Discovering and Visualizing Efficient Patterns in Cost/Utility Sequences

Philippe Fournier-Viger[1], Jiaxuan Li[1], Jerry Chun-Wei Lin[2], Tin Truong Chi[3]

[1]Harbin Institute of Technology (Shenzhen), China

[2]University of Applied Sciences (HVL), Bergen, Norway

[3] University of Dalat, Vietnam

High-utility sequential pattern mining

Input

Quantitative sequences with purchase quantities (internal utility)
sequence 1: $\langle (a,3),(b,3),(c,1),(b,4)\rangle$
sequence 2: $\langle (a,1),(e,3)\rangle$
sequence 3: $\langle (a,6),(c,7),(b,8),(d,9)\rangle$
sequence 4: $\langle (b,3),(c,1)\rangle$
Unit profits (external utility)
$a = 5\$$, $b = 1\$$, $c = 2\$$, $d = 1\$$

a minimum utility threshold ***(e.g. minutil = 30)***

Output

All **sequences** having a $\boldsymbol{utility \geq minutil}$)

High-utility sequential pattern mining

Input

Quantitative sequences with purchase quantities (internal utility)
sequence 1: $\langle \boxed{(a,3),(b,3)},(c,1),(b,4)\rangle$
sequence 2: $\langle (a,1),(e,3)\rangle$
sequence 3: $\langle \boxed{(a,6)},(c,7),\boxed{(b,8)},(d,9)\rangle$
sequence 4: $\langle (b,3),(c,1)\rangle$
Unit profits (external utility)
$\boxed{a=5\$}\ \boxed{b=1\$},\ c=2\$,\ d=1\$$

a minimum utility threshold ***(e.g. minutil = 30)***

Output

All **sequences** having a $\boldsymbol{utility \geq minutil}$)

The **sequence <ab> is a high utility pattern because:**

$$u(<ab>) = \underbrace{3\times5+3\times1}_{\text{Sequence 1}} + \underbrace{6\times5+8\times1}_{\text{Sequence 3}} = 56 > minutil$$

Limitations

- **High utility pattern mining** aims at discovering patterns that have a high utility.
- But it ignores the cost or effort required to obtain these benefits.
- May find patterns that have:
  - **a high utility but a very high cost**
- **Cost of a pattern:** *time, money, resources consumed* or *effort*.

Our proposal: Find Cost-effective Patterns →

Sequential Activity Database

- A **sequence** is a series of activities, each having a cost.
- The **utility** of a sequence is a binary class or a positive number.

Sid	<Activity : cost>	Utility
S_1	<(a:4)(b:2)(e:4)(c:4)(d:5)>	Positive
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	Negative
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	Positive
S_4	<(a:2)(b:2)(c:1)(f:2)>	Negative

(e.g. cured or died after some medical treatments)

Sid	<Activity : cost>	Utility
S_1	<(a:4)(b:2)(e:4)(c:4)(d:5)>	40
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	50
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	60
S_4	<(a:2)(b:2)(c:1)(f:2)>	70

(e.g. score obtained at an exam)

The *support* measure

Sid	<Activity : cost>	Utility
S_1	<(**a**:**4**)(**b**:**2**)(e:4)(c:4)(d:5)>	...
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	...
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	...
S_4	<(**a**:**2**)(**b**:**2**)(c:1)(f:2)>	...

The **support** of a pattern p:

$$\sup(p) = |S_s|p \subseteq S_s \in DB|$$

(number of sequences containing p)

e.g. $sup(<\mathbf{ab}>)=|\{S_1, S_4\}| = \mathbf{2}$ sequences

This measure is used to remove noise.

The *cost* measure

Sid	<Activity : cost>	...
S_1	<(**a**:**4**)(**b**:**2**)(e:4)(c:4)(d:5)>	...
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	...
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	...
S_4	<(**a**:**2**)(**b**:**2**)(c:1)(f:2)>	...

The **cost** of a pattern p:

$$c(p, S_s) = \sum_{v_i \in first(p, S_s)} c(v_i, S_s)$$

$$\mathbf{c(ab,}\ \boldsymbol{S_1}\mathbf{)} = 4 + 2 = \mathbf{6}$$

The *cost* measure

Sid	<Activity : cost>	...
S_1	<(**a**:**4**)(**b**:**2**)(e:4)(c:4)(d:5)>	...
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	...
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	...
S_4	<(**a**:**2**)(**b**:**2**)(c:1)(f:2)>	...

