Discovering Multiple Clustering Solutions: Grouping Objects in Different Views of the Data

Emmanuel Müller[•], Stephan Günnemann[◦], Ines Färber[◦], Thomas Seidl[◦]

[•] Karlsruhe Institute of Technology, Germany
[◦] RWTH Aachen University, Germany

Tutorial at SDM 2011

download slides: `http://dme.rwth-aachen.de/DMCS`

Overview

Tradition Cluster Detection

Abstract cluster definition

"Group similar objects in one group,
separating dissimilar objects in different groups."

- Several instances focus on:
  different similarity functions, cluster characteristics, data types, ...
- Most definitions provide only a **single clustering solution**

For example, *K*-MEANS

- Aims at a **single partitioning** of the data
  Each **object** is assigned to exactly **one cluster**
- Aims at **one clustering** solution
  One set of K **clusters** forming the resulting **groups of objects**

$\Rightarrow$ In contrast, we focus on **multiple clustering solutions**...

What are Multiple Clusterings?

Informally, **Multiple Clustering Solutions** are...

- **Multiple sets of clusters** providing more insights than only one solution
- One given solution and a **different grouping** forming alternative solutions

Goals and objectives:

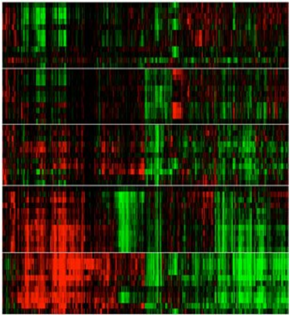
- Each object should be grouped in multiple clusters, representing different perspectives on the data.
- The result should consist of many alternative solutions. Users may choose one or use multiple of these solutions.
- Solutions should differ to a high extend, and thus, each of these solutions provides additional knowledge.

$\Rightarrow$ Overall, enhanced extraction of knowledge.

$\Rightarrow$ Objectives are motivated by various application scenarios...

Application: Gene Expression Analysis

Cluster detection in gene databases to derive multiple functional roles...



- Objects are genes described by their expression (behavior) under different conditions.
- Aim:
  Groups of genes with similar function.
- Challenge:
  One gene may have multiple functions

⇒ There is not a single grouping.

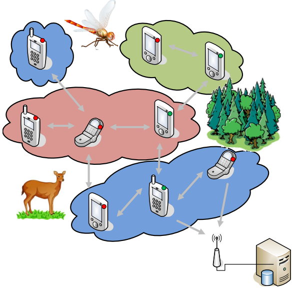
- Biologically motivated,
  clusters have to represent multiple functional roles for each object.

Each object may have several roles in multiple clusters (1)

⇒ Multiple Clustering Solutions required...

Application: Sensor Surveillance

Cluster detection in sensor networks to derive environmental conditions...



- Objects are sensor nodes described by their measurements.
- Aim:
  Groups of sensors in similar environments.
- Challenge:
  One cluster might represent high temperature, another cluster might represent low humidity

$\Rightarrow$ There is not a single perspective.

- Clusters have to represent the different sensor measurements, and thus, clusters represent the different views on the data.

Clusters are hidden in different views on the data (2)

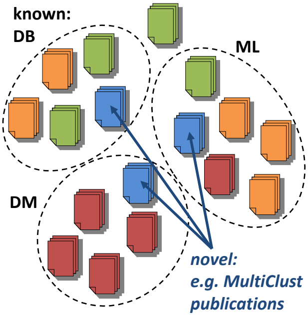
$\Rightarrow$ Multiple Clustering Solutions required...

Application: Text Analysis

Detecting novel topics based on given knowledge...

- Objects are text documents described by their content.
- Aim:
  Groups of documents on similar topic.
- Challenge:
  Some topics are well known (e.g. DB/DM/ML).
  In contrast, one is interested in detecting novel topics not yet known.

⇒ There are multiple alternative clustering solutions.



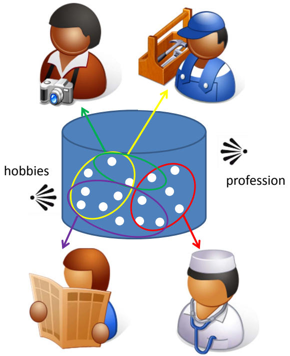
- Documents describe different topics: Some of them are well known, others form the desired alternatives to be detected

Multiple clusters describe alternative solutions (3)

⇒ Multiple Clustering Solutions required...

Application: Customer Segmentation

Clustering customer profiles to derive their interests...



- Objects are customers described by profiles.
- Aim:
  Groups of customers with similar behavior.
- Challenge:
  Customers show common musical interest but show different sport activities

$\Rightarrow$ Groups are described by subsets of attributes.

- Customers seem to be unique on all available attributes, but show multiple groupings considering subsets of the attributes.

Multiple clusterings hidden in projections of the data (4)

$\Rightarrow$ Multiple Clustering Solutions required...

General Application Demands

Several properties can be derived out of these applications,
they raise new research questions and give hints how to solve them:

Why should we aim at multiple clustering solutions?

(1) Each object may have **several roles in multiple clusters**
(2) Clusters are hidden in **different views** of the data

How should we guide our search to find these multiple clusterings?

(3) Model the **difference of clusters** and search for **alternative groups**
(4) Model the **difference of views** and search in **projections of the data**

$\Rightarrow$ In general, this occurs due to

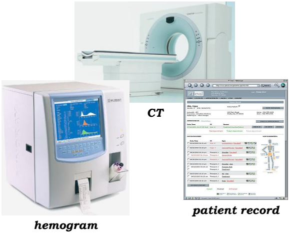
- data integration, merging multiple sources providing a complete picture ...
- evolutionary databases, providing more and more attributes per object...

... in high dimensional databases

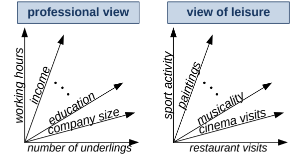
Integration of Multiple Sources

Usually it can be expected that there exist different views on the data:

- Information about the data is collected from different domains
  $\rightarrow$ different features are recorded
  - medical diagnosis (CT, hemogram,...)
  - multimedia (audio, video, text)
  - web pages (text of this page, anchor texts)
  - molecules (amino acid sequence, secondary structure, 3D representation)



- For **high dimensional data** different views/perspectives on the data may exist
- Multiple data sources provide us with **multiple given views on the data**



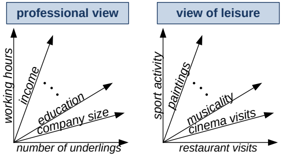
Lost Views due to Evolving Databases

Huge databases are gathered over time, adding more and more information into existing databses...

- Extending the stored information may lead to huge data dumps
- Relations between individual tables get lost
- Overall, different views are merged to one universal view on the data

⇒ Resulting in high dimensional data, as well.

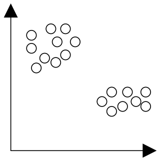
- Given some knowledge about one view on the data, one is interested in **alternative view** on the same data.



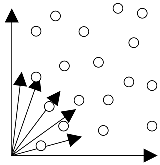
Challenge: High Dimensional Data

- Considering more and more attributes...
- **Objects become unique**, known as the

  "curse of dimensionality" (Beyer *et al.*, 1999)



$$\lim_{|D|\to\infty} \frac{\max_{p\in DB} dist_D(o,p) - \min_{p\in DB} dist_D(o,p)}{\min_{p\in DB} dist_D(o,p)} \to 0$$



- Object tend to be very dissimilar to each other...

⇒ How to cope with this effect in data mining?

⇒ identify relevant dimensions (views/subspaces/space transformations)

⇒ restrict distance computation to these views

⇒ enable detection of patterns in projection of high dimensional data

Challenge: Comparison of Clusterings

Requirements for Multiple Clustering Solutions:

- Identify only one solution is too restrictive

$\Rightarrow$ Multiple solutions are desired

- However, one searches for different / alternative / orthogonal clusterings

$\Rightarrow$ Novel definitions of difference between clusterings

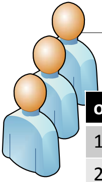
- Search for multiple sets of clusters (multiple clusterings), in contrast to one optimal set of clusters

$\Rightarrow$ Novel objective functions required

In contrast to (dis-)similarity between objects

- Define (dis-)similarity between clusters
- Define (dis-)similarity between views
- No common definitions for both of these properties!

Example Customer Analysis – Abstraction



object ID	age	income	blood pres.	sport activ.	profession
1	XYZ	XYZ	XYZ	XYZ	XYZ
2	XYZ	XYZ	XYZ	XYZ	XYZ
3	XYZ	XYZ	XYZ	XYZ	XYZ
4	XYZ	XYZ	XYZ	XYZ	XYZ
5	XYZ	XYZ	XYZ	XYZ	XYZ
6	XYZ	XYZ	XYZ	XYZ	XYZ
7	XYZ	XYZ	XYZ	XYZ	XYZ
8	XYZ	XYZ	XYZ	XYZ	XYZ
9	XYZ	XYZ	XYZ	XYZ	XYZ

- Consider each customer as a row in a database table
- Here a selection of possible attributes (example)

Example Customer Analysis – Clustering

object ID	age	income	blood pres.	sport activ.	profession
1					
2					
3	50	59.000	130	comp. game	CS
4	51	61.000	129	comp. game	CS
5	49	58.500	...	...	...
6	47	62.000	...	...	...
7	52	60.000	...	...	...
8					
9					

- Group similar objects in one “cluster”
- Separate dissimilar objects in different clusters
- Provide one clustering solution, for each object one cluster

Example Customer Analysis – Multiple Clusterings

<table>
<tr><th>object ID</th><th>age</th><th>income</th><th>blood pres.</th><th>sport activ.</th><th>profession</th></tr>
<tr><td>1</td><td rowspan="4" colspan="2">rich oldies</td><td colspan="3"></td></tr>
<tr><td>2</td><td rowspan="2" colspan="2">healthy sporties</td><td></td></tr>
<tr><td>3</td><td></td></tr>
<tr><td>4</td><td colspan="3">sport professionals</td></tr>
<tr><td>5</td><td></td><td></td><td rowspan="3" colspan="2">unhealthy gamers</td><td></td></tr>
<tr><td>6</td><td rowspan="2" colspan="2">average people</td><td rowspan="2"></td></tr>
<tr><td>7</td></tr>
<tr><td>8</td><td rowspan="2" colspan="2">unemployed people</td><td></td><td></td><td rowspan="2"></td></tr>
<tr><td>9</td><td colspan="2"></td></tr>
</table>

- Each object might be clustered by using multiple views
- For example, considering combinations of attributes

⇒ For each object multiple clusters are detected

⇒ Novel challenges in **cluster definition**, i.e. not only similarity of objects

Example Customer Analysis – Multiple Views

- Cluster of customers which show high similarity in **health behavior**
- Cluster of customers which show high similarity in **music interest**
- Cluster of customers which show high similarity in **sport activities**
- Cluster of customers which show high similarity in **...**

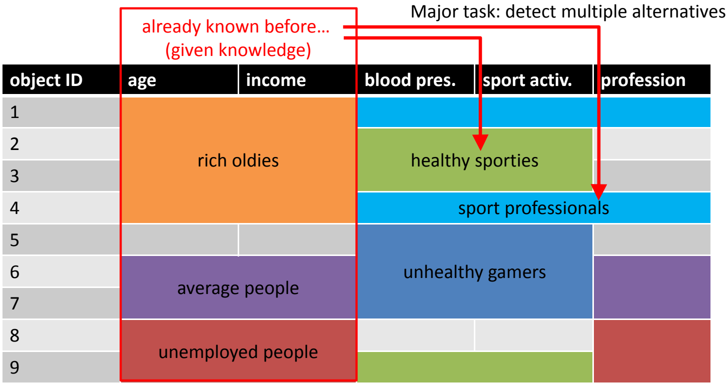
$\Rightarrow$ Group all objects according to these criteria.

Challenge:

- These criteria (views, perspectives, etc.) have to be detected
- Criteria depend on the possible cluster structures
- Criteria enforce different grouping although similarity of objects (without these criteria) shows only one optimal solution

$\Rightarrow$ Task: Enforce clustering to detect multiple solutions

Example Customer Analysis – Alternative Clusterings



- Assume a given knowledge about one clustering
- How to find the residual (alternative clustering solutions) that describe additional knowledge?

⇒ Novel challenges in **defining differences between clusterings**

Overview of Challenges and Techniques

One can observe general challenges:

- Clusters hidden in integrated data spaces from multiple sources
- Single data source with clusters hidden in multiple perspectives
- High dimensional data with clusters hidden in low dimensional projections

General techniques covered by this tutorial...

- Cluster definitions enforcing **multiple clustering solutions**
- Cluster definitions providing **alternatives to given knowledge**
- Cluster definitions **selecting relevant views** on the data

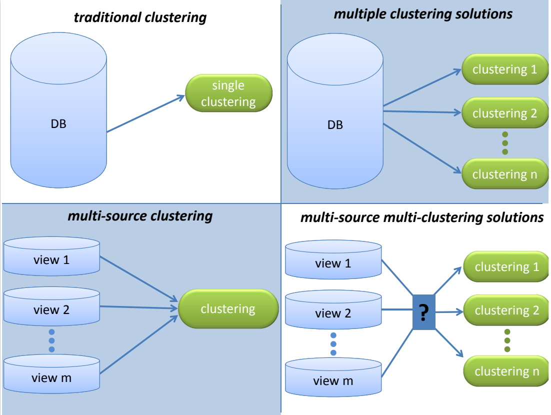
- First step for characterization and overview of existing approaches...

$\Rightarrow$ Taxonomy of paradigms and methods

Taxonomy of Approaches I

Basic taxonomy

- ONE database:
  ONE clustering
  (traditional clustering)
- ONE database:
  MULTIPLE clusterings
  (tutorial: major focus)
- MULTIPLE databases:
  ONE clustering
  (tutorial: given views)
- MULTIPLE databases:
  MULTIPLE clusterings
  (? still unclear ?)



