

---

# Temporal Relationships Among Clusters for Data Streams (TRACDS)

By Georg Eutermoser

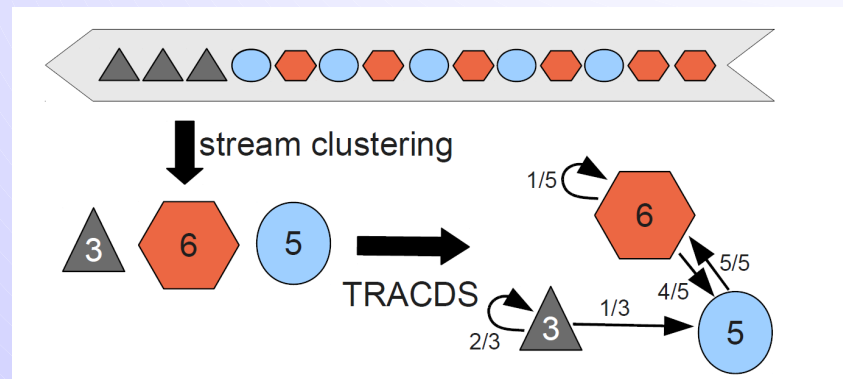
# Outline

---

- Problem
  - Problem Definition
  - Data Stream Clustering
- TRACDS
  - Markov Chains
  - TRACDS Framework
  - Implementation hints
- Experiments

# Problem

- Data stream clustering loses temporal information
- This is important in many applications
  - Anomaly detection
  - Intrusion detection in networks



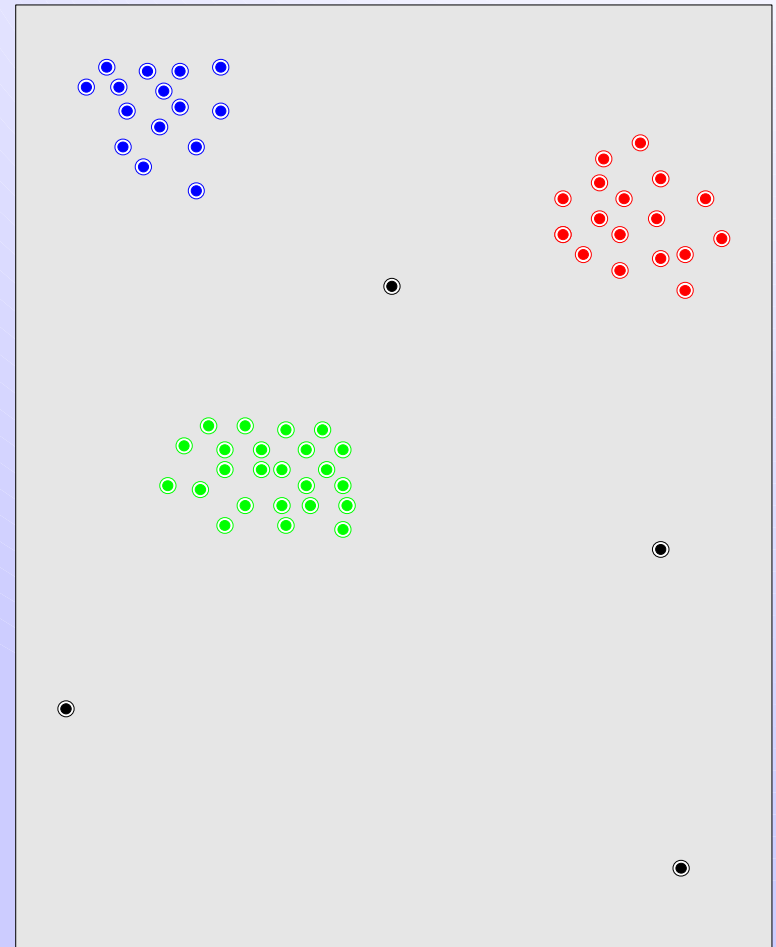
Credit cards:

- Use card and pay with it
- No usage, break for some time

... B B U B B B B U B B U U U 

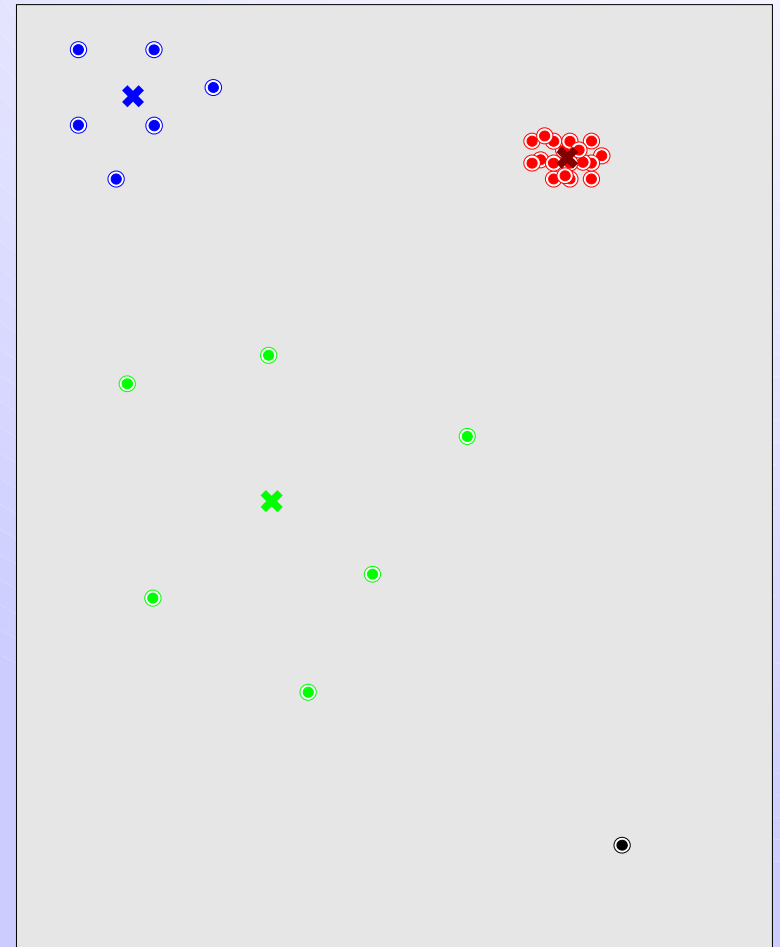
# Clustering $\zeta$

- Partitioning of data into  $k$  subsets  $C_1, \dots, C_k$
- Hard clustering
- Minimized cost function  $f_c(\zeta)$
- Points can be outliers



# Data Stream Clustering $\zeta_t$

- Clustering as defined before
- All data until  $t$
- $k$  can change over time
- Synopsis  $c_i$  for every Cluster  $C_i$ 
  - Size
  - Distribution
  - Location

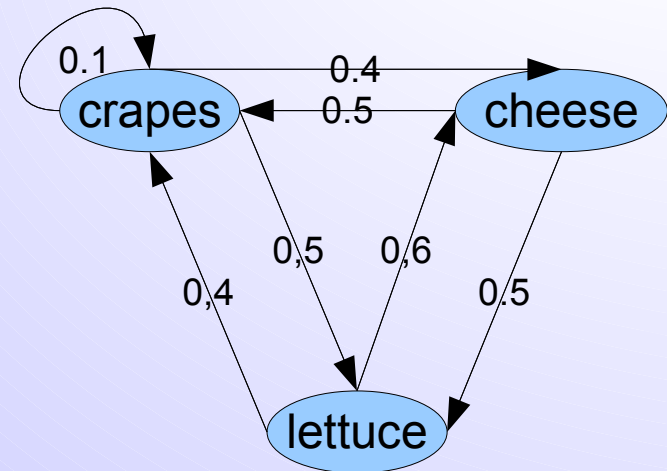


# Outline

---

- Problem
  - Problem Definition
  - Data Stream Clustering
- **TRACDS**
  - Markov Chains
  - TRACDS Framework
  - Implementation hints
- Experiments

