HUE-Span: Fast High Utility Episode Mining

Philippe Fournier-Viger[1], Peng Yang[1], Jerry Chun-Wei Lin[2], Unil Yun[3]

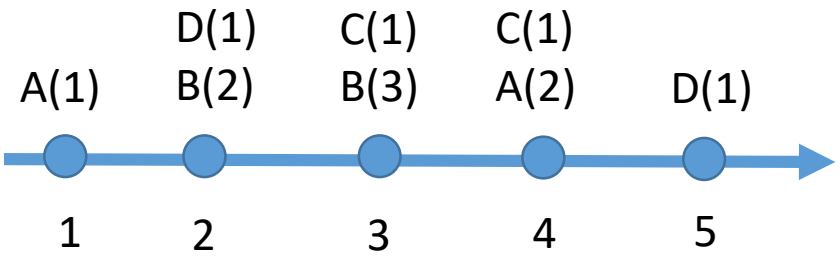
[1]Harbin Institute of Technology (Shenzhen), China
[2]Western Norway University of Applied Sciences (HVL), Bergen, Norway
[3]Sejong University, Seoul, Republic of Korea

philfv8@yahoo.com, pengyeung@163.com,
jerrylin@ieee.org, yunei@sejong.ac.kr

High Utility Episode Mining
(Wu et al, 2013 and recent papers)

Input:

A event sequence



A unit profit table

Event	A	B	C	D
Profit	2	1	3	2

$minUtil$: minimum utility threhold
$maxDur$: maximum time duration

Output:

High utility episodes (with utility $\geq minUtil$ & duration $\leq maxDur$)

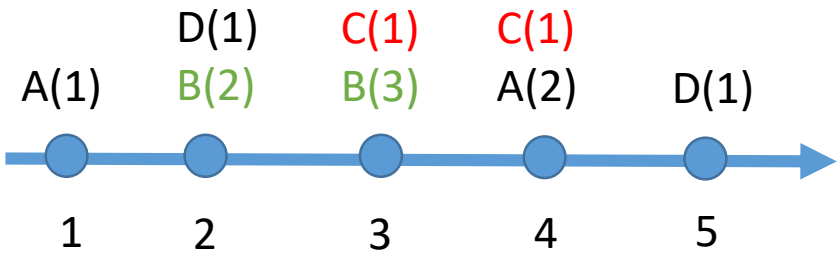
If set $minUtil$ = 15 and $maxDur$ = 3, HUEs are:

Episode	Minimal Occurrences	Utility
< (BC), (AC), (D) >	[3, 5]	15
<(B), (BC), (AC)>	[2, 4]	15
<(BD), (BC), (AC)>	[2, 4]	17
<(D), (BC), (AC)>	[2, 4]	15

What is a Minimal Occurrence?

- **Episode** $<(SE_1), (SE_2), \ldots, (SE_k)>$:
  - a non-empty totally ordered set of simultaneous events

- **Occurrence** $([t_s, t_e])$:
  - (i) SE_1 occurs at t_s and (ii) SE_k occurs at t_e
  - **e.g.:** occSet(<(B), (C)>) = [2, 3], [2, 4], [3, 4]

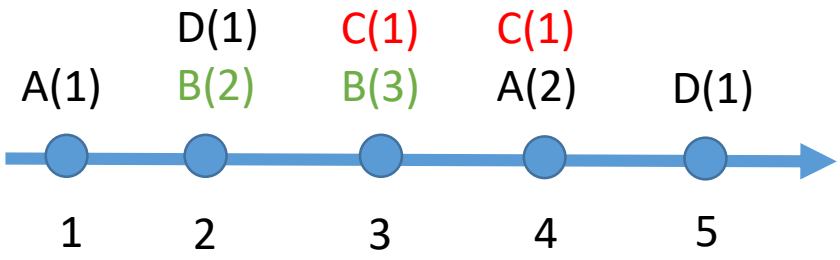


What is a Minimal Occurrence?

