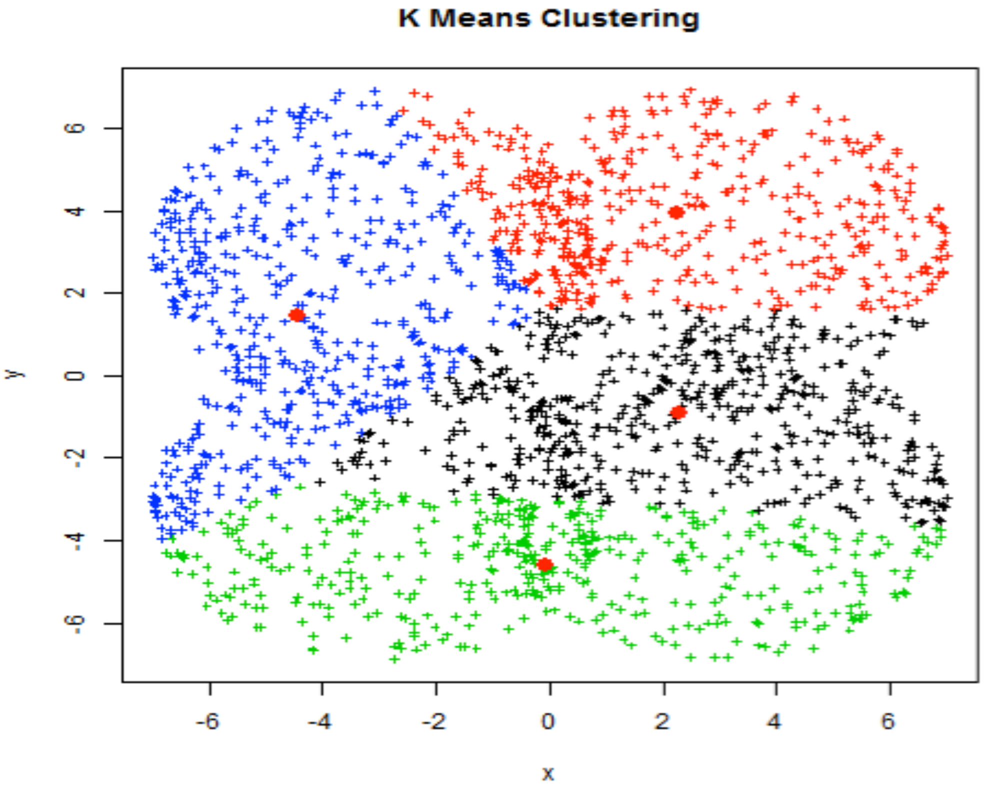
| | Cluster 1 | | Cluster 2 | |
|---|---|---|---|---|
| Step | Individual | Mean Vector (centroid) | Individual | Mean Vector (centroid) |
| 1 | 1 | (1.0, 1.0) | 4 | (5.0, 7.0) |
| 2 | 1, 2 | (1.2, 1.5) | 4 | (5.0, 7.0) |
| 3 | 1, 2, 3 | (1.8, 2.3) | 4 | (5.0, 7.0) |
| 4 | 1, 2, 3 | (1.8, 2.3) | 4, 5 | (4.2, 6.0) |
| 5 | 1, 2, 3 | (1.8, 2.3) | 4, 5, 6 | (4.3, 5.7) |
| 6 | 1, 2, 3 | (1.8, 2.3) | 4, 5, 6, 7 | (4.1, 5.4) |

| | Individual | Mean Vector (centroid) |
|---|---|---|
| Cluster 1 | 1, 2, 3 | (1.8, 2.3) |
| Cluster 2 | 4, 5, 6, 7 | (4.1, 5.4) |

| Individual | Distance to mean (centroid) of Cluster 1 | Distance to mean (centroid) of Cluster 2 |
|---|---|---|
| 1 | 1.5 | 5.4 |
| 2 | 0.4 | 4.3 |
| 3 | 2.1 | 1.8 |
| 4 | 5.7 | 1.8 |
| 5 | 3.2 | 0.7 |
| 6 | 3.8 | 0.6 |
| 7 | 2.8 | 1.1 |

| | Individual | Mean Vector (centroid) |
|---|---|---|
| Cluster 1 | 1, 2 | (1.3, 1.5) |
| Cluster 2 | 3, 4, 5, 6, 7 | (3.9, 5.1) |

# K-MEANS EXAMPLE




http://rossfarrelly.blogspot.com/2012/12/k-means-clustering.html

# ADVANTAGES AND DISADVANTAGES

**Advantages**

1. Easyto implement.
2. With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small)

**Disadvantages**

1. Difficult to predict the number of clusters (K-Value).
2. Can converge on local minima
3. Sensitive to outliers

# K-MEDOIDS CLUSTERING

The mean in k-means clustering is sensitive to outliers. Since an object with an extremely high value may substantially distort the distribution of data.

