Temporal Structure Learning for Clustering Massive Data Streams in Real-Time

Michael Hahsler & Margaret H. Dunham

Intelligent Data Analysis Group
Southern Methodist University

2011 SIAM Conference on Data Mining
April 28–30, 2011

Table of Contents

Data Stream Clustering

Algorithms for clustering data streams [5, 1, 2, 3, 6, 7, 9, 8, 10] have focused on many characteristics of stream data, e.g.,

- limited storage but potentially unbounded size of data stream,
- single pass over the data,
- store only summaries for (micro-)clusters,
- real-time processing,
- concept drift.

Data Stream Clustering

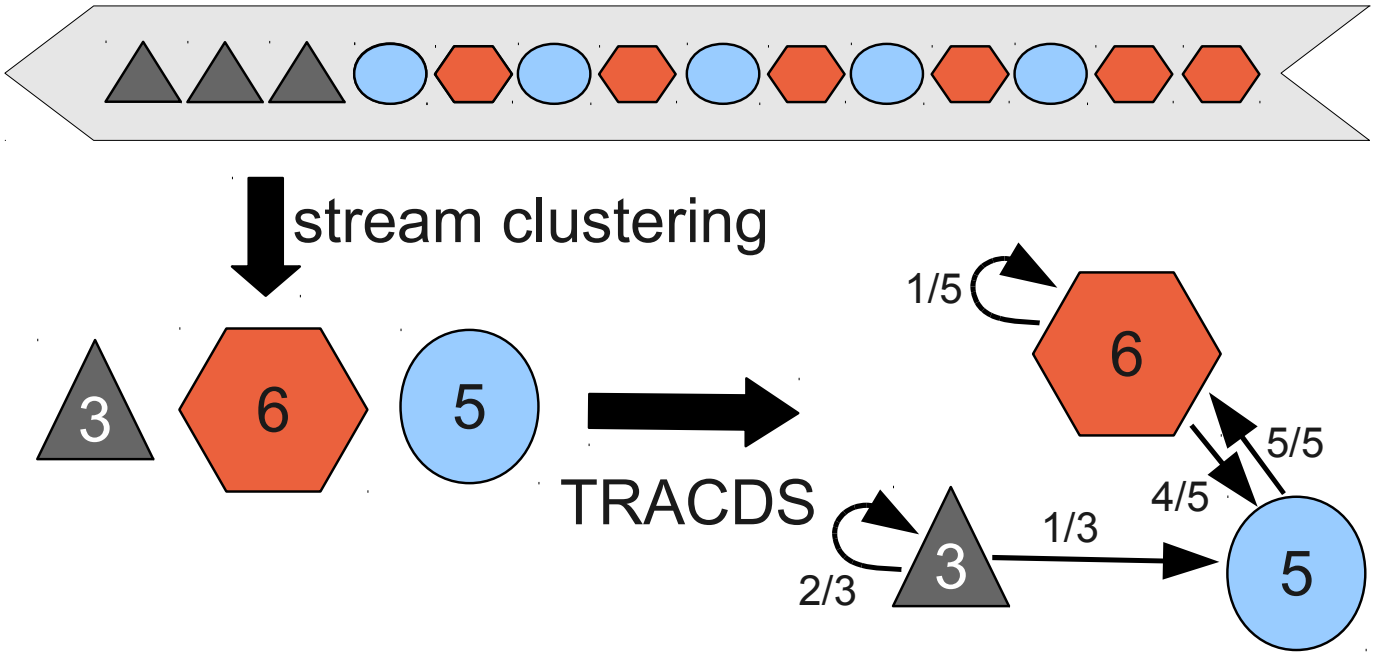
Algorithms for clustering data streams [5, 1, 2, 3, 6, 7, 9, 8, 10] have focused on many characteristics of stream data, e.g.,

- limited storage but potentially unbounded size of data stream,
- single pass over the data,
- store only summaries for (micro-)clusters,
- real-time processing,
- concept drift.