Taxonomy of Approaches II

Taxonomy for **MULTIPLE CLUSTERING SOLUTIONS**

From the perspective of the underlying data space:

- Detection of multiple clustering solutions...
  - in the Original Data Space
  - by Orthogonal Space Transformations
  - by Different Subspace Projections
  - in Multiple Given Views/Sources

<table>
<tr><th></th><th></th><th>search space taxonomy</th><th>processing</th><th>knowledge</th><th>flexibility</th></tr>
<tr><td rowspan="4">Sec. 2</td><td>algorithm1</td><td rowspan="4">original space</td><td></td><td></td><td>exch. def.</td></tr>
<tr><td>alg2</td><td>iterative</td><td>given k.</td><td rowspan="3">specialized</td></tr>
<tr><td>alg3</td><td rowspan="2">simultan.</td><td rowspan="2">no given k.</td></tr>
<tr><td>alg4</td></tr>
<tr><td rowspan="2">Sec. 3</td><td>alg5</td><td rowspan="2">orthogonal transformations</td><td rowspan="2">iterative</td><td rowspan="2">given k.</td><td rowspan="2">exch. def.</td></tr>
<tr><td>alg6</td></tr>
<tr><td rowspan="4">Sec. 4</td><td>alg7</td><td rowspan="4">subspace projections</td><td></td><td rowspan="2">no given k.</td><td rowspan="3">specialized</td></tr>
<tr><td>alg8</td><td rowspan="2">simultan.</td></tr>
<tr><td>alg9</td><td>given k.</td></tr>
<tr><td>alg10</td><td></td><td rowspan="4">no given k.</td><td>exch. def.</td></tr>
<tr><td rowspan="3">Sec. 5</td><td>alg11</td><td rowspan="3">multiple views/sources</td><td rowspan="2">simultan.</td><td rowspan="2">specialized</td></tr>
<tr><td>alg12</td></tr>
<tr><td>alg13</td><td></td><td>exch. def.</td></tr>
</table>

Taxonomy of Approaches III

Further characteristics

From the perspective of the given knowledge:

- No clustering is given
- One or multiple clusterings are given

From the perspective of cluster computation:

- Iterative computation of further clustering solutions
- Simultaneous computation of multiple clustering solutions

From the perspective of parametrization/flexibility:

- Detection of a fixed number of clustering solutions
- The number of clusterings to be detected is not specified by the user
- The underlying cluster definition can be exchanged (flexible model)

Common Notions vs. Diversity of Terms I

CLUSTER vs. CLUSTERING

CLUSTER = a set of similar objects
CLUSTERING = a set of clusters

alternative clusters

disparate clusters

subspace search

different clusters

MULTIPLE CLUSTERING SOLUTIONS

multi-source clustering

multi-view clustering

subspace clustering

orthogonal clustering

Common Notions vs. Diversity of Terms II

ALTERNATIVE CLUSTERING

with a given knowledge used to find alternative clusterings

ORTHOGONAL CLUSTERING

transforming the search space based on previous results

SUBSPACE CLUSTERING

using different subspace projections to find clusters in lower dimensional projections

SIMILARITY and DISSIMILARITY are used in several contexts:

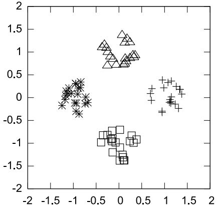
- OBJECTS: to define similarity of objects in one cluster
- CLUSTERS: to define the dissimilarity of clusters in multiple clusterings
- SPACES: to define the dissimilarity of transformed or projected spaces

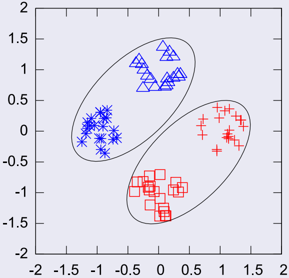
Overview

Motivation: Multiple Clusterings in a Single Space

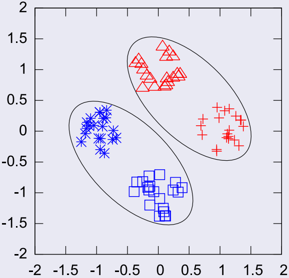
A frequently used toy example

- Note: In real world scenarios the clustering structure is more difficult to reveal
- Let's assume we want to partition the data in two clusters





multiple meaningful solutions possible



Abstract Problem Definition

General notions

- $DB \subseteq Domain$ — set of objects (usually $Domain = \mathbb{R}^d$)
- $Clust_i$ — clustering (set of clusters C_j) of the objects DB
- $Clusterings$ — theoretical set of all clusterings
- $Q : Clusterings \rightarrow \mathbb{R}$ — function to measure the quality of a clustering
- $Diss : Clusterings \times Clusterings \rightarrow \mathbb{R}$ — function to measure the dissimilarity between clusterings

Aim: Detect clusterings $Clust_1, \ldots, Clust_m$ such that

- $Q(Clust_i)$ is high $\forall i \in \{1, \ldots, m\}$
- $Diss(Clust_i, Clust_j)$ is high $\forall i, j \in \{1, \ldots, m\}, i \neq j$

Comparison to Traditional Clustering

Multiple Clusterings

Detect clusterings $Clust_1, \ldots, Clust_m$ such that

- $Q(Clust_i)$ is high $\forall i \in \{1, \ldots, m\}$
- $Diss(Clust_i, Clust_j)$ is high $\forall i, j \in \{1, \ldots, m\}, i \neq j$

Traditional clustering

- traditional clustering is special case
- just one clustering, i.e. $m = 1$
- dissimilarity trivially fulfilled
- consider e.g. k-Means:
  - quality function $Q \rightarrow$ compactness/total distance

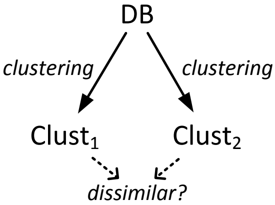
First approach: Meta Clustering

Meta clustering (Caruana *et al.*, 2006)

1. generate many clustering solutions
   - use of non-determinism or local minima/maxima
   - use of different clustering algorithms
   - use of different parameter settings
2. group similar clusterings by some dissimilarity function
   - e.g. Rand Index

- intuitive and powerful principle
- however: blind / undirected / unfocused / independent generation of solutions
  - → risk of determining highly similar clusterings
  - → inefficient

⇒ more systematic approaches required



Clustering Based on Given Knowledge

Basic idea

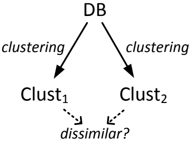
- generate a single clustering solution (or assume it is given)
- based on first clustering generate a **dissimilar** clustering

→ check dissimilarity **during** clustering process

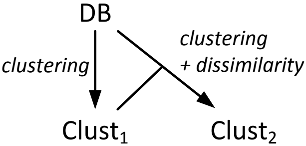
→ guide clustering process by given knowledge

→ similar clusterings are directly avoided

so far:



now:



General aim of Alternative Clustering

- given clustering $Clust_1$ and functions Q, $Diss$
- find clustering $Clust_2$ such that $Q(Clust_2)$ & $Diss(Clust_1, Clust_2)$ are high

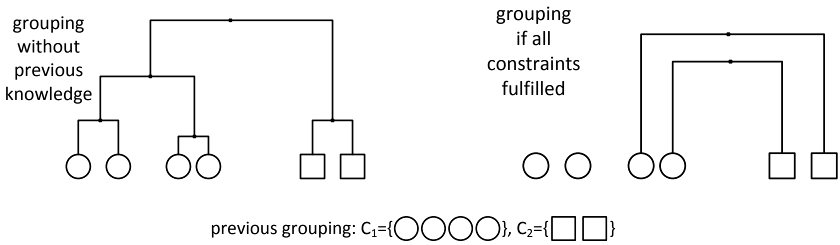
COALA (Bae & Bailey, 2006)

General idea of COALA

- avoid similar grouping of objects by using **instance level constraints**

$\rightarrow$ add cannot-link constraint $\mathit{cannot}(o, p)$ if $\{o, p\} \subseteq C \in \mathit{Clust}_1$

- hierarchical agglomerative average link approach
- try to group objects such that constraints are mostly satisfied
  - 100% satisfaction not meaningful
  - trade off quality vs. dissimilarity of clustering

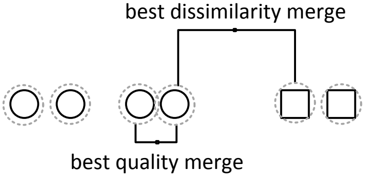


COALA: Algorithm

Determine which sets to merge

- given current grouping $P_1, \ldots, P_l$
- quality merge
  - assume no constraints are given
  - determine P_a, P_b with smallest average link distance d_{qual}
- dissimilarity merge
  - determine $(P_a, P_b) \in \mathit{Dissimilar}$ with smallest average link distance d_{diss}
  - $(P_i, P_j) \in \mathit{Dissimilar} \Leftrightarrow$ constraints **between** sets are fulfilled

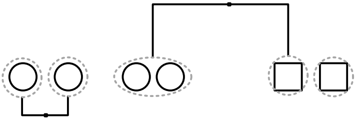
    $\Leftrightarrow \neg\exists o \in P_i, p \in P_j : \mathit{cannot}(o, p)$
- if $d_{qual} < w \cdot d_{diss}$ perform quality merge; otherwise dissimilarity merge



COALA: Algorithm

Determine which sets to merge

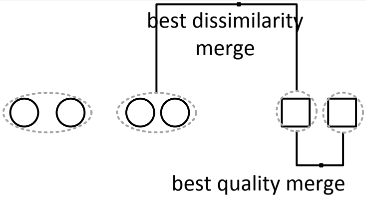
- given current grouping $P_1, \ldots, P_l$
- quality merge
  - assume no constraints are given
  - determine P_a, P_b with smallest average link distance d_{qual}
- dissimilarity merge
  - determine $(P_a, P_b) \in \mathit{Dissimilar}$ with smallest average link distance d_{diss}
  - $(P_i, P_j) \in \mathit{Dissimilar} \Leftrightarrow$ constraints **between** sets are fulfilled
    $\Leftrightarrow \neg\exists o \in P_i, p \in P_j : \mathit{cannot}(o, p)$
- if $d_{qual} < w \cdot d_{diss}$ perform quality merge; otherwise dissimilarity merge



COALA: Algorithm

Determine which sets to merge

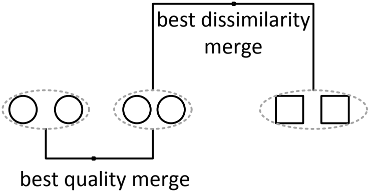
- given current grouping $P_1, \ldots, P_l$
- quality merge
  - assume no constraints are given
  - determine P_a, P_b with smallest average link distance d_{qual}
- dissimilarity merge
  - determine $(P_a, P_b) \in \mathit{Dissimilar}$ with smallest average link distance d_{diss}
  - $(P_i, P_j) \in \mathit{Dissimilar} \Leftrightarrow$ constraints **between** sets are fulfilled
    $\Leftrightarrow \neg\exists o \in P_i, p \in P_j : \mathit{cannot}(o, p)$
- if $d_{qual} < w \cdot d_{diss}$ perform quality merge; otherwise dissimilarity merge



COALA: Algorithm

Determine which sets to merge

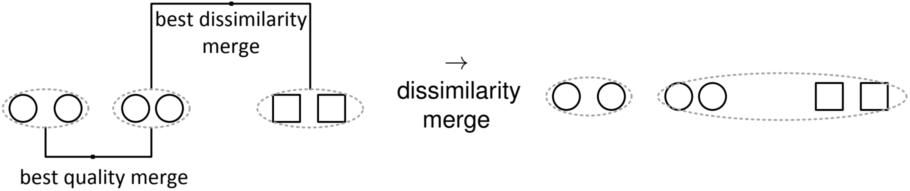
- given current grouping $P_1, \dots, P_l$
- quality merge
  - assume no constraints are given
  - determine P_a, P_b with smallest average link distance d_{qual}
- dissimilarity merge
  - determine $(P_a, P_b) \in \mathit{Dissimilar}$ with smallest average link distance d_{diss}
  - $(P_i, P_j) \in \mathit{Dissimilar} \Leftrightarrow$ constraints **between** sets are fulfilled
    $\Leftrightarrow \neg\exists o \in P_i, p \in P_j : \mathit{cannot}(o, p)$
- if $d_{qual} < w \cdot d_{diss}$ perform quality merge; otherwise dissimilarity merge



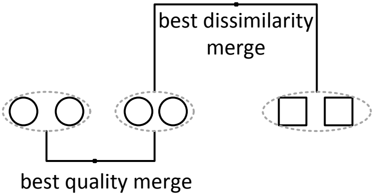
COALA: Algorithm

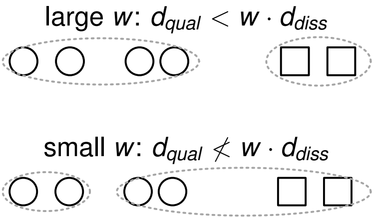
Determine which sets to merge

- given current grouping $P_1, \ldots, P_l$
- quality merge
  - assume no constraints are given
  - determine P_a, P_b with smallest average link distance d_{qual}
- dissimilarity merge
  - determine $(P_a, P_b) \in \mathit{Dissimilar}$ with smallest average link distance d_{diss}
  - $(P_i, P_j) \in \mathit{Dissimilar} \Leftrightarrow$ constraints **between** sets are fulfilled
    $\Leftrightarrow \neg\exists o \in P_i, p \in P_j : \mathit{cannot}(o, p)$
- if $d_{qual} < w \cdot d_{diss}$ perform quality merge; otherwise dissimilarity merge



COALA: Discussion





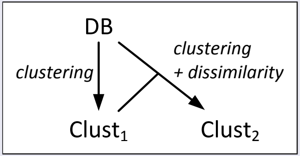
Discussion

- large w: prefer quality; small w: prefer dissimilarity
  - possible to trade off quality vs. dissimilarity
- hierarchical and/or flat partitioning of objects
- only distance function between objects required
- heuristic approach

Taxonomy

Classification into taxonomy

- COALA:
  - assumes given clustering
  - iteratively computes alternative
  - two clustering solutions are achieved



- further approaches from this category
  - (Chechik & Tishby, 2002; Gondek & Hofmann, 2003; Gondek & Hofmann, 2004): based on information bottleneck principle, able to incorporate arbitrary given knowledge
  - (Gondek & Hofmann, 2005): use of ensemble methods
  - (Dang & Bailey, 2010b): information theoretic approach, use of kernel density estimation, able to detect non-linear shaped clusters
  - (Gondek *et al.*, 2005): likelihood maximization with constraints, handels only binary data, able to use a set of clusterings as input
  - (Bae *et al.*, 2010): based upon comparison measure between clusterings, alternative should realize different density profile/histogram
  - (Vinh & Epps, 2010): based on conditional entropy, able to use a set of clusterings as input