# Markov chains



- Sequence of random variables  $\{X_T\} = \langle X_1, X_2, \dots \rangle$

- Same domain  
 $dom(X) = S = \langle s_1, s_2, s_k \rangle$

- Markov Property:  
$$P(X_{t+1} = s_{t+1} | X_t = s_t, \dots, X_1 = s_1)$$
$$= P(X_{t+1} = s_{t+1} | X_t = s_t) = a_{t,t+1}$$

- $S = \{\text{crapes, cheese, lettuce}\}$
- Possible sequences:
  - $\langle \text{crapes, lettuce, cheese} \rangle$
  - $\langle \text{crapes, crapes, cheese, lettuce} \rangle$
- Impossible sequence:
  - $\langle \text{lettuce, lettuce, crapes} \rangle$

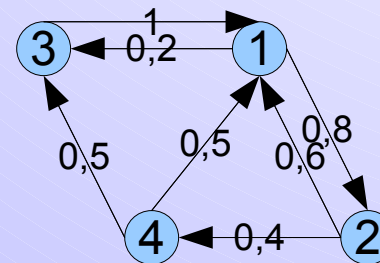
# TRACDS - Idea

- MC can be displayed as transition matrix  $A$
- Store into transition count matrix  $C$
- $A$  estimated with maximum likelihood:

$$a_{ij} = \frac{c_{ij}}{\sum_{i=0}^k c_{ij}}$$

$$C = \begin{pmatrix} 0 & 12 & 4 & 0 \\ 8 & 0 & 0 & 4 \\ 6 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 0,8 & 0,2 & 0 \\ 0,6 & 0 & 0 & 0,4 \\ 1 & 0 & 0 & 0 \\ 0,5 & 0 & 0,5 & 0 \end{pmatrix}$$



$$A' = \begin{pmatrix} 0 & \frac{3}{4} & \frac{1}{4} & 0 \\ \frac{2}{3} & 0 & 0 & \frac{1}{3} \\ 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

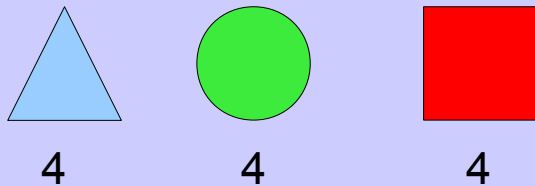


# TRACDS Definition

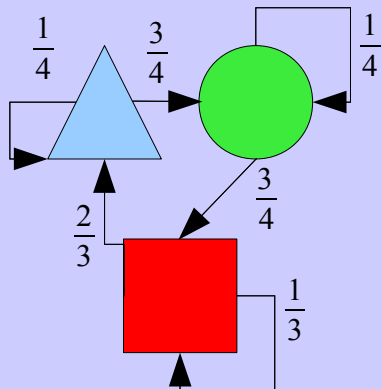
Stream:



Clustering:



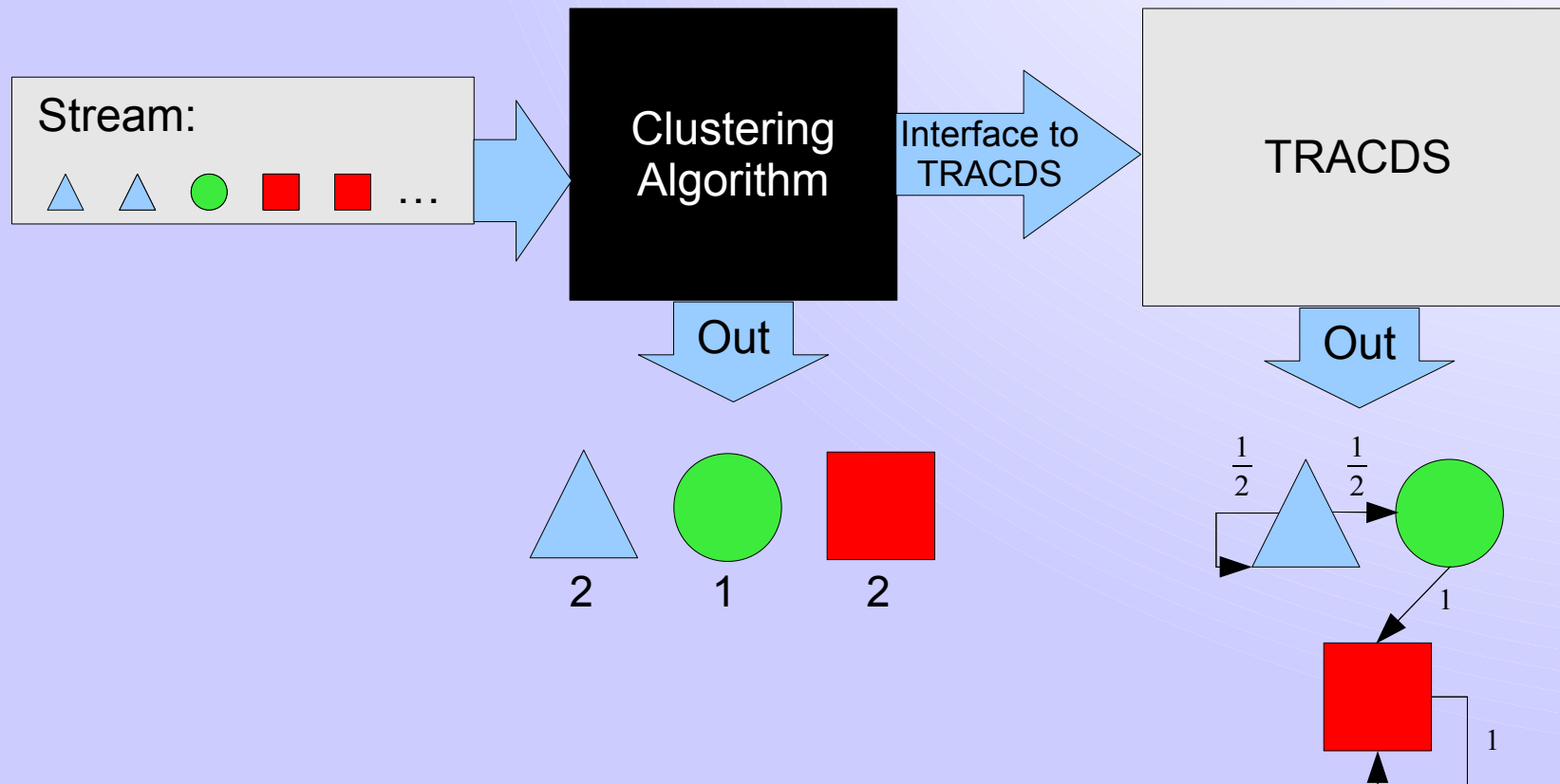
TRACDS:  $T = (\{1,2,3\}, C, 3)$



$$C = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

- Any clustering algorithm
- Clusters of the data stream as states of a MC
- Temporal information as transitions
- $T = (S, C, s_c)$ 
  - State space  $S$  with a state for each cluster
  - Transition count matrix  $C$
  - $s_c =$  current state

# TRACDS Framework



# Clustering Operations

- 6 Clustering Operations:  
(assign, create, remove, merge, fade, split)
- Appropriate TRACDS Operations  $r: T_{t+1} = r(T_t, y)$ :
  - $r_{\text{assign}}(T_t, y)$ :
    - $y = s_i$ , the state of the cluster
    - Update C:  $c_{sc,si} = c_{sc,si} + 1$
  - $r_{\text{create}}(T_t, y)$ :
    - $y$  is empty
    - Add new state to S; Enlarge C

