Clustering

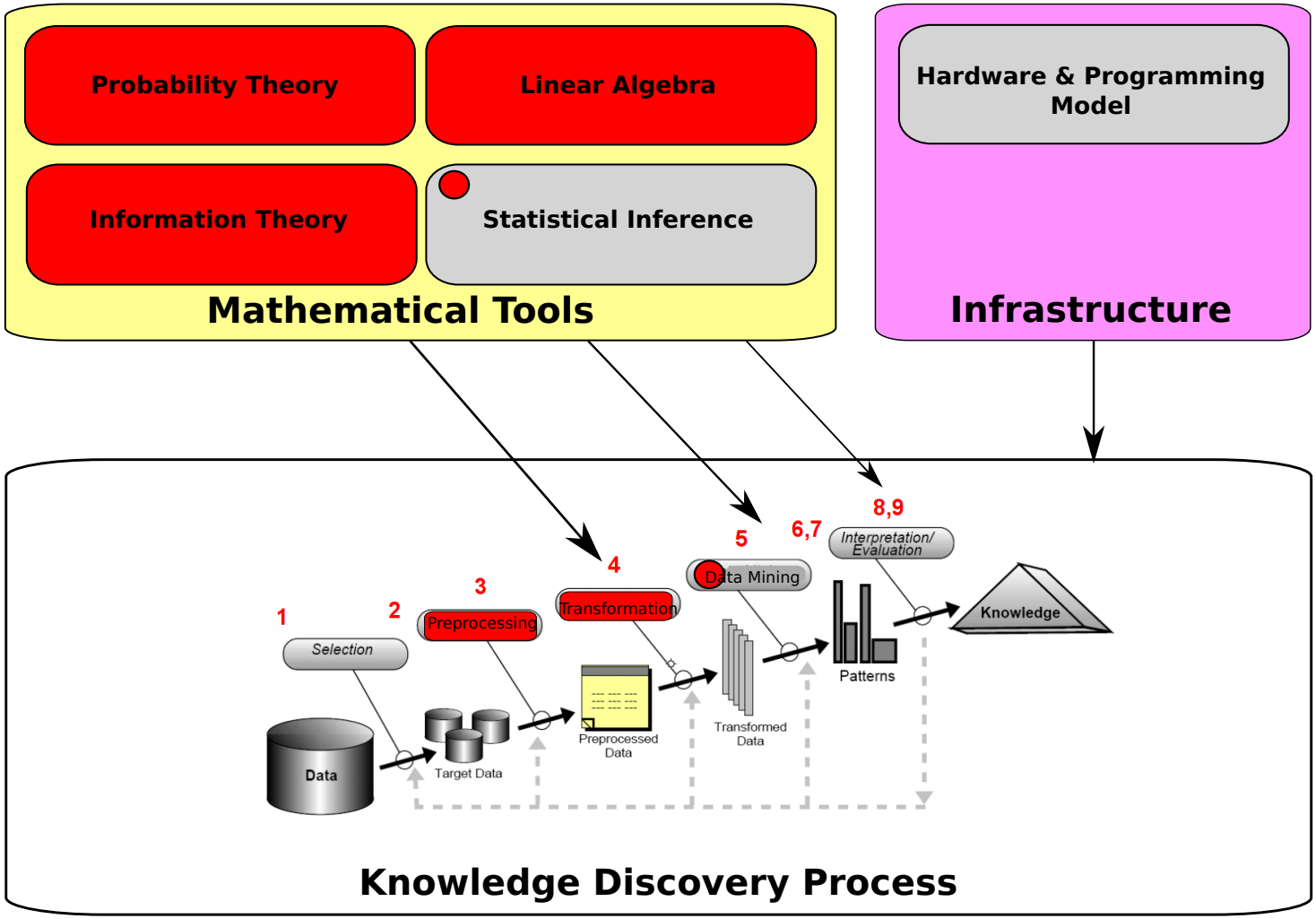
Knowledge Discovery and Data Mining 1

Roman Kern

KTI, TU Graz

2015-12-17

Big picture: KDDM



Outline

Clustering

An unsupervised machine learning technique

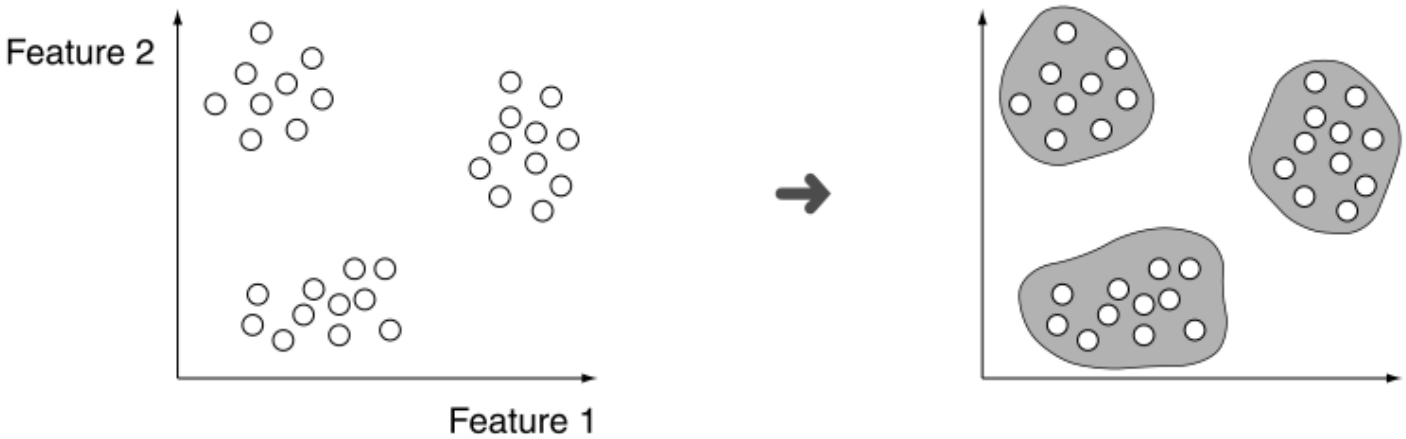
Introduction

Introduction

- Identify group of instances with the following objectives
  1. Maximize the similarity with a group (intra group)
  2. Minimize the similarity between the groups (inter group)
- Clustering is an unsupervised method
- ... in contrast to supervised methods no training data is needed (i.e. no external teacher)
- ... the optimisation criterion is often task and domain-independent

Introduction

Example



Introduction - Use Cases

Use Cases & Applications

- Group similar items
  - e.g. similar buyers
- Text categorization
  - e.g. for search engine results
- Pattern recognition
- Supporting task, e.g. outlier detection
- Vector quantisation
  - e.g. for lossy compression

Introduction - Basics

Basics

- Similarity is often expressed as distance function, $d(x, y)$
- ... the higher the distance, the lower the similarity
- Definition of the distance function depends on the types of features (categorial, numeric, ...)
- All inter-instance distances build a $n \times n$ distance matrix (takes $\mathcal{O}(n^2)$ time) for n instances

Instead of a distance function, often proximity or (dis-)similarity functions are used (often the Euclidean distance or the Cosine similarity are used).

