

# Detecting Clusters in Moderate-to-high Dimensional Data:

## Subspace Clustering, Pattern-based Clustering, Correlation Clustering

Hans-Peter Kriegel, Peer Kröger, Arthur Zimek

Ludwig-Maximilians-Universität München

Munich, Germany

<http://www.dbs.ifi.lmu.de>

{kriegel,kroegerp,zimek}@dbs.ifi.lmu.de



1. Please feel free to ask questions at any time during the presentation
2. Aim of the tutorial: get the big picture
  - NOT in terms of a long list of methods and algorithms
  - BUT in terms of the basic algorithmic approaches
  - Sample algorithms for these basic approaches will be sketched briefly
    - The selection of the presented algorithms is somewhat arbitrary
    - Please don't mind if your favorite algorithm is missing
    - Anyway you should be able to classify any other algorithm not covered here by means of which of the basic approaches is implemented

1. Introduction
2. Axis-parallel Subspace Clustering
3. Pattern-based Clustering
4. Arbitrarily-oriented Subspace Clustering
5. Summary

# Outline: Introduction

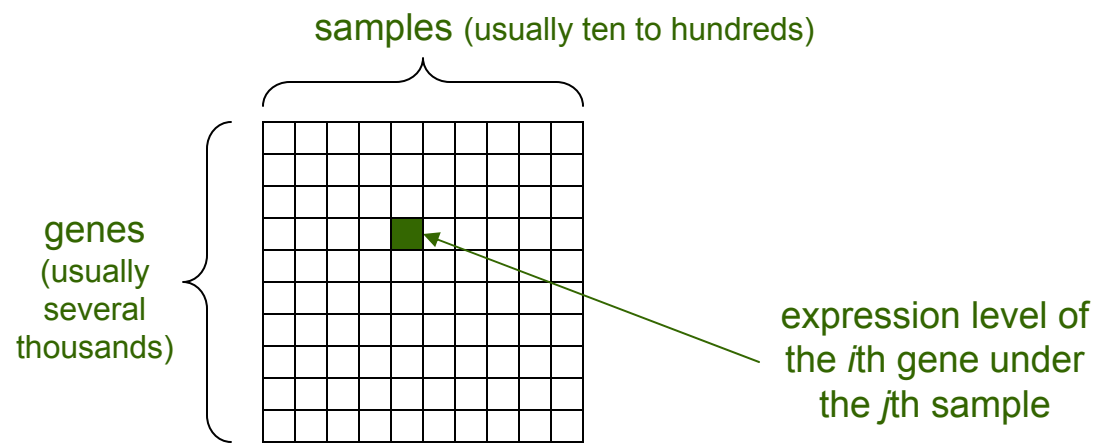
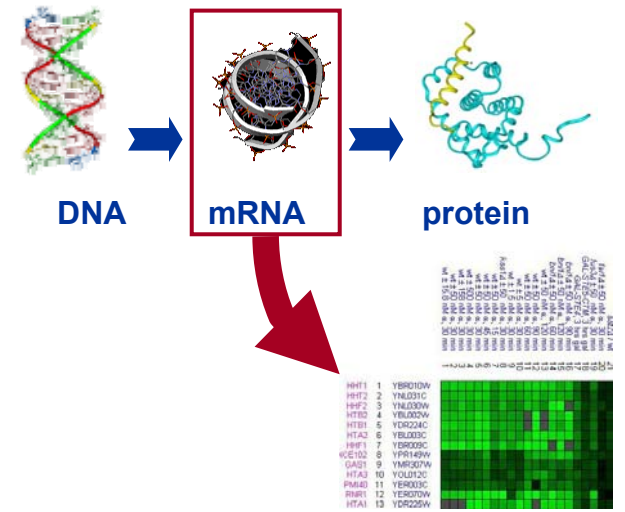
- Sample Applications
- General Problems and Challenges
- A First Taxonomy of Approaches

- Gene Expression Analysis

- Data:

- Expression level of genes in different samples such as
  - different individuals (patients)
  - different time slots after treatment
  - different tissues
  - different experimental environments

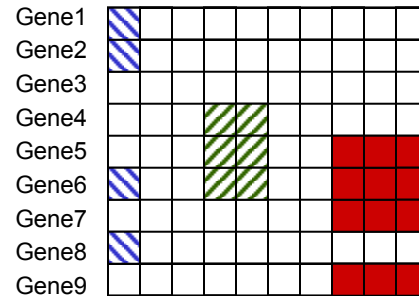
- Data matrix:



# Sample Applications

- Task 1: Cluster the rows (i.e. genes) to find groups of genes with similar expression profiles indicating homogeneous functions

- *Challenge:*  
genes usually have different functions under varying (combinations of) conditions



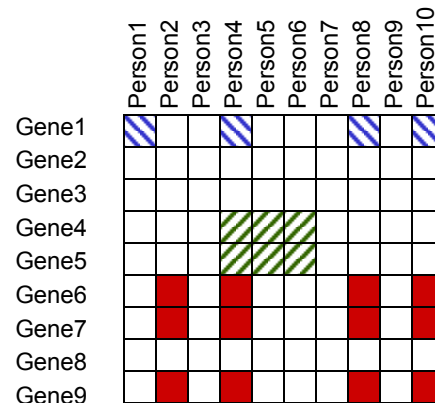
Cluster 1: {G1, G2, G6, G8}

Cluster 2: {G4, G5, G6}

Cluster 3: {G5, G6, G7, G9}

- Task 2: Cluster the columns (e.g. patients) to find groups with similar expression profiles indicating homogeneous phenotypes

- *Challenge:*  
different phenotypes depend on different (combinations of) subsets of genes



Cluster 1: {P1, P4, P8, P10}

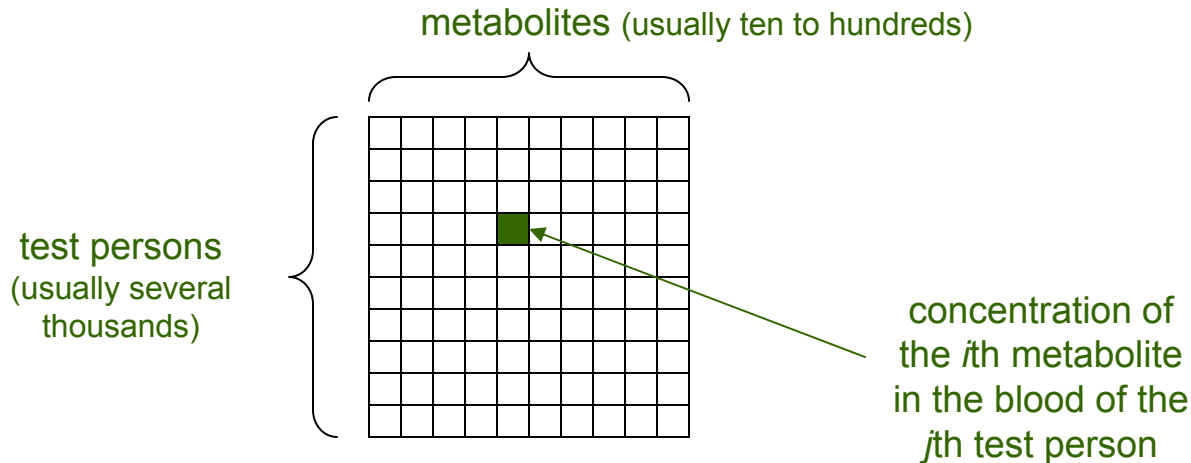
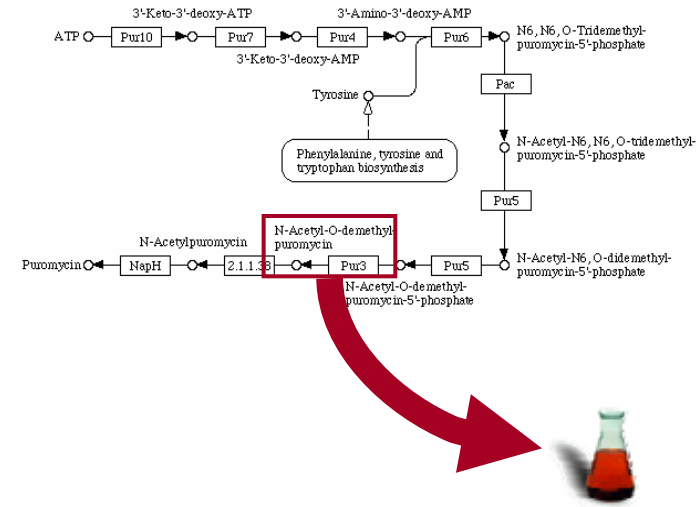
Cluster 2: {P4, P5, P6}

Cluster 3: {P2, P4, P8, P10}

- Metabolic Screening

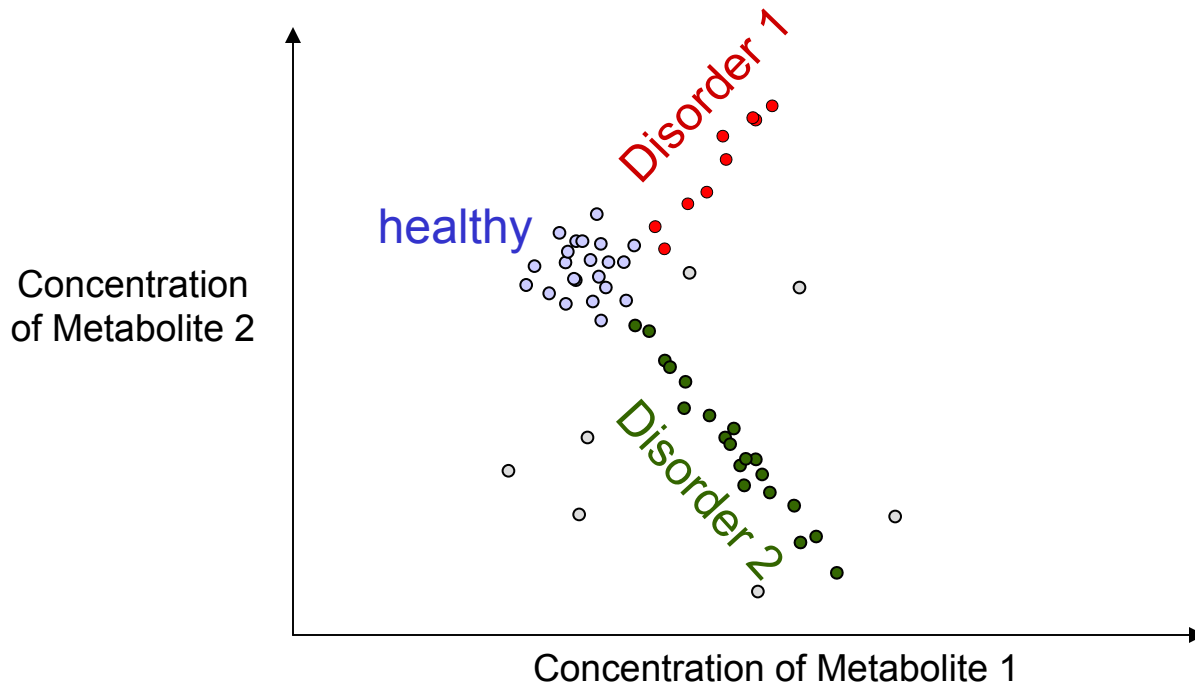
- Data

- Concentration of different metabolites in the blood of different test persons
- Example: *Bavarian Newborn Screening*
- Data matrix:



# Sample Applications

- Task: Cluster test persons to find groups of individuals with similar correlation among the concentrations of metabolites indicating homogeneous metabolic behavior (e.g. disorder)
  - *Challenge:*  
different metabolic disorders appear through different correlations of (subsets of) metabolites

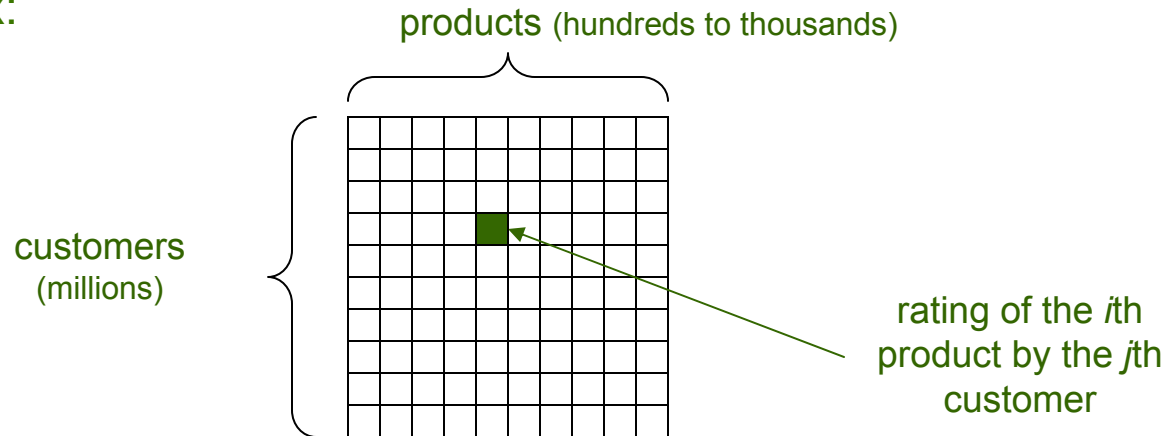




- Customer Recommendation / Target Marketing

- Data

- Customer ratings for given products
- Data matrix:



- Task: Cluster customers to find groups of persons that share similar preferences or disfavor (e.g. to do personalized target marketing)

- *Challenge:*

customers may be grouped differently according to different preferences/disfavors, i.e. different subsets of products

- And many more ...
- In general, we face a steadily increasing number of applications that require the analysis of moderate-to-high dimensional data
- Moderate-to-high dimensional means from apprx. 10 to hundreds or even thousands of dimensions

- The curse of dimensionality  
(from a clustering perspective)
  - Ratio of  $(Dmax_d - Dmin_d)$  to  $Dmin_d$  converges to zero with increasing dimensionality  $d$  (see e.g. [BGRS99,HAK00])
    - $Dmin_d$  = distance to the nearest neighbor in  $d$  dimensions
    - $Dmax_d$  = distance to the farthest neighbor in  $d$  dimensions

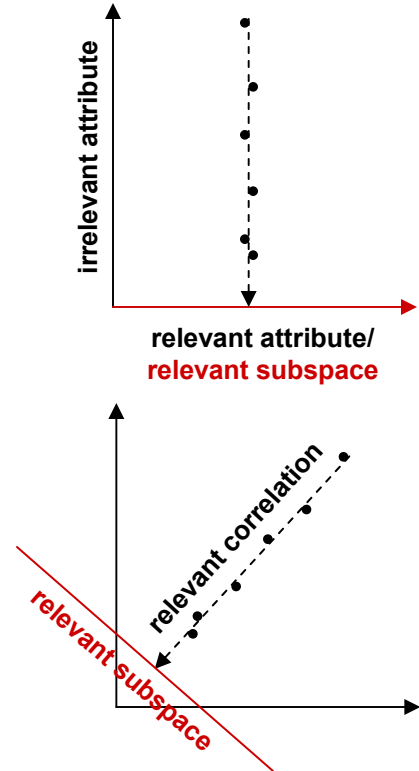
Formally:

$$\forall \varepsilon > 0 : \lim_{d \rightarrow \infty} \mathbf{P} \left[ dist_d \left( \frac{Dmax_d - Dmin_d}{Dmin_d}, 0 \right) \leq \varepsilon \right] = 1$$

- Observable for a wide range of data distributions and distance functions

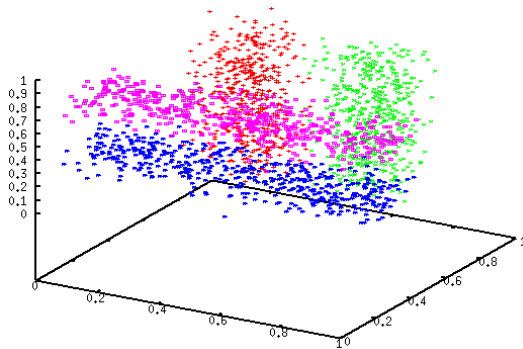
- Consequences?
  - The relative difference of distances between different points decreases with increasing dimensionality
  - The distances between points cannot be used in order to differentiate between points
  - The more the dimensionality is increasing, the more the data distribution degenerates to random noise
  - ***All points are almost equidistant from each other — there are no clusters to discover in high dimensional spaces!***

- Feature relevance and feature correlation
  - Feature relevance
    - A subset of the features may be relevant for clustering
    - Groups of similar (“dense”) points may be identified when considering these features only
  - Feature correlation
    - A subset of features may be correlated
    - Groups of similar (“dense”) points may be identified when considering this correlation of features only

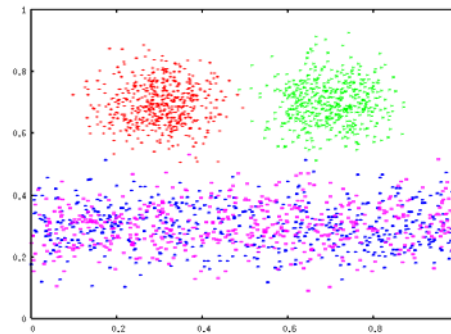


=> Clusters exist in **arbitrarily oriented subspaces** of the data space

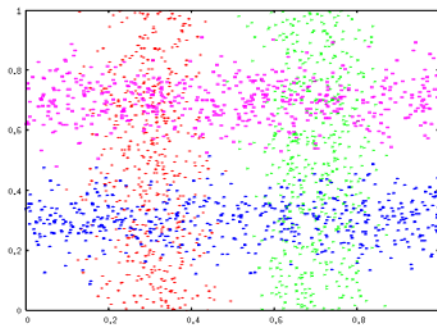
- Local feature relevance/correlation
  - Different (correlations of/subsets of) features are relevant for different clusters



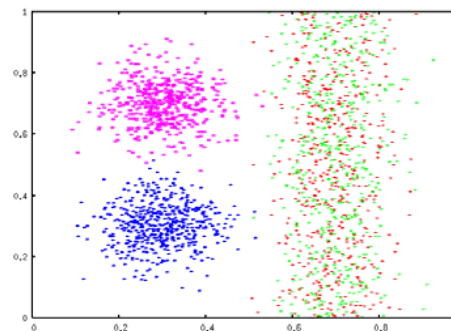
3D data set with four clusters



projection on x/z (relevant for red/green cluster)



projection on x/y

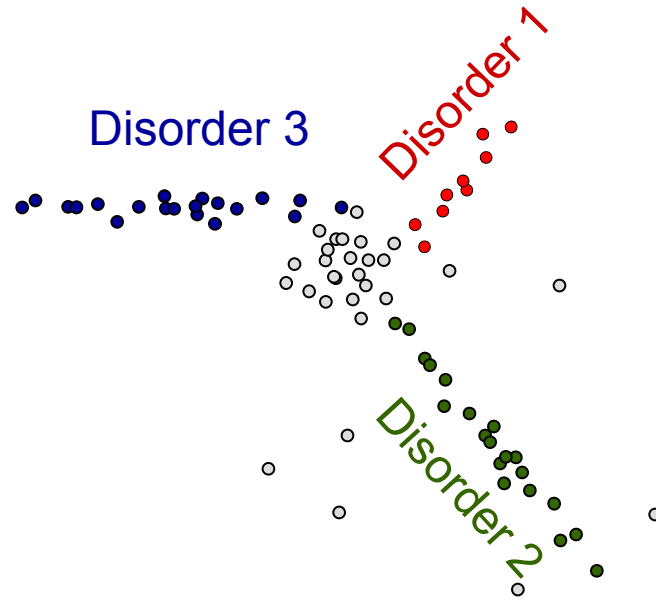


projection on y/z (relevant for blue/purple cluster)

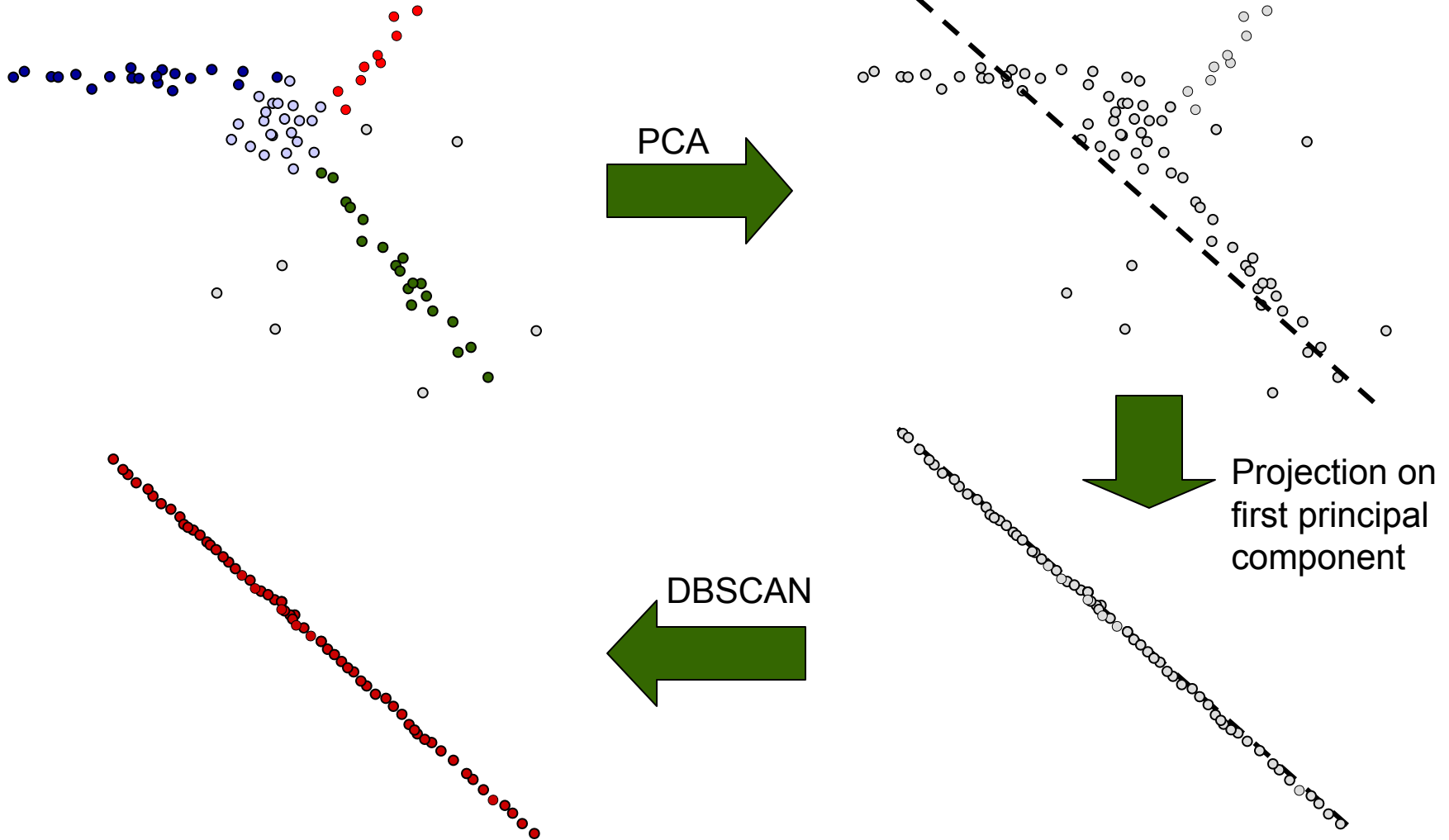
**NOTE:**

*This is problem can already occur in rather low dimensional data (Here: 3D data)*

