Data Mining

Munawar, PhD

Agenda

01 Cluster Analysis

02 Flat Clustering

03 Clustering in Data Warehouse

04 QA ?

Cluster Analysis

Cluster Analysis

- Supervised learning
  - The training data is accompanied by labels indicating the class of the observations
  - Major application: classification
- Unsupervised learning
  - The class labels of training data are unknown
  - Major application: **Cluster Analysis**

Cluster Analysis

- Clustering?
  - Deals with finding some structure in a collection of **unlabeled data**
- Definition
  - **Clustering** is the process of **organizing objects** into groups, whose members are **similar** in some way

Cluster Analysis

- Clustering in human life
  - Early in childhood we learn how to distinguish between cats and dogs, or between animals and plants
    - By continuously improving subconscious clustering schemes

Cluster Analysis

- Clustering (also called **data segmentation**)
  - A form of **learning by observation** rather than learning by example
  - Is used in numerous applications
    - Market research
    - Pattern recognition
    - Data analysis
    - Information retrieval
    - Image processing

Cluster Analysis

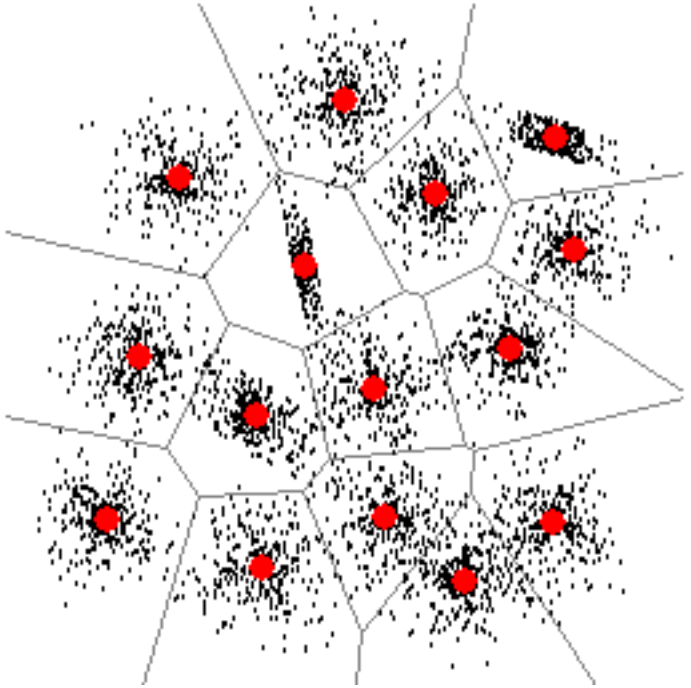
- **Requirements** of cluster analysis
  - Scalability
    - Highly scalable algorithms are needed for clustering on large data sets
  - Ability to deal with different types of attributes
    - Clustering may be performed also on binary, categorical and ordinal data
  - Discovery of clusters with arbitrary shape
    - Most algorithms tend to find spherical clusters
  - Ability to deal with noisy data

Requirements

- High dimensionality
  - D W can contain several dimensions
- Minimal requirements for domain knowledge
  - Clustering results are quite sensitive to the input parameters
  - Parameters are often difficult to determine

Issues in clustering

- Clustering is quite challenging!
  - **How many clusters?**
  - **Flat** or **hierarchical?**
  - **Hard** or **soft?**
  - What's a **good** clustering?
  - How to **find** it?

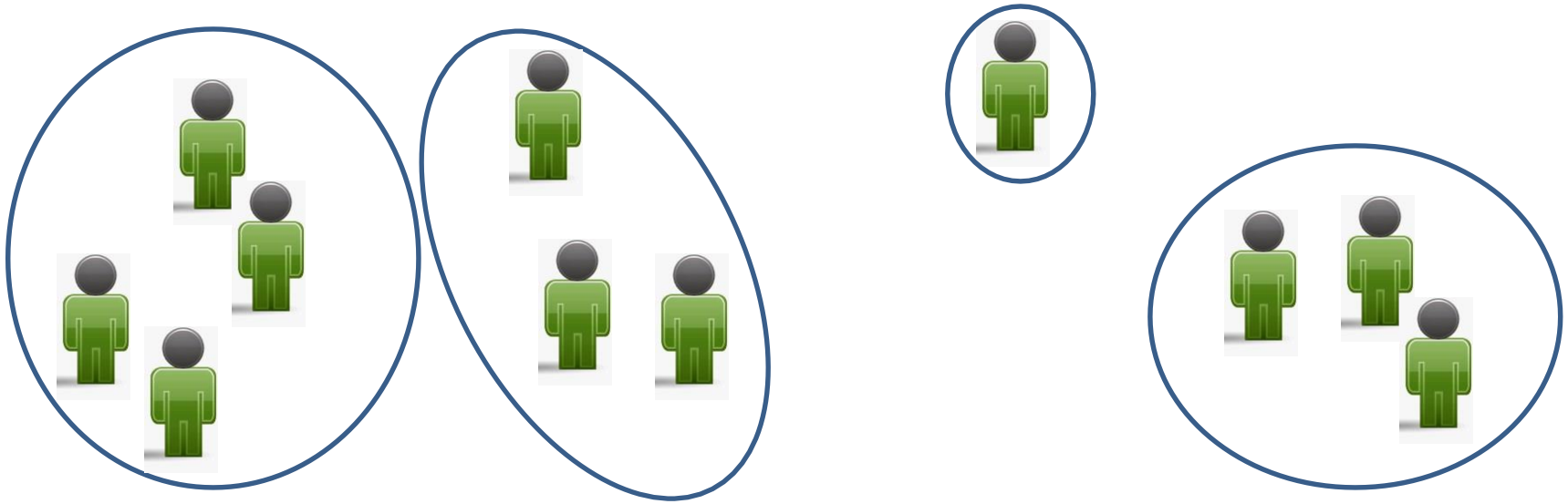


Issues in clustering

- **How many clusters?**
  - Let **k** denote the **number of clusters** from now on
  - Basically, there are two different approaches regarding the choice of k
    - **Define k** before searching for a clustering, then only consider clusterings having exactly k clusters
    - **Do not define a fixed k**, i.e. let the number of clusters depend on some measure of clustering quality to be defined
  - The "right" choice depends on the problem you want to solve…