Unsolved Problem

Temporal structure is an important feature of a data stream.
This structure is *lost* during clustering!

Data Stream Clustering with Temporal Structure



Partitioning of a data stream using standard (data stream) clustering neglects the temporal aspect of the data.

Learn a temporal structure model with transition probabilities between clusters (arcs).

Table of Contents

Temporal Relationships Among Clusters for Data Streams (TRACDS)

1. Identify the set of clustering operations which is sufficient to describe state-of-the-art data stream clustering algorithms.
2. Select an appropriate temporal structure model.
3. Define a set of operations to dynamically update a temporal structure model when clustering operations are performed.

Contribution

- First attempt to model the temporal structure of massive data streams in real-time.
- Combines data stream clustering and Markov chains which are dynamically changed by a set of operations in a new and efficient way.

Data Stream Clustering Operations

Operation	*CluStream*[1]	*HPStream*[2]	*DenStream*[3]	*WSTREAM*[6]
q_{assign}	x	x	x	x
q_{create}	x	x	x	x
q_{remove}	x	x	x	x
q_{merge}	offline		offline	x
q_{fade}		x	x	x
q_{split}				

Operation	*OpticsStream*[7]	*E-Stream*[9]	*D-Stream*[8]	*MR-Stream*[10]
q_{assign}	x	x	x	x
q_{create}	x	x	x	x
q_{remove}	x	x	x	x
q_{merge}	x	offline	offline	
q_{fade}	x	x	x	x
q_{split}		x		

Temporal Structure Model: Markov Chain for Clusters

Definition

First order discrete parameter Markov Chain

$\{X_t\} = \langle X_1, X_2, X_3, \ldots \rangle$ and $\mathrm{dom}(X_t) = S = \{cl_1, cl_2, \ldots, cl_k\}$

Markov property:

$$P(X_{t+1} = s_{t+1} \mid X_t = s_t, \ldots, X_1 = s_1) = P(X_{t+1} = s_{t+1} \mid X_t = s_t)$$

where $s_1, \ldots, s_t, s_{t+1} \in S$

- Representation as a $k \times k$ transition matrix:
  $\mathbf{A} = (a_{ij})$ with $a_{ij} = P(X_{t+1} = s_j \mid X_t = s_i)$, $i, j = 1, 2, \ldots, k$
- Estimation of $\mathbf{A}$ from observed transition counts c_{ij}:
  - Maximum likelihood method: $a_{ij} = c_{ij} / \sum_{j=1}^{k} c_{ij}$
  - Laplace estimates: $a_{ij} = \frac{c_{ij}+1}{k+\sum_{j=1}^{k} c_{ij}}$
  - Bayes estimates with known prior distribution.

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1			
2	2			
3	3			
4	2			
5	3			
6	4			
7	4			
8	2			
9	3			
10	4			

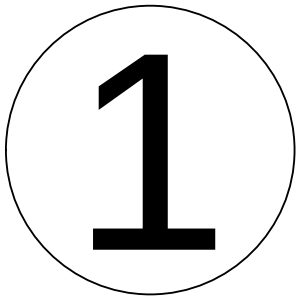
Graph Representation
Empty Clustering

Transition Count Matrix $\mathbf{C}$
0×0 *matrix*

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	 1
2	2			
3	3			
4	2			
5	3			
6	4			
7	4			
8	2			
9	3			
10	4			

Graph Representation



Transition Count Matrix $\mathbf{C}$

	1
1	0

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3			
4	2			
5	3			
6	4			
7	4			
8	2			
9	3			
10	4			

Graph Representation

Transition Count Matrix $\mathbf{C}$

	1	2
1	0	1
2	0	0

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 3×3 $c_{2,3} \leftarrow c_{2,3} + 1$	3
4	2			
5	3			
6	4			
7	4			
8	2			
9	3			
10	4			