Information Bottleneck Approaches

- information theoretic clustering approach
- enrich traditional approach by given knowledge/clustering

Information bottleneck principle

- two random variables: X (objects) and Y (their features/attribute values)
- find (probabilistic) clustering C that minimizes $F(C) = I(X, C) - \beta I(Y, C)$
- trade-off between
  - compression $\approx$ minimize mutual information $I(X, C)$
  - and preservation of information $\approx$ maximize mutual information $I(Y, C)$
- mutual information $I(Y, C) = H(Y) - H(Y|C)$ with entropy H
  - intuitively: how much is the uncertainty about Y decreased by knowing C

IB with Given Knowledge

Incorporate given clustering

- assume clustering D is already given, X objects, Y features
- (Chechik & Tishby, 2002): minimize $F_1(C) = I(X, C) - \beta I(Y, C) + \gamma I(D, C)$
- (Gondek & Hofmann, 2003): minimize $F_2(C) = I(X, C) - \beta I(Y, C|D)$
- (Gondek & Hofmann, 2004): maximize $F_3(C) = I(Y, C|D)$ such that $I(X, C) \leq c$ and $I(Y, C) \geq d$

- $I(X, C) \approx$ compression, $I(Y, C) \approx$ preservation of information
- $I(D, C) \approx$ similarity between D and C
- $I(Y, C|D) \approx$ preservation of information if C **and** D are used

Discussion

- able to incorporate arbitrary knowledge
- joint distributions have to be known

Drawbacks of Alternative Clustering Approaches

Drawback 1: Single alternative

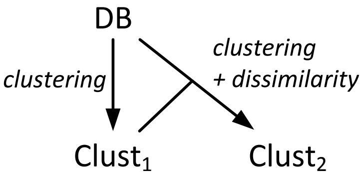
- usually only one alternative is extracted
- given $Clust_1 \rightarrow$ extract $Clust_2$
- thus, two clusterings determined
- however, multiple (≥ 2) clusterings possible

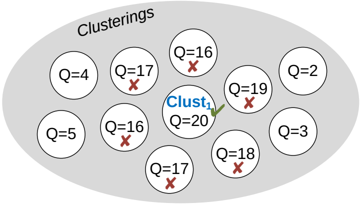
- naive extension problematic
  - given $Clust_1 \rightarrow$ extract $Clust_2$, given $Clust_2 \rightarrow$ extract $Clust_3$, ...
  - one ensures: $Diss(Clust_1, Clust_2)$ and $Diss(Clust_2, Clust_3)$ high
  - but no conclusion about $Diss(Clust_1, Clust_3)$ possible
  - often/usually they should be very similar
- more complex extension necessary
  - given $Clust_1 \rightarrow$ extract $Clust_2$
  - given $Clust_1$ **and** $Clust_2 \rightarrow$ extract $Clust_3$
  - ...

Drawbacks of Alternative Clustering Approaches

Drawback 2: Iterative processing

- already generated solutions cannot be modified anymore
- greedy selection of clustering solutions
- $\sum_i Q(Clust_i)$ need not to be high
  - clusterings with very low quality possible



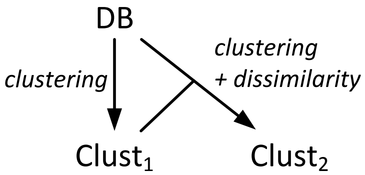


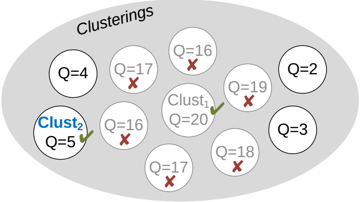
Other approach: Detect all clusterings **simultaneously**

Drawbacks of Alternative Clustering Approaches

Drawback 2: Iterative processing

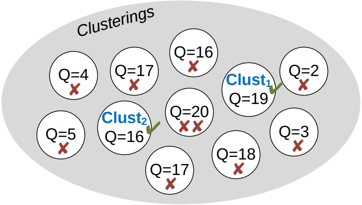
- already generated solutions cannot be modified anymore
- greedy selection of clustering solutions
- $\sum_i Q(Clust_i)$ need not to be high
  - clusterings with very low quality possible

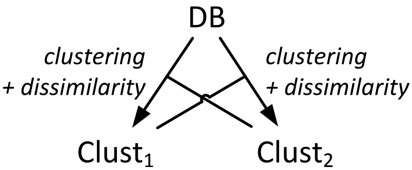




Other approach: Detect all clusterings **simultaneously**

Simultaneous Generation of Multiple Clusterings





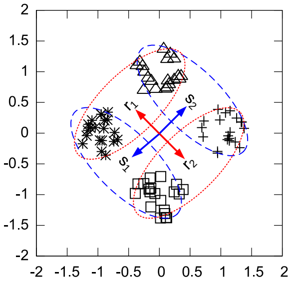
Basic idea

- simultaneous generation of clusterings $Clust_1, \ldots, Clust_m$
- make use of a combined objective function
- informally: maximize $\sum_i Q(Clust_i) + \sum_{i \neq j} Diss(Clust_i, Clust_j)$

Decorrelated k-Means (Jain *et al.*, 2008)

Decorrelated k-Means: Notions

- k clusters of $Clust_1$ are represented by vectors $r_1, \ldots, r_k$
  - objects are assigned to its nearest representative
  - yielding clusters $C_1, \ldots, C_k$
  - note: representatives may not be the mean vectors of clusters
  - means denoted with $\alpha_1, \ldots, \alpha_k$
- analogously: representatives $s_1, \ldots, s_l$ for $Clust_2$
  - clusters $D_1, \ldots, D_l$ and mean vectors of clusters $\beta_1, \ldots, \beta_l$



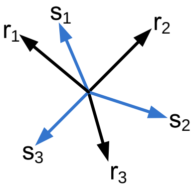
intuition:

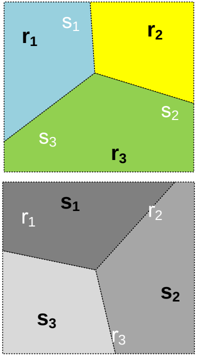
- each cluster should be compact and
- representatives should be different (mostly orthogonal)

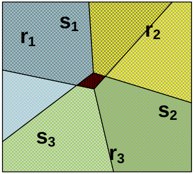
Decorrelated k-Means: Objective Function

minimize objective function $G(r_1, \ldots, r_k, s_1, \ldots, s_l) =$

$$\underbrace{\sum_i \sum_{x \in C_i} \|x - r_i\|^2 + \sum_j \sum_{x \in D_j} \|x - s_j\|^2}_{\text{compactness of both clusterings}} + \underbrace{\lambda \sum_{i,j} (\beta_j^T \cdot r_i)^2 + \lambda \sum_{i,j} (\alpha_i^T \cdot s_j)^2}_{\text{difference/orthogonality of representatives}}$$







intuition of orthogonality: cluster labels generated by nearest-neighbor assignments are independent

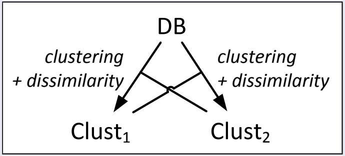
Decorrelated k-Means: Discussion

Discussion

- enables parametrization of desired number of clusterings
  - $T \geq 2$ clusterings can be extracted
- discriminative approach

Classification into taxonomy

- Decorrelated k-Means:
  - no clustering given
  - simultaneous computation of clusterings
  - T alternatives



- further approaches from this category
  - CAMI (Dang & Bailey, 2010a): generative model based approach, each clustering is a Gaussian mixture model
  - (Hossain *et al.*, 2010): use of contingency tables, detects only 2 clusterings, can handle two different databases (relational clustering)

A Generative Model Based Approach

Idea of CAMI (Dang & Bailey, 2010a)

- generative model based approach
- each clustering $Clust_i$ is a Gaussian mixture model (parameter Θ_i)
  - $p(x|\Theta_i) = \sum_{j=1}^{k} \lambda_i^j \mathcal{N}(x, \mu_i^j, \Sigma_i^j) = \sum_{j=1}^{k} p(x|\theta_i^j)$
- quality of clusterings is measured by likelihood
  - $L(\Theta_i, DB) = \sum_{x \in DB} \log p(x|\Theta_i)$
- (dis-)similarity by mutual information (KL divergence)
  - $I(Clust_1, Clust_2) = \sum_{j,j'} I(p(x|\theta_1^j), p(x|\theta_2^{j'}))$
- combined objective function
  - maximize $\underbrace{L(\Theta_1, DB) + L(\Theta_2, DB)}_{\text{likelihood}} - \underbrace{\mu I(\Theta_1, \Theta_2)}_{\text{mutual information}}$
- expectation maximization framework to determine clusterings

Contingency tables to model dissimilarity

Idea of (Hossain *et al.*, 2010)

- contingency table for clusterings: highest dissimilarity if uniform distribution

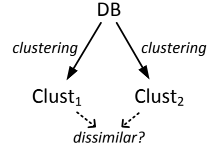
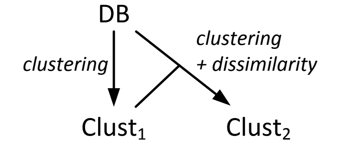
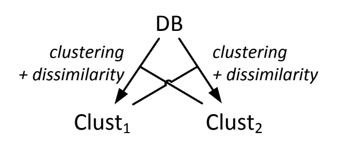
→ maximize uniformity of contingency table

- however: arbitrary clusterings not meaningful due to quality properties
- solution: represent clusters by prototypes
  → quality of clusterings ensured
- determine prototypes (and thus clusterings) that maximize uniformity

Discussion

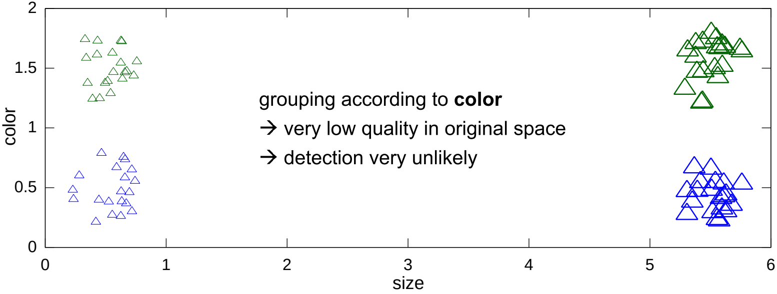
- detects only 2 clusterings
- but presents more general framework
  - can handle two different databases → relational clustering
  - also able to solve dependent clustering (diagonal matrix)

Preliminary Conclusion for this Paradigm

<table>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td>independent computation</td><td colspan="2">focused on dissimilarity</td></tr>
<tr><td></td><td>iterative computation</td><td>simultaneous computation</td></tr>
<tr><td></td><td>based on previous knowledge</td><td>no knowledge required</td></tr>
<tr><td></td><td>usually just 2 clusterings</td><td>often $\geq$ 2 clusterings possible</td></tr>
<tr><td>arbitrary clustering definition</td><td colspan="2">specialized clustering definitions</td></tr>
<tr><td></td><td colspan="2">methods are designed to detect multiple clusterings in the same data space</td></tr>
</table>

Open Challenges w.r.t. this Paradigm

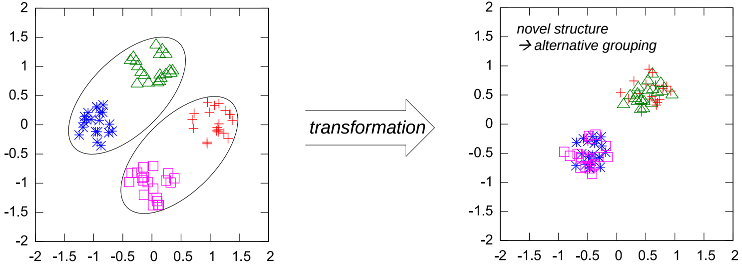
- methods are designed for individual clustering algorithms
- can good alternatives be expected in the same space?
  - consider clustering as aggregation of objects
  - main factors/components/characteristics of the data are captured
  - alternative clusterings should group according to different characteristics
  - main factors obfuscate these structures in the original space



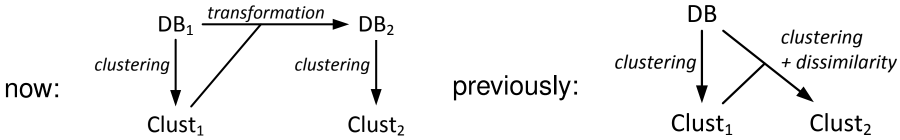
Overview

Motivation: Multiple Clusterings by Transformations

- previously: clustering in the same data space
  - → explicit check of dissimilarity during clustering process
  - → dependent on selected clustering definition
- now: iteratively transform and cluster database
  - "learn" transformation based on previous clustering result
  - → transformation can highlight novel structures
  - → any algorithm can be applied to (transformed) database
  - → dissimilarity only implicitly ensured



General idea



General aim

- given database DB and clustering $Clust_1$
- find transformation T, such that
  - clustering of $DB_2 = \{T(x) \mid x \in DB\}$ yields $Clust_2$ and
  - $Diss(Clust_1, Clust_2)$ is high

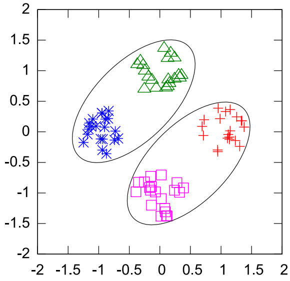
Observation: One has to avoid complete distortion of the original data

- approaches focus on linear transformations of the data
- find **transformation matrix** M; thus, $T(x) = M \cdot x$

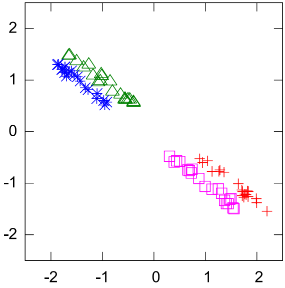
A Metric Learning Approach

Basic idea of approach (Davidson & Qi, 2008)

