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

Emmanuel Müller○, Stephan Günemann●, Ines Färber□, Thomas Seidl□

○ Karlsruhe Institute of Technology, Germany
● Carnegie Mellon University, USA
□ RWTH Aachen University, Germany

 Tutorial at ICML 2013

download slides: http://dme.rwth-aachen.de/DMCS
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
Traditional Cluster Detection

Abstract cluster definition

“Group similar objects in one group, separate 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

⇒ 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.

⇒ Overall, enhanced extraction of knowledge.

⇒ 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: Customer Segmentation

Clustering of 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 (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: 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
  ⇒ 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 (4)
⇒ 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**

⇒ 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
  → 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
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
Lost Views due to Evolving Databases

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

- Extending the stored information may lead to huge data dumps
- Relationships 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 views on the same data.
**Example Customer Analysis – Abstraction**

<table>
<thead>
<tr>
<th>object ID</th>
<th>age</th>
<th>income</th>
<th>blood pres.</th>
<th>sport activ.</th>
<th>profession</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
<tr>
<td>2</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
<tr>
<td>3</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
<tr>
<td>4</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
<tr>
<td>5</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
<tr>
<td>6</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
<tr>
<td>7</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
<tr>
<td>8</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
<tr>
<td>9</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
<td>XYZ</td>
</tr>
</tbody>
</table>

- Consider each customer as a row in a database table
- Here a selection of possible attributes (example)
**Example Customer Analysis – Clustering**

<table>
<thead>
<tr>
<th>object ID</th>
<th>age</th>
<th>income</th>
<th>blood pres.</th>
<th>sport activ.</th>
<th>profession</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>59.000</td>
<td>130</td>
<td>comp. game</td>
<td>CS</td>
</tr>
<tr>
<td>4</td>
<td>51</td>
<td>61.000</td>
<td>129</td>
<td>comp. game</td>
<td>CS</td>
</tr>
<tr>
<td>5</td>
<td>49</td>
<td>58.500</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>6</td>
<td>47</td>
<td>62.000</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>7</td>
<td>52</td>
<td>60.000</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 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

- 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 – Alternative Clusterings

Already known before... (given knowledge)

Major task: detect multiple alternatives

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

Assume a given knowledge about one clustering

How to find further solutions (alternative clusterings) that describe additional knowledge?

⇒ Novel challenges in defining differences between clusterings
Challenge: Comparison of Clusterings

Requirements for Multiple Clustering Solutions:

- Identify only one solution is too restrictive
  → Multiple solutions are desired
- However, one searches for different / alternative / orthogonal clusterings
  → Novel definitions of difference between clusterings
- Search for multiple sets of clusters (multiple clusterings), in contrast to one optimal set of clusters
  → 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 – 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 …

⇒ 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

⇒ Task: Enforce clustering to detect multiple solutions
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...

⇒ Taxonomy of paradigms and methods
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>
<thead>
<tr>
<th>Algorithm</th>
<th>Search Space Taxonomy</th>
<th>Processing</th>
<th>Knowledge</th>
<th>Flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>alg1</td>
<td>original space</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg2</td>
<td>iterative</td>
<td>given k.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg3</td>
<td>simultan.</td>
<td>no given k.</td>
<td>specialized</td>
<td></td>
</tr>
<tr>
<td>alg4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg5</td>
<td>orthogonal transformations</td>
<td>iterative</td>
<td>given k.</td>
<td>exch. def.</td>
</tr>
<tr>
<td>alg6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg8</td>
<td>subspace projections</td>
<td>simultan.</td>
<td>given k.</td>
<td>specialized</td>
</tr>
<tr>
<td>alg9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg11</td>
<td>multiple views/sources</td>
<td>simultan.</td>
<td>no given k.</td>
<td>specialized</td>
</tr>
<tr>
<td>alg12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

**MULTIPLE CLUSTERING SOLUTIONS**

- alternative clusters
- disparate clusters
- different clusters
- subspace search
- multi-source clustering
- multi-view clustering
- subspace clustering
- orthogonal clustering
# Common Notions vs. Diversity of Terms II

<table>
<thead>
<tr>
<th>ALTERNATIVE CLUSTERING</th>
<th>with a given knowledge used to find alternative clusterings</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORTHOGONAL CLUSTERING</td>
<td>transforming the search space based on previous results</td>
</tr>
<tr>
<td>SUBSPACE CLUSTERING</td>
<td>using different subspace projections to find clusters in lower dimensional projections</td>
</tr>
</tbody>
</table>

**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

Müller, Günnemann, Färber, Seidl

*Discovering Multiple Clustering 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 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 ⊆ Domain**
  - set of objects (usually $\text{Domain} = \mathbb{R}^d$)
- **Clust\textsubscript{i}**
  - clustering (set of clusters $C_j$) of the objects $DB$
- **Clusterings**
  - theoretical set of all clusterings $\text{Clust}_i$
- **$Q : \text{Clusterings} \rightarrow \mathbb{R}$**
  - function to measure the quality of a clustering
- **$\text{Diss} : \text{Clusterings} \times \text{Clusterings} \rightarrow \mathbb{R}$**
  - function to measure the dissimilarity between clusterings

**Aim:** Detect clusterings $\text{Clust}_1, \ldots, \text{Clust}_m$ such that

- $Q(\text{Clust}_i)$ is high $\forall i \in \{1, \ldots, m\}$
- $\text{Diss}(\text{Clust}_i, \text{Clust}_j)$ is high $\forall i, j \in \{1, \ldots, m\}, i \neq j$
Comparison to Traditional Clustering

Multiple Clusterings

Detect clusterings $\text{Clust}_1, \ldots, \text{Clust}_m$ such that

- $Q(\text{Clust}_i)$ is high $\forall i \in \{1, \ldots, m\}$
- $\text{Diss}(\text{Clust}_i, \text{Clust}_j)$ 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

General aim of Alternative Clustering
- given clustering $Clust_1$ and functions $Q$, $Diss$
- find clustering $Clust_2$ such that $Q(\text{Clust}_2) \& Diss(\text{Clust}_1, \text{Clust}_2)$ are high
COALA (Bae & Bailey, 2006)

**General idea of COALA**

- avoid similar grouping of objects by using **instance level constraints**
- add cannot-link constraint $cannot(o, p)$ if $\{o, p\} \subseteq C \in Clust_1$
- 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

