Towards Subspace Clustering on Dynamic Data: An Incremental Version of PreDeCon

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

Ludwig-Maximilians-Universität (LMU),

Munich, Germany

www.dbs.ifi.lmu.de

StreamKDD, 25/7/2010, Washington DC

DATABASE SYSTEMS GROUP

Overview

- Motivation
- Related work
- Density based subspace clustering – PreDeCon
- Incremental PreDecon
- Evaluation
- Summary and next steps

Motivation

- Modern applications:
  - Web (navigation data, content data, traffic data)
  - Telecommunication, Banks, Health care systems
  - Sensor networks, Position tracking systems …
- Data characteristics:
  - High dimensionality
  - Dynamic nature
  - Huge amounts of data
- Need for mining over high dimensional, dynamic, huge amounts of data !!!

High dimensionality

- The curse of dimensionality:
  - All points are almost equidistant from each other in high dimensional spaces.
  - The distances between points cannot be used to differentiate them → clustering does not make sense!
- Different features may be relevant for different clusters
  - Feature selection methods, e.g. PCA fail because are global

⇨ Subspace clustering
  - Searches for clusters of objects and subspaces where these clusters exist.

Dynamic data

- As new data arrive, the so far built clustering should be updated to reflect these changes:
- Lines of research:
  - Incremental methods
    - e.g., incDBSCAN, incOPTICS
  - Adaptive methods
    - e.g., STREAM, DUCStream (CLIQUE based), CLIQUE$^+$(DEMON framework)
      - Might also work over streams
  - Stream methods (summary based)
    - e.g., CluStream, DenStream, HPStream (subspace clustering)

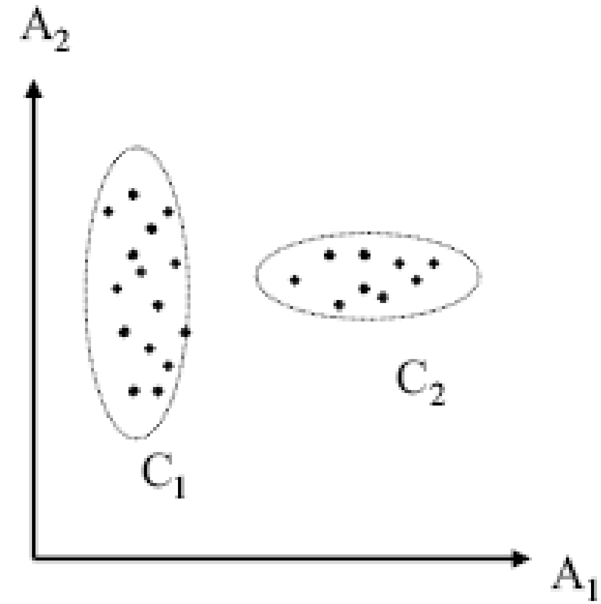
Our method

- We choose:
  - Subspace clustering for high dimensionality
  - Incremental clustering for dynamic data
- We work upon algorithm PreDeCon:
  - a subspace clustering algorithm
  - relies on a density based clustering model, so updates usually cause only limited local changes.

PreDeCon

- It adapts the density-based cluster model of DBSCAN to projected clustering

- PreDeCon applies DBSCAN with a *weighted Euclidean distance function* which distinguish between preferable and non-preferable dimensions

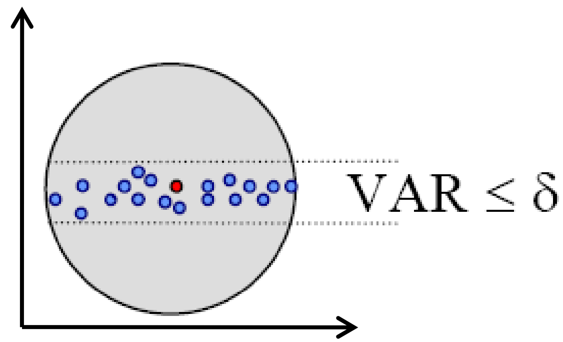


Dimension preferences

- For each point p, its subspace preference vector is defined:

$$\bar{\mathbf{w}}_p = (w_1, w_2, \ldots w_d)$$

$$w_i = \begin{cases} 1 & \text{if} \quad \mathrm{VAR}_i > \delta \\ \kappa & \text{if} \quad \mathrm{VAR}_i \leq \delta \end{cases}$$



