Mining Locally Trending High Utility Itemsets

Philippe Fournier-Viger¹, Yanjun Yang¹, Jerry Chun-Wei Lin², Jaroslav Frenda³

¹Harbin Institute of Technology (Shenzhen), China
²Western Norway University of Applied Sciences (HVL), Norway
³University of Zilina, Slovakia

PAKDD 2020
Outline

• Introduction
• Definition
• LTHUI-Miner
• Experiment
• Conclusion
High Utility Itemset Mining

Input:
A transaction database

| TID | Items                  |
|-----|------------------------|
| $T_1$ | b(2),c(2),e(1)       |
| $T_2$ | b(4),c(3),d(2),e(1)  |
| $T_3$ | b(5),c(1),e(1)       |
| $T_4$ | a(2),b(10),c(2)      |
| $T_5$ | a(2),c(6),e(2)       |
| $T_6$ | b(4),c(3)            |
| $T_7$ | b(16),c(2)           |
| $T_8$ | a(2),c(6),e(2)       |
| $T_9$ | b(5),c(2),e(1)       |

A unit profit table

| Item | Unit profit |
|------|-------------|
| a    | 5$          |
| b    | 2$          |
| c    | 1$          |
| d    | 2$          |
| e    | 3$          |

Output:
High-utility itemsets (with utility $\geq$ minutil)

if $\text{minutil} = 40\$, the $HUIs$ are:

| Item set | Utility |
|----------|---------|
| $\{a, c, e\}$ | 44$     |
| $\{a, c\}$   | 44$     |
| $\{b, e\}$   | 44$     |
| $\{b, c, e\}$ | 52$     |
| $\{c, e\}$   | 44$     |
| $\{b\}$      | 92$     |
| $\{b, c\}$   | 107$    |
Trending High Utility Itemset Mining

- Time is considered

Ackman et al. (2018) proposed to discover **trending high utility itemsets**, i.e. itemsets that yield a high profit that follows an increasing or decreasing trend in the whole database.

A Transaction Database with **Time**

| TID | Items          | Time |
|-----|----------------|------|
| $T_1$ | b(2),c(2),e(1) | d1   |
| $T_2$ | b(4),c(3),d(2),e(1) | d3   |
| $T_3$ | b(5),c(1),e(1) | d4   |
| $T_4$ | a(2),b(10),c(2) | d5   |
| $T_5$ | a(2),c(6),e(2) | d6   |
| $T_6$ | b(4),c(3)      | d7   |
| $T_7$ | b(16),c(2)     | d9   |
| $T_8$ | a(2),c(6),e(2) | d10  |
| $T_9$ | b(5),c(2),e(1) | d12  |

- Transactions $T_1, T_2 \ldots T_9$ have **timestamps** $d_1, d_3 \ldots d_{12}$.
- Transactions can be simultaneous.
Limitation

- Trending High Utility Itemset Mining
  - This problem only focuses on discovering itemsets that have trends spanning over the whole database.
  - Rely on the unrealistic assumption that trends must be stable over the whole database.

- We propose a new type of pattern: locally trending high utility itemsets (LTHUI)
  - eg. \{schoolbag, pen, notebook\} yields a high profit and have an upward trend during the back-to-school shopping season rather than the whole year.
# Binned Database

A Transaction Database With **Time**

| TID | Items          | Time |
|-----|----------------|------|
| $T_1$ | b(2),c(2),e(1) | d1   |
| $T_2$ | b(4),c(3),d(2),e(1) | d3   |
| $T_3$ | b(5),c(1),e(1)  | d4   |
| $T_4$ | a(2),b(10),c(2) | d5   |
| $T_5$ | a(2),c(6),e(2)  | d6   |
| $T_6$ | b(4),c(3)      | d7   |
| $T_7$ | b(16),c(2)     | d9   |
| $T_8$ | a(2),c(6),e(2)  | d10  |
| $T_9$ | b(5),c(2),e(1)  | d12  |

A **bin** denoted as $B_{i,j}$ is the set of transactions from time $i$ to $j$, i.e. $B_{i,j} = \{ T | i \leq t(T) \leq j \}$

A **binned database** denoted as $BS$ is the sequence of consecutive non-overlapping bins of length $binlen$ in the database, eg, $BS = \langle B_{1,3}, B_{4,6}, B_{7,9}, B_{10,12} \rangle$
## Window

| TID | Items          | Time |
|-----|----------------|------|
| $T_1$ | b(2),c(2),e(1) | d1   |
| $T_2$ | b(4),c(3),d(2),e(1) | d3   |
| $T_3$ | b(5),c(1),e(1) | d4   |
| $T_4$ | a(2),b(10),c(2) | d5   |
| $T_5$ | a(2),c(6),e(2) | d6   |
| $T_6$ | b(4),c(3) | d7   |
| $T_7$ | b(16),c(2) | d9   |
| $T_8$ | a(2),c(6),e(2) | d10  |
| $T_9$ | b(5),c(2),e(1) | d12  |

A **window** denoted as $W_{[i,j]}$ is the set of bins from $i$-th bin to the $j$-th bin of the sequence $BS$, i.e. $W_{[i,j]} = \{BS[k] | i \leq k \leq j\}$. 