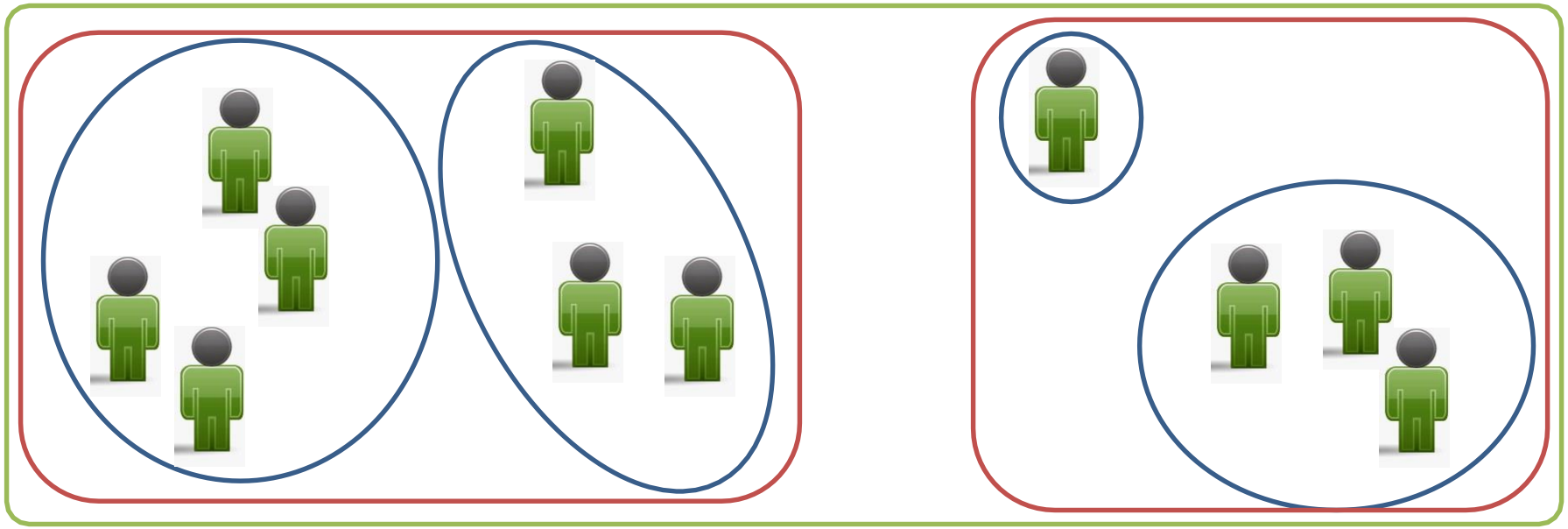
Issues in clustering

- Clustering approaches: **flat** or **hierarchical**?
  - Flat clustering: finding all clusters at once
    - Partition the items into **k clusters**
    - **Iteratively** reallocate items to improve the clustering



Issues in clustering

- Hierarchical clustering: finding new clusters using previously found ones
  - **Agglomerative**: each item forms a cluster, merge clusters to form larger ones
  - **Divisive**: all items are in one cluster, split it up into smaller clusters



Issues in clustering

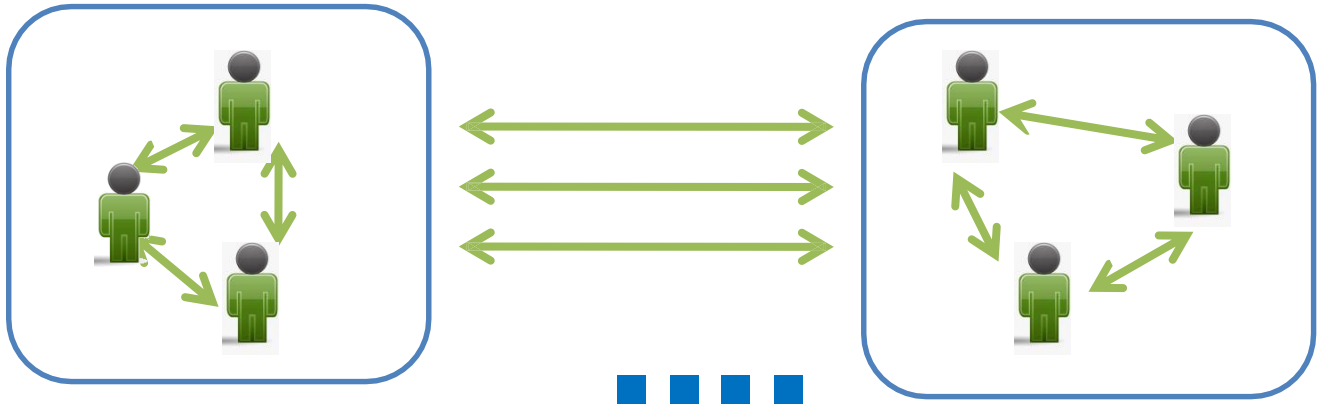
- Hard or soft?
  - **Hard clustering:**
    - Every item is assigned to exactly one cluster (at the lowest level, if the clustering is hierarchical)
    - More common and easier to do
  - **Soft clustering:**
    - An items assignment is a **distribution** over all clusters (fuzzy, probabilistically, or something else)
    - Better suited for creating browsable hierarchies

Issues in clustering

- Abstract problem statement
  - **Given:**
    - A **collection** of items
    - The **type** of clustering to be done (hard/soft)
    - An **objective function** f that assigns a number to any possible clustering of the collection
  - **Task:**
    - Find a clustering that minimizes the objective function (or maximizes, respectively)
  - Exclude a special case: we don't want empty clusters!

Issues in clustering

- The **overall quality** of a clustering is measured by f
  - Usually, f is closely related to a **measure of distance**
- Popular **primary goals:**
  - **Low inter-cluster similarity**, i.e. customers from different clusters should be dissimilar
  - **High intra-cluster similarity**, i.e. all customers within a cluster should be mutually similar



Issues in clustering

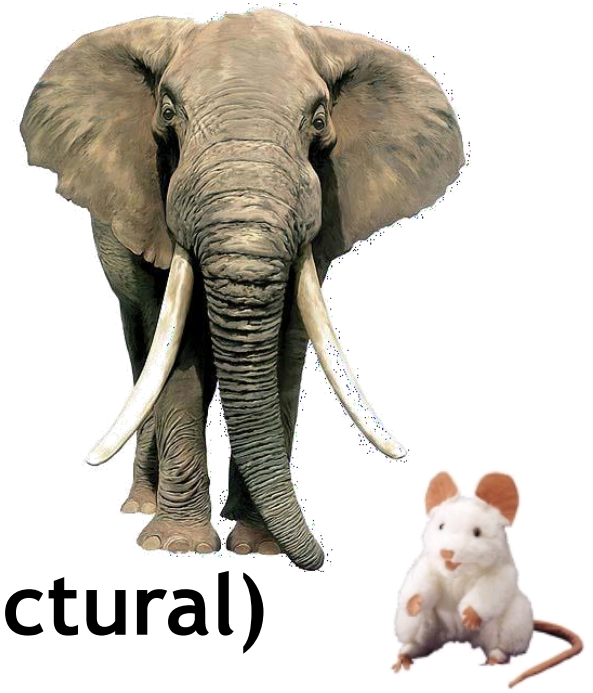
- Inter-cluster similarity and intra-cluster similarity:

BAD:

GOOD:

Issues in clustering

- **Common secondary goals:**
  - Avoid very small clusters
  - Avoid very large clusters
  - …
- All these goals are **internal (structural) criteria**
- **External criteria:** compare the clustering against a hand-crafted reference clustering



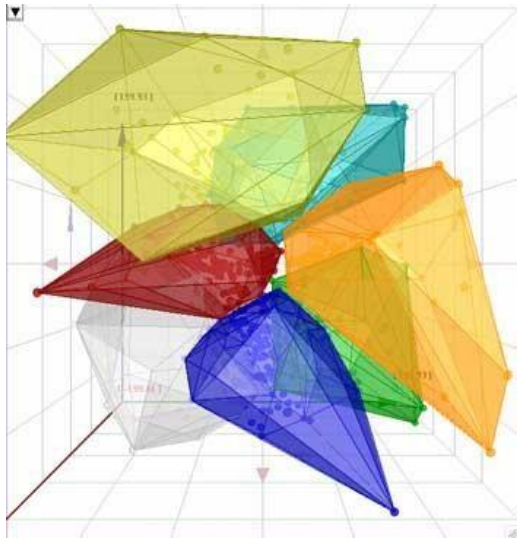
Issues in clustering

- Naïve approach:
  - Try **all possible** clusterings
  - Choose the one minimizing/maximizing f
- How many different clusterings are there?
  - There are S(n, k) distinct hard, flat clusterings of a n-element set into exactly k clusters
    - S(·, ·) are the **Stirling numbers of the second kind**
    - Roughly: S(n, k) is exponential in n
- Better use some heuristics…

Flat Clustering

Flat Clustering

- **Flat clustering**
  - **K-means**
    - A cluster is represented by its center
  - K-medoids or PAM (partition around medoids)
    - Each cluster is represented by one of the objects in the cluster



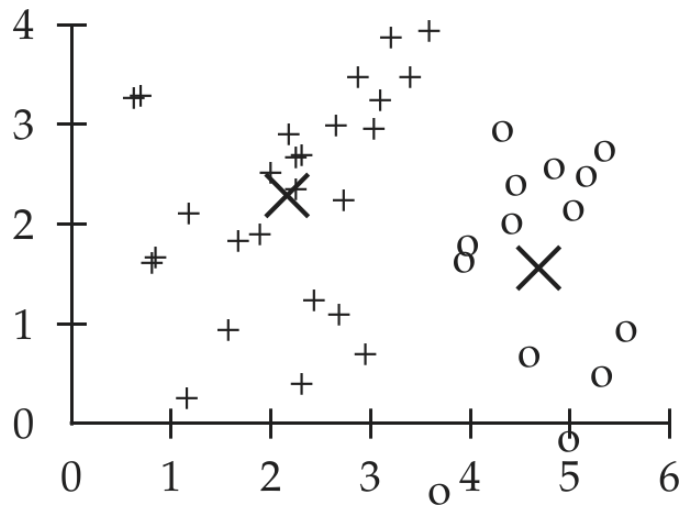
Flat Clustering

- **K-means** clustering
  - The most important **(hard) flat clustering** algorithm, i.e. every cluster is a set of data points (items)
  - The number of clusters k is defined **in advance**
  - Data points usually are represented as **unit vectors**
  - **Objective**
    - **Minimize** the average distance from each node in a cluster to its respective center!

K-means clustering

- **Center** of a cluster
  - Let $A = \{d_1, \ldots, d_m\}$ be a data set cluster (a set of unit vectors)
  - The **centroid** of A is defined as:

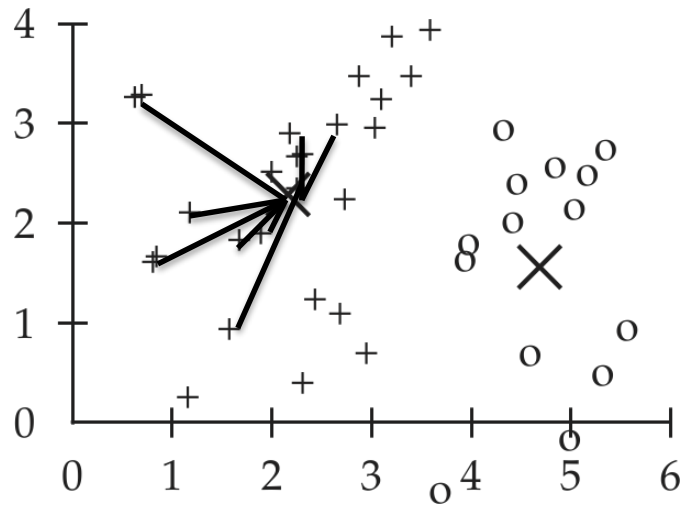
$$\mu(A) = \frac{1}{m} \sum_{i=1}^{m} d_i$$



K-means clustering

- **Quality** of a cluster
  - Again, let A be a data set cluster with m items
  - The **residual sum of squares** (RSS) of A is defined as

$$\mathrm{RSS}(A) = \sum_{i=1}^{m} \left\| d_i - \mu(A) \right\|^2$$

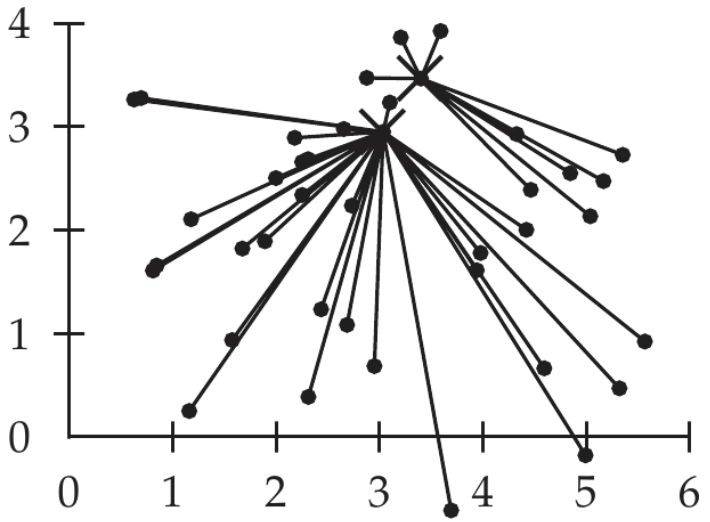


K-means clustering

- In k-means clustering, the **quality of the clustering** into (disjoint) clusters $A_1, \ldots, A_k$ is measured by:

$$\text{RSS}(A_1, \ldots, A_k) = \sum_{j=1}^{k} \text{RSS}(A_j)$$

- K-means clustering tries to **minimize this value**
  - Minimizing $\text{RSS}(A_1, \ldots, A_k)$ is equivalent to **minimizing the average squared distance** between each data point and its cluster's centroid



Bad quality clusters

K-means clustering

- The **k-means algorithm** (aka Lloyd's algorithm):