Hence we move to k-medoids.

Instead of taking mean of cluster we take the most centrally located point in cluster as it's center.

These are called medoids.

# K-MEANS & K-MEDOIDS Clustering- Outliers Comparison

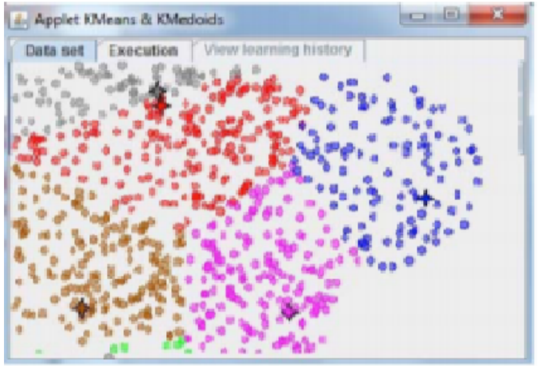


**Fig.6** Outliers in K-Means

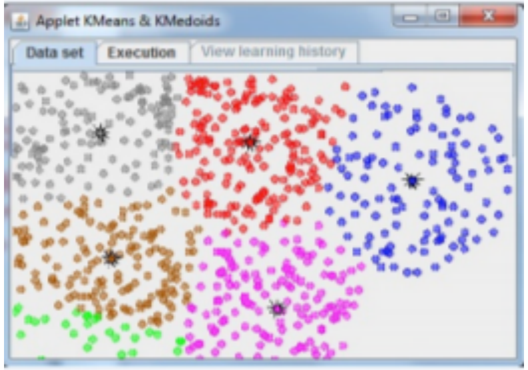


**Fig.7** Outliers in K-Medoids

# K-MEDOIDS - BASIC ALGORITHM

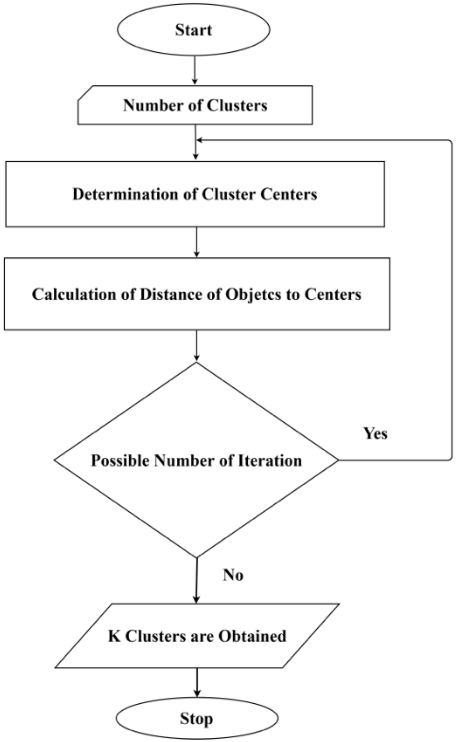
**Input** : Number of K (the clusters to form)

**Initialize**:
Select K points as the initial representative objects i.e initial K-medoids of our K clusters.

**Repeat**:
**Assign** each point to the cluster with the closest medoid m.
Randomly select a non-representative object $o_i$
Compute the total cost of **swapping** S, the medoid m with $o_i$
If $S < 0$:
Swap m with $o_i$ to form new set of medoids.

