C | A | U Christian-Albrechts-Universität zu Kiel Technische Fakultät

Inf-KDDM: Knowledge Discovery and Data Mining

Winter Term 2019/20

Lecture 6: Clustering

Lectures: Prof. Dr. Matthias Renz

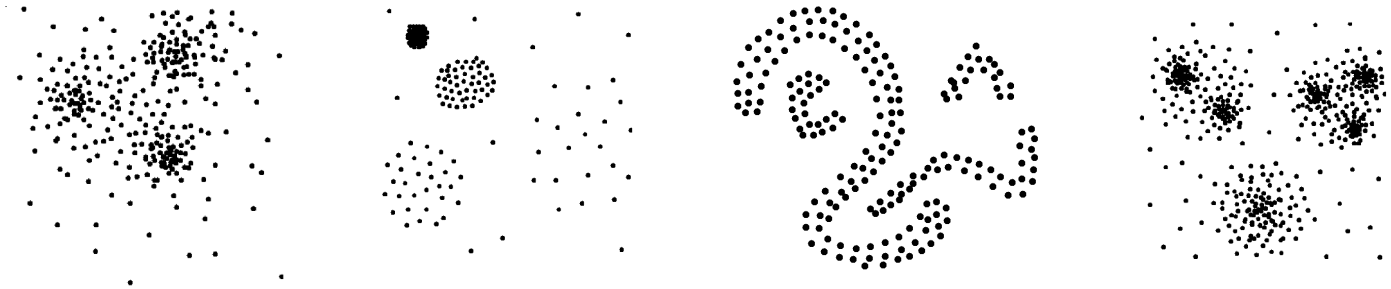
Exercises: Christian Beth

Outline

- Unsupervised learning vs supervised learning
- A categorization of major clustering methods
- Partitioning-based clustering

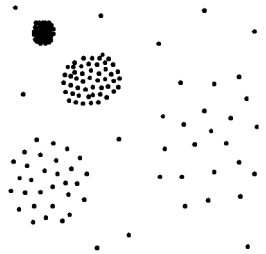
What is cluster analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters



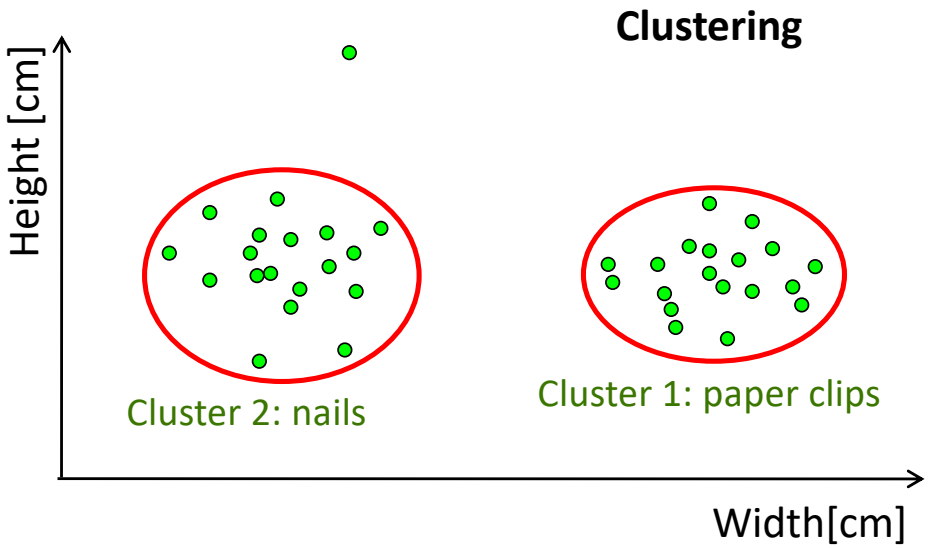
An unsupervised learning task

- Clustering is an unsupervised learning task
  - Given a set of measurements, observations, etc., the goal is to group the data into groups of similar data (clusters)
  - We are given a dataset as input which we want to cluster but there are no class labels
  - We don't know how many clusters exist in the data
  - We don't know the characteristics of the individual clusters



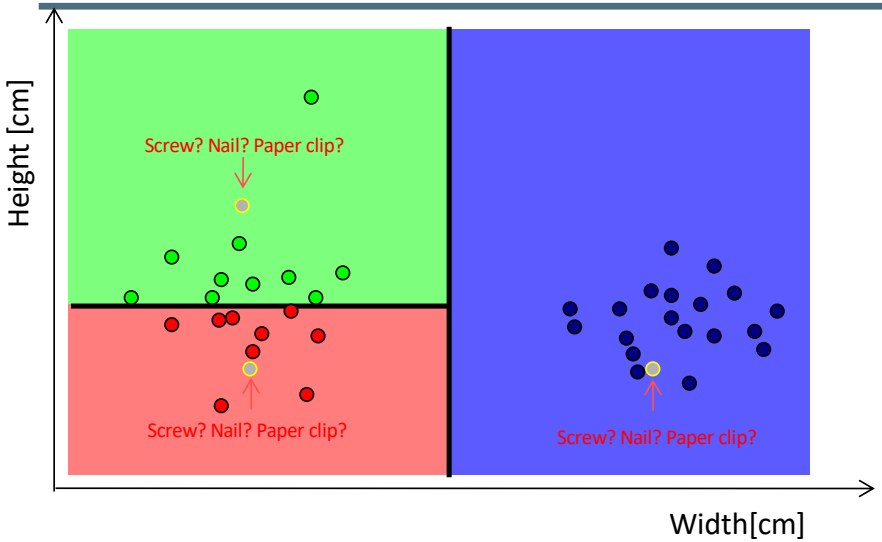
- In contrast to classification, which is a supervised learning task
  - Supervision: The training data (observations, measurements, etc.) are accompanied by *labels* indicating the *class* of the observations
  - New data is classified based on the training set

Unsupervised learning example



Question:
Is there any structure in data (based on their characteristics, i.e., width, height)?