1. Create k **empty clusters**
2. Assign exactly one centroid to each cluster (seed points)
3. Iterate over the whole data points: assign each data point to the cluster with the nearest centroid
4. Recompute cluster centroids based on contained data points
5. Recalculate the distance between each data point and newly computed cluster centroids.
6. Check if clustering is **good enough**; return to (2) if not

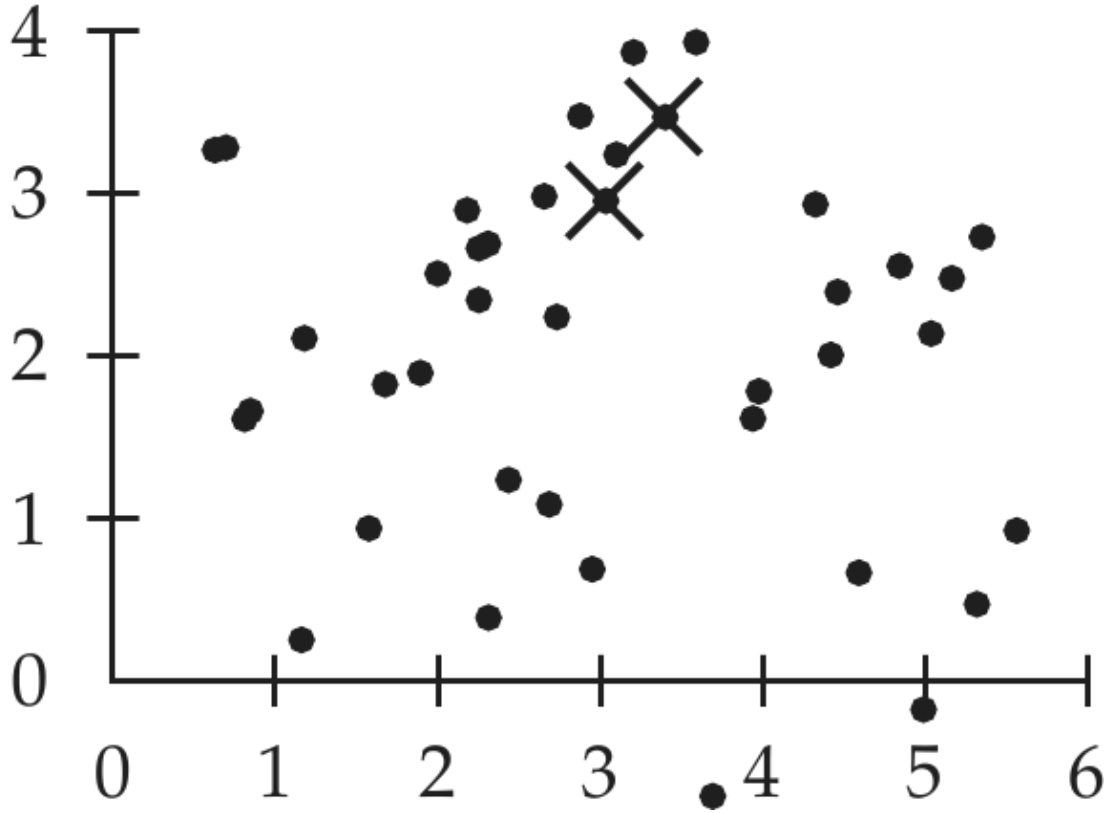
K-means clustering

- What's **good enough**?
  - **Small change** since previous iteration
  - **Maximum number** of iterations reached
  - Set a threshold for a convenient **RSS**

K-means clustering

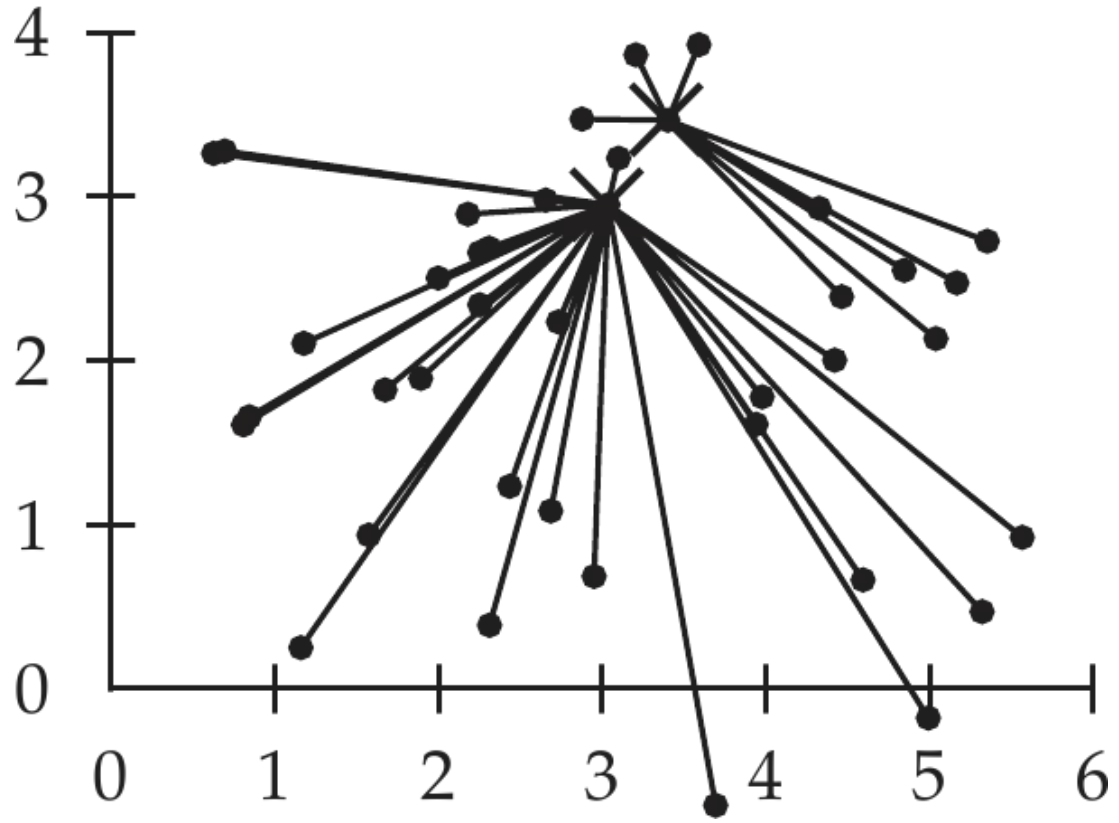
- Example from (Manning et al., 2008):