Graph Representation

Transition Count Matrix $\mathbf{C}$

	1	2	3
1	0	1	0
2	0	0	1
3	0	0	0

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 3×3 $c_{2,3} \leftarrow c_{2,3} + 1$	3
4	2	$r_{assign\ point}$	$c_{3,2} \leftarrow c_{3,2} + 1$	2
5	3			
6	4			
7	4			
8	2			
9	3			
10	4			

Graph Representation

Transition Count Matrix $\mathbf{C}$

	1	2	3
1	0	1	0
2	0	0	1
3	0	1	0

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 3×3 $c_{2,3} \leftarrow c_{2,3} + 1$	3
4	2	$r_{assign\ point}$	$c_{3,2} \leftarrow c_{3,2} + 1$	2
5	3	$r_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
6	4			
7	4			
8	2			
9	3			
10	4			

Graph Representation

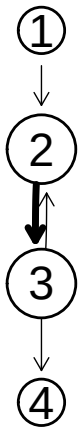
Transition Count Matrix $\mathbf{C}$

	1	2	3
1	0	1	0
2	0	0	2
3	0	1	0

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 3×3 $c_{2,3} \leftarrow c_{2,3} + 1$	3
4	2	$r_{assign\ point}$	$c_{3,2} \leftarrow c_{3,2} + 1$	2
5	3	$r_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
6	4	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 4×4 $c_{3,4} \leftarrow c_{3,4} + 1$	4
7	4			
8	2			
9	3			
10	4			

Graph Representation



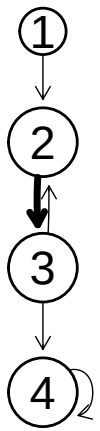
Transition Count Matrix $\mathbf{C}$

	1	2	3	4
1	0	1	0	0
2	0	0	2	0
3	0	1	0	1
4	0	0	0	0

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 3×3 $c_{2,3} \leftarrow c_{2,3} + 1$	3
4	2	$r_{assign\ point}$	$c_{3,2} \leftarrow c_{3,2} + 1$	2
5	3	$r_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
6	4	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 4×4 $c_{3,4} \leftarrow c_{3,4} + 1$	4
7	4	$r_{assign\ point}$	$c_{4,4} \leftarrow c_{4,4} + 1$	4
8	2			
9	3			
10	4			

Graph Representation



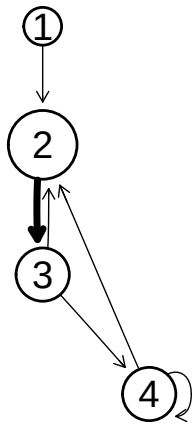
Transition Count Matrix $\mathbf{C}$

	1	2	3	4
1	0	1	0	0
2	0	0	2	0
3	0	1	0	1
4	0	0	0	1

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 3×3 $c_{2,3} \leftarrow c_{2,3} + 1$	3
4	2	$r_{assign\ point}$	$c_{3,2} \leftarrow c_{3,2} + 1$	2
5	3	$r_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
6	4	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 4×4 $c_{3,4} \leftarrow c_{3,4} + 1$	4
7	4	$r_{assign\ point}$	$c_{4,4} \leftarrow c_{4,4} + 1$	4
8	2	$r_{assign\ point}$	$c_{4,2} \leftarrow c_{4,2} + 1$	2
9	3			
10	4			

Graph Representation



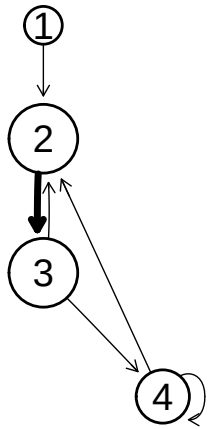
Transition Count Matrix $\mathbf{C}$

	1	2	3	4
1	0	1	0	0
2	0	0	2	0
3	0	1	0	1
4	0	1	0	1

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 3×3 $c_{2,3} \leftarrow c_{2,3} + 1$	3
4	2	$r_{assign\ point}$	$c_{3,2} \leftarrow c_{3,2} + 1$	2
5	3	$r_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
6	4	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 4×4 $c_{3,4} \leftarrow c_{3,4} + 1$	4
7	4	$r_{assign\ point}$	$c_{4,4} \leftarrow c_{4,4} + 1$	4
8	2	$r_{assign\ point}$	$c_{4,2} \leftarrow c_{4,2} + 1$	2
9	3	$r_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
10	4			