**Stop** when convergence criteria is meet.

# K-MEDOIDS - BASIC FIOWCHART




https://www.researchgate.net/figure/Flowchart-of-K-means-algorithm_fig1_314078156

# K-MEDOIDS - PAM ALGORITHM

**PAM** stands for **Partitioning Around Medoids**.

**GOAL**: To find Clusters that have minimum average dissimilarity between objects that belong to same cluster.

**ALGORITHM:**

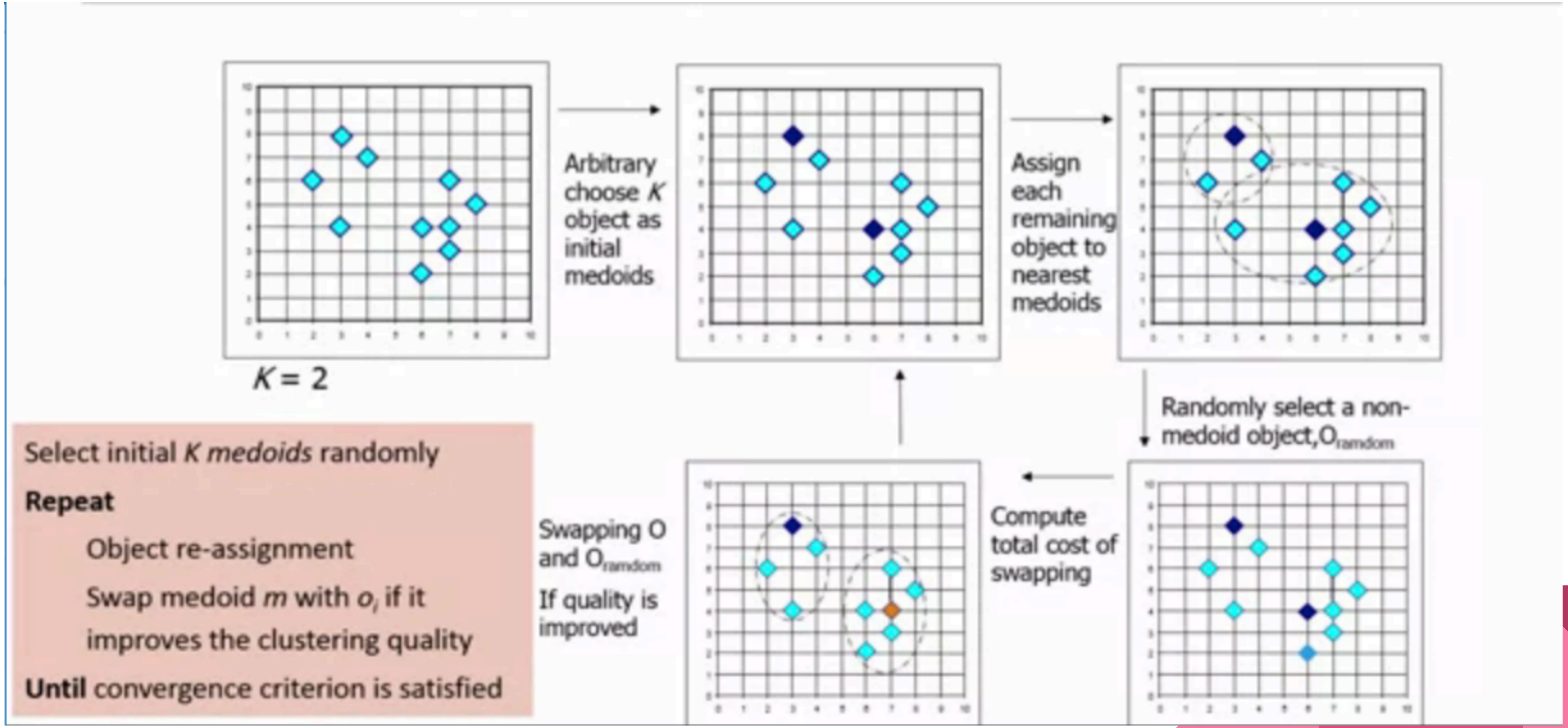
1. Start with initial set of medoids.
2. Iteratively replace one of the medoids with a non-medoid if it reduces total sum of SSE of resulting cluster.

