Mining local and peak high utility itemsets

Yimin Zhang
Philippe Fournier-Viger
Jerry Chun-Wei Lin
Hamido Fujita
Yun Sing Koh

Fournier-Viger, P., Zhang, Y., Lin, J. C.W., Fujita, H., Koh, Y.S. (2019). Mining Local and Peak High Utility Itemsets. Information Sciences, Elsevier, 481: 344-367.

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(2),c(2),e(1)
T_4	a(2),b(10),c(2),d(10),e(2)
T_5	a(2),c(6),e(2)
T_6	b(4),c(3),e(1)
T_7	a(2),c(2),d(2)
T_8	a(2),c(6),e(2)

a unit profit table

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

a $minutil$ threshold

Output:

High-utility itemsets (with utility≥ $minutil$)

if $minutil = 60\$,$ the $HUIs$ are:

$\{b, e\}$: 62$	$\{a, c, e\}$: 62$
$\{b, d, e\}$: 78$	$\{b, c, d, e\}$: 85$
$\{b, c, e\}$: 74$	

How to calculate the utility?

TID	Items
T_1	b(2),c(2),e(1)
T_2	**b(4),c(3),d(2)**,e(1)
T_3	b(2),c(2),e(1)
T_4	a(2),**b(10),c(2),d(10)**,e(2)
T_5	a(2),c(6),e(2)
T_6	b(4),c(3),e(1)
T_7	a(2),c(2),d(2)
T_8	a(2),c(6),e(2)

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

The utility of the itemset $\{b, c, d\}$ is calculated as follows:

$$u(\{b,c,d\}) = \underbrace{(4 \times 2) + (3 \times 1) + (2 \times 2)}_{\text{utility in transaction } T_2} + \underbrace{(10 \times 2) + (2 \times 1) + (10 \times 2)}_{\text{utility in transaction } T_4} = 57$$

Previous Work

- **Several algorithms:**
  - Two-Phase (PAKDD 2005)
  - IHUP (TKDE, 2010),
  - UP-Growth (KDD 2011),
  - HUI-Miner (CIKM 2012),
  - FHM (ISMIS 2014)
  - EFIM (KAIS 2017)
  - mHUIMiner (PAKDD 2017)
- **Key idea:**

  Calculate an upper-bound on the utility of itemsets (**e.g.** the TWU) that is anti-monotonic to be able to prune the search space.

$$u(\{b, c, d\}) = u(\{b, c, d\}, T_2) + u(\{b, c, d)\}, T_4)$$

$$TWU(\{b, c, d\}) = u(T_2) + u(T_4)$$

Limitation

- **High utility itemset mining**
  - ➢is **useful** for discovering profitable itemsets in a **whole database**
  - ➢but it ignores the time **when** transactions were made
  - ➢and it fails to find itemsets that have a high utility in **some time periods**

- We propose a new pattern type:

  Local High Utility Itmeset (LHUI)

  e.g.$\{schoolbag, pen, notebook\}$ yields a high profit during the back-to-school shopping season, while not being a HUI in the whole year.

Database with time, Window

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(2),c(2),e(1)	d3
T_4	a(2),b(10),c(2),d(10),e(2)	d5
T_5	a(2),c(6),e(2)	d6
T_6	b(4),c(3),e(1)	d7
T_7	a(2),c(2),d(2)	d9
T_8	a(2),c(6),e(2)	d10

$W_{1,5} = \{T_1, T_2, T_3, T_4\}$

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

- Transactions $T_1, T_2 \ \ldots T_8$ have **timestamps** $d_1, d_3, \ldots d_{10}$.
- Transactions can be simultaneous.
- A **window** denoted as $W_{i,j}$ is the set of transactions from time i to j, i.e. $W_{i,j} = \{T | i \leq t(T) \leq j\}$

Problem Definition

- An itemset X is a **local high utility itemset (LHUI)** if there exists a window $W_{i,j}$ such that $length(W_{i,j}) = minLength$ and $u_{i,j}(X) > lMinutil$

TID	Items	timestamp
T_1	b(4),c(2),e(3)	d_1
T_2	b(8),c(3),d(4),e(3)	d_3
T_3	b(4),c(2),e(3)	d_3
T_4	a(10),b(20),c(2),d(20),e(6)	d_5

Theorem. If $minutil = lMinutil \times \lceil W_D / minLength \rceil$, then $HUIs \subseteq LHUIs$.