Introduction - Types of Clusterings

Exclusive clustering

- Each instances are only assigned to a **single cluster**
- ... there is no overlap between the clusters

Let X be a set of instances. An exclusive clustering $\mathcal{C}$ of X, $\mathcal{C} = \{C_1, C_2, ..., C_n\}$, $C_i \subseteq X$, is a partitioning of X into non-empty, mutually exclusive subsets C_i with $\bigcup_{C_i \in \mathcal{C}} C_i = X$

Introduction - Types of Clusterings

Fuzzy clustering

- Instances are assigned to **more than one cluster**
- ... includes a weight, how strong the relationship is
- Thus clusters do overlap

Also known as soft clustering

Introduction - Types of Clusterings

Hierarchical clustering

- Clusters are further refined with **sub-clusters**
- ... thus an instance is assigned to a cluster and all its parent clusters
- Thus clusters overlap with their parent clusters

Introduction - Types of Clustering Algorithms

Clustering algorithms

- Iterative
  - Exemplar based (e.g. k-means)
  - Exchange based (e.g. Kerninghan-Lin)
- Hierarchical
  - Agglomerative (e.g. HAC)
  - Devisive (e.g. min-cut)
- Density based
  - Point density based (e.g. DBSCAN)
  - Attraction based (e.g. MajorClust)
- Meta-search controlled
  - Gradient based (e.g. simulated annealing)
  - Competitive (e.g. evolutionary strategies)
- Stochastic
  - Gaussian mixtures (e.g. EM)
- Information theory based
- Subspace clustering

Introduction - Types of Clustering Algorithms

Clustering algorithms - on graphs

- Clique based
- Spectral clustering approaches
- Markov clustering

Clustering approaches - on matrices

- Co-clustering

Note: this is just a selection and one possible way to categorise clustering algorithms

Iterative Clustering

e.g. k-Means

Iterative Clustering

Model-Based Clustering

- Assume a model and then try to fit the parameters
- $\rightarrow$ optimize the fit between the model and the data
- Clustering as a problem to infer these parameters
- Typically each cluster (group) will be presented by a parametric distribution
- $\rightarrow$ for more than one cluster it will be a mixture of distributions

Iterative Clustering

Clustering a Mixture of Gaussians

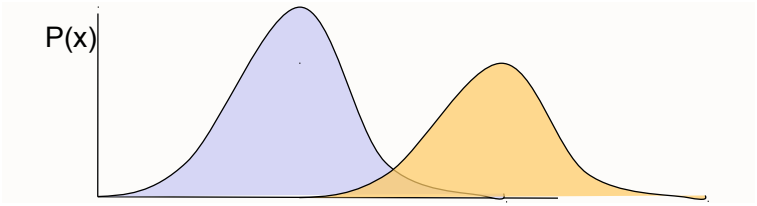
- The clustering assumes, that the data is generated by a mixture of Gaussians
- Each cluster is then a Gaussian distribution
- The task is to find the parameters, μ_i and σ_i for each cluster $C_i \in \mathcal{C}$
- $\rightarrow$ the mean and the standard deviation for each cluster

Plus as an optional task, find out the true number of clusters

Iterative Clustering

Connection to noisy channel

- Transmit two states over a noisy channel
- Two states as input, a real value as output
- The output will look like this:



- Knowing the parameters of the distributions, one could compute the probabilities

Iterative Clustering

k-Means

- Example for an iterative clustering algorithm
- Input:
  - Set of instances - V
  - Distance function - $d(r_i, v_j)$
  - Minimization criteria, based on d
  - Number of desired clusters - k
- Output:
  - Cluster representatives - $r_1, ..., r_k$, the so called centroids
- Each centroid is the mean of the instances within the cluster

k-Means is a partitioning clustering algorithm
In contrast to centroids, **medoids** are the best representative instance within a cluster

Iterative Clustering

k-Means - Algorithm

1. Initialise the centroids ($r_1, ..., r_k$ - seed selection)
2. Iterate over all instances ($v \in V$)
   - Assign each instance to the closest centroid (using d)
3. Update the centroids (using minimization criteria - $r_i = minimize(e(C_i))$)
4. GOTO 2 (until convergence)

Iterative Clustering

k-Means - Open questions

- Which is the correct cluster number?
- Which initialisation to choose?
- Which distance measure?
- Which minimization criteria?