- VAR_i is the variance of the ε-neighborhood of p in the entire d-dimensional space

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

δ, κ (κ>>1) are input parameters

Preference weighted distance

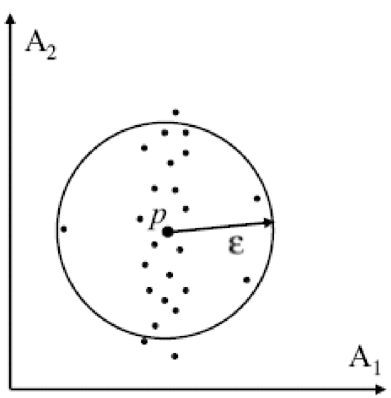
- Preference weighted distance function:

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

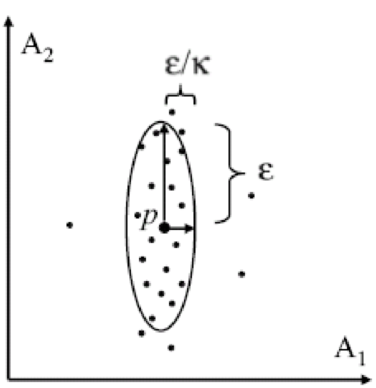
$$dist_{pref}(p,q) = \max\{dist_p(p,q), dist_q(q,p)\}$$

- Preference weighted ε-neighborhood:

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



Simple Euclidean ε-neighborhood



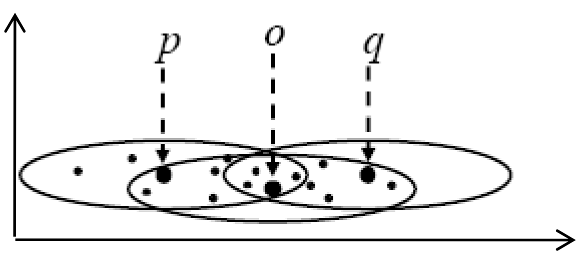
Preference weighted Euclidean ε-neighborhood

Preference weighted core points

- Preference weighted core points:

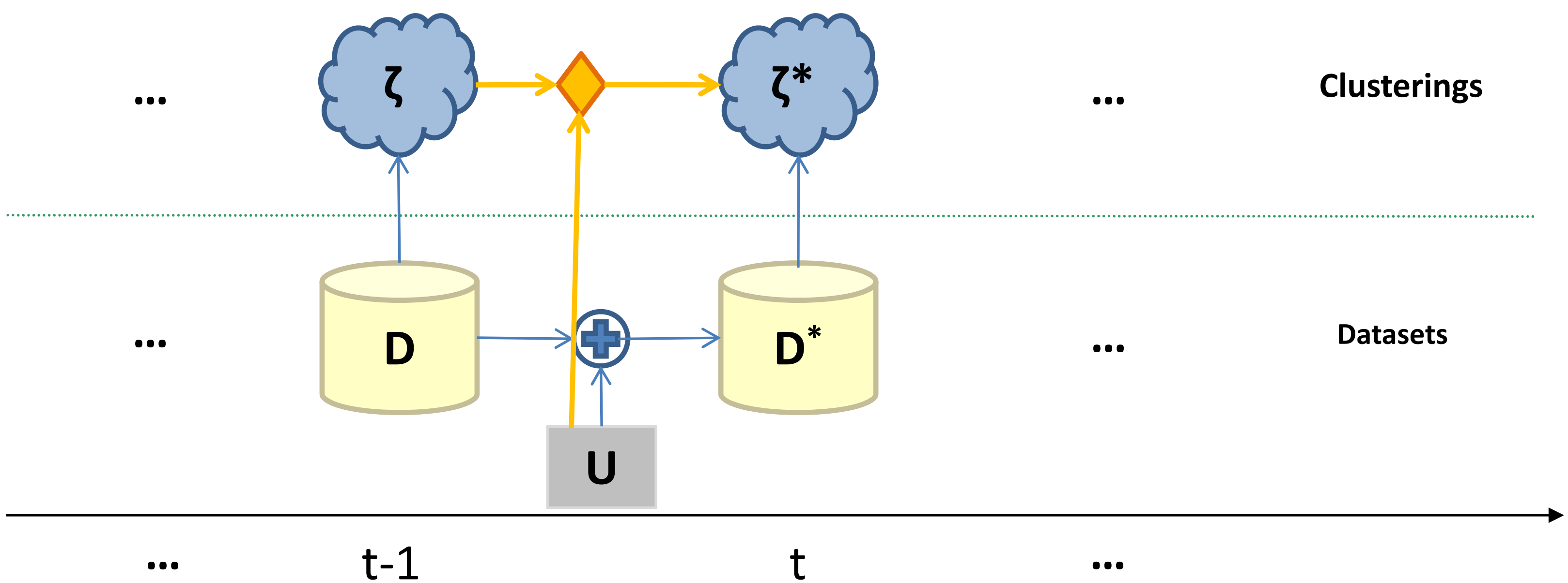
$$\text{CORE}_{\text{den}}^{\text{pref}}(p) \Leftrightarrow \underbrace{\boxed{\text{PDIM}(\mathcal{N}_\varepsilon(p)) \leq \lambda}}_{\text{Condition 1}} \wedge \underbrace{\boxed{|\mathcal{N}_\varepsilon^{\bar{\mathbf{w}}_o}(p)| \geq \mu}}_{\text{Condition 2}}$$