$$u_{d_1,d_3}(\{b,c\}) = 6 + 11 + 6 = 23 > 20$$

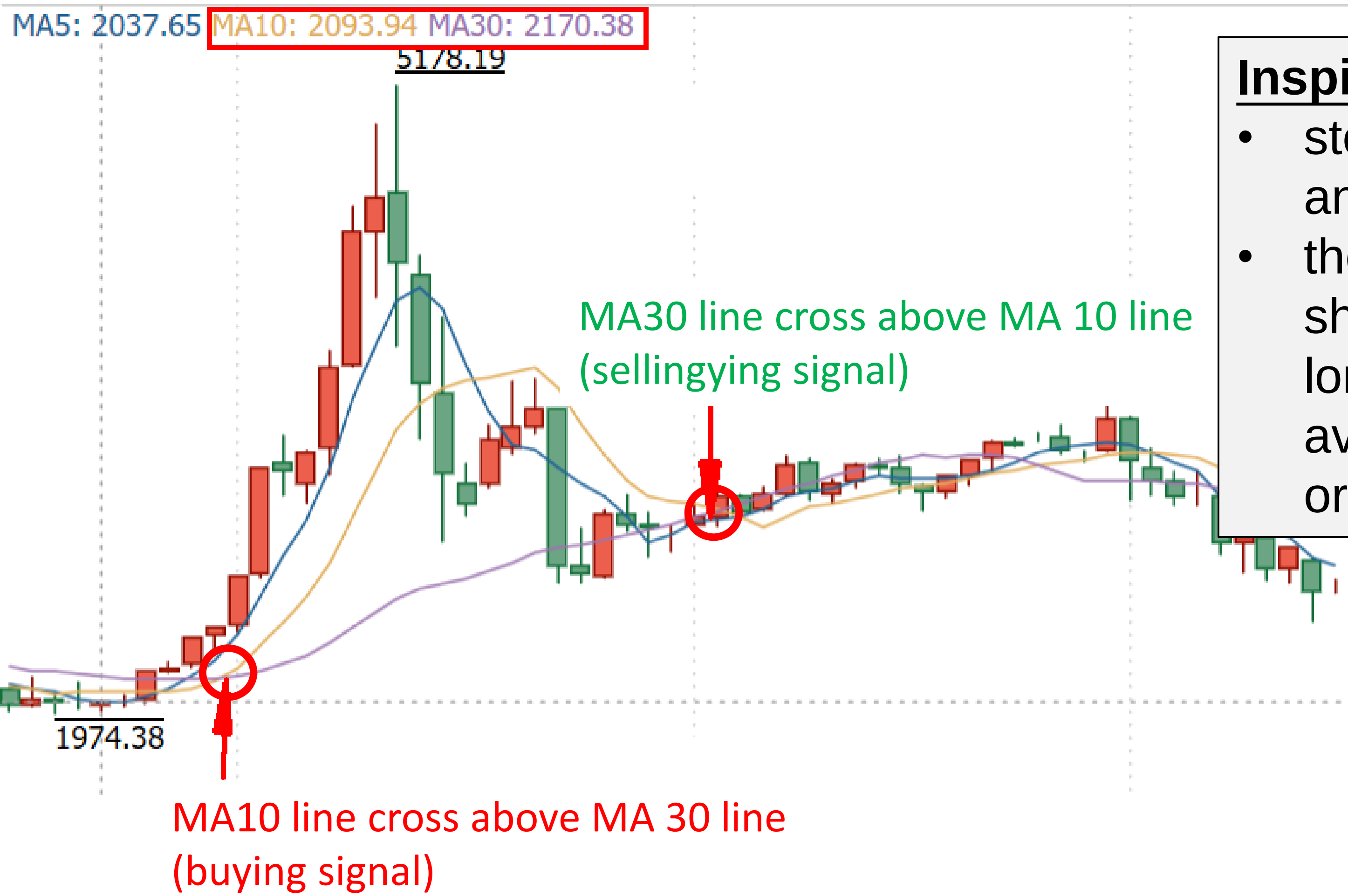
e.g. for $minLength = 3, lMinutil = 20$, then $\{b, c\}$ is a LHUI

When high utility?

- Since the utility vary over time, it is necessary to identify when the utility is significantly higher than usual.
- Thus, Peak high utility itemset is defined.
- How to define peak?
  - Naïve way: Time periods higher than threshold
  - Or greater than n standard deviations

How to define peak?