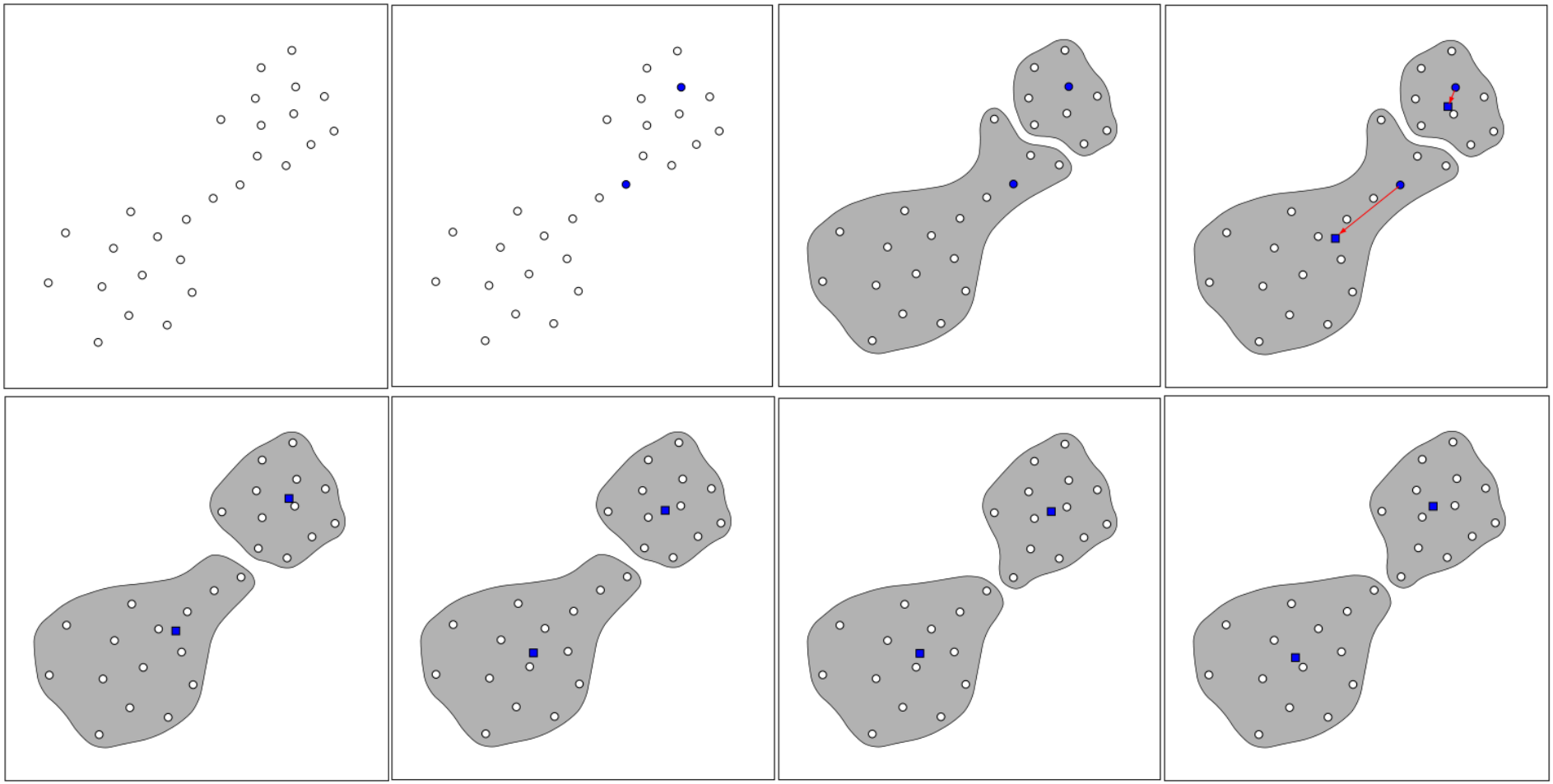
Iterative Clustering

k-Means - Basic algorithm

- Which is the correct cluster number?
  - e.g. predefined
- Which initialisation to choose?
  - e.g. random initialisation
- Which distance measure?
  - e.g. Euclidean distance
- Which minimization criteria?
  - e.g. component wise arithmetic mean of the cluster members

If the data is from a metric space, then as minimization criterion e the sum of the squared distances to the cluster representatives (= variance criterion) is usually chosen $\rightarrow$ vector of minimum variance

Iterative Clustering



Iterative Clustering

k-Means - Advantages

- Conceptually simple
- Efficient, low computational complexity - $\mathcal{O}(n)$, for n instances
- The basic algorithm produces hard clustering, but can be extended to fuzzy clustering
- Can be extended to work iteratively (online k-Means)
- Partitions the instances $\rightarrow$ Voronoi diagram

k-Means - Disadvantages

- Only finds local minimal
  - Start multiple times with random seeds
  - Devise "clever" seeding methods
- Needs fixed k (cluster number)
- Does not work well for elongated clusters or density based clusters
- Does not deal well with noisy data, outliers

Iterative Clustering

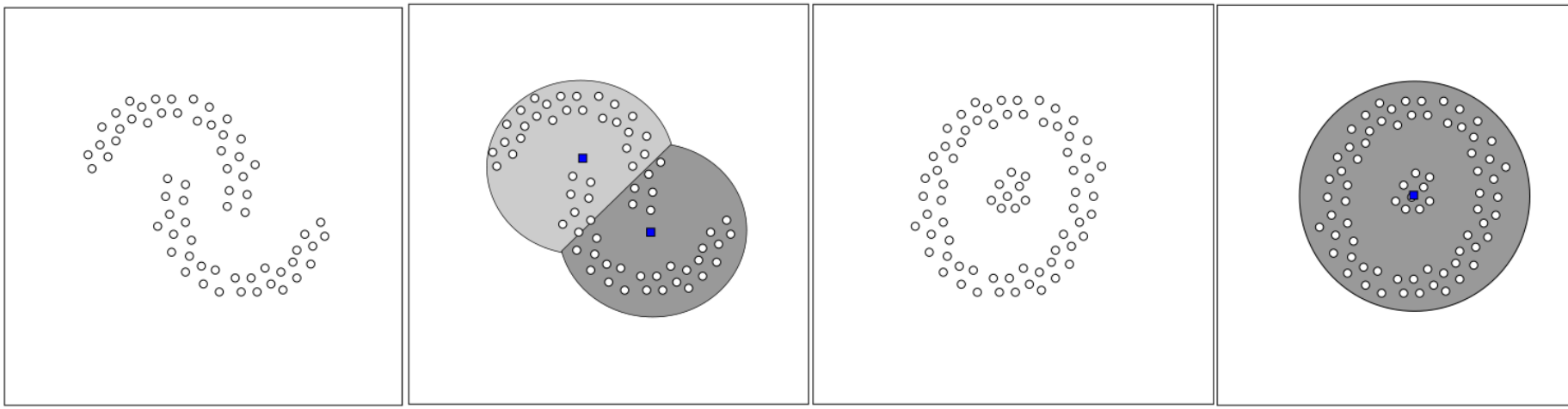
k-Means - Advanced algorithms

- Which is the correct cluster number?
  - e.g. Bisecting k-Means (start with 2 clusters, split the largest one)
- Which initialisation to choose?
  - e.g. k-Means++ (try to cover the feature space)
- Which distance measure?
  - e.g. Spherical k-Means (cosine similarity)
- Which minimization criteria?
  - e.g. k-center ($max_{v \in C_i} |v - r_i|$)

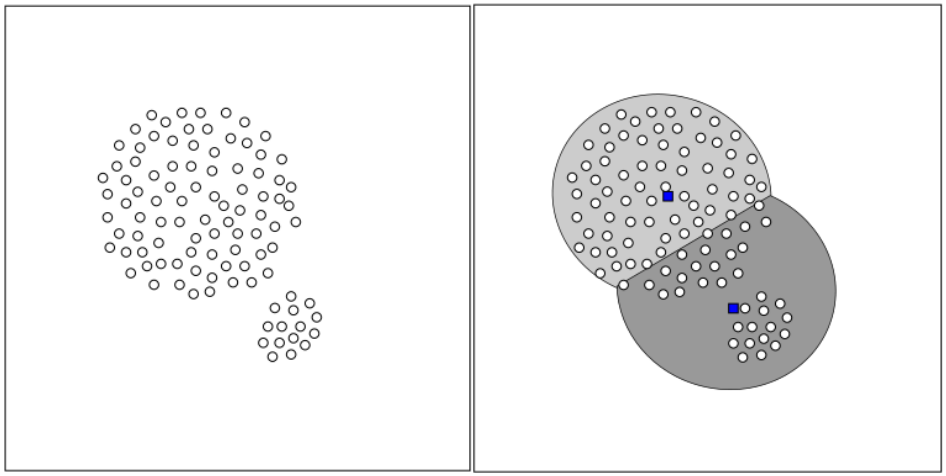
Iterative Clustering

k-Means - Limitations

- Nested clusters



- Clusters with large differences in size



Iterative Clustering

Expectation–Maximization (EM) algorithm

- Model based clustering algorithm
- ... iteratively compute the Maximum Likelihood estimate for the model parameters, in the presence of hidden or missing data (latent variables)
- Algorithm
  1. Initialise parameters
  2. E-step (estimate missing data by observations)
  3. M-step (maximize likelihood function)
  4. GOTO 2 (until convergence)
- Used for mixture of Gaussians
- Soft clustering