---

previous grouping: $C_1=\{\bigcirc\circ\bigcirc\bigcirc\bigcirc\}$, $C_2=\{\square \bigcirc\square\square\}$
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 \( d_{\text{qual}} \)
- dissimilarity merge
  - determine \((P_a, P_b) \in \text{Dissimilar}\) with smallest average link distance \( d_{\text{diss}} \)
  - \((P_i, P_j) \in \text{Dissimilar} \iff \) constraints between sets are fulfilled
    \[ \iff \neg \exists o \in P_i, p \in P_j : \text{cannot}(o, p) \]
  - if \( d_{\text{qual}} < w \cdot d_{\text{diss}} \) 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 $(P_a, P_b) \in Dissimilar$ with smallest average link distance $d_{diss}$
  - $(P_i, P_j) \in Dissimilar \iff$ constraints between sets are fulfilled
  - $\iff \neg \exists o \in P_i, p \in P_j : cannot(o, p)$
- 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 $(P_a, P_b) \in \text{Dissimilar}$ with smallest average link distance $d_{diss}$
  - $(P_i, P_j) \in \text{Dissimilar} \iff$ constraints between sets are fulfilled
  - $\iff \neg \exists o \in P_i, p \in P_j : \text{cannot}(o, p)$
- 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 $(P_a, P_b) \in \text{Dissimilar}$ with smallest average link distance $d_{diss}$
  - $(P_i, P_j) \in \text{Dissimilar} \iff$ constraints between sets are fulfilled
    - $\iff \neg \exists o \in P_i, p \in P_j: \text{cannot}(o, p)$
- 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 $(P_a, P_b) \in \text{Dissimilar}$ with smallest average link distance $d_{diss}$
  - $(P_i, P_j) \in \text{Dissimilar} \iff$ constraints between sets are fulfilled
    \[\iff \neg \exists o \in P_i, p \in P_j : \text{cannot}(o, p)\]
- 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

Mathematical formulation:

- For large $w$: $d_{qual} < w \cdot d_{diss}$
  - Prefer quality, merge objects
  - Large circles indicate high quality
  - Small squares indicate low dissimilarity

- For small $w$: $d_{qual} \not< w \cdot d_{diss}$
  - Prefer dissimilarity, merge objects
  - Large circles indicate high dissimilarity
  - Small squares indicate low quality
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
  - (Bae et al., 2010): based upon comparison measure between clusterings, alternative should realize different density profile/histogram
Information Bottleneck Approaches

Information bottleneck principle

- information theoretic clustering approach
- random variables: \( X \) (objects) and \( Y \) (their features/attribute values)
- extract a *compact representation* of \( X \), i.e. the (probabilistic) clustering \( C \), with *minimal loss of mutual information* to the relevance variable \( Y \)
- variational principle: 
  \[
  \min_{p(c|x)} \left[ I(X, C) - \beta I(Y, C) \right]
  \]
- trade-off between
  - complexity (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 \( H(Y) = -\int p(y) \log p(y) dy \)
  - and conditional entropy \( H(Y|C) = -\iint p(y, c) \log p(y|c) dy dc \)
  - intuitively: how much is the uncertainty about \( Y \) decreased by knowing \( C \)
IB for Unsupervised Clustering

here: \( X = \) object ids, \( Y = \) features, \( C = \) clustering

Solution of IB

- since \( C \) is a compressed representation of \( X \), its distribution should be determined given \( X \) alone \( \Rightarrow p(c|x, y) = p(c|x) \)
- equivalently: \( p(y|x, c) = p(y|x) \)
  feature vector of given object does not depend on how we cluster
  \( \Rightarrow p(y|c) = \frac{1}{p(c)} \sum_x p(y|x)p(x)p(c|x) \)
  \( \Rightarrow p(c) = \sum_x p(x)p(c|x) \)
- \( p(c|x) \) is stationary point of \( I(X, C) - \beta I(Y, C) \) iff
  \[
p(c|x) = \frac{p(c)}{Z(x, \beta)} \exp(-\beta \cdot D_{KL}[p(y|x)\|p(y|c)])
  \]
IB with Given Knowledge

**Enrich traditional IB by given knowledge/clustering**

- Assume clustering $D$ is already given, $X$ objects, $Y$ features
- (Chechik & Tishby, 2002): $\min_{p(c|x)}[I(X, C) - \beta I(Y, C) + \gamma I(D, C)]$
- (Gondek & Hofmann, 2003): $\min_{p(c|x)}[I(X, C) - \beta I(Y, C|D)]$
- (Gondek & Hofmann, 2004): $\max_{p(c|x)}[I(Y, C|D)]$ such that $I(X, C) \leq c$ and $I(Y, C) \geq d$

- $I(X, C) \approx$ complexity, $I(Y, C) \approx$ 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**

- previous methods: only one alternative is extracted
- given $Clust_1 \rightarrow$ extract $Clust_2$
- thus, two clusterings determined
- however, multiple ($\geq 2$) 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 $Clust_3$
  - ...
Possible Solution

'Simple' Solution

- assume multiple clusterings $\mathcal{D} = \{D_1, \ldots, D_m\}$ are given
- use sum of dissimilarity in objective function
  - e.g. minimize $I(X, C) - \beta I(Y, C) + \gamma \sum_{D \in \mathcal{D}} I(D, C)$
  - where $C$ is the new clustering, $X$ are the objects, $Y$ are the features
- iteratively enlarge the set $\mathcal{D}$
- approaches from this category
  - (Gondek et al., 2005): likelihood maximization with constraints, handles only binary data
  - (Vinh & Epps, 2010): based on conditional entropy and kernel density estimation
  - (Dang & Bailey, 2013a): uses mixture models and mutual information between distributions
Drawbacks of Alternative Clustering Approaches

Drawback 2: Iterative processing

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

Clusterings

DB

$\text{clustering} + \text{dissimilarity}$

Clust$_1$

Clust$_2$

Other approach: Detect all clusterings *simultaneously*
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 $\text{Clust}_1, \ldots, \text{Clust}_m$
- make use of a combined objective function
- informally: maximize $\sum_i Q(\text{Clust}_i) + \sum_{i \neq j} \text{Diss}(\text{Clust}_i, \text{Clust}_j)$
Decorrelated k-Means (Jain et al., 2008)

Decorrelated k-Means: Notions

- $k$ clusters of $\text{Clust}_1$ 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 $\text{Clust}_2$
  - 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 $G(r_1, \ldots, r_k, s_1, \ldots, s_l) =$

$$
\sum_i \sum_{x \in C_i} \|x - r_i\|^2 + \sum_j \sum_{x \in D_j} \|x - s_j\|^2 + \lambda \sum_{i,j} (\beta_j^T \cdot r_i)^2 + \lambda \sum_{i,j} (\alpha_i^T \cdot s_j)^2
$$

compactness of both clusterings

$$\text{difference/orthogonality of representatives}$$

intuition of orthogonality: cluster labels generated by nearest-neighbor assignments are independent
**Decorrelated k-Means: Discussion**

**Discussion**
- Enables parametrization of desired number of clusterings
- 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)
  - (Kontonasios & Bie, 2013): finds clusters which are unexpected regarding prior beliefs, based on maximum entropy model, uses iterative processing
A Generative Model Based Approach

Idea of CAMI (Dang & Bailey, 2010a)

- generative model based approach
- each clustering $\text{Clust}_i$ is a Gaussian mixture model (parameter $\Theta_i$)
  
