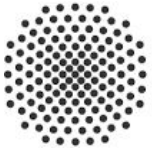


Universität Stuttgart

Hauptseminar: **Advanced Data Engineering**
Term: **Winter 2018/2019**
Professor: Dr. rer. nat. Melanie Herschel
Supervisors: Houssem Ben Lahmar, Leonardo Gazzarri

Clustering of evolving stream data

Jagan Shanmugam

Motivation

- Large amounts of data generated nowadays - data evolves over time - considered as an unbounded stream
- To gain insight from stream data, we need to cluster the streaming data efficiently in real-time
- Use case: **News recommendation system** - provide users with news from the same/similar cluster
    - News is generated continuously and will fade out over time

Outline

- Basic concepts
- Existing work
- Core idea
  - Summarization of data
  - Density Peaks clustering
  - Dependency Tree
  - Formulation of problem statement
- EDMStream - Overview and Operations
- Comparison - DP Clustering vs DBSCAN
- Evaluation on datasets and Summary

Basic concepts

- **Stream data clustering** - groups streaming data on the basis of similarity
- Challenges in comparison to classical batch mode clustering:
  - Stream data arrives in high speed - how to incrementally update the clustering results?
  - Capture cluster evolution activities - how to track cluster evolution?
- **Evolution Tracking** - as the stream data evolves, track how the clusters evolve to gain insights about the clusters

Existing work

Divides the clustering process into two steps:

- Reduce processing overhead by summarizing data points in stream using summary structures e.g., Micro-clusters, grids
- Offline batch mode clustering algorithm periodically performed
    - They do not support incremental update

Additional offline cluster evolution tracking:

- MONIC, MEC: Detects cluster evolutions and tracks it
- Expensive offline clustering and offline tracking, cannot be used in real-time

Core idea

EDMStream: Stream clustering algorithm based on **Density Peak (DP)** clustering

Assumptions:

- Cluster centers are surrounded by points with lower density
- Cluster center has relatively larger distance to higher density points (other cluster centers)

1. Summarize the set of close points as a cluster-cell
2. Efficient hierarchical tree like structure (Dependency Tree) to abstract density mountains
3. Dependency Tree (DP-Tree) is updated to reflect the relationship between cluster-cells
4. Track the cluster evolution activities by tracking updates of DP-Tree

Summarization of data

Cluster-cell: Set of close points summarized as a Cluster cell

- Cluster cells as basic units instead of data points

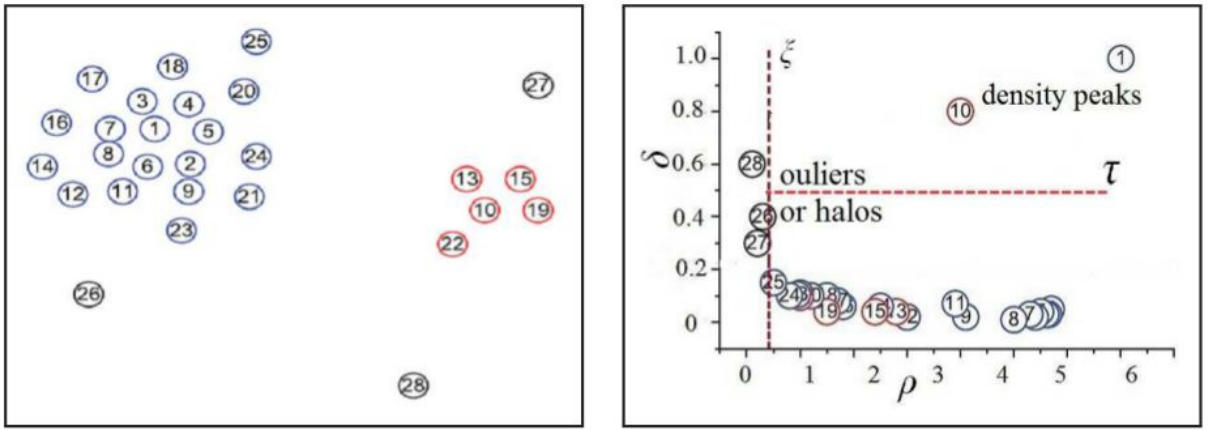
characterized by a three-tuple {*Seed, Density, Dependent distance*} at time *t*

- Seed point: Summarizes a set of points whose distance to Sc is less than or equal to a predefined radius
- Density: Summarization of all cluster-cell points' timely density
- Dependent distance: From its seed point to its nearest cluster-cell seed point with higher cluster-cell density

Density Peaks Clustering

Computes two metrics:

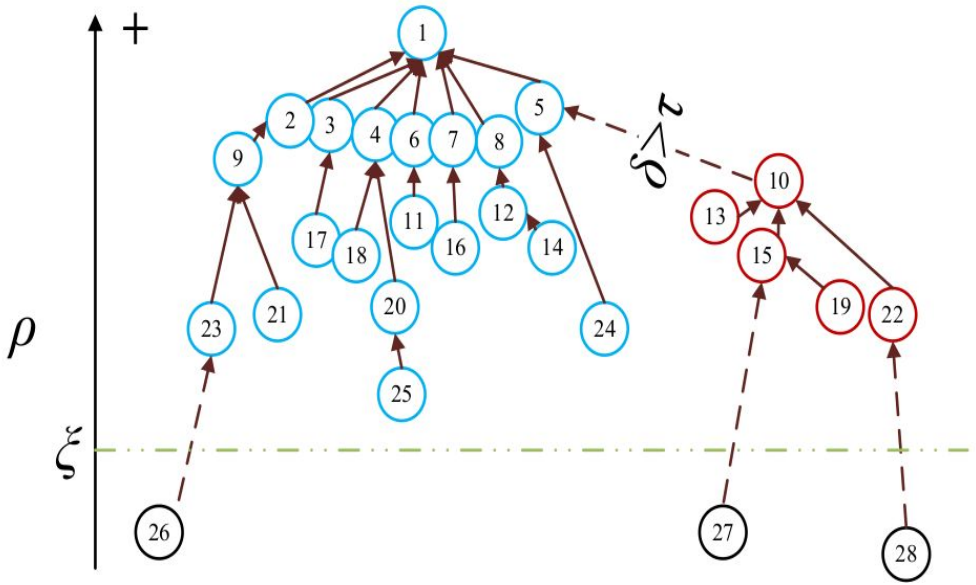
- Local density : $\rho_i = |\{p_j \big| |p_i, p_j| < d_c\}|$
- Dependent distance : $\delta_i = \min_{j:\rho_j > \rho_i} (|p_i, p_j|)$



(a) Plane view (b) Decision graph

Density peaks clustering. Figure from A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. Science, 344(6191):1492–1496, 2014.

Dependency Tree



- Point to point dependencies abstracted by a tree like structure - DP Tree (since each point is only dependent on a single point)
- If all links in a subtree are strongly dependent, then the subtree is a **strongly dependent subtree**

Illustrative example for DP-Tree. Figure from G. Shufeng, Y. Zhang, and G. Yu. Clustering Stream Data by Exploring the Evolution of Density Mountain. VLDB 2018.

Dependency Tree

- If there is no other strongly dependent subtree Tj such that Ti is a subtree of Tj, then Ti is **Maximal Strongly Dependent SubTree (MSDSubTree)**

- **Clustering based on DP-Tree**:
  - Each MSDSubTree corresponds to a cluster - Root of a MSDSubTree is the cluster center
  - Stream Clustering using DP-Tree: Find all MSDSubTrees from a dynamic DP-Tree
  - Evolution Tracking using DP-Tree: Monitoring DP-Tree changes
  - Goal: To monitor and maintain the dynamic DP-Tree and return the MSDSubTrees

Formulation of problem statement

- **Decay model**: Importance of data decayed over time, exponential time dependent weighting function
  - If no nearby points arrive, point density decreases over time
  - All points decay at same pace
- **Stream clustering**: Given a data stream and their freshness from decay model, return disjoint clusters at any time
- **Cluster Evolution:** Track five types of cluster evolution: Emerge, Disappear, Split, Merge, Adjust