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



Incremental rationale

- At time **t-1**: **D** (dataset), **ζ** (clustering derived uppon D)
- At time **t**: **U** (new coming data)
- **Goal:** Update ζ, so as to derive the valid clustering **ζ*** at t.

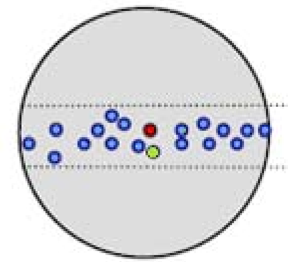


Incremental PreDeCon

- **Observation:** A preference weighted cluster is determined uniquely by one of its preference weighted core points.
- **Idea:** Check whether the update affects the core member property of some point
- **Sketch of the algorithm:**
  – Find affected core points
  – Find affected points
  – Update the clustering model

Affected core points

- The insertion of p, directly affects the points q its ε-neighborhood.
  - Nε(q) is affected because p is now a member of it



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

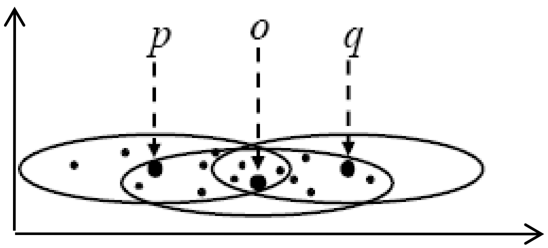
$$\boxed{\mathrm{CORE}_{\mathrm{den}}^{\mathrm{pref}}(q) \Leftrightarrow \mathrm{PDIM}(\mathcal{N}_\varepsilon(q)) \leq \lambda \wedge |\mathcal{N}_\varepsilon^{\bar{\mathbf{w}}_o}(q)| \geq \mu}$$

The core property of q might be affected

- Effect on the core member property of q:
  - core → non-core
  - non-core → core
  - core → core but under different preferences

Affected points

- The insertion of p might cause indirect effects to points that are preference weighted reachable from p:
  - if q: core→non-core after insertion, any density connectivity relying on q is destroyed
  - if q: non-core→core after insertion, some new density connectivity might arise



- Affected points:

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

From where to start restructuring?

- Note that changes in $\text{AFFECTED}_D(p)$ are initiated by points in the ε-neighborhood of p
  - No need to consider all points, 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.

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

Update the clustering model

- Call *expandCluster()* starting with UPDSEED set.
- The pseudoce of the algorithm:

```
algorithm incPreDeCon(𝒟, 𝒰, ε, μ, λ, δ)
  for each p ∈ 𝒰 do
    1. 𝒟* = 𝒟 ∪ p;
    2. compute the subspace preference vector w̄_p;
    // update preferred dimensionality and
    // check changes in the core member property in 𝒩_ε(p)
    3. for each q ∈ 𝒩_ε(p) do
    4.     update w̄_q;
    5.     check changes in the core member property of q;
    6.     if change exists, add q to AffectedCore;
    7. compute UpdSeed based on AffectedCore
    8. for each q ∈ UpdSeed do
    9.     expandCluster(𝒟*, UpdSeed, q, ε, μ, λ);
  end;
```