  $p(x|\Theta_i) = \sum_{j=1}^{k} \lambda_j^i \mathcal{N}(x, \mu_j^i, \Sigma_j^i) = \sum_{j=1}^{k} p(x|\theta_j^i)$

- quality of clusterings is measured by log-likelihood
  
  $L(\Theta_i, DB) = \sum_{x \in DB} \log p(x|\Theta_i)$

- (dis-)similarity by mutual information (KL divergence) between mixtures
  
  $I(\text{Clust}_1, \text{Clust}_2) = \sum_{j,j'} I(p(x|\theta_j^1), p(x|\theta_j^2))$

- combined objective function
  
  maximize $L(\Theta_1, DB) + L(\Theta_2, DB) - \mu I(\Theta_1, \Theta_2)$

- 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)
Maximum Entropy Modeling

Idea of (Kontonasios & Bie, 2013)

- Based on information theoretic model (De Bie, 2011) developed for pattern mining (here: pattern = cluster)
- Clusters are deemed more interesting if their probability is small under the prior beliefs, and thus, unexpected by the user
- Probability is derived based on maximum entropy model of prior beliefs
- Optimizing this set of alternative clusters is an NP-hard problem

→ Proposes a greedy heuristic with incremental processing, which approximates the simultaneous computation

Discussion

- Maximum entropy distribution potentially difficult to derive when using complex prior beliefs
- Potentially endless output of alternative clusters (stopping criterion?)
## Preliminary Conclusion for this Paradigm

<table>
<thead>
<tr>
<th>independent computation</th>
<th>focused on dissimilarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>iterative computation</td>
<td>simultaneous computation</td>
</tr>
<tr>
<td>based on previous knowledge</td>
<td>no knowledge required</td>
</tr>
<tr>
<td>new methods: ≥ 2 clusterings</td>
<td>often ≥ 2 clusterings possible</td>
</tr>
<tr>
<td>arbitrary clustering definition</td>
<td>specialized clustering definitions</td>
</tr>
<tr>
<td>methods are designed to detect multiple clusterings in the same data space</td>
<td></td>
</tr>
</tbody>
</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

![Diagram]

- grouping according to color
  - very low quality in original space
  - detection very unlikely
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 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

![Graph showing transformation and resulting novel structure and alternative grouping.](image)
General idea

Given a database \( DB \) and a clustering \( Clust_1 \), find a transformation \( T \) such that:
- The 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 the 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 $D$

\[
D = \begin{pmatrix}
1.5 & -1 \\
-1 & 1
\end{pmatrix}
\]
Determine the "alternative" transformation

- given learned transformation metric $D$
- SVD provides a decomposition: $D = H \cdot S \cdot A$
- informally: $D = \text{rotate} \cdot \text{stretch} \cdot \text{rotate}$
- $\rightarrow$ 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
Classification into taxonomy

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

Further approaches from this category:
- (Dang & Bailey, 2013b): regularized PCA, finds subspace highly independent to previous clustering, additional extension to non-linear shaped clusters
- (Qi & Davidson, 2009): constrained optimization problem, able to specify which parts of clustering to keep or to reject, trade-off between alternativeness and quality
**Regularized PCA Method**

**Idea of (Dang & Bailey, 2013b)**

- learn a subspace that
  - is independent (dissimilar) to the previous clustering
  - naturally preserves quality/characteristics of data

→ combine PCA with Hilbert Schmidt Independence Criterion (HSIC)

- advantages:
  - HSIC does not require to compute joint distribution to assess independence (in contrast to, e.g., mutual information)
  - using PCA with HSIC leads (for some instantiations) to an eigendecomposition problem

→ globally optimal subspace can be computed
given two random variables $A$ and $B$, the HSIC computes

- the squared Hilbert-Schmidt norm of the cross-covariance operator
- in the reproducing kernel Hilbert spaces $A$ and $B$
- based on the (universal) kernel functions $k$ and $l$

low HSIC $\Rightarrow$ variables are highly independent

empirical estimation of HSIC:

- given $n$ observations $(a_1, b_1), \ldots, (a_n, b_n)$

$$HSIC(A, B) = (n - 1)^{-2} \cdot \text{tr}(KHLH)$$

- where $K$ and $L$ are the Gram matrices, i.e. $K_{i,j} = k(a_i, a_j)$, $L_{i,j} = l(b_i, b_j)$

$$H = I - \frac{1}{n} 1_n 1_n^T$$ for centering of data

here: $A/K$ is the transformed data (in new subspace)

$B/L$ cluster membership of previous clustering
Instantiation of Gram Matrices

\[ HSIC(A, B) = (n - 1)^{-2} \cdot tr(KHLH) \]
\[ A/K \text{ transformed data, } B/L \text{ cluster membership} \]

Gram matrices selected in (Dang & Bailey, 2013b)

matrix \( K \)
- linear mapping, corresponding to linear projection of PCA
- \( \phi(x) = W^T x \)
- \( k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \Rightarrow K = X^T W W^T X \)

matrix \( L \)
- binary cluster membership indicators
- e.g. \( \theta(x) = (0, 1, 0, 0)^T \) if \( x \) belongs to cluster 2 out of 4
- \( l(x_i, x_j) = \langle \theta(x_i), \theta(x_j) \rangle \)
Overall Solution

Resulting Transformation

- Regularized PCA

\[ M = \arg \max_{W \in \mathbb{R}^{d \times q}} \left[ \text{var}(W^T X) - \text{HSIC}(W^T X, \text{Clust}_1) \right] \]

\[ = \arg \max_{W \in \mathbb{R}^{d \times q}} \left[ W^T XX^T W - \text{tr}(HX^T WW^T XHL) \right] \]

\[ = \arg \max_{W \in \mathbb{R}^{d \times q}} \left[ W^T (XX^T - XHLHX^T) W \right] \]

⇒ eigendecomposition problem

⇒ matrix \( M \) equals to \( q \) most important eigenvectors (largest eigenvalues)

- computation of \( \text{Clust}_2 \): perform clustering on \( M^T X \)
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
  \[ \rightarrow \text{add constraint } \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} x_i \notin C_j \| x_i - m_j \|_B \leq \beta \]
  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 $x_i \notin C_j$
    \[ \rightarrow \text{distance to 'old' mean } m_i \text{ should be high after transformation} \]
    \[ \rightarrow \text{novel clusters are expected} \]
Resulting Transformation

Solution

- optimal solution of constraint optimization problem

\[ M = \tilde{\Sigma}^{-1/2} \quad \text{with} \quad \tilde{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} (x_i - m_j)(x_i - m_j)^T \]

- 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 approaches: 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?
How to obtain novel structure after each iteration?