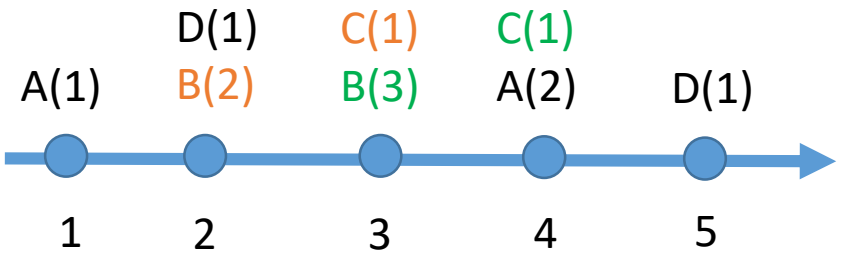
- **Episode** $<(SE_1), (SE_2), \ldots, (SE_k)>$:
  - a non-empty totally ordered set of simultaneous events

- **Occurrence** $([t_s, t_e])$:
  - (i) SE_1 occurs at t_s and (ii) SE_k occurs at t_e
  - e.g.: occSet(<(B), (C)>) = [2, 3], [2, 4], [3, 4]

- **Minimal Occurrence:**
  - no alternative occurrence $[t'_s, t'_e]$ is a sub-time interval of $[t_s, t_e]$
  - e.g.: moSet(<(B), (C)>) = [2, 3], [3, 4]



How to Calculate the Utility?



Event	A	B	C	D
Profit	2	1	3	2

$minUtil$ = 10 and $maxDur$ = 4

The utility of the episode <(B), (C)> is calculated as follows:

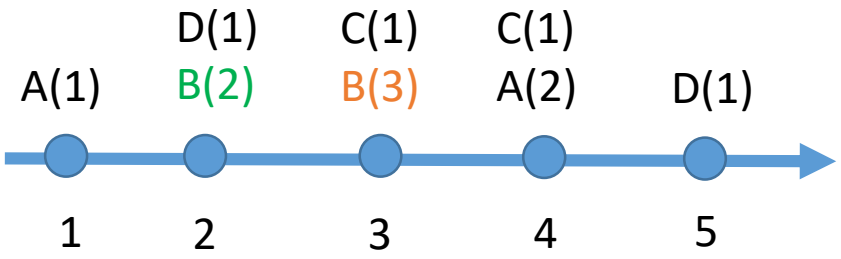
u(<(B),(C)>) = u(<(B),(C)>, [2, 3]) + u(<(B),(C)>, [3, 4])

= (2*1) + (1*3) + (3*1) + (1*3)

= 11 > 10

So, <(B), (C)> is a High Utility Episode (HUE)

A Problem with the Utility Calculation



Event	A	B	C	D
Profit	2	1	3	2

$minUtil$ = 10 and $maxDur$ = 4

Consider the utility of <(A), (B), (A)>:

- Previous works would choose the first B:
  - u(<(A), (B), (A)>) = 1*2 + 2*1 + 2*2 = 8
  - the episode's utility may be underestimated

- We choose the highest utility:
  - u(<(A), (B), (A)>) = 1*2 + 3*1 + 2*2 = 9

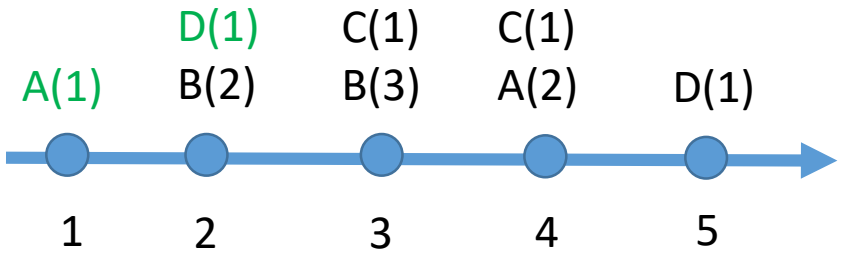
The EWU Upper-Bound on the Utility
(Wu et al, 2013 and recent papers)

Let an episode $\alpha =< (SE_1), (SE_2), \dots, (SE_k) >$ satisfying $maxDur$, where simultaneous event sets are associated with the time points $t_1, t_2, \dots t_k$.

- EWU of a MO : $\sum_{i=1}^{k-1} u(SE_i, t_i) + \sum_{j=t_k}^{t_1+maxDur-1} u(tSE_j,\ j)$
  - tSE_i : simultaneous event set at j
  - Don't need to keep the order of events