Supervised learning example



Classification model

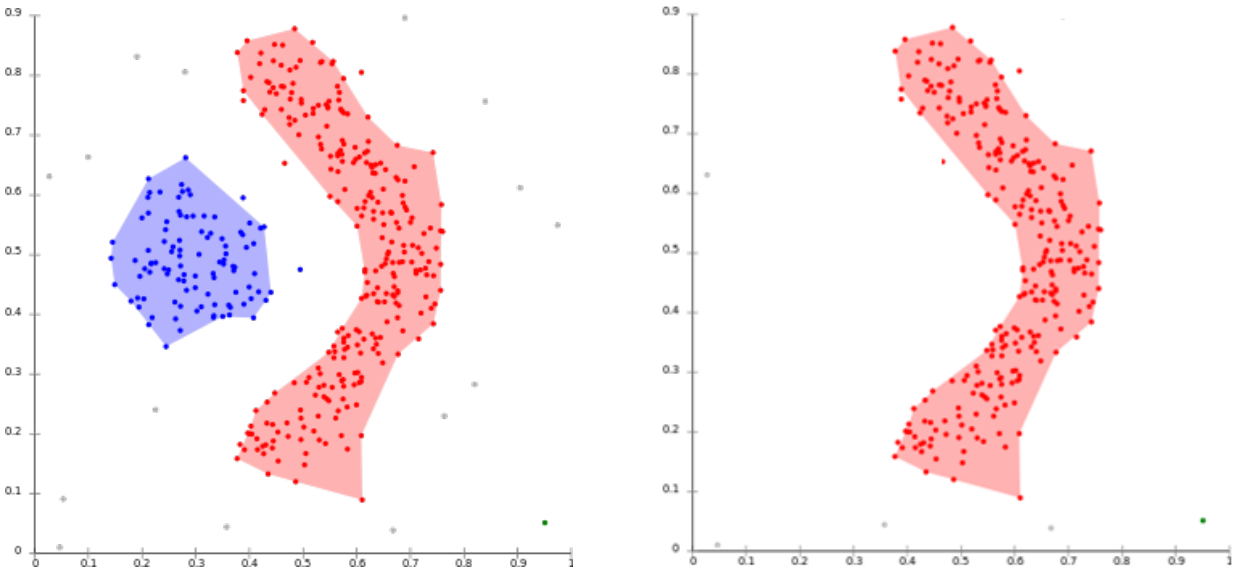
- Screw 
- Nails 
- Paper clips 

- New object (unknown class)

Question:
What is the class of a new object???
Screw, nail or paper clip?

Why clustering?

- Clustering is widely used as:
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms



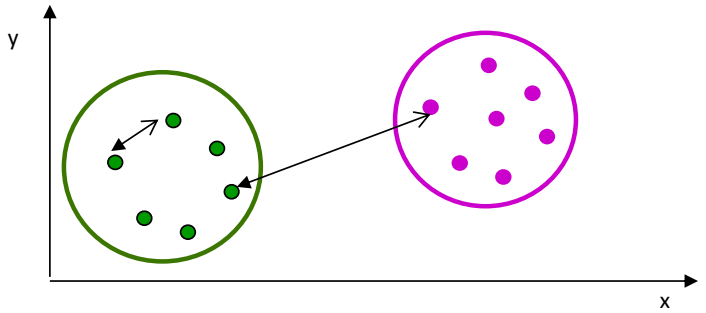
http://en.wikipedia.org/wiki/Cluster_analysis

Example applications

- Marketing:
  - Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Telecommunications:
  - Build user profiles based on usage and demographics and define profile specific tariffs and offers
- Land use:
  - Identification of areas of similar land use in an earth observation database
- City-planning:
  - Identifying groups of houses according to their house type, value, and geographical location
- Bioinformatics:
  - Cluster similar proteins together (similarity wrt chemical structure and/or functionality etc)
- Web:
  - Cluster users based on their browsing behavior
  - Cluster pages based on their content (e.g. News aggregators)

The clustering task

- **Goal:** Group objects into groups so that the objects belonging in the same group are similar (high intra-cluster similarity), whereas objects in different groups are different (low inter-cluster similarity)
- A good clustering method will produce high quality clusters with
  - high intra-cluster similarity
  - low inter-cluster similarity



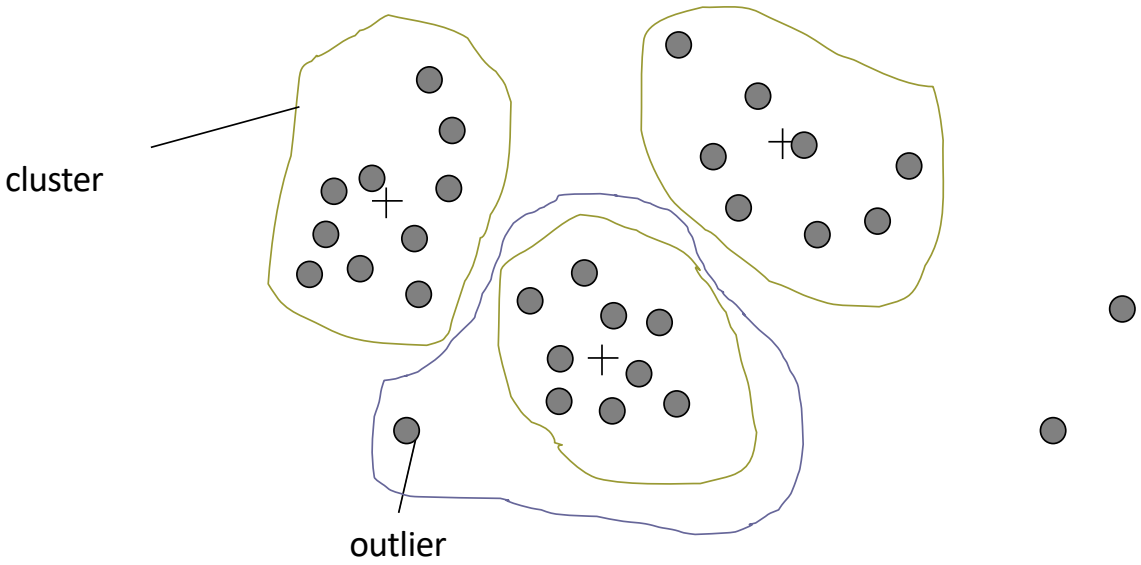
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation

Requirements for clustering

- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Incorporation of user-specified constraints
- Interpretability and usability
- Insensitive to order of input records
- Scalability
- Ability to deal with different types of attributes
- Ability to handle dynamic data
- High dimensionality

Outliers

- There might be objects that do not belong to any cluster



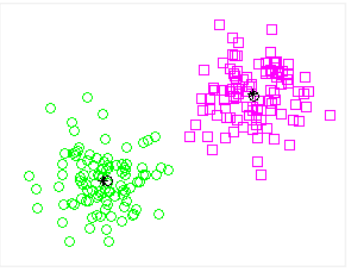
- There are cases where we are interested in detecting outliers not clusters
- Outlier analysis is related to clustering but considered as a different problem!