Graph Representation



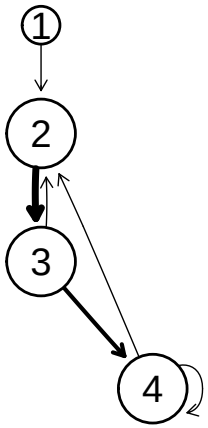
Transition Count Matrix $\mathbf{C}$

	1	2	3	4
1	0	1	0	0
2	0	0	3	0
3	0	1	0	1
4	0	1	0	1

Example: Creation of a Simple TRACDS Model

Incoming data point	Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	s_c
		initial	$\mathbf{C}$ is 0×0	ϵ
1	1	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 1×1 no manipulation	1
2	2	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 2×2 $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	3	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 3×3 $c_{2,3} \leftarrow c_{2,3} + 1$	3
4	2	$r_{assign\ point}$	$c_{3,2} \leftarrow c_{3,2} + 1$	2
5	3	$r_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
6	4	$r_{new\ cluster}$ $r_{assign\ point}$	expand $\mathbf{C}$ to 4×4 $c_{3,4} \leftarrow c_{3,4} + 1$	4
7	4	$r_{assign\ point}$	$c_{4,4} \leftarrow c_{4,4} + 1$	4
8	2	$r_{assign\ point}$	$c_{4,2} \leftarrow c_{4,2} + 1$	2
9	3	$r_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
10	4	$r_{assign\ point}$	$c_{3,4} \leftarrow c_{3,4} + 1$	4

Graph Representation



Transition Count Matrix $\mathbf{C}$

	1	2	3	4
1	0	1	0	0
2	0	0	3	0
3	0	1	0	2
4	0	1	0	1

Complexity of TRACDS

Space Complexity

- Transition count matrix $\mathbf{C}$ has size $k \times k$.
- DS clustering algorithms typically limit k.
- $\mathbf{C}$ is typically sparse and can be stored more efficiently at the expense of time complexity.
- Fading and pruning rare transitions makes matrix sparser.

Time Complexity

- Simple counting—most operations take $O(k)$ time.
- Fading and transition matrix reorganization take $O(k^2)$.
  - Minimize reorganization by using a larger matrix with unused rows/columns.
  - Postpone fading using timestamps.

Table of Contents

Evaluation Setting: Anomaly Detection

TRACDS is potentially useful for many applications. Compare TRACDS to unsupervised anomaly detection via regular DS clustering.

Simple Baseline Approach (proposed by Eskin et al [4])