- EWU(<(A),(D)>, [1,2]) = u(A, 1) + u(BD, 2) + u(BC, 3) + u(AC,4)
  = 19

$minUtil$ = 10
$maxDur$ = 4



Event	A	B	C	D
Profit	2	1	3	2

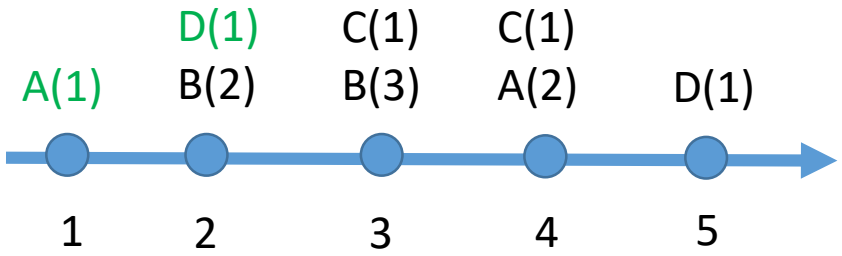
A Tighter Upper-Bound called ERU

Let an episode $\alpha =< (SE_1), (SE_2), \dots, (SE_k) >$ satisfying $maxDur$, where simultaneous event sets are associated with the time points $t_1, t_2, \dots t_k$.

- ERU of a MO: $\sum_{i=1}^{k} u(SE_i, t_i) + \text{u}(rSE_k, t_k) + \sum_{j=t_k+1}^{t_1+maxDur-1} u(tSE_j,\ j)$
  - rSE_k : the remaining event set of (SE_k) at t_k
  - Need to keep the order of events

- ERU(<(A),(D)>, [1,2]) = u(A, 1) + u(D, 2) + 0 + u(BC, 3) + u(AC,4)
  = 17

$minUtil$ = 10
$maxDur$ = 4



Event	A	B	C	D
Profit	2	1	3	2

The Utility Co-occurrence Structures

- $EECUS_{simult}$: stores the action-window utilization of all pairs of events by **i-extension**
- $EEUCS_{serial}$: stores the action-window utilization of all pairs of events by **s-extension**

Event	A	D	B	C
A		0	0	19
D			19	0
B				21
C				

$(a)\ EEUCS_{simult}$

Event	A	D	B	C
A	0	27	12	12
D	17	0	19	19
B	36	15	19	38
C	19	30	0	19

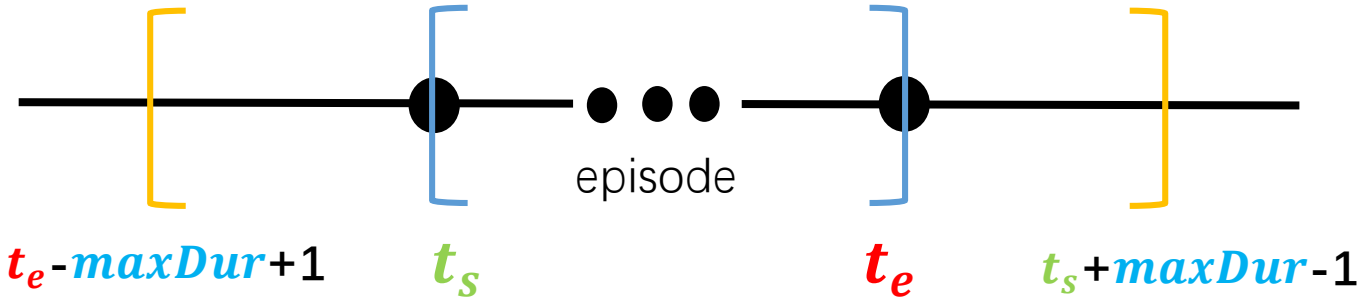
$(b)\ EEUCS_{serial}$

This figure shows *EEUCS* for ***maxDur*** = 3

The action-window utilization is the loosest upper-bound on the utility

Action-Window Utilization

- Action-Window:



- Action-Window Utilization of $[t_s, t_e]$:
  - AWU(α, $[t_s, t_e]$) = $\sum_{t_i=t_s-maxDur+1}^{t_e+maxDur-1} \mathrm{u}(tSE_i, t_i)$

Pruning Properties

- **Pruning an i-extension**: The i-extension of an episode α with an event x is **not** high utility if there exist an event i in α such that AWU(<(i, x)>) in $EECUS_{simult}$ less than $minUtil$.

- **Pruning an s-extension**: The s-extension of an episode α with an event x is **not** high utility if there exist an event i in α such that AWU(<$(i),(x)$>) in $EECUS_{serial}$ less than $minUtil$.

