Density Based Subspace Clustering over Dynamic Data

Hans-Peter Kriegel, Peer Kröger, Irene Ntoutsi, Arthur Zimek

Ludwig-Maximilians-Universität (LMU)
Munich, Germany
www.dbs.ifi.lmu.de

SSDBM, 20-22/7/2011, Portland OR

Outline

- Motivation
- Subspace clustering for static data (PreDeCon)
- Subspace clustering for dynamic data (incPreDeCon)
- Experiments
- Conclusions & Outlook

Modern data properties

- High dimensionality
  - We tend to store more and more details regarding our applications
- Dynamic nature
  - We tend to keep track of the population evolution over time
- Due to the advances in hardware/ software, we have nowadays the ability to store all these data

- Applications: Telco, Banks, Retail industry, WWW ...

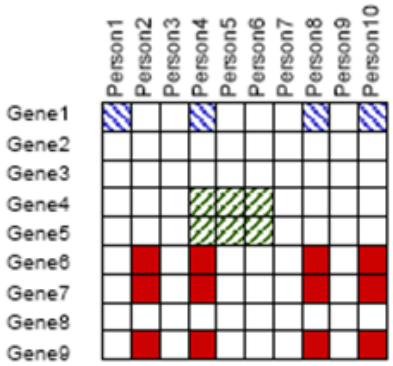
The clustering problem becomes even harder under these characteristics!

Dynamic nature of data

- Data evolution ➔ cluster evolution
  - How can we maintain online the clusters as new data arrive over time?
- Different lines of research, corresponding to different application requirements:
  - Incremental methods
    - Appropriate for data arriving at a low rate
    - Require access to the raw data for the re-arrangement of the clustering, but produces lossless results (e.g. incDBSCAN, incOPTICS)
  - Stream methods
    - Appropriate for data arriving at a rapid rate
    - No random access to the raw data, work upon summaries, thus produce lossy results (e.g. CluStream, DenStream)

High dimensionality of data

1. The "curse of dimensionality"
   - $(D_{max\_d} - D_{min\_d}) / D_{min\_d}$ converges to zero with increasing dimensionality *d*
     - D_{min_d}: distance to the nearest neighbor in d dimensions
     - D_{max_d}: distance to the farthest neighbor in d dimensions
2. Different features may be relevant for different clusters



- Different solutions have been proposed:
  - Feature selection methods (e.g. PCA): fail in 2., because they are global
  - Subspace clustering methods: search for both clusters and subspaces where these clusters exist

Our approach

- We chose:
  - *Incremental* clustering to deal with the dynamic nature of the data
  - *Subspace* clustering to deal with the high dimensionality of the data
- We work with the (static) algorithm PreDeCon:
  - a subspace clustering algorithm
  - relies on a density based model → updates are expected to cause only limited local changes
- We propose an incremental version that maintains density based subspace clusters as new data arrive over time
  - Allows both points and dimensions associated to a cluster to evolve
  - Deals with both single and batch updates
  - Can serve as a framework for monitoring changes in dynamic environments

Outline

- Motivation
- Subspace clustering for static data (PreDeCon)
- Subspace clustering for dynamic data (incPreDeCon)
- Experiments
- Conclusions & Outlook

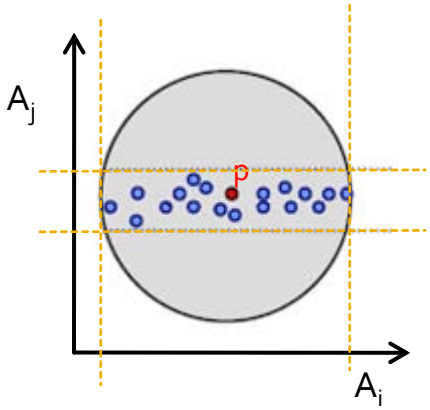
PreDeCon basics - I

- Extends DBSCAN to high dimensional spaces by incorporating the notion of dimension preferences in the distance function
- For each point p, it defines its subspace preference vector:

$$\bar{\mathbf{w}}_p = (w_1, w_2, \ldots w_d) \qquad w_i = \begin{cases} 1 & if \quad \text{VAR}_i > \delta \\ \kappa & if \quad \text{VAR}_i \leq \delta \end{cases}$$