- Why not feature selection?
  - (Unsupervised) feature selection is **global** (e.g. PCA), i.e. always returns only one (reduced) feature space
  - The **local** feature relevance/correlation problem states that we usually need multiple feature spaces (possibly one for each cluster)
  - Example: Simplified metabolic screening data (2D)

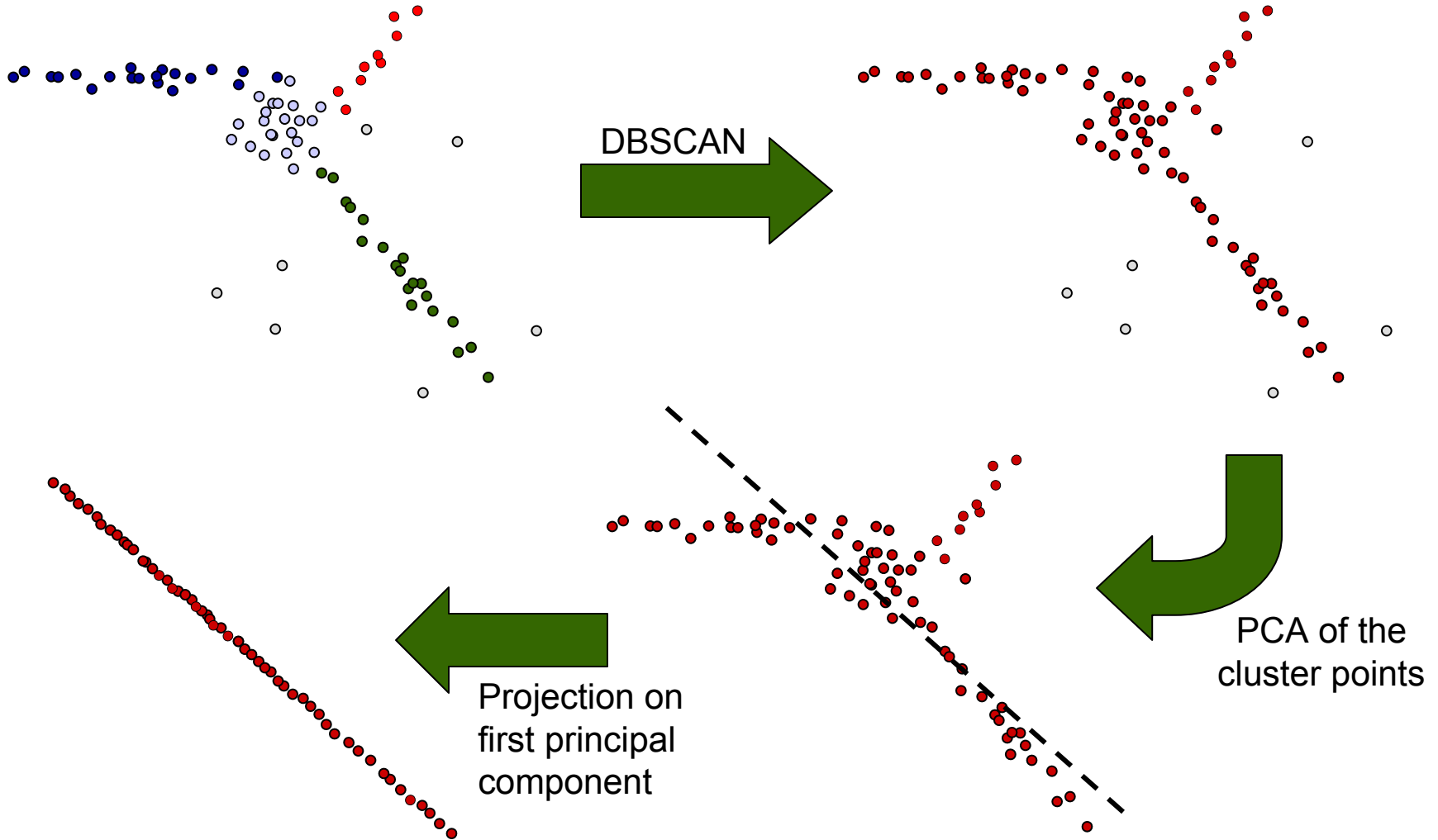


– Use feature selection before clustering





– Cluster first and then apply PCA



- Problem Summary
  - Curse of dimensionality/Feature relevance and correlation
    - Usually, no clusters in the full dimensional space
    - Often, clusters are hidden in subspaces of the data, i.e. only a subset of features is relevant for the clustering
    - E.g. a gene plays a certain role in a subset of experimental conditions
  - Local feature relevance/correlation
    - For each cluster, a different subset of features or a different correlation of features may be relevant
    - E.g. different genes are responsible for different phenotypes
  - Overlapping clusters
    - Clusters may overlap, i.e. an object may be clustered differently in varying subspaces
    - E.g. a gene plays different functional roles depending on the environment

- General problem setting of clustering high dimensional data

*Search for clusters in  
(in general arbitrarily oriented) subspaces  
of the original feature space*

- Challenges:
  - Find the correct subspace of each cluster
    - Search space:
      - all possible arbitrarily oriented subspaces of a feature space
      - infinite
  - Find the correct cluster in each relevant subspace
    - Search space:
      - “Best” partitioning of points (see: minimal cut of the similarity graph)
      - NP-complete [SCH75]

- Even worse: ***Circular Dependency***
    - Both challenges depend on each other
    - In order to determine the correct subspace of a cluster, we need to know (at least some) cluster members
    - In order to determine the correct cluster memberships, we need to know the subspaces of all clusters
  
  - How to solve the circular dependency problem?
    - Integrate subspace search into the clustering process
    - Thus, we need heuristics to solve
      - the clustering problem
      - the subspace search problem
- simultaneously***

# A First Taxonomy of Approaches

- One common assumption independent of the circular dependency problem:
  - Search space is restricted to axis-parallel subspaces only
- Thus, we can distinguish between
  - Approaches detecting clusters in axis-parallel subspaces
    - “subspace clustering algorithms”
    - “projected clustering algorithms”
    - “bi-clustering or co-clustering algorithms”
  - Approaches detecting clusters in arbitrarily oriented subspaces
    - “bi-clustering or co-clustering algorithms”
    - “pattern-based clustering algorithms”
    - “correlation clustering algorithms”

- Notes

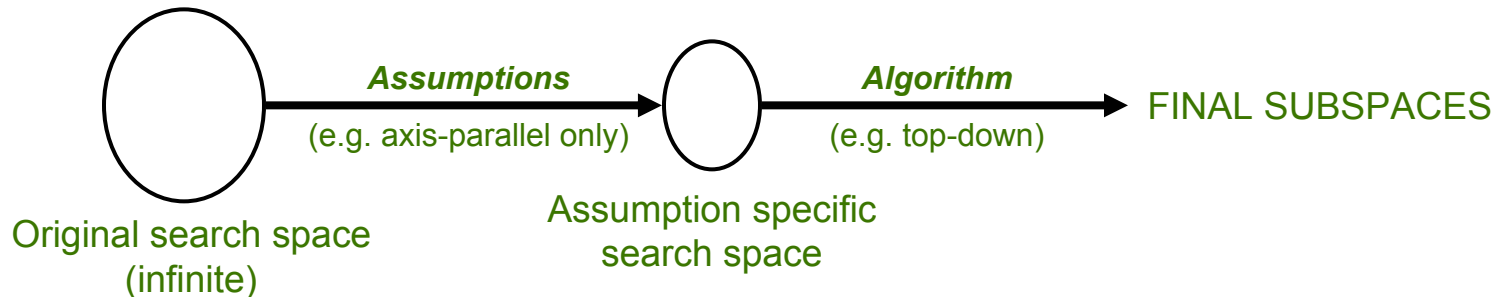
- This taxonomy considers only the subspace search space
- The clustering search space is also important
- Generally, other aspects for classifying existing approaches are e.g.
  - The underlying cluster model that usually involves
    - Input parameters
    - Assumptions on number, size, and shape of clusters
    - Noise (outlier) robustness
  - Determinism
  - Independence w.r.t. the order of objects/attributes
  - Assumptions on overlap/non-overlap of clusters/subspaces
  - Efficiency

# A First Taxonomy of Approaches

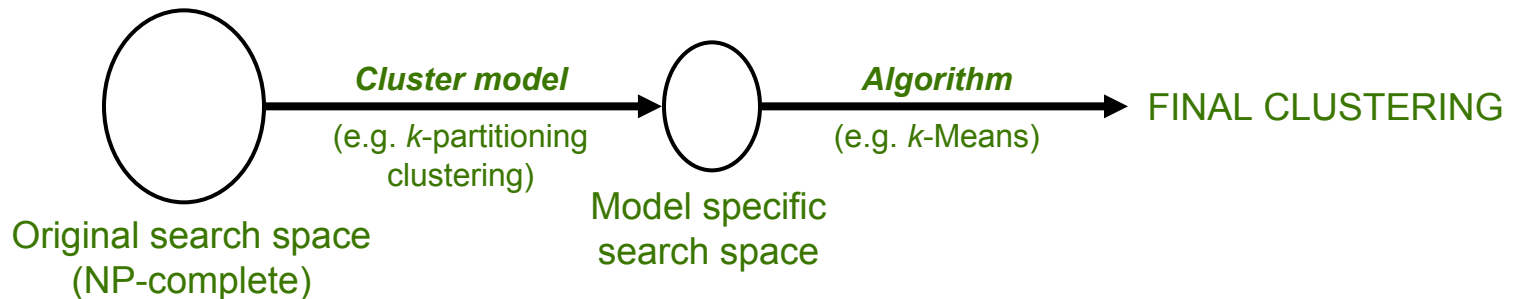
- A first big picture

- We have two problems to solve
- For both problems we need heuristics that have huge influence on the properties of the algorithms

- **Subspace search**



- **Cluster search**



1. Introduction
2. Axis-parallel Subspace Clustering
3. Pattern-based Clustering
4. Arbitrarily-oriented Subspace Clustering
5. Summary



# Outline:

## Axis-parallel Subspace Clustering

- Challenges and Approaches
- Bottom-up Algorithms
- Top-down Algorithms
- Summary

- We are searching for ***axis-parallel*** subspace clusters only
  - Overlapping clusters: points may be grouped differently in different subspaces
    - ⇒ “***subspace clustering***”
  - Disjoint partitioning: assign points uniquely to clusters (or noise)
    - ⇒ “***projected clustering***”

*Note: the terms **subspace clustering** and **projected clustering** are not used in a unified or consistent way in the literature*

- The naïve solution:
  - Given a cluster criterion, explore each possible subspace of a  $d$ -dimensional dataset whether it contains a cluster
  - Runtime complexity: depends on the search space, i.e. the number of all possible subspaces of a  $d$ -dimensional data set

- What is the number of all possible (axis-parallel) subspaces of a  $d$ -dimensional data set?
  - How many  $k$ -dimensional subspaces ( $k \leq d$ ) do we have?
    - A  $k$ -dimensional subspace has  $k$  relevant and  $d-k$  irrelevant attributes
    - The number of all  $k$ -tuples of a set of  $d$  elements is

$$\binom{d}{k}$$

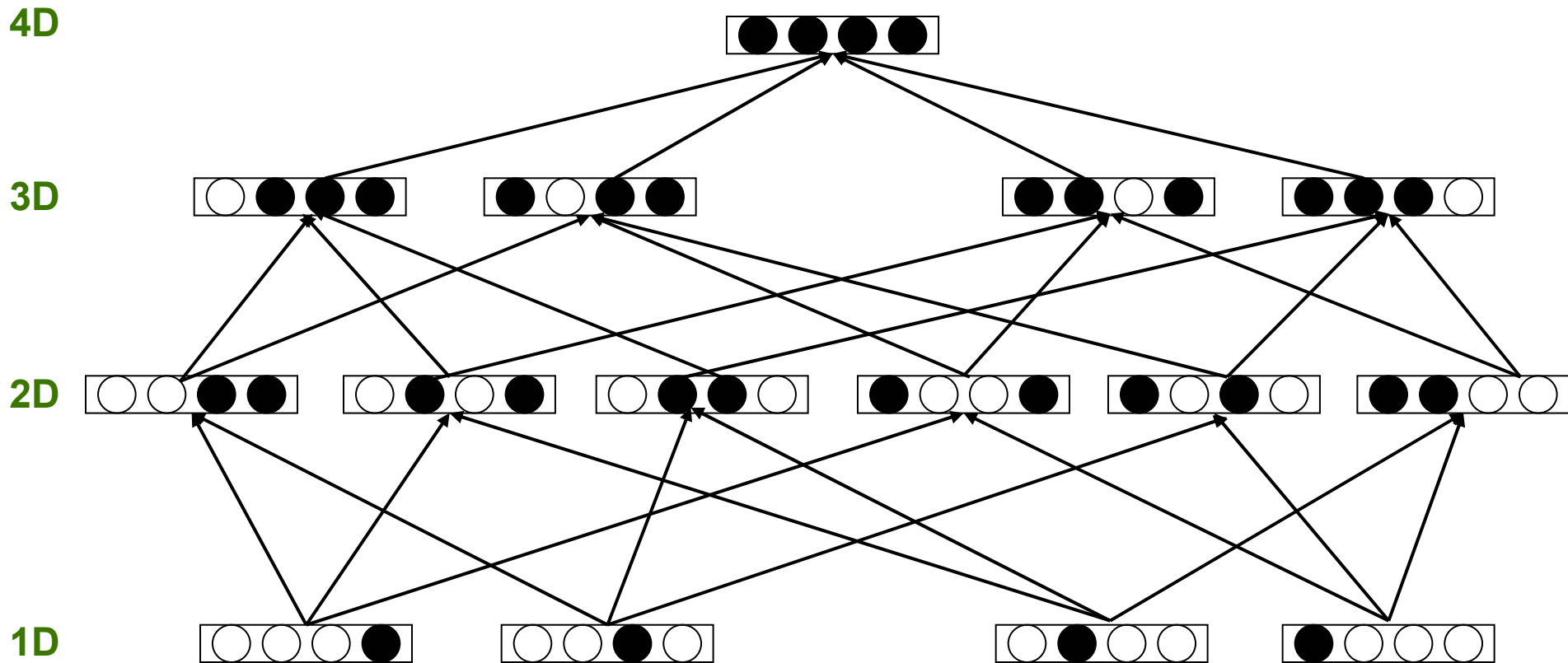
- Overall:

$$\sum_{k=1}^d \binom{d}{k} = 2^d - 1$$

- So the naïve solution is computationally infeasible:

**We face a runtime complexity of  $O(2^d)$**

- Search space for  $d = 4$



- Basically, existing approaches implement two different ways to efficiently navigate through the search space of possible subspaces
  - Bottom-up:
    - Start with 1D subspaces and iteratively generate higher dimensional ones using a “suitable” merging procedure
      - If the cluster criterion implements the downward closure property, one can use any bottom-up frequent itemset mining algorithm (e.g. APRIORI [AS94])
    - *Key*: downward-closure property OR merging procedure
  - Top-down:
    - The search starts in the full  $d$ -dimensional space and iteratively learns for each point or each cluster the correct subspace
    - *Key*: procedure to learn the correct subspace

- Rational:
  - Start with 1-dimensional subspaces and merge them to compute higher dimensional ones
  - Most approaches transfer the problem of subspace search into frequent item set mining
    - The cluster criterion must implement the downward closure property
      - If the criterion holds for any  $k$ -dimensional subspace  $S$ , then it also holds for any  $(k-1)$ -dimensional projection of  $S$
      - Use the reverse implication for pruning:  
If the criterion does not hold for a  $(k-1)$ -dimensional projection of  $S$ , then the criterion also does not hold for  $S$
    - Apply any frequent itemset mining algorithm (like APRIORI)
  - Few approaches use other search heuristics like best-first-search, greedy-search, etc.
    - Better average and worst-case performance
    - No guaranty on the completeness of results

- The key limitation: ***global density thresholds***
  - Usually, the cluster criterion relies on density
  - In order to ensure the downward closure property, the density threshold must be fixed
  - Consequence: the points in a 20-dimensional subspace cluster must be as dense as in a 2-dimensional cluster
  - This is a rather optimistic assumption since the data space grows exponentially with increasing dimensionality
  - Consequences:
    - A strict threshold will most likely produce only lower dimensional clusters
    - A loose threshold will most likely produce higher dimensional clusters but also a huge amount of (potentially meaningless) low dimensional clusters

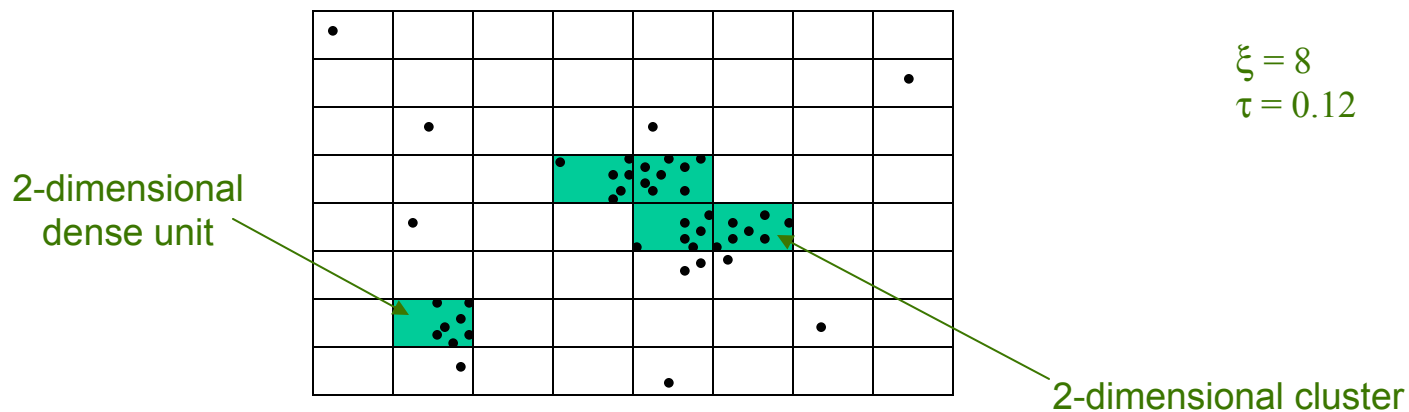
- Properties (APRIORI-style algorithms):
  - Generation of all clusters in all subspaces  $\Rightarrow$  overlapping clusters
  - Subspace clustering algorithms usually rely on bottom-up subspace search
  - Worst-case: complete enumeration of all subspaces, i.e.  $O(2^d)$  time
  - Complete results
- See some sample bottom-up algorithms coming up ...



