An Empirical Comparison of Stream Clustering Algorithms

Matthias Carnein, Dennis Assenmacher, Heike Trautmann
Clustering aims to discover groups of similar objects
Traditional algorithms assume a fixed set of data
If new data arrives: Repeat the entire procedure
Stream Clustering

Stream Model

1. Data points arrive as a continuous stream
2. The size of the stream is large and potentially unbounded
3. Data points can only be evaluated once
4. The order of data points cannot be influenced
5. The distribution of the data can change over time
Outline

1. Stream Clustering
2. Setup
3. Results
4. Conclusion
Outline

1. Stream Clustering
2. Setup
3. Results
4. Conclusion
Two-Stage Clustering

Data stream

Online

'Micro-clusters'

Offline

'Reclustering'

'Macro-clusters'

[Aggarwal et al. 2003]
Distance-Based

- **BIRCH** [Zhang et al. 1996]
- **STREAM** [O'Callaghan et al. 2002]
- **CluStream** [Aggarwal et al. 2003]
- **DenStream** [Cao et al. 2006]
- **ClusTree** [Kranen et al. 2009]
- **streamKM++** [Ackermann et al. 2012]
- **BICO** [Fichtenberger et al. 2013]
- **DBSTREAM** [Hahsler et al. 2016]

Grid-Based

- **D-Stream** [Chen et al. 2007]
- **D-Stream + attraction** [Tu et al. 2009]

Challenges

- Many parameters
- No clear guidelines how to set them
DenStream

- **Clustering Feature** $CF = (n, L\bar{S}, S\bar{S})$ describes micro-cluster
- **Decay the weight of clusters over time**: $\omega(t) = 2^{-\lambda t}$
- **Core clusters can decay to outlier clusters**
D-Stream

- Maintains density of grid cells
- **Reclustering**: combine neighbouring dense cells
- Extension using attraction: only combine cells that share points at the border
DBSTREAM

- Assign points based on distance threshold
- Move clusters towards new points (competitive learning)
- Maintains shared density between micro-clusters
- **Reclustering**: Merge clusters with high shared density
Remaining Algorithms

- **BIRCH**: maintain Clustering Features in a tree structure
- **STREAM**: repeatedly apply $k$-median approximation on chunks of data
- **CluStream**: store latest snapshots for various time intervals to extract portion of entire stream
- **streamKM++**: divisive clustering approach using medoids
- **BICO**: similar tree structure as BIRCH but with medoids similar to streamKM++

Research Goal

Fair evaluation and comparison of ten popular stream clustering algorithms
Outline

1. Stream Clustering
2. Setup
3. Results
4. Conclusion
## Data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Description</th>
<th>Observations</th>
<th>Dimension</th>
<th>Cluster</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>square1</td>
<td>Spherical, 1,000</td>
<td>1,000</td>
<td>2</td>
<td>4</td>
<td>0%</td>
</tr>
<tr>
<td>spiral</td>
<td>Connected, 1,000</td>
<td>1,000</td>
<td>2</td>
<td>2</td>
<td>0%</td>
</tr>
<tr>
<td>complex9</td>
<td>Various shapes, 3,031</td>
<td>3,031</td>
<td>2</td>
<td>9</td>
<td>0%</td>
</tr>
<tr>
<td>t8.8k</td>
<td>Shapes + Noise, 8,000</td>
<td>8,000</td>
<td>2</td>
<td>8</td>
<td>4%</td>
</tr>
<tr>
<td>KDD Cup’99</td>
<td>Network Intrusion, 4,898,431</td>
<td>4,898,431</td>
<td>34</td>
<td>23</td>
<td>–</td>
</tr>
<tr>
<td>Sensor</td>
<td>Environment sensors, 2,219,803</td>
<td>2,219,803</td>
<td>4</td>
<td>54</td>
<td>–</td>
</tr>
<tr>
<td>Covertype</td>
<td>Forest type, 581,012</td>
<td>581,012</td>
<td>10</td>
<td>6</td>
<td>–</td>
</tr>
</tbody>
</table>
Evaluation

- Static Data: Evaluate clusters at the end of the stream
- Real world Data: Prequential evaluation [Cao et al. 2006]

1. Evaluate
2. Update
3. Evaluate
4. Update
5. Evaluate
6. Update

External Evaluation
- Micro-clusters: Purity
  - Proportion of points that belong to the majority class in a cluster
- Macro-clusters: Adjusted Rand Index (ARI)
  - Similarity between true groups and generated clusters
Parameter-Tuning

- Choice of parameters is important for a fair comparison
- Automatic parameter-configuration using Iterated Racing (irace) [López-Ibáñez et al. 2016]
- Search for configurations that optimize the ARI of the macro-clustering
- Upper limit on the number of micro-clusters
  - Static data: 5% of the number of observations in the stream
  - Real world data: fixed upper limit of 5000
Outline

1. Stream Clustering
2. Setup
3. Results
4. Conclusion
Quality – square1

- BIRCH
- STREAM
- CluStream
- DenStream
- D-Stream
- D-Stream + attr.
- ClusTree
- StreamKM++
- BICO
- DBSTREAM

Micro-clusters: generally high purity, inconsistency for DenStream

Macro-clusters: good results but lowest ARI for DBSTREAM

Example: Clustering result of DBSTREAM
Quality – spiral

Micro-clusters: high purity for CluStream, DBSTREAM, BICO, streamKM++

Macro-clusters: only decent results for DBSTREAM

Example: clustering result of STREAM
Quality – Sensor

- BIRCH
- STREAM
- CluStream
- DenStream
- D-Stream
- D-Stream + attr.
- ClusTree
- StreamKM++
- BICO
- DBSTREAM

Time (in 1000)

Purity

ARI

Time (in 1000)
Runtime

- Various programming languages (R, C, C++, Java)
- Different interfaces, different authors
Outline

1. Stream Clustering
2. Setup
3. Results
4. Conclusion
Conclusion

▶ Empirical comparison of 10 stream clustering algorithms
▶ **DBSTREAM** performed best but dependent on insertion order
▶ **D-Stream** good results but requires more micro-cluster
▶ **DenStream** poor results and very inconsistent

Future Work

▶ More algorithms and data sets and runtime experiments
▶ Alternative configuration methods
▶ Use results to design an improved algorithm
▶ Application to Multi-Channel Customer Relationship Management
Questions?
References I

Ackermann, Marcel R., Marcus Märtens, Christoph Raupach, Kamil Swierkot, Christiane Lammersen and Christian Sohler (2012). ‘StreamKM++: A Clustering Algorithm for Data Streams’. In: *J. Exp. Algorithmics* 17, 2.4:2.1–2.4:2.30. (Visited on 16/02/2017).

Aggarwal, Charu C., Jiawei Han, Jianyong Wang and Philip S. Yu (2003). ‘A Framework for Clustering Evolving Data Streams’. In: *Proceedings of the 29th International Conference on Very Large Data Bases*. Vol. 29. VLDB ’03. Berlin, Germany: VLDB Endowment, pp. 81–92.


References II


Hahsler, Michael and Matthew Bolaños (2016). ‘Clustering Data Streams Based on Shared Density between Micro-Clusters’. In: IEEE Transactions on Knowledge and Data Engineering 28.6, pp. 1449–1461. DOI: 10.1109/TKDE.2016.2522412.

References III


