Mining Frequent Items in a Stream using Flexible Windows

Toon Calders, *Eindhoven Technical University*  
Nele Dexters, Bart Goethals, *University of Antwerp*

DBDBD 2006 – Brussels – November 15
Finding frequent items in a continuous stream of items

- a b b a a c d e e b c d a a b a b a a c d b a b a a c a d a a

↑ timestamp $t=1$

→ New Frequency Measure: Max-Frequency
→ Incremental Algorithm
→ Worst-Case Analysis
New Frequency: Example

Timestamp 6
Target item a

Stream a b a a a b

Time 1  Time 6

mfreq(a, abaaab) = \max(f_{\text{freq}}(a, \text{last}(k, abaaab)))

= \max(0/1, 1/2, 2/3, 3/4, 3/5, 4/6)

= 3/4
For each item, we consider the window in which it has the highest probability:

Max-Frequency:

\[
mfreq(i, S) := \max_{k=1..|S|}(\text{freq}(i, \text{last}(k, S)))
\]
Properties

Checking all possible windows to find the maximal one: **infeasible**

**BUT**: not every point needs to be checked

Only some **special** points = the **borders**

<table>
<thead>
<tr>
<th>a a a b b b a b b a b a b a b b a b b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Unversiteit Antwerpen
How to find the borders?

Target item $a$

$a_i = \#$ occurrences of $a$ in that block

If $a_1/l_1 \geq a_2/l_2$, position $p$ is never the border again!

**Very powerful pruning criterion!**

If a position $p$ is not a border in $S$, then it neither can be a border in any extension from $S$.

Universiteit Antwerpen
Example

On timestamp 27, we have $S_{27}$:

The only borders that need to be remembered:

$$\begin{array}{|c|c|c|}
\hline
1 & 21 & 27 \\
8/20 & 3/6 & 1/1 \\
\hline
\end{array}$$
**Algorithm**

**Output**: on every timestamp $t$: $\text{Summary}(S_t)$

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$\ldots$</th>
<th>$p_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1/y_1$</td>
<td>$x_r/y_r$</td>
<td></td>
</tr>
</tbody>
</table>

$p_1 < \ldots < p_r$

$x_1/y_1 < \ldots < x_r/y_r$

Most recent

= the largest

= the current max-freq

**How**: on every timestamp, the algo adjusts the stored values based on the newly entered item

Universiteit Antwerpen
Example

Target item = a

Universiteit Antwerpen
Worst-Case Analysis

For a specific streamlength \( l \), we will identify a stream of length \( l \) that maximizes the number of borders: the Farey stream.

The idea is to have as many blocks as possible, causing as many borders as possible.

\[ \frac{a_1}{l_1} < \frac{a_2}{l_2} < \ldots < \frac{a_r}{l_r} \]
What Farey has to do with it

The challenge is for each streamlength $k = l_1 + l_2 + \ldots + l_r$ to find such an increasing array of fractions

$a_1/l_1 < a_2/l_2 < \ldots < a_r/l_r$

Solution: Farey sequences

$F_1 = 1/1$
$F_2 = 1/2, 1/1$
$F_3 = 1/3, 1/2, 2/3, 1/1$
$F_4 = 1/4, 1/3, 1/2, 2/3, 3/4, 1/1$
Farey Streams

The Farey Sequence $F_n$ introduces the Farey Stream $S_n$.

$F_5$:
$1/5 < 1/4 < 1/3 < 2/5 < 1/2 < 3/5 < 2/3 < 3/4 < 4/5 < 1/1$

$S_5$:
|abbb|abbb|abb|aabbb|ab|aaabb|aab|aaab|aaaab|a
Most Important Result

Theorem:
For streams of length $L$, the maximal number of borders is given by $N$:

$$N = \left( \frac{\pi^2 L}{2} \right)^{2/3} \frac{3}{\pi^2}$$

Remark:
Experiments show that the worst case never happens!
Further Work

- Minimum Window Length
- Focus on multiple targets in the stream
- Make the extension to itemset mining