Summarizing Data Stream's History

Baptiste Csernel, 3rd year PhD Student.
Fabrice Clérot, FT R&D Supervisor.
Georges Hébrail, ENST Supervisor.
Plan

- Data Stream Summaries
  - Definition
  - Motivation
- Single Stream Summary: StreamSamp
  - Method Presentation
  - Method Evaluation
- Relational Stream Summary: CrossStream
  - Problematic and Motivation
  - Method Presentation
  - Method Evaluation
- Conclusion and Perspective
Definition of a data stream

Golab & Oszu (2003): A *data stream* is a real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor is it feasible to locally store a stream in its entirety.”

Structured records

Timestamps

- Physical/logical
- Explicit/implicit

<table>
<thead>
<tr>
<th>DATE-HEURE</th>
<th>Puis. A (kW)</th>
<th>Puis. R (kVAR)</th>
<th>U 1 (V)</th>
<th>I 1 (A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>16/12/2006-17:26</td>
<td>5,374</td>
<td>0,498</td>
<td>233,29</td>
<td>23</td>
</tr>
<tr>
<td>16/12/2006-17:27</td>
<td>5,388</td>
<td>0,502</td>
<td>233,74</td>
<td>23</td>
</tr>
<tr>
<td>16/12/2006-17:28</td>
<td>3,666</td>
<td>0,528</td>
<td>235,68</td>
<td>15,8</td>
</tr>
<tr>
<td>16/12/2006-17:29</td>
<td>3,52</td>
<td>0,522</td>
<td>235,02</td>
<td>15</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Definition of a Data Stream Summary

- Building in small space a summary of the stream’s content suitable for a large variety of data mining tasks.

- Being able to reconstruct from the general summary a summary concerning only a selected part of the stream’s history.
Motivation

- Due to their nature it is not possible to query or mine a stream after its passage.
- In many cases in the light of new information, new queries or analysis are needed on lost stream content.
- To be able to give answers to these queries, it is necessary to build data stream summaries.
- Summaries can also be used as light version of the stream’s content to provide approximate answers at fast speeds and with little effort.
Existing Summary Methods

- Samples
- Histograms
- Sketches
- Micro Clusters
StreamSamp: Single Stream Summary

- Method Presentation

- Method Evaluation
  - Artificial Data
  - KDD 98 Dataset
  - KDD 99 Dataset
Algorithm (1/3) (Online component)

Data Stream

Sampling at a constant rate $\alpha$

Up to $L$ samples at any given order $o$.

Samples of size $T$ and order $o$

$\ldots 1117 \ 1116 \ 1115 \quad 1114 \ 1110$

$1107 \ 1001 \ 1091 \ 1083$

$1078 \ 1069 \ 1061 \ 1053$
Algorithm (2/3) (Online component)

$L = 3$

Samples of size $T$ and order $o$.

Samples of size $T$ and order $o+1$.

Random sampling of $T/2$ elements from each parent sample.
Algorithm (3/3) **(Offline component)**

- Samples relating to the stream part to be analyzed are fused together.

- Each element receives a weight of $2^o$ where $o$ is the order of the sample.
Method Evaluation

- Method evaluation through the task of clustering.
- The scoring use is intra cluster inertia (SSQ).
- Comparison with the CluStream Algorithm.
- Evaluation on 3 Datasets:
  - Artificial Data.
  - KDD 98 Charitable Donations Dataset.
  - KDD99 Network Intrusion Dataset.
Artificial Dataset

- 100,000 elements
- 30 Variables following Gaussian distributions.
- A total of 5 equal sized macro clusters.
  - One cluster present throughout the stream.
  - Two clusters only present in the first half of the stream.
  - Two clusters only present in the second half of the stream.
KDD 98 Charitable Donation Dataset

- Information about people having made donations following a campaign of mailing.
- Very Stable dataset.

- 65,000 elements.
- 20 variables.
- Donators grouped into 5 clusters
KDD 99 Network Intrusion Dataset

- A dataset recording TCP connections records for 2 weeks of LAN network traffic managed by MIT Lincoln Labs.

- 500,000 elements
- 34 quantitative variables
- Scores computed for 5 final macro clusters.
  - 4 Different attack types and standard traffic.
Result Scores for Various Chunks of the Stream
Result Curve for an Increasing Quantity of Stream Data Starting with the First Element

![Graph showing the result curve for an increasing quantity of stream data starting with the first element. The graph compares Intra Cluster Inertia for CluStream and StreamSamp. The x-axis represents the number of elements in the horizon, and the y-axis represents the Intra Cluster Inertia. The graph shows a clear trend of increasing Intra Cluster Inertia as the number of elements in the horizon increases.]
Result Curve for an Increasing Quantity of Stream Data Starting with the Last Element