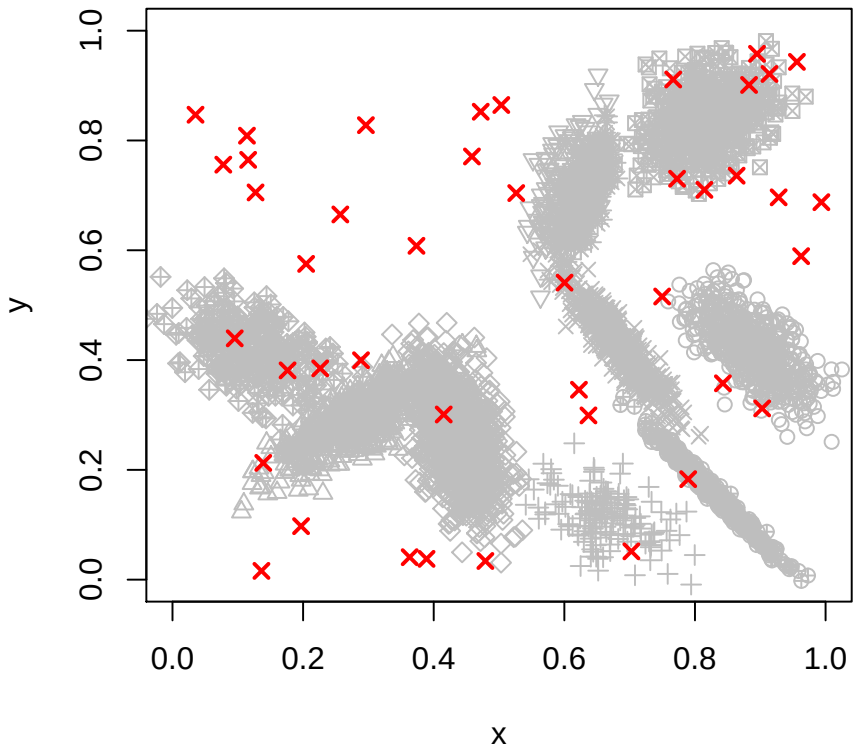
1. Start with an empty clustering
2. Assign the next data point to the closest cluster if it is within a fixed threshold from the cluster center, otherwise create a new cluster with the data point as its center.
3. Continue with step 2 for the next data point.

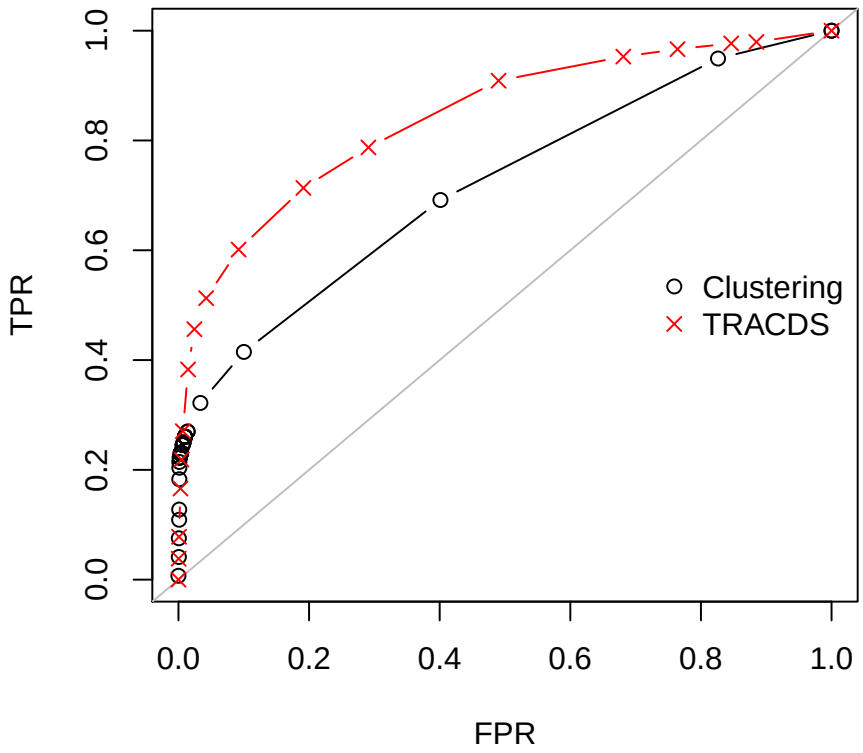
A data point is classified as an outlier if the density of the assigned cluster is low ($n_i < \delta_c$).

TRACDS Approach

- Lean a TRACDS model for the clustering above.
- A data point is classified as an outlier if the transition probability from the cluster of the previous point to the cluster of this data point is low ($a_{ij} < \delta_T$).