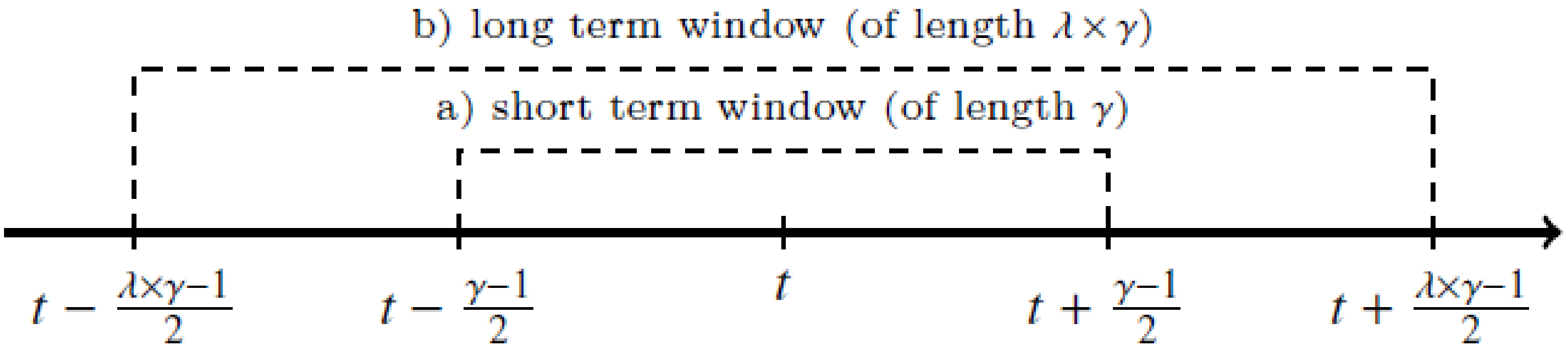
Shanghai composite index
(上证指数 2014.7-2018.11)



Inspiration:
- stock market analysis,
- the crossover of a short-term and a long-term moving average is a buying or selling signal.

MA= Moving Average

Problem Definition



Moving average utility (mau):
the average utility of X before and after t in a window of length γ

Peak: period when short-term moving average utility line is above **long-term** moving average utility line

Problem Definition

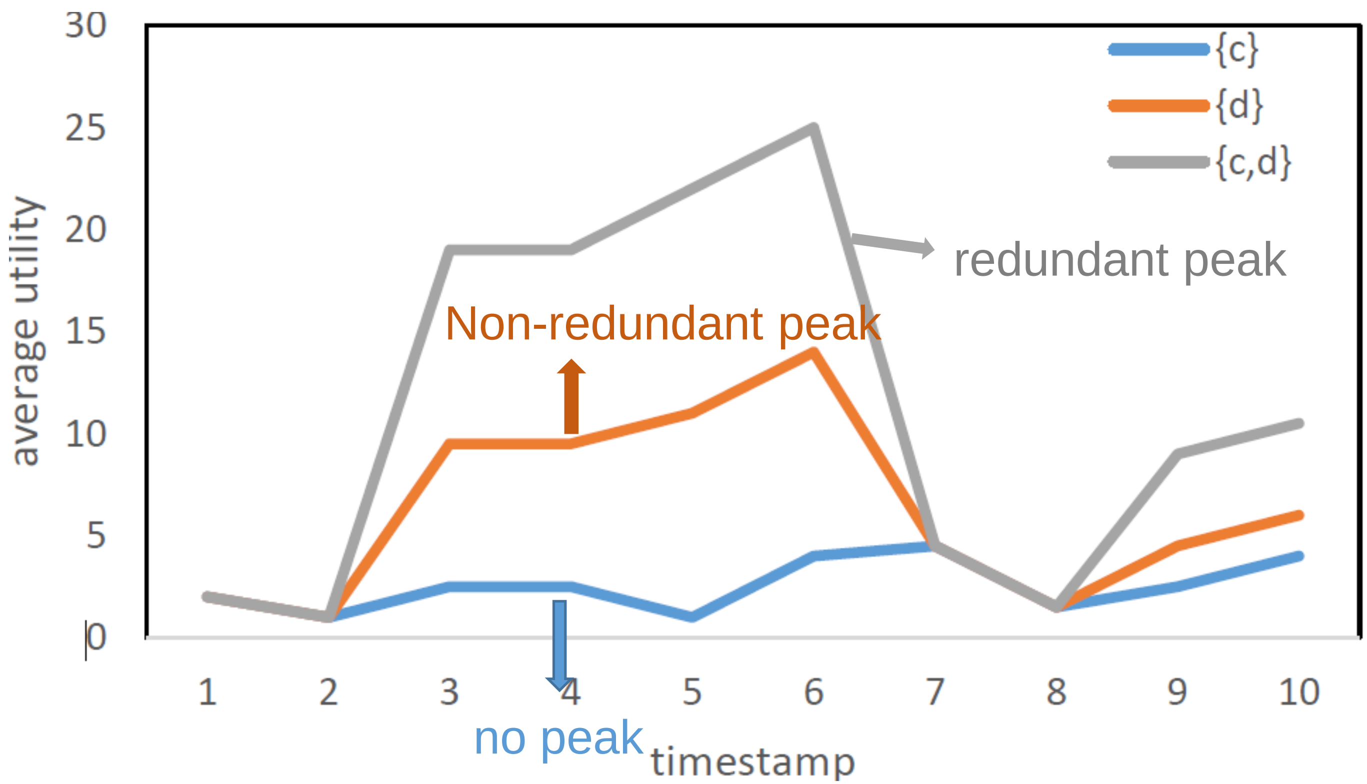
Goal: discover all LHUIs and their peak windows given parameters $minLength$, $lMinutil$ and λ.