1. Randomly select k = 2 seeds (initial centroids)



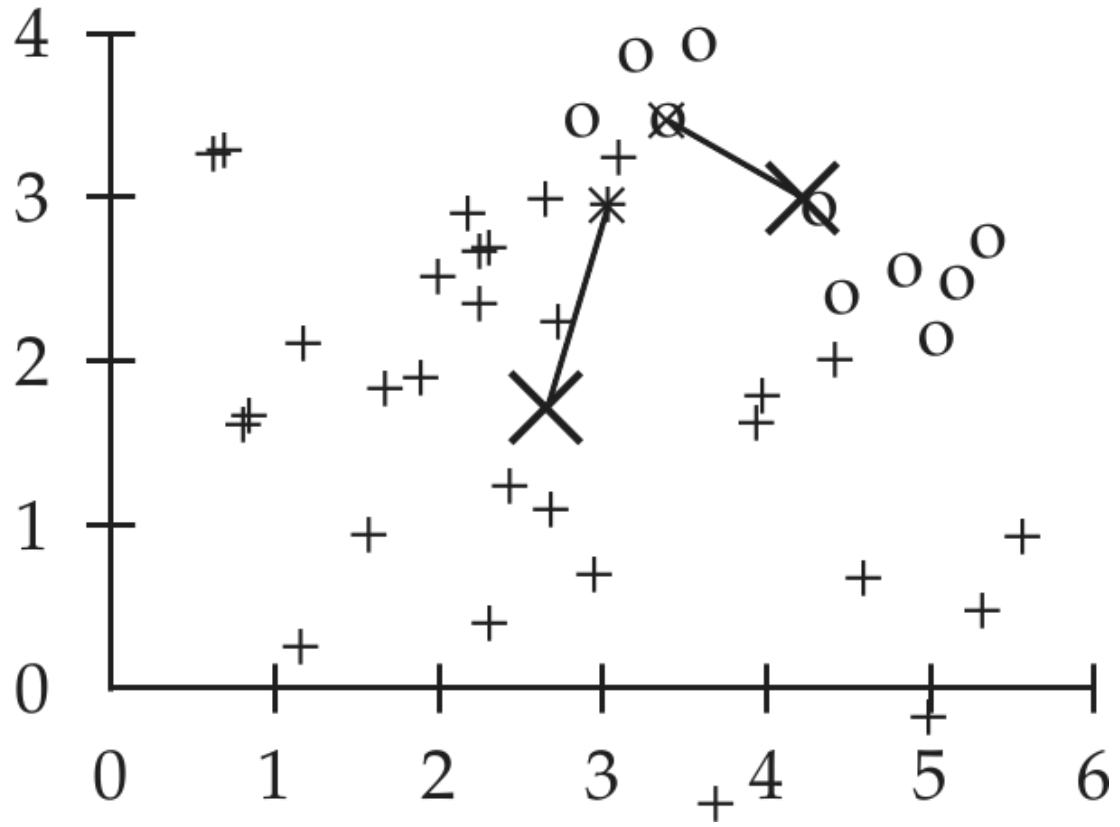
K-means clustering

4. Assign each data set to the cluster having the nearest centroid



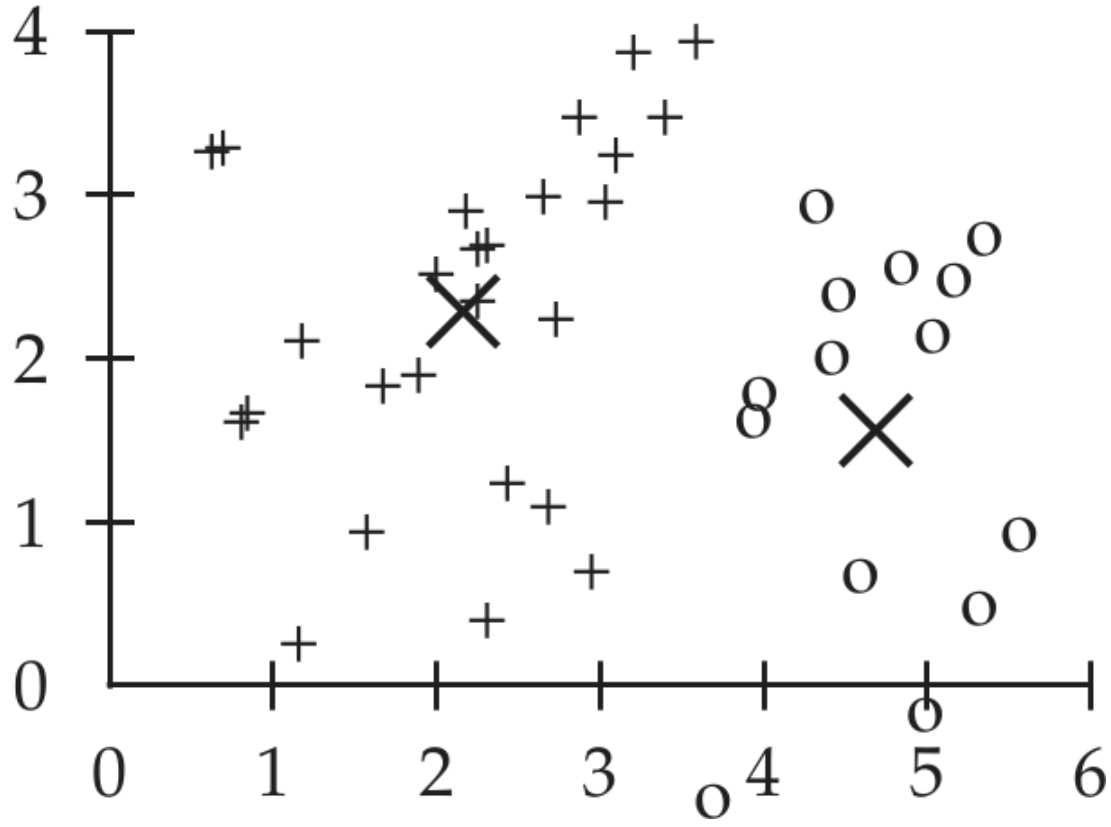
K-means clustering

5. Recompute centroids



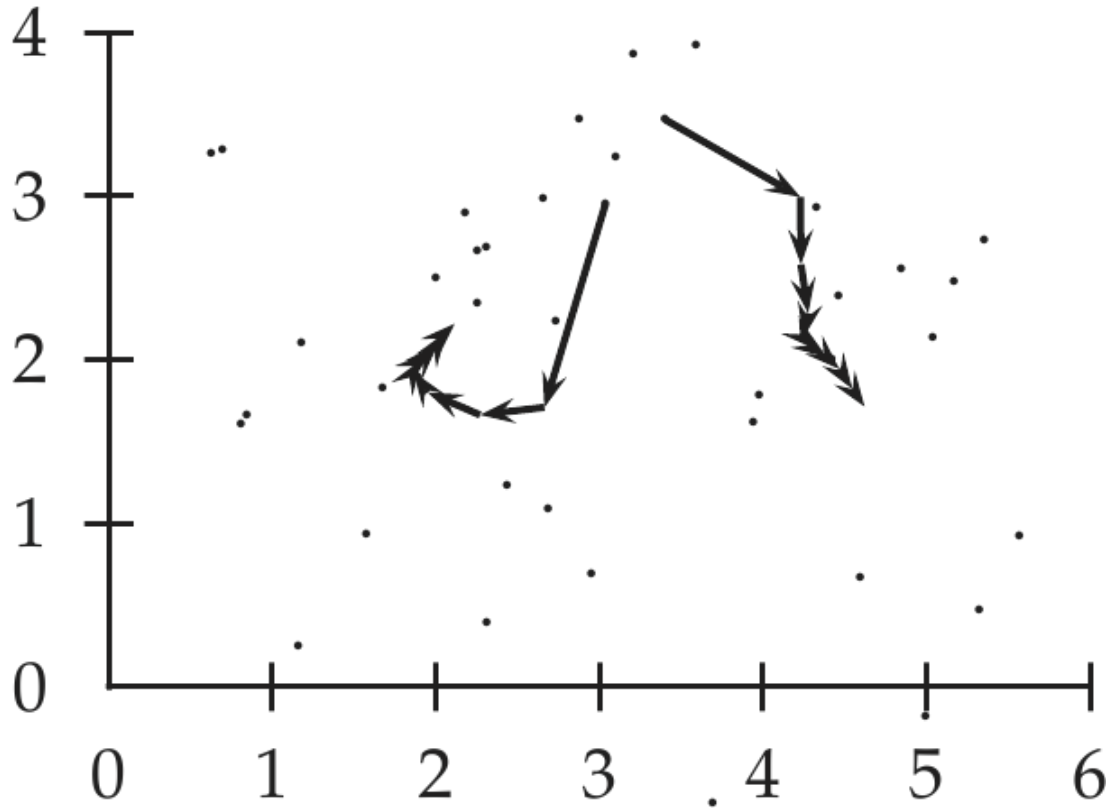
K-means clustering

Result after
9 iterations:



K-means clustering

Movement of centroids in 9 iterations:



K-means clustering

- Advantages
  - Relatively efficient: $O(nkt)$
    - n: # objects, k: # clusters, t: # iterations; k, t << n
  - Often terminates at a local optimum
- Disadvantages
  - Applicable only, when the mean is defined
  - What about **categorical data**?
  - Need to specify the **number of clusters**