EDMStream - Algorithm overview

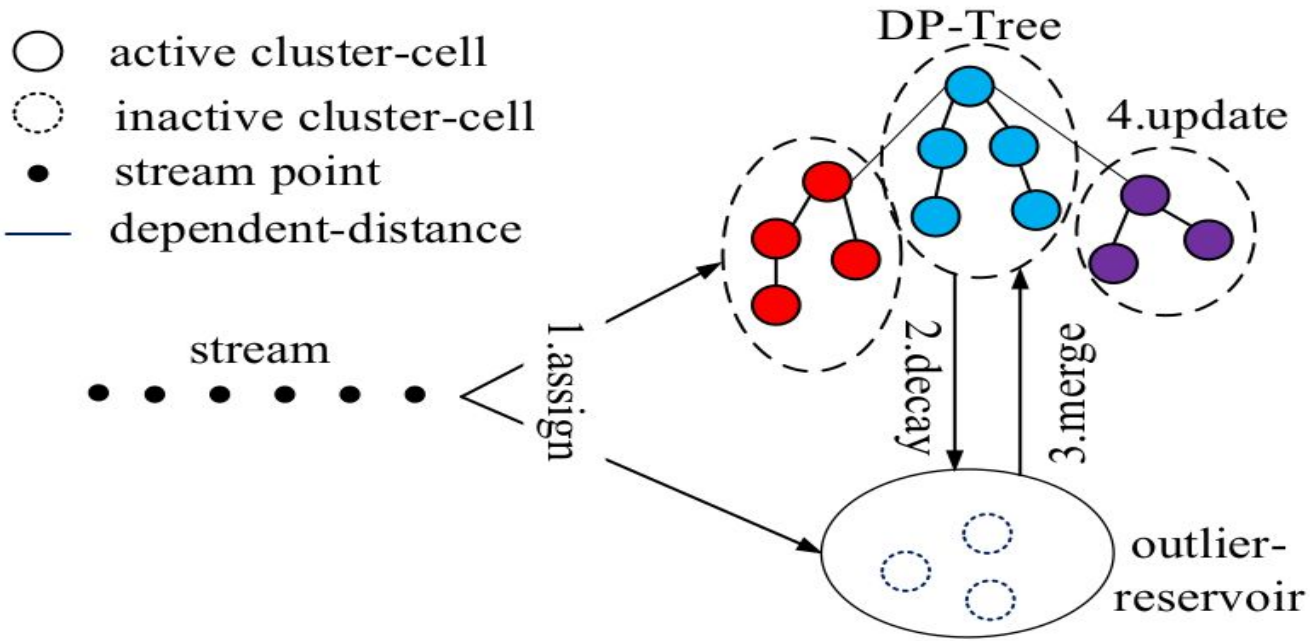
Storage structures used:

1. DP-Tree
2. Outlier Reservoir: to limit the size of DP-Tree

Key operations:

- New point assignment
- Dependencies update
- Cluster-cell emergence
- Cluster-cell decay

EDMStream - Algorithm overview



EDMStream algorithm overview. Figure from G. Shufeng, Y. Zhang, and G. Yu. Clustering Stream Data by Exploring the Evolution of Density Mountain. VLDB 2018.

EDMStream - Operations

New point assignment:

- Newly arrived point is not directly inserted to the DP-Tree but used to generate a new cluster-cell or increasing an existing cluster-cell's density
  -> Leads to DP-Tree update

- New points from data stream is added to the existing cluster-cell (in DP-Tree or outlier reservoir) or forms a new outlier cluster-cell
  - Added to a cluster-cell *c* if,
    - Distance to cluster seed *Sc* is smaller or equal to *r*
    - *Sc* is the closest seed

EDMStream - Operations

Dependencies update (DP-Tree update)

- Densities update: $\rho_c^{t_{j+1}} = a^{\lambda(t_{j+1}-t_j)}\rho_c^{t_j} + 1.$
- Dependencies update: Density changes result in dependency changes - computational cost increases
- Set of cluster-cells whose density higher than *c* at time tj: $F_c^{t_j} = \{c' | \rho_c^{t_j} < \rho_{c'}^{t_j}\}$
- Dependent cluster cell of c at time tj: $D_c^{t_j} = \arg \min_{c':c' \in F_c^{t_j}} |s_c, s_{c'}|.$

- **Update analogy**: If c' absorbs a new point, for each *c*,
  - Local density: Previously higher than or equal to but currently lower than should be updated
  - **DP-Tree point of view**: Nodes previously at higher levels than c' but now at lower levels

EDMStream - Operations

Dependencies update (DP-Tree update)

To reduce update cost, filter out the cluster-cells which does not require update

- Density Filter: If $\rho_c^{t_j} < \rho_{c'}^{t_j}$ *or* $\rho_c^{t_{j+1}} \geq \rho_{c'}^{t_{j+1}}$
  - then $D_c^{t_j} = D_c^{t_{j+1}}$.
- Triangle Inequality Filter: If $\left| |p, s_c| - |p, s_{c'}| \right| > \delta_c^{t_j}$
  - then $D_c^{t_j} = D_c^{t_{j+1}}$.