Outline

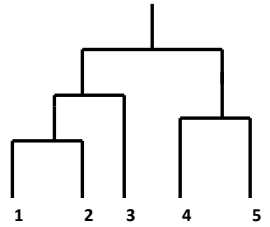
- Unsupervised learning vs supervised learning
- A categorization of major clustering methods
- Partitioning-based clustering

Major clustering methods 1/2

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS



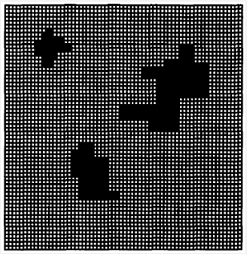
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CHAMELEON



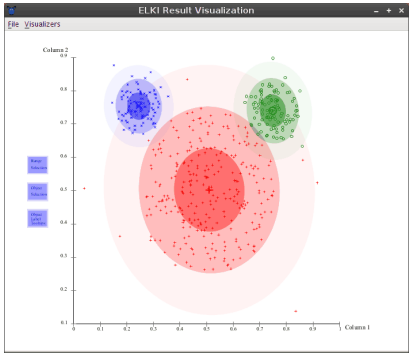
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue

Major clustering methods 2/2

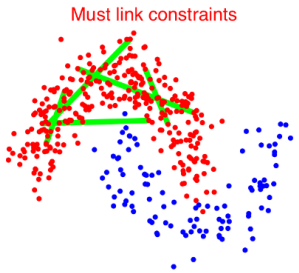
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE



- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: EM, SOM, COBWEB



- Frequent pattern-based:
  - Based on the analysis of frequent patterns
  - Typical methods: pCluster
- User-guided or constraint-based:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering



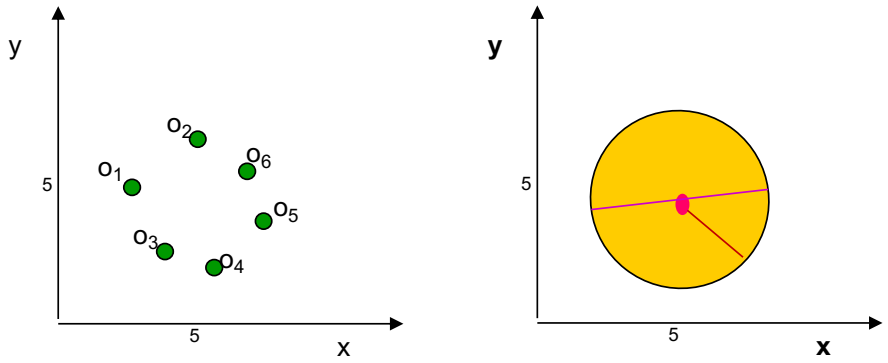
Cluster descriptors (numerical data)

- Centroid: the "middle" of a cluster
- Radius: square root of average distance from any point of the cluster to its centroid
- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$c_m = \frac{\sum_{i=1}^{n} p_i}{n}$$

$$r_m = \sqrt{\frac{\sum_{i=1}^{n} (p_i - c_m)^2}{n}}$$

$$d_m = \sqrt{\frac{\sum_{i=1}^{n} \sum_{i=1}^{n} (p_i - p_j)^2}{n(n-1)}}$$



Outline

- Unsupervised learning vs supervised learning
- A categorization of major clustering methods
- Partitioning-based clustering

Partitioning methods idea

- Construct a partition of a database *D* of *n* objects into a set of *k* clusters
  - Each object belongs to exactly one cluster (*hard* or *crisp* clustering)
  - The number of clusters *k* is given in advance
- The partition should optimize the chosen partitioning criterion
  - e.g., minimize the intra-cluster variance, i.e., the sum of the squared distances from each data point to its cluster center.
  - Possible solutions:
    - Global optimal: exhaustively enumerate all partitions
    - Heuristic methods: *k*-means and *k*-medoids algorithms
    - *k*-means: Each cluster is represented by the center of the cluster
    - *k*-medoids: Each cluster is represented by one of the objects in the cluster .

The *k*-Means problem

- Given a database *D* of *n* points in a *d*-dimensional space and an integer *k*

- Task: choose a set of *k* points $\{c_1, c_2, \ldots, c_k\}$ in the *d*-dimensional space to form clusters $\{C_1, C_2, \ldots, C_k\}$ such that the clustering cost is minimized:

$$Cost(C) = \sum_{i=1}^{k} \underbrace{\sum_{x \in C_i} (x - c_i)^2}_{\text{Cluster cost}}$$

Clustering cost

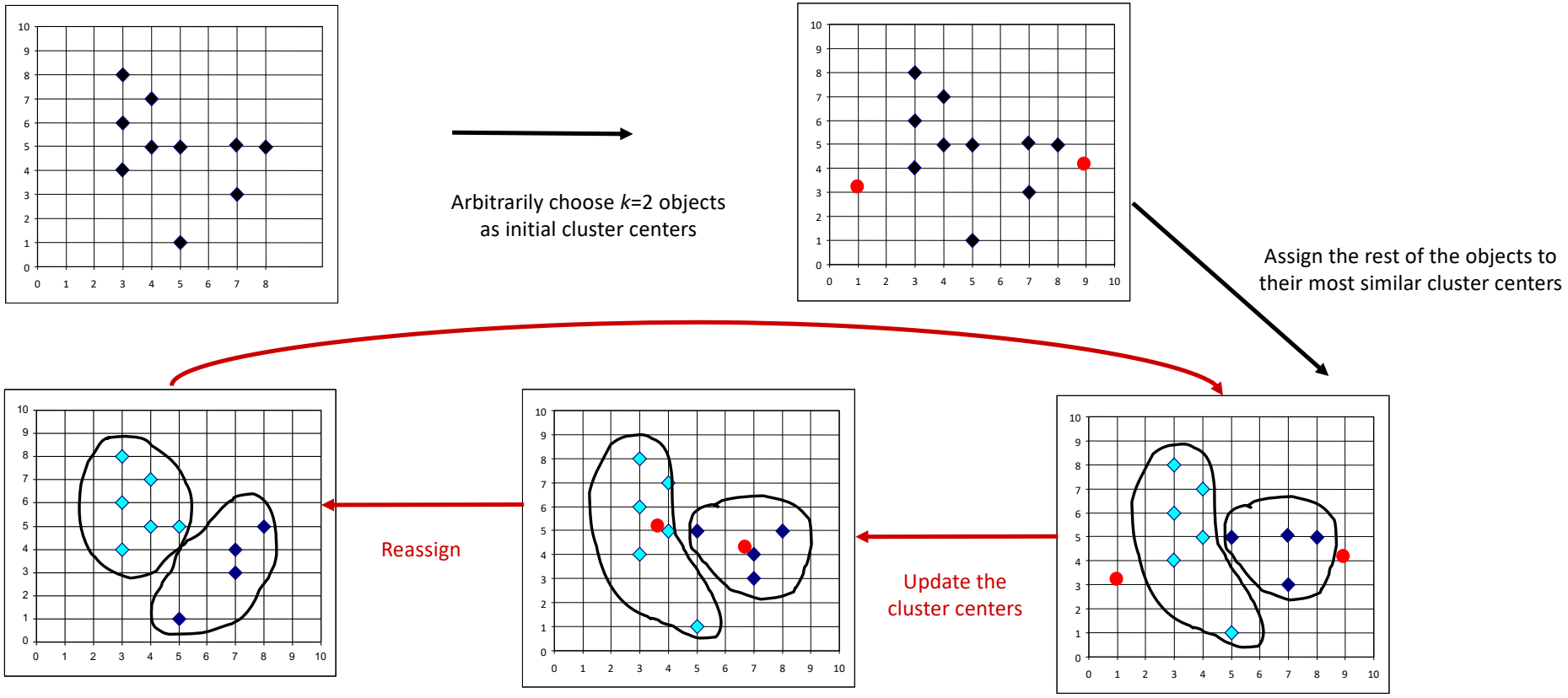
- This is an optimization problem, with the objective function to minimize the cost
- Enumerating all possible solutions and choosing the global optimum is infeasible.