- make use of dimensionality reduction techniques
- first clustering determines main factors/principal 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 $\text{Clust}_i$ of $\text{DB}_i \rightarrow$ 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: $\text{DB}_i^A = \{A \cdot x \mid x \in \text{DB}_i\}$ yields the **same clustering**

---

Motivation Original Data Space Orthogonal Spaces Subspace Projections Multiple Sources Summary
Orthogonalization

Step 2: Determine the orthogonal subspace

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

![Diagram showing orthogonalization process with data points and orthogonal subspaces.](image)
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

focused on dissimilarity (implicitly by transformation)

iterative computation

(transformation is) based on previous knowledge

2 clusterings extracted

≥ 2 clusterings extracted
(by using dimensionality reduction)

independent of the used clustering algorithm

detect multiple clusterings based on space transformations
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

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
Clustering in Subspace Projections

- Irrelevant attributes might obfuscate the clustering in the fullspace (curse of dimensionality)

⇒ Only the data’s projection into subspaces reveals the clustering structure
- For different clusters **differing subspaces** might be important

⇒ Global dimensionality reduction is inappropriate

**Aim:** Detect a **group of objects and subset of attributes** per cluster
Clustering in Subspace Projections

- Clusters are observed in arbitrary attribute combinations (subspaces) using the original attributes (no transformations)
  ⇒ Cluster interpretation based on relevant attributes
- Detect multiple clusterings in different subspace projections as each object can be clustered differently in each projection
  ⇒ Enables multiple groupings of objects given different projections
Abstract Problem Definition

Abstract subspace clustering definition

- Specify cluster properties of object set $O$ grouped in subspace $S$
  \[ C = (O, S) \text{ with } O \subseteq DB, S \subseteq DIM \]
- Specify result set $M$ as 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 $(O, S)$
  \[ \Rightarrow \text{Output all (multiple) valid subspace clusters } M = ALL \]
- “Projected Clustering”: focus on definition of disjoint clusters in $M$
  \[ \Rightarrow \text{Unable to detect objects in multiple clusterings} \]
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

⇒ A single clustering solution only!
⇒ Might miss clusters and views due to this restriction

- ORCLUS: arbitrary oriented projected clusters (Aggarwal & Yu, 2000)
- DOC: monte carlo processing (Procopiuc et al., 2002)
- PreDeCon: locally preference weighted clusters (Böhm et al., 2004)
- MrCC: multi-resolution indexing technique (Cordeiro et al., 2010)
"Subspace Clustering" 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 \(S \subseteq DIM\)
  - Pruning the *exponential number of cluster candidates*
- Clusters as subsets of the database \(O \subseteq DB\)
  - 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 \( \tau \)
- 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...

- Interleaved processing (object set and dimension set)
- Detection of dense cells in any of the $2^{\text{DIM}}$ projections
- Puning of subspaces based on the monotonicity:
  - $O$ is dense in $S \Rightarrow \forall T \subseteq S : O$ is dense in $T$
  - Idea based on the apriori principle (Agrawal & Srikant, 1994)

$\Rightarrow$ CLIQUE detects overlapping clusters in different projections
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 $Pr[X_s \geq n_s] \leq e^{-2n_s t^2} \leq \tau$
  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 $\xi$ intervals

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

\[
\Rightarrow \frac{E[X_s]}{n} = \left(\frac{1}{\xi}\right)^s
\]
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 can be 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 \(S_2 \neq S_3\)
  - Produces **highly redundant output**
  - **Is not flexible** w.r.t. different cluster models
  - **Does not ensure dissimilarity** of subspace clusters
  - **Not able to compute alternatives** w.r.t. a given clustering
Open Challenges for Traditional Subspace Clustering

**Flexibility**
- Basic techniques such as PROCLUS (based on k-Means) or SUBCLU (based on DBSCAN) are dependent on the underlying cluster model.
- Each algorithm re-invents a new solution for *subspace selection*!

**Redundancy**
- Looking at the output of subspace clustering techniques, many of the detected clusters are redundant.
- Results are *overwhelming the users* and could be reduced!

**Dissimilarity**
- The traditional research area of subspace clustering focuses on detecting multiple views and multiple clustering solutions.
- However, *enforcing different clustering solutions* is not in its focus!
Subspace Search: Flexible Selection

- Assess the quality of a subspace as a whole
- Selection of interesting subspaces
- Comparison to subspace clustering
  - Subspace Clustering: individual subspace per cluster
  - Subspace Search: global set of subspaces
  ⇒ Provides multiple views on the data
  Decoupling subspace and cluster detection
  Flexibility in using different cluster models

- **ENCLUS**: entropy-based subspace search (Cheng *et al.*, 1999)
- **RIS**: density-based subspace search (Kailing *et al.*, 2003)
- **mSC**: enforces different spectral clustering views (Niu & Dy, 2010)
- **HiCS**: statistical selection of high contrast subspaces (Keller *et al.*, 2012)
- **CMI**: based on cumulative mutual information (Nguyen *et al.*, 2013)
Comparison of Subspace Mining Paradigms

<table>
<thead>
<tr>
<th></th>
<th>convex shapes</th>
<th>density-based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Full Space</strong></td>
<td>• K-Means (MacQueen, 1967)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DBSCAN (Ester et al., 1996)</td>
<td></td>
</tr>
<tr>
<td><strong>Coupled Subspace Mining</strong></td>
<td>• PROCLUS (Aggarwal et al., 1999)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• SUBCLU (Kailing et al., 2004b)</td>
<td></td>
</tr>
<tr>
<td><strong>Decoupled Subspace Mining</strong></td>
<td>• ENCLUS (Cheng et al., 1999)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• HiCS (Keller et al., 2012)</td>
<td></td>
</tr>
</tbody>
</table>
## Comparison of Subspace Mining Paradigms

Database → Clustering Method → Result

**Full Space**
- **convex shapes**
  - K-Means (MacQueen, 1967)
  - ...
- **density-based**
  - DBSCAN (Ester et al., 1996)
  - ...

**Coupled Subspace Mining**
- **convex shapes**
  - PROCLUS (Aggarwal et al., 1999)
  - ...
- **density-based**
  - SUBCLU (Kailing et al., 2004b)
  - ...

**Decoupled Subspace Mining**
- **convex shapes**
  - ...
- **density-based**
  - ENCLUS (Cheng et al., 1999)
  - HiCS (Keller et al., 2012)
Comparison of Subspace Mining Paradigms

- **Database** → **Subspace Search + Clustering Method** → **Result**

**Convex Shapes**
- **Full Space**
  - K-Means (MacQueen, 1967)
  - DBSCAN (Ester et al., 1996)
- **Coupled Subspace Mining**
  - PROCLUS (Aggarwal et al., 1999)
- **Decoupled Subspace Mining**
  - . . .

**Density-Based**
- **Full Space**
  - DBSCAN (Ester et al., 1996)
  - SUBCLU (Kailing et al., 2004b)
- **Coupled Subspace Mining**
  - PROCLUS (Aggarwal et al., 1999)
- **Decoupled Subspace Mining**
  - . . .
  - ENCLUS (Cheng et al., 1999)
  - HiCS (Keller et al., 2012)
Comparison of Subspace Mining Paradigms

- **Database** → **Subspace Search** → **Clustering Method** → **Result**

### Full Space
- **convex shapes**
  - K-Means (MacQueen, 1967)
  - ...
- **density-based**
  - DBSCAN (Ester et al., 1996)
  - ...

