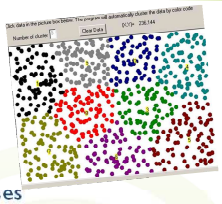
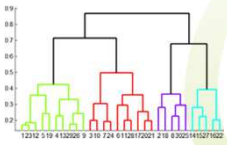
ifis
Institut für Informationssysteme
Technische Universität Braunschweig

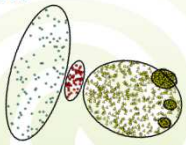
Data Warehousing & Data Mining

Wolf-Tilo Balke
Kinda El Maarry
Institut für Informationssysteme
Technische Universität Braunschweig
http://www.ifis.cs.tu-bs.de

12. Data Mining







12.0 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**

12.0 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

12.0 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

12.0 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

12.0 Cluster Analysis

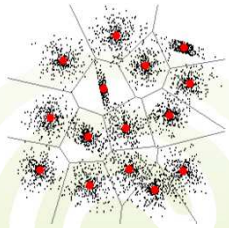
- **Requirements** of cluster analysis
  - Scalability
    - Highly scalable algorithms are needed for clustering on large data sets
  - Ability to deal with different types of attributes
    - Clustering may be performed also on binary, categorical and ordinal data
  - Discovery of clusters with arbitrary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

12.0 Requirements

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

12.0 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?

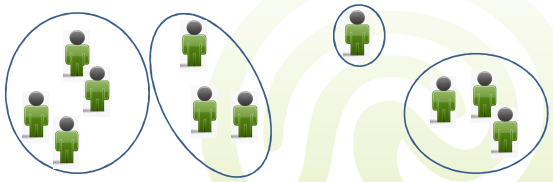


12.0 Issues in clustering

- **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…

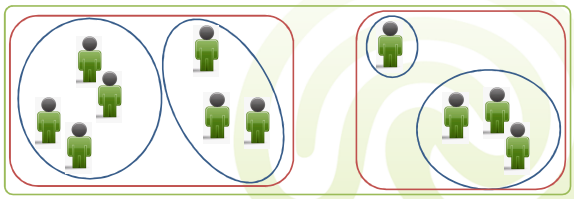
12.0 Issues in clustering

- 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



12.0 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



12.0 Issues in clustering

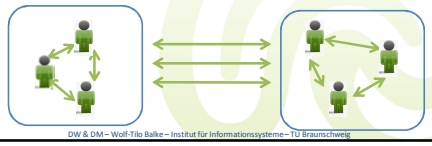
- Hard or soft?
  - **Hard clustering:**
    - Every item is assigned to exactly one cluster (at the lowest level, if the clustering is hierarchical)
    - More common and easier to do
  - **Soft clustering:**
    - An items assignment is a **distribution** over all clusters (fuzzy, probabilistically,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

12.0 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!

12.0 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



12.0 Issues in clustering

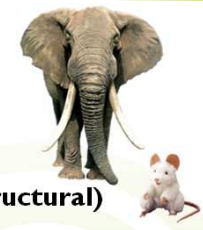
- Inter-cluster similarity and intra-cluster similarity:

BAD:

GOOD:

12.0 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 (later)



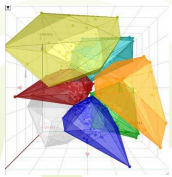
12.0 Issues in clustering

- 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…

12.1 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



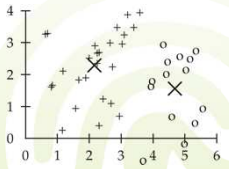
12.1 Flat Clustering

- **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!