The *k*-Means algorithm

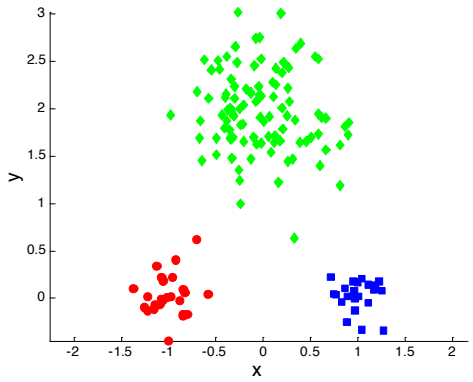
- Given *k*, the *k*-Means algorithm is implemented in four steps:
  - Randomly pick *k* objects as cluster centers $\{c_1, \ldots, c_k\}$.
  - Assign the rest of the points to their closest cluster centers.
  - Update the center of each cluster based on the new point assignments.
  - Repeat until convergence.
    - E.g., cluster centers do not change, cost is not improved significantly, after *t* iterations, etc.

- Complexity
  - Relatively efficient: $O(tkn)$, where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, $k, t << n$.

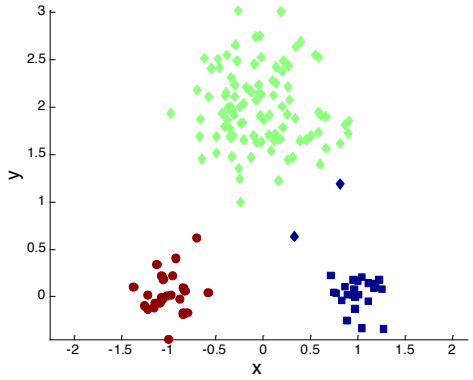
k-Means example



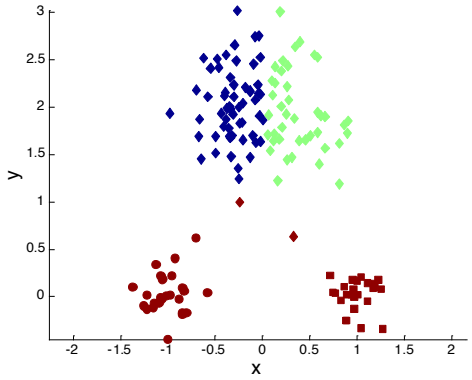
k-Means finds a local optimum



original points

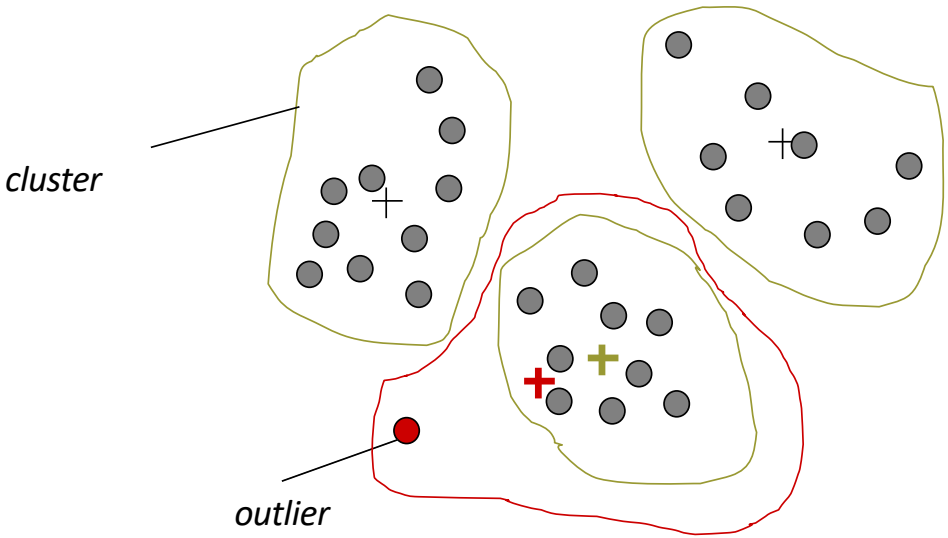


optimal clustering



sub-optimal clustering

k-Means is sensitive to outliers



k-Means variations

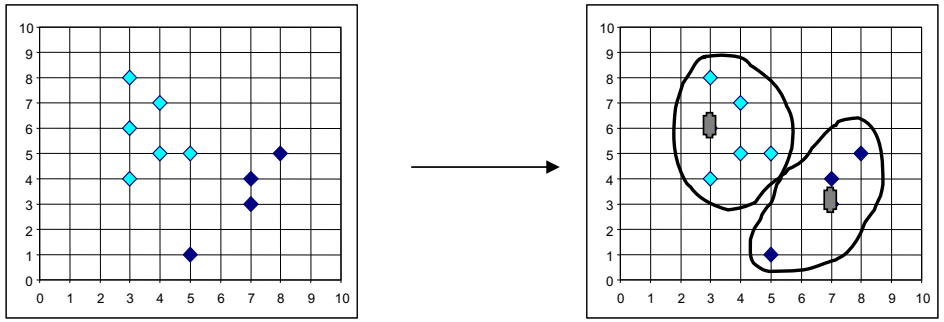
- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
    - Multiple runs
    - Not random selection of centers. e.g., pick the most distant (from each other) points as cluster centers (*k*Means++ algorithm)
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes (mode = value that occurs more often)
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters

k-Means overview

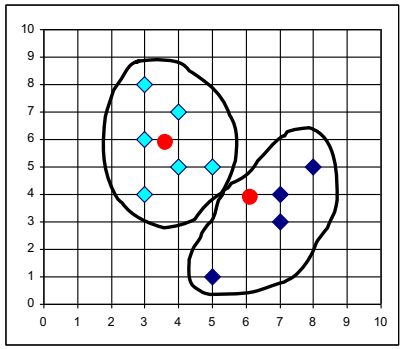
- Relatively efficient: O(*tkn*), *n*: # objects, *k*: # clusters, *t:* # iterations. Normally, *k*, *t* << *n*.
  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Finds a local optimum
- The choice of initial points can have large influence in the result
- Weaknesses
  - Need to specify *k*, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes
  - Applicable only when mean is defined, then what about categorical data?