### Coupled Subspace Mining
- **convex shapes**
  - PROCLUS (Aggarwal et al., 1999)
  - ...
- **density-based**
  - SUBCLU (Kailing et al., 2004b)
  - ...

### Decoupled Subspace Mining
- **convex shapes**
  - ...
- **density-based**
  - ENCLUS (Cheng et al., 1999)
  - HiCS (Keller et al., 2012)
ENCLUS: Subspace Quality Assessment

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

- Idea based on the subspace selection in CLIQUE

⇒ Low entropy indicates highly interesting subspaces...

**Entropy of a subspace**

\[ H(X) = - \sum_{x \in X} d(x) \cdot \log d(x) \]

with the density \( d(x) \) of each cell \( x \in \text{grid } X \)
(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
HiCS: Assessment of High Contrast

A subspace with high contrast versus an irrelevant subspace
HiCS: Assessment of High Contrast

Idea: Search for violation of statistical independence

- More general than correlation analysis
- Statistical dependence in higher dimensions
HiCS: Main Idea

Comparing marginal and conditional distributions:
Subspace Contrast

For a low contrast subspace, we have:

\[
\frac{p_{s_1 \mid s_2, \ldots, s_d}(x_{s_1} \mid x_{s_2}, \ldots, x_{s_d})}{p_{s_2, \ldots, s_d}(x_{s_2}, \ldots, x_{s_d})} = \frac{p_{s_1, \ldots, s_d}(x_{s_1}, \ldots, x_{s_d})}{p_{s_2, \ldots, s_d}(x_{s_2}, \ldots, x_{s_d})} = p_{s_1}(x_{s_1})
\]

(1)

Definition: Subspace Contrast

Evaluate a subspace by a Monte Carlo algorithm with \( M \) iterations:

- pick a random marginal attribute \( s_i \)
- generate a random condition set \( C_i \)
- determine violation of equation 1.

\[
\text{contrast}(S) \equiv \frac{1}{M} \sum_{i}^{M} \text{deviation} \left( p_{s_i}^{(m)}, p_{s_i \mid C_i}^{(c)} \right)
\]
Subspace Contrast

For a low contrast subspace, we have:

\[
p_{s_1 \mid s_2, \ldots, s_d}(x_{s_1} \mid x_{s_2}, \ldots, x_{s_d}) = \frac{p_{s_1, \ldots, s_d}(x_{s_1}, \ldots, x_{s_d})}{p_{s_2, \ldots, s_d}(x_{s_2}, \ldots, x_{s_d})} = p_{s_1}(x_{s_1})
\]

\[
p_{s_{i \mid C_i}}^{(c)} = \frac{p_{s_i}^{(m)}}{p_{s_1}(x_{s_1})}
\]

Definition: Subspace Contrast

Evaluate a subspace by a Monte Carlo algorithm with \( M \) iterations:

- pick a random marginal attribute \( s_i \)
- generate a random condition set \( C_i \)
- determine violation of equation 1.

\[
\text{contrast}(S) \equiv \frac{1}{M} \sum_{i}^{M} \text{deviation} \left( p_{s_i}^{(m)}, p_{s_i \mid C_i}^{(c)} \right)
\]

How to generate the condition sets \( C_i \)?

How to instantiate \( \text{deviation} \left( p_{s_i}^{(m)}, p_{s_i \mid C_i}^{(c)} \right) \)?
The deviation function

Apply established methods for comparing samples ⇒ statistical test

- **Null hypothesis** \( H_0 \):
  - Samples are drawn from the same underlying distribution.
- Welch-t-Test and Kolmogorov-Smirnov are used, others are possible.

Adaptive Condition Sets

Given a subspace \( S \) with \( d \) dimensions.

⇒ number of conditions is: \( |C| = d - 1 \)

- **Challenge:**
  - Condition sets should scale to subspaces of arbitrary dimension \( d \).

- **Solution in HiCS:**
  - Ensure a fixed size of the selected sample:
    - select a random object as centroid for the selection slice
    - scale the selection slice according to \( N \sqrt[|C|]{\alpha} \)
Contribution of Subspace Search

- Decoupled subspace selection from clustering
- Statistical assessment of subspace contrast
- General pre-processing step useful for a variety of methods:
  - Clustering
  - Outlier Mining
  - Consensus / Ensemble Mining
  - Classification

Remaining Open Challenges:

- Produces **highly redundant set** of subspace clusters
- **Does not ensure dissimilarity** of subspace clusters
- Not able to compute alternatives w.r.t. a given clustering
Open Challenges for Multiple Clusterings

- Ensuring the dissimilarity 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

⇒ Novel (general) techniques for redundancy elimination required...

- **DUSC**: local pairwise comparison of redundancy (Assent et al., 2007)
- **StatPC**: statistical selection of non-red. clusters (Moise & Sander, 2008)
- **RESCU**: including interesting and excluding redundant clusters (Müller et al., 2009c)
- **OSCLU**: redundancy criteria based on subspace similarity (Günnemann et al., 2009)
Abstract redundancy model (e.g. RESCU):

- all possible clusters \( \text{ALL} \)
- relevance model
  - interestingness of clusters
  - redundancy of clusters
- relevant clustering \( M \subset \text{ALL} \)

Removing redundancy

Including multiple views

+ Model the dissimilarity between subspaces

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

\( \text{DB} \rightarrow 2^{[\text{DIM}]} \rightarrow \text{DBs} \)

\( \text{subspace clustering} \)

\( \text{ALL} = \text{Clust}_1, \ldots, \text{Clust}_n \)

⇒ result optimization

\( M = \text{Clust}_1, \text{Clust}_2 \)
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
OSCLU: Clustering in almost Orthogonal Subspaces

Idea:

- Redundancy based on coverage of clusters
  ⇒ A non-redundant cluster covers suff. many new objects in its view
  \[ M \subseteq ALL \text{ is redundancy free } \iff \forall C \in M : C \not\simred M \setminus \{ C \}_{\text{view}_C} \]
- Clusters in different views (suff. different subspaces) present new information
  ⇒ Select an optimal set of clusters in orthogonal subspaces
OSCLU: Clustering in almost Orthogonal Subspaces

Idea:
- Redundancy based on **coverage** of clusters
  - A non-redundant cluster covers **suff. many new** objects in its **view**
    - $M \subseteq ALL$ is redundancy free $\iff \forall C \in M : C \not\perp_{\text{red}} M \setminus \{C\}_{\text{view}_C}$
- Clusters in different views (**suff. different subspaces**) present new information
- Select an optimal set of clusters in orthogonal subspaces
OSCLU: Final Clustering Solution

Combinatorial Optimization

Opt $\subseteq$ ALL is an optimal, non-redundant clustering iff

$$Opt = \arg \max_{M \in \text{NonRed}} \left\{ \sum_{C \in M} q(C) \right\}$$

// objective fct.

with NonRed = \{ M \subseteq ALL | M is redundancy free \} // constraint

Computing Opt is NP-hard w.r.t. the number of clusters
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 \( \text{Res} \) for a given clustering \( \text{Known} \)

A valid clustering \( \text{Res} \) has to fulfill all properties defined in OSCLU but additionally has to be a valid alternative to \( \text{Known} \).