SSE is calculated as below:

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

Where k is number of clusters and x is a data point in cluster $C_i$ and $M_i$ is medoid of $C_i$

# TYPICAL PAM EXAMPLE




https://www.coursera.org/learn/cluster-analysis/lecture/nJ0Sb/3-4-the-k-medoids-clustering-method

# K-MEDOIDS (PAM) EXAMPLE

**Data Objects**

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |

For K = 2

Randomly Select m1 = (3,4) and m2 =(7,4)

Using Manhattan as similarity metric we get,

C1 = ( o1, o2, o3, o4 )

C2 = ( o5, o6, o7, o8, o9, o10)

# K-MEDOIDS (PAM) EXAMPLE

**Data Objects**

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |

Compute absolute error as follows,

E = (o1-o2) + (o3-o2) + (o4-o2)

+

(o5-o8) +(o6-o8)+(o7-o8) +(o9-o8) + (o10-o8)

E = (3+4+4) + (3+1+1+2+2)

Therefore,

E = 20

# K-MEDOIDS (PAM) EXAMPLE

**Data Objects**

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |

Swapping o8 with o7

Compute absolute error as follows,

E = (o1-o2) + (o3-o2) + (o4-o2)

+

(o5-o7) +(o6-o7)+(o8-o7) +(o9-o7) + (o10-o7)

E = (3+4+4) + (2+2+1+3+3)

Therefore,

E = 22

# K-MEDOIDS (PAM) EXAMPLE

**Data Objects**

| | $A_1$ | $A_2$ |
|---|---|---|
| $O_1$ | 2 | 6 |
| $O_2$ | 3 | 4 |
| $O_3$ | 3 | 8 |
| $O_4$ | 4 | 7 |
| $O_5$ | 6 | 2 |
| $O_6$ | 6 | 4 |
| $O_7$ | 7 | 3 |
| $O_8$ | 7 | 4 |
| $O_9$ | 8 | 5 |
| $O_{10}$ | 7 | 6 |

Let’s now calculate cost function S for this swap,

S = E for (o2,07) - E for (o2, o8)

S = 22- 20

Therefore S > 0,

This swap is undesirable.

# ADVANTAGES and DISADVANTAGES of PAM

**Advantages:**
PAM is more flexible as it can use any similarity measure.
PAM is more robust than k-means as it handles noise better.

**Disadvantages:**
PAM algorithm for K-medoid clustering works well for dataset but cannot scale well for large data set due to high computational overhead.

PAM COMPLEXITY : $O(k(n-k)^2)$ this is because we compute distance of n-k points with each k point, to decide in which cluster it will fall and after this we try to replace each of the medoid with a non medoid and find it's distance with n-k points.

To overcome this we make use of CLARA.

# CLARA - CLUSTERING  LARGE APPLICATIONS

- Improvement over PAM
- Finds medoids in a sample from the dataset

[Idea]: If the samples are sufficiently random, the medoids of the sample approximate the medoids of the dataset

[Heuristics]: 5 samples of size 40+2k gives satisfactory results

- Works well for large datasets (n=1000, k=10)

https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/CLARA

# CLARA ALGORITHM

1. Split randomly the data sets in multiple subsets with fixed size (sampsize)
2. Compute PAM algorithm on each subset and choose the corresponding k representative objects (medoids). Assign each observation of the entire data set to the closest medoid.
3. Calculate the mean (or the sum) of the dissimilarities of the observations to their closest medoid. This is used as a measure of the goodness of the clustering.
4. Retain the sub-dataset for which the mean (or sum) is minimal. A further analysis is carried out on the final partition.

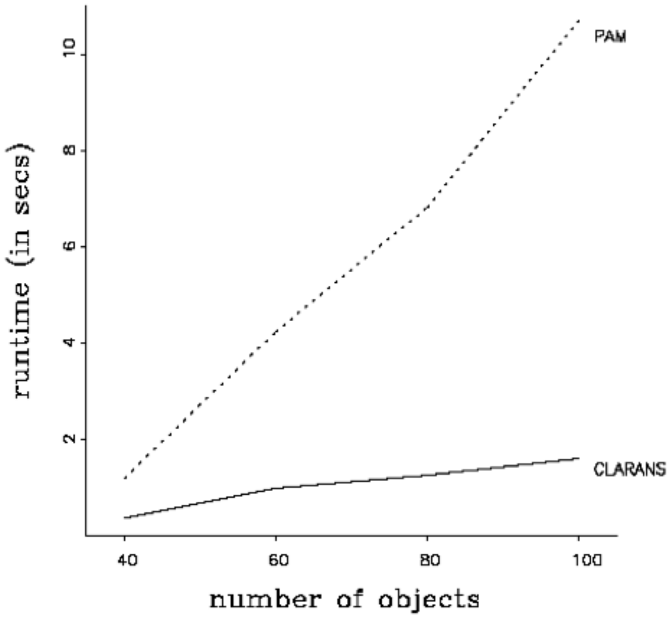
# COMPARISON CLARA vs PAM

<u>Strength</u>:

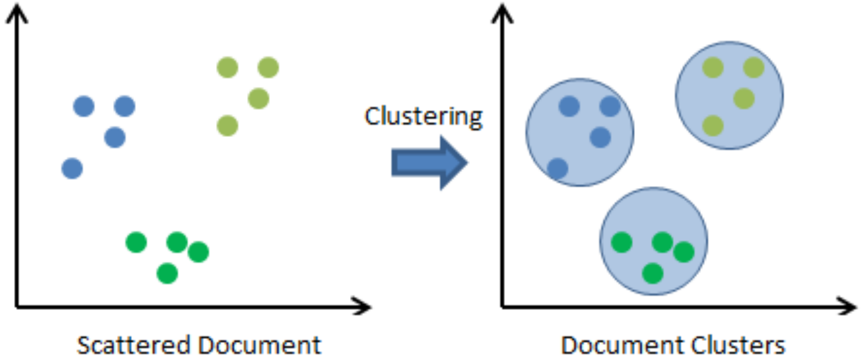
- deals with larger data sets than *PAM*
- CLARA Outperforms PAM in terms of running time and quality of clustering

<u>Weakness:</u>

- Efficiency depends on the sample size
- A good clustering based on samples will not necessarily represent a good clustering of the whole



# APPLICATIONS




Document Clustering

Social Network

# GENERAL APPLICATIONS OF CLUSTERING

1. Recognition
2. Spatial Data Analysis
    a. create thematic maps in GIS by clustering feature spaces
    b. detect spatial clusters and explain them in spatial data mining
1. Image Processing
2. Economic Science (especially market research)
3. WWW
    a. Document classification
    b. Cluster Weblog data to discover groups of similar access patterns

# PART 2

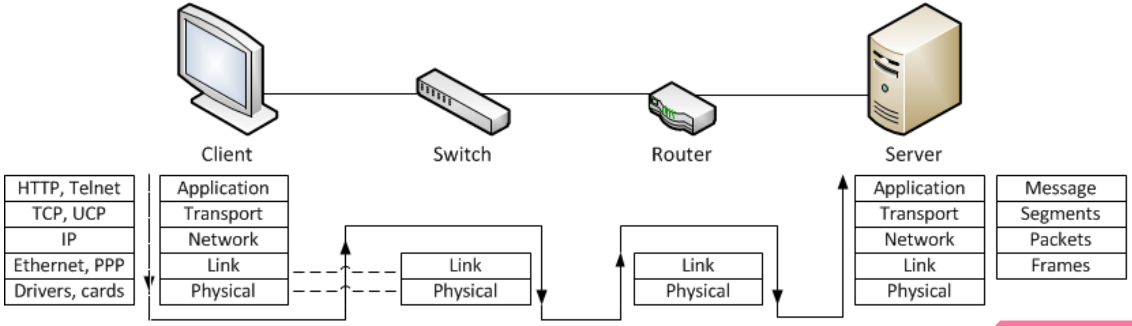
# TRAFFIC ANOMALY DETECTION USING K-MEANS CLUSTERING

Authors:
Gerhard Munz, Sa Li, Georg Carle,
Computer Networks and Internet,
Wilhelm Schickard Institute for Computer Science,
University of Tuebingen, Germany

Published in GI/ITG Workshop MMBnet, 2007.

# NETWORK DATA MINING

- Knowledge about monitoring data. Helps in determining dominant characteristics and outliers.
- Deployed to define rules or patterns that are typical for specific kinds of traffic helps to analyze new sets of monitoring data(labeling).



# NOVEL NDM APPROACH

❖ K-means clustering of monitoring data

Aggregate and transform flow records into datasets for equally spaced time intervals

- ❖ Raw Data and Extracted Features
  - ➢ Total number of packets sent
  - ➢ Total number of bytes sent
  - ➢ Number of different source-destination pairs