- VAR_i is the variance along dimension j in $N_\varepsilon(p)$:

$$\text{VAR}_{A_i}(\mathcal{N}_\varepsilon(p)) = \frac{\sum_{q \in \mathcal{N}_\varepsilon(p)} (dist(\pi_{A_i}(p), \pi_{A_i}(q)))^2}{|\mathcal{N}_\varepsilon(p)|}$$



δ, κ (κ>>1) are input parameters

PreDeCon basics - II

- Preference weighted distance function:

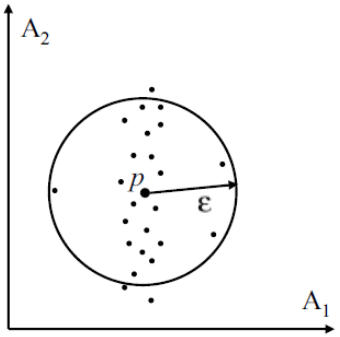
$$dist_p(p,q) = \sqrt{\sum_{i=1}^{d} \frac{1}{w_i} \cdot (\pi_{A_i}(p) - \pi_{A_i}(q))^2}$$

$$dist_{pref}(p,q) = \max\{dist_p(p,q), dist_q(q,p)\}$$

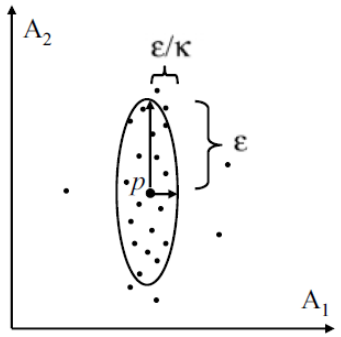
- Preference weighted ε-neighborhood:

$$\mathcal{N}_{\varepsilon}^{\bar{w}_p}(p) = \{x \in \mathcal{D} \mid dist_{pref}(p,x) \leq \varepsilon\}$$

simple ε-neighborhood



↔



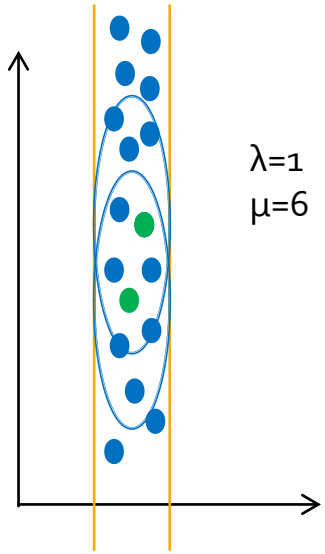
preference weighted ε-neighborhood

PreDeCon: Subspace preference clusters

- Preference weighted core points:

$$\text{CORE}_{\text{den}}^{\text{pref}}(p) \Leftrightarrow \boxed{\text{PDIM}(\mathcal{N}_\varepsilon(p)) \leq \lambda} \wedge \boxed{|\mathcal{N}_\varepsilon^{\bar{w}_o}(p)| \geq \mu}$$

1 2



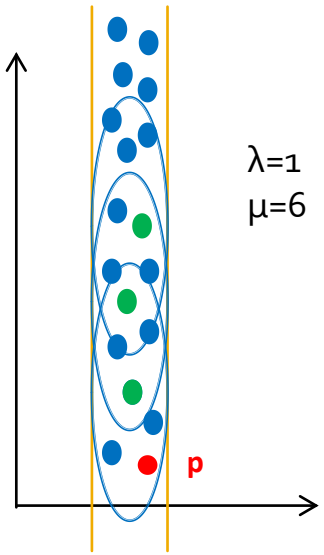
- Direct density reachability, reachability and connectivity are defined based on preference weighted core points
- A *subspace preference cluster* is a maximal density connected set of points associated with a certain subspace preference vector.

Outline

- Motivation
- Subspace clustering for static data (PreDeCon)
- Subspace clustering for dynamic data (incPreDeCon)
- Experiments
- Conclusions & Outlook

PreDeCon over dynamic data

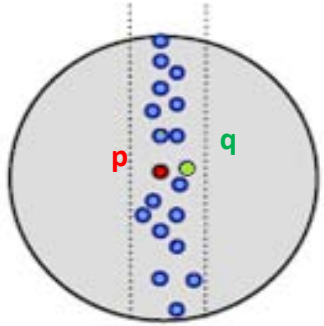
- **Observation:** A subspace preference cluster is uniquely determined by one of its preference weighted core points.
- **Idea:** Check whether the insertion of a new point p affects the core member property of the points in the dataset