$$W_{[1,2]} = \{B_{1,3}, B_{4,6}\}$$

\[ B_{1,3}, B_{4,6}, B_{7,9}, B_{10,12} \]
Slope

- Let $BN_{i,j}$ be the sequence of bins that are contained in $W_{i,j}$.
- Let $AN(X)_{i,j}$ denotes the sequence of average utilities of an itemset $X$ for the bins of $BN_{i,j}$.
- Let $AT_{i,j}$ denotes the sequence of average timestamps corresponding to bins in $BN_{i,j}$.
- The slope of an itemset $X$ in a sliding window $W$ is denoted as $Slope(X, W)$, i.e. $slope(X, W) = \frac{\sum_{k=1\ldots|BN|}(AU(X)[k] - \text{avg}(AU(X))) \times (AT[k] - \text{avg}(AT))}{\sum_{t \in AT}(t - \text{avg}(AT))^2}$ iff the itemset $X$ appears in each bin of the sliding window $W$, i.e., $AU(X)[k] \neq 0$. 
Problem Definition

An itemset X is a **locally trending high utility itemset (LTHUI)** if there exists a window $W_{[i,j]}$ such that $\text{length}(W_{[i,j]}) = \text{winlen}$, $u_{[i,j]}(X) \geq \text{minutil}$ and $\text{slope}(X, W_{[i,j]}) \geq \text{minslope}$.

| TID | Items       | Time |
|-----|-------------|------|
| $T_1$ | b(2),c(2),e(1) | d1   |
| $T_2$ | b(4),c(3),d(2),e(1) | d3   |
| $T_3$ | b(5),c(1),e(1) | d4   |
| $T_4$ | a(2),b(10),c(2) | d5   |
| $T_5$ | a(2),c(6),e(2) | d6   |

$\text{slope}\{b,c\}, W_{[1,2]} = \frac{(5.67 - 8.33) \times (2 - 3.5) + (11 - 8.33) \times (5 - 3.5)}{(2 - 3.5)^2 + (5 - 3.5)^2} = 1.78 > 0.15$

**e.g.** for binlen = 3, winlen = 6, minutil = 20 and minslope = 0.15, then $\{b,c\}$ is a LTHUI.
Problem Definition

For an itemset $X$, a window $W_{i,j}$ is a **trending high utility period (THUP)** if for each window $W_{k,l} \subseteq W_{i,j}$ where $\text{length}(W_{k,l}) = \text{winlen}$, $u(X, W_{k,l}) \geq \text{minutil}$ and $\text{slope}(X, W_{k,l}) \geq \text{minslope}$.

| TID | Items       | Time |
|-----|-------------|------|
| $T_1$ | b(2),c(2),e(1) | d1   |
| $T_2$ | b(4),c(3),d(2),e(1) | d3   |
| $T_3$ | b(5),c(1),e(1)   | d4   |
| $T_4$ | a(2),b(10),c(2) | d5   |
| $T_5$ | a(2),c(6),e(2)   | d6   |
| $T_6$ | b(4),c(3)        | d7   |
| $T_7$ | b(16),c(2)       | d9   |

\[u_{[1,2]}(\{b, c\}) = 50 \geq 20\]
\[\text{slope}(\{b, c\}, W_{[1,2]}) = 1.78 \geq 0.15\]
\[u_{[2,3]}(\{b, c\}) = 78 \geq 20\]
\[\text{slope}(\{b, c\}, W_{[2,3]}) = 1.33 \geq 0.15\]

**e.g.** for $\text{binlen} = 3$, $\text{winlen} = 6$, $\text{minutil} = 20$ and $\text{minslope} = 0.15$, $W_{[1,3]}$ is a THUP of $\{b, c\}$
Problem Definition

The problem of **Locally Trending High Utility Itemset Mining (LTHUIM)** is to find all **Locally Trending High Utility Itemsets (LTHUIs)**, and their maximum **Trending High Utility Periods (THUPs)** given parameters $binlen$, $winlen$, $minutil$, and $minslope$.

◆ **For example**, given the database as mentioned before, and set the **parameters** $binlen=3$, $winlen=6$, $minutil=20$ and $minslope=0.15$

| 3 LTHUIs can be found |
|------------------------|
| $\{b\}: [d_1, d_9]$   |
| $\{b, c\}: [d_1, d_9]$|
| $\{c, e\}: [d_1, d_9]$|
LTHUI-Miner
The LTHUI-Miner Algorithm

- Based on HUI-Miner, find larger itemsets with **depth-first search**.
- Create a vertical structure named **TU-List** for each itemset.