Hierarchical Clustering

e.g. HAC

Hierarchical clustering

Two basic approaches

1. **Agglomerative**: start with each instance as single cluster and then merge (bottom-up)
2. **Divisive**: start with a single cluster of all instances and then split (top-down)

Hierarchical clustering

Hierarchical Agglomerative Clustering (HAC)

- Input:
  - Distance measure for two clusters - d_C
- Output:
  - Cluster hierarchy (dendrogram)

Clusters are not represented by centroids, there is no fixed number of clusters

Hierarchical clustering

HAC - Algorithm

1. Initialise clustering (one cluster per instance)
2. Update the distance matrix
3. Select the pair of clusters with the lowest distance ($argmin_{\{C_i, C_j\} \subset \mathcal{C}: C_i \neq C_j} d_{\mathcal{C}}(C_i, C_j)$)
4. Merge the selected pair
5. GOTO 2 (until stop criteria has been reached)

Many implementation use an explicit distance (similarity, proximity) matrix (see step 2) - but it is not strictly required

Hierarchical clustering

HAC - Open questions

- Which is the distance measure (for clusters)?
- Which stop criteria?

Hierarchical clustering

HAC - Stop criterion

- Alternatives
- # Cluster
- Threshold on distance (e.g. infinity)
- Size of clusters

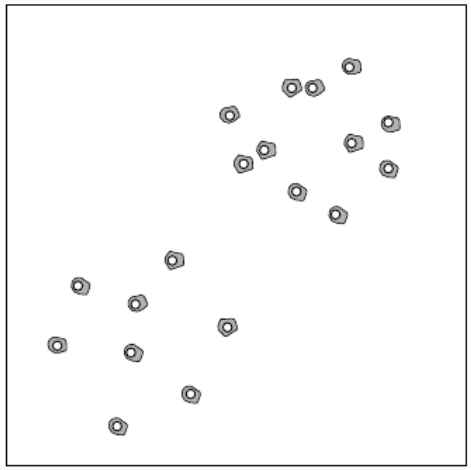
Hierarchical clustering

HAC - Distance Measure

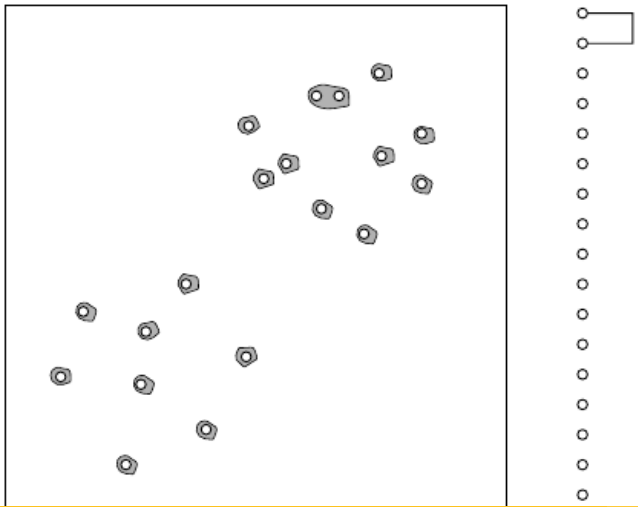
- The distance measure represents the inter-cluster relationship
- ... also the term linkage is used for this distance measure
- The most common used linkage types are
  - **Single link** (nearest neighbour, MIN)
    $d_{\mathcal{C}}(C_i, C_j) = min_{u \in C_i, v \in C_j} d(u, v)$
  - **Complete link** (furthest neighbour, MAX)
    $d_{\mathcal{C}}(C_i, C_j) = max_{u \in C_i, v \in C_j} d(u, v)$
  - **Group average link**
    $d_{\mathcal{C}}(C_i, C_j) = \frac{1}{|C_i||C_j|} sum_{u \in C_i, v \in C_j} d(u, v)$
  - **Ward criterion** (variance)
    $d_{\mathcal{C}}(C_i, C_j) = \sqrt{\frac{2|C_i||C_j|}{|C_i|+|C_j|}} ||\bar{u} - \bar{v}||$
  - **Centroid**
    $d_{\mathcal{C}}(C_i, C_j) = d(r_i, r_j)$

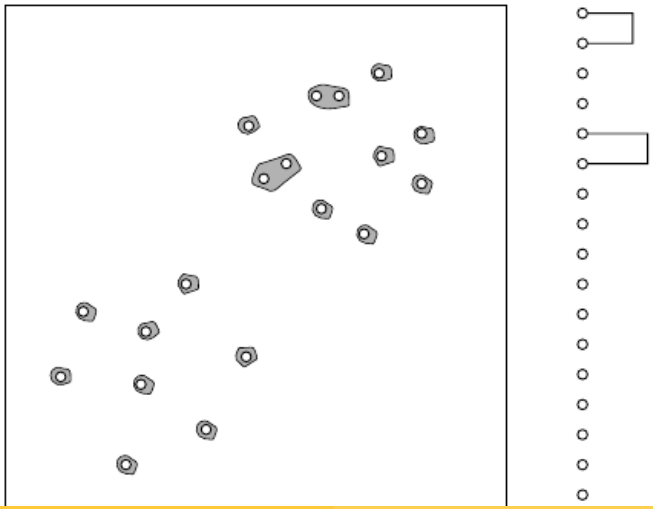
Hierarchical clustering - Single Linkage Example