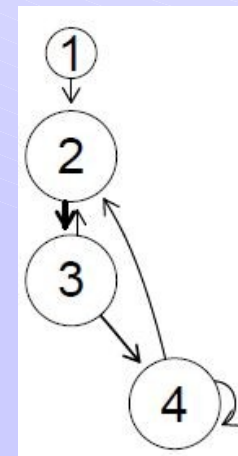
# Clustering Operations

- Appropriate TRACDS Operations (Continued):
  - $r_{\text{remove}}(T_t, y)$ :
    - $y = s_i$ , state of the removed cluster
    - Remove state from  $S$ ; Reduce  $C$
  - $r_{\text{merge}}(T_t, y)$ :
    - $y = s_i, s_j$ , states of the two merged clusters
    - merge states in  $S$ ; Reduce  $C$
  - $r_{\text{fade}}(T_t, y), r_{\text{split}}(T_t, y)$ : depends on Clustering algorithm

# Example Clustering Operations

Cluster assignment	TRACDS operation	Manipulation of $\mathbf{C}$	$s_c$
	initial	$\mathbf{C}$ is $0 \times 0$	$\epsilon$
1	$T_{new\ cluster}$ $T_{assign\ point}$	expand $\mathbf{C}$ to $1 \times 1$ no manipulation	1
2	$T_{new\ cluster}$ $T_{assign\ point}$	expand $\mathbf{C}$ to $2 \times 2$ $c_{1,2} \leftarrow c_{1,2} + 1$	2
3	$T_{new\ cluster}$ $T_{assign\ point}$	expand $\mathbf{C}$ to $3 \times 3$ $c_{2,3} \leftarrow c_{2,3} + 1$	3
2	$T_{assign\ point}$	$c_{3,2} \leftarrow c_{3,2} + 1$	2
3	$T_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
4	$T_{new\ cluster}$ $T_{assign\ point}$	expand $\mathbf{C}$ to $4 \times 4$ $c_{3,4} \leftarrow c_{3,4} + 1$	4
4	$T_{assign\ point}$	$c_{4,4} \leftarrow c_{4,4} + 1$	4
2	$T_{assign\ point}$	$c_{4,2} \leftarrow c_{4,2} + 1$	2
3	$T_{assign\ point}$	$c_{2,3} \leftarrow c_{2,3} + 1$	3
4	$T_{assign\ point}$	$c_{3,4} \leftarrow c_{3,4} + 1$	4

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



# Implementation

- TRACDS separately from Clustering algorithm
- Lightweight interface: Clustering Operations
- C as array  $k' \times k'$  with  $k' \geq k$ : space  $O(k'^2)$ 
  - assign  $O(1)$
  - merge, remove, create:  $O(k)$
  - Fading, Reordering:  $O(k^2)$
- Computational complexity:
  - Depends on amount of clustering operations
  - Negligible compared to clustering Operation