E.g.: If \( \text{Known} = \{C_2, C_5\} \), then \( \text{Res} = \{C_3, C_4, C_7\} \) would be a valid clustering.
Multiple Subspace Clusterings

Motivation

Original Data Space

Orthogonal Spaces

Subspace Projections

Multiple Sources

Summary

Open Challenge:

- Awareness of different clusterings
  - dissimilarity only between clusters not between clusterings
  - grouping of clusters in common subspaces required

Müller, Günnemann, Färber, Seidl

Discovering Multiple Clustering Solutions 100 / 157
MVGen: Multi View Mixture Models in Subspaces

Idea of MVGen (Günnemann et al., 2012)

- Learn **multiple mixture models in different views** of the data:
  - Each cluster is modeled by an **individual set of relevant dimensions**
  - Views, represent summaries of **similar subspace clusters**
  - All objects are grouped in each of these views ⇒ **multiple clusterings**
MVGen: Multi View Mixture Models in Subspaces

Idea of MVGen (Günnemann et al., 2012)

- Learn **multiple mixture models in different views** of the data:
  - Each **cluster** is modeled by an **individual set of relevant dimensions**
  - **Views**, represent summaries of **similar subspace clusters**
  - All objects are grouped in each of these views ⇒ **multiple clusterings**

![Graphical representation of MVGen concept](image)
MVGen: Generative Model

Multi-View Mixture Models

- each mixture describes a specific view
- mixture components in subspaces
  - random variables $S_{m,k,d}$ on $\{0, 1\}$

Bernoulli process:

$$S_{m,k,d} \sim \text{Bern}(V_m, d)$$
MVGen: Generative Model

Multi-View Mixture Models

- each mixture describes a specific view
  ⇒ mixture components in subspaces
- random variables $S_{m,k,d}$ on $\{0, 1\}$

\[
\begin{aligned}
S_{m,k,d} &
\alpha_{m,k,d} \\
\beta_{m,k,d} \\
\pi_{m,k} \\
\text{Dom}_{n,d} \\
\text{Sel}_{n,m} \\
\end{aligned}
\]

\[
\begin{aligned}
d \in D \\
k \in K \\
m \in M \\
n \in N \\
\end{aligned}
\]
MVGen: Generative Model

Multi-View Mixture Models

- each mixture describes a specific view
- mixture components in subspaces
  - random variables $S_{m,k,d}$ on $\{0, 1\}$
- same view $\Rightarrow$ similar subspaces
  - Bernoulli process: $S_{m,k,d} \sim \text{Bern}(V_{m,d})$
MVGEN: Generative Model

Multi-View Mixture Models

- objects follow **multiple** components
  - one per view
- dimensions might occur in several views

View 1

<table>
<thead>
<tr>
<th>Cluster</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Dimensional assignments:

- **View 1:**
  - **Cluster:** D1, D2, D3
  - **Dimensions:** D1, D2
  - **View:** View 1

- **View 2:**
  - **Cluster:** D1, D3
  - **Dimensions:** D3
  - **View:** View 2

- **View 3:**
  - **Cluster:** D1
  - **Dimensions:** D4
  - **View:** View 3

- **View 4:**
  - **Cluster:**
  - **Dimensions:** D2
  - **View:** View 4
MVGEN: Generative Model

Multi-View Mixture Models

- objects follow **multiple** components
  - one per view
- dimensions might occur in several views
  ⇒ for each object and dimension: one **dominant view**
  - $\text{Dom}_{n,d} \sim \text{Uni}(\{m \in M \mid S_{m,\text{Sel}_{n,m,d}} = 1\})$
**MVGen: Overall Multi-View Mixture Distribution**

$$X_{n,d} \mid \text{Dom}, \text{Sel}, S, \alpha, \beta \sim \begin{cases} \text{Beta}(\alpha_{i,j,d}, \beta_{i,j,d}) & \text{if } S_{i,j,d} = 1 \\ \text{Uni}(0, 1) & \text{else} \end{cases}$$

where $\text{Dom}_{n,d} = i$ and $\text{Sel}_{n,i} = j$
MVGen: Learning Objective

- given: observations $X$, aim: learn distributions
- challenge: not reasonable to maximize a posteriori probability

$$p(V, S, \alpha, \beta, \text{Dom}, \text{Sel}, \pi \mid X = X)$$

- all dimensions of a cluster would be relevant
- Beta distrib. always fits data better than Uniform distrib.

⇒ balance models’ goodness of fit and their simplicity
⇒ Bayesian model selection
**Algorithm**

- exact inference intractable
- iteratively update variables
  - cf. Gibbs Sampling, ICM

**Two learning phases:**

- learn $V^*, \text{Dom}^*, \text{Sel}^*, \pi^*$ by maximizing

\[
p(V, \text{Dom}, \text{Sel}, \pi \mid X = X) \propto \sum_S \int \int p(V, S, \alpha, \beta, \text{Dom}, \text{Sel}, \pi \mid X = X)
\]

- learn $S^*, \alpha^*, \beta^*$ by maximizing

\[
p(S, \alpha, \beta \mid X = X, V = V^*, \text{Dom} = \text{Dom}^*, \text{Sel} = \text{Sel}^*, \pi = \pi^*)
\]
Overview for this Paradigm

| DB | 2|DIM| DBs |
|---|---|---|

\[ ALL = \text{Clust}_1, \ldots, \text{Clust}_n \]

| DB | 2|DIM| DBs |
|---|---|---|

\[ M = \text{Clust}_1, \text{Clust}_2 \]

no dissimilarity (\textit{ALL})

consider dissimilarity (e.g. redundancy)

consider dissimilarity only implicitly

simultaneous processing

only recent approaches use previous knowledge

no prev. knowledge

too many clusters

optimized result size

\[ \geq 2 \] clusterings

dependent

mostly dependent

dependent

on the used clustering algorithm

enable interpretation of multiple clusterings
Open Challenges w.r.t. this Paradigm

- **Awareness of different clusterings**
  - first steps for dissimilarity between clusterings have been made
  - how many different clusterings should be output, is still unclear

- **Decoupling of existing solutions with high interdependences**
  - allow different models for flexible mining

- **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

⇒ How to combine results from different views
  - By merging clusterings to one consensus solution
  - Without merging the given sources
Challenge: Heterogeneous Data

- Information about objects is available from different sources
- Data sources are often heterogeneous (*multi-represented data*)
  ⇒ 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
  → Loss of information
- Construction of a feature space comprising all representations
  → Demands a new combined distance function
  → 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

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

⇒ 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 $orall x^{(1)} \in V^{(1)}, \forall x^{(2)} \in V^{(2)}$: $p(x^{(1)}, x^{(2)} | y) = p(x^{(1)} | y) \ast p(x^{(2)} | y)$.