Start state: single cluster for each instance

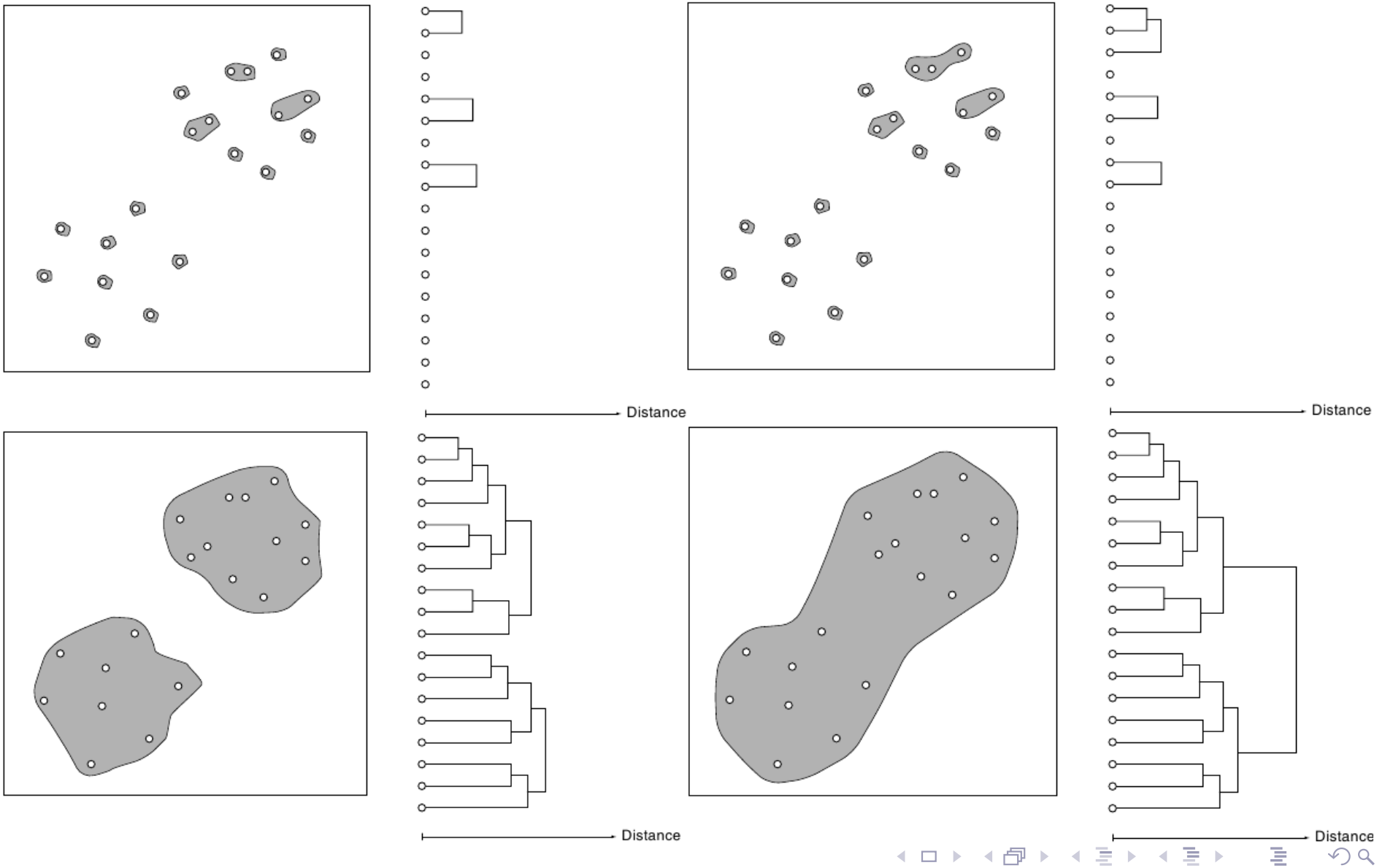


Iteratively merge clusters to build a dendrogram





Hierarchical clustering - Single Linkage Example



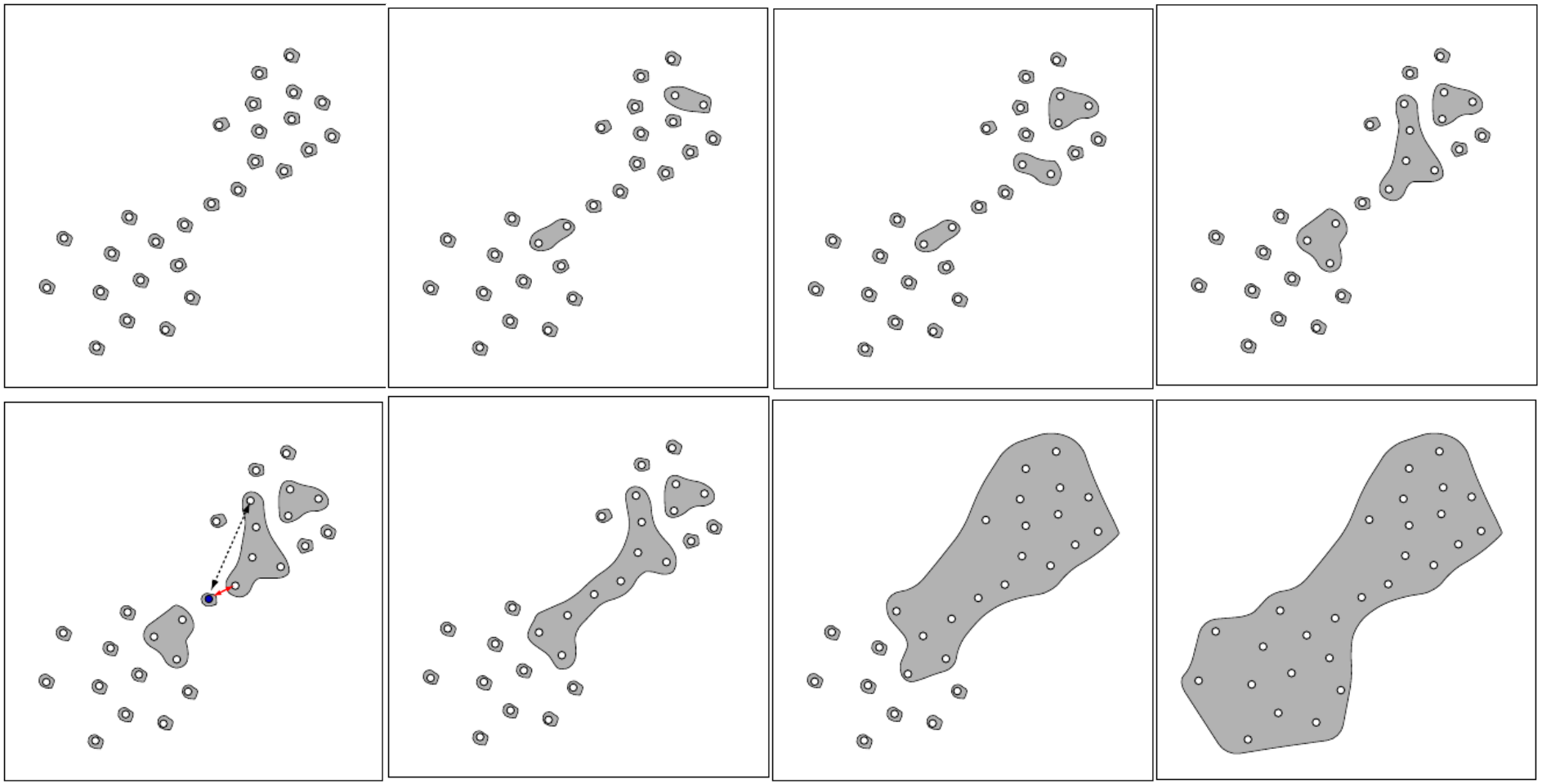
Hierarchical clustering

Comparison of linkage types

	single link	complete link	average link	Ward criterion
characteristic	contractive	dilating	conservative	conservative
cluster number	low	high	medium	medium
cluster form	extended	small	compact	spherical
chaining tendency	strong	low	low	low
outlier-detecting	very good	poor	medium	medium

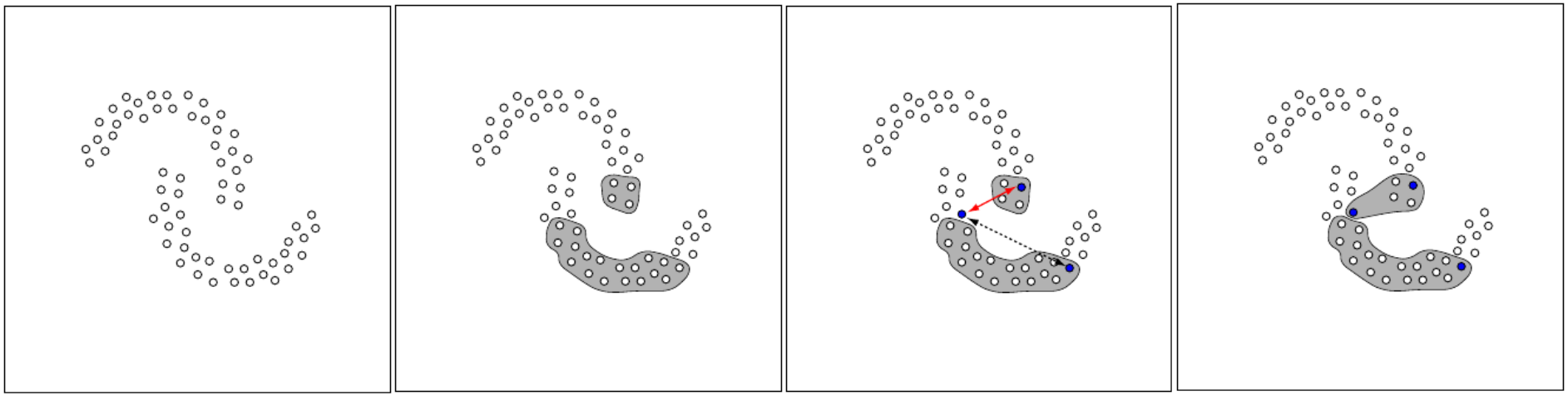
Hierarchical clustering

Chaining issues with single link



Hierarchical clustering

Nesting issues with complete link



Hierarchical clustering

HAC - Advantages

- Arbitrary measures for distance and and similarities
- No need to specify the true number of clusters

HAC - Disadvantages

- Relatively high computational complexity $\mathcal{O}(n^2)$ (to $\mathcal{O}(n^3)$), depending on the linkage
  - In many implementations HAC uses the distance matrix as input
  - Computing this matrix is already $\mathcal{O}(n^2)$
- High memory requirements, if distance matrix is stored in memory ($\mathcal{O}(n^2)$)
- Greedy nature and no backtracking

Density-Based Clustering

e.g. DBSCAN

Density-Based Clustering

Basic approaches

- Density based clustering tries to partition the instances into groups of equal density
- Two types:
- Parameter-based: if the distribution is known
- Parameterless: need to estimate the distribution and parameters