- K-means Clustering
  - Distance metric used is

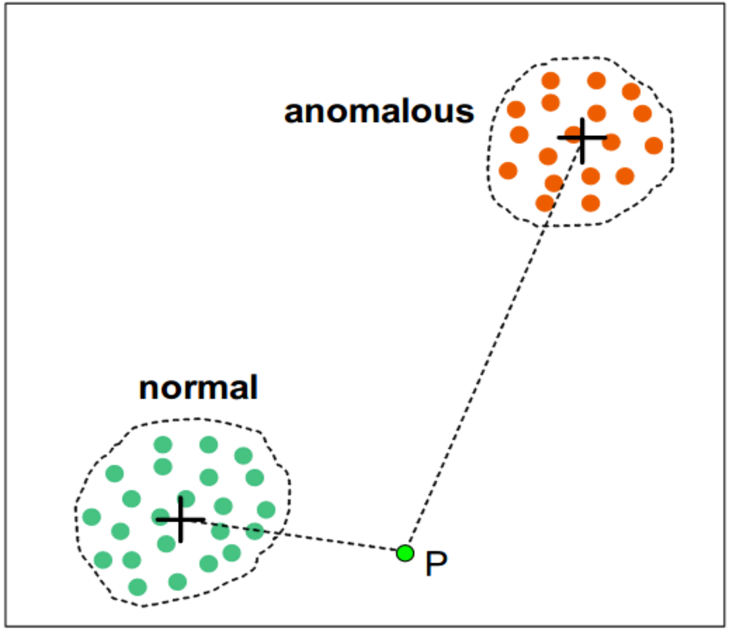
$$d(x,y) = \sqrt{\sum_{i=1}^{m} \left( \frac{x_i - y_i}{s_i} \right)^2}$$