- given clustering poses constraints
  - similar objects in one cluster (must-link)
  - dissimilar objects in different clusters (cannot-link)
- make use of any metric learning algorithm
  - learn a transformation D such that **known** clustering is easily observable
- determine "alternative" transformation M based on D



$$D = \begin{pmatrix} 1.5 & -1 \\ -1 & 1 \end{pmatrix}$$



Transformation

Determine the "alternative" transformation

- given learned transformation metric D
- SVD provides a decomposition: $D = H \cdot S \cdot A$
- informally: $D = \mathit{rotate} \cdot \mathit{stretch} \cdot \mathit{rotate}$

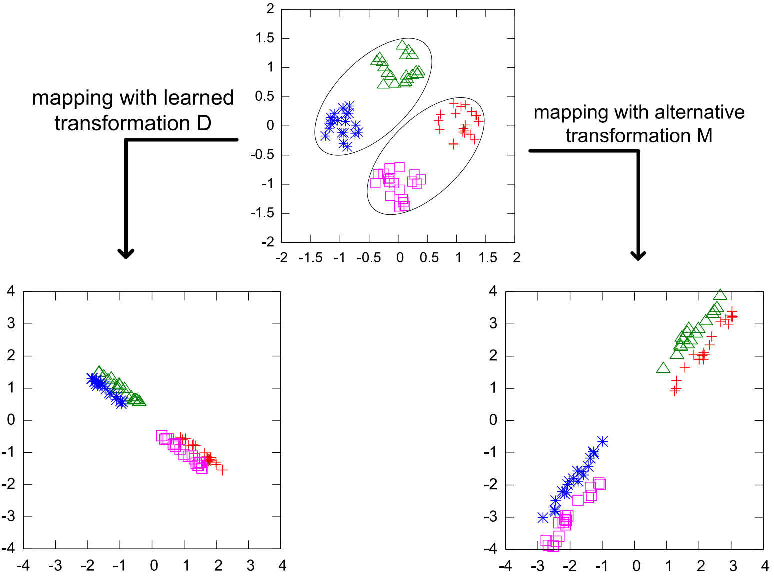
$\rightarrow$ invert stretcher matrix to get alternative M

- $M = H \cdot S^{-1} \cdot A$

$$D = \begin{pmatrix} 1.5 & -1 \\ -1 & 1 \end{pmatrix} = H \cdot S \cdot A = \begin{pmatrix} 0.79 & -0.62 \\ -0.62 & -0.79 \end{pmatrix} \begin{pmatrix} 2.28 & 0 \\ 0 & 0.22 \end{pmatrix} \begin{pmatrix} 0.79 & -0.62 \\ -0.62 & -0.79 \end{pmatrix}$$

$$M = \begin{pmatrix} 2 & 2 \\ 2 & 3 \end{pmatrix} = H \cdot S^{-1} \cdot A = H \cdot \begin{pmatrix} 0.44 & 0 \\ 0 & 4.56 \end{pmatrix} \cdot A$$

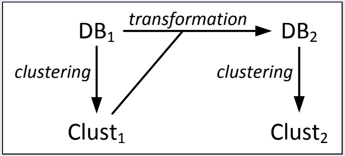
Exemplary transformations



Taxonomy

Classification into taxonomy

- (Davidson & Qi, 2008):
  - assumes given clustering
  - iteratively computes alternative
  - two clustering solutions are achieved



- further approach from this category: (Qi & Davidson, 2009)
  - constrained optimization problem
    - transformed data should preserve characteristics
    - but distance of points to previous cluster means should be high
  - able to specify which parts of clustering to keep or to reject
  - trade-off between alternativeness and quality

A Constraint based Optimization Approach

Basic idea (Qi & Davidson, 2009)

- transformed data should preserve characteristics as much as possible
  - $p(x)$ is probability distribution of the original data space
  - $p_M(y)$ of the transformed data space
- find transformation M that minimizes Kullback-Leibler divergence $min_M KL(p(x)||p_M(y))$
- keep in mind: original clusters should not be detected

→ add constraint $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1,x_i \notin C_j}^{k} \|x_i - m_j\|_B \leq \beta$ with $B = M^T M$ and Mahalanobis distance $\|\cdot\|_B$

- intuition:
  - $\|x_i - m_j\|_B$ is distance in transformed space
  - enforce small distance in new space only for $x_i \notin C_j$

  → distance to 'old' mean m_i should be high after transformation

  → novel clusters are expected

Resulting Transformation

Solution

- optimal solution of constraint optimization problem

$$M = \widetilde{\Sigma}^{-1/2} \text{ with } \widetilde{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1, x_i \notin C_j}^{k}(x_i - m_j)(x_i - m_j)^T$$

- advantage: closed-form

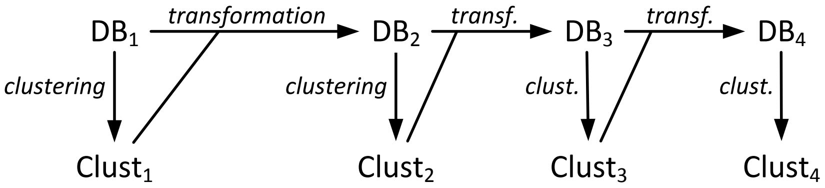
Discussion

- paper presents more general approach
  - able to specify which parts of clustering to keep or to reject
  - trade-off between alternativeness and quality
- as the previous approach: just one alternative

Drawbacks of previous approaches

The problem of just one alternative

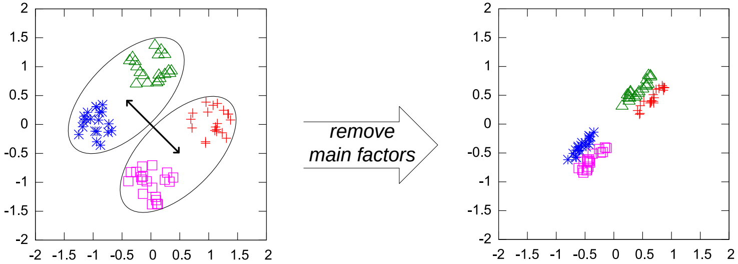
- extension to multiple views non-trivial
  - cf. alternative clustering approaches in the original space
- how to obtain novel structure after each iteration?



Dimensionality Reducing Transformation

How to obtain novel structure after each iteration?

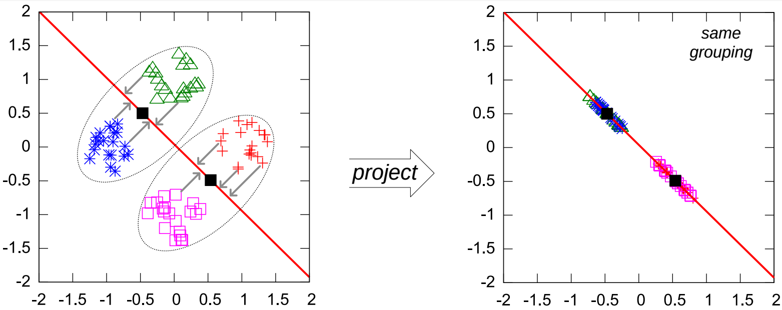
- make use of dimensionality reduction techniques
- first clustering determines main factors/principle components of the data
- transformation "removes" main factors
- retain only residue/orthogonal space
- previously weak factors are highlighted



Orthogonal Subspace Projections (Cui *et al.*, 2007)

Step 1: Determine the 'explanatory' subspace

- given $Clust_i$ of DB_i → determine mean vectors of clusters $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$
- find feature subspace A that captures clustering structure well
  - e.g. use PCA to determine strong principle components of the means
  - $A = [\phi_1, \ldots, \phi_p] \in \mathbb{R}^{d \times p}$ $p < k, p < d$
  - intuitively: $DB_i^A = \{A \cdot x \mid x \in DB_i\}$ yields the **same clustering**

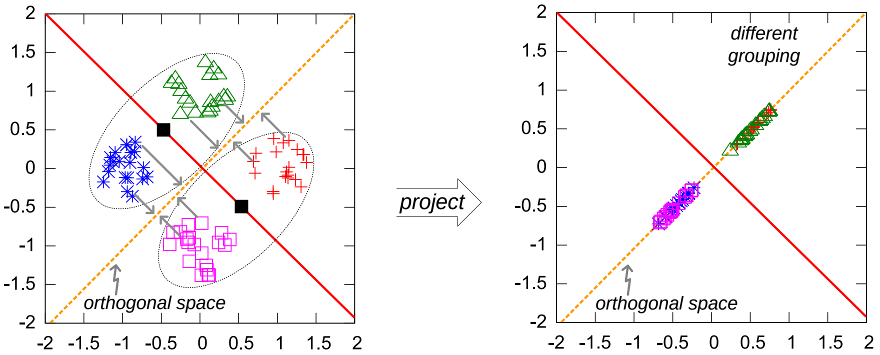


extended version: (Cui *et al.*, 2010)

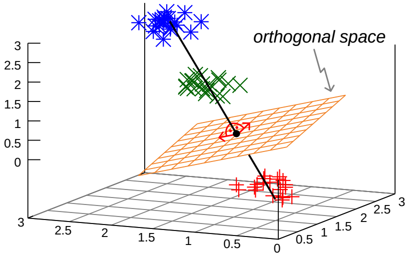
Orthogonalization

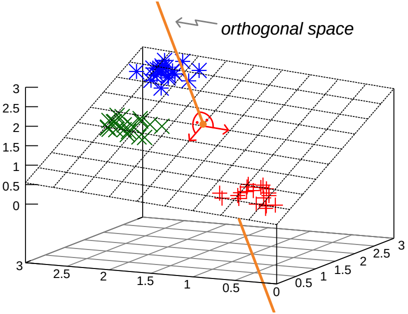
Step 2: Determine the orthogonal subspace

- orthogonalize subspace A to get novel database
  - $M_i = I - A \cdot (A^T \cdot A)^{-1} \cdot A^T \in \mathbb{R}^{d \times d}$
  - $DB_{i+1} = \{M_i \cdot x \mid x \in DB_i\}$



Examples & Discussion

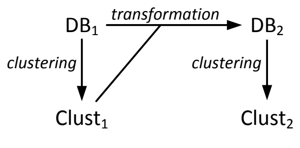
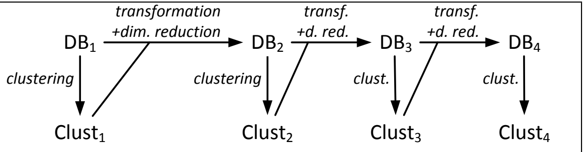




Discussion

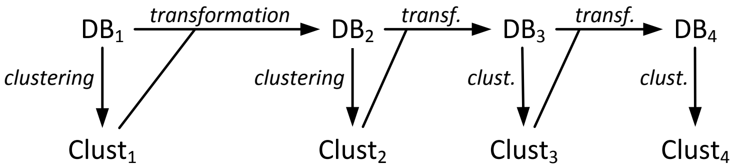
- potentially not appropriate for *low* dimensional spaces
  - dimensionality reduction problematic
- independent of reduction techniques, e.g. use PCA, LDA
- more than two clusterings possible
  - advantage: number of clusterings automatically determined

Preliminary Conclusion for this Paradigm

<table>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">focused on dissimilarity (implicitly by transformation)</td></tr>
<tr><td colspan="2">iterative computation</td></tr>
<tr><td colspan="2">(transformation is) based on previous knowledge</td></tr>
<tr><td>2 clusterings extracted</td><td>$\geq$ 2 clusterings extracted
(by using dimensionality reduction)</td></tr>
<tr><td colspan="2">independent of the used clustering algorithm</td></tr>
<tr><td colspan="2">detect multiple clusterings based on space transformations</td></tr>
</table>

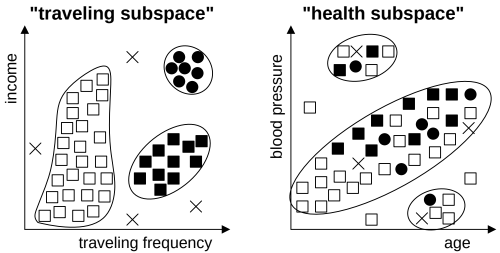
Open Challenges w.r.t. this Paradigm

- potentially very similar/redundant clusterings in subsequent iterations
  - dissimilarity only implicitly ensured for next iteration
- only iterative/greedy processing
  - cf. alternative clustering approaches in a single space
- difficult interpretation of clusterings based on space transformations
- initial clustering is based on the full-dimensional space
  - in high-dimensional spaces not meaningful



Overview

Motivation: Multiple Clusterings in Subspaces



Clustering in Subspace Projections

- Cluster are observed in arbitrary attribute combinations (**subspaces**) **using the original attributes** (no transformations)

$\Rightarrow$ Cluster interpretation based on relevant attributes

- Detect **multiple clusterings in different subspace projections** as each object can be clustered differently in each projection

$\Rightarrow$ Detect a **group of objects and subset of attributes** per cluster

Abstract Problem Definition

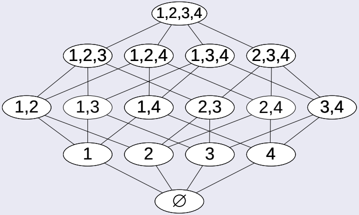
Abstract subspace clustering definition

- Definition of object set O clustered in subspace S

  $$C = (O, S) \text{ with } O \subseteq DB, S \subseteq DIM$$

- Selection of result set M a subset of all valid subspace clusters ALL

  $$M = \{(O_1, S_1) \ldots (O_n, S_n)\} \subseteq ALL$$



Overview of paradigms:

- Subspace clustering: focus on definition of (O, S)

$\Rightarrow$ Output all (multiple) valid subspace clusters $M = ALL$

- Projected clustering: focus on definition of disjoint clusters in M

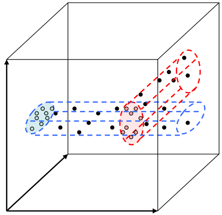
$\Rightarrow$ Unable to detect objects in multiple clusterings

Contrast to the Projected Clustering Paradigm

First approach: PROCLUS (Aggarwal *et al.*, 1999)

- Based on iterative processing of k-Means
- Selection of compact projection
- Exclude highly deviating dimensions

⇒ Basic model, fast algorithm



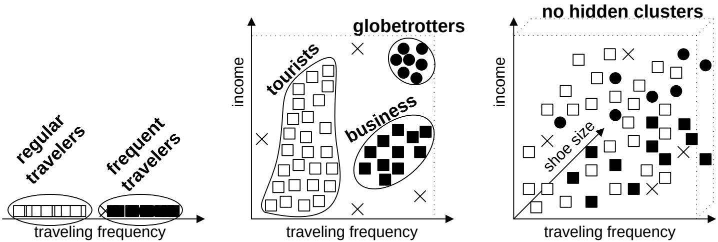
⇒ **Only a single clustering solution!**

- **ORCLUS**: arbitrary oriented projected clusters (Aggarwal & Yu, 2000)
- **DOC**: monte carlo processing (Procopiuc *et al.*, 2002)
- **PreDeCon/4C**: correlation based clusters (Böhm *et al.*, 2004a; Böhm *et al.*, 2004b)
- **MrCC**: multi-resolution indexing technique (Cordeiro *et al.*, 2010)

Subspace Cluster Models (O, S)

Clusters are hidden in arbitrary subspaces with individual (dis-)similarity:

$$dist^S(o,p) = \sqrt{\sum_{i \in S}(o_i - p_i)^2}$$



⇒ How to find clusters in arbitrary projections of the data?