  - Unable to handle noisy data and **outliers**
  - Unsuitable to discover **non-convex clusters**

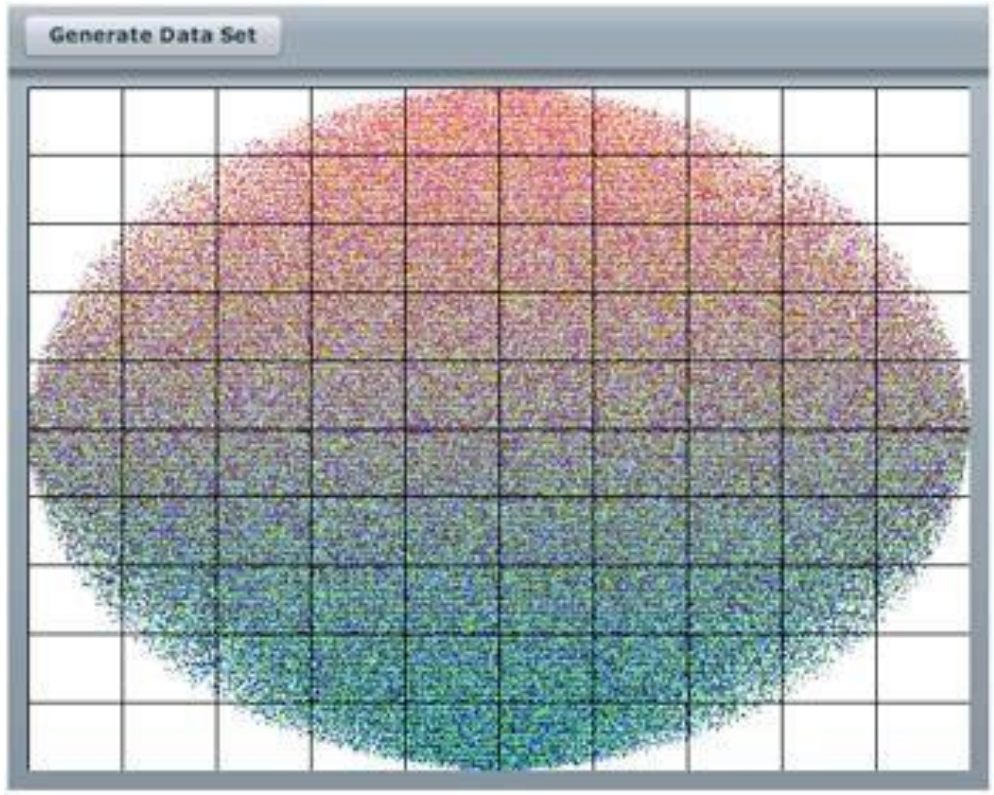
K-means clustering

- Similar approaches:
  - **K-medoids:** like k-means, but use tuples lying closest to the centroid instead of centroid
  - **Fuzzy c-means:** similar to k-means but soft clustering
  - **Model-based clustering:** Assume that data has been generated randomly around k unknown "source points"; find the k points that most likely have generated the observed data **(maximum likelihood)**

Clustering in Data Warehouse

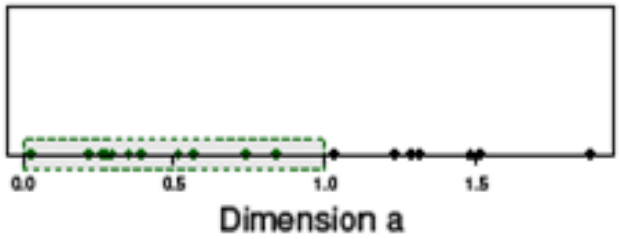
Clustering in DW

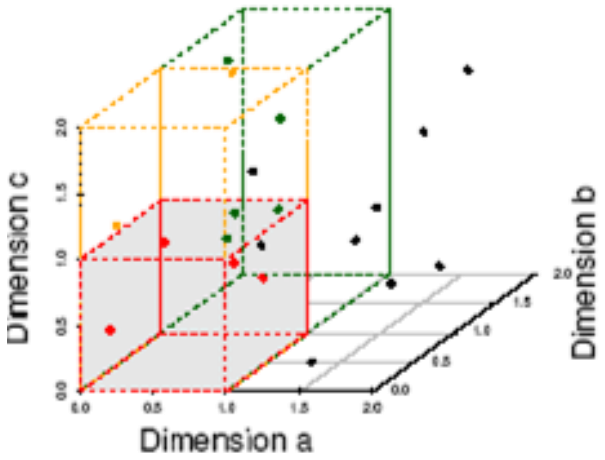
- Clustering in DW
  - High data dimensionality
  - Large data sets