- CLIQUE [AGGR98]

- Cluster model

- Each dimension is partitioned into  $\xi$  equi-sized intervals called units
- A  $k$ -dimensional unit is the intersection of  $k$  1-dimensional units (from different dimensions)
- A unit  $u$  is considered dense if the fraction of all data points in  $u$  exceeds the threshold  $\tau$
- A cluster is a maximal set of connected dense units

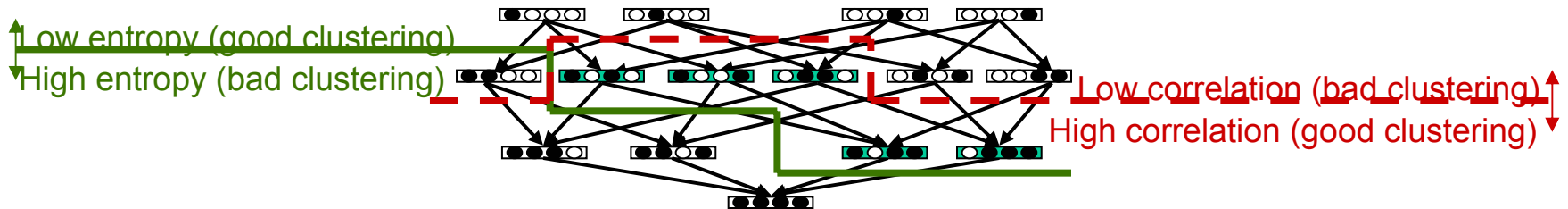


# Bottom-up Algorithms

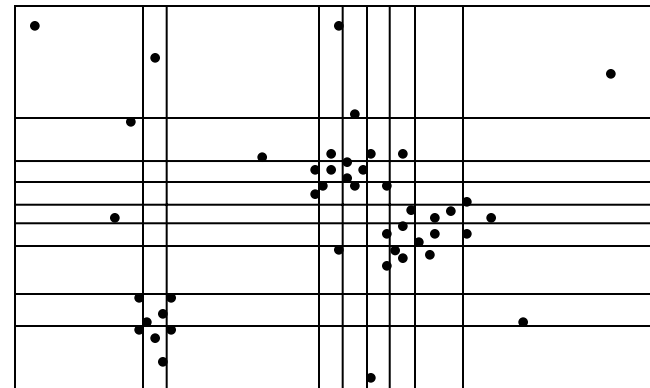
- Downward-closure property holds for dense units
- Algorithm
  - All dense cells are computed using APRIORI-style search
  - All connected dense units in a common subspace are merged to generate the subspace clusters
- Discussion
  - Input:  $\xi$  and  $\tau$  specifying the density threshold
  - Output: all clusters in all subspaces, clusters may overlap
  - Uses a fixed density threshold for all subspaces (in order to ensure the downward closure property)
  - Simple but efficient cluster model

related approaches:

- ENCLUS [CFZ99]
  - fixed grid similar to CLIQUE but subspaces are evaluated



- MAFIA [NGC01]
  - adaptive grid

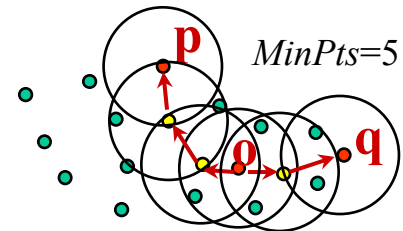
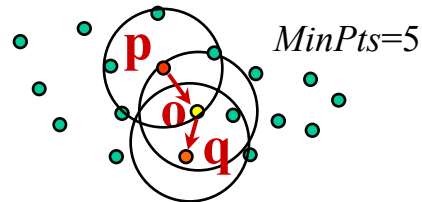
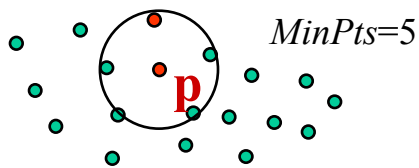


- P3C [MSE06]
  - computes cluster cores (hyper-rectangular approximations of subspace clusters) bottom-up from 1-dimensional intervals

- SUBCLU [KKK04]

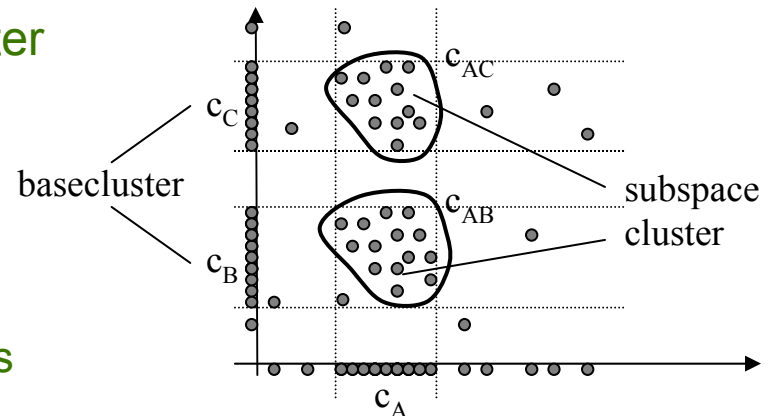
- Cluster model:

- Density-based cluster model of DBSCAN [EKSX96]
- Clusters are maximal sets of density-connected points
- Density connectivity is defined based on core points
- Core points have at least *minPts* points in their  $\epsilon$ -neighborhood



- Detects clusters of arbitrary size and shape (in the corresponding subspaces)
- Downward-closure property holds for sets of density-connected points

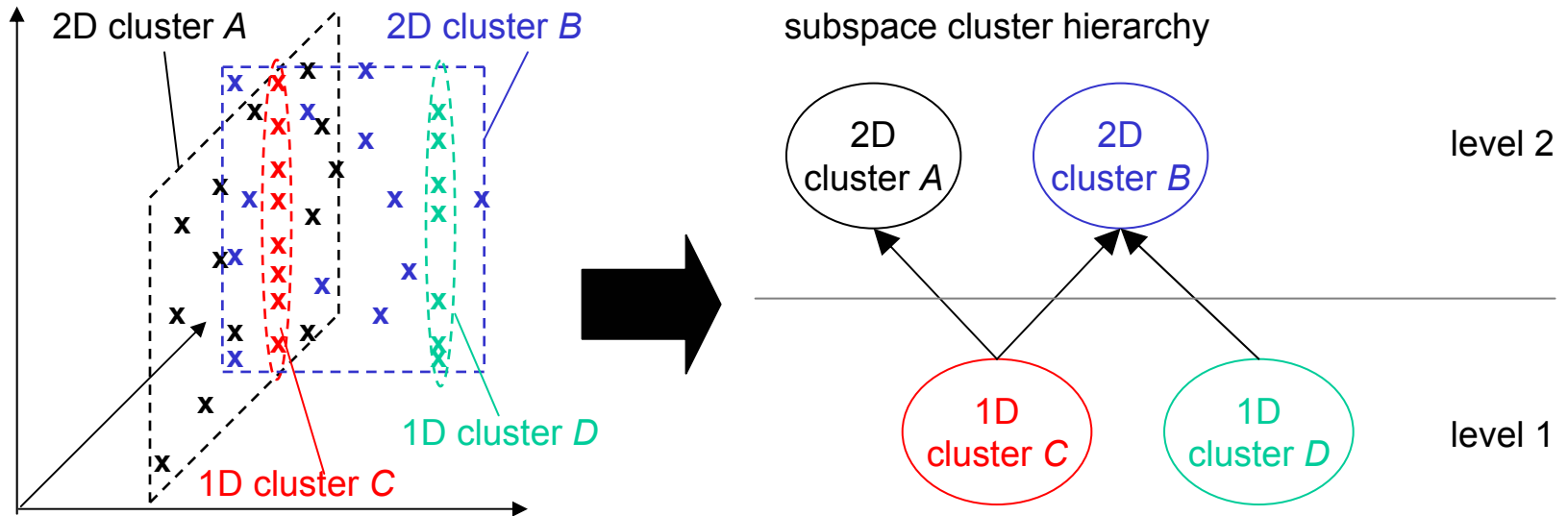
- FIRES [KKRW05]
  - different bottom-up heuristic for subspace search
  - 3-step algorithm
    - Starts with 1-dimensional clusters called *base clusters* (generated by applying any traditional clustering algorithm to each 1-dimensional subspace)
    - Merges these clusters to generate subspace cluster approximations by applying a clustering of the base clusters using a variant of DBSCAN (similarity between two clusters  $C1$  and  $C2$  is defined by  $|C1 \cap C2|$ )  
 ⇒ jump to maximal dimensional subspaces
    - Refines the resulting subspace cluster approximations
      - Apply any traditional clustering algorithm on the points within the approximations
      - Prune lower dimensional projections



- DiSH [ABK+07a]

- Idea:

- Not considered so far: lower dimensional clusters embedded in higher dimensional ones



- Now: find hierarchies of subspace clusters
- Integrate a proper distance function into hierarchical clustering
- learns distance function instance-based bottom-up

- Rational:
  - Cluster-based approach:
    - Learn the subspace of a cluster starting with **full-dimensional** clusters
    - Iteratively refine the cluster memberships of points and the subspaces of the cluster
  - Instance-based approach:
    - Learn for each point its subspace preference in the **full-dimensional** data space
    - The subspace preference specifies the subspace in which each point “clusters best”
    - Merge points having similar subspace preferences to generate the clusters

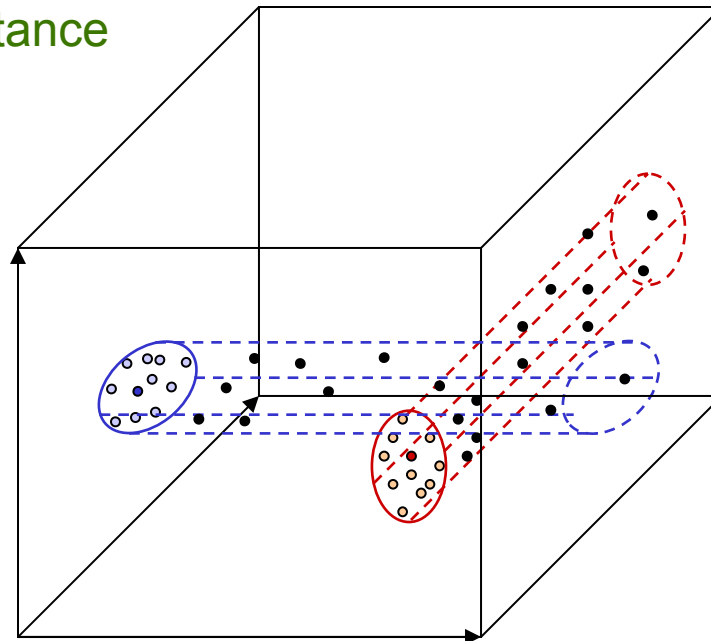
- The key problem: How should we learn the subspace preference of a cluster or a point?
  - Most approaches rely on the so-called “locality assumption”
    - The subspace is usually learned **from the local neighborhood** of cluster representatives/cluster members in the entire feature space:
      - Cluster-based approach: the **local neighborhood** of each cluster representative is evaluated in the  $d$ -dimensional space to learn the “correct” subspace of the cluster
      - Instance-based approach: the **local neighborhood** of each point is evaluated in the  $d$ -dimensional space to learn the “correct” subspace preference of each point
    - **The locality assumption**: the subspace preference can be learned from the **local neighborhood** in the  $d$ -dimensional space
  - Other approaches learn the subspace preference of a cluster or a point **from randomly sampled points**



- Discussion:
  - Locality assumption
    - Recall the effects of the curse of dimensionality on concepts like “local neighborhood”
    - The neighborhood will most likely contain a lot of noise points
  - Random sampling
    - The larger the number of total points compared to the number of cluster points is, the lower the probability that cluster members are sampled
  - Consequence for both approaches
    - The learning procedure is often misled by these noise points

- Properties:
  - Simultaneous search for the “best” partitioning of the data points and the “best” subspace for each partition  $\Rightarrow$  disjoint partitioning
  - Projected clustering algorithms usually rely on top-down subspace search
  - Worst-case:
    - Usually complete enumeration of all subspaces is avoided
    - Worst-case costs are typically in  $O(d^2)$
- See some sample top-down algorithms coming up ...

- PROCLUS [APW+99]
  - $K$ -medoid cluster model
    - Cluster is represented by its medoid
    - To each cluster a subspace (of relevant attributes) is assigned
    - Each point is assigned to the nearest medoid (where the distance to each medoid is based on the corresponding subspaces of the medoids)
    - Points that have a large distance to its nearest medoid are classified as noise



- DOC [PJAM02]
  - random sampling
- PreDeCon [BKKK04]
  - density-based cluster model  $\Rightarrow$  instance-based locality assumption
- COSA [FM04]
  - not a clustering but learns a weight-matrix (instance-based locality assumption)

- The big picture
  - Basic assumption:  
“subspace search space is limited to axis-parallel subspaces”
  - Algorithmic view:
    - Bottom-up subspace search
    - Top-down subspace search
  - Problem-oriented view:
    - Subspace clustering (overlapping clusters)
    - Projected clustering (disjoint partitions)

- How do both views relate?
  - Subspace clustering algorithms compute overlapping clusters
    - Many approaches compute all clusters in all subspaces
      - These methods usually implement a bottom-up search strategy à la itemset mining
      - These methods usually rely on global density thresholds to ensure the downward closure property
      - These methods usually do not rely on the locality assumption
      - These methods usually have a worst case complexity of  $O(2^d)$
    - Other focus on maximal dimensional subspace clusters
      - These methods usually implement a bottom-up search strategy based on simple but efficient heuristics
      - These methods usually do not rely on the locality assumption
      - These methods usually have a worst case complexity of at most  $O(d^2)$

- How do both views relate?
  - Projected clustering algorithms compute a disjoint partitioning of the data
    - They usually implement a top-down search strategy
    - They usually rely on the locality assumption
    - They usually do not rely on global density thresholds
    - They usually scale at most quadratic in the number of dimensions

1. Introduction
2. Axis-parallel Subspace Clustering
3. Pattern-based Clustering
4. Arbitrarily-oriented Subspace Clustering
5. Summary

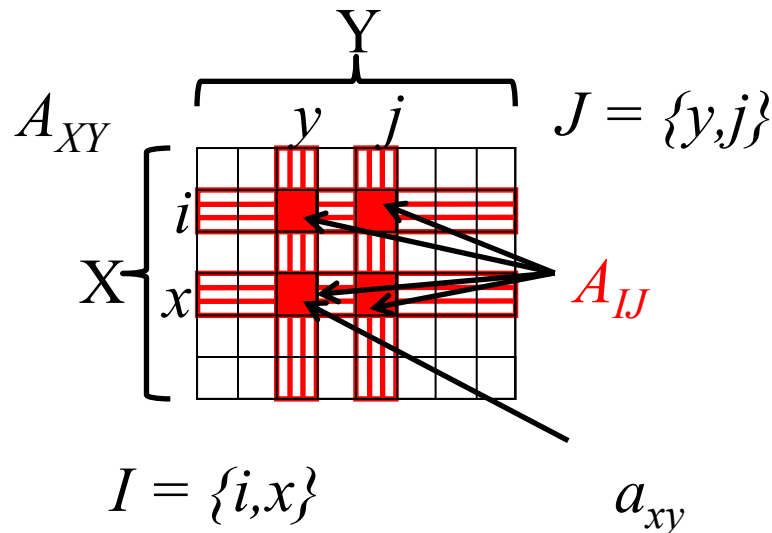


# Outline: Pattern-based Clustering

- Challenges and Approaches, Basic Models
  - Constant Biclusters
  - Biclusters with Constant Values in Rows or Columns
  - Pattern-based Clustering: Biclusters with Coherent Values
  - Biclusters with Coherent Evolutions
- Algorithms for
  - Constant Biclusters
  - Pattern-based Clustering: Biclusters with Coherent Values
- Summary

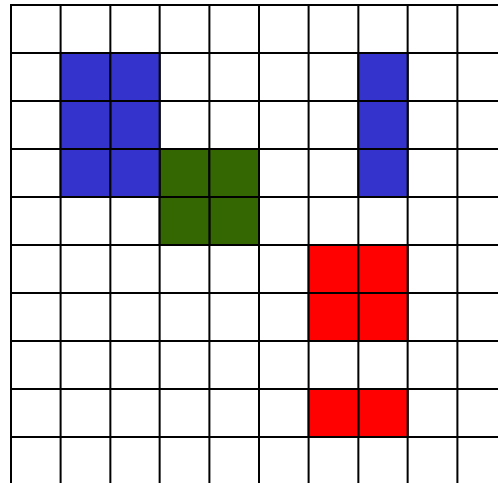
Pattern-based clustering relies on patterns in the data matrix:

- Simultaneous clustering of rows and columns of the data matrix (hence *biclustering*).
  - Data matrix  $\mathbf{A} = (X, Y)$  with set of rows  $X$  and set of columns  $Y$
  - $a_{xy}$  is the element in row  $x$  and column  $y$ .
  - submatrix  $A_{IJ} = (I, J)$  with subset of rows  $I \subseteq X$  and subset of columns  $J \subseteq Y$  contains those elements  $a_{ij}$  with  $i \in I$  und  $j \in J$



General aim of biclustering approaches:

- Find a set of submatrices  $\{(I_1, J_1), (I_2, J_2), \dots, (I_k, J_k)\}$  of the matrix  $A=(X, Y)$  (with  $I_i \subseteq X$  and  $J_i \subseteq Y$  for  $i = 1, \dots, k$ ) where each submatrix (= bicluster) meets a given homogeneity criterion.



- Some values often used by bicluster models:

- mean of row  $i$ :

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}$$

- mean of column  $j$ :

$$a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$$

- mean of all elements:

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij}$$

$$= \frac{1}{|J|} \sum_{j \in J} a_{Ij}$$

$$= \frac{1}{|I|} \sum_{i \in I} a_{iJ}$$

- Different types of biclusters (cf. [MO04]):
  - constant biclusters
  - biclusters with
    - constant values on columns
    - constant values on rows
  - biclusters with coherent values (aka. pattern-based clustering)
  - biclusters with coherent evolutions

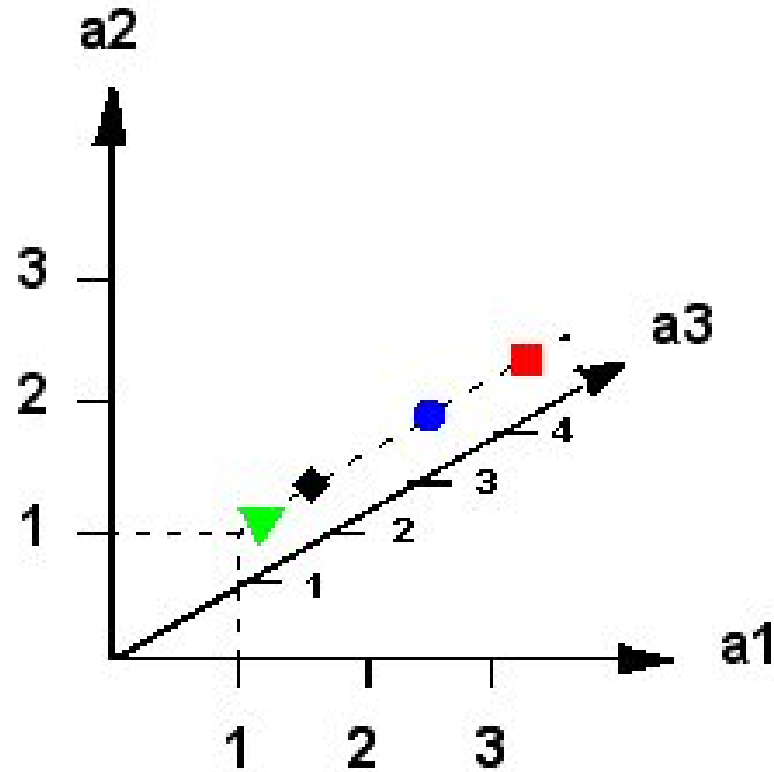
- Constant biclusters
  - all points share an identical value in selected attributes.
  - The constant value  $\mu$  is a typical value for the cluster.
  - Cluster model:

$$a_{ij} = \mu$$

- Obviously a special case of an axis-parallel subspace cluster.