Example:
- Assume that $minLength = 3$, $lMinutil = 10$ and $\lambda = 1.67$,
- **24 LHUIs** are found with their peak windows, including: $\{d\}$: $\{d_3, d_6\}$, $\{a, c\}$: $\{\{d_5, d_7\}, \{d_9, d_{10}\}\}$

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(2),c(2),e(1)	d3
T_4	**a(2)**,b(10),**c(2)**,**d(10)**,e(2)	**d5**
T_5	**a(2),c(6)**,e(2)	**d6**
T_6	b(4),c(3),e(1)	**d7**
T_7	**a(2),c(2)**,d(2)	**d9**
T_8	**a(2),c(6)**,e(2)	**d10**

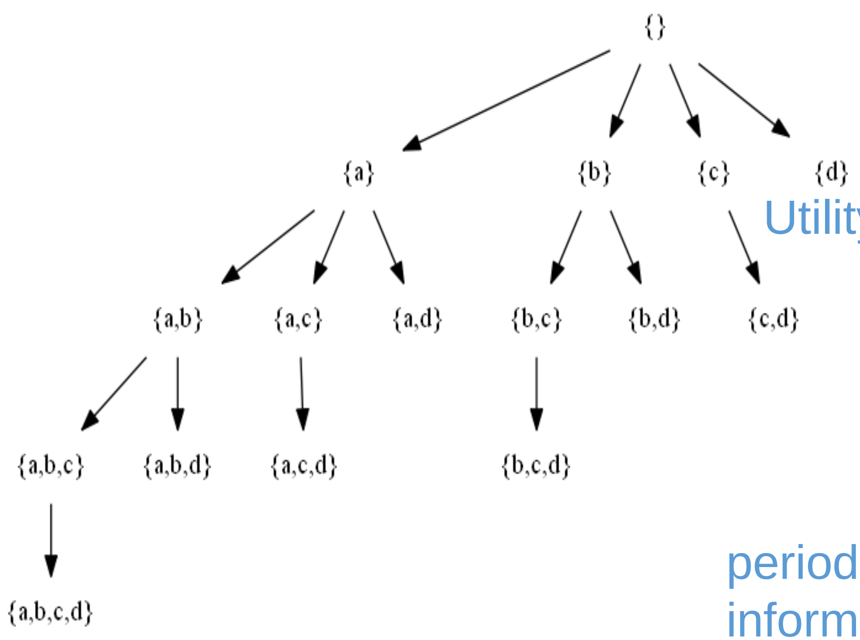
Non-redundant PHUI

However, there are too many peak patterns, the solution is to only keep Non-redundant PHUI (**NPHUI**), which is defined as a PHUI whose all subsets are PHUIs.



The Algorithms

- Based on **HUI-Miner**, it finds larger itemsets using a **depth-first search**;
- Create a vertical structure named **LU-List** for each itemset.



LU-list of $\{b\}$

Utility-list

tid	$iutil$	$rutil$
T_1	4	5
T_2	8	10
T_3	4	15
T_4	20	28
T_6	8	6

periods information

$iutilPeriods$
$[d_1, d_3], [d_3, d_7]$
$utilPeriods$
$[d_1, d_7]$

utility of itemset in a transaction

remaining utility

LHUI periods

promising LHUI periods

Construction of a Utility-list

- The **Utility-list** of a **single item** can be constructed by **scanning the database**.
- For other itemsets, it can be obtained by **joining their child itemset's Utility-lists.**

Utility-list $\{b\}$

tid	$iutil$	$rutil$
T_1	4	5
T_2	8	10
T_3	4	15
T_4	20	28
T_6	8	6

+

Utility-list $\{c\}$

tid	$iutil$	$rutil$
T_1	2	3
T_2	3	7
T_3	2	13
T_4	2	26
T_5	6	6
T_6	3	3
T_7	2	4
T_8	6	6

→

Utility-list $\{b, c\}$

tid	$iutil$	$rutil$
T_1	6	3
T_2	11	7
T_3	7	13
T_4	22	26
T_6	11	3

Construction of LU-list

- Consider that $a \prec b \prec c \prec d \prec e$, $minLength = 3$ and $minMau = 20,$
- using sliding windows to get periods information

Utility-list $\{b\}$			
tid	$iutil$	$rutil$	Time
T_1	4	5	d1
T_2	8	10	d3
T_3	4	15	d3
T_4	20	28	d5
T_6	8	6	d7