![Graph showing the result curve for an increasing quantity of stream data starting with the last element. The x-axis represents the number of elements in the horizon, while the y-axis represents intra-cluster inertia. Two lines are plotted: one for CluStream and one for StreamSamp.](image-url)
## Result Curve for Stream Processing Speed

<table>
<thead>
<tr>
<th>StreamSamp (in seconds)</th>
<th>CluStream (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>45043</td>
</tr>
<tr>
<td>200000</td>
<td>88523</td>
</tr>
<tr>
<td>300000</td>
<td>130854</td>
</tr>
<tr>
<td>400000</td>
<td>175192</td>
</tr>
<tr>
<td>500000</td>
<td>211413</td>
</tr>
</tbody>
</table>
CrossStream : Relational Stream Summary

- Relational Stream Summary
  - Motivation
  - Problematic
  - Goal
- Useful Tools
  - Bloom Filters
- CrossStream Presentation
  - System Overview
  - Entity Stream Summary
  - Relation Stream Summary
- Tests
Motivation

- Much real world data is not composed of single independent data streams.

- Most of the data shares relational links to other data streams or static databases.
  - Example: Clients <-> Service Usage <-> Services

- To properly analyze this data it is necessary to take into account all the information.
Problematic

Entity Stream E of Elements $E_i$

$E_i : (K_e, t, e1, e2, ..., ep)_i$

Relation Stream R of Elements $R_i$

$R_i : (K_e, K_f, t, r1, r2, ..., rd)_i$

Entity Stream F of Elements $F_j$

$F_j : (K_f, t, f1, f2, ..., fq)_j$

Additional Constraints:

- All Streams are insertion only.
- R speed >>> E and F speeds.
- All attributes are numerical.
- References constraints are never broken.
Goal

- Summarizing three data streams sharing a relational link with one another.
- Building separate summaries for each entity stream, and for the relation stream.
- Summarizing the information contained in the relational links between the streams.
Useful Tools

- CluStream (Aggarwal 2003)
  - Cluster Feature Vector (CFV)
  - SnapShot System

- Bloom Filters (Bloom 1970)
Cluster Feature Vector (CFV)

- **Structure:**
  
  \[(n, CF_1(t), CF_2(t), CF_1(a1), CF_2(a1), \ldots, CF_1(ad), CF_2(ad))\].

- **With**
  
  - \[CF_1(ak) = \sum (i, 1, n) (ak_i)\]
  - \[CF_2(ak) = \sum (i, 1, n) (ak_i)^2\]

- **Remark**
  
  - Time has the same role as any other variable.
CluStream (on-line part)

- **Initialization**
  - Off-line initialization of the micro clusters.

- **For each element**
  - Locate the closest micro cluster.
  - Admission test
    - If admitted, update CFV.
    - Otherwise, create a new micro cluster, and remove an outdated one.
CluStream (on-line part)

- Micro cluster removal
  - Remove an old micro cluster.
    (criteria based on the arrival date of the last elements)
  - If none is available, fuse the two closest micro cluster.
    (Update the idlist of the absorbing micro cluster)
CluStream (online part)

- **Storage**
  - Snapshot system with a distribution in $2^\circ$
  - Each snapshot contains
    - The CFV of each micro cluster.
    - The id list of each micro cluster.
SnapShot System

- The state of the system is saved at regular time intervals

- The data structure is chosen in order to allow arithmetic operation between snapshots.

- The time at which snapshots are taken is chosen in accordance to the user’s needs.
**Snapshot System:**

**Distribution example:** $2^o$

<table>
<thead>
<tr>
<th>Order $o$</th>
<th>Snapshots</th>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>69 67 65</td>
<td>$2^1$</td>
</tr>
<tr>
<td>1</td>
<td>70 66 62</td>
<td>$2^2$</td>
</tr>
<tr>
<td>2</td>
<td>68 60 52</td>
<td>$2^3$</td>
</tr>
<tr>
<td>3</td>
<td>56 40 24</td>
<td>$2^4$</td>
</tr>
<tr>
<td>4</td>
<td>48 16</td>
<td>$2^5$</td>
</tr>
<tr>
<td>5</td>
<td>64 32</td>
<td>$2^6$</td>
</tr>
</tbody>
</table>
CluStream (off-line part)

- Use the snapshot to rebuild the stream part to be analyzed. (as a set of micro clusters)

- Apply a classic classification algorithm to the resulting set of micro clusters.

- The resulting clusters represent the final clustering of the stream.
Bloom Filters (Bloom 1970) (1/3)

- Basic Principle:
  Small and fast data structure that can memorize a set of numbers.

- Supports two operations:
  - Learn a new element.
  - Tests whether an element has been previously learned or not.
Bloom Filters (Bloom 1970) (2/3)

- **Structure:**
  - A bloom filter is a simple binary word $B$ of $b$ bytes with $k$ hash functions from $\mathbb{R}$ to $[1,b]$.
  - At initialization, all the bytes are set to 0.