From *k*-Means to *k*-Medoids

- The *k*-Means algorithm is sensitive to outliers!
  - an object with an extremely large value may substantially distort the distribution of the data.
- *k*-Medoids: Instead of taking the mean value of the objects in a cluster as a reference point, medoids can be used, which are the most centrally located object in the clusters.



medoid-based approach

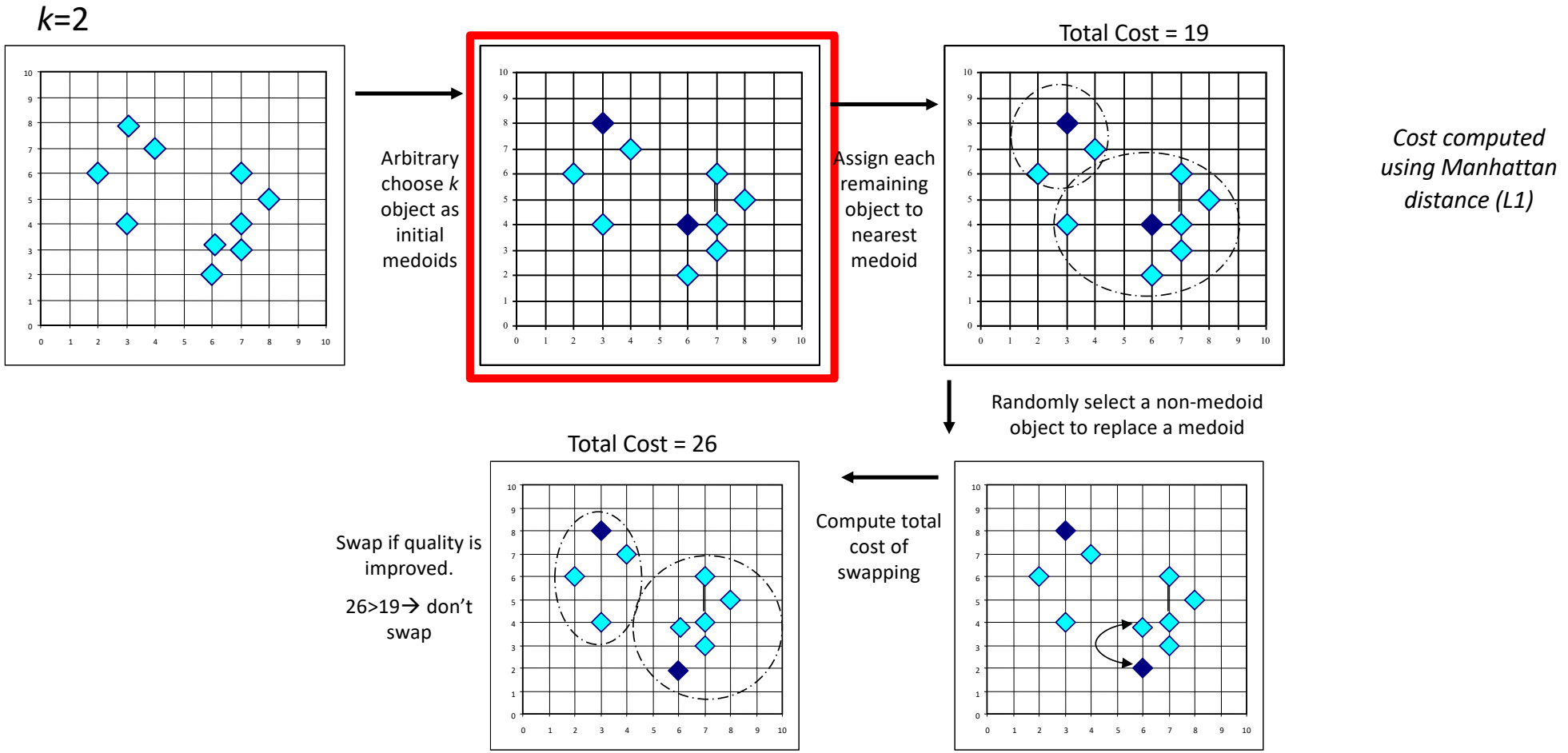


centroid-based approach

The k-Medoids clustering algorithm

- Clusters are represented by real objects called *medoids*.
- PAM (Partitioning Around Medoids, Kaufman and Rousseeuw, 1987)
  - starts from an initial set of *k* medoids and iteratively replaces one of the medoids by one of the non-medoid poins if such a replacement improves the total clustering cost
- Pseudocode:
  - Select *k* representative objects arbitrarily
  - Assign the rest of the objects to the *k* clusters
  - Representative replacement:
    - For each medoid *m* and each non-medoid object *o* do, check whether *o* could replace *m*
    - Replacement is possible if the clustering cost is improved.
  - Repeat until no improvements can be achieved by any replacement

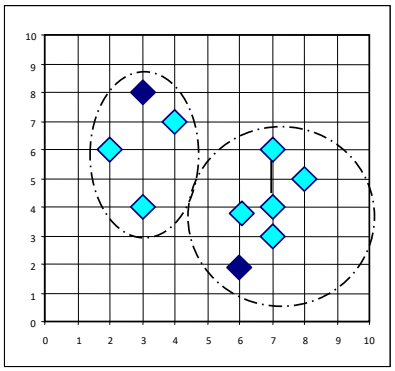
PAM example:


k=2
Arbitrary choose k object as initial medoids
Assign each remaining object to nearest medoid
Total Cost = 19
Randomly select a non-medoid object to replace a medoid
Compute total cost of swapping
Total Cost = 26
Swap if quality is improved.
26>19→ don't swap
Cost computed using Manhattan distance (L1)