$sumIutil = 20$
$sumRutil = 30$
$sumUtil = 50$

$sumIutil = 12$
$sumRutil = 20$
$sumUtil = 32$

$sumIutil = 32$
$sumRutil = 53$
$sumUtil = 85$

$sumIutil = 20$
$sumRutil = 28$
$sumUtil = 58$

$sumIutil = 28$
$sumRutil = 34$
$sumUtil = 62$

$iutilPeriods$
$[d_1, d_3], [d_3, d_7]$
$utilPeriods$
$[d_1, d_7]$

Three optimizations

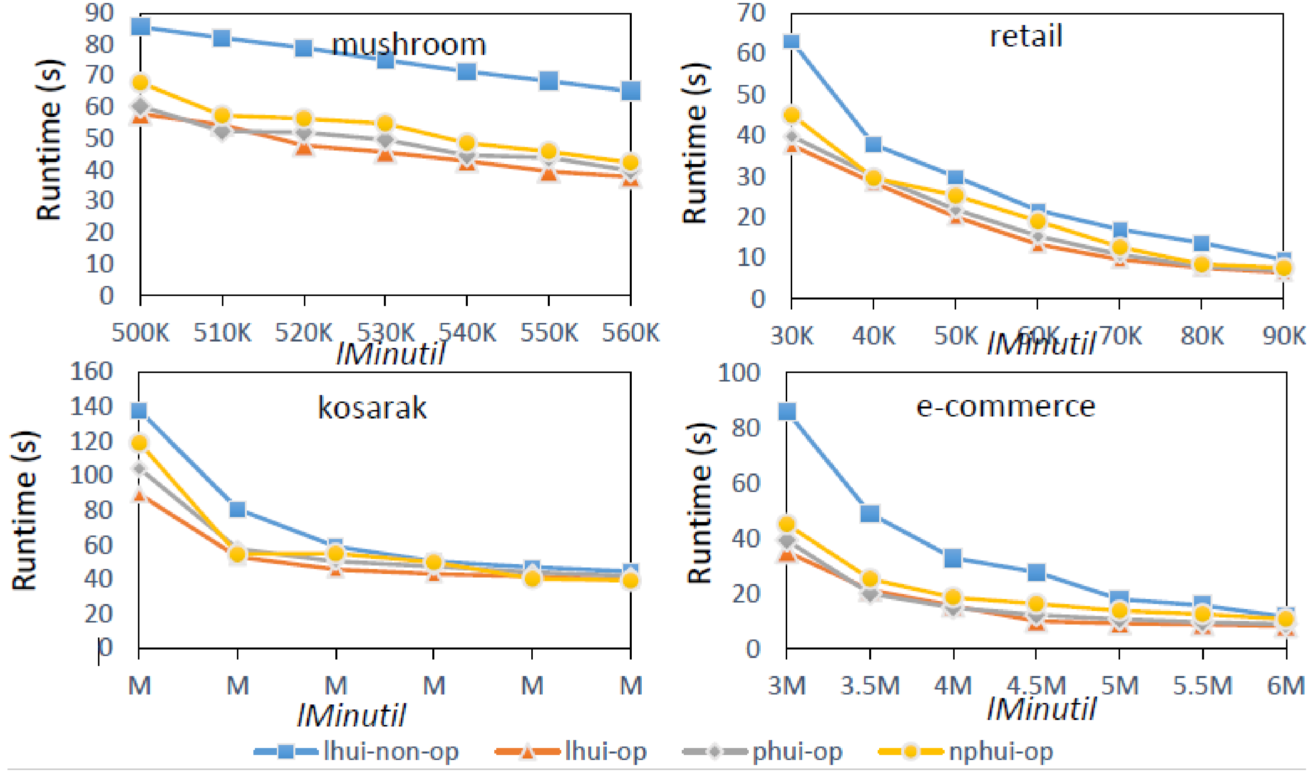
- **Discarding unpromising items using the sliding window**
  - Discard an item i, if for any window $W_{k,l}$ of $minLength$, $TWU_{k,l}(i) < lMinUtil$
- **Discarding irrelevant transactions**
  - Remove transactions that don’t contribute to any LHUI
- **Discarding unpromising tuples in LU-list**
  - Only keep tuples that are in $utilperiods$

Experimental Evaluation

Dataset	Trans count	Item count	Average length	Type
mushroom	8,124	119	23	dense
kosarak	990,000	41,270	8.09	Long transaction
retail	88,162	16,470	10.3	sparse
e_commerce	17,535	3,803	15.4	Real-life data