12.1 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:

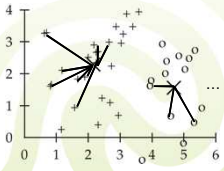
$$\mu(A) = \frac{1}{m}\sum_{i=1}^{m} d_i$$



12.1 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

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

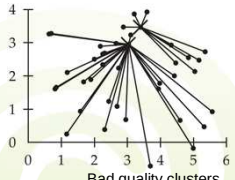


12.1 K-means clustering

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

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

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



Bad quality clusters

12.1 K-means clustering

- The **k-means algorithm** (aka Lloyd's algorithm):
  1. Randomly select k data points (items) as **seeds** (= initial centroids)
  2. Create k **empty clusters**
  3. Assign exactly one centroid to each cluster
  4. Iterate over the whole data points: assign each data point to the cluster with the nearest centroid
  5. Recompute cluster centroids based on contained data points
  6. Check if clustering is **good enough**; return to (2) if not

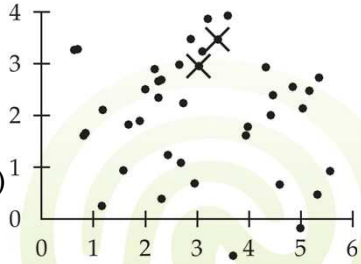
12.1 K-means clustering

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

12.1 K-means clustering

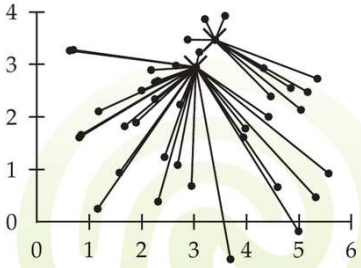
- Example from (Manning et al., 2008):

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



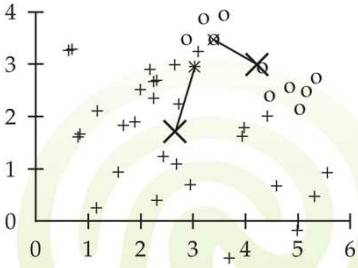
12.1 K-means clustering

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



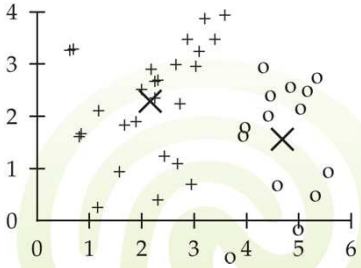
12.1 K-means clustering

5. Recompute centroids



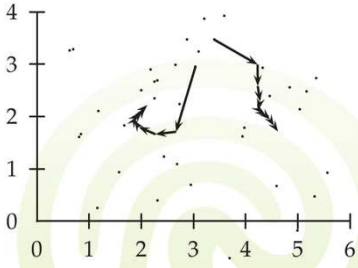
12.1 K-means clustering

Result after 9 iterations:



12.1 K-means clustering

Movement of centroids in 9 iterations:



12.1 K-means clustering

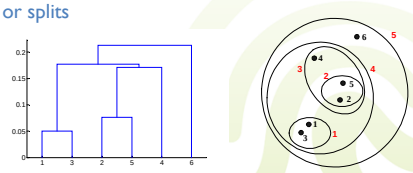
- Advantages
  - Relatively efficient: $O(nkt)$
    - n: # objects, k: # clusters, t: # iterations; k, t << n
  - Often terminates at a local optimum
- Disadvantages
  - Applicable only, when the mean is defined
  - What about **categorical data**?
  - Need to specify the **number of clusters**
  - Unable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

12.1 K-means clustering

- Similar approaches:
  - **K-medoids:** like k-means, but use document 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)**

12.2 Hierarchical Clustering

- 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



12.2 Hierarchical Clustering

- Hierarchical clustering
  - **Agglomerative** (bottom-up)
    - Start with individual items as initial clustering, create parent clusters by **merging**
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left

12.2 Hierarchical Clustering

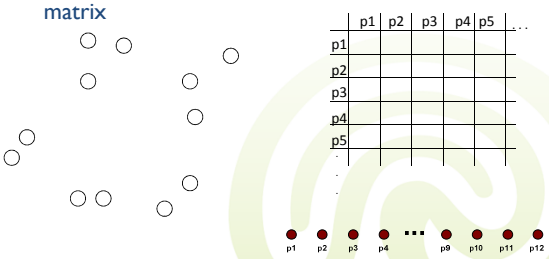
- Hierarchical clustering
  - **Divisive** (top-down)
    - Start with an initial large cluster containing all items, create child clusters by **splitting**
    - At each step, split a cluster until each cluster contains a single point (or there are k clusters)