Synthetic Data I





$10,000$ points, $k = 10$, $d = 2$

Perfect temporal structure, slowly changing data ($p_t = 0.5$)

Anomalies: 1% uniform distribution in $[0, 1]^d$

Synthetic Data II

Table: Slowly changing data ($n = 100$ and $p_t = 0.5$).

d	p_s*	N	AUC_{clust}	AUC_{TRACDS}	Improvement
2	0.0	10,000	0.751	0.855	13.8%
2	0.2	10,000	0.753	0.836	11.0%
5	0.0	10,000	0.859	0.936	9.0%
5	0.2	10,000	0.875	0.929	6.1%
10	0.0	10,000	0.904	0.957	5.9%
10	0.2	10,000	0.897	0.942	5.0%

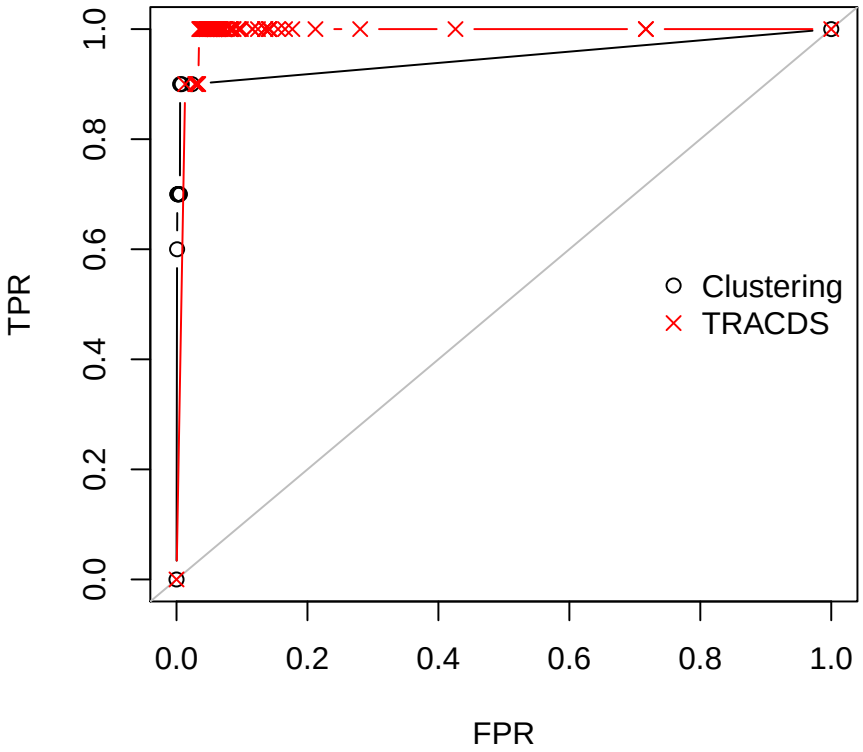
* Weaken the temporal structure by swapping two consecutive data points with probability p_s.

Table: No temporal structure ($n = N$ and $p_t = 1$).

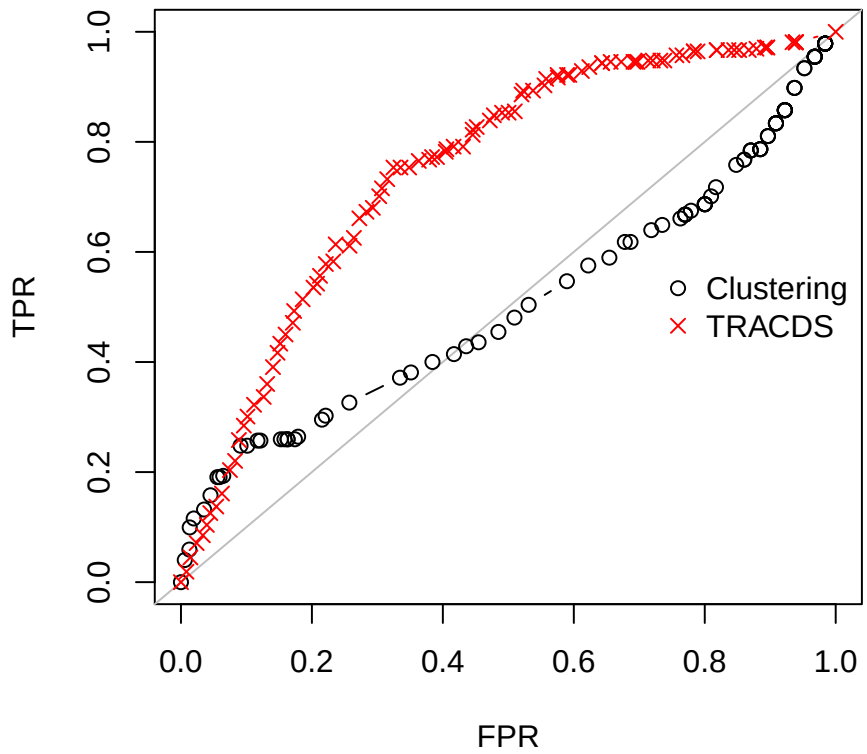
d	p_s	N	AUC_{clust}	AUC_{TRACDS}	Improvement
2	0.0	10,000	0.728	0.746	2.4%
5	0.0	10,000	0.816	0.826	1.2%
10	0.0	10,000	0.841	0.845	0.4%

