Roadmap

- Conceptual Overview
  - What it is
  - Applications
  - Key Challenges
  - Preliminaries and Concepts

- Two Most Classic Algorithms
  - Lossy Counting on Landmark
  - Lossy Counting on Sliding

- Exploration and Conclusion
  - Overall Analysis
  - Important Issues
1. Conceptual Overview
1.1 What is “Frequent Itemset Mining in Data Stream”?

“In a given data stream, find those itemsets which appears more than the expected threshold”

- **Frequency of an Itemset:**
  - $freq(X) = \# \text{ of transactions in window that contain } X$

- Identify all elements whose current frequency exceeds support threshold $s = 0.1\%$.

- Identify all subsets of items whose current frequency exceeds $s = 0.1\%$. 
1.2 Applications

- Web Log and Click-stream Mining
- Fraud Detection in Telecommunications data
- Network Traffic Analysis
- E-business and Stock Market Analysis
- Trend Analysis
- Sensor Networks
1.3 Key Challenges

- Memory Consumption: combinatorial explosion of itemsets
- Processing Efficiency: fast, real-time
- Single Pass: no multiple scans on stored data
- Data Representation: multi-dimensional
1.4 Preliminaries and Concepts

**Itemset**
\[ X = \{x_1, x_2, ..., x_k\} \]

**Transaction**
tuple \( T = (\text{transaction id}, \text{itemset } X) \)

**Bucket**
a sequence of transactions

**Batch**
a sequence of buckets

**Window**
an excerpt of stream

**Data Stream**
a sequence of incoming transactions
1.4 Preliminaries and Concepts

- **Landmark Window Model**
  - Data stream based on landmark windows requires handling disjoint portions of the streams, separated by landmarks.
  - Landmarks can be defined either in terms of time (e.g., on daily or weekly basis) or in terms of the number of elements observed since last landmark.

![Landmark Window Model Diagram](image)
1.4 Preliminaries and Concepts

- **Sliding Window Model**
  - Only the most recent information from the data stream are stored in a data structure.
  - The data structure is usually a first-in first-out (FIFO) structure, which considers objects from the current period of time up to a certain period in the past.

![Sliding-window model](image)
2. Two Most Classic Algorithms
Approximate Frequency Counts over Data Streams

Gurmeet Singh Manku, Rajeev Motwani

VLDB ‘02 Proceedings of the 28th International Conference on Very Large Data Bases. Pages: 346-357.
The Algorithm - Lossy Counting

❖ UpdateEntry (For each itemset X in D)
   ➢ add the frequency count of itemset X in the current batch
   ➢ if (sum of X’s frequency count and X’s error para) is smaller than current batch id, then delete this itemset from D

❖ AddEntry
   ➢ if the frequency count of an itemset X, in current batch, is at least the threshold, then add it into D, and assign its error para as (batch id - threshold)
Divide the stream into ‘Batches’
For each Itemset \( X \) in \( D \), format is a tuple \((X, \text{freq}(X), \text{err}(X))\)
For each Itemset $X$ in $D$, format is a tuple $(X, \text{freq}(X), \text{err}(X))$.

first “update” then “add”
The Guarantees ...

- All Itemsets whose true frequency exceeds $\sigma N$ are output. There are no false negatives.
- No Itemsets whose true frequency is less than $(\sigma - \varepsilon)N$ is output.
- Estimated frequencies are less than the true frequencies by at most $\varepsilon N$
A Lossy-Counting-Based Algorithm over Data Streams

Joong Hyuk Chang, Won Suk Lee

Introduction

❖ In a Data Stream...

➢ Minimum support: $S_{\text{min}} \in (0, 1)$
➢ Error parameter: $\varepsilon \in (0, S_{\text{min}})$
➢ Size of a sliding window: $w$
➢ Recently Frequent Itemset (FI)
➢ Significant Itemset
➢ Maximum possible error count for the itemset = $w \times \varepsilon$
  ■ Also called Pruning Threshold
Introduction

- In Main Memory...
  - Monitoring Lattice: each node contains an entry $(e, f, t)$
    - $e$: Corresponding itemset
    - $f$: Count of the itemset
    - $t$: Transaction where the itemset was newly inserted
  - Current Transaction List (CTL)
    - Maintains all transactions of the current window
Theorem

Given an error parameter $\varepsilon$, when $w_{\text{first}}$ denotes the TID of the first transaction of the current window, the maximum possible count $C_{\text{max}}^k(e)$ of an itemset with its entry $(e, f, t)$ is found as follows:

$$C_{\text{max}}^k(e) = \begin{cases} f & \text{if } t \leq w_{\text{first}} \\ f + \left\lfloor (t - w_{\text{first}}) \times \varepsilon \right\rfloor & \text{otherwise} \end{cases}$$
The Algorithm ...

- Two Different Phases
  - Window Initialization Phase
    - Happens when the number of transactions so far is smaller or equal to window size
    - A new transaction is appended to a CTL
    - No extracted transaction
  - Window Sliding Phase
    - Happens when CTL is full
    - A new transaction is in
    - Oldest transaction is out
The Algorithm ...

- Five Steps
  - Step 1: Appending a Transaction
  - Step 2: Count Updating and Insertion of New Itemsets
  - Step 3: Extracting a Transaction
    - Only in Window Sliding Phase
  - Step 4: Pruning Itemsets
    - Only periodically if needed
  - Step 5: Frequent Itemset Selection
    - Only when Up-to-date set of Recently FI is Requested
3. Exploration and Conclusion
## An Overall Analysis

<table>
<thead>
<tr>
<th>Representative Work</th>
<th>Window Model</th>
<th>Update Interval</th>
<th>Approximation Type</th>
<th>False Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manku &amp; Motwani</td>
<td>Landmark Count-Based</td>
<td>Per Batch</td>
<td>False-Positive</td>
<td>[ X \mid (\sigma - \varepsilon) \leq \sup(X) &lt; \sigma ]</td>
</tr>
<tr>
<td>Chang &amp; Lee</td>
<td>Sliding Count-Based</td>
<td>Per Transaction</td>
<td>False-Positive</td>
<td>[ X \mid (\sigma - \varepsilon) \leq \sup(X) &lt; \sigma ]</td>
</tr>
</tbody>
</table>
Exact Vs Approximate Mining

- **Exact Mining**
  - Keeps all itemsets’ records
  - Number of itemset is large

- **Approximate Mining**
  - Widely adopted
  - Goal: general identification rather than exact result
Load Shedding

- Approximate the Processing Rate
  - Number of transaction per unit of time machine can handle
- Characteristic of Stream
  - Average size of a transaction
  - Average size of an FI
  - Memory requirement at a particular processing rate
- How to do Load Shedding
  - Random sampling
  - Semantic drop
  - Window reduction
Reference

Thanks for Listening