**TU-List of \{b\}**

| tid | iutil | rutil |
|-----|-------|-------|
| T₁  | 4     | 2     |
| T₂  | 8     | 3     |
| T₃  | 10    | 1     |
| T₄  | 20    | 2     |
| T₆  | 8     | 3     |
| T₇  | 32    | 2     |
| T₉  | 10    | 2     |

**maximum trending high utility periods**

| bin | binUtils | binRutils |
|-----|----------|-----------|
| B₁,₃ | 12       | 5         |
| B₄,₆ | 30       | 3         |
| B₇,₉ | 40       | 5         |
| B₁₀,₁₂ | 10      | 2         |

**promising LTHUI periods**
Construction of TU-List

- The Utility-list of a **single item** can be constructed by scanning the database, and others can be obtained by joining their child itemset’s Utility-lists.

| Utility-list \{b\} | Utility-list \{c\} | Utility-list \{b, c\} |
|-------------------|-------------------|-------------------|
| **tid** | **iutil** | **rutil** | **tid** | **iutil** | **rutil** | **tid** | **iutil** | **rutil** |
| $T_1$ | 4 | 2 | $T_1$ | 2 | 0 | $T_1$ | 6 | 0 |
| $T_2$ | 8 | 3 | $T_2$ | 3 | 0 | $T_2$ | 11 | 0 |
| $T_3$ | 10 | 1 | $T_3$ | 1 | 0 | $T_3$ | 11 | 0 |
| $T_4$ | 20 | 2 | $T_4$ | 2 | 0 | $T_4$ | 22 | 0 |
| $T_6$ | 8 | 3 | $T_5$ | 6 | 0 | $T_6$ | 11 | 0 |
| $T_7$ | 32 | 2 | $T_6$ | 3 | 0 | $T_7$ | 34 | 0 |
| $T_9$ | 10 | 2 | $T_7$ | 2 | 0 | $T_9$ | 12 | 0 |
Construction of TU-List

The *binUtils* and *binRutils* can be constructed by scanning the Utility-list.

| tid | iutil | rutil |
|-----|-------|-------|
| $T_1$ | 4 | 2 |
| $T_2$ | 8 | 3 |
| $T_3$ | 10 | 1 |
| $T_4$ | 20 | 2 |
| $T_6$ | 8 | 3 |
| $T_7$ | 32 | 2 |
| $T_9$ | 10 | 2 |

| $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $d_1$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ | $d_7$ | $d_9$ | $d_{10}$ | $d_{12}$ |

| $\text{binUtils}$ | $\text{binRutils}$ |
|-------------------|-------------------|
| $B_{1,3}$          | 12                |
|                   | 5                 |
| $B_{4,6}$          | 30                |
|                   | 3                 |
| $B_{7,9}$          | 40                |
|                   | 5                 |
| $B_{10,12}$        | 10                |
|                   | 2                 |
Construction of TU-List

- Consider that $a < e < b < c$, $binlen = 3$, $winlen = 6$, $minutil = 20$ and $minslope = 0.15$, using sliding window to get periods information.

|        | $binUtils$ | $binRutils$ |
|--------|------------|-------------|
| $B_{1,3}$ | 12         | 5           |
| $B_{4,6}$ | 30         | 3           |
| $B_{7,9}$ | 40         | 5           |
| $B_{10,12}$ | 10        | 2           |

- $sumIUtils = 42$
- $sumRUtils = 8$
- upperbound is 50
- slope = 0.67

- $sumIUtils = 70$
- $sumRUtils = 8$
- upperbound is 78
- slope = 0.37

- $sumIUtils = 50$
- $sumRUtils = 7$
- upperbound is 57
- slope = $-1.11$
Optimizations

- **Pruning a low-TWU item in a database**
  - Remove an item $i$, if for any window $W_{i,j}$ of $\text{winlen}$, $TWU(i) < \text{minutil}$.

- **Pruning an unpromising itemset using its remaining utility in a database**
  - Remove an itemset $X$ and its transitive extensions, if $u(X) + \text{reu}(X) < \text{minutil}$.

- **Pruning an unpromising itemset using its remaining utility in a sliding window**
  - Remove an itemset $X$ and its transitive extensions in a sliding window $W$, if $u(X, W) + \text{reu}(X, W) < \text{minutil}$
Experiment
Experimental Evaluation

| Dataset   | Trans count | Item count | Average length |
|-----------|-------------|------------|----------------|
| retail    | 88,162      | 16,470     | 10.30          |
| foodmart  | 4141        | 1559       | 4.40           |

- We compared the execution time of the algorithm with and without the optimizations.
- Experiments were done by varying the `minutil` and `minslope` parameters to see their influence on runtime and pattern count.
- Java, Windows 10, 64 GB RAM, Intel Xeon E3-1270 v5
Experimental Evaluation

- In some cases, the optimized algorithm is over 150 times faster than the non-optimized algorithm.

- It is observed that as minutil is decreased, runtime increases.

- It is observed that as minslope increases, the number of patterns decreases.

- Some patterns having a strong trend were found. E.g., on retail and foodmart dataset, 179 and 13 patterns have slope values greater than 1.1 and 0.6 respectively.
Conclusion
Conclusion

- A new type of patterns named **Locally Trending High Utility Itemset**;
- A new algorithm with three optimizations;
- Results:
  - the optimizations can reduce running time over 150 times in some cases;
  - some patterns having a strong trend were found.

Open source Java data mining software, 150 algorithms
http://www.phillippe-fournier-viger.com/spmf/
Q & A