- Learn a new element $E$:
  - Hash $E$ with the $k$ hash functions to obtain $k$ indices in $[1,b]$.
  - Set all the bytes at the found indices to 1 in $B$. 
Test a new element $N$:
- Hash $N$ with the $k$ hash functions to obtain $k$ indices in $[1,b]$.
- If all the bytes at the found indices are at 1 in $B$, then, with high probability, $N$ was previously learned.
- Otherwise, $N$ was never learned before.

Remark:
- Bloom filters are additive.
Method Presentation

- System Overview
- Entity Summary
- Relation Summary
- Storage System
System Overview

Entity Stream E
Example: Stream of Services

Micro Clusters with attached Bloom Filters

CFV Cross Table

Entity Stream F
Example: Stream of Clients

Micro Clusters with attached Bloom Filters

Relation Stream R
Example: Stream of Usage
Upon the arrival of each new element $E_i (K_e, t, e1, e2, \ldots, ep)_i$:

- Find the closest micro cluster.
- Test for admission
  - If admitted:
    - Update the micro cluster’s CFV information.
    - Learn $K_e$ with the bloom filter attached to the micro cluster.
  - If not admitted:
    - Create a new micro cluster with $E_i$ as its seed.
    - Make room for it by fusing the two closest micro clusters.
      (which implies fusing their two Bloom filters as well)
Relation Summary

- Upon the arrival of each new element $R_i (K_e, K_f, t, r1, r2, .... rd)_i$:
  - Check all the Bloom filters summarizing $E$ to locate the one containing $K_e$. Mark its associated micro cluster $C_i$.
  - Check all the Bloom filters summarizing $F$ to locate the one containing $K_f$. Mark its associated micro cluster $C_j$.
  - If the couple $(i,j)$ is unique, add the element $R_i$ to the CFV of indices $(i,j)$ in the CFV cross table.
Entity Stream \( E_i \) of Elements \( E_i \):
\[ (K_{ei}, t, e_1, e_2, \ldots, e_p)_i \]

Entity Stream \( F_j \) of Elements \( F_j \):
\[ (K_{fj}, t, f_1, f_2, \ldots, f_q)_j \]

Relation Stream \( R_l \) of Elements \( R_l \):
\[ (K_{el}, K_{fj}, t, r_1, r_2, \ldots, r_d)_l \]

Micro Cluster 1 (CFV, ID, Filter)
Micro Cluster 2 (CFV, ID, Filter)
Micro Cluster 3 (CFV, ID, Filter)

Ei added to Micro Cluster 2
Fj added to Micro Cluster 1
Rl added to Micro Cluster (2,3)
Storage Management

- The storage system used is the same one as the one described in CluStream.
- All three streams are considered to share the same system clock.
- The information saved in each snapshot is:
  - For each entity:
    - The CFV and IdList of each micro cluster.
  - For the relation:
    - All the CFV matrix.
Method Evaluation

- Tests on Artificial Data
  - First Static Test
    - The relational stream’s data distribution **is** related to the entity stream’s data distribution.
  
  - Second Static Test
    - The relational stream’s data distribution **is not** related to the entity stream’s data distribution.
Performance Evaluation

- **Notion of Micro Cluster Purity**
  - The Purity of a micro cluster is the ratio between the number of elements in the cluster originating from the most represented macro cluster in this micro cluster and the total number of elements in the micro cluster.

- **Entity Stream Purity**
  - The average Purity of all the micro clusters summarizing the stream.

- **Relation Stream Purity**
  - The average Purity of all the boxes in the CFV cross table.
Simple Static Dataset

- Favorable relational data distribution
- Favorable temporal condition (static distribution)

Objectives
- Validate CrossStream’s concept.
- Evaluate the impact of Bloom Filter Size on performance.
Simple Relational Data Structure
Numerical Values

- **Relation Stream**
  - 360,000 elements
  - 10 numerical variables following Gaussian distributions.
  - 6 equal size macro classes with limited overlapping.

- **Entity Stream**
  - 180,000 elements
  - 10 numerical variables following Gaussian distributions.
  - 6 equal size macro classes with limited overlapping.
Average Purity Score

Bloom Filters Size in Bits

- Purity
- Entity Stream
- Relation Stream

Values:
- 432832
- 72192
- 43328
- 21696
Complex Static Dataset

- Very defavorable relational data distribution
- Favorable temporal condition (static distribution)

Objectives
- Test the impact of the number of micro classes used for entity stream summaries.
Complex Relational Data Structure
Purity Score

Number of Micro Clusters

0,05 0,1 0,15 0,2 0,25 0,3 0,35 0,4 0,45 0,5
100 150 200
Conclusion and Perspective

- A new efficient and fast method to summarize a single data stream.
- A novel method to summarize several data streams sharing relational information with one another.

Perspectives
- Extending CrossStream to more complex relational models.
- Extending CrossStream to be able to deal with deletion in the streams.
- Integrating both methods as part of a DSMS.