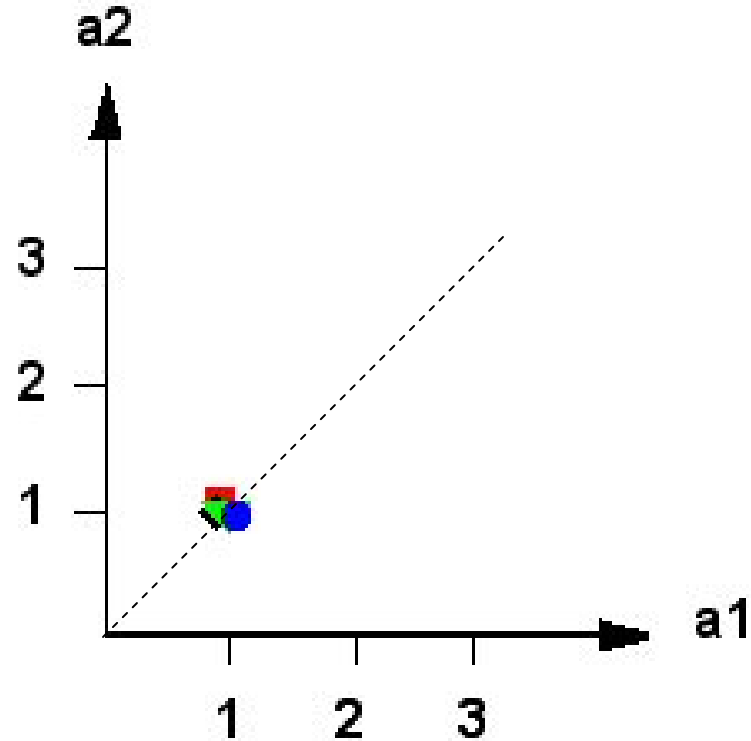
- example – embedding 3-dimensional space:

	a1	a2	a3
P1	1	1	3.5
P2	1	1	2.3
P3	1	1	0.2
P4	1	1	0.7



- example – 2-dimensional subspace:

	a1	a2
P1	1	1
P2	1	1
P3	1	1
P4	1	1

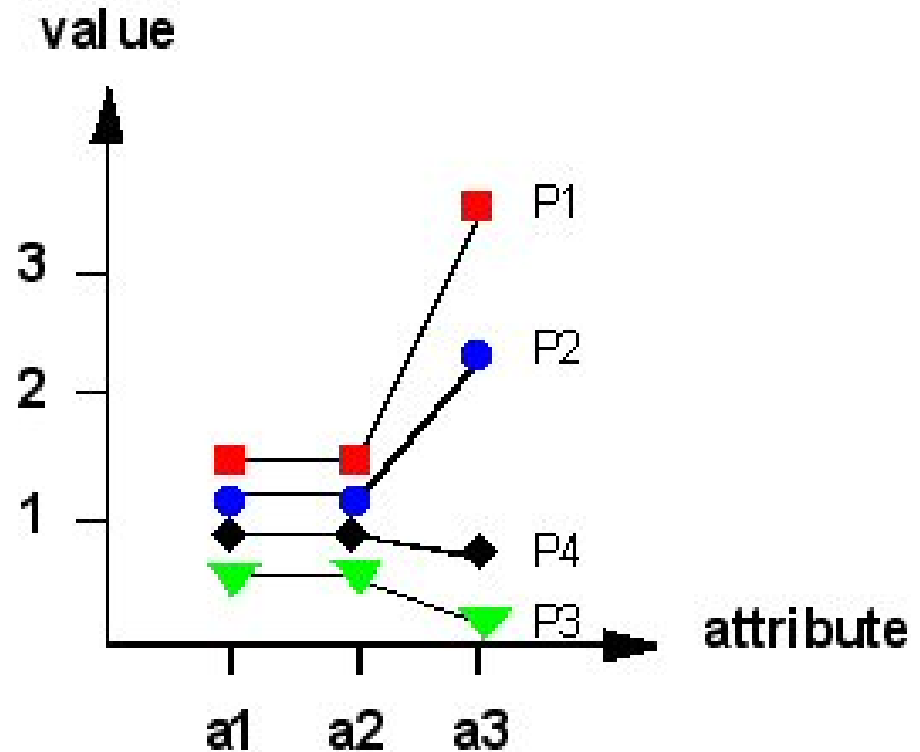


⇒ points located on the bisecting line of participating attributes



- example – transposed view of attributes:

	a1	a2	a3
P1	1	1	3.5
P2	1	1	2.3
P3	1	1	0.2
P4	1	1	0.7



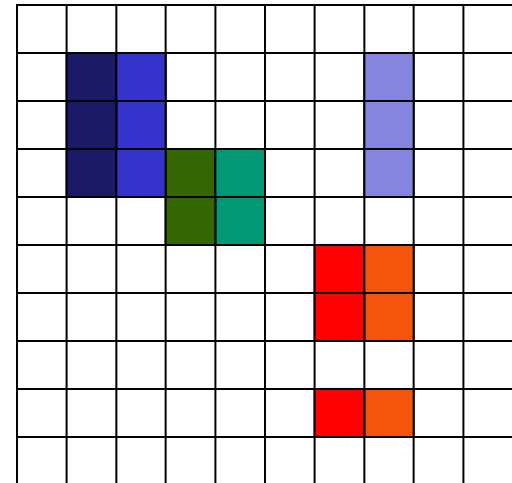
⇒ pattern: identical constant lines

- Biclusters with constant values on columns
  - Cluster model for  $\mathbf{A}_{I,J} = (I,J)$ :

$$a_{ij} = \mu + c_j$$

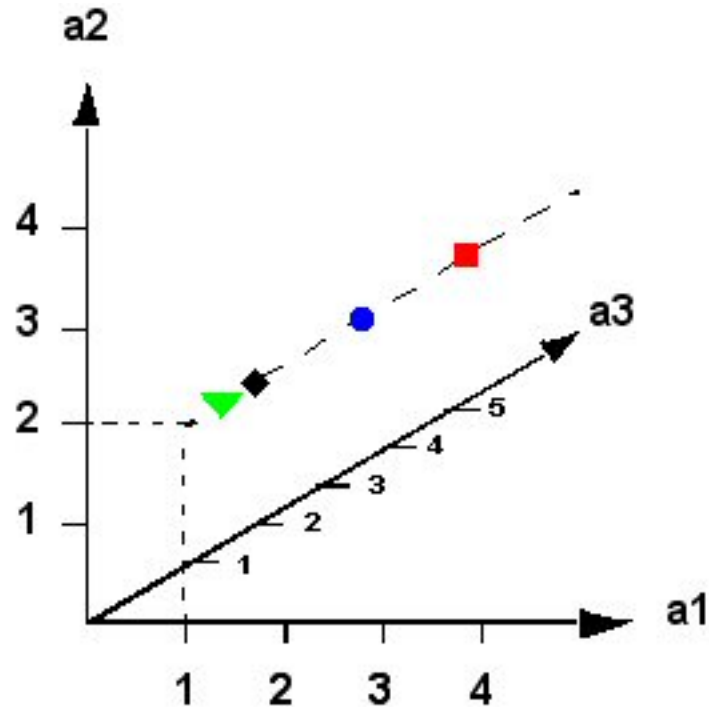
$$\forall i \in I, j \in J$$

- adjustment value  $c_j$  for column  $j \in J$
- results in axis-parallel subspace clusters



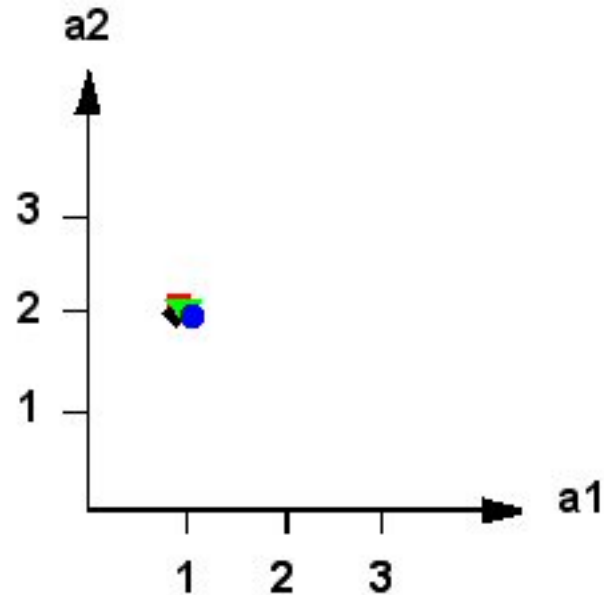
- example – 3-dimensional embedding space:

	a1	a2	a3
P1	1	2	3.5
P2	1	2	2.3
P3	1	2	0.2
P4	1	2	0.7



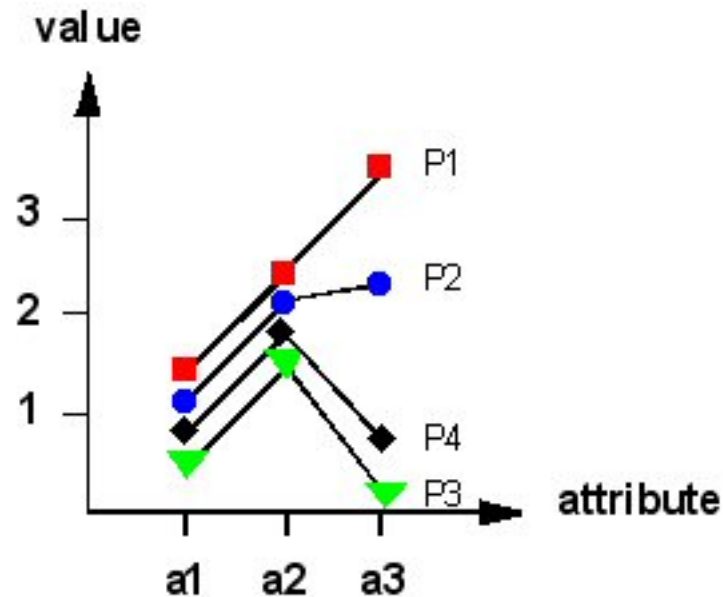
- example – 2-dimensional subspace:

	a1	a2
P1	1	2
P2	1	2
P3	1	2
P4	1	2



- example – transposed view of attributes:

	a1	a2	a3
P1	1	2	3.5
P2	1	2	2.3
P3	1	2	0.2
P4	1	2	0.7



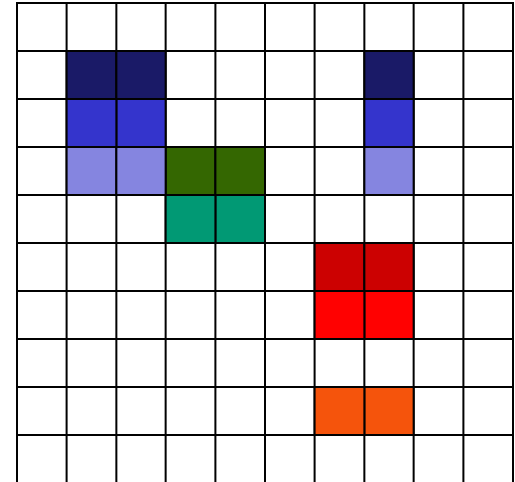
⇒ pattern: identical lines

- Biclusters with constant values on rows
  - Cluster model for  $\mathbf{A}_{I,J} = (I,J)$ :

$$a_{ij} = \mu + r_i$$

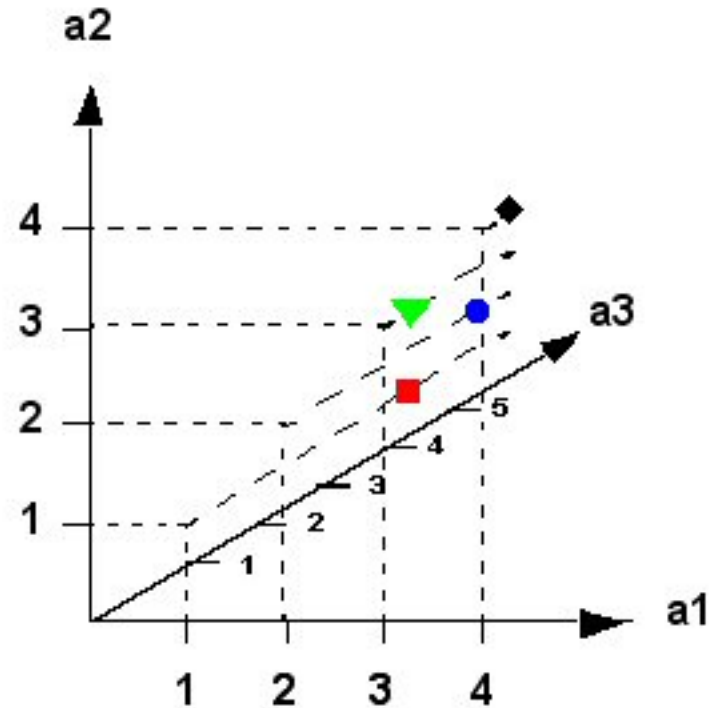
$$\forall i \in I, j \in J$$

- adjustment value  $r_i$  for row  $i \in I$



- example – 3-dimensional embedding space:

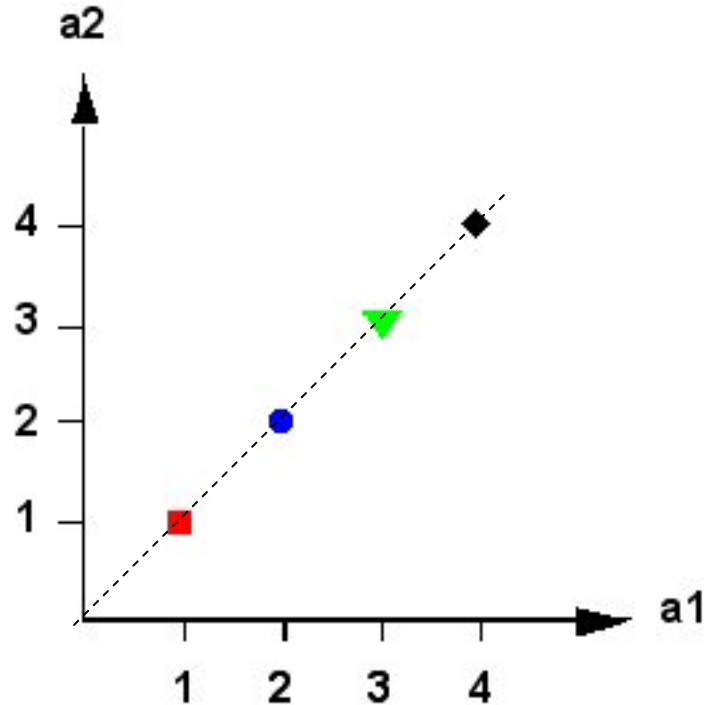
	a1	a2	a3
P1	1	1	3.5
P2	2	2	2.3
P3	3	3	0.2
P4	4	4	0.7



⇒ in the embedding space, points build a sparse hyperplane parallel to irrelevant axes

- example – 2-dimensional subspace:

	a1	a2
P1	1	1
P2	2	2
P3	3	3
P4	4	4

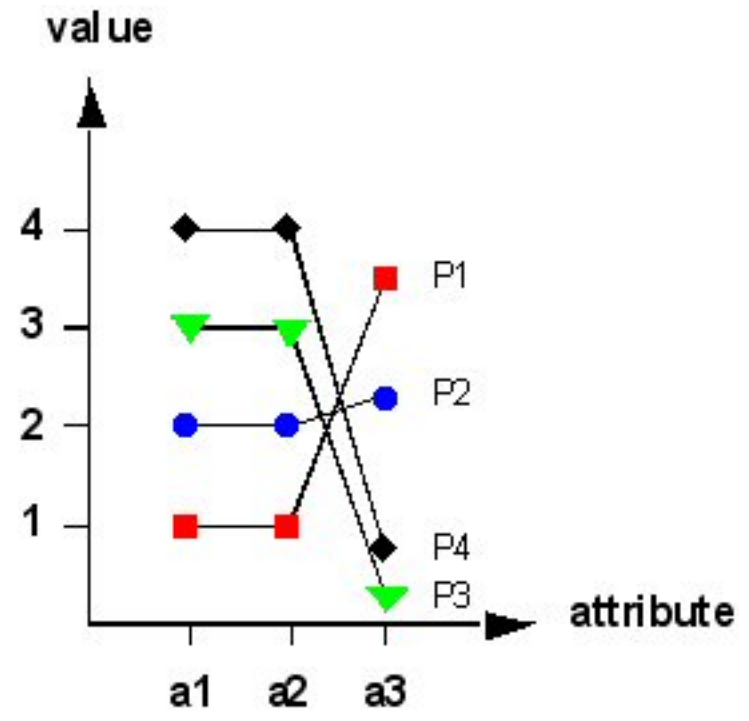


⇒ points are accommodated on the bisecting line of participating attributes



- example – transposed view of attributes:

	a1	a2	a3
P1	1	1	3.5
P2	2	2	2.3
P3	3	3	0.2
P4	4	4	0.7



⇒ pattern: parallel constant lines

- Biclusters with coherent values

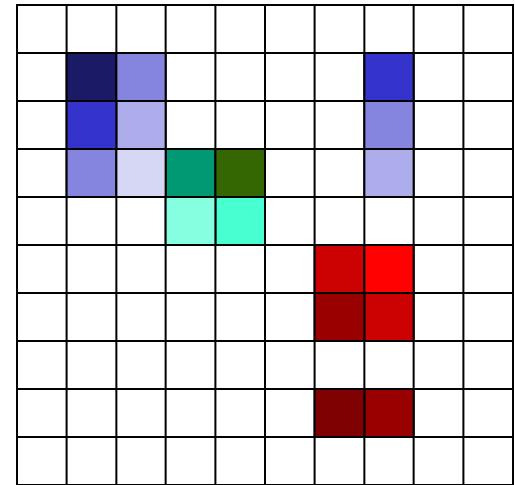
- based on a particular form of covariance between rows and columns

$$a_{ij} = \mu + r_i + c_j$$

$$\forall i \in I, j \in J$$

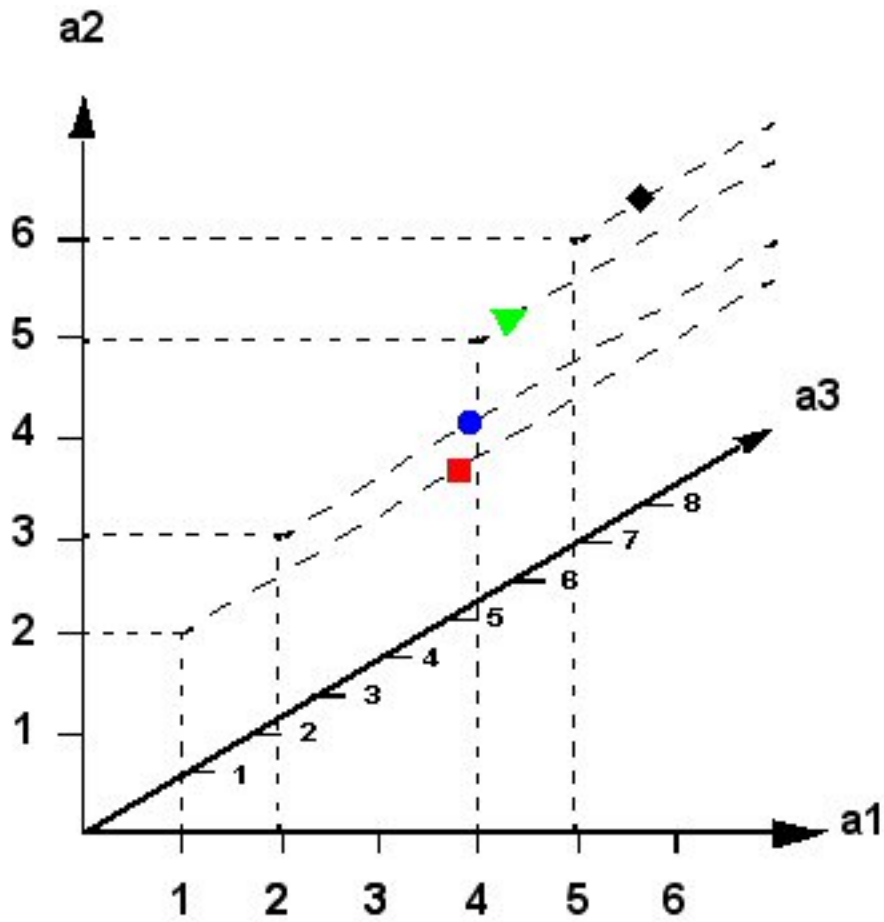
- special cases:

- $c_j = 0$  for all  $j \Rightarrow$  constant values on rows
- $r_i = 0$  for all  $i \Rightarrow$  constant values on columns



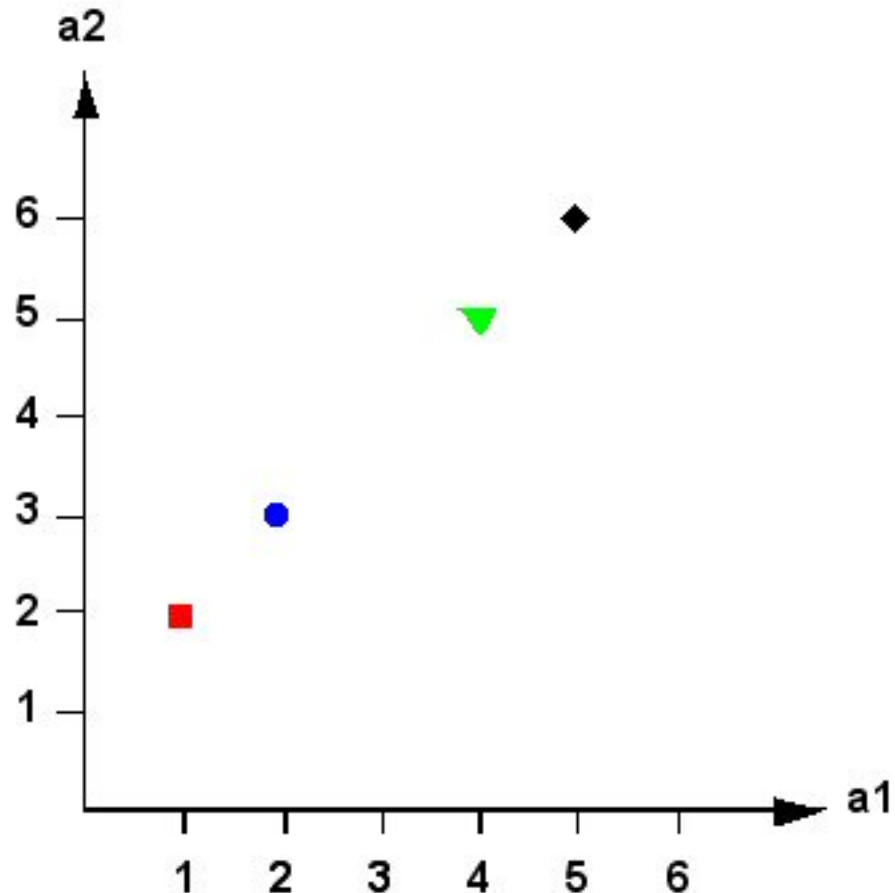
- embedding space: hyperplane parallel to axes of irrelevant attributes

	a1	a2	a3
P1	1	2	3.5
P2	2	3	2.3
P3	4	5	0.2
P4	5	6	0.7



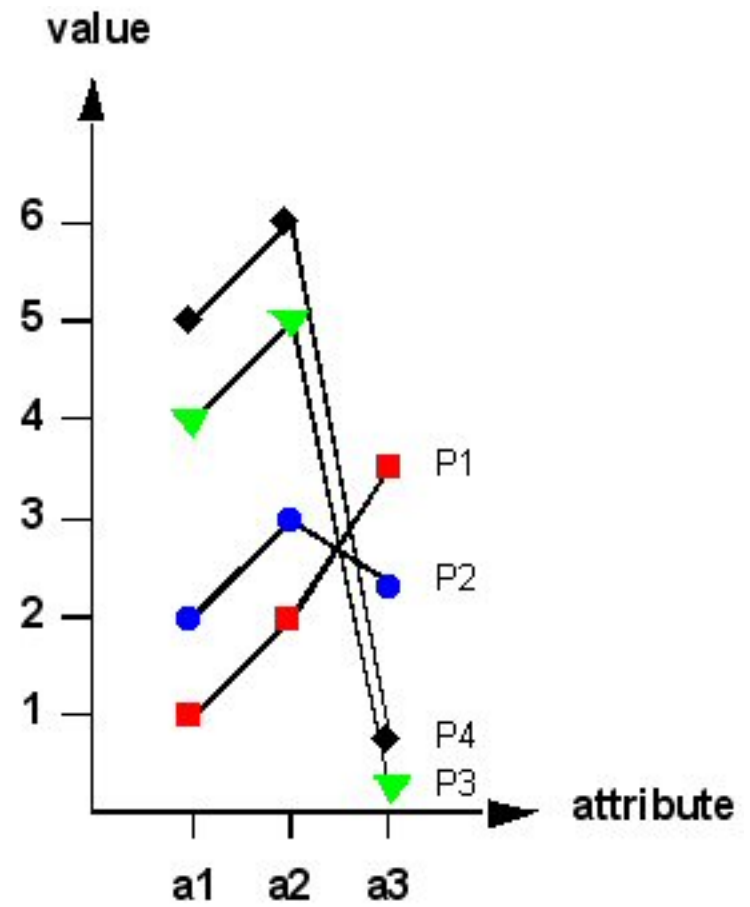
- subspace: increasing one-dimensional line