Real-World Data

KDD Cup 1999
Network Intrusion Detection



16S Ribosomal RNA
Triplet Counts (Mollicutes/Alphaproteobacteria)



Name	N	outliers	d	AUC_{clust}	AUC_{TRACDS}	Improvement
KDD Cup 1999	250,000	10	38	0.893	0.972	8.9%
16S rRNA	402	43	64	0.516	0.696	34.9%

Execution Time

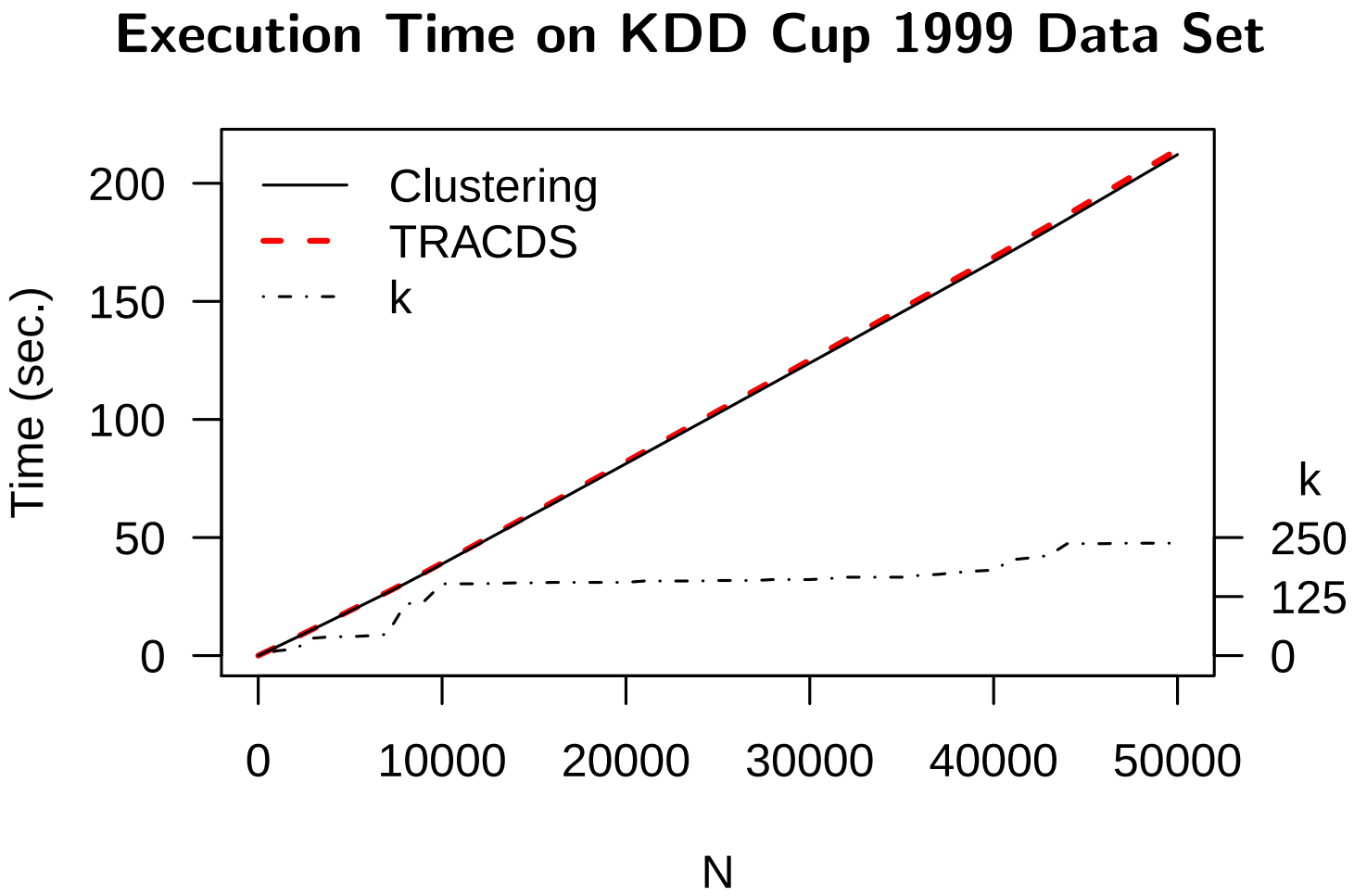


Table of Contents

Conclusion & Future Work

- First attempt to model the temporal structure for massive data streams.
- Evaluation results indicate usefulness of the approach (improvements for anomaly detection, robust against data imperfections).

Future Work

- Study the use of different data stream clustering algorithms as the base for TRACDS.
- Investigate the use of higher order Markov models (issues: space complexity, probability estimation).
- Cluster membership prediction for future data points.
- Evaluate dissimilarities between data streams by using dissimilarities between the learned temporal/sequence models.

All code is available in the R package rEMM at:
http://CRAN.R-Project.org/package=rEMM

References I

C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu.

A framework for clustering evolving data streams.

In *Proceedings of the International Conference on Very Large Data Bases (VLDB '03)*, pages 81–92, 2003.

C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu.

A framework for projected clustering of high dimensional data streams.

In *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB '04)*, pages 852–863, 2004.

F. Cao, M. Ester, W. Qian, and A. Zhou.