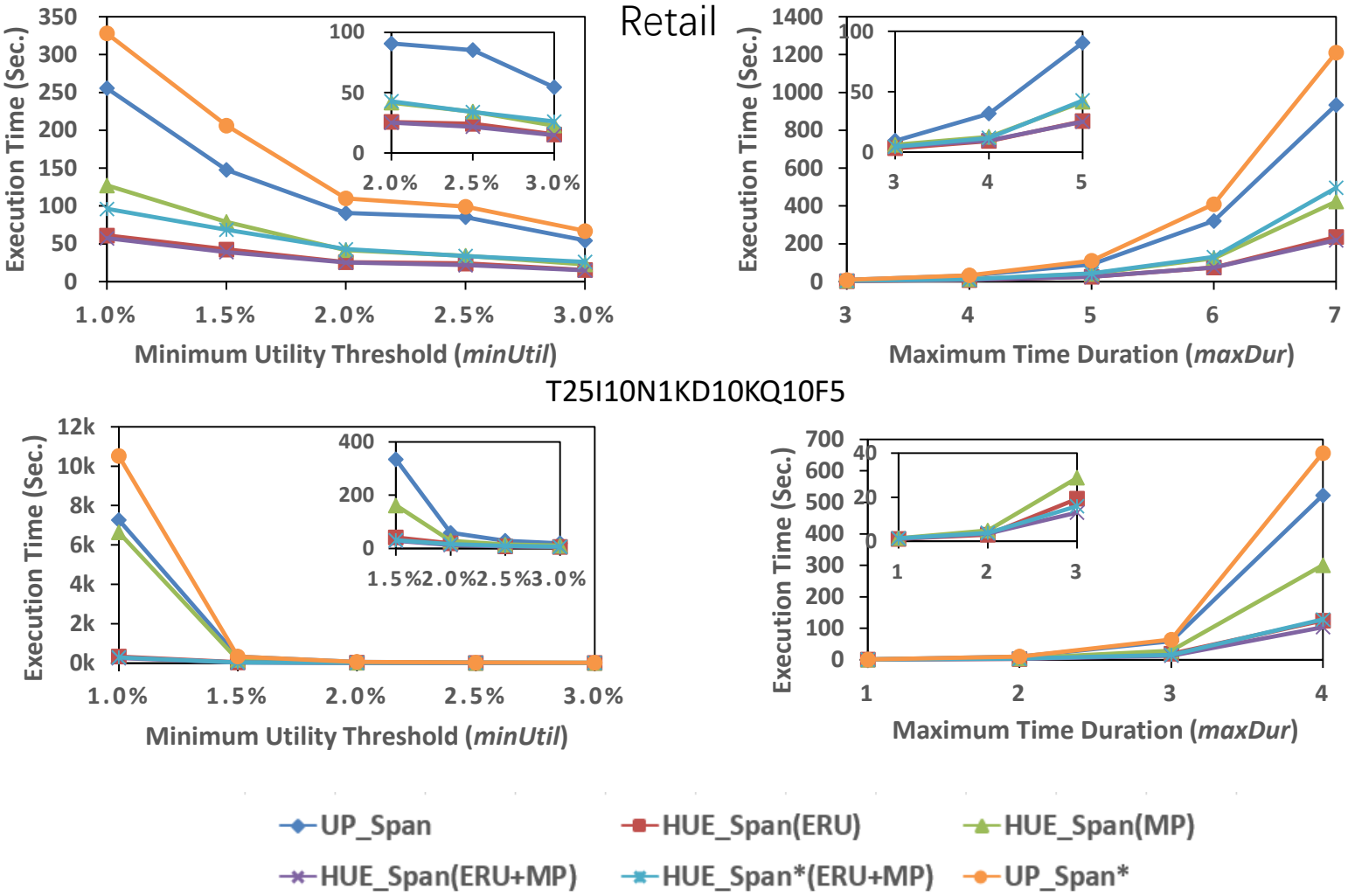
Experimental Evaluation

- **Statistical information about three datasets**

Dataset	#Time Point	#Event	Avg. Length
T25I10N1KD10KQ10F5	9,976	929	24.8
Retail	88,162	16,470	10.3
Kosarak	990,002	41,270	8.1

- Retail and Kosarak are real datasets
- T25I10N1KD10KQ10F5 (synthetic, using SPMF generator):
  - average profit is 5
  - quantities between 1 and 10
- Java, Windows10