	a1	a2
P1	1	2
P2	2	3
P3	4	5
P4	5	6



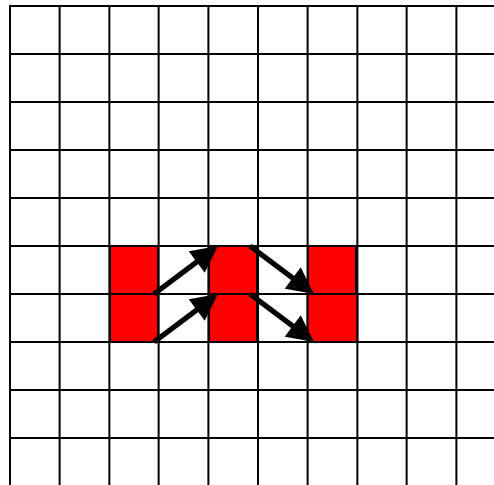
- transposed view of attributes:

	a1	a2	a3
P1	1	2	3.5
P2	2	3	2.3
P3	4	5	0.2
P4	5	6	0.7



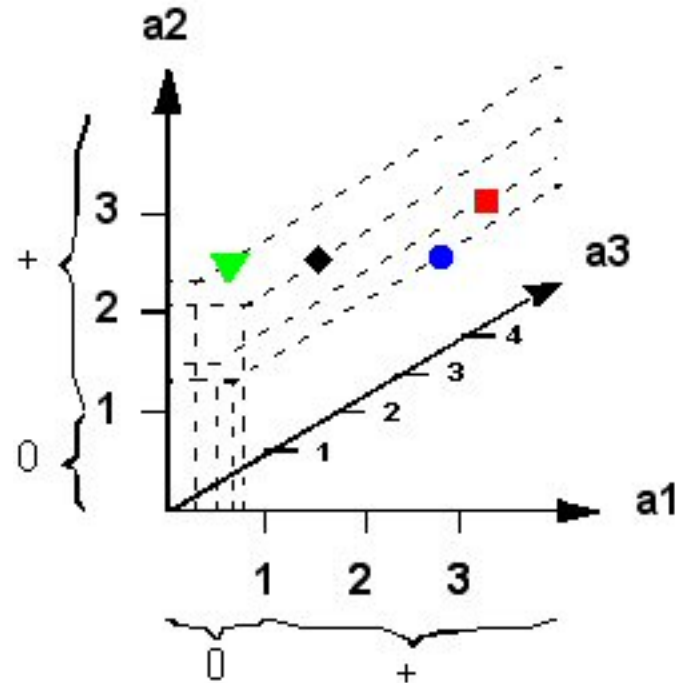
⇒ pattern: parallel lines

- Biclusters with coherent evolutions
  - for all rows, all pairs of attributes change simultaneously
    - discretized attribute space: coherent state-transitions
    - change in same direction irrespective of the quantity



- Approaches with coherent state-transitions: [TSS02,MK03]
  - reduces the problem to grid-based axis-parallel approach:

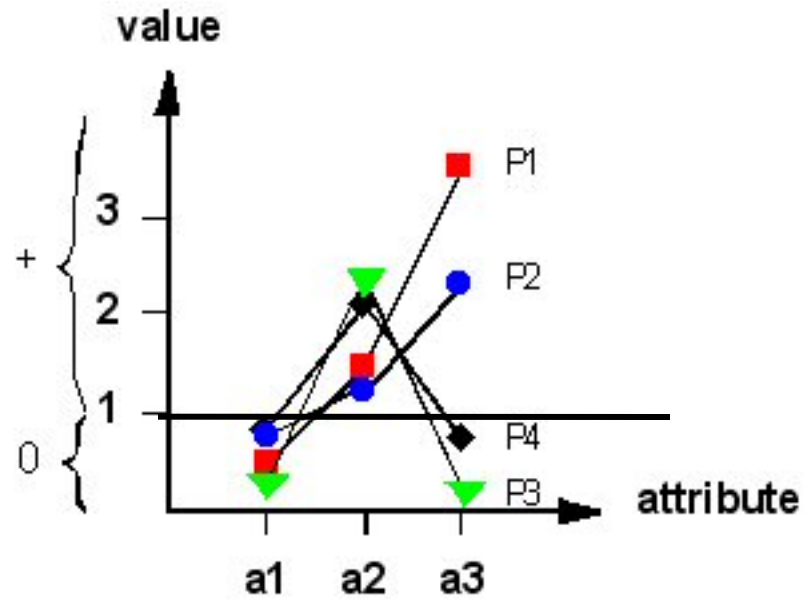
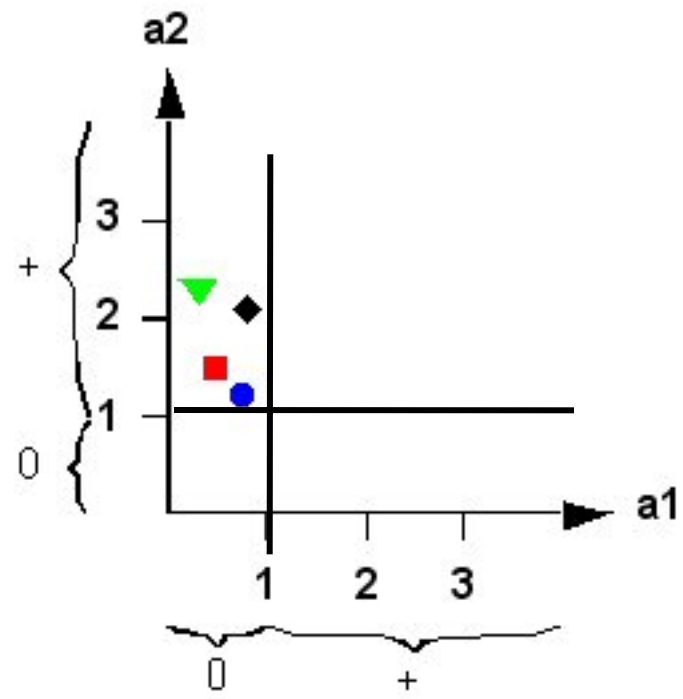
	a1	a2	a3
P1	0.5	1.5	3.5
P2	0.7	1.3	2.3
P3	0.3	2.3	0.2
P4	0.8	2.1	0.7



# Challenges and Approaches, Basic Models

	a1	a2
P1	0	+
P2	0	+
P3	0	+
P4	0	+

	a1	a2	a3
P1	0.5	1.5	3.5
P2	0.7	1.3	2.3
P3	0.3	2.3	0.2
P4	0.8	2.1	0.7



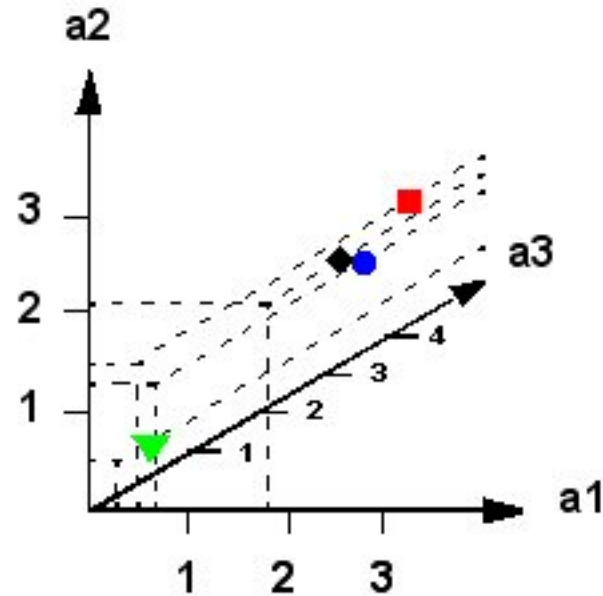
pattern: all lines cross border between states (in the same direction)



- change in same direction – general idea: find a subset of rows and columns, where a permutation of the set of columns exists such that the values in every row are increasing
- clusters do not form a subspace but rather half-spaces
- related approaches:
  - quantitative association rule mining [Web01,RRK04,GRRK05]
  - adaptation of formal concept analysis [GW99] to numeric data [Pfa07]

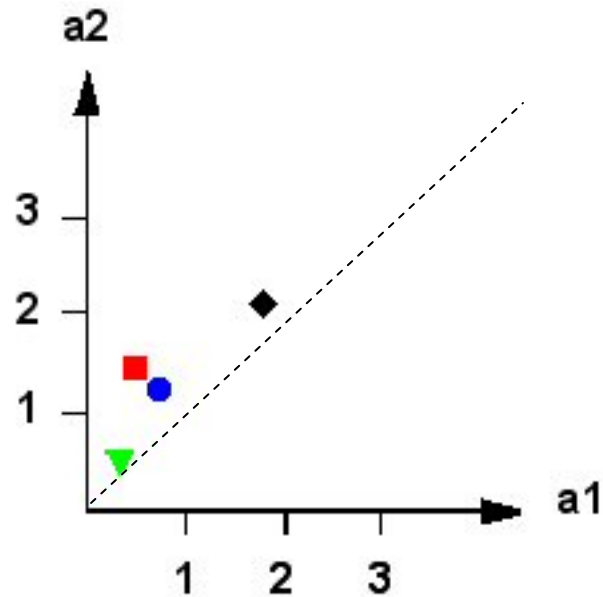
- example – 3-dimensional embedding space

	a1	a2	a3
P1	0.5	1.5	3.5
P2	0.7	1.3	2.3
P3	0.3	0.5	0.2
P4	1.8	2.1	0.7



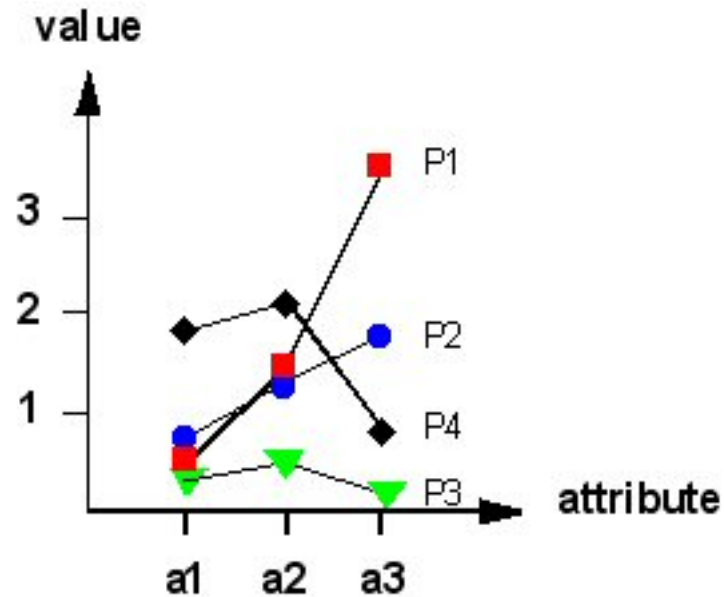
- example – 2-dimensional subspace

	a1	a2
P1	0.5	1.5
P2	0.7	1.3
P3	0.3	0.5
P4	1.8	2.1



- example – transposed view of attributes

	a1	a2	a3
P1	0.5	1.5	3.5
P2	0.7	1.3	2.3
P3	0.3	0.5	0.2
P4	1.8	2.1	0.7



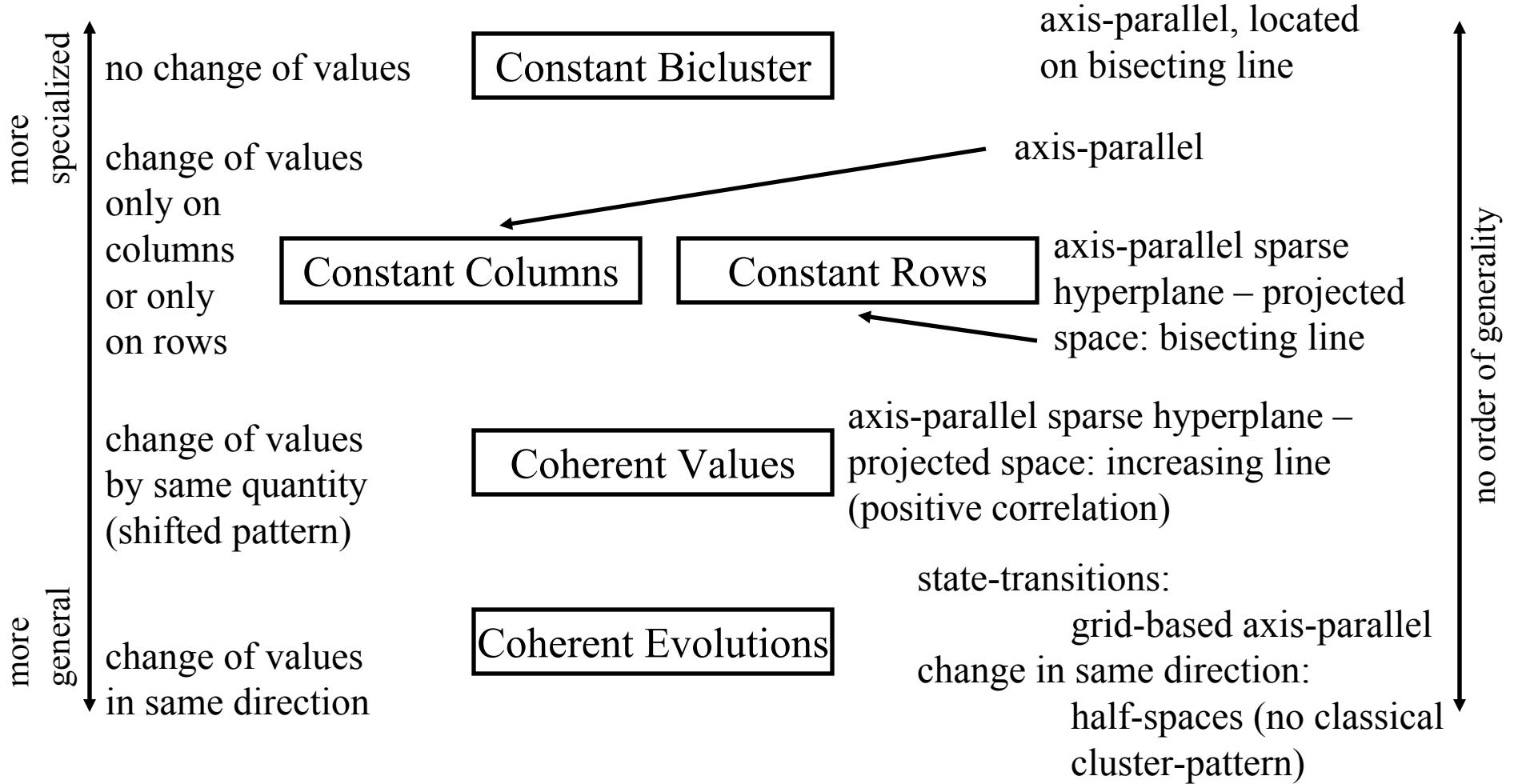
⇒ pattern: all lines increasing

# Challenges and Approaches, Basic Models

Matrix-Pattern

Bicluster Model

Spatial Pattern



- Algorithms for Constant Biclusters

- classical problem statement by Hartigan [Har72]
- quality measure for a bicluster: variance of the submatrix  $A_{IJ}$ :

$$VAR(A_{IJ}) = \sum_{i \in I, j \in J} (a_{ij} - a_{IJ})^2$$

- recursive split of data matrix into two partitions
- each split chooses the maximal reduction in the overall sum of squares for all biclusters
- avoids partitioning into  $|X| \cdot |Y|$  singularity-biclusters (optimizing the sum of squares) by comparing the reduction with the reduction expected by chance

- Algorithms for Biclusters with Constant Values in Rows or Columns
  - simple approach: normalization to transform the biclusters into constant biclusters and follow the first approach (e.g. [GLD00])
  - some application-driven approaches with special assumptions in the bioinformatics community (e.g. [CST00,SMD03,STG+01])
  - constant values on columns: general axis-parallel subspace/projected clustering
  - constant values on rows: special case of general correlation clustering
  - both cases special case of approaches to biclusters with coherent values

- Pattern-based Clustering: Algorithms for Biclusters with Coherent Values
  - classical approach: Cheng&Church [CC00]
    - introduced the term biclustering to analysis of gene expression data
    - quality of a bicluster: *mean squared residue* value  $H$

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

- submatrix (I,J) is considered a bicluster, if  $H(I, J) < \delta$



- $\delta = 0 \Rightarrow$  *perfect* bicluster:
  - each row and column exhibits absolutely consistent bias
  - bias of row  $i$  w.r.t. other rows:

$$a_{iJ} - a_{IJ}$$

- the model for a perfect bicluster predicts value  $a_{ij}$  by a row-constant, a column-constant, and an overall cluster-constant:

$$a_{ij} = a_{iJ} + a_{Ij} - a_{IJ}$$

$$\Updownarrow \mu = a_{IJ}, r_i = a_{iJ} - a_{IJ}, c_j = a_{Ij} - a_{IJ}$$

$$a_{ij} = \mu + r_i + c_j$$

- for a non-perfect bicluster, the prediction of the model deviates from the true value by a residue:

$$a_{ij} = \text{res}(a_{ij}) + a_{iJ} + a_{Ij} - a_{IJ}$$



$$\text{res}(a_{ij}) = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}$$

- This residue is the optimization criterion:

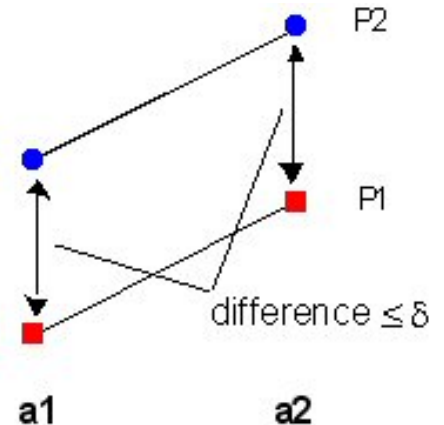
$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

– related approaches:

- p-cluster model [WWYY02]

- specializes  $\delta$ -bicluster-property to a pairwise property of two objects in two attributes:

$$\left| (a_{i_1 j_1} - a_{i_1 j_2}) - (a_{i_2 j_1} - a_{i_2 j_2}) \right| \leq \delta$$



- submatrix (I,J) is a  $\delta$ -p-cluster if this property is fulfilled for any 2x2 submatrix  $(\{i_1, i_2\}, \{j_1, j_2\})$  where  $\{i_1, i_2\} \in I$  and  $\{j_1, j_2\} \in J$ .

- FLOC [YWWY02]: randomized procedure
- MaPle [PZC+03]: improved pruning
- CoClus [CDGS04]: *k*-means-like approach

- Biclustering models do not fit exactly into the spatial intuition behind subspace, projected, or correlation clustering.
- Models make sense in view of a data matrix.
- Strong point: the models generally do not rely on the locality assumption.
- Models differ substantially  $\Rightarrow$  fair comparison is a non-trivial task.
- Comparison of five methods: [PBZ+06]
- Rather specialized task – comparison in a broad context (subspace/projected/correlation clustering) is desirable.
- Biclustering performs generally well on microarray data – for a wealth of approaches see [MO04].