⇒ Consider multiple valid clusters in different subspaces

Challenges

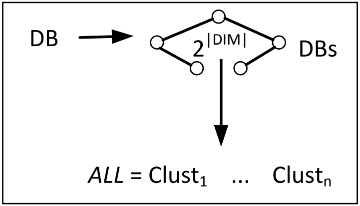
Traditional focus on ($O \subseteq DB$, $S \subseteq DIM$)

- Cluster detection in arbitrary subspaces $S \subseteq DIM$

$\Rightarrow$ Pruning the **exponential number of cluster candidates**

- Clusters as subsets of the database $O \subseteq DB$

$\Rightarrow$ Overcome **excessive database access** for cluster computation

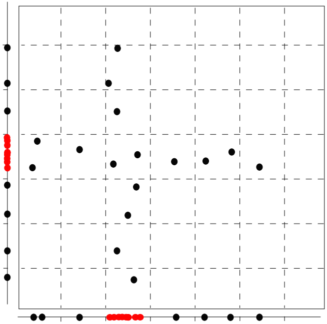


Surveys cover basically this traditional perspective on subspace clustering: (Parsons *et al.*, 2004; Kriegel *et al.*, 2009)

Additional challenge: ($M \subseteq ALL$)

- Selection of **meaningful** (e.g. non-redundant) **result set**

First approach: CLIQUE (Agrawal *et al.*, 1998)



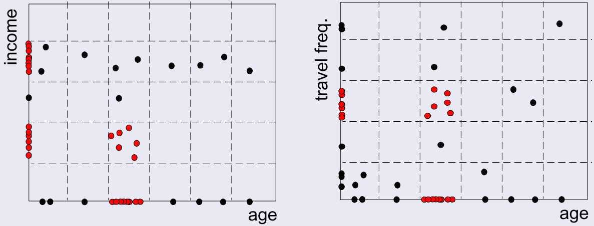
- First subspace clustering algorithm
- Aims at automatic identification of subspace clusters in high dimensional databases
- Divide data space into fixed grid-cells by equal length intervals in each dimension

- Cluster model:
  Clusters (dense cells) contain more objects than a threshold τ
- Search for all dense cells in all subspaces...

Multiple Clusters in Any Subspace Projection

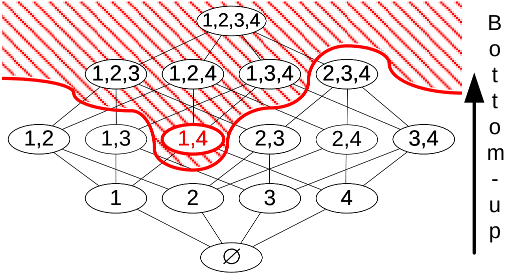
Multiple clustering solutions

- CLIQUE detects each object in multiple dense cells...



- Based on definition of dense cells one has to search in all subspaces... Do we have to check all of the $2^{|DIM|}$ projections?
- No. The search space can be pruned (without loss of results).
- Interleaved processing (**object set** and **dimension set**): Detection of dense cells in a bottom-up search on the subspace lattice...

Basic Idea for Search Space Pruning



Pruning based on monotonicity

- Monotonicity (e.g. in CLIQUE):

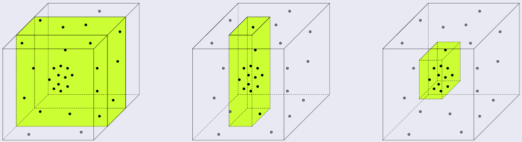
$$O \text{ is dense in } S \Rightarrow \forall T \subseteq S \ : \ O \text{ is dense in } T$$

- Higher dimensional projections of a non-dense region are pruned.
- Density has to be checked via an expensive database scan.
- Idea based on the apriori principle (Agrawal & Srikant, 1994)

Enhancements based on grid-cells

SCHISM (Sequeira & Zaki, 2004)

- Observation in subspace clustering:
  Density (number of objects) decreases with increasing dimensionality
- Fixed thresholds are not meaningful,
  enhanced techniques adapt to the dimensionality of the subspace
- SCHISM introduced the first **decreasing threshold function**



- **MAFIA**: enhanced grid positioning (Nagesh *et al.*, 2001)
- **P3C**: statistical selection of dense-grid cells (Moise *et al.*, 2006)
- **DOC** / **MineClus**: enhanced quality by flexible positioning of cells
  (Procopiuc *et al.*, 2002; Yiu & Mamoulis, 2003)

SCHISM - Threshold Function

Goal: define efficiently computable threshold function

Idea: Chernoff-Hoeffding bound: $Pr[Y \geq E[Y] + nt] \leq e^{-2nt^2}$

- X_s is a random variable denoting
  the number of points in grid-cell of dimensionality s

$\Rightarrow$ A cluster with n_s objects has $Pr[X_s \geq n_s] \leq e^{-2nt_s^2} \leq \tau$
i.e. the probability of observing so many object is very low...

- Derive $\tau(|S|)$ as a **non-linear monotonically decreasing** function in the number of dimensions

$$\tau(s) = \frac{E[X_s]}{n} + \sqrt{\frac{1}{2n} \ln \frac{1}{\tau}}$$

- Assumption: d-dimensional space is independent and uniformly distributed and discretized into ξ intervals

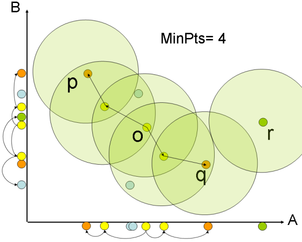
$\Rightarrow$ $Pr[\text{a point lies in a s-dimensional cell}] = (\frac{1}{\xi})^s$

$\Rightarrow$ $\frac{E[X_s]}{n} = (\frac{1}{\xi})^s$

Density-Based Subspace Clustering

SUBCLU (Kailing *et al.*, 2004b)

- Subspace clustering extension of DBSCAN (Ester *et al.*, 1996)
- Enhanced density notion compared to grid-based techniques
- Arbitrary shaped clusters and noise robustness
- However, highly inefficient for subspace clustering



- **INSCY**: efficient indexing of clusters (Assent *et al.*, 2008)
- **FIRES**: efficient approximate computation (Kriegel *et al.*, 2005)
- **DensEst**: efficient density estimation (Müller *et al.*, 2009a)

Preliminary Conclusion on Subspace Clustering

- **Benefits** of subspace clustering methods:
  - each object is clustered in multiple subspace clusters
  - selection of relevant attributes in high dimensional databases
  - focus on cluster definitions (O, S) in any subspace S
- **Drawbacks** of subspace clustering methods:
  - Provides only one set of clusters $\{(O_1, S_1), (O_2, S_2), \ldots, (O_n, S_n)\}$
  - Not aware of the different clusterings: $\{(O_1, S_1), (O_2, S_2)\} vs. \{(O_3, S_3), (O_4, S_4)\}$
  - Not aware of the different subspaces: $S_1 = S_2$ and $S_3 = S_4$ while $S_2 \neq S_3$

  $\Rightarrow$ **Does not ensure dissimilarity** of subspace clusters

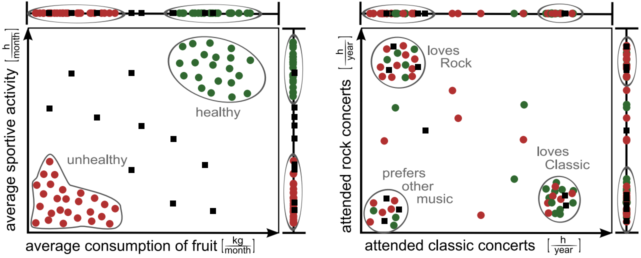
  $\Rightarrow$ **Not able to compute alternatives** w.r.t. a given clustering

$\Rightarrow$ This research area is contributing by a variety of established clustering models detecting multiple clustering solutions.

- However, **enforcing different clustering solutions** is not in its focus!

Open Challenges for Multiple Clusterings

- Ensuring the difference of subspace projections
- Eliminating redundancy of subspace clusters



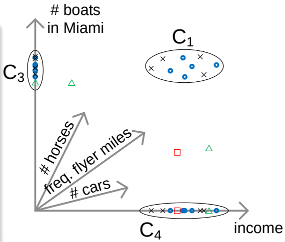
Results out of evaluation study (Müller *et al.*, 2009b)

- Redundancy is the reason for:
  - low quality results
  - high runtimes (not scaling to high dimensional data)

Non-Redundant Subspace Clustering Overview

Redundant results

- Exponentially many **redundant projections** of one hidden subspace cluster
  - No benefit by these redundant clusters
  - Computation cost (scalability)
  - Overwhelming result sets



$\Rightarrow$ Novel (general) techniques for redundancy elimination required...

- **DUSC**: local pairwise comparison of redundancy (Assent *et al.*, 2007)
- **StatPC**: statistical selection of non-redundant clusters (Moise & Sander, 2008)
- **RESCU**: including interesting and excluding redundant clusters (Müller *et al.*, 2009c)

STATPC: Selection of Representative Clusters

General idea:

- Result should be able to **explain all other clustered regions**

Underlying cluster definition

- Based on P3C cluster definition (Moise *et al.*, 2006)
- Could be exchanged in more general processing...

Statistical selection of clusters

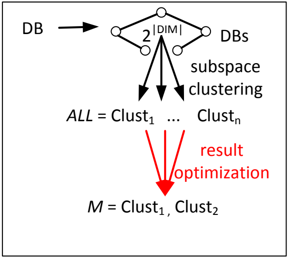
- A redundant subspace cluster can be explained by
  a set of subspace clusters in the result set
- Current subspace cluster result set defines a mixture model
- Test explain relation by statistical significance test:
  Explained, if the true number of clustered objects is not significantly
  larger or smaller than what can be expected under the given model

Result Optimization for Multi View Clustering

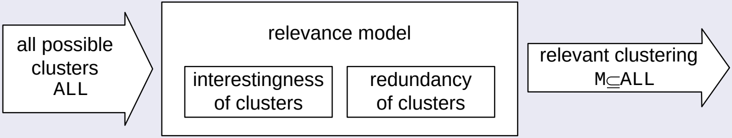
- **Removing redundancy**
- **Including multiple views**

\+ Model the difference between subspaces

⇒ Exclude redundant clusters in similar subspaces
Allow novel knowledge represented in dissimilar subspaces



Abstract redundancy model: RESCU (Müller *et al.*, 2009c)



...does not include similarity of subspaces!

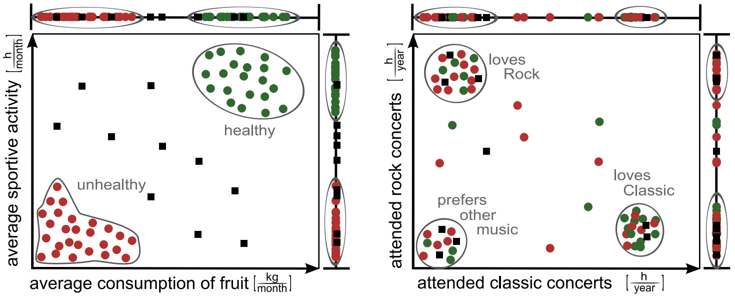
Orthogonal Concepts in Subspace Projections

OSCLU (Günnemann *et al.*, 2009)

- Orthogonal concepts share no or only few common attributes

⇒ We prune the detection of similar concepts (in similar subspaces)

⇒ We select an optimal set of clusters in orthogonal subspaces



Optimal Choice of Orthogonal Subspaces

Abstract subspace clustering definition

- Definition of object set O clustered in subspace S

$$C = (O, S) \text{ with } O \subseteq DB, S \subseteq DIM$$

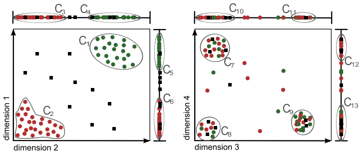
- Selection of result set M a subset of all valid subspace clusters ALL

$$M = \{(O_1, S_1) \dots (O_n, S_n)\} \subseteq ALL$$

Definition of cluster $C = (O, S)$ and clustering $M = \{C_1, \dots, C_n\} \subseteq All$

$\Rightarrow$ Choose optimal subset $Opt \subseteq All$ out of all subspace clusters

1. avoid similar concepts (subspaces) in the result
2. each cluster should provide novel information



Almost Orthogonal Concepts

Extreme cases:

1. Allow only disjoint attribute selection
2. Exclude only lower dimensional projections

$\Rightarrow$ allow overlapping concepts, but avoid too many shared dimensions

$\Rightarrow$ similar concepts: high fraction of common dimensions

Covered Subspaces (β fraction of common dimensions)

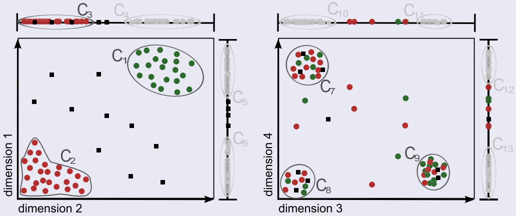
$$\mathit{coveredSubspaces}_\beta(S) = \{T \subseteq \mathit{Dim} \mid |T \cap S| \geq \beta \cdot |T|\}$$

with $0 < \beta \leq 1$. For $\beta \to 0$ we get the first, for $\beta = 1$ the second definition.