Density-based clustering over an evolving data stream with noise.

In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 328–339. SIAM, 2006.

E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo.

A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data.

In *Data Mining for Security Applications*. Kluwer, 2002.

S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan.

Clustering data streams: Theory and practice.

IEEE Transactions on Knowledge and Data Engineering, 15(3):515–528, 2003.

D. Tasoulis, N. Adams, and D. Hand.

Unsupervised clustering in streaming data.

In *IEEE International Workshop on Mining Evolving and Streaming Data. Sixth IEEE International Conference on Data Mining (ICDM 2006)*, pages 638–642, Dec. 2006.

D. K. Tasoulis, G. Ross, and N. M. Adams.

Visualising the cluster structure of data streams.

In *Advances in Intelligent Data Analysis VII*, Lecture Notes in Computer Science, pages 81–92. Springer, 2007.

References II

L. Tu and Y. Chen.

Stream data clustering based on grid density and attraction.
ACM Transactions on Knowledge Discovery from Data, 3(3):1–27, 2009.

K. Udommanetanakit, T. Rakthanmanon, and K. Waiyamai.

E-stream: Evolution-based technique for stream clustering.
In *ADMA '07: Proceedings of the 3rd international conference on Advanced Data Mining and Applications*, pages 605–615. Springer-Verlag, Berlin, Heidelberg, 2007.

L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang.

Density-based clustering of data streams at multiple resolutions.
ACM Transactions on Knowledge Discovery from Data, 3(3):1–28, 2009.