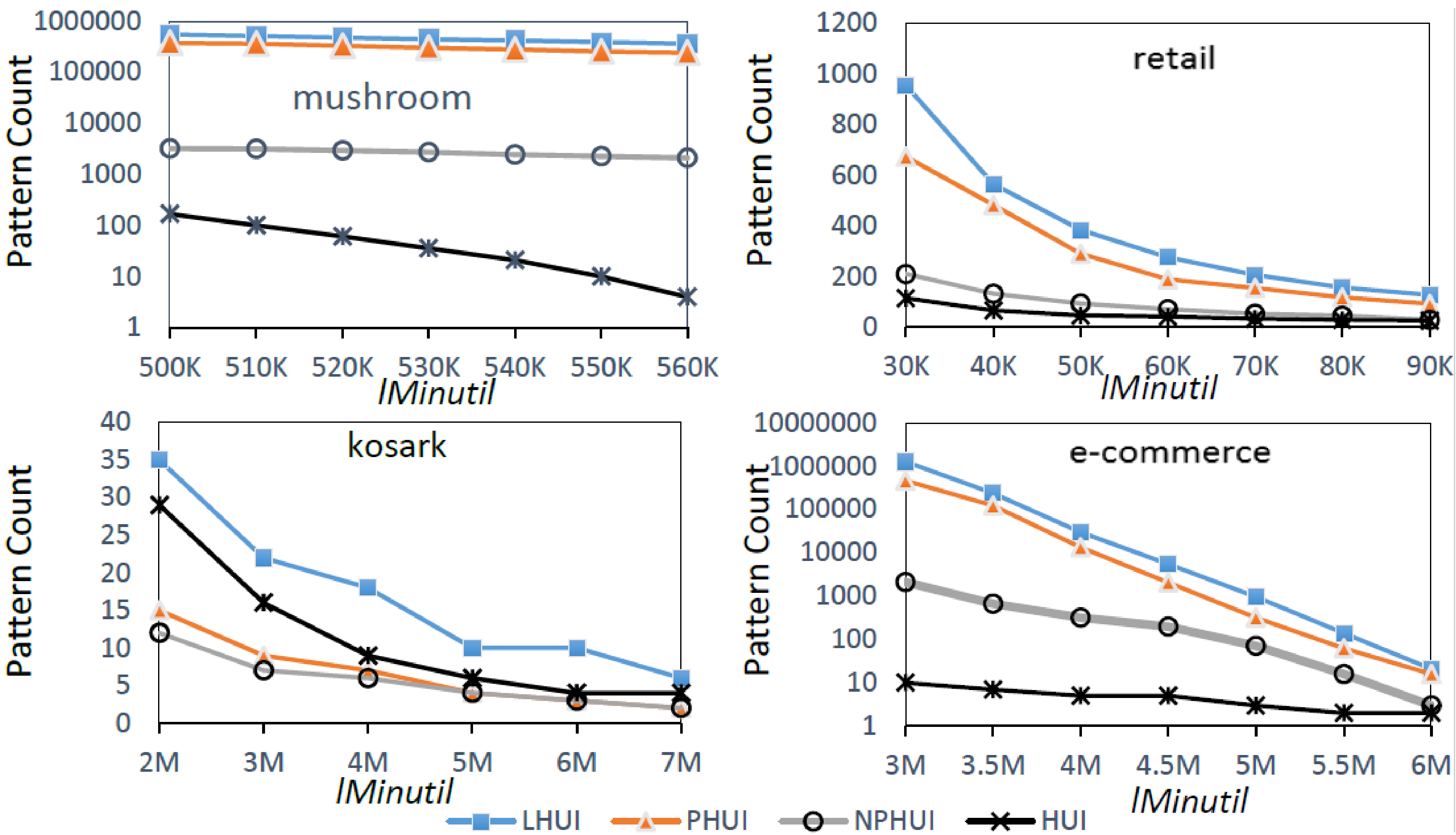
- **We compared the execution time of the algorithm with and without the optimizations**
- **We also compared the number of patterns found (LHUI and HUI)**
- **java, Windows 10, 16 GB RAM, Intel Xeon E3-1270 v5**

Run time



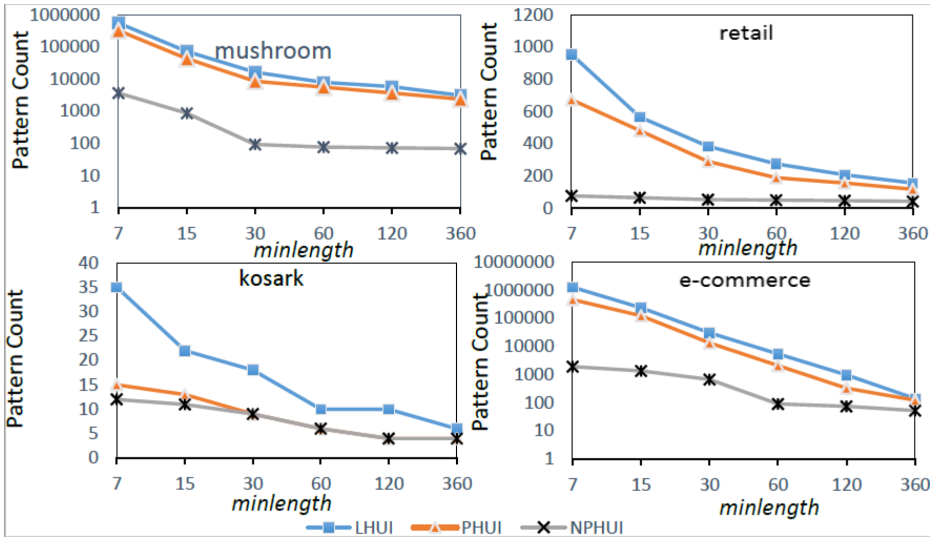
In some cases, the optimized algorithm is one time faster than the non-optimized algorithm