EDMStream - Operations

Cluster-cells Emergence:

- Cluster-cells:
  - In DP-Tree: **Active** cluster-cells
  - In outlier reservoir: **Inactive** cluster-cells
- Low density cluster-cells might absorb new points and become dense
- Declare a cluster-cell as active if $\rho_c^t \geq \frac{\beta \cdot v}{1-a^\lambda}$
  - β tunable parameter

EDMStream - Operations

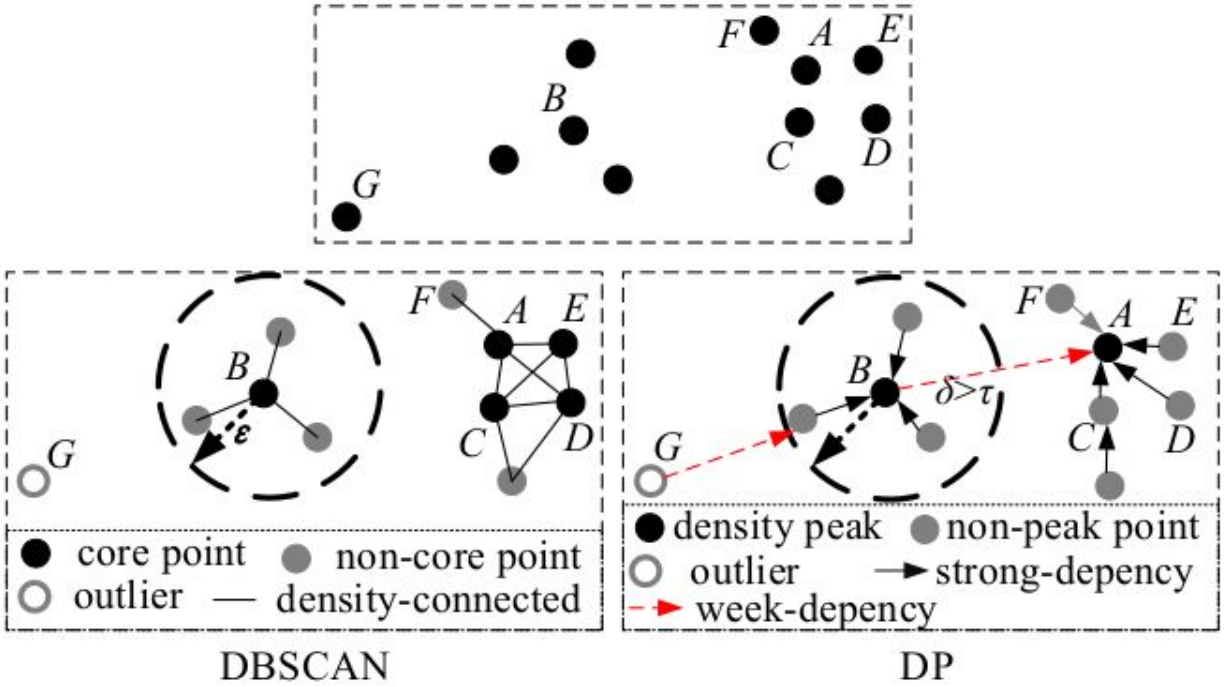
Cluster-cells Decay:

- Active cluster-cell becomes inactive if it doesn’t absorb points
- Moved from DP-Tree to outlier reservoir
- Successors of the corresponding cluster-cell should also be moved to the outlier reservoir
- Memory recycling is done to prevent overflow issues in outlier reservoirs

DP Clustering vs DBSCAN

- Clustering in DBSCAN is to find all the connected components from the density-connected graph
- Density-connected relationship between points abstracted as an undirected graph
- In DP Clustering, Density-dependent relationship between points abstracted with a tree-like structure
- Both rely on point density information

DP Clustering vs DBSCAN



DP Clustering vs DBSCAN. Figure from G. Shufeng, Y. Zhang, and G. Yu. Clustering Stream Data by Exploring the Evolution of Density Mountain. VLDB 2018.