- 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 $\theta$ 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

initialization

iteration 1 – Maximization $V^{(1)}$

iteration 1 – Maximization $V^{(2)}$

iteration 1 – Expectation $V^{(1)}$

iteration 1 – Expectation $V^{(2)}$
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
  ⇒ 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 $\varepsilon$-neighborhood of each view independently

$$\mathcal{N}_{\varepsilon_i}^{V(i)}(o) = \{ x \in DB \mid \text{dist}_i(o^{(i)}, x^{(i)}) \leq \varepsilon_i \}$$

- 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: 

\[
\text{COREU}^k_{\varepsilon_1, \ldots \varepsilon_m}(o) \Leftrightarrow \left| \bigcup_{o(i) \in o} N^{V(i)}_{\varepsilon_i}(o) \right| \geq k
\]

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 \( N^{V(i)}_{\varepsilon_i}(q) \), formally:

\[
\text{DIRREACHU}^k_{\varepsilon_1, \ldots \varepsilon_m}(q, p) \Leftrightarrow \text{COREU}^k_{\varepsilon_1, \ldots \varepsilon_m}(q) \land \exists i \in \{1, \ldots, m\} : p^{(i)} \in N^{V(i)}_{\varepsilon_i}(q)
\]
Intersection of Different Views

- well suited for data containing unreliable 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 → 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: $\text{COREIS}^k_{\varepsilon_1, \ldots, \varepsilon_m}(o) \iff \left| \bigcap_{i \in \{1, \ldots, m\}} N_{\varepsilon_i}(o) \right| \geq k$

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 $N_{\varepsilon_i}(q)$, formally:

$\text{DIRREACHIS}^k_{\varepsilon_1, \ldots, \varepsilon_m}(q, p) \iff \text{COREIS}^k_{\varepsilon_1, \ldots, \varepsilon_m}(q) \land \forall i \in \{1, \ldots, m\} : p(i) \in N_{\varepsilon_i}(q)$
Consensus Clustering on Subspace Projections

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

⇒ In contrast to previous paradigms, stabilize **one clustering solution**
⇒ 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

⇒ 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 $\theta$:

$$P_{i,j}^\theta = \sum_{l=1}^{k} P(l|i, \theta) \cdot P(l|j, \theta)$$
## Overview for this Paradigm

<table>
<thead>
<tr>
<th>consensus basis:</th>
<th>sources are known</th>
<th>low dimensional projections</th>
</tr>
</thead>
<tbody>
<tr>
<td>consensus transfer:</td>
<td>internal cluster model parameter</td>
<td>external objective function</td>
</tr>
<tr>
<td>consensus objective:</td>
<td>stable clusters</td>
<td>enable clustering in high dimensions</td>
</tr>
<tr>
<td>cluster model:</td>
<td>specific adaption</td>
<td>generalized consensus</td>
</tr>
</tbody>
</table>

⇒ 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

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
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

traditional clustering

multiple clustering solutions

multi-source clustering

multi-source multi-clustering solutions

DB
single clustering
clustering 1
clustering 2
clustering n
view 1
view 2
view m
clustering
view 1
view 2
view m
clustering 1
clustering 2
clustering n
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

<table>
<thead>
<tr>
<th>Space</th>
<th>Processing</th>
<th>Given Know.</th>
<th># Clusterings</th>
<th>Subspace Detect.</th>
<th>Flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caruana et al., 2006</td>
<td>original</td>
<td></td>
<td>m &gt;= 2</td>
<td></td>
<td>exchang. def.</td>
</tr>
<tr>
<td>Bae &amp; Bailey, 2006</td>
<td>original</td>
<td>iterative</td>
<td>m == 2</td>
<td></td>
<td>specialized</td>
</tr>
<tr>
<td>Gondek &amp; Hofmann, 2004</td>
<td>original</td>
<td>iterative</td>
<td>m == 2</td>
<td></td>
<td>specialized</td>
</tr>
<tr>
<td>Jain et al., 2008</td>
<td>original</td>
<td>simultaneous</td>
<td>m == 2</td>
<td></td>
<td>specialized</td>
</tr>
<tr>
<td>Hossain et al., 2010</td>
<td>original</td>
<td>simultaneous</td>
<td>m == 2</td>
<td></td>
<td>specialized</td>
</tr>
<tr>
<td>Dang &amp; Bailey, 2010a</td>
<td>original</td>
<td>simultaneous</td>
<td>m == 2</td>
<td></td>
<td>specialized</td>
</tr>
<tr>
<td>Davidson &amp; Qi, 2008</td>
<td>transformed</td>
<td>iterative</td>
<td>m == 2</td>
<td>dissimilarity</td>
<td>exchang. def.</td>
</tr>
<tr>
<td>Qi &amp; Davidson, 2009</td>
<td>transformed</td>
<td>iterative</td>
<td>m == 2</td>
<td>dissimilarity</td>
<td>exchang. def.</td>
</tr>
<tr>
<td>Cui et al., 2007</td>
<td>transformed</td>
<td>iterative</td>
<td>m == 2</td>
<td>dissimilarity</td>
<td>exchang. def.</td>
</tr>
<tr>
<td>Agrawal et al., 1998</td>
<td>subspaces</td>
<td></td>
<td>m &gt;= 2</td>
<td>no dissimilarity</td>
<td>specialized</td>
</tr>
<tr>
<td>Sequeira &amp; Zaki, 2004</td>
<td>subspaces</td>
<td></td>
<td>m &gt;= 2</td>
<td>no dissimilarity</td>
<td>specialized</td>
</tr>
<tr>
<td>Moise &amp; Sander, 2008</td>
<td>subspaces</td>
<td>simultaneous</td>
<td>m &gt;= 2</td>
<td>no dissimilarity</td>
<td>specialized</td>
</tr>
<tr>
<td>Müller et al., 2009b</td>
<td>subspaces</td>
<td>simultaneous</td>
<td>m &gt;= 2</td>
<td>no dissimilarity</td>
<td>specialized</td>
</tr>
<tr>
<td>Günneumann et al., 2009</td>
<td>subspaces</td>
<td>simultaneous</td>
<td>m &gt;= 2</td>
<td>dissimilarity</td>
<td>specialized</td>
</tr>
<tr>
<td>Günneumann et al., 2010</td>
<td>subspaces</td>
<td>simultaneous</td>
<td>m &gt;= 2</td>
<td>dissimilarity</td>
<td>specialized</td>
</tr>
<tr>
<td>Cheng et al., 1999</td>
<td>subspaces</td>
<td></td>
<td>m == 2</td>
<td>no dissimilarity</td>
<td>specialized</td>
</tr>
<tr>
<td>Niu &amp; Dy, 2010</td>
<td>subspaces</td>
<td></td>
<td>m &gt;= 2</td>
<td>dissimilarity</td>
<td>exchang. def.</td>
</tr>
<tr>
<td>Bickel &amp; Scheffer, 2004</td>
<td>multi-source</td>
<td>simultaneous</td>
<td>m = 1</td>
<td>given views</td>
<td>specialized</td>
</tr>
<tr>
<td>Kailing et al., 2004</td>
<td>multi-source</td>
<td>simultaneous</td>
<td>m = 1</td>
<td>given views</td>
<td>specialized</td>
</tr>
<tr>
<td>Fern &amp; Brodley, 2003</td>
<td>multi-source</td>
<td></td>
<td>m = 1</td>
<td>no dissimilarity</td>
<td>exchang. def.</td>
</tr>
</tbody>
</table>