12.2 Hierarchical Clustering

- **Agglomerative clustering**
  - Assume that we have some measure of similarity between clusters
  - A simple agglomerative clustering algorithm:
    1. For each data point: create a new cluster containing only this data point
    2. Compute the similarity between every pair of clusters (if there are m clusters, we get an m × m **similarity matrix**)
    3. **Merge** the two clusters having **maximal similarity**
    4. If there is more than one cluster left, go back to (2)
- Key operation is the computation of the **proximity** of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

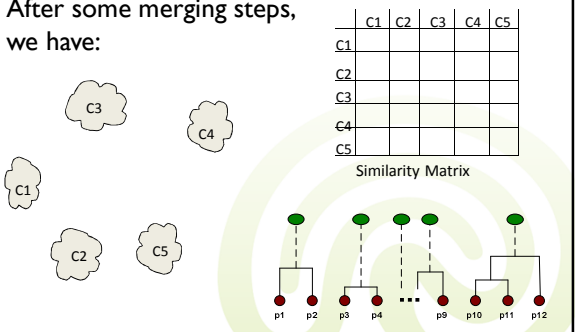
12.2 Agglomerative Clustering

- Starting situation
  - Start with clusters of individual points and a similarity matrix



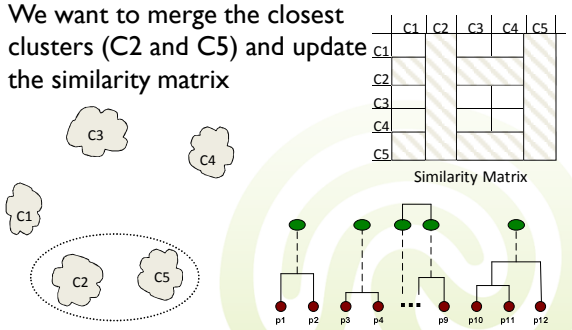
12.2 Agglomerative Clustering

- After some merging steps, we have:



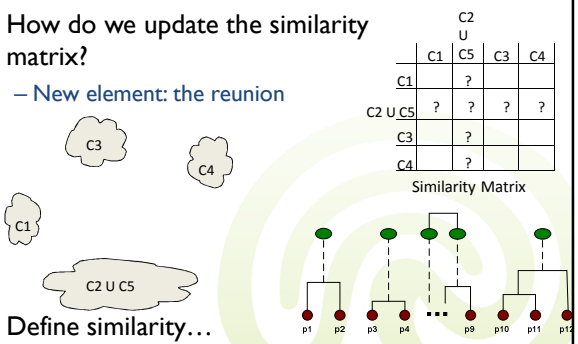
12.2 Agglomerative Clustering

- We want to merge the closest clusters (C2 and C5) and update the similarity matrix



12.2 Agglomerative Clustering

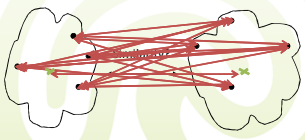
- How do we update the similarity matrix?
  - New element: the reunion



- Define similarity…

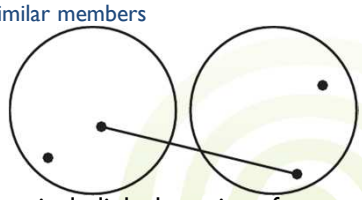
12.2 Agglomerative Clustering

- Inter-cluster **similarity**
  - Single-link clustering (MIN)
  - Complete-link clustering (MAX)
  - Group average
  - Distance between centroids
  - …