Evaluation on datasets

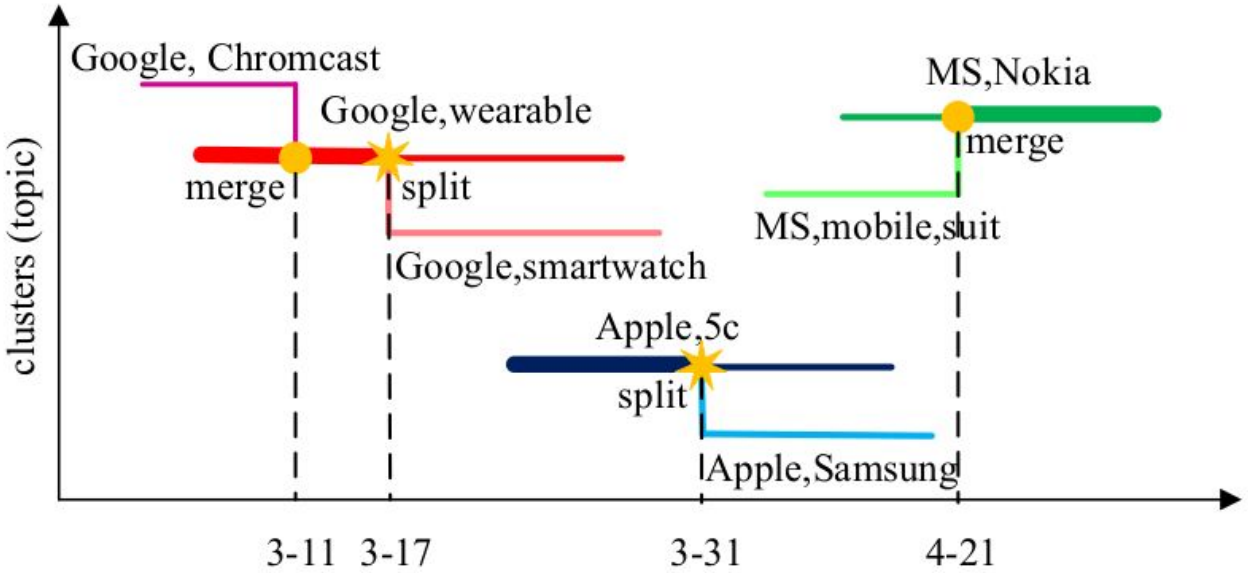
EDMStream evaluated on synthetic (SDS, HDS) and real world datasets (NADS, KDDCUP99, CoverType, PAMAP2)

- **Case study**: Monitoring Cluster Evolution in News Recommendation
- Using NADS news stream dataset, cluster evolution is tracked
- EDMStream can identify different types of cluster evolution activities
- Reflects the trends in real world

Evaluation on datasets

Case study: Monitoring Cluster Evolution in News Recommendation

Cluster evolution:

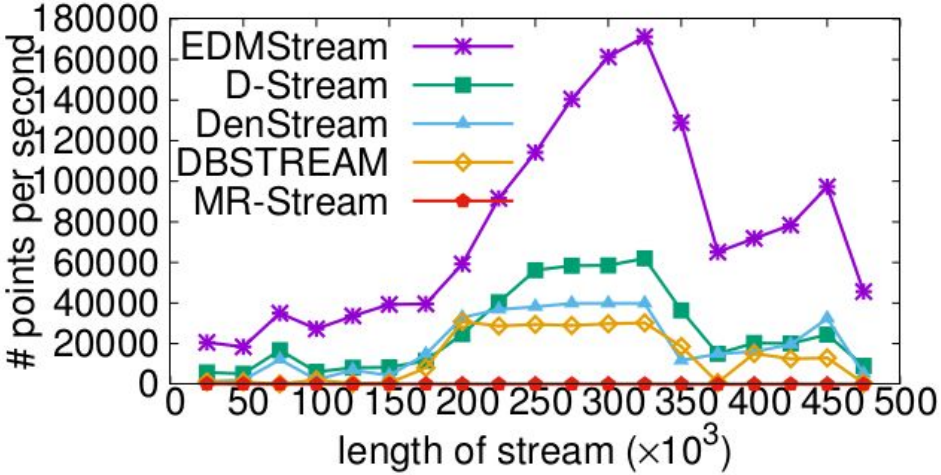


Cluster evolution acitivities. Figure from G. Shufeng, Y. Zhang, and G. Yu. Clustering Stream Data by Exploring the Evolution of Density Mountain. VLDB 2018.

Evaluation on datasets

Efficiency of EDMStream: Measured in terms of Response time, throughput with/without filtering and varying data dimensions

For KDDCUP99 dataset, comparison of throughput:



Comparison of throughput. Figure from G. Shufeng, Y. Zhang, and G. Yu. Clustering Stream Data by Exploring the Evolution of Density Mountain. VLDB 2018.

Summary

- Robust algorithm - EDMStream for stream data clustering is proposed and evaluated
- DP-Tree structure used to effectively update and monitor the cluster results
- Cluster evolution is tracked successfully
- Experimental evaluation conducted proves that the algorithm is efficient than its counterparts (D-Stream, DBStream, DenStream, MRStream)

References

1. G. Shufeng, Y. Zhang, and G. Yu. Clustering Stream Data by Exploring the Evolution of Density Mountain. VLDB 2018
2. A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. Science, 344(6191):1492–1496, 2014

Thank you for your attention!