  - 3 interesting changes might occur w.r.t. core property:
    - a non-core point might turn into core ➔ new density connections
    - a core point might turn into non-core ➔ demolished density connections
    - a core point might remain core, but under different dimension preferences ➔ both cases

Affected core points

- The insertion of p, *directly* affects the points q in its ε-neighborhood.
  - $N_\varepsilon(q)$ is affected because p is now a member of it

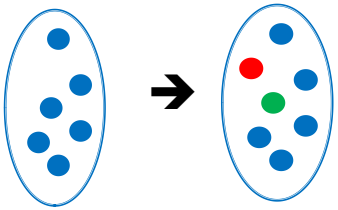


$$\mathcal{N}_\varepsilon(q) \rightarrow \text{VAR}_{A_i}(\mathcal{N}_\varepsilon(q)) \rightarrow \bar{\mathbf{w}}_q \rightarrow \text{PDIM}(\mathcal{N}_\varepsilon(q)),\ \mathcal{N}_\varepsilon^{\bar{\mathbf{w}}}(q)$$

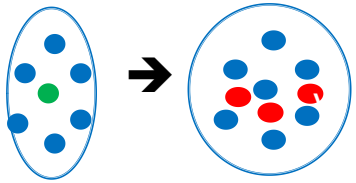
$$\text{CORE}_{\text{den}}^{\text{pref}}(p) \Leftrightarrow \text{PDIM}(\mathcal{N}_\varepsilon(p)) \leq \lambda \wedge |\mathcal{N}_\varepsilon^{\bar{\mathbf{w}}_o}(p)| \geq \mu$$

The core property of q might be affected

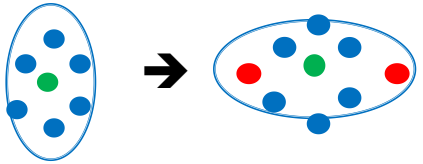
- Effect in core property:
  - non-core ➔ core



  - core ➔ non-core



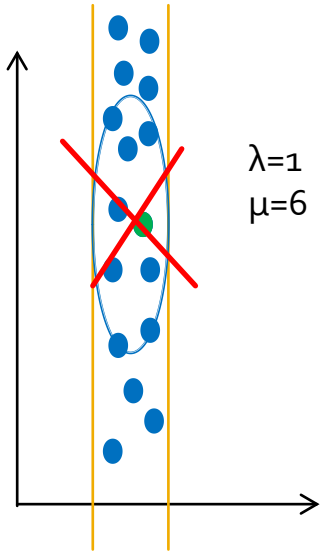
  - core ➔ core, under different preferences



λ=1
μ=6

Affected points

- The change in the core property of q, might cause changes to points that are preference weighted reachable from q:
  - if q: core→non-core, any density connectivity relying on q is destroyed
  - if q: non-core→core, some new density connectivity might arise
  - If q core → core but under different dimension preferences, both might occur



- Affected points:

$$\text{AFFECTED}_{\mathcal{D}}(p) = \mathcal{N}_{\varepsilon}(p) \quad \cup \quad \{q | \exists o \in \mathcal{N}_{\varepsilon}(p) : REACH_{den}^{pref}(o, q) \text{ in } \mathcal{D}^*\}$$

$$\mathcal{D}^* = \mathcal{D} \cup \{p\}$$

Restructuring the affected objects

- Naïve solution: cluster $AFFECTED_D(p)$ from scratch
- But, any changes in $AFFECTED_D(p)$ are initiated by points in $N_\varepsilon(p)$
  - No need to consider all points in $N_\varepsilon(p)$, just those with affected core member property (AFFECTEDCORE)
  - If a point q' is an affected core point, we consider as seeds points for its update any core point q in its preferred neighborhood.

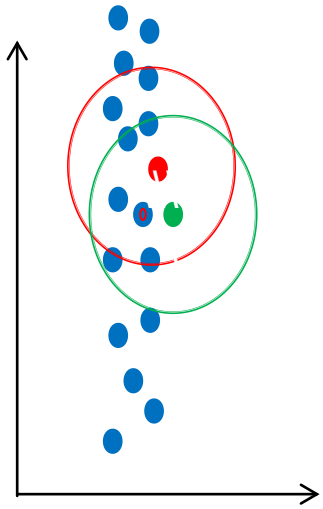
$$\text{UPDSEED} = \{q \mid q \text{ is core in } \mathcal{D}^*, \exists q' : q \in \mathcal{N}_\varepsilon(q') \text{ and } q' \text{ changes his core member property } \bar{\bar{w}} \text{ in } \mathcal{D}^*\}$$