Execution times

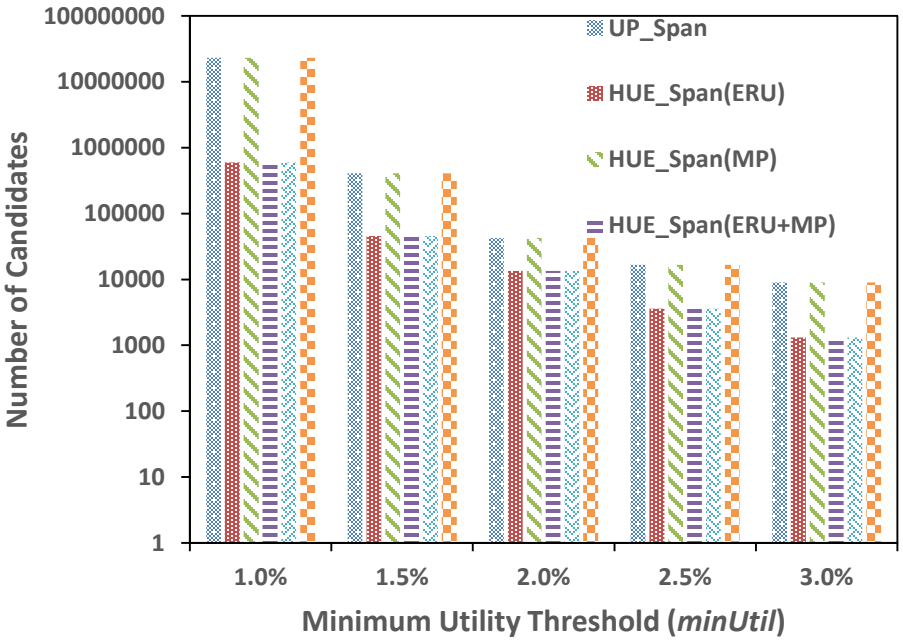


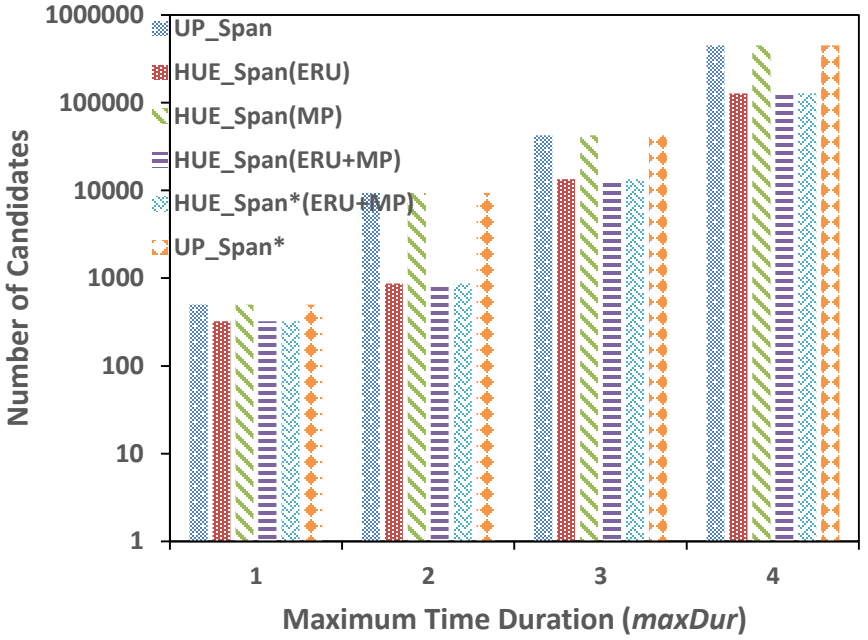
The notation * means that it mines maximal high utility episodes

Increasing $minUtil$ or decreasing $maxDur$ ofen increase the runtime.