The **cost** of a pattern p:

$$c(p, S_s) = \sum_{v_i \in first(p, S_s)} c(v_i, S_s)$$

$$\mathbf{c(ab, \boldsymbol{S_1})} = 4 + 2 = \mathbf{6}$$

The **average cost** of a pattern p:

$$ac(p) = \frac{\sum_{p \subseteq S_s \in DB} c(p, S_s)}{|\sup(p)|}$$

$$\mathbf{ac(ab)} = \underbrace{6}_{\textbf{Sequence 1}} + \underbrace{4}_{\textbf{Sequence 4}} / 2 = \mathbf{5}$$

This measure is used to assess the effort or resource spent.

The *occupancy* measure

Sid	<Activity : cost>	...
S_1	<(**a**:**4**)(**b**:**2**)(e:4)(c:4)(d:5)>	...
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	...
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	...
S_4	<(**a**:**2**)(**b**:**2**)(c:1)(f:2)>	...

The **occupancy** of a pattern p:

$$occup(p) = \frac{1}{sup(p)} \sum_{p \subseteq S_s \in SEL} \frac{|p|}{|S_s|}$$

$$\mathbf{occup}(\boldsymbol{ab}) = \frac{1}{2} \cdot \left(\underbrace{\frac{2}{5}}_{\textbf{Sequence 1}} + \underbrace{\frac{2}{4}}_{\textbf{Sequence 4}} \right) = \mathbf{0.45}$$

This measure is used to remove patterns that are short and non-representative of the containing sequences.

Problem 1:
Finding all cost-effective patterns in a **binary DB**

Sid	<Activity : cost>	Utility
S_1	<(a:4)(b:2)(e:4)(c:4)(d:5)>	Positive
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	Negative
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	Positive
S_4	<(a:2)(b:2)(c:1)(f:2)>	Negative

A **pattern** p is **cost-effective** if:

$$\sup(p) \geq minsup$$

$$ac(p) \leq \mathrm{maxcost}$$

$$occup(p) \geq minoccup$$

Problem 1:
Finding all cost-effective patterns in a **binary DB**

Sid	<Activity : cost>	Utility
S_1	<(a:4)(b:2)(e:4)(c:4)(d:5)>	Positive
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	Negative
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	Positive
S_4	<(a:2)(b:2)(c:1)(f:2)>	Negative

A **pattern** p is **cost-effective** if:

$$sup(p) \geq minsup$$

$$ac(p) \leq \text{maxcost}$$

$$occup(p) \geq minoccup$$

Furthermore, we measure the **correlation of** a **pattern** p with the desirable outcome:

$$cor(p) = \frac{ac(D_p^+) - ac(D_p^-)}{Std} \sqrt{\frac{|D_p^+||D_p^-|}{|D_p^+ \cup D_p^-|}} \quad \in [-1,1]$$

a positive correlation is desirable

Pattern	support	average cost	correlation
<ac>	3	5.3	0.80

More details…

The **correlation** of a pattern p:

$$cor(p) = \frac{ac(D_p^+) - ac(D_p^-)}{Std}\sqrt{\frac{|D_p^+||D_p^-|}{|D_p^+ \cup D_p^-|}}$$

$where, ac(D_p^+), ac(D_p^-)\ denotes\ pattern\ p's\ average\ cost\ in\ positive\ and\ negative\ sequences, respectively.$

- $ac(D_p^+) - ac(D_p^-)$, indicates the difference in terms of average cost for positive and negative sequences.
- *Std*, standard deviation of the cost to avoid absolute values.
- $\sqrt{\frac{|D_p^+||D_p^-|}{|D_p^+ \cup D_p^-|}}$, measures distribution difference to indicate patterns' effect on the outcome.
- The *cor* measure values are in the [-1,1] interval.
- The greater positive(negative) the *cor* measure is, the more a pattern is correlated with a positive (negative) utility.