$$\mathcal{D}^* = \mathcal{D} \cup \{p\}$$

  - Apply the expand procedure of PreDeCon using the points in the UPDSEED

Batch updates

- **Idea:** Don't treat insertions separately, rather insert the whole batch and update the clustering based on the whole batch
- Is more efficient since some computations might take place only once:
  - The preference vector computation for each point
  - The set of affected core points
  - The affected points
- Efficient for cases where the updates are related to each other
  - e.g. they might correspond to patients suffering from a specific disease



Outline

- Motivation
- Subspace clustering for static data (PreDeCon)
- Subspace clustering for dynamic data (incPreDeCon)
- Experiments
- Conclusions & Outlook

Experiments

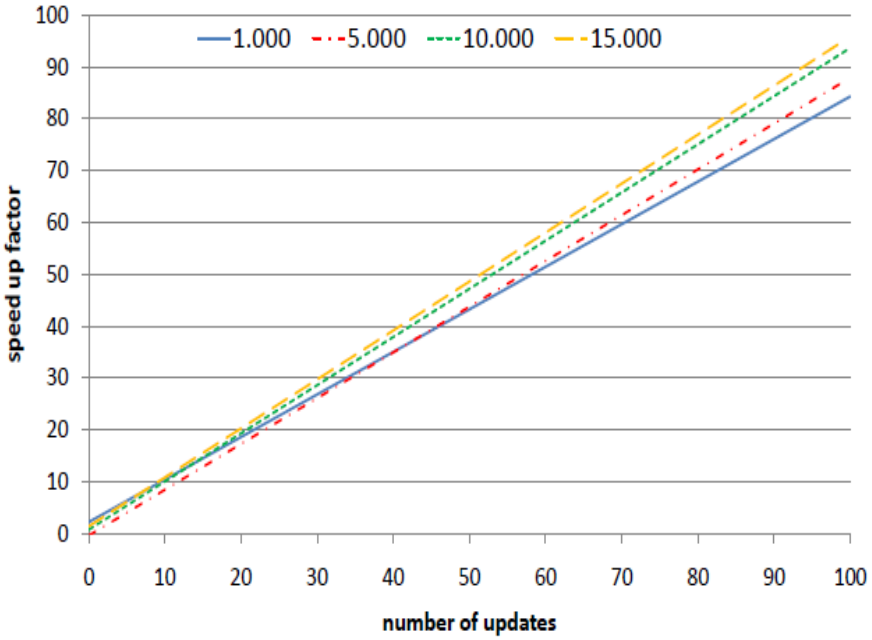
- We compared incPreDeCon to PreDeCon
- We evaluated # range queries required by each algorithm

- For each dataset,
  - Perform 100 random inserts
  - Compute the range queries for PreDeCon
  - Compute the range queries for incPreDeCon

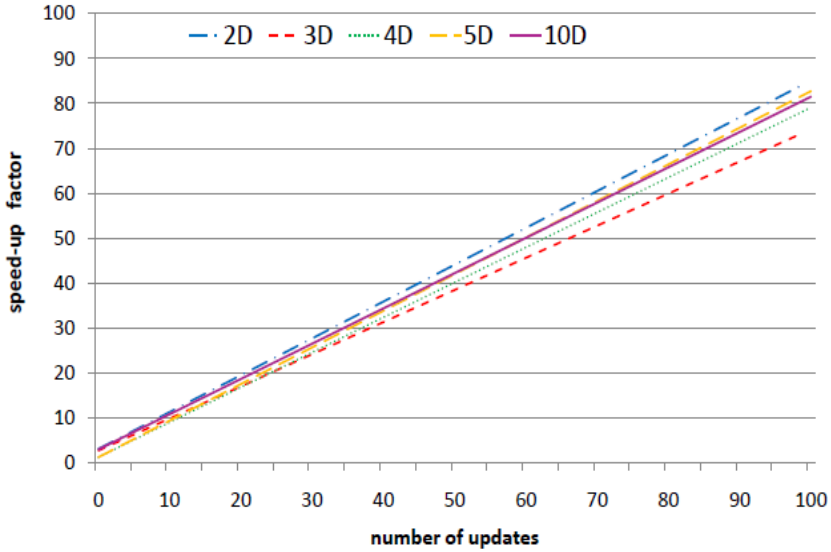
$$SpeedupFactor = \frac{COST_{\text{PREDECON}}(\mathcal{D}^*)}{COST_{\text{INCPREDECON}}(\mathcal{D} \cup \mathcal{U})}$$