12.2 Agglomerative Clustering

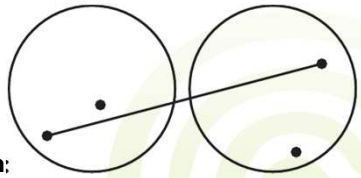
- Single-link similarity (MIN)
  - Similarity of two clusters represents similarity of their most similar members



- **Problem:** single-link clustering often produces **long chains**

12.2 Agglomerative Clustering

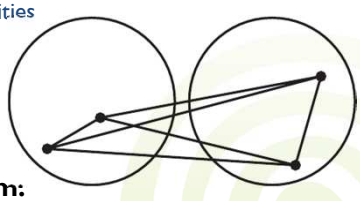
- Complete-linkage similarity (MAX)
  - Similarity of two clusters represents similarity of their most dissimilar members



- **Problem**: complete-link clustering is sensitive to outliers

12.2 Agglomerative Clustering

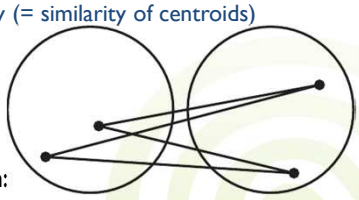
- Group average clustering
  - Similarity of two clusters represents average of all similarities



- **Problem:** computation is expensive

12.2 Agglomerative Clustering

- Centroid clustering
  - Similarity of two clusters represents average inter-similarity (= similarity of centroids)



- **Problem**: similarity to other clusters can improve by merging (leads to overlaps in dendrogram)

12.2 Agglomerative Clustering

- **Divisive clustering**
  - How does **divisive clustering** work?
  - We won't go into details here
  - But there is a simple method:
    - Use a flat clustering algorithm as a subroutine to split up clusters (e.g. 2-means clustering)
  - Again, there might be **constraints** on clustering quality:
    - Avoid very small clusters
    - Avoid splitting into clusters of extremely different cardinalities

12.3 Outlier Analysis Detour

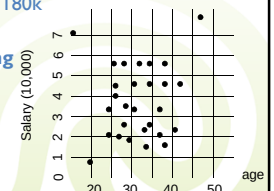
- Outlier analysis
  - Often there exist data objects that do not comply with the **general behavior** of the data
  - Such data which are grossly different from, or inconsistent with the remaining data are called **outliers**

12.3 Outlier Analysis Detour

- Sources of outliers
  - Correct data variability
    - E.g., the salary of a CEO could stand out as an outlier among other salaries in the company
  - Bad data
    - E.g., persons age is 999
- Outliers can dramatically affect analysis resulting in erroneous interpretations

12.3 Outlier Analysis Detour

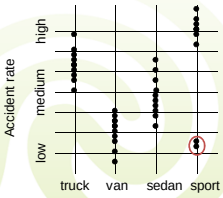
- Why are outliers important?
  - Knowledge generated from databases can be divided into three categories
    - **Incorrect** e.g., 10 years old CTO
    - **Useless** e.g., our CEO earns 180k a year
    - **New, surprising, interesting** e.g., hire lots of students because they are cheap



12.3 Outlier Analysis Detour

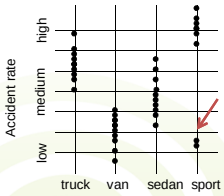
- Niche detection
  - E.g., Farmers Insurance Group

Farmers Insurance found a previously unnoticed niche of sports car enthusiasts: married boomers with a couple of kids and a second family car, maybe a minivan, parked in the driveway. Claim rates among these customers were much lower than other sports car drivers, yet they were paying the same surcharges. Farmers relaxed its underwriting rules and cut rates on certain sports cars for people who fit the profile



12.3 Outlier Detection Detour

- Detecting outliers seems easy: just **visualize** the data and here they are…
  - What about when dealing with large data sets and multiple dimensions as it is the case in DW?
    - Car types, accident rates, age, marital status, children, financial status