A full example

Database

Sid	<Activity : cost>	Utility
S_1	<(a:4)(b:2)(e:4)(c:4)(d:5)>	Positive
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	Negative
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	Positive
S_4	<(a:2)(b:2)(c:1)(f:2)>	Negative

$minsup = 3$

$maxcost = 10$

Cost-effective patterns

Pattern	support	average cost	correlation
a	3	2.7	0.50
b	3	2.3	-0.50
c	4	2.5	0.89
d	3	3.7	0.99
e	3	2.3	0.19
f	3	1.7	0.50
ac	3	5.3	0.80
bc	3	4.7	0.76
cd	3	6.7	0.99

Problem 2:
Finding all cost-effective patterns in a **numeric DB**

Sid	<Activity : cost>	Utility
S1	<(a:4)(b:2)(e:4)(c:4)(d:5)>	40
S2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	50
S3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	60
S4	<(a:2)(b:2)(c:1)(f:2)>	70

A **pattern** p is **cost-effective** if:

$$\text{sup}(p) \geq minsup$$

$$ac(p) \leq \text{maxcost}$$

$$occup(p) \geq minoccup$$

Problem 2:
Finding all cost-effective patterns in a **numeric DB**

Sid	<Activity : cost>	Utility
S1	<(a:4)(b:2)(e:4)(c:4)(d:5)>	40
S2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	50
S3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	60
S4	<(a:2)(b:2)(c:1)(f:2)>	70

A **pattern** p is **cost-effective** if:

$$\sup(p) \geq minsup$$

$$ac(p) \leq \mathrm{maxcost}$$

$$occup(p) \geq minoccup$$

Furthermore, we measure the **trade-off** between the **cost** and **utility** of a **pattern** p :

$$tf(p) = \frac{ac(p)}{u(p)}$$

Average cost

Utility

$$u(p) = \frac{\sum_{p \subseteq S_s \in DB} su(S_s)}{|\sup(p)|}$$

Trade-off values are in the $(0, \infty]$ interval. Lower means more efficient.

More details…

Sid	<Activity : cost>	Utility
S1	<(**a:4**)(**b:2**)(e:4)(**c:4**)(**d:5**)>	40
S2	<(b:3)(**c:2**)(f:1)(**d:1**)(e:2)>	50
S3	<(a:2)(f:2)(e:1)(**c:3**)(**d:5**)>	60
S4	<(**a:2**)(**b:2**)(c:1)(f:2)>	70

Utility of a pattern *p*:

$$u(p) = \frac{\sum_{p \subseteq S_s \in SADB} su(S_s)}{|\text{sup}(p)|}$$

$$u(\mathbf{ab}) = \underbrace{40}_{\textbf{Sequence 1}} + \underbrace{70}_{\textbf{Sequence 3}} /2 = \mathbf{55}$$

Trade-off of a pattern *p*:

$$tf(p) = \frac{ac(p)}{u(p)}$$

$tf(\mathbf{ab}) = 5 /55 = \mathbf{0.09}$

$tf(\mathbf{cd}) = 6.7 /50 = \mathbf{0.13}$

Thus, pattern (***ab***) is more efficient than (***cd***).

A full example

Database

Sid	<Activity : cost>	Utility
S1	<(a:4)(b:2)(e:4)(c:4)(d:5)>	40
S2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	50
S3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	60
S4	<(a:2)(b:2)(c:1)(f:2)>	70

minsup=3
maxcost=10

Cost-effective patterns

Utility:50		Utility:53		Utility:55		Utility:56		Utility:60	
pattern	**tf**	**pattern**	**tf**	**pattern**	**tf**	**pattern**	**tf**	**pattern**	**tf**
e	**0.05**	**b**	**0.04**	**c**	**0.05**	**a**	**0.05**	**f**	**0.03**
d	0.07	bc	0.09			ac	0.09		
cd	0.13								

How to reduce the search space? (1)

Sid	<Activity : cost>	Utility
S_1	<(a:4)(**b:2**)(e:4)(**c:4**)(d:5)>	…
S_2	<(**b:3**)(**c:2**)(f:1)(d:1)(e:2)>	…
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	…
S_4	<(a:2)(**b:2**)(**c:1**)(f:2)>	…

We propose a **lower-bound** on the **average cost**:

$$AMSC(p) = \frac{1}{minsup} \sum_{i=1,2,..,minsup} c(p, S_i)$$

where c(p,S_i) are sorted in ascending order.