Evaluation

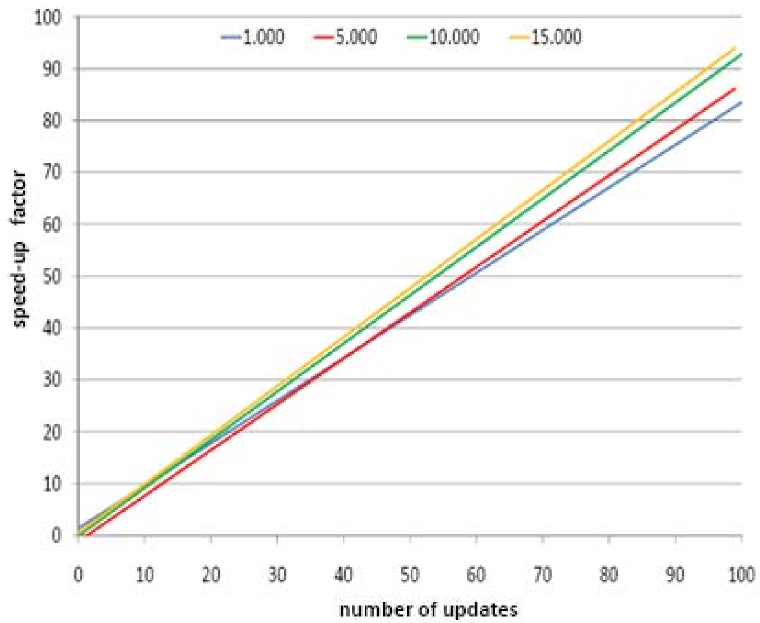
- We evaluated incPreDeCon vs PreDeCon w.r.t. the number of range queries
- For each dataset, we performed 100 random inserts, and counted the number of range queries required by incPreDeCon and PreDeCon.

$$SpeedupFactor = \frac{COST_{\textsc{PreDeCon}}(\mathcal{D}^*)}{COST_{\textsc{incPreDeCon}}(\mathcal{D} \cup \mathcal{U})}$$

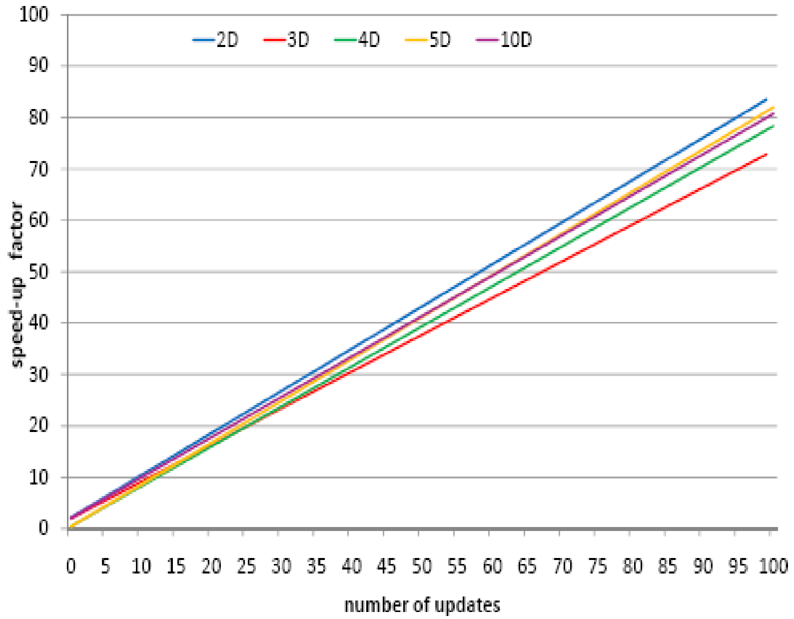
- Costs:
  - PreDeCon: 2|D|
  - incPreDeCon: $1+ 2|N\varepsilon(p)|+|AFFECTED_D(p)|$

Evaluation

- Comparison w.r.t. cluster population



- Comparison w.r.t. dimensionality



Summary and next steps

- We presented the first incremental subspace clustering algorithm, based on PreDeCon
  - The update strategy manages to restructure only the affected part of the old clustering
- Future work:
  - Subspace clustering over fast changing environments like data streams where access to raw data is not allowed
  - A unified framework for turning static subspace clustering methods into dynamic methods
  - Change detection in subspace clusters, e.g. create, delete, split, merge … but what about subspace preferences also (e.g. move in a new subspace, “losing” some dimension)?

Questions?

Thank you

for your attention!

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

http://www.humboldt-foundation.de