Let us discuss the secondary characteristics of our taxonomy...
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?
From the perspective of **how many clusterings** are provided:

- $m = 1$ (traditional clustering)
- $m = 2$ 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
  ⇒ 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 similar 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 similar views?
- Selection of views independent of cluster models?
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).  
- 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>
<thead>
<tr>
<th>Section</th>
<th>search space taxonomy</th>
<th>processing</th>
<th>knowledge</th>
<th>flexibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec. 2</td>
<td>original space</td>
<td>iterative</td>
<td>given k.</td>
<td>exch. def.</td>
</tr>
<tr>
<td>alg2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg3</td>
<td></td>
<td>simultan.</td>
<td>no given k.</td>
<td>specialized</td>
</tr>
<tr>
<td>alg4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sec. 3</td>
<td>orthogonal transformations</td>
<td>iterative</td>
<td>given k.</td>
<td>exch. def.</td>
</tr>
<tr>
<td>alg5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sec. 4</td>
<td>subspace projections</td>
<td>simultan.</td>
<td>given k.</td>
<td>specialized</td>
</tr>
<tr>
<td>alg7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg9</td>
<td></td>
<td></td>
<td>no given k.</td>
<td></td>
</tr>
<tr>
<td>alg10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sec. 5</td>
<td>multiple views/sources</td>
<td>simultan.</td>
<td>no given k.</td>
<td>specialized</td>
</tr>
<tr>
<td>alg11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>alg13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</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 sguennem@cs.cmu.edu{faerber, seidl}@cs.rwth-aachen.de

Thanks for attending the tutorial! Any questions?

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


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

DUSC: Dimensionality Unbiased Subspace Clustering.
In: ICDM.

INSCY: Indexing Subspace Clusters with In-Process-Removal of Redundancy.
In: ICDM.

COALA: A Novel Approach for the Extraction of an Alternate Clustering of High Quality and High Dissimilarity.
In: ICDM.

Bae, E., Bailey, J., & Dong, G. 2010.
A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings.
Data Min. Knowl. Discov., 21(3).
When is nearest neighbors meaningful.
_In: IDBT._

Multi-View Clustering.
_In: ICDM._

Combining labeled and unlabeled data with co-training.
_In: COLT._

Density Connected Clustering with Local Subspace Preferences.
_In: ICDM._

Meta Clustering.
_In: ICDM._
References IV

Extracting Relevant Structures with Side Information.
*In: NIPS.*

Entropy-based subspace clustering for mining numerical data.
*In: SIGKDD.*

Finding Clusters in Subspaces of Very Large Multi-dimensional Datasets.
*In: ICDE.*

Non-redundant Multi-view Clustering via Orthogonalization.
*In: ICDM.*

Learning multiple nonredundant clusterings.
*TKDD, 4*(3).
References V

Generation of Alternative Clusterings Using the CAMI Approach. 
*In: SDM.*

A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. 
*In: SIGKDD.*

Dang, XuanHong, & Bailey, James. 2013a. 
A framework to uncover multiple alternative clusterings. 

Dang, XuanHong, & Bailey, James. 2013b. 
Generating multiple alternative clusterings via globally optimal subspaces. 
*Data Mining and Knowledge Discovery*, 1–24.
References VI

Davidson, I., & Qi, Z. 2008.
Finding Alternative Clusterings Using Constraints.
In: ICDM.

De Bie, T. 2011.
An information theoretic framework for data mining.
In: KDD.

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

Weighted cluster ensembles: Methods and analysis.
TKDD, 2(4).
A density-based algorithm for discovering clusters in large spatial databases.  
*In: SIGKDD.*

Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach.  
*In: ICML.*

Conditional information bottleneck clustering.  
*In: ICDM, Workshop on Clustering Large Data Sets.*

Non-Redundant Data Clustering.  
*In: ICDM.*
Non-redundant clustering with conditional ensembles.  
*In: SIGKDD.*

Clustering with Model-level Constraints.  
*In: SDM.*

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.*
ASCLU: Alternative Subspace Clustering.
*In: MultiClust Workshop at SIGKDD.*

Günnemann, S., Färber, I., & Seidl, T. 2012.
Multi-View Clustering Using Mixture Models in Subspace Projections.
*In: KDD.*

Unifying dependent clustering and disparate clustering for non-homogeneous data.
*In: SIGKDD.*

Simultaneous Unsupervised Learning of Disparate Clusterings.
*In: SDM.*
References

Scalable Density-Based Distributed Clustering. 
_In: PKDD._

Ranking interesting subspaces for clustering high dimensional data. 
_In: PKDD._

Clustering Multi-Represented Objects with Noise. 
_In: PAKDD._

Density-Connected Subspace Clustering for High-Dimensional Data. 
_In: SDM._

HiCS: High Contrast Subspaces for Density-Based Outlier Ranking. 
_In: ICDE._


MacQueen, J. 1967.
Some methods for classification and analysis of multivariate observations.
_in: Berkeley Symp. Math. stat. & prob._

Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering.
_in: SIGKDD._

P3C: A Robust Projected Clustering Algorithm.
_in: ICDM._

DensEst: Density Estimation for Data Mining in High Dimensional Spaces.
_in: SDM._


Nguyen, Hoang Vu, Müller, Emmanuel, Vreeken, Jilles, Keller, Fabian, & Böhm, Klemens. 2013. CMI: An Information-Theoretic Contrast Measure for Enhancing Subspace Cluster and Outlier Detection. 
*In: SDM.*

*In: ICML.*

*SIGKDD Explorations, 6*(1).

*In: SIGMOD.*
Qi, Z., & Davidson, I. 2009.  
A principled and flexible framework for finding alternative clusterings.  
In: SIGKDD.

SCHISM: A New Approach for Interesting Subspace Mining.  
In: ICDM.

Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions.  

Vinh, N. Xuan, & Epps, J. 2010.  
minCEntropy: A Novel Information Theoretic Approach for the Generation of Alternative Clusterings.  
In: ICDM.

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