e.g. For *minsup = 2*

c(**bc**,S_1) = 6 c(**bc**,S_2) = 5 c(**bc**,S_4) = 3

AMSC(**bc**) = (3+5) / 2 = 4

Properties of *AMSC*

$$AMSC(p) = \frac{1}{minsup} \sum_{i=1,2,..,minsup} c(p, S_i)$$

Properties of the AMSC:

I. **Underestimation**: The AMSC of a pattern p is smaller than or equal to its cost, $AMSC(p) \leq c(p)$

II. **Monotonicity**: Let p_x and p_y be two patterns, $If\ \ p_x \subset p_y\ \ then\ \ AMSC(p_x) \leq AMSC(p_y)$

III. **Pruning**: For a pattern p, if $AMSC(p) >$ maxcost, then pattern p can be eliminated as well as its supersequences.

How to reduce the search space? (2)

We use an upper bound on the occupancy of a pattern p:

$$uo(p) = \frac{1}{sup(p)} \cdot \max_{S_1,\ldots,S_{sup(p)}} \sum_{i=1}^{sup(p)} \frac{psl[S_i] + ssl[S_i]}{sl[S_i]}$$

where $psl[S_i]$, $ssl[S_i]$ and $Sl[S_i]$ is p's length in S_i, the length of the subsequence after p in S_i, and S_i's length, respectively.

e.g. $minsup = 2, p = \langle a, b, c \rangle$

$psl[S_1]$=psl,$[S_4]$=3,$ssl[S_1]$=1, $ssl[S_4]$=1,

$sl[S_1]$=5, $sl[S_4]$=4,

$$uo(p) = \frac{1}{2}\left(\frac{3+1}{5} + \frac{3+1}{4}\right) = 0.9$$

Sid	<Activity : cost>	Utility
S_1	<(**a:4**)(**b:2**)(e:4)(**c:4**)(d:5)>	...
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	...
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	...
S_4	<(**a:2**)(**b:2**)(**c:1**)(f:2)>	...

Properties of *uo*

$$uo(p) = \frac{1}{sup(p)} \cdot \max_{S_1,\ldots,S_{sup(p)}} \sum_{i=1}^{sup(p)} \frac{psl[S_i] + ssl[S_i]}{sl[S_i]}$$

I. **Overestimation**: The *uo* of a pattern *p* is greater than or equal to its occupancy, $uo\,(p) \geq occup(p)$

II. **Anti-monotonicity**: Let p_x and p_y be two patterns, $If\ \ p_x \subset p_y\ \ then\ \ uo(p_x) \geq uo(p_y)$

III. **Pruning**: For a pattern p, if $uo(p) < minoccup$, then pattern p can be eliminated as well as its supersets.

How to reduce the search space? (3)

We use an upper bound on the utility of a pattern p in a numeric DB:

$$upperu = \frac{1}{minsup} \sum_{i=1,2,\ldots,n} u(p, S_i)$$

Sid	<Activity : cost>	Utility
S_1	<(**a:4**)(**b:2**)(e:4)(**c:4**)(d:5)>	40
S_2	<(b:3)(c:2)(f:1)(d:1)(e:2)>	50
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	60
S_4	<(**a:2**)(**b:2**)(**c:1**)(f:2)>	70

e.g. $minsup = 2$ $\qquad p = \langle a, b, c \rangle$

$u(p, S_1) = 40 \qquad u(p, S_4) = 70$

$$upperu(p) = \frac{1}{2}(40 + 70) = 55$$

Properties of *upperu*:

$$upperu = \frac{1}{minsup} \sum_{i=1,2,\ldots,n} u(p, S_i)$$

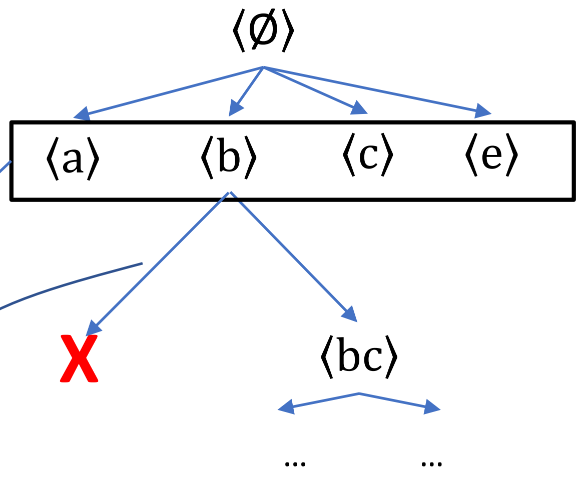
I. **Overestimation**: The *upperu* of a pattern p is greater than or equal to its cost, $upperu(p) \geq u(p)$