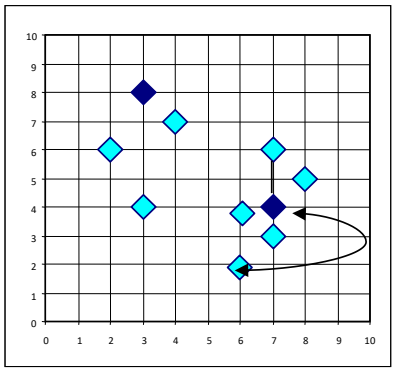
PAM example: swap case

k=2

Total Cost = 26

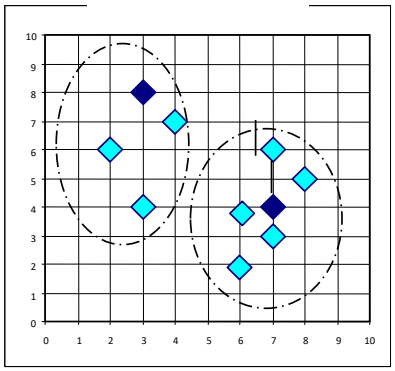


Randomly select a non-medoid object to replace a medoid



Assign each remaining object to nearest medoid

Total Cost = 18



Cost computed using Manhattan distance (L1)

Swap if quality is improved

18<26 → swap

Do loop

Until no change

PAM overview

- Very similar to *k*-Means
- PAM is more robust to outliers comparing to *k*-Means because a medoid is less influenced by outliers or other extreme values than a centroid.
- PAM works efficiently for small data sets but does not scale well for large data sets.
  - $O(k(n-k)^2)$ for each iteration

    where *n* is # of data, *k* is # of clusters

- Sampling based method:
  - CLARA(Clustering LARge Applications)
  - CLARANS ("Randomized" CLARA)

CLARA (Clustering Large Applications)

- CLARA (Kaufmann and Rousseeuw, 1990)
- It draws multiple samples of the dataset, applies PAM on each sample, and gives the best clustering as the output.
- Strength: deals with larger datasets than PAM
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole dataset if the sample is biased

What is the right number of clusters 1/2

- The number of clusters *k* is required as input by the partitioning algorithms. Choosing the right *k* is challenging.
- Silhouette coefficient (Kaufman & Rousseeuw 1990)
  - Let *a(o)* the distance of an object *o* to the representative of its cluster and *b(o)* the distance to the representatives of its "second best" cluster
  - Silhouette *s*(*o*) of an object *o:*

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

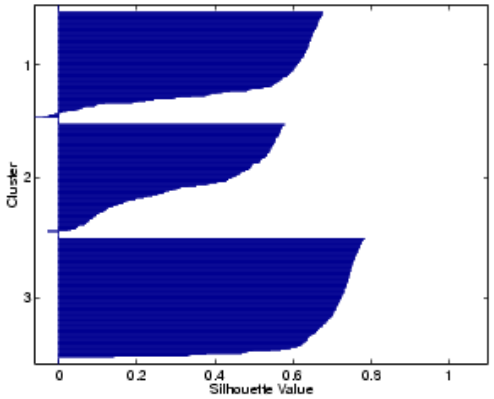
-1 ≤ s(o) ≤ +1

s(o) ~ -1 / 0 / +1 : bad / indifferent / good assignment

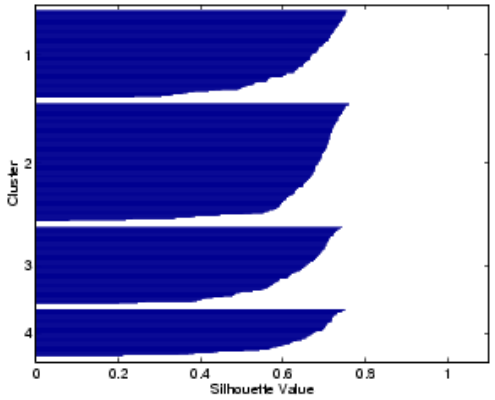
  - s(o)~1→ *a*(*o*) < < *b*(*o*). Small *a*(*o*) means it is well matched to its own cluster. Large b(o) means is badly matched to its neighbouring cluster.
  - s(o)~-1→ the neighbor cluster seems more appropriate
  - s(o)~0→ in the border between two natural clusters

What is the right number of clusters 2/2

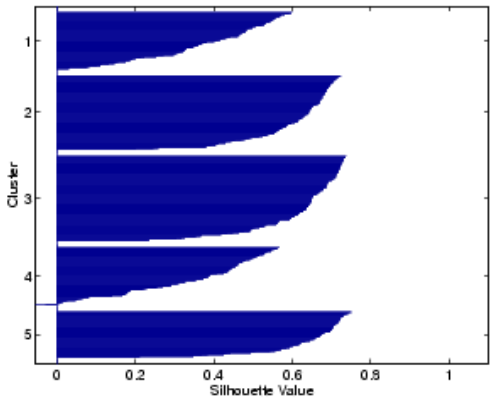
- The Silhouette coefficient of a cluster is the avg silhouette of all its objects
  - Is a measure of how tightly grouped all the data in the cluster are.
  - > 0,7: strong structure, > 0,5: usable structure ….
- The Silhouette coefficient of a clustering is the avg silhouette of all objects
  - is a measure of how appropriately the dataset has been clustered



K=3



K=4



K=5

Things you should know from this lecture

Things you should know …

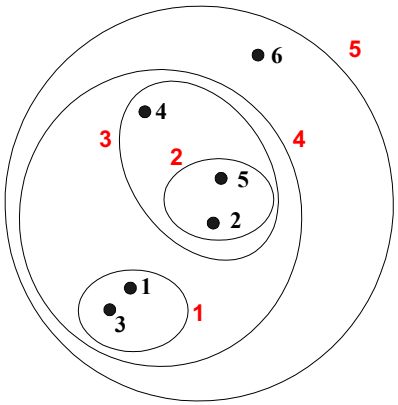
- Principal difference between unsupervised learning and supervised learning
- Categories of diverse clustering methods
- Principal partitioning-based clustering approaches
  - k-Means
  - k-Medoid

Outline

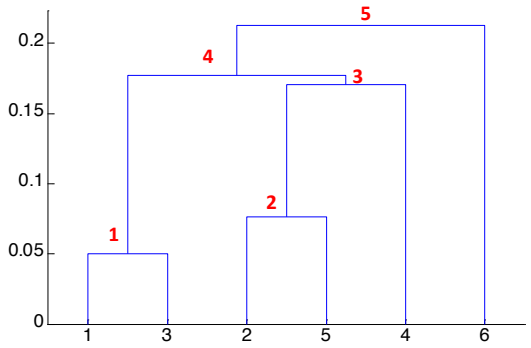
- Hierarchical-based clustering

Hierarchical methods idea

- 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
  - The height at which two clusters are merged in the dendrogram reflects their distance