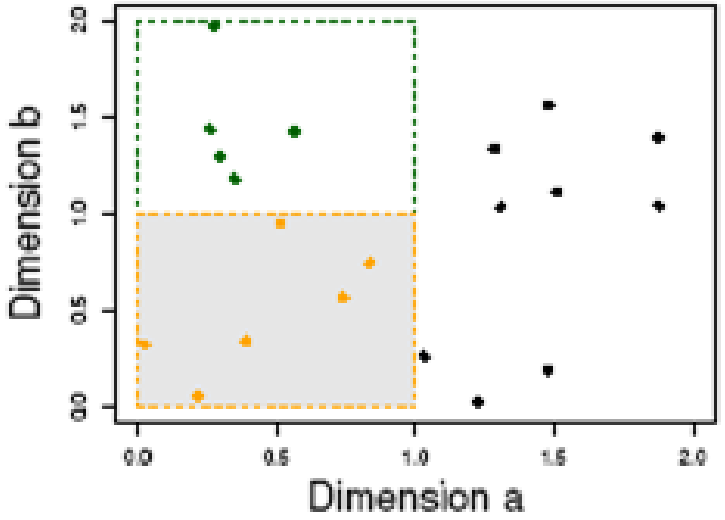


Clustering in DW

- **Major challenges** in clustering high-dimensional data
  - Many irrelevant dimensions
  - Clusters may exist only in some subspaces







Clustering in DW

- Handling high-dimensional data
  - Feature transformation: only effective if most dimensions are relevant
    - Singular Value Decomposition: useful only when features are highly correlated/redundant
  - Subspace-clustering: find clusters in all the possible subspaces
    - CLIQUE, ProClus, and frequent pattern-based clustering

CLIQUE

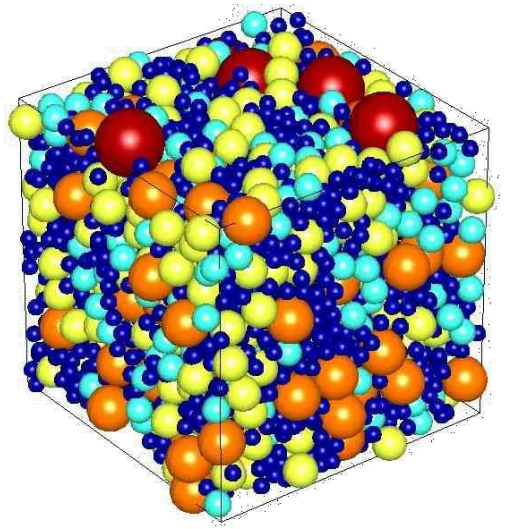
- Clustering in QUEST (**CLIQUE**)
  - Automatically identify those **subspaces** of a high dimensional data space that allow better clustering than the original space
  - CLIQUE is both density- and grid-based
    - It partitions **each dimension** into the same number of equal length intervals: a grid structure

CLIQUE

- A unit is **dense** if the fraction of total data points contained in the unit exceeds the input model parameter
- A **cluster** is a maximal set of **connected dense units** within a subspace
- Two units are **connected** if they have 'a common face' (i.e. they are adjacent) or if there is a third unit having a common face with each of them

CLIQUE

- A-priori principle in CLIQUE
  - If k-dimensional unit is dense then so are its projections in (k-1)-dimensional space
  - Therefore, if one of the (k-1)-dimensional projections of a k-dimensional unit is **not dense**, we can prune the k-dimensional unit,
    since it cannot be dense

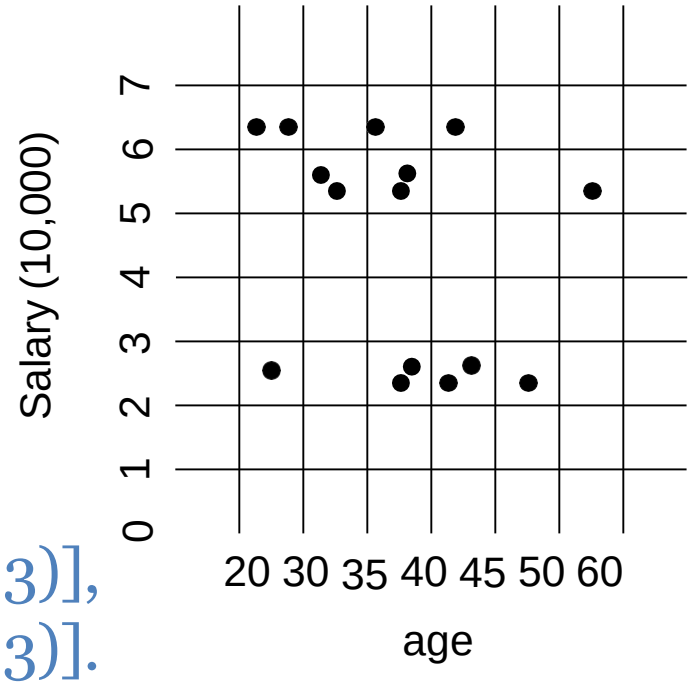


CLIQUE

- Step 1: **identification of subspaces** that contain clusters
  - Find dense units in different subspaces
    - Proceed level by level
    - Start with 1-dimensional subspace, and build higher-dimensional subspaces with dense units
    - Generate k-dimensional candidates, from the k-1 dense units

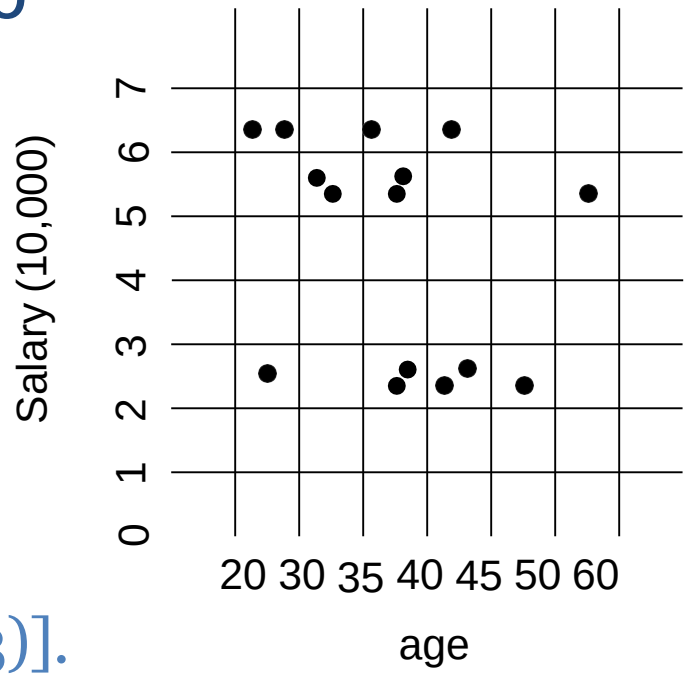
CLIQUE

- Example: density parameter 2 elements
  - Dense units in 1dimensional Space:
    - On X: (20;30), (30;35), (35;40), (40;45)
    - On Y: (2;3), (5;6), (6;7)
  - Build 2D candidates:
    - Build the 12 combinations
    - Read the data, and eliminate 2D non dense units
    - Result: [(20;30), (6,7)], [(30;35), (5,6)], [(35;40), (2,3)], [(35;40), (5,6)], [(40;45), (2,3)].