II. **Anti-monotonicity**: Let p_x and p_y be two patterns, $If\ \ p_x \subset p_y\ then\ upperu(p_x) \geq upperu(p_y)$

III. **Pruning**: For a pattern p, if $upperu(p) < minutility$, then pattern p can be eliminated as well as its supersets.

The CEPDO and CEPHU Algorithms

Sid	<Activity : cost>	Utility
S_1	<~~(a:4)(b:2)(e:4)(c:4)~~(d:5)>	P/40
S_2	<~~(b:3)(c:2)~~(f:1)(d:1)(e:2)>	N/50
S_3	<(a:2)(f:2)(e:1)(c:3)(d:5)>	P/60
S_4	<~~(a:2)(b:2)(c:1)~~(f:2)>	N/70



P	*sup*	*ac*	*occup*	*cor / tf*
a	3	2.7	0.22	0.5/
b	3	2.3	0.22	-0.5/
…	…	…	…	…

P	*sup* ∧ *AMSC* ∧ *uo* ∧ *upperu(case2 only)*			
a	3	2.67	0.11	56.7
b	3	2.33	0.11	53.3
…	…	…	…	…

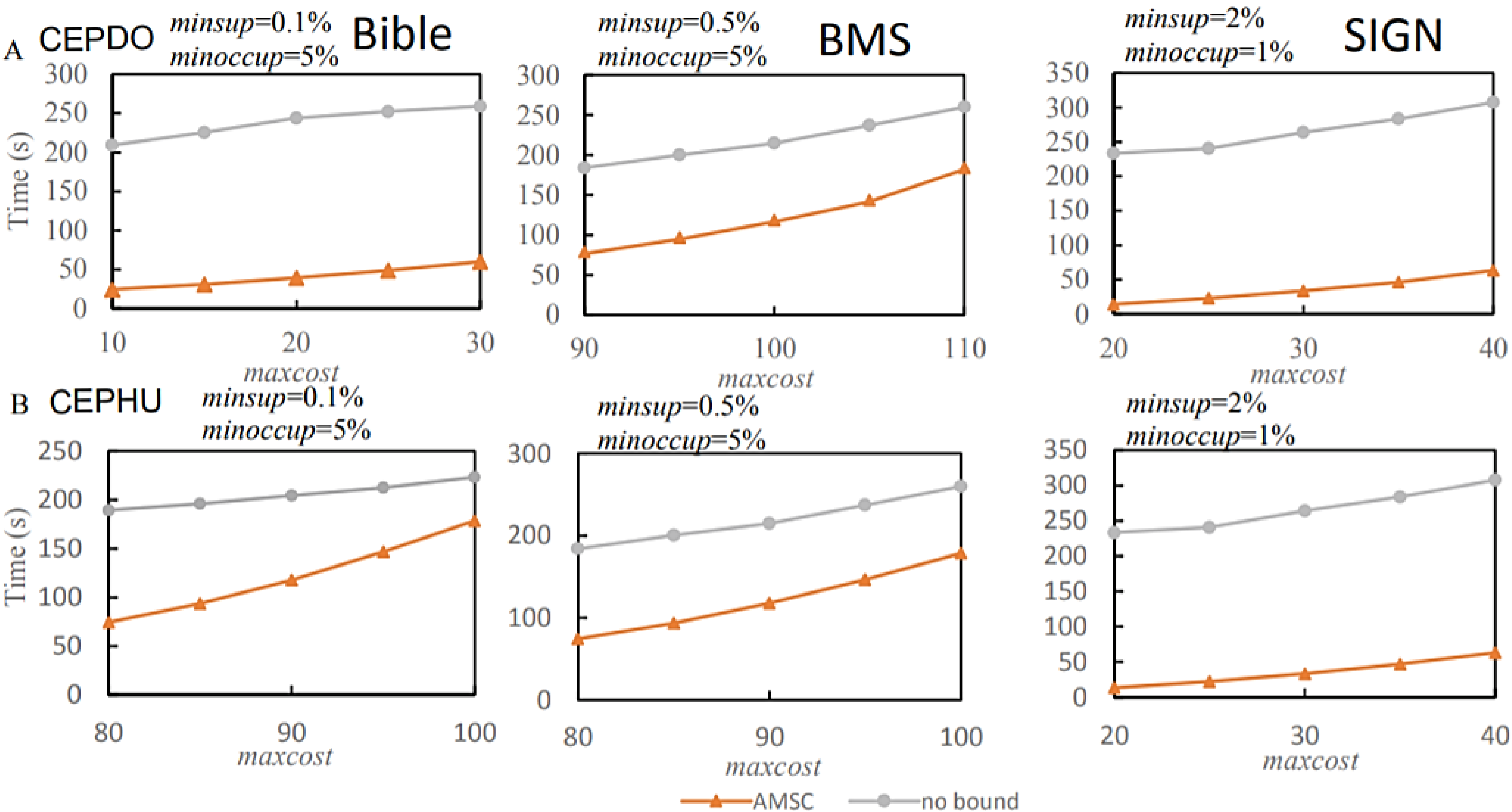
$\mathrm{sup}(p) \geq minsup \quad \wedge$