Nested clusters



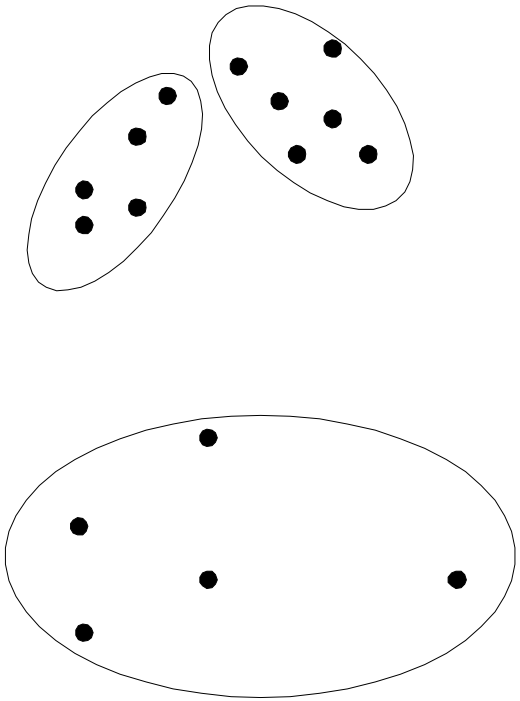
Dendrogram

Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

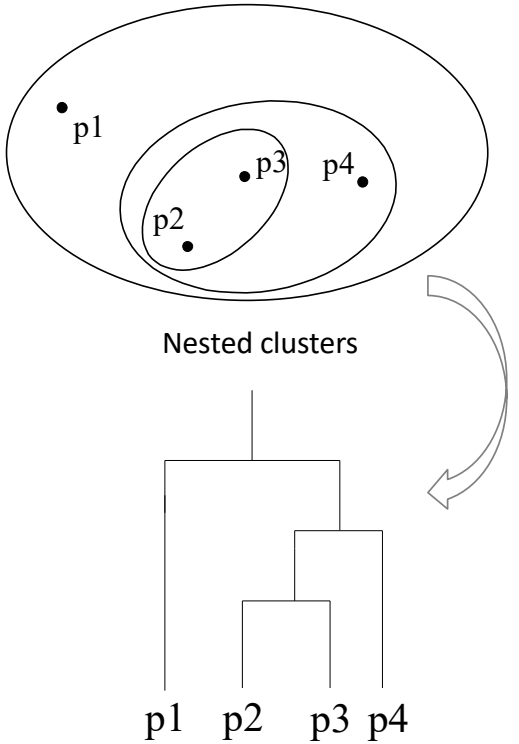
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical vs Partitioning



Partitioning clustering

Partitioning algorithms typically have global objectives



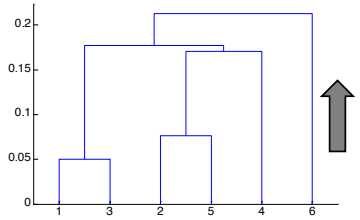
Nested clusters

Dendrogram

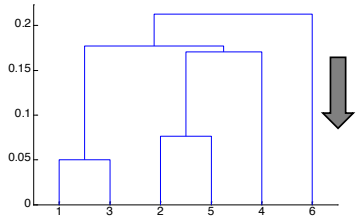
Hierarchical clustering algorithms typically have local objectives

Hierarchical clustering methods

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or *k* clusters) left
    - e.g., AGNES



  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are *k* clusters)
    - e.g., DIANA



- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative clustering algorithm

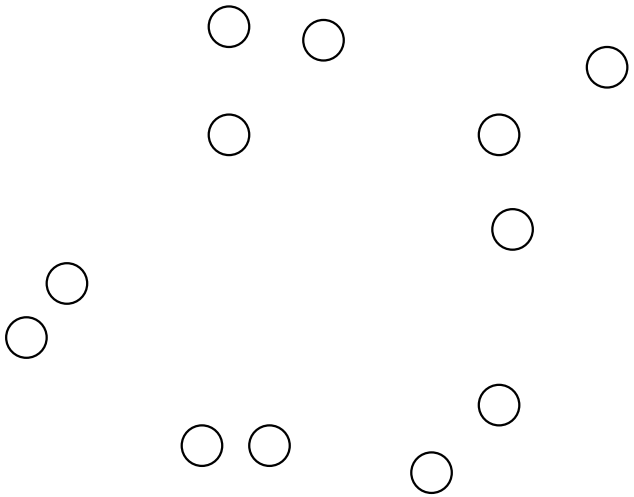
- More popular hierarchical clustering technique
- Basic algorithm is straightforward

1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

- Key points:
  - the computation of the proximity of two clusters
    - Different approaches to defining the distance between clusters distinguish the different algorithms (single link, complete link, …..)
  - the update of the proximity matrix due to cluster merges

Starting situation

- Start with clusters of individual points and a proximity matrix

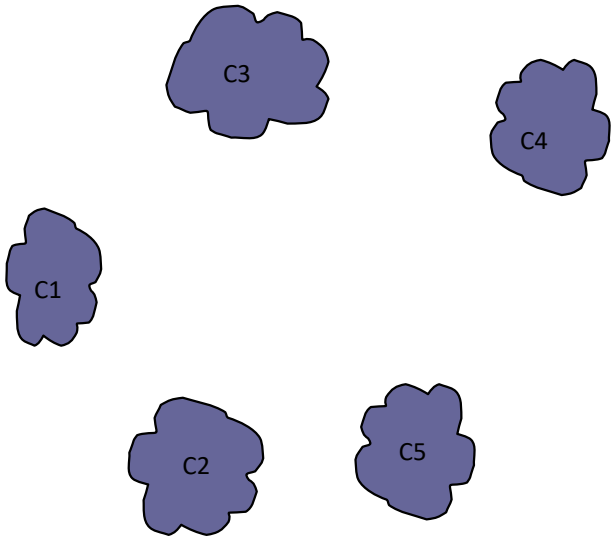


	p1	p2	p3	...	p12	
p1						
p2						
p3						
...						
p12						

Proximity matrix

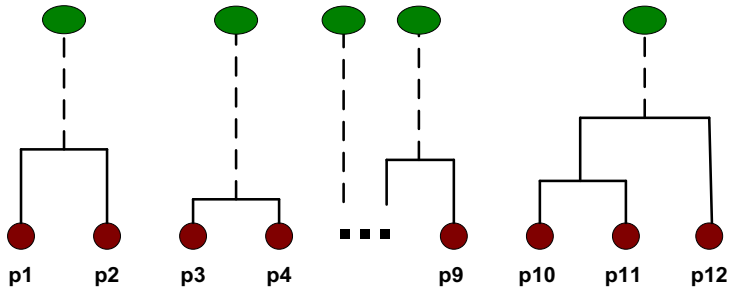
Intermediate situation I

- After some merging steps, we have some clusters



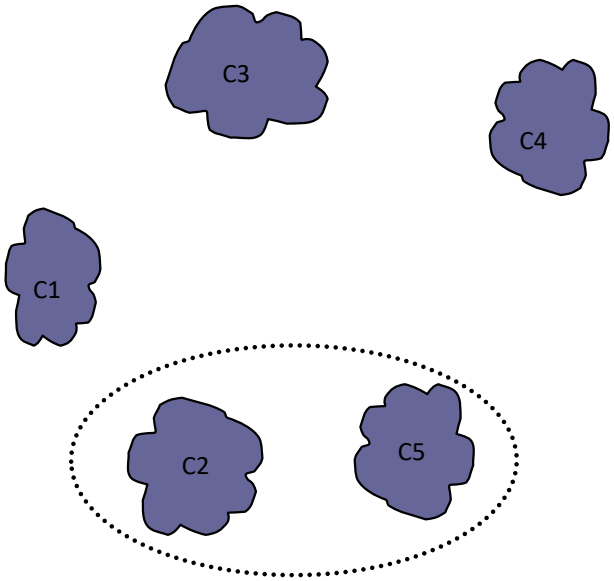
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