Number of Candidates

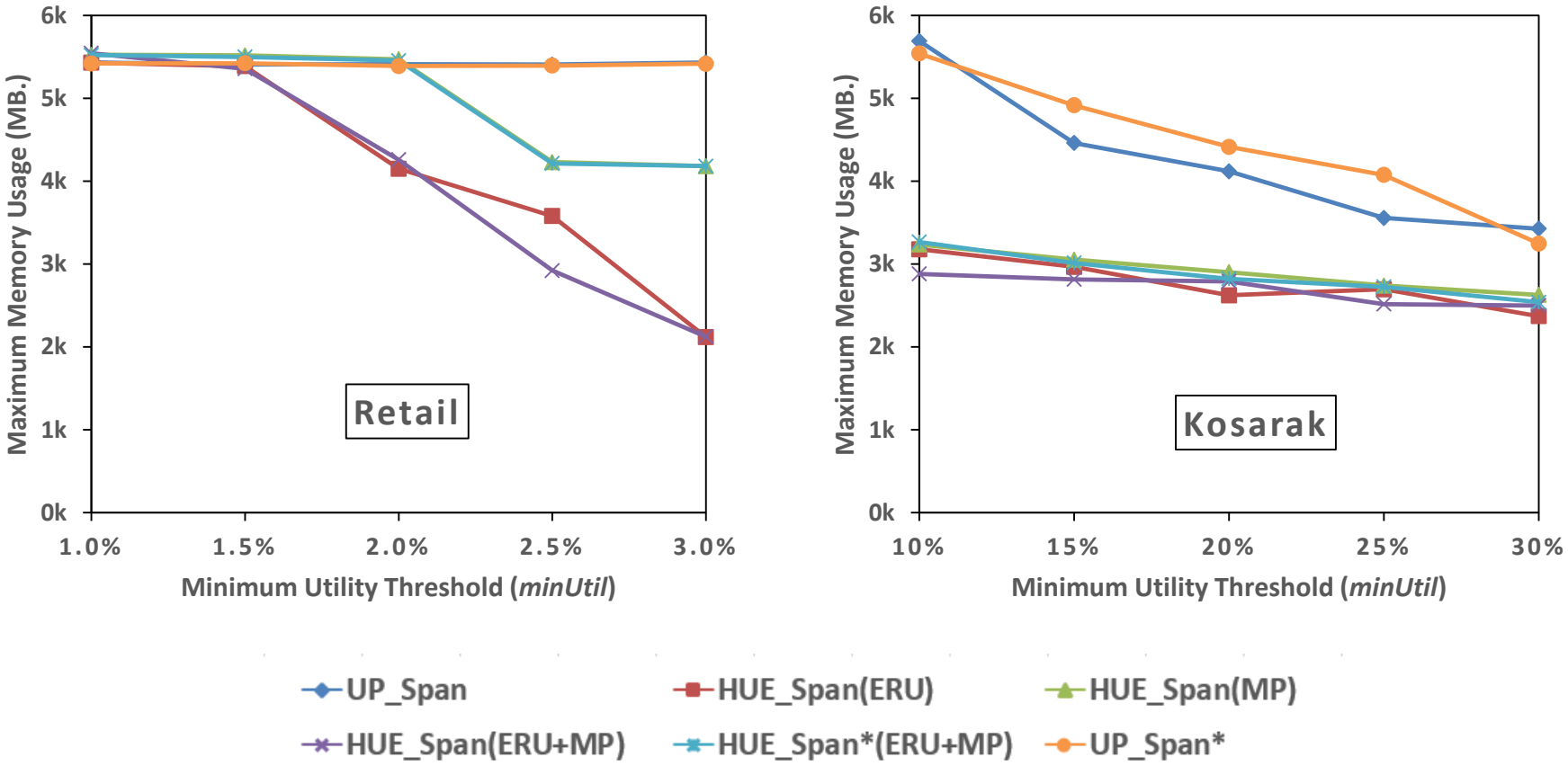
T25I10N1KD10KQ10F5





The number of candidates grows rapidly when
$minUtil$ is decreased or
$maxDur$ is increased.

Peak memory usage



HUE-Span with pruning strategies uses less memory than UP-Span since the proposed pruning strategies reduce the number of candidates.

Number of patterns found

Dataset	$minUtil$	$maxDur$	$\#HUE^*$	$\#HUE$	$\#HUE^-$
Retail	1%	5	1,556	1,174	745
	1.5%	5	523	422	196
	2%	5	179	170	98
	2%	6	730	439	296
	2%	7	2,084	1,077	858
Kosarak	10%	5	105	73	29
	15%	5	27	22	5
	20%	5	3	2	0
	20%	6	21	8	1
	20%	7	81	28	16

HUE^* : high utility episode with maximal utility
HUE : high utility episode
HUE^- : high utility episode that its utility is not maximal utility

UP-Span finds much less HUEs than the proposed HUE-Span* algorithm
UP-Span underestimates the utility of up to 79% of the (maximal) HUEs.

Conclusion

- Contributions:
  - Redefined utility: **highest (maximal) utility**
  - A Tighter Upper-Bound on the utility: **ERU**
  - A novel pruning strategy based on event co-occurrences
  - An efficient algorithm, named **HUE-Span**

- Future work:
  - Design other optimizations for high utility episode mining
  - consider using high utility episodes to derive high utility episode rules

- Source code and datasets available as part of the SPMF data mining library (GPL 3).

Open source Java data mining software, 178 algorithms
http://www.philippe-fournier-viger.com/spmf/

Thank you. Questions?

Open source Java data mining software, 178 algorithms
http://www.philippe-fournier-viger.com/spmf/