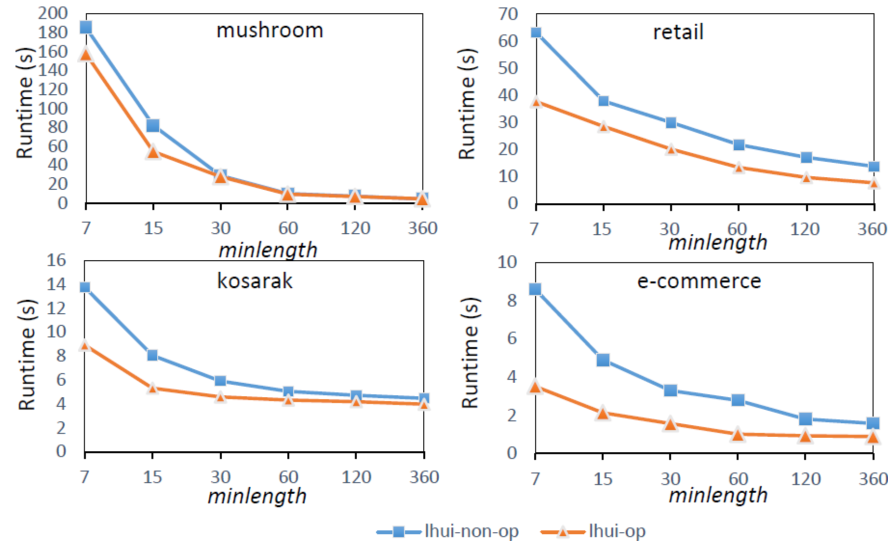
Pattern count

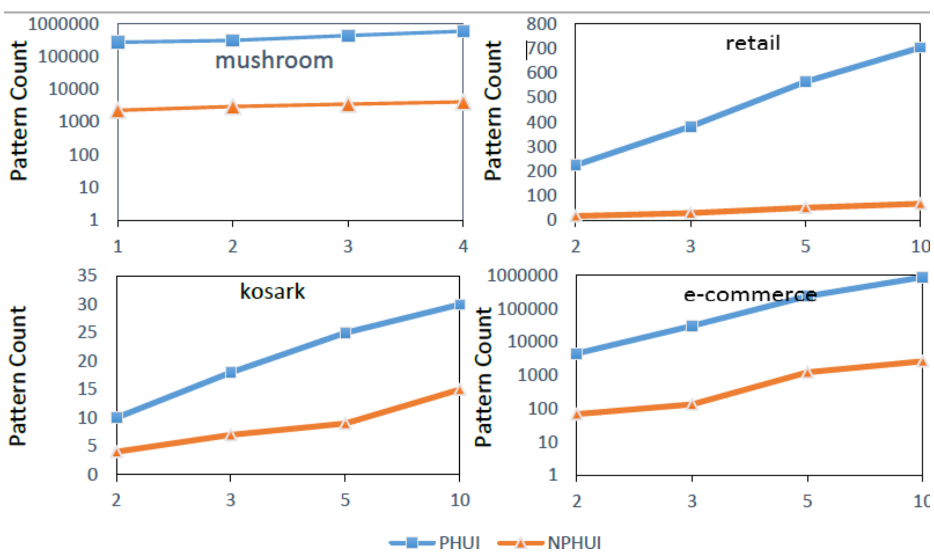


The number of LHUI and PHUI is much more than HUI, and NPHUI is much less than PHUI