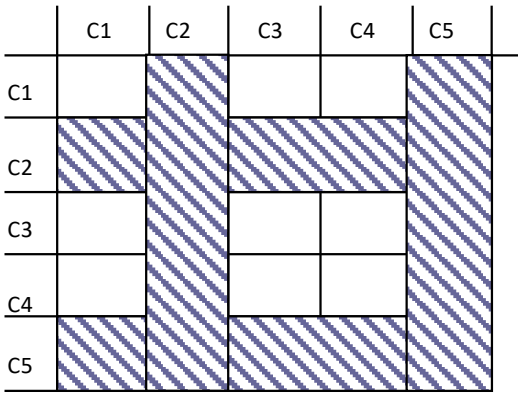
Proximity matrix



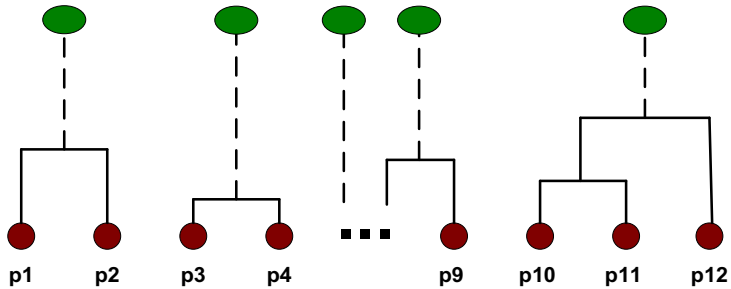
Intermediate situation II

- We want to merge the two closest clusters (C_2 and C_5) and update the proximity matrix.



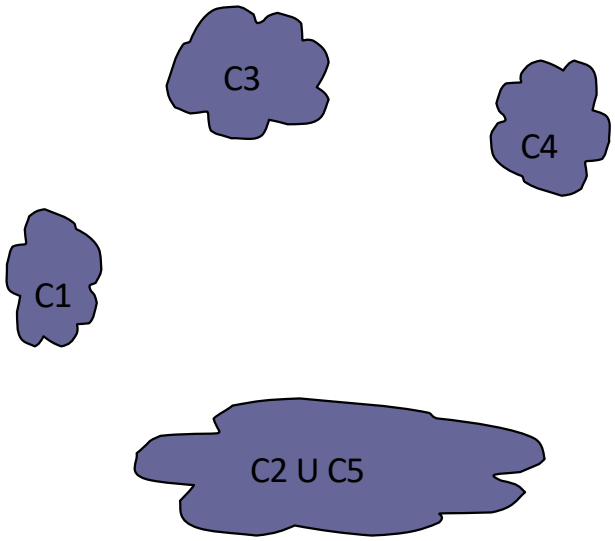


Proximity matrix



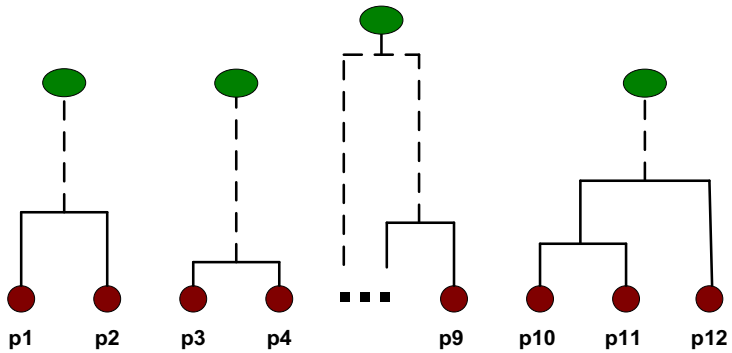
After merging

- The question is “How do we update the proximity matrix?” Or, in other words, what is the similarity between two clusters?

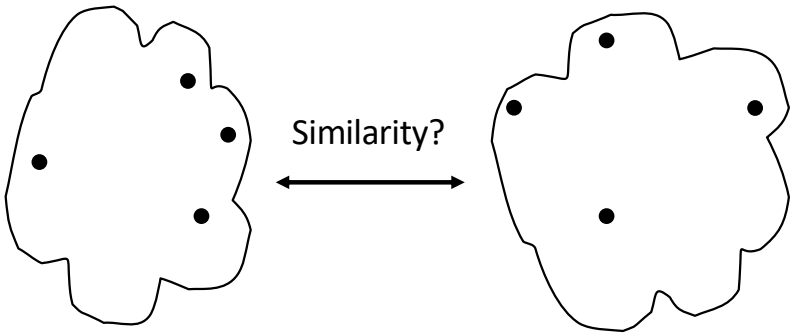


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity matrix



Measures of inter-cluster similarity I



	p1	p2	p3	…	p12	
p1						
p2						
p3						
…						
p12						

Proximity matrix

- A variety of different measures:
- Single link (or MIN)
- Complete link (or MAX)
- Group average
- Distance between centroids
- Distance between medoids
- Other methods driven by an objective function
  - Ward's Method uses squared error

Typical alternatives to calculate the distance between clusters

- Single link: smallest distance between an element in one cluster and an element in the other, i.e.,

$$dis_{sl}(C_i, C_j) = \min_{x,y}\{d(x,y) | x \in C_i, y \in C_j\}$$

- Complete link: largest distance between an element in one cluster and an element in the other, i.e.,

$$dis_{cl}(C_i, C_j) = \max_{x,y}\{d(x,y) | x \in C_i, y \in C_j\}$$

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

$$dis_{avg}(C_i, C_j) = \frac{\sum_{x \in C_i, y \in C_j} d(x,y)}{|C_i||C_j|}$$

- Centroid: distance between the centroids of two clusters, i.e.,

$$dis_{centroids}(C_i, C_j) = d(c_i, c_j)$$

- Medoid: distance between the medoids of two clusters, i.e., dis(K_i, K_j) = dis(M_i, M_j)
  - Medoid: one chosen, centrally located object in the cluster