Sequence Clustering in Data Streams

A.Marascu F. Masseglia

AxIS Project-Team
INRIA – Sophia Antipolis

Goals:

- Extract sequential patterns from data streams. Applied to: behaviour of a Web site's users.
- Identifying problems arising with this pattern extraction. Particularly the management of their history.

Framework: The SCDS method
(Sequence Clustering in Data Streams)

Data Streams: a few words…

- New elements are generated continuously.
- Data have to be considered as fast as possible.
- No blocking operator can be performed.
- Data can be examined only once.
- Memory usage is restricted.

Sequential Pattern Mining: some definitions.

- *Item*: bought by a customer
- *Transaction*: a customer + an item + a timestamp
- *Sequence*: ordered list of itemsets

- *Data sequence*: stands for the activities of a customer. Let T1, T2, …, T*n* be the transactions of C*j*, the data sequence of C*j* is:
  < itemset(T1) itemset(T2) … itemset(T*n*)>

- *Minimum support* : the minimum number of occurrences of a sequential pattern to be considered as *frequent*.

Illustration:

U1	Publications	Paper1	Paper2	Paper3
U2	Publications	Paper1	List	Paper2
U3	Research	Theme1	Theme3	Theme4
U4	Publications	List	Paper1	List
U5	Research	Theme1	Theme2	Theme3

Question : « Can we find a *behavior* that would be shared by (at least) 40% of the users recorded in the log file? »

behaviour : a series a requests performed during a navigation on the site.

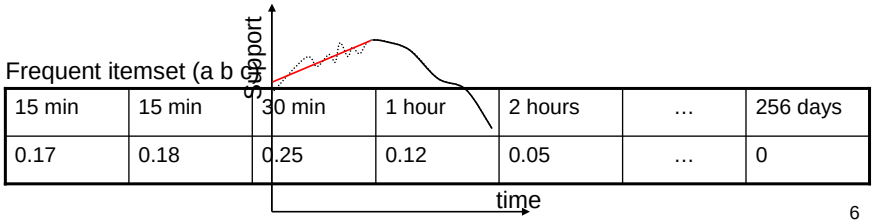
5

European Workshop on Data Stream Analysis
Caserta March 15, 2007

Extracting patterns from data streams

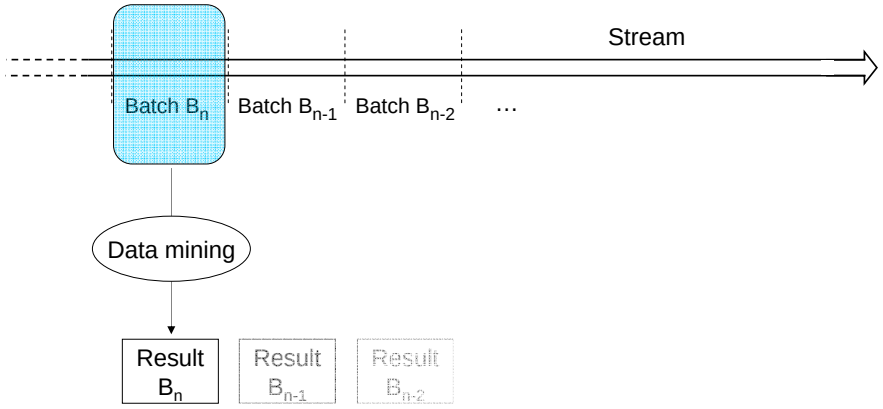
1) Satisfy the constraints of a data stream environment.
High speed algorithms.
Sampling with an estimation of the quality.
etc.

2) Managing the history of frequencies
Logarithmic Tilted Time Window (Han et al.)
Segment Tuning and Relaxation (Teng et al.)

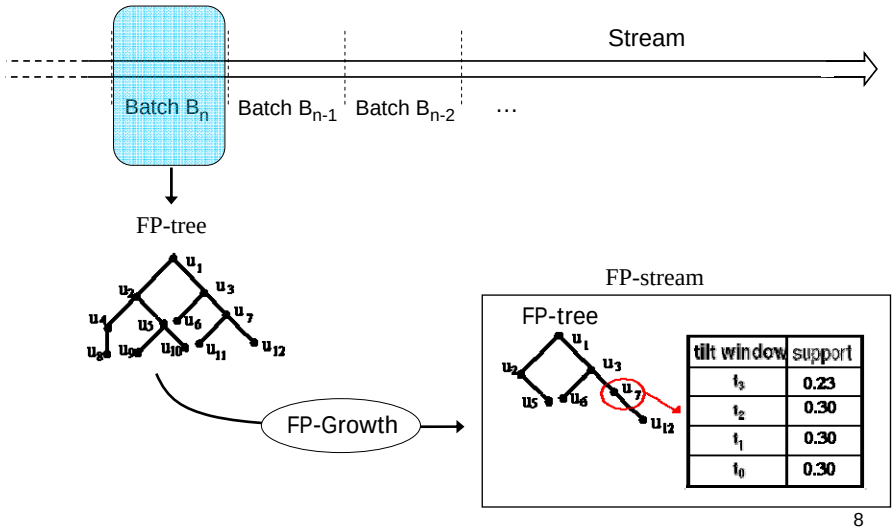


6

Overview



Example: FTPStream



Why is this such a deal to extract sequential patterns from a data stream?