CLIQUE

- Step 2: identification of clusters
  - Input: the set of dense units U of the same subspace
  - Output: partition U into $U_1 \ldots U_q$ such that all units in U_i, $1 \leq i \leq q$ are connected and no two units belonging to different partitions are connected
  - Depth-first search algorithm
  - Result:
    - U_1: [(20;30), (6,7)], [(30;35), (5,6)], [(35;40), (5,6)].
      U_2: [(35;40), (2,3)], [(40;45), (2,3)].

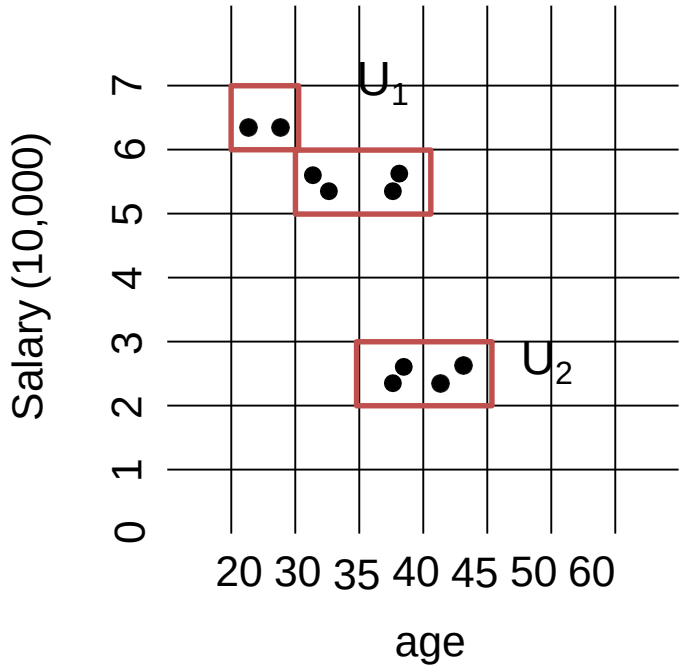


CLIQUE

- Step 3: Generation of minimal description for each of the clusters
  - Take U_1: [(20;30),(6,7)], [(30;35),(5,6)], [(35;40),(5,6)] and U_2: [(35;40), (2,3)], [(40;45), (2,3)] as input
  - Generate a concise description of the clusters
  - Problem: cover all units with the minimum number of regions (rectangles only containing connected units)
    - NP hard
  - Solution: **greedy** algorithm

CLIQUE

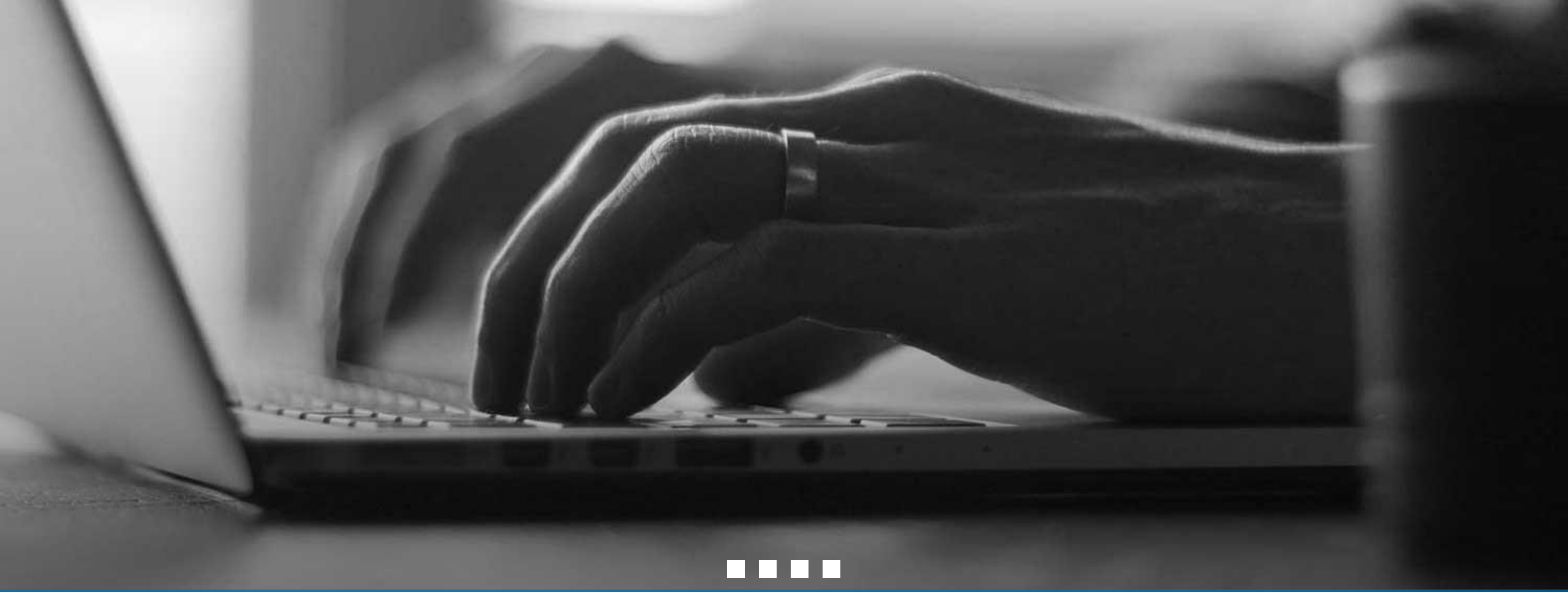
- Minimum Coverage: greedy algorithm
  - Start with U_1, and take a random seed
  - From the seed, grow a rectangle in all directions covering only units from U_1
  - Continue with not covered units from U_1
  - Repeat the process for U_2



CLIQUE

- Strength
  - Automatically finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
  - Insensitive to the order of records in input and does not presume some canonical data distribution
  - Scales linearly with the size of input and has good scalability as the number of dimensions in the data increases
- Weakness
  - The accuracy of the clustering result may be degraded at the expense of simplicity of the method


QA ???



THANK YOU

Munawar, PhD – moenawar@gmail.com – www.moenawar.web.id