1. Introduction
2. Axis-parallel Subspace Clustering
3. Pattern-based Clustering
4. Arbitrarily-oriented Subspace Clustering
5. Summary



## Outline:

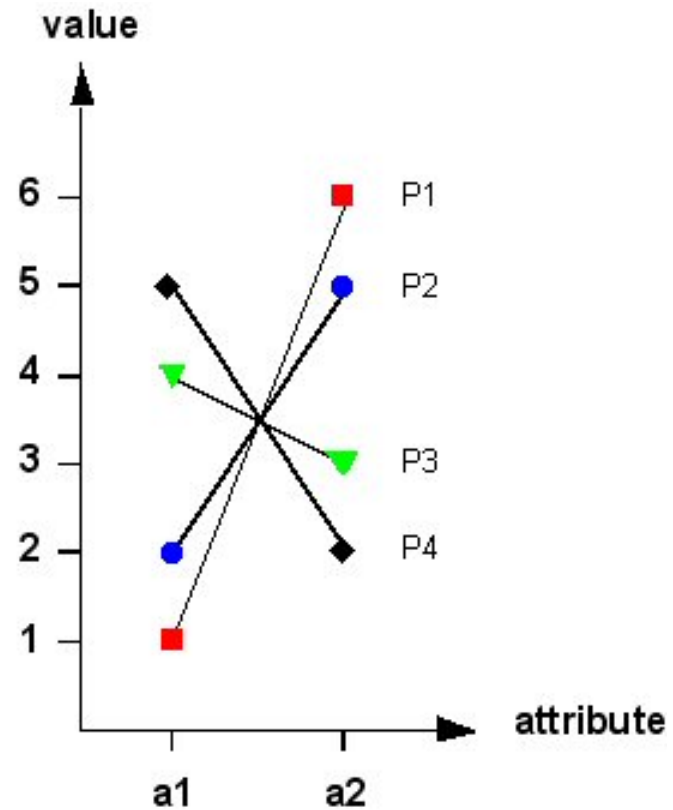
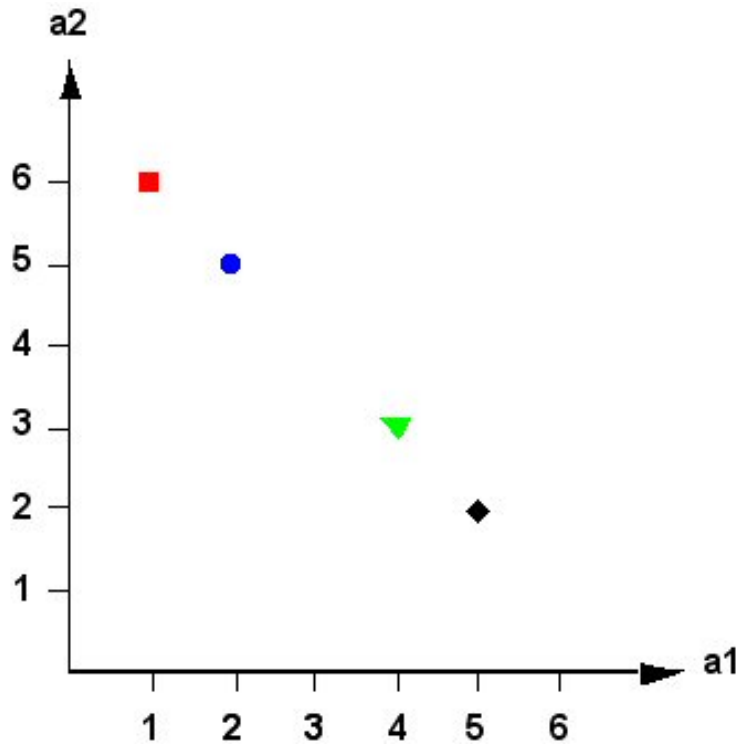
# Arbitrarily-oriented Subspace Clustering



- Challenges and Approaches
- Correlation Clustering Algorithms
- Summary and Perspectives

# Challenges and Approaches

- Pattern-based approaches find simple positive correlations
- negative correlations: no additive pattern

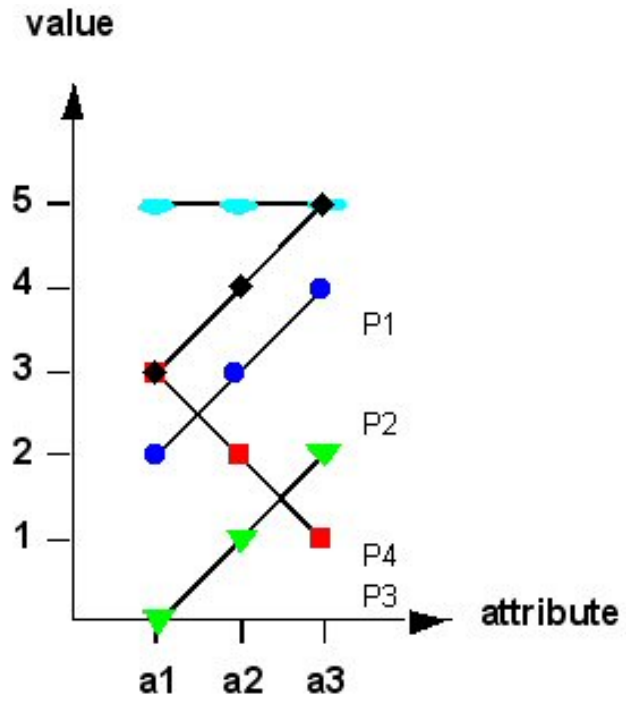
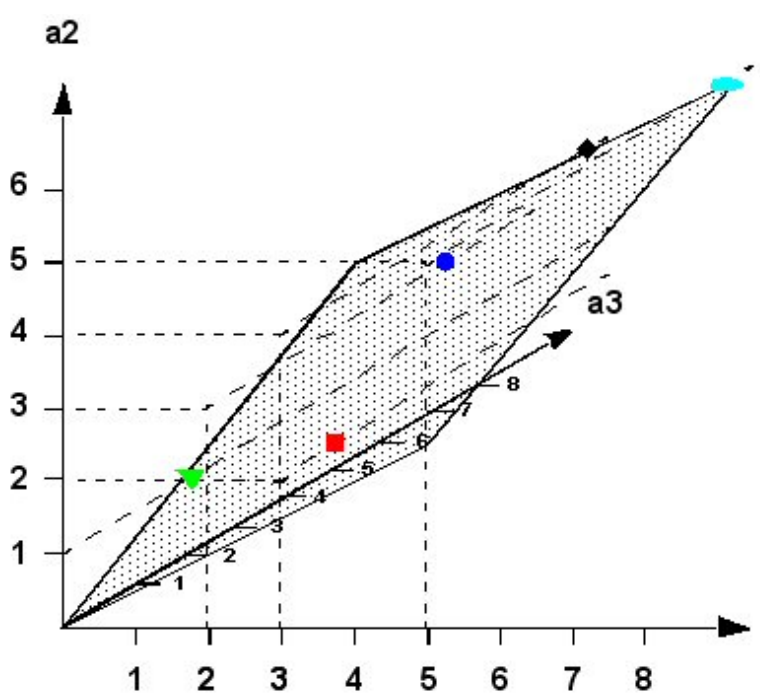


# Challenges and Approaches

- more complex correlations: out of scope of pattern-based approaches

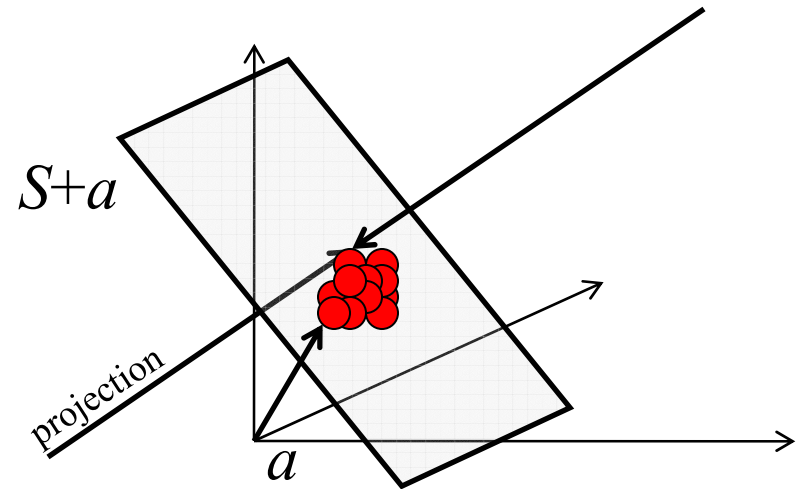
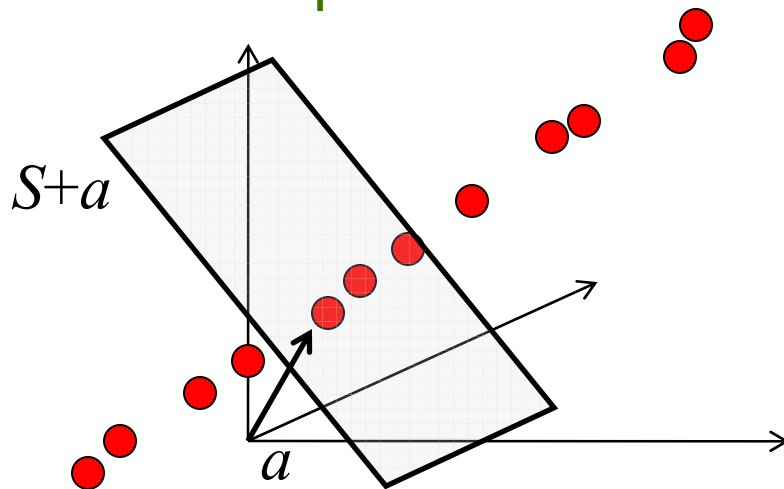
$$a1 - 2 \cdot a2 + a3 = 0$$

	a1	a2	a3
P1	3	2	1
P2	2	3	4
P3	0	1	2
P4	3	4	5
P5	5	5	6



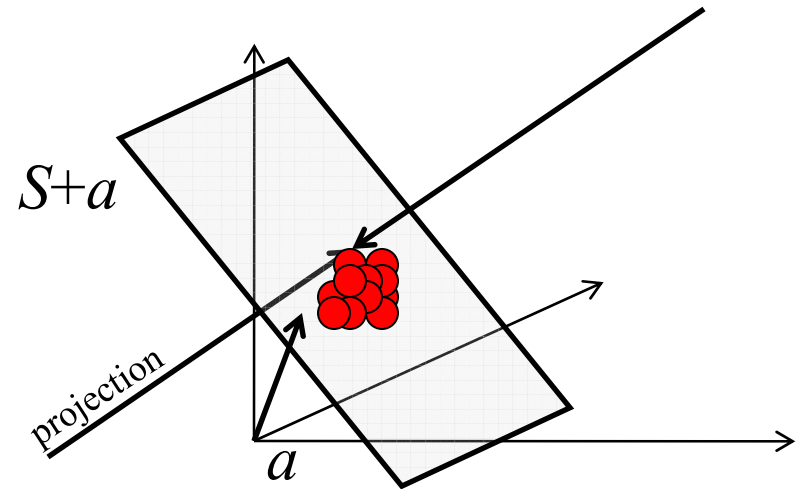
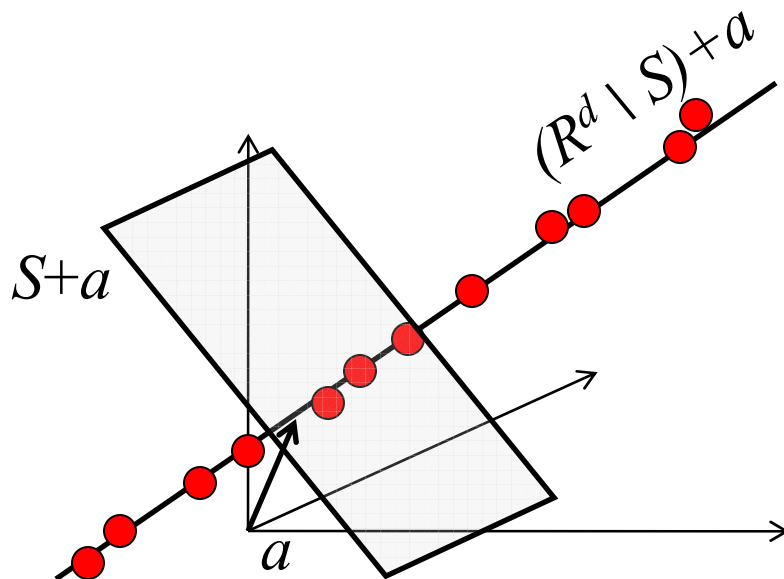


- Pattern-based approaches find simple positive correlations
- More general approach: oriented clustering aka. generalized subspace/projected clustering aka. correlation clustering
  - Note: different notion of “Correlation Clustering” in machine learning community, e.g. cf. [BBC04]
- Assumption: any cluster is located in an arbitrarily oriented affine subspace  $S+a$  of  $R^d$

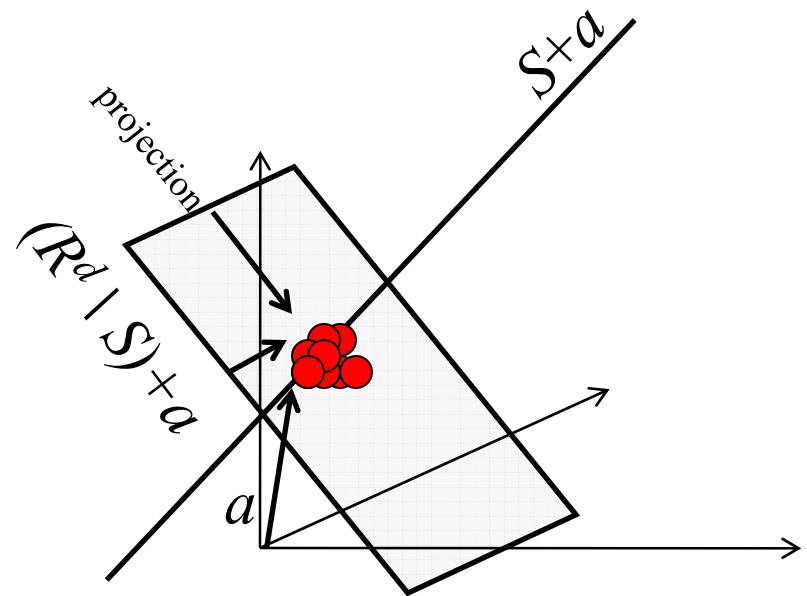
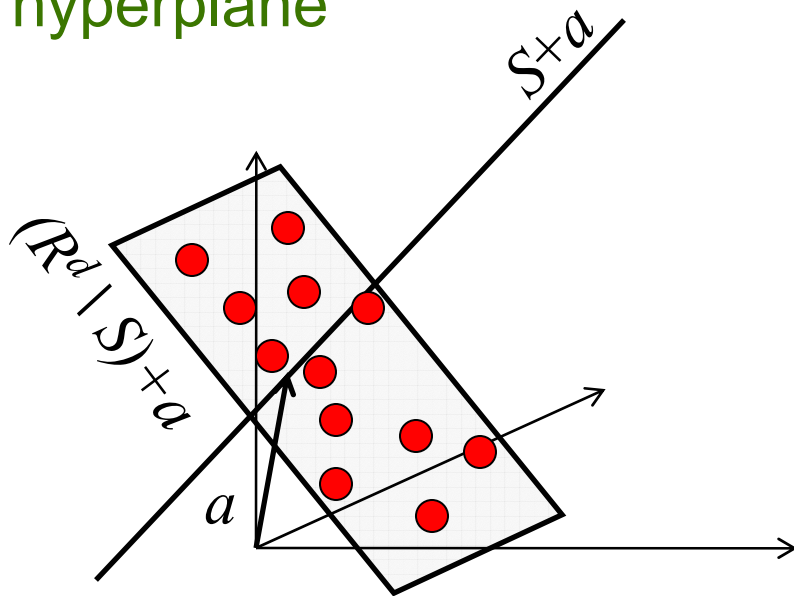


# Challenges and Approaches

- Affine subspace  $S+a$ ,  $S \subset R^d$ , affinity  $a \in R^d$  is interesting if a set of points clusters within this subspace
- Points may exhibit high variance in perpendicular subspace  $(R^d \setminus S)+a$

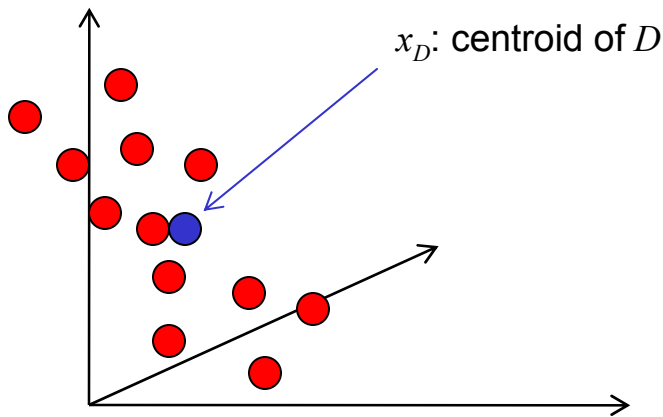


- high variance in perpendicular subspace  $(R^d \setminus S) + a \rightarrow$  points form a hyperplane within  $R^d$  located in this subspace  $(R^d \setminus S) + a$
- Points on a hyperplane appear to follow linear dependencies among the attributes participating in the description of the hyperplane



- Directions of high/low variance: PCA (local application)
- locality assumption: local selection of points sufficiently reflects the hyperplane accommodating the points
- general approach: build covariance matrix  $\Sigma_D$  for a selection  $D$  of points (e.g.  $k$  nearest neighbors of a point)

$$\Sigma_D = \frac{1}{|D|} \sum_{x \in D} (x - x_D)(x - x_D)^T$$



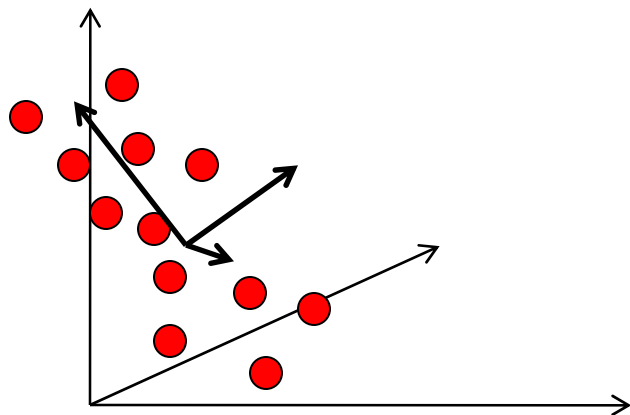
properties of  $\Sigma_D$ :

- $d \times d$
- symmetric
- positive semidefinite
- $\sigma_{D_{ij}}$  (value at row  $i$ , column  $j$ ) = covariance between dimensions  $i$  and  $j$
- $\sigma_{D_{ii}}$  = variance in  $i$ th dimension

- decomposition of  $\Sigma_D$  to eigenvalue matrix  $E_D$  and eigenvector matrix  $V_D$ :

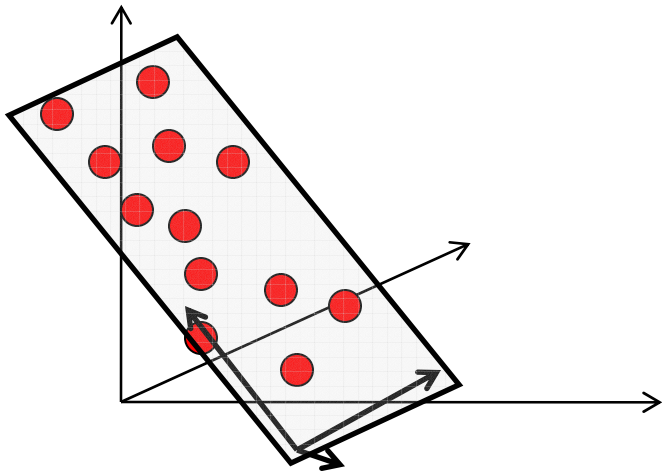
$$\Sigma_D = V_D E_D V_D^T$$

- $E_D$ : diagonal matrix, holding eigenvalues of  $\Sigma_D$  in decreasing order in its diagonal elements
- $V_D$ : orthonormal matrix with eigenvectors of  $\Sigma_D$  ordered correspondingly to the eigenvalues in  $E_D$



- $V_D$ : new basis, first eigenvector = direction of highest variance
- $E_D$ : covariance matrix of  $D$  when represented in new axis system  $V_D$

- points forming  $\lambda$ -dimensional hyperplane  $\rightarrow$  hyperplane is spanned by the first  $\lambda$  eigenvectors (called “strong” eigenvectors – notation:  $\check{V}_D$ )
- subspace where the points cluster densely is spanned by the remaining  $d-\lambda$  eigenvectors (called “weak” eigenvectors – notation:  $\hat{V}_D$ )



for the eigensystem, the sum of the smallest  $d-\lambda$  eigenvalues  $\sum_{i=\lambda+1}^d e_{D_i}$  is minimal under all possible transformations  $\rightarrow$  points cluster optimally dense in this subspace

- model for correlation clusters [ABK+06]:
  - $\lambda$ -dimensional hyperplane accommodating the points of a correlation cluster  $C \subset R^d$  is defined by an equation system of  $d-\lambda$  equations for  $d$  variables and the affinity (e.g. the mean point  $x_C$  of all cluster members):

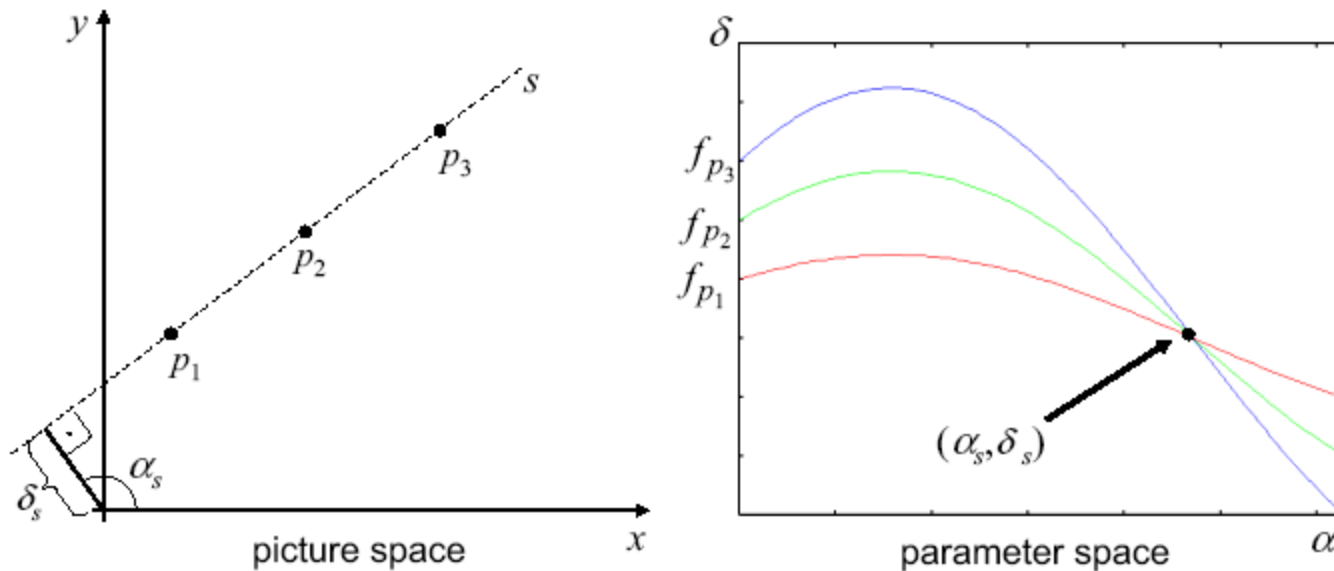
$$\hat{V}_C^T x = \hat{V}_C^T x_C$$

- equation system approximately fulfilled for all points  $x \in C$
- quantitative model for the cluster allowing for probabilistic prediction (classification)
- Note: correlations are observable, linear dependencies are merely an assumption to explain the observations – predictive model allows for evaluation of assumptions and experimental refinements