$\{1,2\}\ \cancel{\mathit{covers}}\ \{3,4\}$	different concepts	☺
$\{1,2\}\ \cancel{\mathit{covers}}\ \{2,3,4\}$	different concepts	☺
$\{1,2,3,4\}\ \mathit{covers}\ \{1,2,3\}$	similar concepts	☺
$\{1,\ldots,9,10\}\ \mathit{covers}\ \{1,\ldots,9,11\}$	similar concepts	☺

Allowing overlapping clusters

1. avoid similar subspaces (concept group)
2. each cluster should provide novel information (within its concept group)



Global interestingness

Cluster $C = (O, S)$ and clustering $M = \{C_1, \ldots, C_n\} \subseteq All$

$I_{global}(C, M) =$ fraction of new objects in C within its concept group

Orthogonal clustering

The clustering $M = \{C_1, \ldots, C_n\} \subseteq All$ is orthogonal iff

$$\forall C \in M : I_{global}(C, M \backslash \{C\}) \geq \alpha$$

Optimal Orthogonal Clustering

Formal Definition

Given the set *All* of all possible subspace clusters, a clustering $Opt \subseteq All$ is an optimal orthogonal clustering iff

$$Opt = \arg \max_{M \in Ortho} \left\{ \sum_{C \in M} I_{local}(C) \right\}$$

with

$$Ortho = \{M \subseteq All \mid M \text{ is an orthogonal clustering}\}$$

Local interestingness

- dependent on application, flexibility
- size, dimensionality, ...

NP-hard Problem

Theorem: Computing an Optimal Orth. Clustering is NP-hard

- Idea of Proof: Reduction to *SetPacking* problem
  - given several finite sets O_i
  - find maximal number of disjoint sets
- each set O_i is mapped to the cluster $C_i = (O_i, \{1\})$
- disjoint sets: choose $\alpha = 1$
- maximal number of sets: $I_{local}(C) = 1$

$\Rightarrow$ our model generates valid *SetPacking* solution

Optimal Orthogonal Clustering is a more general problem

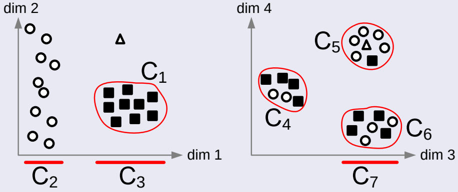
Optimal Orthogonal Clustering is NP-hard $\Rightarrow$ approximate algorithm

Alternative Subspace Clustering

ASCLU (Günnemann *et al.*, 2010)

- Aim: extend the idea of alternative clusterings to subspace clustering
- Intuition: subspaces represent views; differing views may reveal different clustering structures
- Idea: utilize the principle of OSCLU to find an alternative clustering *Res* for a given clustering *Known*

A valid clustering *Res* has to fulfill all properties defined in OSCLU but additionally has to be a valid alternative to *Known*.



E.g.: If $Known = \{C_2, C_5\}$, then $Res = \{C_3, C_4, C_7\}$ would be a valid clustering.

Extending Subspace Clustering by Given Knowledge

A valid clustering *Res* has to fulfill all properties defined in OSCLU but additionally has to be a valid alternative to *Known*.

Given a cluster $C \in Res$, $C = (O, S)$ is a valid alternative cluster to *Known* iff

$$\frac{|O \setminus AlreadyClustered(Known, C)|}{|O|} \geq \alpha$$

where $0 < \alpha \leq 1$ and

$$AlreadyClustered(Known, C) = \bigcup_{(O,S)=K \in Known} \{O \mid K \in ConceptGroup(C, Known)\}$$

Valid alternative subspace Clustering

Given a clustering $Res \subseteq All$, *Res* is a valid alternative clustering to *Known* iff all clusters $C \in Res$ are valid alternative clusters to *Known*.

Subspace Search: Selection Techniques

- Estimating the quality of a whole subspace
- Selection of interesting subspaces

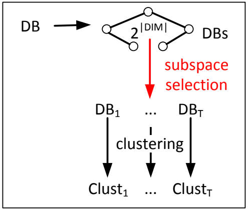
$\Rightarrow$ Decoupling subspace and cluster detection

- However, quality might be only locally visible in each subspace

$\Rightarrow$ Is global estimation meaningful?

Subspace Clustering: individual subspace per cluster

Subspace Search: restricted set of subspaces

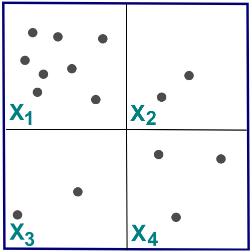


- **ENCLUS**: entropy-based subspace search (Cheng *et al.*, 1999)
- **RIS**: density-based subspace search (Kailing *et al.*, 2003)
- **mSC**: multiple spectral clustering views
  enforce different subspaces (Niu & Dy, 2010)

ENCLUS: Subspace Quality Estimation

- Based on the CLIQUE subspace clustering model
- Entropy as a measure for:
  - High coverage of the CLIQUE clustering
  - High density of individual subspace clusters
  - High correlation between the relevant dimensions

$\Rightarrow$ Low entropy indicates highly interesting subspaces...



Entropy of a subspace

$$H(X) = -\sum_{x \in \mathcal{X}} d(x) \cdot \log d(x)$$

with the density $d(x)$ of each cell $x \in$ grid $\mathcal{X}$
(i.e. percentage of objects in x)

mSC: Enforcing Different Subspaces

General idea:

- Optimize cluster quality and subspace difference (cf. simultaneous objective function (Jain *et al.*, 2008))

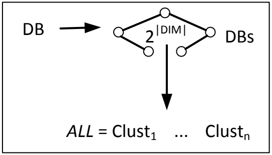
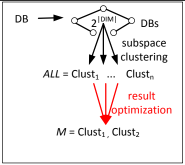
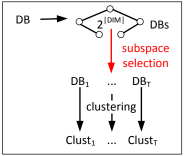
Underlying cluster definition

- Using spectral clustering (Ng *et al.*, 2001)
- Could be exchanged in more general processing...

Measuring subspace dependencies

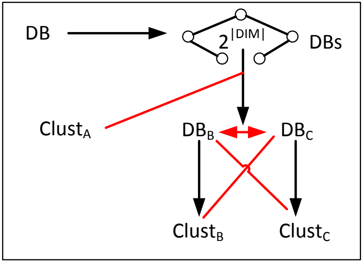
- Based on the Hilbert-Schmidt Independence Criterion (Gretton *et al.*, 2005)
- Measures the statistical dependence between subspaces
- Steers subspace search towards independent subspaces
- Includes this as penalty into spectral clustering criterion

Overview for this Paradigm

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>no dissimilarity
(ALL)</td>
<td>consider dissimilarity
(e.g. redundancy)</td>
<td>first approach
with dissimilarity</td>
</tr>
<tr>
<td colspan="3">simultaneous processing</td>
</tr>
<tr>
<td colspan="3">only recent approaches use previous knowledge</td>
</tr>
<tr>
<td>too many clusters</td>
<td>optimized result size</td>
<td>(clustering step)</td>
</tr>
<tr>
<td colspan="2">dependent on the used clustering algorithm</td>
<td>independent step</td>
</tr>
<tr>
<td colspan="3">enable interpretation of multiple clusterings</td>
</tr>
</table>

Open Challenges w.r.t. this Paradigm

- Awareness of different clusterings
  - dissimilarity only between clusters not between clusterings
  - grouping of clusters in common subspaces required
- Simultaneous processing
  - decoupling of existing solutions with high interdependences
- Including knowledge about previous clustering solutions
  - steering of subspace clustering to alternative solutions



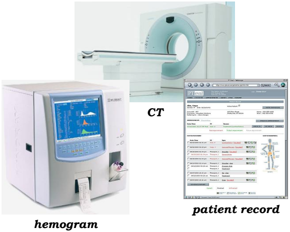
Overview

1. Motivation, Challenges and Preliminary Taxonomy
2. Multiple Clustering Solutions in the Original Data Space
3. Multiple Clustering Solutions by Orthogonal Space Transformations
4. Multiple Clustering Solutions by Different Subspace Projections
5. Clustering in Multiple Given Views/Sources
6. Summary and Comparison in the Taxonomy

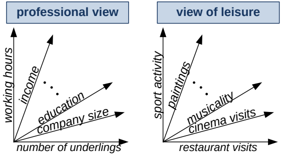
Motivation: Multiple Data Sources

Usually it can be expected that there exist different data sources:

- Information about the data is collected from different domains
  → different features are recorded
  - medical diagnosis (CT, hemogram,...)
  - multimedia (audio, video, text)
  - web pages (text of this page, anchor texts)
  - molecules (amino acid sequence, secondary structure, 3D representation)



⇒ Multiple data sources provide us with **multiple given views on the data**



Given Views vs. Previous Paradigms

Multiple Sources vs. One Database

- Each object is described by multiple sources
- Each object might have multiple representations

⇒ Multiple views on each object are given in the data

Given Views vs. View Detection

- For each object the relevant views are already given
- Traditional clustering can be applied on each view

⇒ Multiple clusterings exist due to the given views

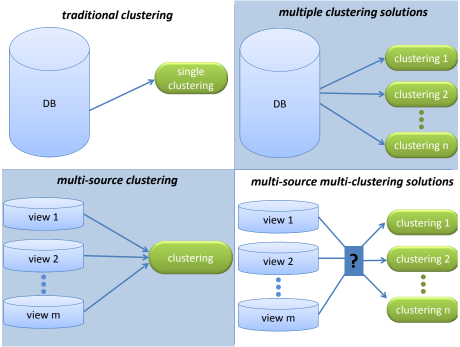
Consensus Clustering vs. Multiple Clusterings

- Clusterings are not alternatives but parts of a consensus solution

⇒ Focus on techniques to establish a consensus solutions

Consensus Clustering on Multiple Views

- Generate one consistent clustering from multiple views of the data



$\Rightarrow$ How to combine results from different views

1. By merging clusterings to one consensus solution
2. Without merging the given sources

Challenge: Heterogeneous Data

- Information about objects is available from different sources
- Data sources are often heterogeneous (*multi-represented data*)

$\Rightarrow$ Traditional methods do not provide a solution...

Reduction to Traditional Clustering

Clustering multi-represented data by traditional clustering methods requires:

- Restriction of the analysis to a single representation / source
  - $\rightarrow$ Loss of information
- Construction of a feature space comprising all representations
  - $\rightarrow$ Demands a new combined distance function
  - $\rightarrow$ Specialized data access structures (e.g. index structures) for each representation would not be applicable anymore

General Idea of Multi-Source Clustering

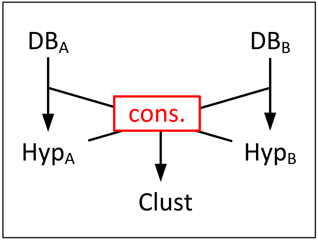
Aim: determine a clustering that is consistent with all sources

$\Rightarrow$ Idea: train different hypotheses from the different sources, which bootstrap by providing each others with parameters

$\Rightarrow$ Consensus between all hypotheses and all sources is achieved

General Assumptions:

- Each view in itself is sufficient for a single clustering solution
- All views are compatible
- All views are conditional independent



Principle of Multi-Source Learning

Co-Training (Blum & Mitchell, 1998)

Bootstrapping method, which trains two hypotheses on distinct views

- originally developed for classification
- the usage of unlabeled together with labeled data has often shown to substantially improve the accuracy of the training phase
- multi-source algorithms train two independent hypotheses, that bootstrap by providing each other with labels for the unlabeled data
- the training algorithms tend to maximize the agreement between the two independent hypotheses
- disagreement of two independent hypothesis is an upper bound on the error rate of one hypothesis

Overview of Methods in Multi-Source Paradigm

Adaption of Traditional Clustering

- co-EM: iterates interleaved **EM** over two given views (Bickel & Scheffer, 2004)
- multi-represented **DBSCAN** for sparse or unreliable sources (Kailing *et al.*, 2004a)

Further Approaches:

- Based on different **cluster definitions**:
  e.g. spectral clustering (de Sa, 2005; Zhou & Burges, 2007)
  or fuzzy clustering in parallel universes (Wiswedel *et al.*, 2010)
- Consensus of **distributed sources** or **distributed clusterings**
  e.g. (Januzaj *et al.*, 2004; Long *et al.*, 2008)
- Consensus of **subspace clusterings**
  e.g. (Fern & Brodley, 2003; Domeniconi & Al-Razgan, 2009)

co-EM Method (Bickel & Scheffer, 2004)

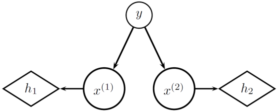
Assumption: The attributes of the data are given in two disjoint sets $V^{(1)}$, $V^{(2)}$. An object x is defined as $x := (x^{(1)}, x^{(2)})$, with $x^{(1)} \in V^{(1)}$ and $x^{(2)} \in V^{(2)}$.

- For each view $V^{(i)}$ we define a hypothesis space $H^{(i)}$
- the overall hypothesis will be combined of two consistent hypotheses $h_1 \in H^{(1)}$ and $h_2 \in H^{(2)}$.
- To restrict the set of consistent hypotheses h_1, h_2, both views have to be conditional independent:

Conditional Independence Assumption

Views $V^{(1)}$ and $V^{(2)}$ are conditional independent given the target value y, if $\forall x^{(1)} \in V^{(1)}, \forall x^{(2)} \in V^{(2)}$: $p(x^{(1)}, x^{(2)} | y) = p(x^{(1)} | y) * p(x^{(2)} | y)$.