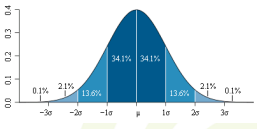


12.3 Outlier Detection Detour

- Automatic outlier detection
  - Based on the point of view
    - Outliers as points which **do not lie in clusters**
    - Outliers as points which behave very **differently from norm**
  - Methods
    - Statistical approaches
    - Distance-based approaches
    - Deviation-based approaches

12.3 Outlier Detection Detour

- **Statistical** approaches
  - Assume a model for the data set e.g., normal distribution



  - Drawbacks
    - Most tests are for one attribute
    - In many cases, the data distribution is unknown

12.3 Outlier Detection Detour

- **Distance-based** approaches
  - We need **multi-dimensional** analysis without knowing **data distribution**
  - Distance-based outlier
    - An object is an outlier if it doesn't have enough **neighbors**
    - Neighbors are defined based on the **distance** from self
  - And there are different algorithms for mining distance-based outliers e.g., index-based, nested-loop, cell-based algorithm, …

12.3 Outlier Detection Detour

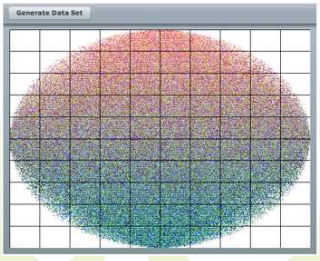
- **Deviation-based** approaches
  - Identifies outliers by examining the main characteristics of objects in a group
    - Objects that "deviate" from this description are considered outliers
  - OLAP data cube technique
    - Uses data cubes to identify regions of anomalies in large multidimensional data

12.3 Outlier Detection Detour

- OLAP data cube technique
  - A cell is an outlier if the measure (aggregate) of the cell differs significantly from its **expected value**
  - The expected value is calculated based on a statistical model e.g., regression analysis
  - If the difference between the actual value and its expected value is greater than 2.5 standard deviation, the cell is an outlier
    - OLAP version of the 3σ rule

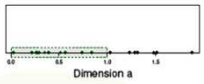
12.4 Clustering in DW

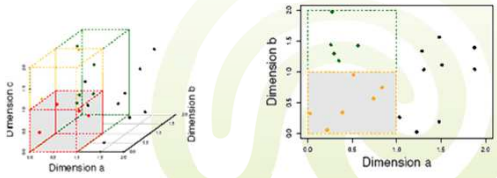
- Clustering in DW
  - **High data dimensionality**
  - **Large data sets**



12.4 Clustering in DW

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





12.4 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

12.4 CLIQUE

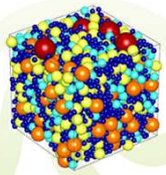
- Clustering in QUEST (**CLIQUE**)
  - Automatically identify those **subspaces** of a high dimensional data space that allow better clustering than the original space
  - CLIQUE is both density- and grid-based
    - It partitions **each dimension** into the same number of equal length intervals: a grid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

12.4 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

12.4 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

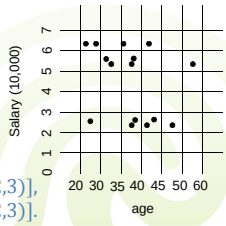


12.4 CLIQUE

- 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

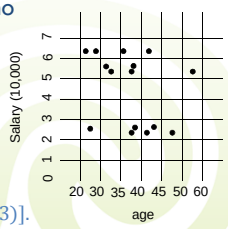
12.4 CLIQUE

- 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)].



12.4 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)].
    -

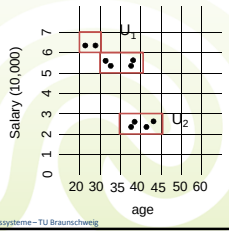


12.4 CLIQUE

- 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

12.4 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



12.4 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

Summary

- Clustering
  - Flat: K-means
  - Hierarchical: Agglomerative, Divisive
- Outlier Analysis
- Clustering high-dimensional data
  - CLIQUE

Next lecture

- Advanced Classification Algorithms
  - Bootstrap Aggregating
  - Boosting
  - Adaptive Boosting