Density-Based Clustering

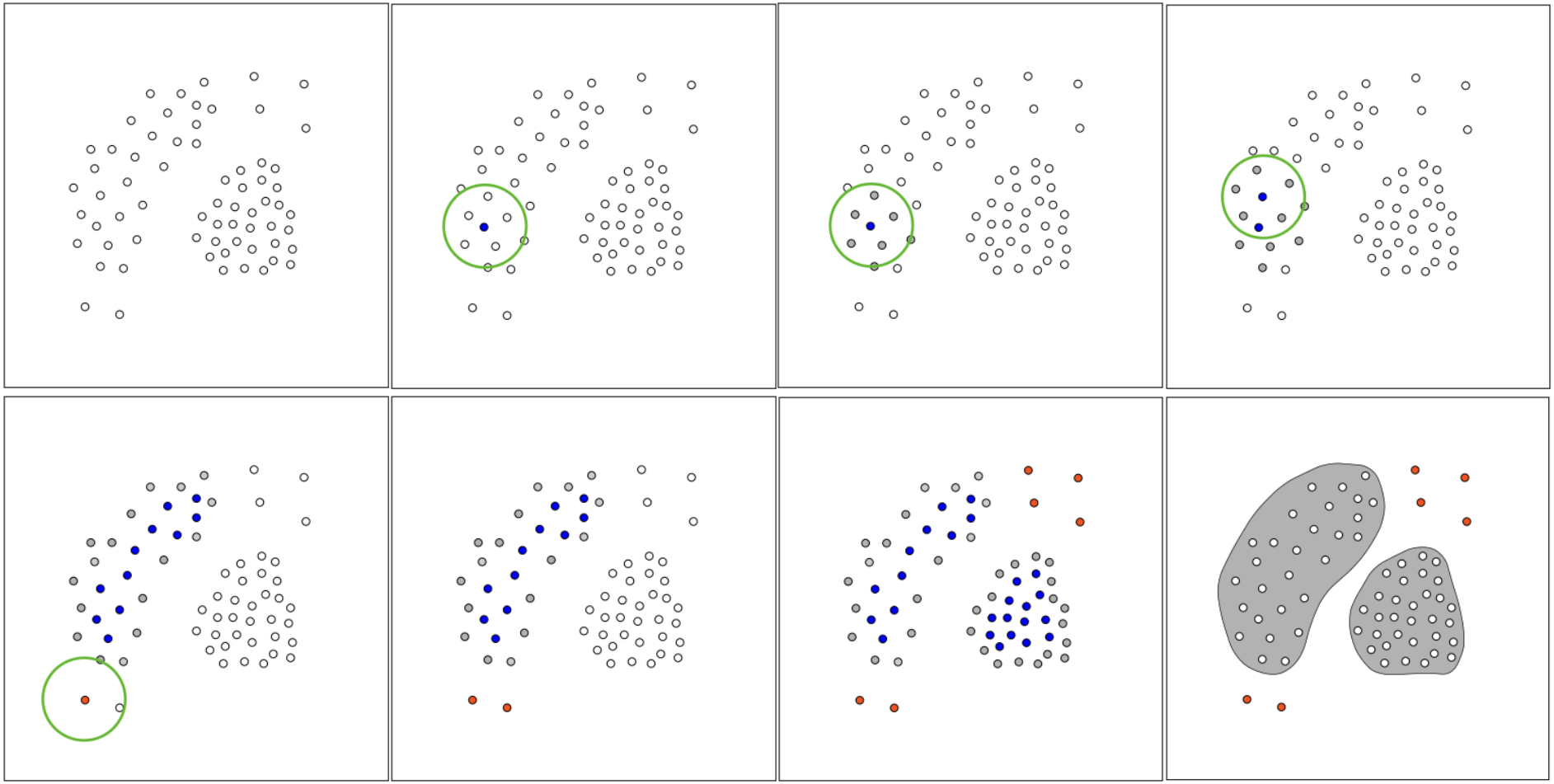
Example: DBSCAN

- Tries to identify three types of instances, based on their neighbourhood $N_\varepsilon(v)$
- Core points: if $|N_\varepsilon(v)| \geq t_{min}$
- Noise point: if not density reachable from any core point
- Border point: otherwise
- Density-reachable
  - If at least a single point in the neighbourhood is a core point,
  - or there is a chain of density-reachable points

In German these types are named: Kern-, Grenz-, and Geräuschpunkte

Density-Based Clustering - DBSCAN

Example of a DBSCAN clustering



Density-Based Clustering - DBSCAN

Advantages

- Low computational complexity (linear)
- No predefined number of clusters needed
- Stable results
  - Deterministic for the most part (expect border points)
  - Independent from the sequence of instances

Disadvantages

- Does not work well with high dimensional data
- Assumption of equal density does not always hold

Other Clustering Approaches

Alternatives & Additional Steps

Other Approaches

Spectral clustering

- Starting with the similarity matrix
- Compute the spectrum (eigenvalues)
- Reduce the dimensionality
- As a preprocessing step for further clustering

Graph clustering

- Make use of the graph structure
- Many proposed algorithms
- e.g. Affinity propagation, Markov clustering

Introduction

Co-clustering

- Cluster more than one dimension at the same time (mutual reinforcement)
- Consider a matrix, e.g. *document* $\times$ *terms*
- Clustering now tries to find topics
- Co-clustering groups the documents to topics and the terms to topics

High-dimensionality

- Subspace clustering targets high dimensional data
- ... as many approaches fail (curse of dimensionality)
- Typically need techniques to keep the computational complexity low

Preprocessing for clustering

Prepare the data for clustering

- Remove noise and outliers
  - Some algorithms do not cope well (e.g. k-Means)
- Normalise the data
- Compute an approximation of clustering to speed up the computation
  - e.g. Canopy clustering to initialise the seeds for k-means
  - Also helps to partition the data in a map/reduce environment
- Estimate the number of clusters
  - Some algorithms need the number of clusters
  - Techniques from model selection
- Reduce the dimensionality & transformation of features
  - e.g. Singular Value Decomposition (SVD)

Postprocessing for clustering

Postprocessing the output of clustering

- Remove spurious clusters (empty, small) - could be outliers, noise
- Validation of results
  - e.g. reject solutions
- Merge clusters (if close to each other)
- Split clusters (if inhomogeneous)

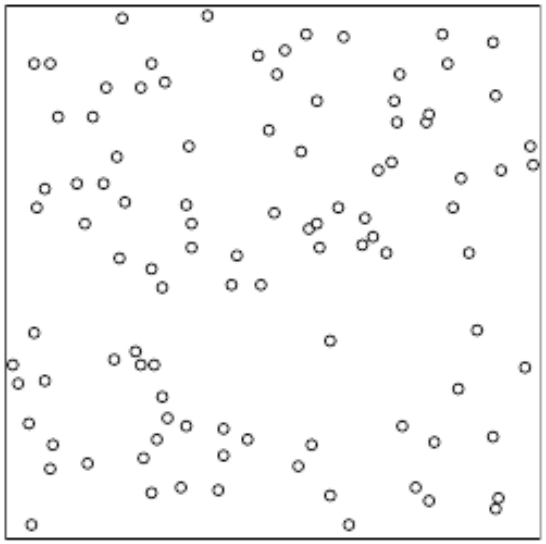
Evaluation

How to assess the quality of clustering

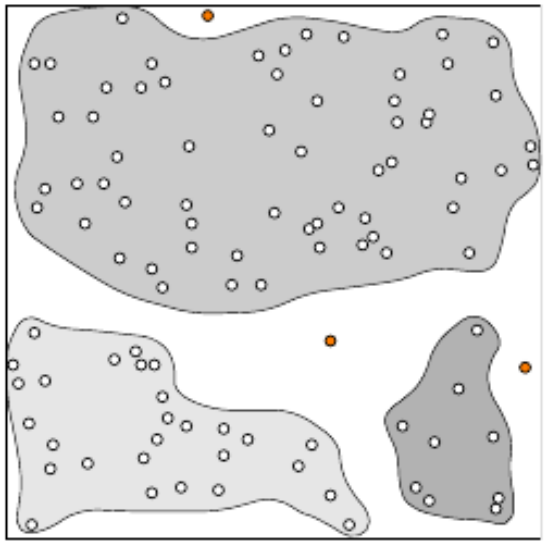
Evaluation

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage." [Jain/Dubes 1990]