- the only dependence between two objects from $V^{(1)}$ and $V^{(2)}$ is given by their target value.



co-EM Algorithmic Steps

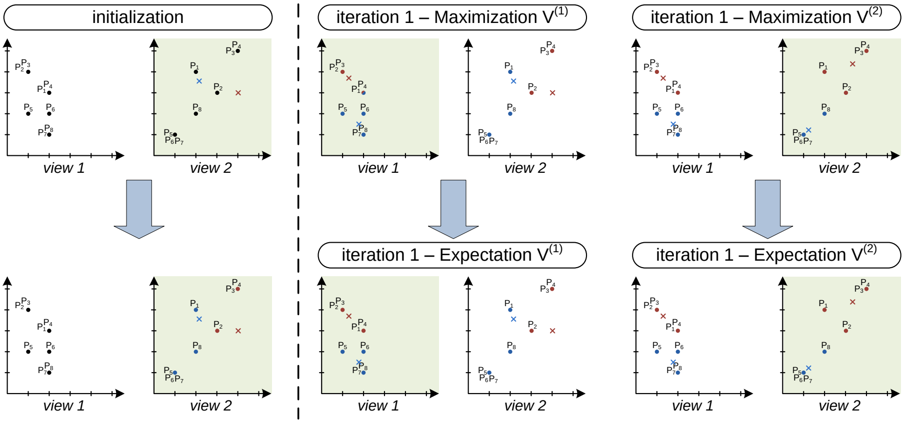
EM revisited:

- **Expectation:** calculate the expected posterior probabilities of the objects based on the current model estimation (assignment of points to clusters)
- **Maximization:** recompute the model parameters θ by maximizing the likelihood of the obtained cluster assignments

Now bootstrap this process by the two views:
For $v = 0, 1$

1. **Maximization:** maximize the likelihood of the data over the model parameters $\theta^{(v)}$ using the posterior probabilities according to view $V^{(\bar{v})}$
2. **Expectation:** compute the expectation of the posterior probabilities according to the new obtained model parameters

co-EM Example



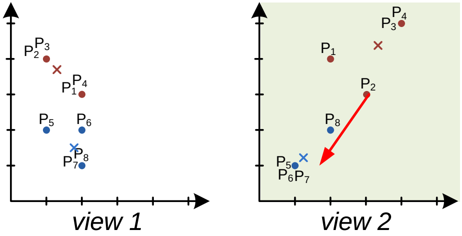
Discussion on co-EM Properties

- Clustering on a single view yields a higher likelihood
- However, initializing single-view with final parameters of multi-view yields even higher likelihood

$\Rightarrow$ Multi-view techniques enable higher clustering quality

Termination Criterion

- Iterative co-EM might not terminate
- Additional termination criterion required



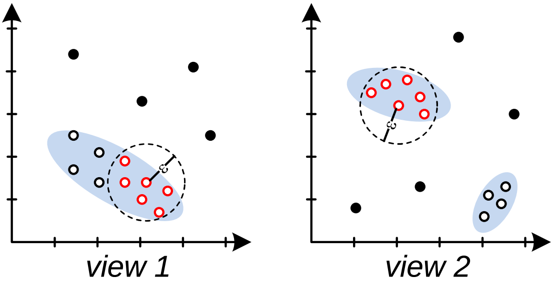
Multi-View DBSCAN (Kailing *et al.*, 2004a)

Idea: adapt the core object property proposed for DBSCAN

- Determine the local ε-neighborhood of each view independently

$$\mathcal{N}_{\varepsilon_i}^{V^{(i)}}(o) = \left\{ x \in DB \,\middle|\, dist_i(o^{(i)}, x^{(i)}) \leq \varepsilon_i \right\}$$

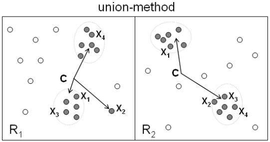
- Combine the results to a **global neighborhood**
  - Sparse spaces: **union method**
  - Unreliable data: **intersection method**



Union of Different Views

- especially useful for sparse data, where each single view provides several small clusters and a large amount of noise

- two objects are assigned to the same cluster if they are similar in at least one of the views



union core object

Let $\varepsilon_1, \ldots \varepsilon_m \in \mathbb{R}^+, k \in \mathbb{N}$. An object $o \in DB$ is formally defined as *union core object* as follows: $\text{CoreU}^k_{\varepsilon_1,\ldots\varepsilon_m}(o) \Leftrightarrow \left|\bigcup_{o^{(i)}\in o} \mathcal{N}^{V^{(i)}}_{\varepsilon_i}(o)\right| \geq k$

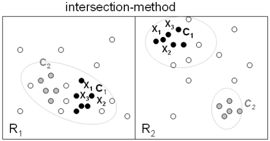
direct union-reachability

Let $\varepsilon_1, \ldots \varepsilon_m \in \mathbb{R}^+, k \in \mathbb{N}$. An object $p \in DB$ is *directly union-reachable* from $q \in DB$ if q is a union core object and p is an element of at least one local $\mathcal{N}^{V^{(i)}}_{\varepsilon_i}(q)$, formally:

$$\text{DirReachU}^k_{\varepsilon_1,\ldots\varepsilon_m}(q,p) \Leftrightarrow \text{CoreU}^k_{\varepsilon_1,\ldots\varepsilon_m}(q) \wedge \exists i \in \{1,\ldots,m\} : p^{(i)} \in \mathcal{N}^{V^{(i)}}_{\varepsilon_i}(q)$$

Intersection of Different Views

- well suited for data containing unrealiable views (providing questionable descriptions of the objects)
- two objects are assigned to the same cluster only if they are similar in all of the views
  $\rightarrow$ finds purer clusters



intersection core object

Let $\varepsilon_1, \ldots \varepsilon_m \in \mathbb{R}^+, k \in \mathbb{N}$. An object $o \in DB$ is formally defined as *intersection core object* as follows: $\text{CoreIS}^k_{\varepsilon_1,\ldots\varepsilon_m}(o) \Leftrightarrow \left|\bigcap_{i\in\{1,\ldots,m\}} \mathcal{N}^{V^{(i)}}_{\varepsilon_i}(o)\right| \geq k$

direct intersection-reachability

Let $\varepsilon_1, \ldots \varepsilon_m \in \mathbb{R}^+, k \in \mathbb{N}$. An object $p \in DB$ is *directly intersection-reachable* from $q \in DB$ if q is a intersection core object and p is an element of all local $\mathcal{N}^{V^{(i)}}_{\varepsilon_i}(q)$, formally:
$\text{DirReachIS}^k_{\varepsilon_1,\ldots\varepsilon_m}(q,p) \Leftrightarrow \text{CoreIS}^k_{\varepsilon_1,\ldots\varepsilon_m}(q) \wedge \forall i \in \{1,\ldots,m\} : p^{(i)} \in \mathcal{N}^{V^{(i)}}_{\varepsilon_i}(q)$

Consensus Clustering on Subspace Projections

Motivation

- One high dimensional data source (cf. subspace clustering paradigm)
- Extract lower dimensional projections (views)

$\Rightarrow$ In contrast to previous paradigms, stabilize **one clustering solution**

$\Rightarrow$ One consensus clustering not multiple alternative clusterings

General Idea (View Extraction + Consensus)

- Split one data source in multiple views (view extraction)
- Cluster each view, and thus, build multiple clusterings
- Use external consensus criterion as post-processing on multiple clusterings in different views

$\Rightarrow$ One consensus clustering over multiple views of a single data source

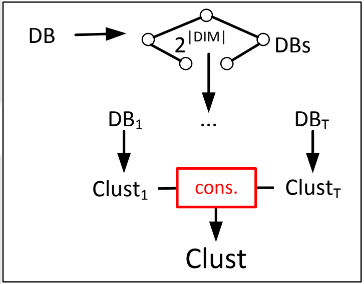
Given vs. Extracted Views

Given Sources

- Clustering on each given source
- Consensus over multiple sources

Extracted Views

- One high dimensional data source
- Virtual views by lower dimensional subspace projections



- Enable consensus mining on one data source:

⇒ Use subspace mining paradigm for space selection

⇒ Use common objective functions for consensus clustering

Consensus on Subspace Projections

Consensus Mining on One Data Source

- Create basis for consensus mining:
  - By *random projections* + EM clustering (Fern & Brodley, 2003)
  - By *soft feature selection* techniques (Domeniconi & Al-Razgan, 2009)
- Consensus objectives for subspace clusterings

Consensus objective from **ensemble clustering** (Strehl & Ghosh, 2002)

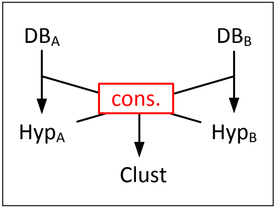
- Optimizes shared mutual information of clusterings:
  Resulting clustering shares most information with original clusterings

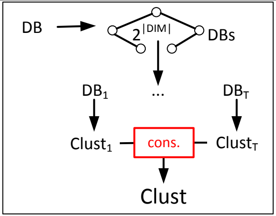
Instantiation in (Fern & Brodley, 2003)

- Compute consensus by
  similarity measure between partitions and *reclustering* of objects
- Probability of objects i and j in the same cluster under model θ:

$$P^{\theta}_{i,j} = \sum_{l=1}^{k} P(l|i,\theta) \cdot P(l|j,\theta)$$

Overview for this Paradigm





consensus basis:	sources are known	low dimensional projections
consensus transfer:	internal cluster model parameter	external objective function
consensus objective:	stable clusters	enable clustering in high dimensions
cluster model:	specific adaption	generalized consensus
$\Rightarrow$ consensus solution for multiple clusterings		

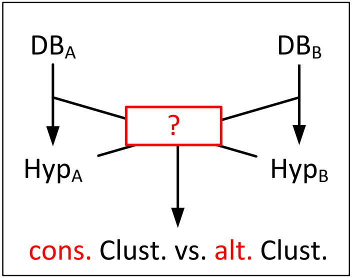
Open Challenges w.r.t. this Paradigm

Generalization to Multiple Clustering Solutions

- Incorporate given/detected views into consensus clustering
- Generalize post-processing steps to multiple clustering solutions

- Utilize consensus techniques in redundancy elimination
- Consensus clustering vs. different clustering solutions

⇒ Highlight alternatives by compressing common structures



Overview

Scope of the Tutorial

Focus of the tutorial

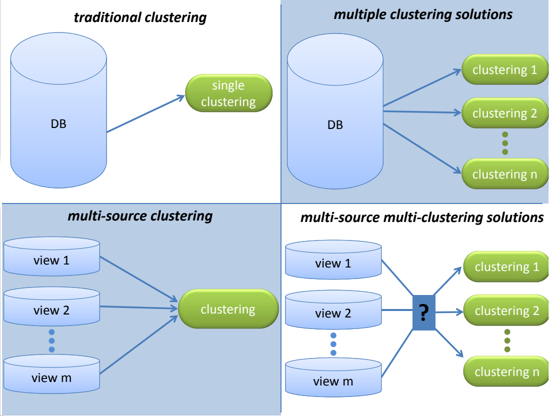
ONE database:
MULTIPLE CLUSTERINGS

+ extensions to
MULTIPLE SOURCES

Major objective

Overview of
- Challenges
- Taxonomy / notions

Comparison of paradigms:
- Underlying techniques
- Pros and Cons



Discussion of Approaches based on the Taxonomy I

Taxonomy for MULTIPLE CLUSTERING SOLUTIONS

From the perspective of the underlying data space:

- Detection of multiple clustering solutions...
  - in the Original Data Space
  - by Orthogonal Space Transformations
  - by Different Subspace Projections
  - in Multiple Given Views/Sources

Main focus on this categorization...

- Differences in **cluster definitions**
- Differences in **modeling the views** on the data
- Differences in **similarity between clusterings**
- Differences in modeling **alternatives to given knowledge**

Discussion of Approaches based on the Taxonomy II

	space	processing	given know.	# clusterings	subspace detec.	flexibility
(Caruana et al., 2006)	original			m >= 2		exchang. def.
(Bae & Bailey, 2006)	original	iterative	given clustering	m == 2		specialized
(Gondek & Hofmann, 2004)	original	iterative	given clustering	m == 2		specialized
(Jain et al., 2008)	original	simultaneous	no	m >= 2		specialized
(Hossain et al., 2010)	original	simultaneous	no	m == 2		specialized
(Dang & Bailey, 2010a)	original	simultaneous	no	m >= 2		specialized
(Davidson & Qi, 2008)	transformed	iterative	given clustering	m == 2	dissimilarity	exchang. def.
(Qi & Davidson, 2009)	transformed	iterative	given clustering	m == 2	dissimilarity	exchang. def.
(Cui et al., 2007)	transformed	iterative	given clustering	m >= 2	dissimilarity	exchang. def.
(Agrawal et al., 1998)...	subspaces		no	m >= 2	no dissimilarity	specialized
(Sequeira & Zaki, 2004)	subspaces		no	m >= 2	no dissimilarity	specialized
(Moise & Sander, 2008)	subspaces	simultaneous	no	m >= 2	no dissimilarity	specialized
(Müller et al., 2009b)	subspaces	simultaneous	no	m >= 2	no dissimilarity	specialized
(Günnemann et al., 2009)	subspaces	simultaneous	no	m >= 2	dissimilarity	specialized
(Günnemann et al., 2010)	subspaces	simultaneous	given clustering	m >= 2	dissimilarity	specialized
(Cheng et al., 1999)	subspaces		no	m >= 2	no dissimilarity	specialized
(Niu & Dy, 2010)	subspaces		no	m >= 2	dissimilarity	exchang. def.
(Bickel & Scheffer, 2004)	multi-source	simultaneous	no	m = 1	given views	specialized
(Kailing et al., 2004)	multi-source	simultaneous	no	m = 1	given views	specialized
(Fern & Brodley, 2003)	multi-source		no	m = 1	no dissimilarity	exchang. def.

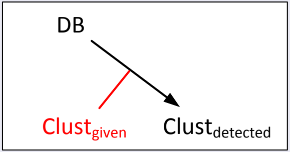
Let us discuss the secondary characteristics of our taxonomy...

Discussion of Approaches based on the Taxonomy III

From the perspective of the **given knowledge**:

- No clustering is given
- One or multiple clusterings are given

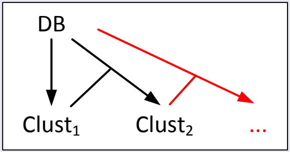
- If some knowledge is given
  it enables **alternative cluster detection**
- Users can steer algorithms to novel knowledge
- How is such prior knowledge provided?
- How to model the differences
  (to the given and the detected clusters)?
- How many alternatives clusterings are desired?



Discussion of Approaches based on the Taxonomy IV

From the perspective of **how many clusterings** are provided:

- $m = 1$ (traditional clustering) VS. $m = 2$ OR $m > 2$ (multiple clusterings)
- $m = T$ fixed by parameter OR open for optimization



- Multiple clusterings are enforced ($m \geq 2$)
- Each clustering should contribute!

$\Rightarrow$ Enforcing many clusterings leads to redundancy

- How set the number of desired clusterings (automatically / manually)?
- How to model redundancy of clusterings?
- How to ensure that the overall result is a high quality combination of clusterings?