$s_i$ is an empirical normalization

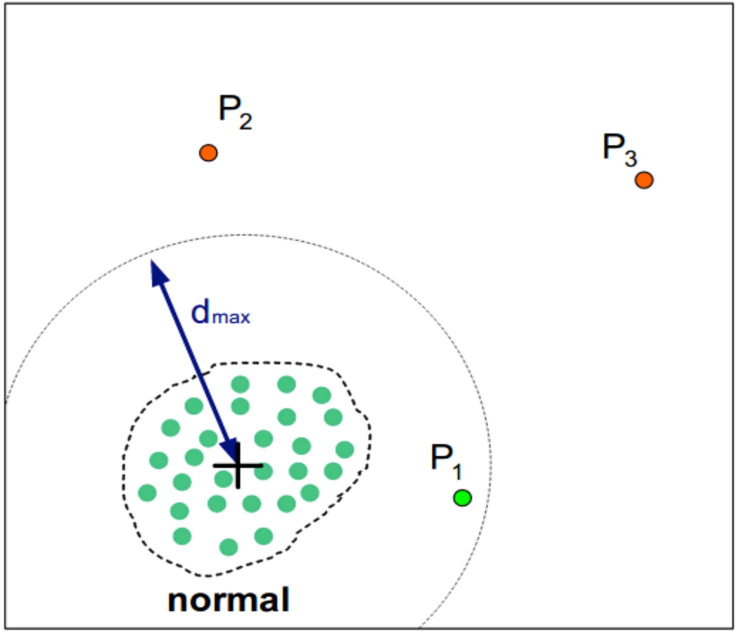
$s_{packets} = s_{bytes} = 5$

$s_{src-dst} = 1.$

# CLASSIFICATION AND OUTLIER DETECTION

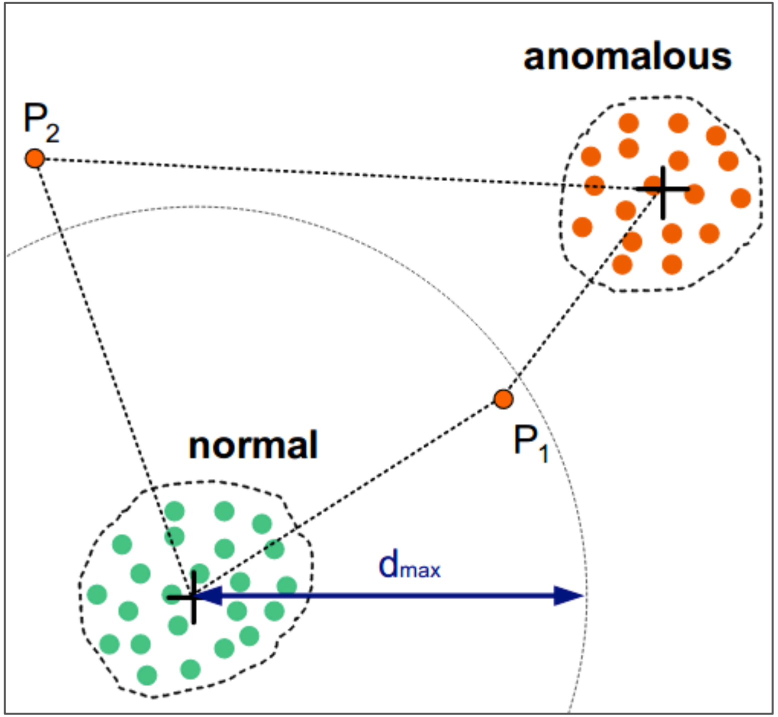



Classification



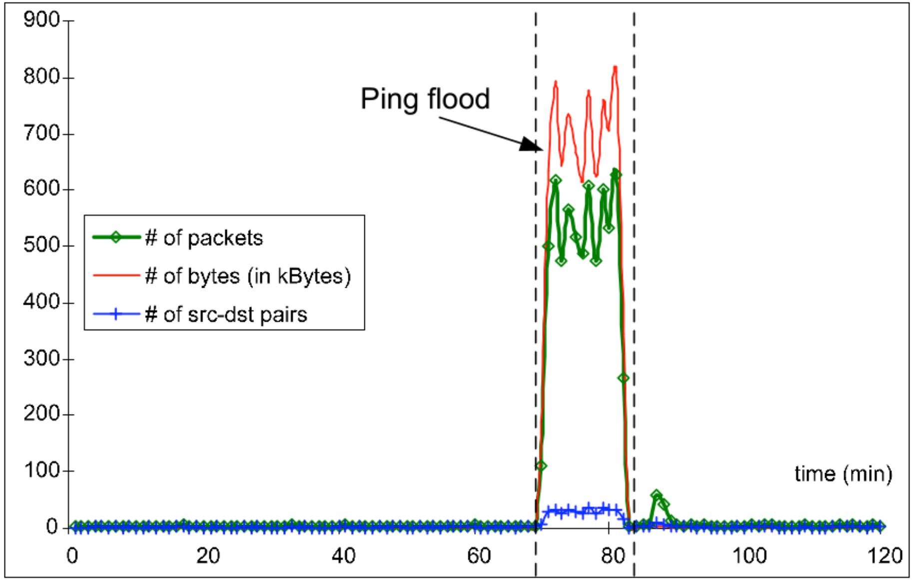

Outlier detection

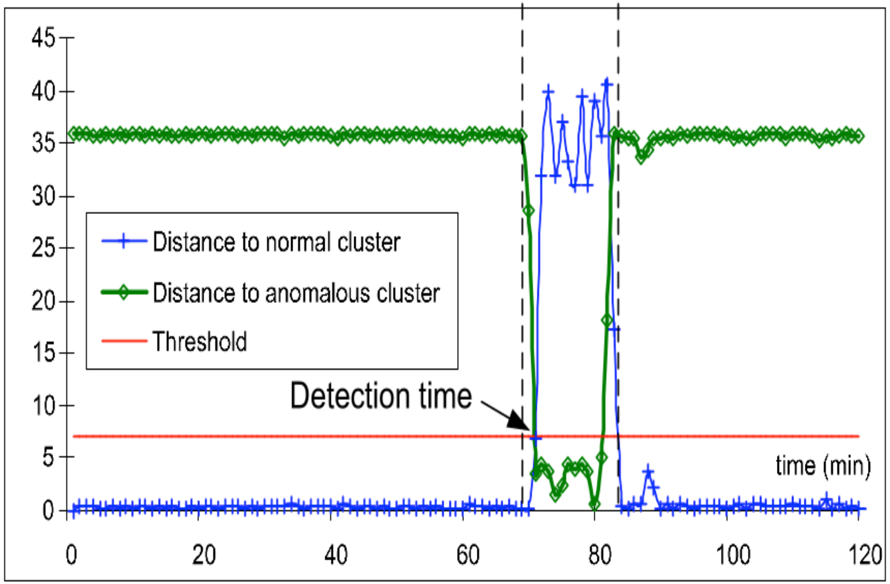
# CLASSIFICATION AND OUTLIER DETECTION




Combined approach

# RESULTS : PING FLOOD DETECTION



# CONCLUSIONS

- The resulting cluster centroids can be used to detect anomalies in new on-line monitoring data with a small number of distance calculations.
- Applying the clustering algorithm separately for different services improves the detection quality.
- The algorithm is scalable.
- Optimum number of clusters K is difficult to decide.