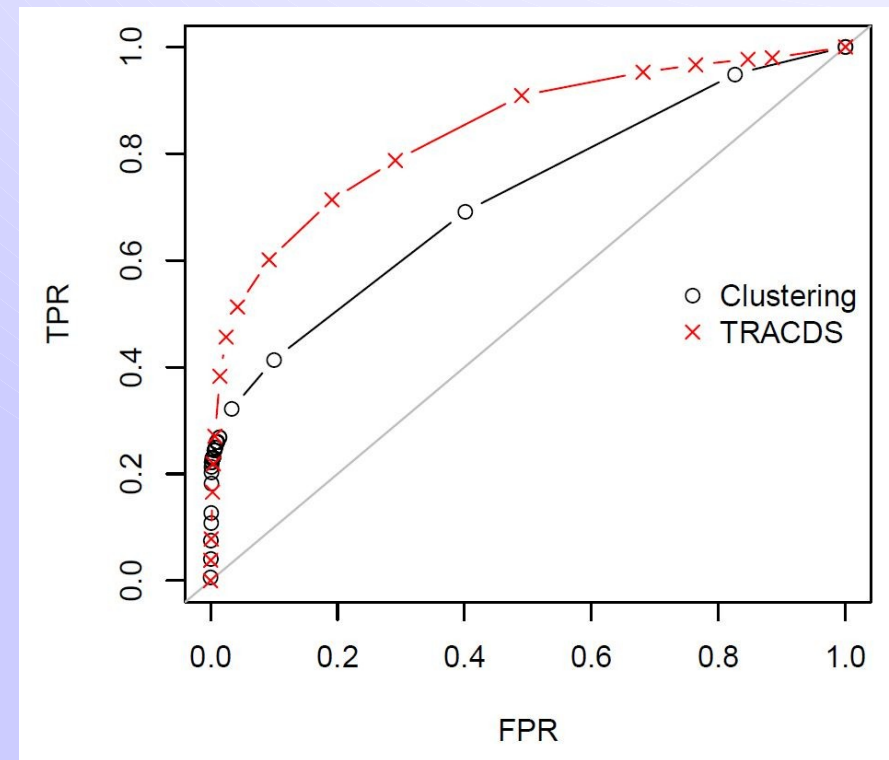
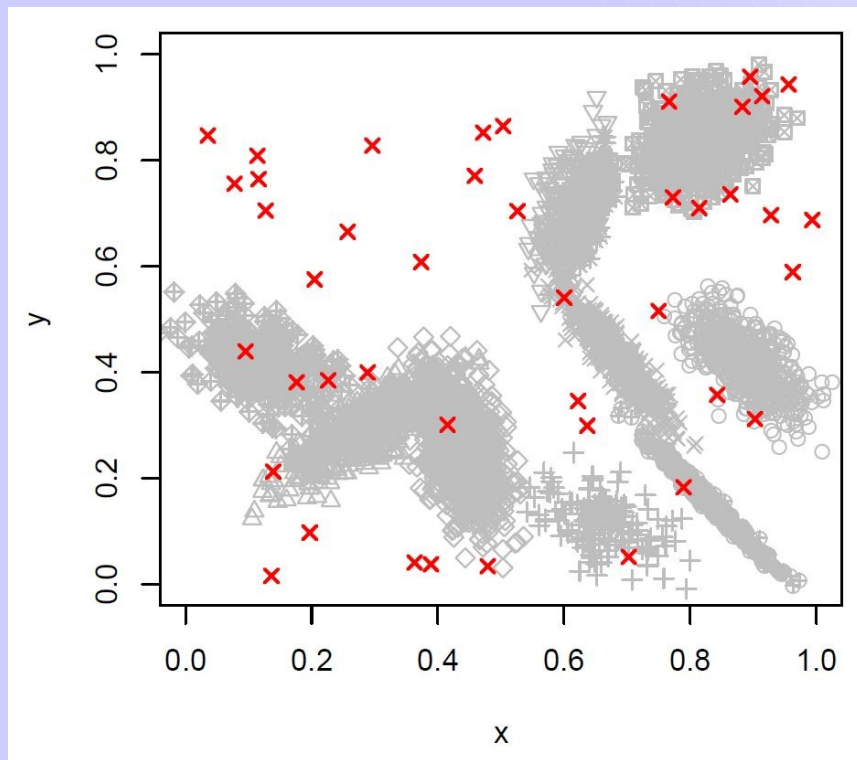
# Outline

---

- Problem
  - Problem Definition
  - Data Stream Clustering
- TRACDS
  - Markov Chains
  - TRACDS Framework
  - Implementation hints
- **Experiments**