Discussion of Approaches based on the Taxonomy V

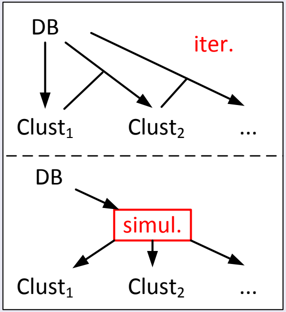
From the perspective of **cluster computation**:

- Iterative computation of further clustering solutions
- Simultaneous computation of multiple clustering solutions

- Iterative techniques are useful in generalized approaches
- However, iterations select one optimal clustering and might miss the global optimum for the resulting set of clusterings

$\Rightarrow$ Focus on quality of all clusterings

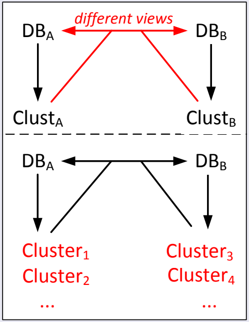
- How to specify such an objective function?
- How to efficiently compute global optimum without computing all possible clusterings?
- How to find the optimal views on the data?



Discussion of Approaches based on the Taxonomy VI

From the perspective of **view / subspace detection**:

- One view vs. different views
- Awareness of common views for several clusters



- Multiple views might lead to better distinction between multiple different clusterings
- Transformations based on given knowledge or search in all possible subspaces?

- Definition of dissimilarity between views?
- Efficient computation of relevant views?
- Groups of clusters in common views?
- Selection of views independent of cluster models?

Discussion of Approaches based on the Taxonomy VII

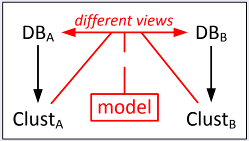
From the perspective of **flexibility**:

- View detection and multiple clusterings are bound to the cluster definition
- The underlying cluster definition can be exchanged (flexible model)

- Specialized algorithms are hard to adapt (e.g. to application demands)

$\Rightarrow$ Tight bounds/integrations might be decoupled

- How to detect orthogonal views only based on an abstract representation of clusterings?
- How to define dissimilarity between views and clusterings?
- What are the common objectives (independent of the cluster definition)?



Correlations between taxonomic views

<table>
<tr><th></th><th></th><th>search space taxonomy</th><th>processing</th><th>knowledge</th><th>flexibility</th></tr>
<tr><td rowspan="4">Sec. 2</td><td>algorithm1</td><td rowspan="4">original space</td><td></td><td></td><td>exch. def.</td></tr>
<tr><td>alg2</td><td>iterative</td><td>given k.</td><td rowspan="3">specialized</td></tr>
<tr><td>alg3</td><td rowspan="2">simultan.</td><td rowspan="2">no given k.</td></tr>
<tr><td>alg4</td></tr>
<tr><td rowspan="2">Sec. 3</td><td>alg5</td><td rowspan="2">orthogonal transformations</td><td rowspan="2">iterative</td><td rowspan="2">given k.</td><td rowspan="2">exch. def.</td></tr>
<tr><td>alg6</td></tr>
<tr><td rowspan="4">Sec. 4</td><td>alg7</td><td rowspan="4">subspace projections</td><td></td><td rowspan="2">no given k.</td><td rowspan="3">specialized</td></tr>
<tr><td>alg8</td><td rowspan="2">simultan.</td></tr>
<tr><td>alg9</td><td>given k.</td></tr>
<tr><td>alg10</td><td></td><td rowspan="4">no given k.</td><td>exch. def.</td></tr>
<tr><td rowspan="3">Sec. 5</td><td>alg11</td><td rowspan="3">multiple views/sources</td><td rowspan="2">simultan.</td><td rowspan="2">specialized</td></tr>
<tr><td>alg12</td></tr>
<tr><td>alg13</td><td></td><td>exch. def.</td></tr>
</table>

⇒ Might reveal some open research questions... (?)

Open Research Questions I

- Most approaches are **specialized to a cluster model**
- Even more important: Most approaches focus on **non-naive solutions only in one part of the taxonomy**!

Generalization as major topic...

- Exchangeable cluster model, decoupling view and cluster detection
- Abstraction from how knowledge is given
- Enhanced view selection (aware of differences between views)
- Simultaneous computation with given knowledge

Open challenges to the community:

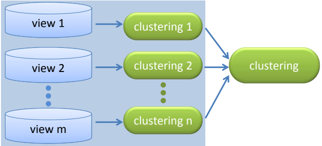
- Common **benchmark data** and **evaluation framework**
- Common **quality assessment** (for multiple clusterings)

Open Research Questions II

How **multiple clustering solutions** can **contribute to enhanced mining**?

First solutions...

- Given views/sources for clustering
- Stabilizing results (one final clustering)

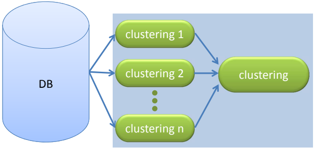


Further ideas

- Observed in ensemble clustering

⇒ Summarizing multiple clustering solutions

⇒ Converging multiple clustering solutions



Multiple clustering solutions is still an open research field...

Discovering Multiple Clustering Solutions: Grouping Objects in Different Views of the Data

contact information:
emmanuel.mueller@kit.edu
{guennemann, faerber, seidl }@cs.rwth-aachen.de

or during the conference:

Karlsruhe Institute of Technology

RWTH AACHEN UNIVERSITY

Discovering Multiple Clustering Solutions: Grouping Objects in Different Views of the Data

contact information:
emmanuel.mueller@kit.edu
{guennemann, faerber, seidl }@cs.rwth-aachen.de

or during the conference:




Thanks for attending the tutorial! Any questions?

download slides: `http://dme.rwth-aachen.de/DMCS`

References I

Aggarwal, C., & Yu, P. 2000.
Finding generalized projected clusters in high dimensional spaces.
In: SIGMOD.

Aggarwal, C., Wolf, J., Yu, P., Procopiuc, C., & Park, J. 1999.
Fast algorithms for projected clustering.
In: SIGMOD.

Agrawal, R., & Srikant, R. 1994.
Fast Algorithms for mining Association Rules.
In: VLDB.

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. 1998.
Automatic subspace clustering of high dimensional data for data mining applications.
In: SIGMOD.

References II

Assent, I., Krieger, R., Müller, E., & Seidl, T. 2007.
DUSC: Dimensionality Unbiased Subspace Clustering.
In: ICDM.

Assent, I., Krieger, R., Müller, E., & Seidl, T. 2008.
INSCY: Indexing Subspace Clusters with In-Process-Removal of Redundancy.
In: ICDM.

Bae, Eric, & Bailey, James. 2006.
COALA: A Novel Approach for the Extraction of an Alternate Clustering of High Quality and High Dissimilarity.
In: ICDM.

Bae, Eric, Bailey, James, & Dong, Guozhu. 2010.
A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings.
Data Min. Knowl. Discov., **21**(3).

References III

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. 1999.
When is nearest neighbors meaningful.
In: IDBT.

Bickel, Steffen, & Scheffer, Tobias. 2004.
Multi-View Clustering.
In: ICDM.

Blum, A., & Mitchell, T. 1998.
Combining labeled and unlabeled data with co-training.
In: COLT.

Böhm, C., Kailing, K., Kriegel, H.-P., & Kröger, P. 2004a.
Density Connected Clustering with Local Subspace Preferences.
In: ICDM.

Böhm, Christian, Kailing, Karin, Kröger, Peer, & Zimek, Arthur. 2004b.
Computing Clusters of Correlation Connected objects.
In: SIGMOD.

References IV

Caruana, Rich, Elhawary, Mohamed Farid, Nguyen, Nam, & Smith, Casey. 2006.
Meta Clustering.
In: ICDM.

Chechik, Gal, & Tishby, Naftali. 2002.
Extracting Relevant Structures with Side Information.
In: NIPS.

Cheng, C.-H., Fu, A. W., & Zhang, Y. 1999.
Entropy-based subspace clustering for mining numerical data.
In: SIGKDD.

Cordeiro, R., Traina, A., Faloutsos, C., & Traina, C. 2010.
Finding Clusters in Subspaces of Very Large Multi-dimensional Datasets.
In: ICDE.

References V

Cui, Ying, Fern, Xiaoli Z., & Dy, Jennifer G. 2007.
Non-redundant Multi-view Clustering via Orthogonalization.
In: ICDM.

Cui, Ying, Fern, Xiaoli Z., & Dy, Jennifer G. 2010.
Learning multiple nonredundant clusterings.
TKDD, **4**(3).

Dang, Xuan Hong, & Bailey, James. 2010a.
Generation of Alternative Clusterings Using the CAMI Approach.
In: SDM.

Dang, Xuan Hong, & Bailey, James. 2010b.
A hierarchical information theoretic technique for the discovery of non linear alternative clusterings.
In: SIGKDD.

References VI

Davidson, Ian, & Qi, Zijie. 2008.
Finding Alternative Clusterings Using Constraints.
In: ICDM.

de Sa, Virginia R. 2005.
Spectral clustering with two views.
In: ICML Workshop on Learning with Multiple Views.

Domeniconi, Carlotta, & Al-Razgan, Muna. 2009.
Weighted cluster ensembles: Methods and analysis.
TKDD, **2**(4).

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. 1996.
A density-based algorithm for discovering clusters in large spatial databases.
In: SIGKDD.

References VII

Fern, Xiaoli Zhang, & Brodley, Carla E. 2003.
Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach.
In: ICML.

Gondek, D., & Hofmann, T. 2003.
Conditional information bottleneck clustering.
In: ICDM, Workshop on Clustering Large Data Sets.

Gondek, David, & Hofmann, Thomas. 2004.
Non-Redundant Data Clustering.
In: ICDM.

Gondek, David, & Hofmann, Thomas. 2005.
Non-redundant clustering with conditional ensembles.
In: SIGKDD.

References VIII

Gondek, David, Vaithyanathan, Shivakumar, & Garg, Ashutosh. 2005.
Clustering with Model-level Constraints.
In: SDM.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. 2005.
Measuring statistical dependence with hilbertschmidt norms.
In: Algorithmic Learning Theory.

Günnemann, S., Müller, E., Färber, I., & Seidl, T. 2009.
Detection of Orthogonal Concepts in Subspaces of High Dimensional Data.
In: CIKM.

Günnemann, S., Färber, I., Müller, E., & Seidl, T. 2010.
ASCLU: Alternative Subspace Clustering.
In: MultiClust Workshop at SIGKDD.

References IX

Hossain, M. Shahriar, Tadepalli, Satish, Watson, Layne T., Davidson, Ian, Helm, Richard F., & Ramakrishnan, Naren. 2010.
Unifying dependent clustering and disparate clustering for non-homogeneous data.
In: SIGKDD.

Jain, Prateek, Meka, Raghu, & Dhillon, Inderjit S. 2008.
Simultaneous Unsupervised Learning of Disparate Clusterings.
In: SDM.

Januzaj, Eshref, Kriegel, Hans-Peter, & Pfeifle, Martin. 2004.
Scalable Density-Based Distributed Clustering.
In: PKDD.

Kailing, K., Kriegel, H.-P., Kröger, P., & Wanka, S. 2003.
Ranking interesting subspaces for clustering high dimensional data.
In: PKDD.

References X

Kailing, K., Kriegel, H.-P., Pryakhin, A., & Schubert, M. 2004a.
Clustering Multi-Represented Objects with Noise.
In: PAKDD.

Kailing, K., Kriegel, H.-P., & Kröger, P. 2004b.
Density-Connected Subspace Clustering for High-Dimensional Data.
In: SDM.

Kriegel, Hans-Peter, Kröger, Peer, Renz, Matthias, & Wurst, Sebastian. 2005.
A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data.
In: ICDM.

Kriegel, Hans-Peter, Kröger, Peer, & Zimek, Arthur. 2009.
Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering.
TKDD, **3**(1).

References XI

Long, Bo, Yu, Philip S., & Zhang, Zhongfei (Mark). 2008.
A General Model for Multiple View Unsupervised Learning.
In: SDM.

Moise, Gabriela, & Sander, Jörg. 2008.
Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering.
In: SIGKDD.

Moise, Gabriela, Sander, Joerg, & Ester, Martin. 2006.
P3C: A Robust Projected Clustering Algorithm.
In: ICDM.

Müller, E., Assent, I., Krieger, R., Günnemann, S., & Seidl, T. 2009a.
DensEst: Density Estimation for Data Mining in High Dimensional Spaces.
In: SDM.

References XII

Müller, E., Günnemann, S., Assent, I., & Seidl, T. 2009b.
Evaluating Clustering in Subspace Projections of High Dimensional Data.
In: VLDB.

Müller, E., Assent, I., Günnemann, S., Krieger, R., & Seidl, T. 2009c.
Relevant Subspace Clustering: Mining the Most Interesting Non-Redundant Concepts in High Dimensional Data.
In: ICDM.

Nagesh, H., Goil, S., & Choudhary, A. 2001.
Adaptive grids for clustering massive data sets.
In: SDM.

Ng, A., Jordan, M., & Weiss, Y. 2001.
On spectral clustering: Analysis and an algorithm.
Advances in Neural Information Processing Systems, **14**.

References XIII

Niu, Donglin, & Dy, Jennifer G. 2010.
Multiple Non-Redundant Spectral Clustering Views.
In: ICML.

Parsons, Lance, Haque, Ehtesham, & Liu, Huan. 2004.
Subspace clustering for high dimensional data: a review.
SIGKDD Explorations, **6**(1).

Procopiuc, C. M., Jones, M., Agarwal, P. K., & Murali, T. M. 2002.
A Monte Carlo algorithm for fast projective clustering.
In: SIGMOD.

Qi, Zijie, & Davidson, Ian. 2009.
A principled and flexible framework for finding alternative clusterings.
In: SIGKDD.

References XIV

Sequeira, K., & Zaki, M. 2004.
SCHISM: A New Approach for Interesting Subspace Mining.
In: ICDM.

Strehl, Alexander, & Ghosh, Joydeep. 2002.
Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions.
Journal of Machine Learning Research, **3**, 583–617.

Vinh, Nguyen Xuan, & Epps, Julien. 2010.
minCEntropy: a Novel Information Theoretic Approach for the Generation of Alternative Clusterings.
In: ICDM.

Wiswedel, Bernd, Höppner, Frank, & Berthold, Michael R. 2010.
Learning in parallel universes.
Data Min. Knowl. Discov., **21**(1).

References XV

Yiu, M. L., & Mamoulis, N. 2003.
Frequent-pattern based iterative projected clustering.
In: ICDM.

Zhou, D., & Burges, C. J. C. 2007.
Spectral clustering and transductive learning with multiple views.
In: ICML.