- PCA-based algorithms:
  - ORCLUS [AY00]:
    - first approach to *generalized projected clustering*
    - similar ideas to PROCLUS [APW+99] (k-means-like)
  - 4C [BKKZ04]
    - density-based cluster-paradigm (cf. DBSCAN [EK SX96])
    - enhancements:
      - COPAC [ABK+07c]: more efficient and robust
      - ERiC [ABK+07b]: finds hierarchies of correlation clusters



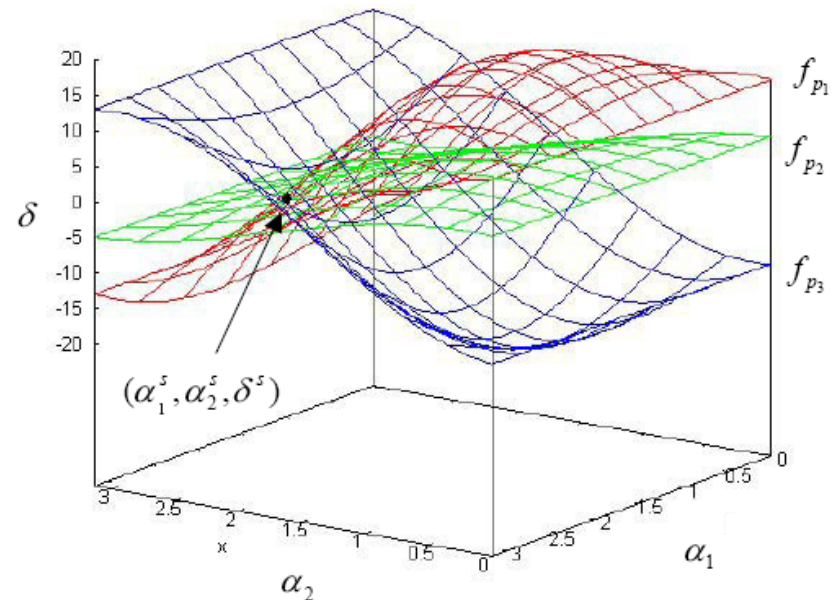
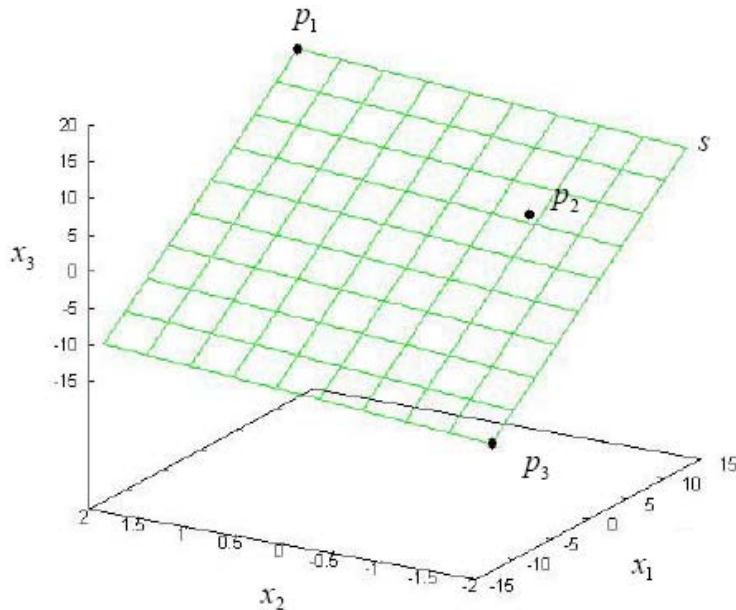
- different correlation primitive: Hough-transform
  - points in data space are mapped to functions in the parameter space



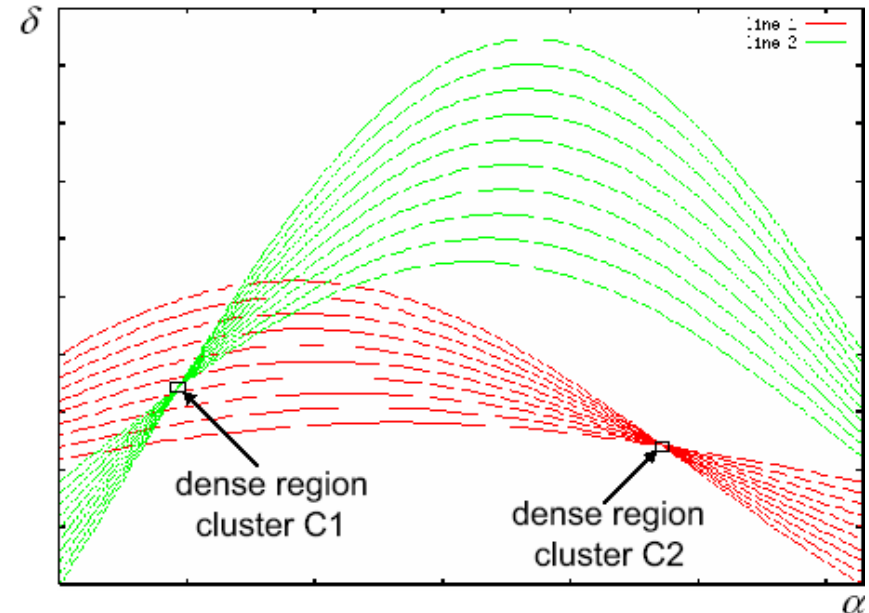
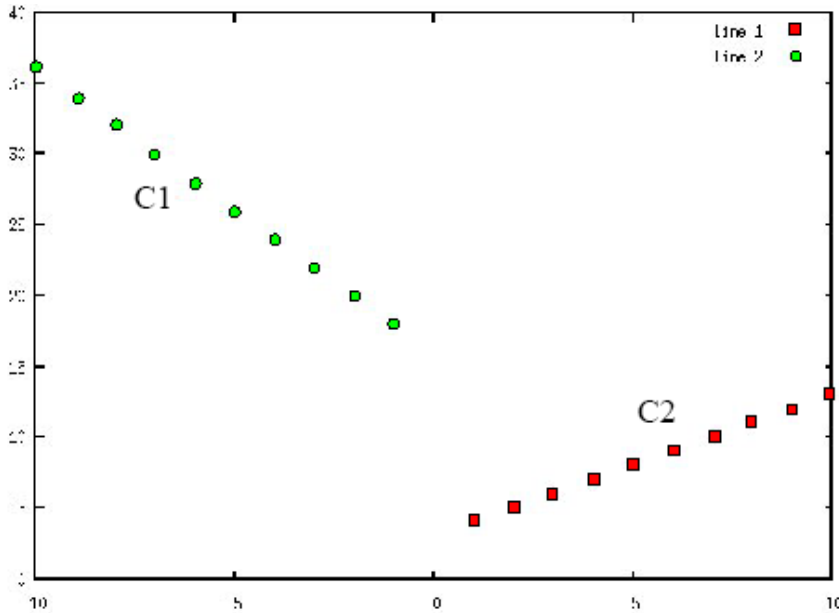
- functions in the parameter space define all lines possibly crossing the point in the data space

## – Properties of the transformation

- Point in the data space = sinusoidal curve in parameter space
- Point in parameter space = hyper-plane in data space
- Points on a common hyper-plane in data space = sinusoidal curves intersecting in a common point in parameter space
- Intersections of sinusoidal curves in parameter space = hyper-plane accommodating the corresponding points in data space



- Algorithm based on the Hough-transform: CASH [ABD+08]



- dense regions in parameter space (right) correspond to linear structures in data space (left)

- Idea: find dense regions in parameter space
  - construct a grid by recursively splitting the parameter space (best-first-search)
  - identify dense grid cells as intersected by many parametrization functions
  - dense grid cell represents  $(d-1)$ -dimensional linear structure
  - transform corresponding data objects in corresponding  $(d-1)$ -dimensional space and repeat the search recursively
- properties:
  - finds arbitrary number of clusters
  - requires specification of depth of search (number of splits per axis)
  - requires minimum density threshold for a grid cell
  - Note: this minimum density does not relate to the locality assumption: CASH is a global approach to correlation clustering
  - search heuristic: linear in number of points, but  $\sim d^4$
  - But: complete enumeration in worst case (exponential in  $d$ )

- PCA: mature technique, allows construction of a broad range of similarity measures for local correlation of attributes
- drawback: all approaches suffer from locality assumption
- successfully employing PCA in correlation clustering in “really” high-dimensional data requires more effort henceforth
  - for example: auto-tuning of local neighborhood, robustification of PCA [KKSZ08]
- new approach based on Hough-transform:
  - does not rely on locality assumption
  - but worst case again complete enumeration

- some preliminary approaches base on concept of self-similarity (intrinsic dimensionality, fractal dimension):  
[BC00,PTTF02,GHPT05]
  - interesting idea, provides quite a different basis to grasp correlations in addition to PCA
  - drawback: self-similarity assumes locality of patterns even by definition

- comparison: correlation clustering – biclustering:
  - model for correlation clusters more general and meaningful
  - models for biclusters rather specialized
  - in general, biclustering approaches do not rely on locality assumption
  - non-local approach and specialization of models may make biclustering successful in many applications
  - usually, biclustering algorithms follow a greedy approach
  - correlation clustering is the more general approach but the approaches proposed so far are rather a first draft to tackle the complex problem

1. Introduction
2. Axis-parallel Subspace Clustering
3. Pattern-based Clustering
4. Arbitrarily-oriented Subspace Clustering
5. Summary



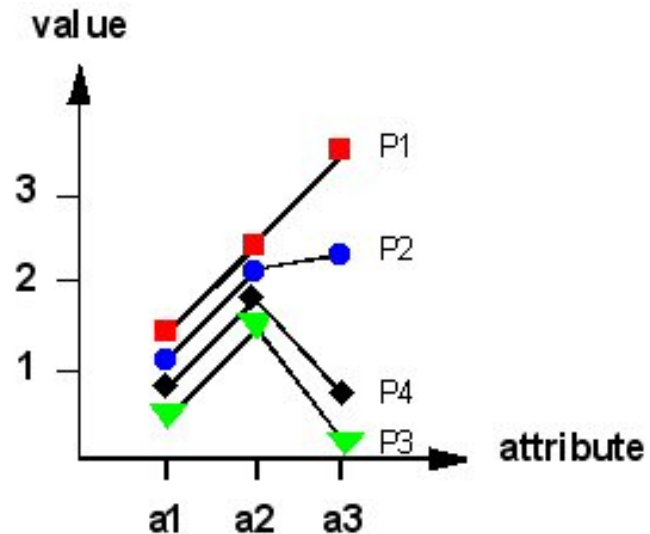
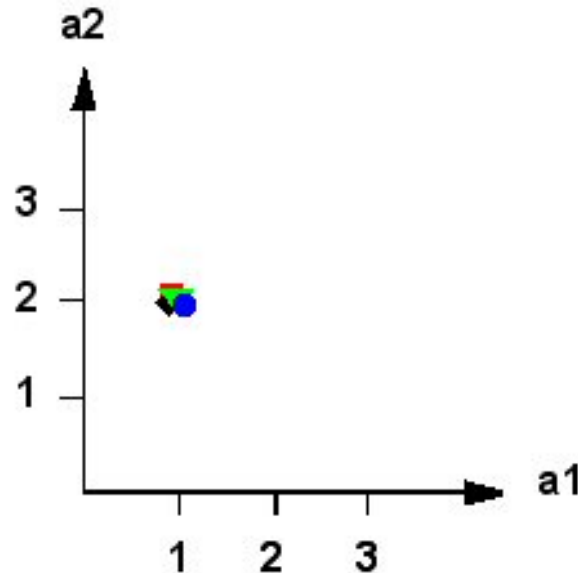
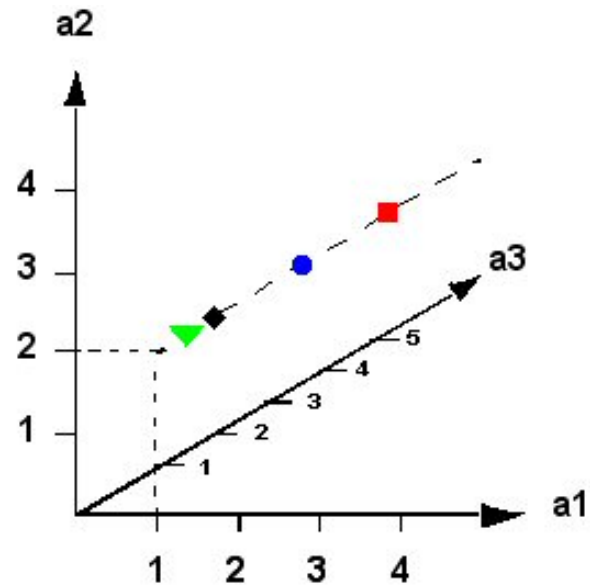
- Let's take the global view again
  - Traditional clustering in high dimensional spaces is most likely meaningless with increasing dimensionality (curse of dimensionality)
  - Clusters may be found in (generally arbitrarily oriented) subspaces of the data space
  - So the general problem of clustering high dimensional data is:  
“find a partitioning of the data where each cluster may exist in its own subspace”
    - The partitioning need not be unique (clusters may overlap)
    - The subspaces may be axis-parallel or arbitrarily oriented
  - Analysis of this general problem:
    - Sub-problem 1: search for clusters
    - Sub-problem 2: search for subspaces

- Analysis of the 2nd sub-problem (subspace search)
  - A naïve solution would examine all possible subspaces to look for clusters
  - The search space of all possible arbitrarily oriented subspaces is infinite
  - We need assumptions and heuristics to develop a feasible solution
- What assumptions did we get to know here to solve the subspace search problem?
  - The search space is restricted to certain subspaces
  - A clustering criterion that implements the downward closure property enables efficient search heuristics
  - The locality assumption enables efficient search heuristics
  - Assuming simple additive models (“patterns”) enables efficient search heuristics
  - ...

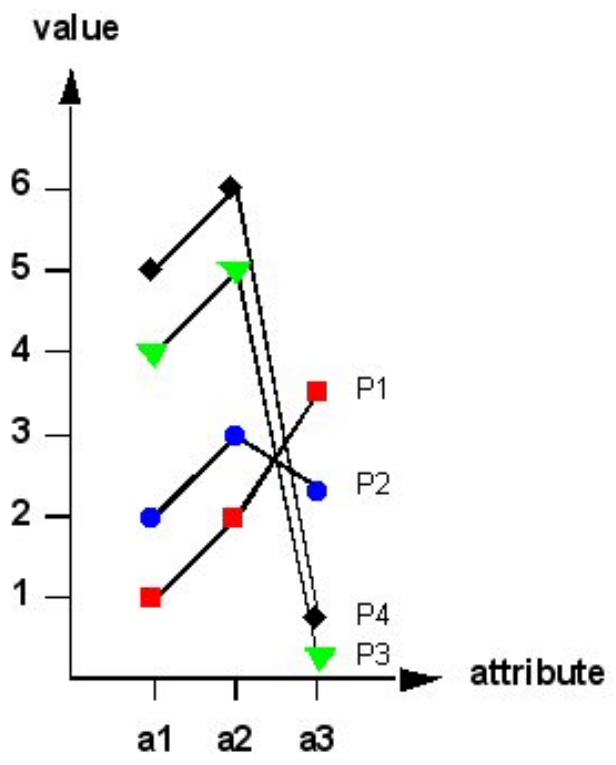
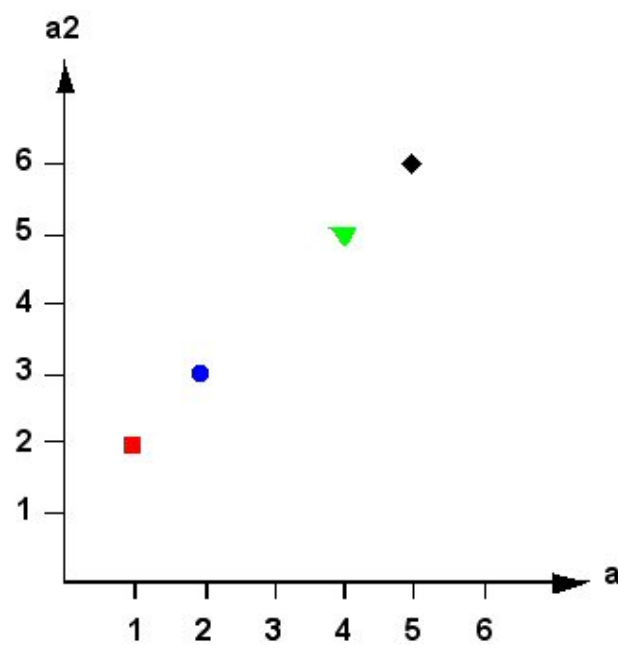
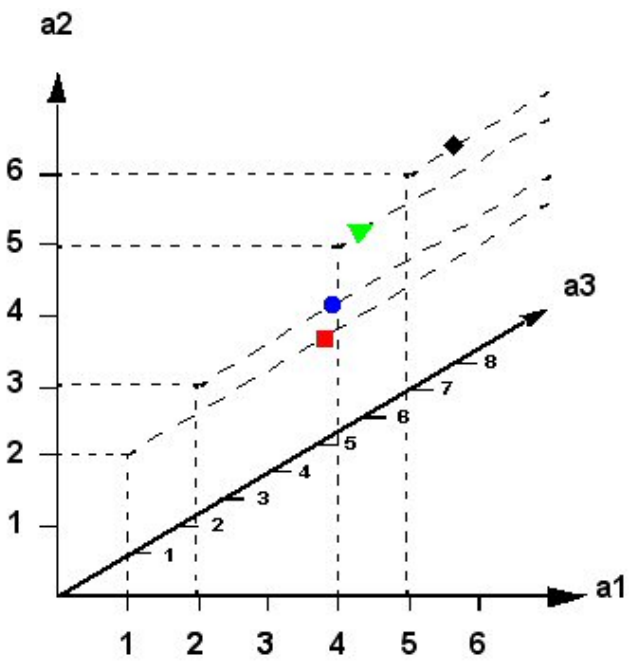
- Remember: also for the clustering problem (1st sub-problem) we need assumptions and heuristics that have an impact on the algorithms' properties
  - Number of clusters need to be specified
  - Results are not deterministic e.g. due to randomized procedures
  - ...
  
- Here, we classify the existing approaches according to the assumptions they made to conquer the infinite subspace search space (sub-problem 2)

## – The global view

- Subspace clustering/projected clustering:
  - Search space restricted to axis-parallel subspaces
  - Clustering criterion implementing the downward closure property (usually based on a global density threshold)
  - Locality assumption
  - ...