Evaluation (single updates)

- Varying the population

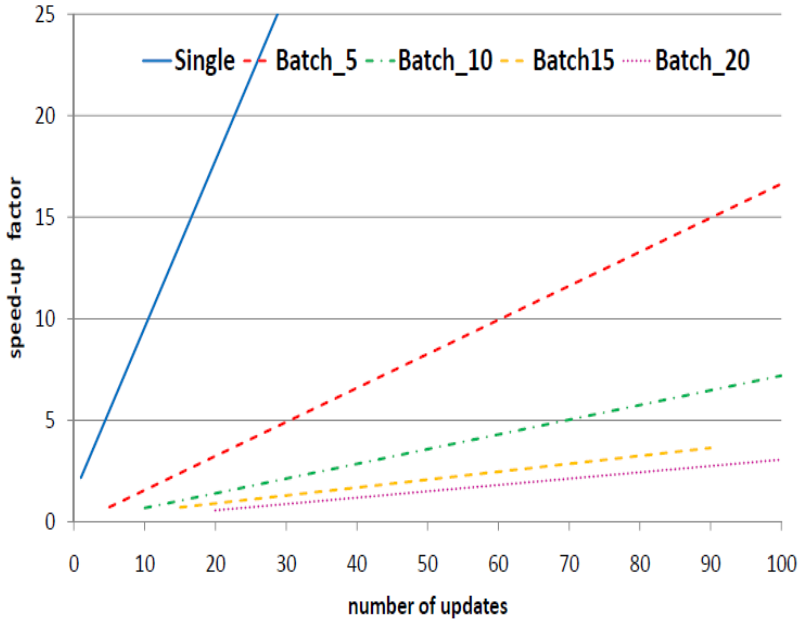


- Varying the dimensions

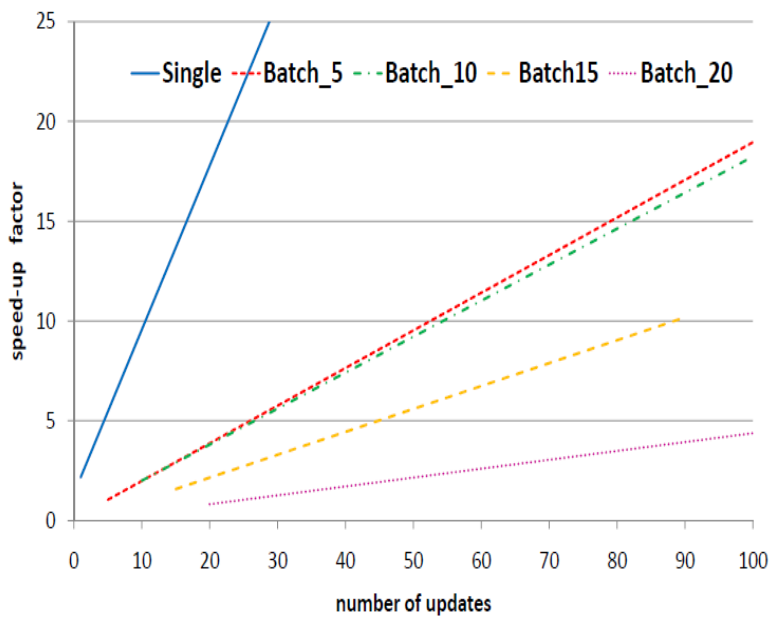


Evaluation (batch updates)