Influence of parameters

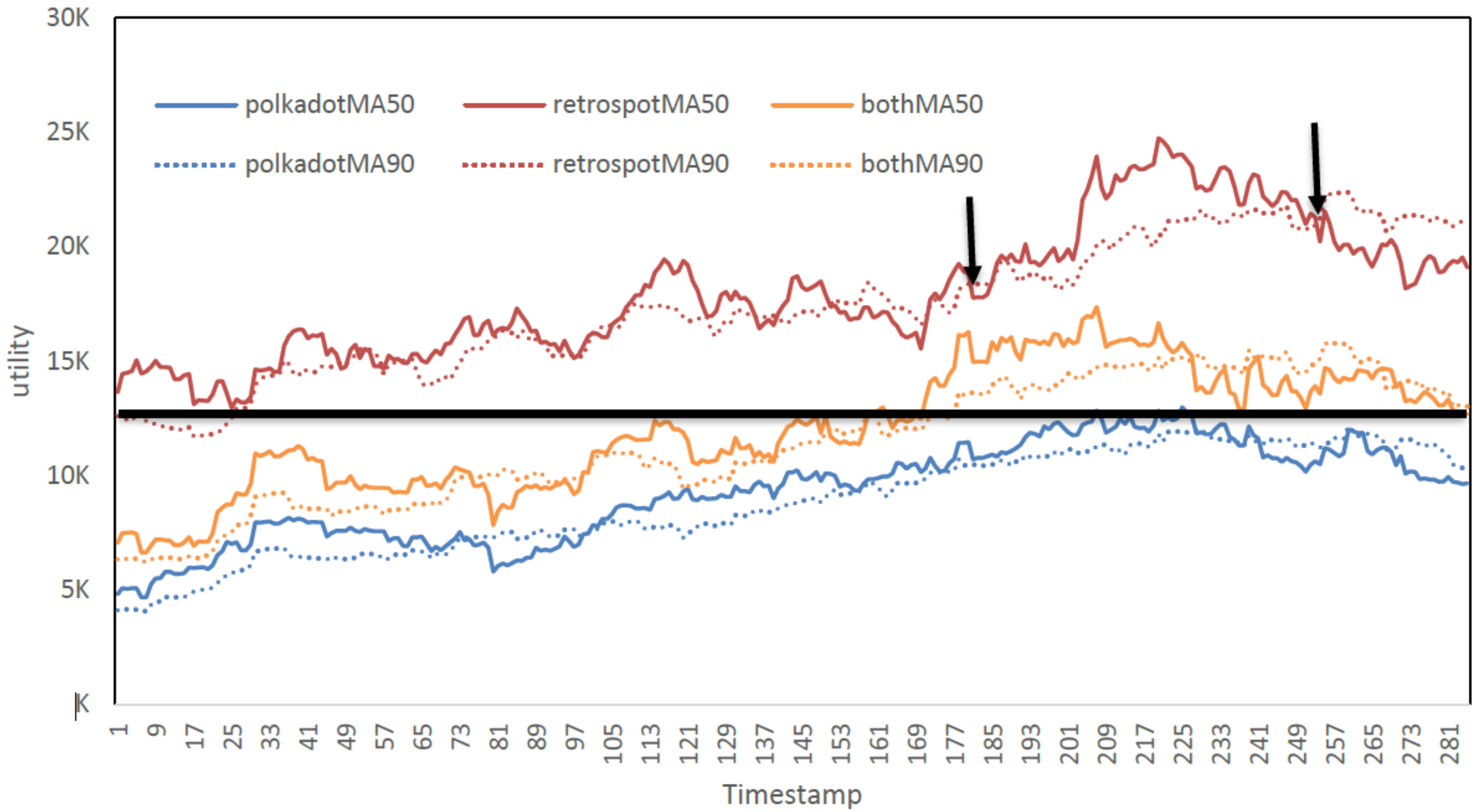






The running time and pattern number decrease when $minlength$ Increase. The pattern number increase when λ increase.

Usefulness Analysis



Example: for $lMinutil = 665{,}000$ and $\mathrm{min}Length = 50\ days$, the itemset $\{retro\ spot\ bag\}$, $\{retro\ spot\ bag,\ polka\ dot\ bag\}$ are two LHUIs, but $\{retro\ spot\ bag,\ polka\ dot\ bag\}$ is not a HUI if we set $minutil = 665{,}000 \times \frac{375}{50} = 5{,}320{,}000$.

let $\lambda = 1.8$, itemset $\{retro\ spot\ bag\}$ has a peak window from day 180 to day 257. And $\{retro\ spot\ bag,\ polka\ dot\ bag\}$ has a peak window from day 171 to day 227.

Itemset $\{retro\ spot\ bag\}$ is a NPHUI, while $\{retro\ spot\ bag,\ polka\ dot\ bag\}$ is not.

Conclusion

- New patterns: **LHUI, PHUI and NPHUI**
- New algorithms: **LHUI-Miner, PHUI-Miner and NPHUI-Miner**
- **Results:**
  - optimizations can reduce runtime by half in some cases,
  - generally, there is much more LHUI and PHUI than HUIs.
- **Future work:**
  - parallel processing
  - Adapt to other problems