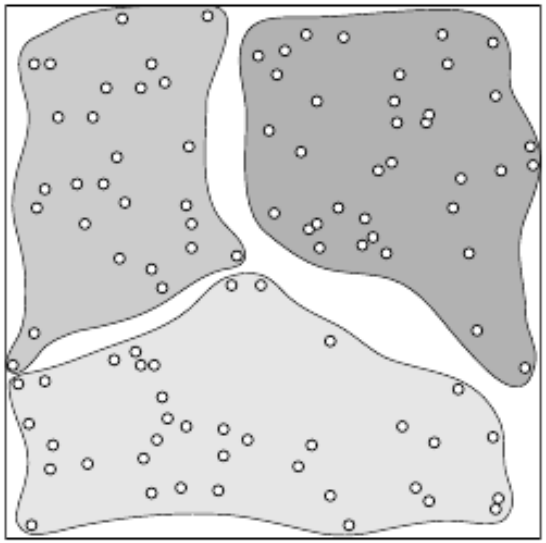
Evaluation



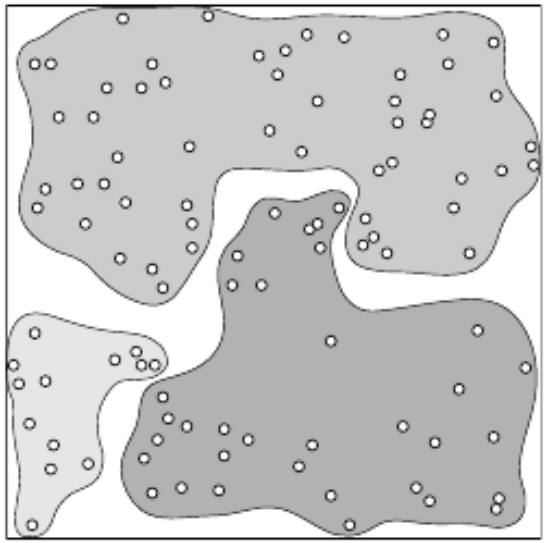
Random points



DBSCAN



k-means



Complete link

Evaluation

Cluster evaluation approaches

- Human evaluation
- Algorithmic evaluation

Cluster evaluation goals

- Runtime vs. quality of results

Evaluation - Human Evaluation

Human clustering evaluation

- Domain expert judges results
- ... has a certain degree of subjectivity
- ... hard to reproduce
- ... labour (and time) intensive

Evaluation - Algorithmic Evaluation

Algorithmic evaluation

- **External validity information**
- ... compare the clustering with a reference
- **Internal validity information**
- ... analyse intrinsic characteristics of the clustering
- **Relative validity measures**
- ... analyse the sensitivity during clustering generation

Evaluation - Algorithmic Evaluation - External

Algorithmic evaluation - external class information

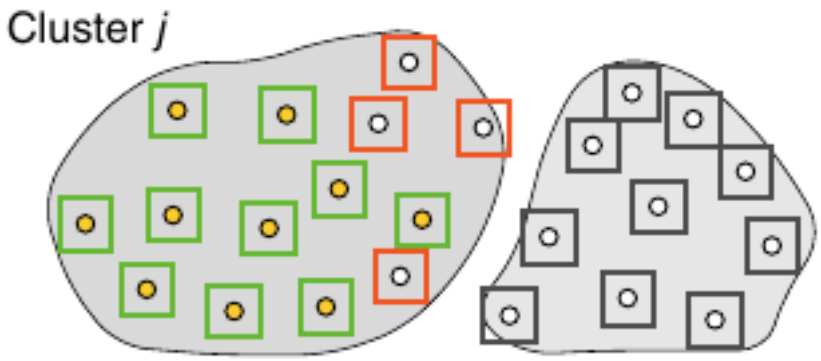
- If the instances contain a class information
- ... compare the clustering with the class
- Link to supervised classification
- Clusters: $\mathcal{C} = \{C_1, ..., C_n\}$
- Classes: $\mathcal{C}^* = \{C_1^*, ..., C_l^*\}$

Evaluation - Algorithmic Evaluation - External

Purity

- Ratio between the dominant class and the cluster size
- $Purity(C_i) = \frac{1}{|C_i|} max_j(|C_i \cap C_j^*|)$
- ... measures how "pure" a cluster is
- Favours smaller clusters

Evaluation - Algorithmic Evaluation - External



		Truth	
		P	N
Hypothesis	P	TP (a)	FP (b)
	N	FN (c)	TN (d)

TP ... true positives, FP ... false positives,
FN ... false negatives, TN ... true negatives

Evaluation - Algorithmic Evaluation - External

Per cluster analysis

- Precision: $\frac{TP}{TP+FP}$
- Recall: $\frac{TP}{TP+FN}$
- F_1: harmonic mean of precision and recall

Evaluation measures known from the field of Information Retrieval

Evaluation - Algorithmic Evaluation - External

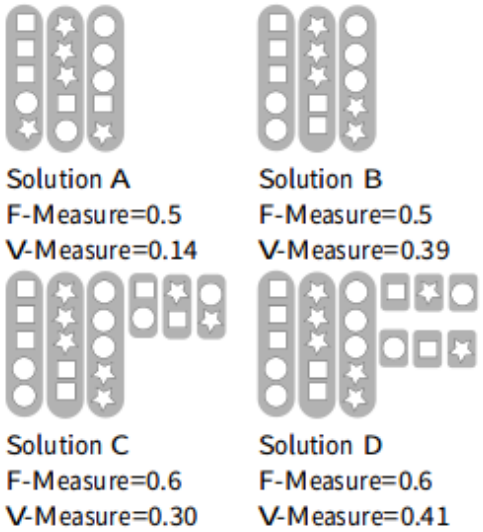
Pairwise comparison

- Pick pairs of instances
- Compare clustering and classes
  - Pair in same cluster and same class $\rightarrow$ TP
  - Pair in same cluster and different classes $\rightarrow$ FP
  - Pair in different cluster and same class $\rightarrow$ FN
  - Pair in different cluster and different classes $\rightarrow$ TN
- Rand Index: $\frac{TP+TN}{TP+TN+FP+FN}$

Evaluation - Algorithmic Evaluation - External

V-Measure

- How much does knowing the clustering tell us about the classes - and vice versa?
- Homogeneity: $h = 1 - \frac{H(\mathcal{C}^*|\mathcal{C})}{H(\mathcal{C}^*)}$
  (0, if $H(\mathcal{C}^*, \mathcal{C}) = 0$)
- Completeness: $c = 1 - \frac{H(\mathcal{C}|\mathcal{C}^*)}{H(\mathcal{C})}$
  (0, if $H(\mathcal{C}, \mathcal{C}^*) = 0$)
- The V-Measure is the weighted harmonic mean of homogeneity and completeness
- $V_\beta = \frac{(1+\beta)hc}{\beta h + c}$



Evaluation - Algorithmic Evaluation - Internal

Internal validity measure - Edge correlation

1. Compute the similarity matrix
2. Compute the occurrence matrix
   - Place a 1 into a cell, if two instances are in the same cluster
3. Compute the correlation between the two matrices

Thank You!

Next up: Examination

Further information

Special thanks to Benno Stein for his slides:
http://www.uni-weimar.de/en/media/chairs/webis/teaching/lecturenotes/#machine-learning

http://www.slideshare.net/pierluca.lanzi/machine-learning-and-data-mining-06-clustering-introduction