- Random updates

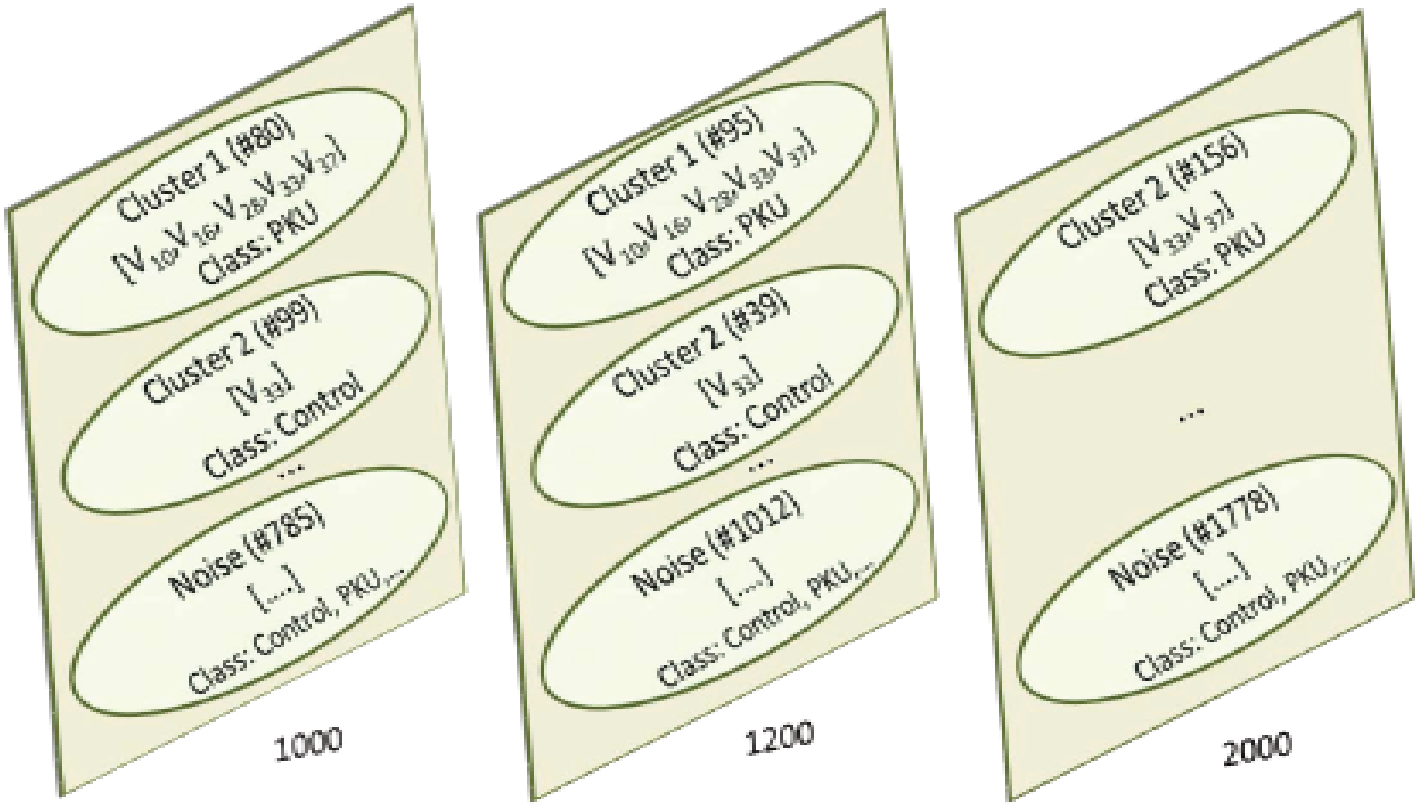


- "Local" updates



Bavarian newborn screening data

- Concentration of different metabolites in the blood of newborns in Bavaria



Outline

- Motivation
- Subspace clustering for static data (PreDeCon)
- Subspace clustering for dynamic data (incPreDeCon)
- Experiments
- Conclusions & Outlook

Conclusions

- We proposed a density based subspace clustering algorithm for dynamic data
  - It allows both points and dimensions associated to a cluster to evolve over time
  - It deals with single and batch updates
  - It can serve as a framework for monitoring changes in dynamic environments

Open issues

(instead of listing them, a categorization of the existing approaches)

	Static clustering	Inc/ Stream clustering	Subspace clustering	Inc/ Stream Subspace clustering
Partitioning methods	• k-Means • k-Medoids	• Single-pass k-Means • STREAM k-Means • CluStream	•PROCLUS	•HPStream
Density-based methods	• DBSCAN • OPTICS	• DenStream • incDBSCAN • incOPTICS	•PreDeCon	•incPreDeCon
Grid-based methods	• STING	• DStream	•CLIQUE	•DUCStream

Questions?

Thank you
for your attention!

Alexander von Humboldt
Stiftung/Foundation

The speaker‘s attendance at this conference was sponsored by the Alexander von Humboldt Foundation

http://www.humboldt-foundation.de