A sequential pattern mining algorithm may be based on:

- Breadth-first search
- Depth-first search

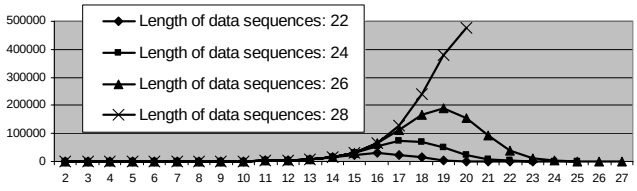
Size of the result!

- Without candidate generation
- Sampling

Size of the batch!

Why is this such a deal to extract sequential patterns from a data stream?

	T1	T2	T3	T4	…	T30
C1	1	2	1	2	…	1
C2	1	2	1	2	…	1



"We have to find the balance between the execution time and the quality of the extracted patterns."

Our proposal relies on two compromises:

1. A greedy sequence clustering algorithm.
2. A sequence alignment method applied to each cluster.

A Greedy Algorithm for clustering streaming sequences

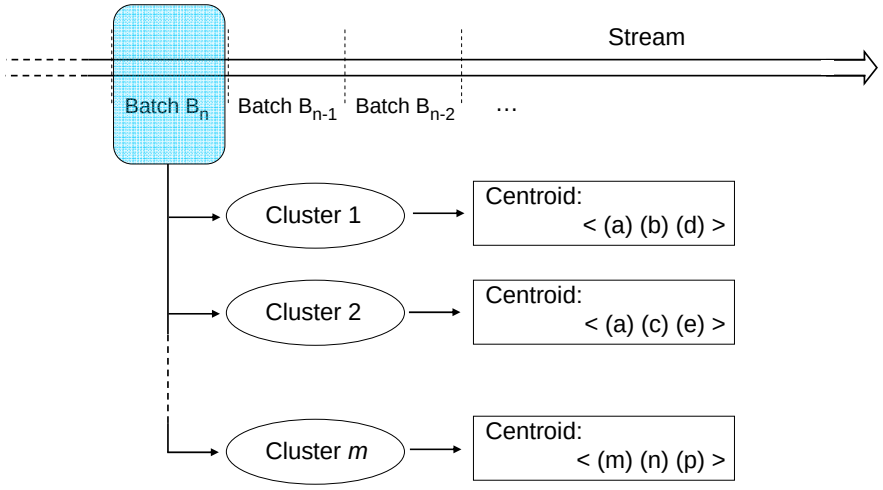
Builds the clusters on the fly.

Let ***S*** be the current sequence.
Scan ***C***, the set of clusters.
Let C_i be the cluster having the most similar centroid to ***S***,
insert (***S***, C_i);
update_centroid(C_i);

If no cluster has been found then
create_cluster (C_j, ***S***);

When "n" sequences have been processed: next batch.

Overview



Sequence alignment for each cluster

The centroid is the result of an alignment applied to the sequences of each cluster.

< (a) (b) (d) >
< (a) (c) (d) >

$\Rightarrow$

< (a) (b) (d) >
< (a) (c) (d) >

< (a:2) (b:1, c:1) (d:2) >

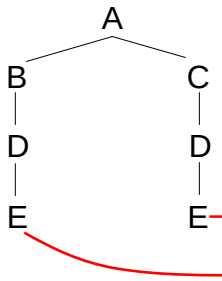
Filter *k* =1: < (a:2) (b:1, c:1) (d:2) >
Filter *k* =2: < (a:2) (d:2) >

Managing the history of the extracted patterns

- On static databases, the knowledge is stable
- On data streams, the knowledge is evolving with the stream

Ongoing work: an incremental clustering.

Motivation: The division of the stream into batches “blurs” the history of the frequent patterns.



Sequence	15 min	15 min	30 min	1h	…	256 jours
ACDE	5%	0%	4%	0%		0%
ABDE	0%	3%	0%	1%		0%
A(BC)DE	5%	3%	4%	1%		0%

We propose to perform an incremental clustering in order to maintain a coherent history.

- **Idea** : have the cluster evolving.
- **Objective** : be independent from slight variations when managing the history of extracted patterns.
- **Principle** : keep the centroid (aligned sequence) of the clusters from one batch to another.

15 min	15 min	30 min	1h	…
A(BC)D	A(BC)D	A(BC)DE	A(BC)E	…
3%	4%	2%	2%	…

17

A few questions motivating this work:

- Is data mining able to help summarizing a stream?

- What should this summary look like?

- Where does the approximation stop?

18