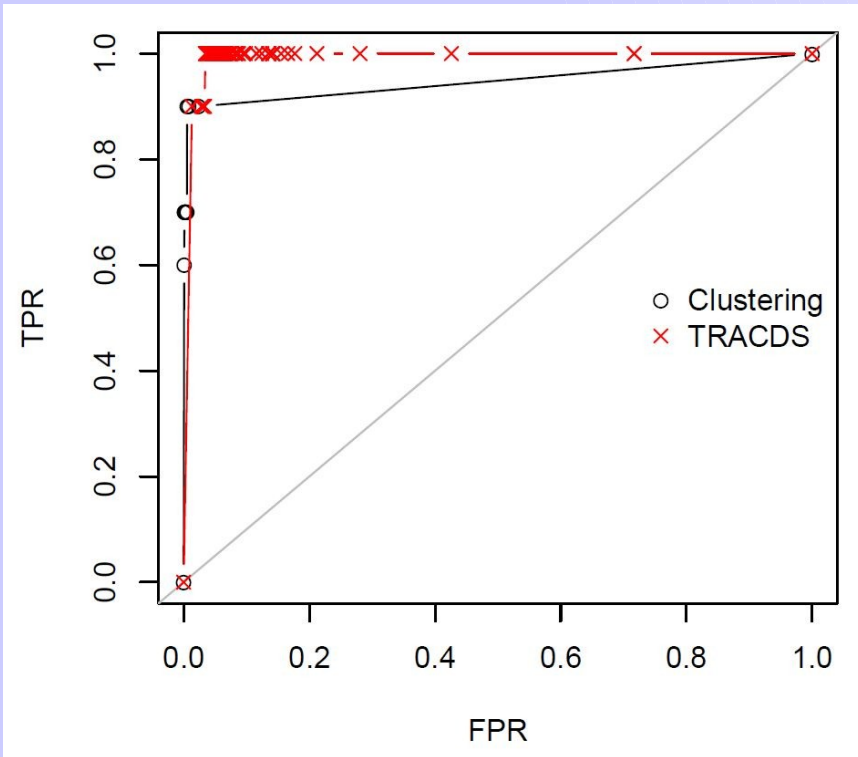
# Experiment – Artificial Data

- 2-Dimensional data stream
- Anomalies in its order, shown as X

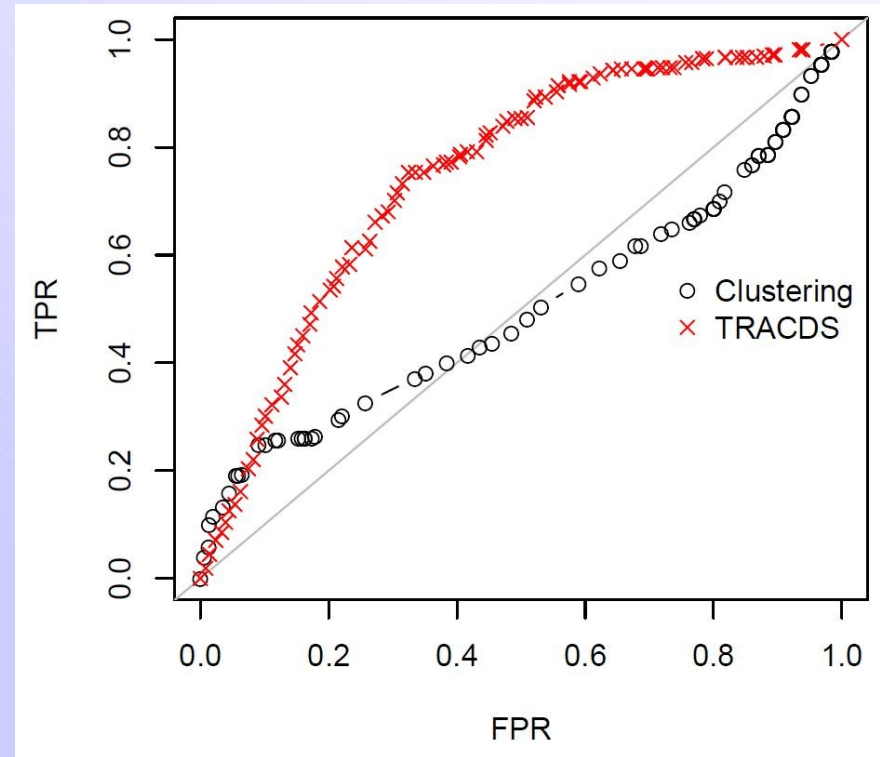




# Real World Data Sets



ROC curves for the KDD-99 data set.



Averaged ROC curves for 10 runs of the 16SrRNA data set.

# Conclusion and Future Work

---

- Advantages:
  - Temporal order stored
  - Independent of Clustering algorithm
- Disadvantage:
  - Much space for transition matrix
- Future work:
  - Better structures as model
  - Prediction of missing values in a stream
  - Better evaluation of dissimilarities



# END

Thank you for your attention.