- The global view
  - Bi-clustering/pattern-based clustering:
    - Search space restricted to special forms and locations of subspaces or half-spaces
    - Greedy-search heuristics based on statistical assumptions

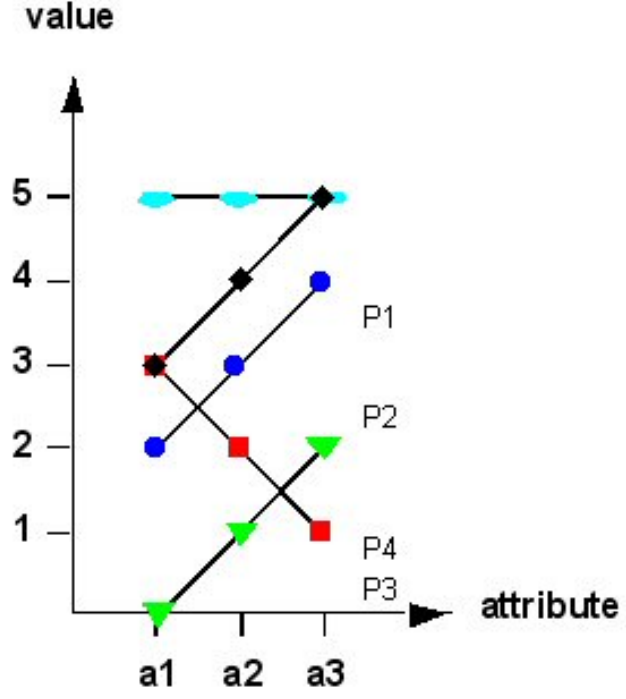
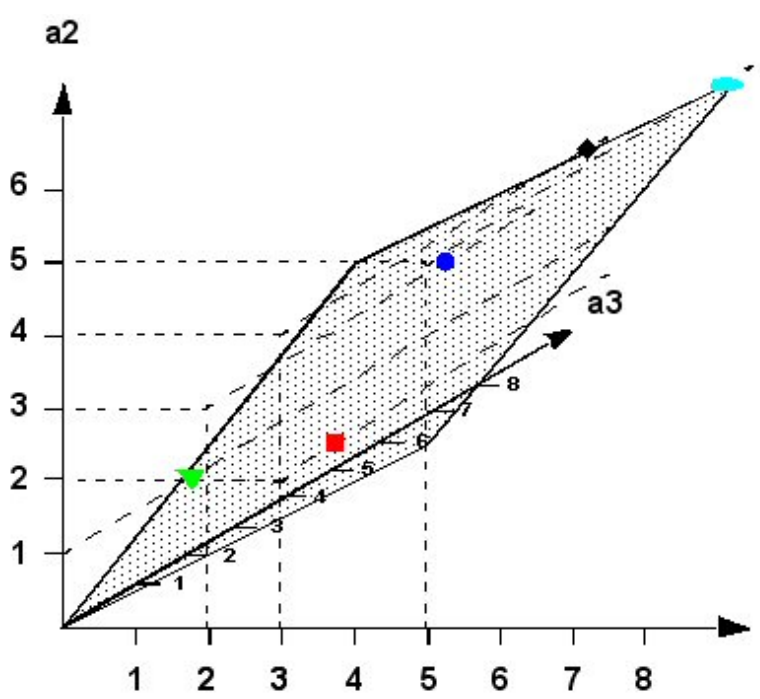


# Summary

- The global view
  - Correlation clustering:
    - Locality assumption
    - Greedy-search heuristics

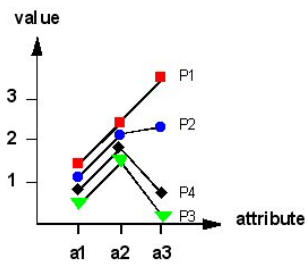
$$a1 - 2 \cdot a2 + a3 = 0$$

	a1	a2	a3
P1	3	2	1
P2	2	3	4
P3	0	1	2
P4	3	4	5
P5	5	5	6

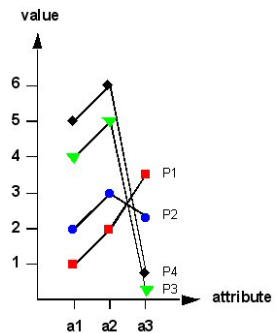


## – The global view

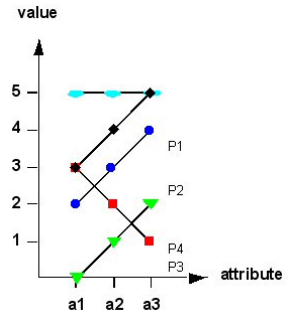
### Matrix-Pattern



Constant values  
in columns,  
change of values  
only on rows



From constant  
values in rows  
and columns (no  
change of values)  
to arbitrary  
change of values  
in common  
direction



No particular  
pattern

### Problem

Subspace / Projected  
Clustering

Pattern-based / Bi-  
Clustering

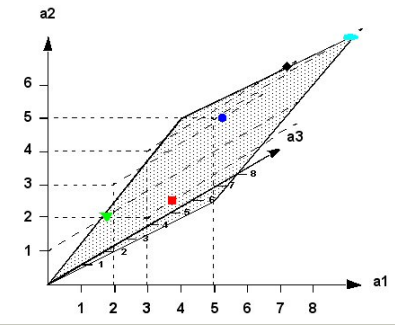
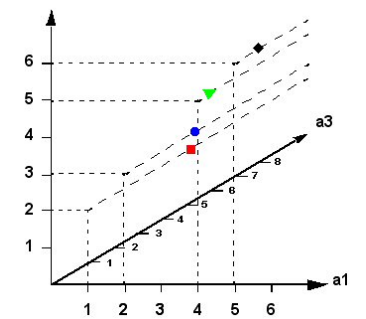
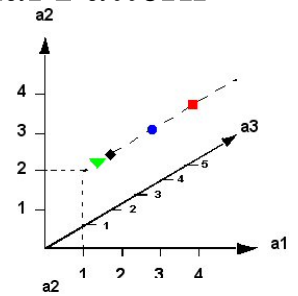
Correlation  
Clustering

### Spatial Pattern

Axis-parallel  
hyperplanes

Special cases  
of axis-parallel  
to special  
cases of  
arbitrarily  
oriented  
hyperplanes

Arbitrarily  
oriented  
hyperplanes



# Summary

Algorithm	complex correlations	simple positive correlation	simple negative correlation	axis parallel	not relying on locality assumption	adaptive density threshold	independent w.r.t. order of attributes	independent w.r.t. order of objects	deterministic	arbitrary number of clusters	overlapping clusters	overlapping subspaces	simultaneously overlapping clusters and subspaces	arbitrary subspace dimensionality	hierarchical structure	avoiding complete enumeration	noise robust
CLIQUE [AGGR98]				✓	✓		✓	✓	✓	✓	✓	✓	✓	✓			✓
ENCLUS [CFZ99]				✓	✓		✓	✓	✓	✓	✓	✓	✓	✓			✓
MAFIA [NGC01]				✓	✓		✓	✓	✓	✓	✓	✓	✓	✓			✓
SUBCLU [KKK04]				✓	✓		✓	✓	✓	✓	✓	✓	✓	✓			✓
PROCLUS [APW <sup>+</sup> 99]				✓		✓						✓				✓	
PreDeCon [BKKK04]				✓			✓	✓	✓	✓		✓				✓	✓
P3C [MSE06]				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
COSA [FM04]				✓			✓	✓	✓			✓		✓		✓	✓
DOC [PJAM02]				✓	✓		✓	✓		✓	✓	✓	✓	✓			
DiSH [ABK <sup>+</sup> 07a]				✓	✓	✓		✓	✓	✓		✓		✓	✓	✓	✓
FIRES [KKRW05]				✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓





# Summary



Algorithm	complex correlations	simple positive correlation	simple negative correlation	axis parallel	not relying on locality assumption	adaptive density threshold	independent w.r.t. order of attributes	independent w.r.t. order of objects	deterministic	arbitrary number of clusters	overlapping clusters	overlapping subspaces	simultaneously overlapping clusters and subspaces	arbitrary subspace dimensionality	hierarchical structure	avoiding complete enumeration	noise robust
Block clustering [Har72]					✓	<i>na</i>	✓	✓	✓					✓	✓		✓
$\delta$ -bicluster [CC00]		✓	✓	✓	✓	<i>na</i>	✓	✓	✓		✓	✓		✓		✓	✓
FLOC [YWWY02]		✓		✓	✓	<i>na</i>					✓	✓	✓	✓		✓	✓
p-Cluster [WWYY02]		✓		✓	✓	<i>na</i>	✓	✓	✓	✓	✓	✓	✓	✓			✓
MaPle [PZC <sup>+</sup> 03]		✓		✓	✓	<i>na</i>	✓	✓	✓	✓	✓	✓	✓	✓			✓
CoClus [CDGS04]		✓		✓	✓	<i>na</i>								✓		✓	
OP-Cluster [LW03]					✓	<i>na</i>	✓	✓	✓	✓	✓	<i>na</i>	<i>na</i>	<i>na</i>			✓



# Summary



Algorithm	complex correlations	simple positive correlation	simple negative correlation	axis parallel	not relying on locality assumption	adaptive density threshold	independent w.r.t. order of attributes	independent w.r.t. order of objects	deterministic	arbitrary number of clusters	overlapping clusters	overlapping subspaces	simultaneously overlapping clusters and subspaces	arbitrary subspace dimensionality	hierarchical structure	avoiding complete enumeration	noise robust
ORCLUS [AY00]	✓	✓	✓	✓			✓					✓				✓	
4C [BKKZ04]	✓	✓	✓	✓			✓	✓	✓	✓		✓				✓	✓
COPAC [ABK <sup>+</sup> 07c]	✓	✓	✓	✓			✓	✓	✓	✓		✓		✓		✓	✓
ERiC [ABK <sup>+</sup> 07b]	✓	✓	✓	✓			✓	✓	✓	✓		✓		✓	✓	✓	✓
CASH [ABD <sup>+</sup> 08]	✓	✓	✓	✓	✓	<i>na</i>		✓	✓	✓		✓		✓	✓		✓

- So, any of the proposed methods is based on at least one assumption because otherwise, it would not be applicable
- But how can we evaluate which assumption is better under which conditions?
  - General issues
    - Basically there is no comprehensive comparison on the accuracy or efficiency of the discussed methods
    - A fair comparison on the efficiency is only possible in sight of the assumptions and heuristics used by the single methods
    - An algorithm performs bad if it has more restrictions AND needs more time
    - Being less efficient but more general should be acceptable

- What we find in the papers
  - Head-to-head comparison with at most one or two competitors that do have similar assumptions
  - But that can be really misleading!!!
  - Sometimes there is even no comparison at all to other approaches
  - Sometimes the experimental evaluations are rather poor
  - At least, we are working on that ...  
see [AKZ08] or visit <http://www.dbs.ifi.lmu.de/research/KDD/ELKI/>
  
- So how can we decide which algorithm to use for a given problem?
  - Actually, we cannot ☹
  - However, we can sketch what makes a sound evaluation

- How should a sound experimental evaluation of the accuracy look like – an example using gene expression data

[Thanks to the anonymous reviewers for their suggestions even though we would have preferred an ACCEPT ;-)]

- Good:

- Apply your method to cluster the genes of a publicly available gene expression data set => you should get clusters of genes with similar functions
- Do not only report that your method has found some clusters (because even e.g. the full-dimensional  $k$ -means would have done so)
- Analyze your clusters: do the genes have similar functions?
  - Sure, we are computer scientists, not biologists, but ...
  - In publicly available databases you can find annotations for (even most of) the genes
  - These annotations can be used as class labels, so consistency measures can be computed

- Even better
  - Identify competing methods (that have similar assumptions like your approach)
  - Run the same experiments (see above) with the competing approaches
  - Your method is very valuable if
    - your clusters have a higher consistency score  
[OK, you are the winner]

OR

- your clusters have a lower (but still reasonably high) score and represent functional groups of genes that clearly differ from that found by the competitors
  - [you can obviously find other biologically relevant facts that could not be found by your competitors]
- Open question: what is a suitable consistency score for subspace clusters?

## – Premium

- You have a domain expert as partner who can analyze your clustering results in order to
  - Prove and/or refine his/her existing hypothesis
  - Derive new hypotheses

Lucky you – that's why we should make data mining 😊



## List of References



- [ABD+08] E. Aichert, C. Böhm, J. David, P. Kröger, and A. Zimek.  
**Robust clustering in arbitrarily oriented subspaces.**  
In Proceedings of the 8th SIAM International Conference on Data Mining (SDM),  
Atlanta, GA, 2008
- [ABK+06] E. Aichert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek.  
**Deriving quantitative models for correlation clusters.**  
In Proceedings of the 12th ACM International Conference on Knowledge Discovery  
and Data Mining (SIGKDD), Philadelphia, PA, 2006.
- [ABK+07a] E. Aichert, C. Böhm, H.-P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek.  
**Detection and visualization of subspace cluster hierarchies.**  
In Proceedings of the 12th International Conference on Database Systems for  
Advanced Applications (DASFAA), Bangkok, Thailand, 2007.
- [ABK+07b] E. Aichert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek.  
**On exploring complex relationships of correlation clusters.**  
In Proceedings of the 19th International Conference on Scientific and Statistical  
Database Management (SSDBM), Banff, Canada, 2007.
- [ABK+07c] E. Aichert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek.  
**Robust, complete, and efficient correlation clustering.**  
In Proceedings of the 7th SIAM International Conference on Data Mining (SDM),  
Minneapolis, MN, 2007.

- [AGGR98] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. **Automatic subspace clustering of high dimensional data for data mining applications.**  
In Proceedings of the ACM International Conference on Management of Data (SIGMOD), Seattle, WA, 1998.
- [AHK01] C. C. Aggarwal, A. Hinneburg, and D. Keim.  
**On the surprising behavior of distance metrics in high dimensional space.**  
In Proceedings of the 8th International Conference on Database Theory (ICDT), London, U.K., 2001.
- [AKZ08] E. Aichert, H.-P. Kriegel, A. Zimek.  
**ELKI: A Software System for Evaluation of Subspace Clustering Algorithms.**  
In Proceedings of the 20th International Conference on Scientific and Statistical Database Management (SSDBM), Hong Kong, China, 2008.
- [APW+99] C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park.  
**Fast algorithms for projected clustering.**  
In Proceedings of the ACM International Conference on Management of Data (SIGMOD), Philadelphia, PA, 1999.
- [AS94] R. Agrawal and R. Srikant. **Fast algorithms for mining association rules.**  
In Proceedings of the ACM International Conference on Management of Data (SIGMOD), Minneapolis, MN, 1994.

- [AY00] C. C. Aggarwal and P. S. Yu.  
**Finding generalized projected clusters in high dimensional space.**  
In Proceedings of the ACM International Conference on Management of Data (SIGMOD), Dallas, TX, 2000.
- [BBC04] N. Bansal, A. Blum, and S. Chawla. **Correlation clustering.**  
Machine Learning, 56:89–113, 2004.
- [BC00] D. Barbara and P. Chen.  
**Using the fractal dimension to cluster datasets.**  
In Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Boston, MA, 2000.
- [BDCKY02] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini.  
**Discovering local structure in gene expression data: The order-preserving submatrix problem.**  
In Proceedings of the 6th Annual International Conference on Computational Molecular Biology (RECOMB), Washington, D.C., 2002.
- [BGRS99] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft.  
**When is “nearest neighbor” meaningful?**  
In Proceedings of the 7th International Conference on Database Theory (ICDT), Jerusalem, Israel, 1999.

- [BKkk04] C. Böhm, K. Kailing, H.-P. Kriegel, and P. Kröger.  
**Density connected clustering with local subspace preferences.**  
In Proceedings of the 4th International Conference on Data Mining (ICDM),  
Brighton, U.K., 2004.
- [BKkZ04] C. Böhm, K. Kailing, P. Kröger, and A. Zimek.  
**Computing clusters of correlation connected objects.**  
In Proceedings of the ACM International Conference on Management of Data  
(SIGMOD), Paris, France, 2004.
- [CC00] Y. Cheng and G. M. Church. **Biclustering of expression data.**  
In Proceedings of the 8<sup>th</sup> International Conference Intelligent Systems for Molecular  
Biology (ISMB), San Diego, CA, 2000.
- [CDGS04] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra.  
**Minimum sum-squared residue co-clustering of gene expression data.**  
In Proceedings of the 4th SIAM International Conference on Data Mining (SDM),  
Orlando, FL, 2004.
- [CFZ99] C. H. Cheng, A. W.-C. Fu, and Y. Zhang.  
**Entropy-based subspace clustering for mining numerical data.**  
In Proceedings of the 5th ACM International Conference on Knowledge Discovery  
and Data Mining (SIGKDD), San Diego, CA, pages 84–93, 1999.

- [CST00] A. Califano, G. Stolovitzky, and Y. Tu.  
**Analysis of gene expression microarrays for phenotype classification.**  
In Proceedings of the 8th International Conference Intelligent Systems for Molecular Biology (ISMB), San Diego, CA, 2000.
- [EKSX96] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu.  
**A density-based algorithm for discovering clusters in large spatial databases with noise.**  
In Proceedings of the 2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR, 1996.
- [FM04] J. H. Friedman and J. J. Meulman.  
**Clustering objects on subsets of attributes.**  
Journal of the Royal Statistical Society: Series B (Statistical Methodology), 66(4):825–849, 2004.
- [GHPT05] A. Gionis, A. Hinneburg, S. Papadimitriou, and P. Tsaparas.  
**Dimension induced clustering.**  
In Proceedings of the 11th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Chicago, IL, 2005.

- [GLD00] G. Getz, E. Levine, and E. Domany.  
**Coupled two-way clustering analysis of gene microarray data.**  
Proceedings of the National Academy of Sciences of the United States of America,  
97(22):12079–12084, 2000.
- [GRRK05] E. Georgii, L. Richter, U. Rückert, and S. Kramer.  
**Analyzing microarray data using quantitative association rules.**  
Bioinformatics, 21(Suppl. 2):ii1–ii8, 2005.
- [GW99] B. Ganter and R. Wille.  
**Formal Concept Analysis.**  
Mathematical Foundations. Springer, 1999.
- [HAK00] A. Hinneburg, C. C. Aggarwal, and D. A. Keim.  
**What is the nearest neighbor in high dimensional spaces?**  
In Proceedings of the 26th International Conference on Very Large Data Bases  
(VLDB), Cairo, Egypt, 2000.
- [Har72] J. A. Hartigan.  
**Direct clustering of a data matrix.**  
Journal of the American Statistical Association, 67(337):123–129, 1972.

- [IBB04] J. Ihmels, S. Bergmann, and N. Barkai.  
**Defining transcription modules using large-scale gene expression data.**  
Bioinformatics, 20(13):1993–2003, 2004.
- [Jol02] I. T. Jolliffe. **Principal Component Analysis.**  
Springer, 2nd edition, 2002.
- [KKK04] K. Kailing, H.-P. Kriegel, and P. Kröger.  
**Density-connected subspace clustering for highdimensional data.**  
In Proceedings of the 4th SIAM International Conference on Data Mining (SDM),  
Orlando, FL, 2004.
- [KKRW05] H.-P. Kriegel, P. Kröger, M. Renz, and S. Wurst.  
**A generic framework for efficient subspace clustering of high-dimensional data.**  
In Proceedings of the 5th International Conference on Data Mining (ICDM),  
Houston, TX, 2005.
- [KKSZ08] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek.  
**A general framework for increasing the robustness of PCA-based correlation clustering algorithms.**  
In Proceedings of the 20th International Conference on Scientific and Statistical Database Management (SSDBM), Hong Kong, China, 2008.

- [LW03] J. Liu and W. Wang.  
**OP-Cluster: Clustering by tendency in high dimensional spaces.**  
In Proceedings of the 3th International Conference on Data Mining (ICDM),  
Melbourne, FL, 2003.
- [MK03] T. M. Murali and S. Kasif.  
**Extracting conserved gene expression motifs from gene expression data.**  
In Proceedings of the 8th Pacific Symposium on Biocomputing (PSB), Maui, HI,  
2003.
- [MO04] S. C. Madeira and A. L. Oliveira.  
**Biclustering algorithms for biological data analysis: A survey.**  
IEEE Transactions on Computational Biology and Bioinformatics, 1(1):24–45, 2004.
- [MSE06] G. Moise, J. Sander, and M. Ester.  
**P3C: A robust projected clustering algorithm.**  
In Proceedings of the 6th International Conference on Data Mining (ICDM),  
Hong Kong, China, 2006.
- [NGC01] H.S. Nagesh, S. Goil, and A. Choudhary.  
**Adaptive grids for clustering massive data sets.**  
In Proceedings of the 1st SIAM International Conference on Data Mining (SDM),  
Chicago, IL, 2001.



- [PBZ+06] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Guissem, L. Hennig, L. Thiele, and E. Zitzler.  
**A systematic comparison and evaluation of biclustering methods for gene expression data.**  
Bioinformatics, 22(9):1122–1129, 2006.
- [Pfa07] J. Pfaltz. **What constitutes a scientific database?**  
In Proceedings of the 19th International Conference on Scientific and Statistical Database Management (SSDBM), Banff, Canada, 2007.
- [PHL04] L. Parsons, E. Haque, and H. Liu.  
**Subspace clustering for high dimensional data: A review.**  
SIGKDD Explorations, 6(1):90–105, 2004.
- [PJAM02] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali.  
**A Monte Carlo algorithm for fast projective clustering.**  
In Proceedings of the ACM International Conference on Management of Data (SIGMOD), Madison, WI, 2002.
- [PTTF02] E. Parros Machado de Sousa, C. Traina, A. Traina, and C. Faloutsos.  
**How to use fractal dimension to find correlations between attributes.**  
In Proc. KDD-Workshop on Fractals and Self-similarity in Data Mining: Issues and Approaches, 2002.

- [PZC+03] J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu.  
**MaPle: A fast algorithm for maximal pattern-based clustering.**  
In Proceedings of the 3th International Conference on Data Mining (ICDM),  
Melbourne, FL, 2003.
- [RRK04] U. Rückert, L. Richter, and S. Kramer.  
**Quantitative association rules based on half-spaces: an optimization  
approach.**  
In Proceedings of the 4th International Conference on Data Mining (ICDM),  
Brighton, U.K., 2004.
- [SCH75] J.L. Slagle, C.L. Chang, S.L. Heller.  
**A Clustering and Data-Reorganization Algorithm.**  
IEEE Transactions on Systems, Man and Cybernetics, 5: 121-128, 1975
- [SLGL06] K. Sim, J. Li, V. Gopalkrishnan, and G. Liu.  
**Mining maximal quasi-bicliques to co-cluster stocks and financial ratios for  
value investment.**  
In Proceedings of the 6th International Conference on Data Mining (ICDM), Hong  
Kong, China, 2006.
- [SMD03] Q. Sheng, Y. Moreau, and B. De Moor.  
**Biclustering microarray data by Gibbs sampling.**  
Bioinformatics, 19(Suppl. 2):ii196–ii205, 2003.

- [STG+01] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller.  
**Rich probabilistic models for gene expression.**  
Bioinformatics, 17(Suppl. 1):S243–S252, 2001.
- [SZ05] K. Sequeira and M. J. Zaki.  
**SCHISM: a new approach to interesting subspace mining.**  
International Journal of Business Intelligence and Data Mining, 1(2):137–160, 2005.
- [TSS02] A. Tanay, R. Sharan, and R. Shamir.  
**Discovering statistically significant biclusters in gene expression data.**  
Bioinformatics, 18 (Suppl. 1):S136–S144, 2002.
- [TXO05] A. K. H. Tung, X. Xu, and C. B. Ooi.  
**CURLER: Finding and visualizing nonlinear correlated clusters.**  
In Proceedings of the ACM International Conference on Management of Data (SIGMOD), Baltimore, ML, 2005.
- [Web01] G. I. Webb.  
**Discovering associations with numeric variables.**  
In Proceedings of the 7<sup>th</sup> ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Francisco, CA, pages 383–388, 2001.

- [WLKL04] K.-G. Woo, J.-H. Lee, M.-H. Kim, and Y.-J. Lee.  
**FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting.**  
Information and Software Technology, 46(4):255–271, 2004.
- [WWYY02] H. Wang, W. Wang, J. Yang, and P. S. Yu.  
**Clustering by pattern similarity in large data sets.**  
In Proceedings of the ACM International Conference on Management of Data (SIGMOD), Madison, WI, 2002.
- [YWWY02] J. Yang, W. Wang, H. Wang, and P. S. Yu.  
 **$\delta$ -clusters: Capturing subspace correlation in a large data set.**  
In Proceedings of the 18th International Conference on Data Engineering (ICDE), San Jose, CA, 2002.