$\mathrm{AMSC}(p) \leq \mathrm{maxcost} \quad \wedge$

$uo(p) \geq minoccup \quad \wedge$

$upperu(p) \geq minutility$

Execution times of the CEPHU and CEPDO algorithms



BMS, Bible and SIGN are benchmark datasets

Case study 1: binary e-learning DB

Database

- 115 students
- A **sequence** is a series of learning sessions, e_1 to e_6.
- **Cost**: time to complete a session.
- **Utility**: to *pass* or *fail* the final exam.

Cost-efficient patterns

Pattern	Correlation	Average Cost	Support
$\langle e_1, e_6 \rangle$	0.210	250.2	39
$\langle e_1, e_2, e_5, e_6 \rangle$	0.209	485.7	34
$\langle e_2, e_6 \rangle$	0.208	298.4	41
$\langle e_1, e_2, e_6 \rangle$	0.204	391.9	36
$\langle e_1, e_5, e_6 \rangle$	0.194	344.3	37
$\langle e_6 \rangle$	0.193	157.2	50
$\langle e_1, e_4 \rangle$	−0.004	169.1	41
$\langle e_1, e_5 \rangle$	0.002	186.0	41
$\langle e_2, e_3 \rangle$	0.001	284.1	40
$\langle e_3, e_4, e_5, e_6 \rangle$	0.001	469.5	40
$\langle e_1, e_4, e_5 \rangle$	0.003	263.2	38
$\langle e_1, e_2, e_4 \rangle$	−0.003	311.5	36
$\langle e_2, e_3, e_4 \rangle$	−0.005	358.2	38
$\langle e_5 \rangle$	−0.147	96.3	53
$\langle e_4, e_5 \rangle$	−0.109	171.0	49
$\langle e_1, e_3 \rangle$	−0.099	234.6	37
$\langle e_1, e_3, e_4 \rangle$	−0.081	311.2	35

Case study 2: numeric e-learning DB

Database

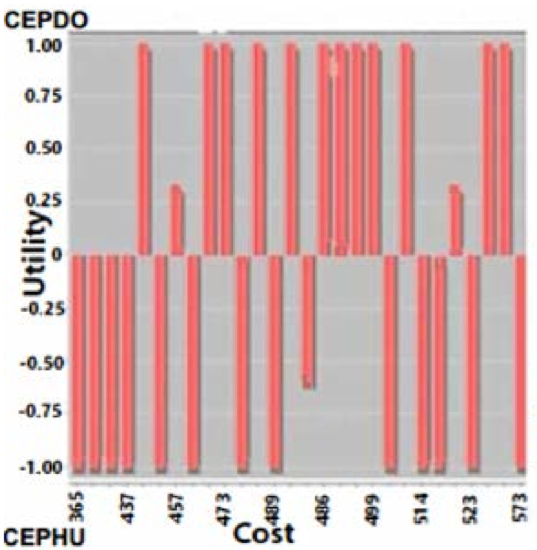
- A sequence is the learning activities of a session.
- **Cost**: time to complete an activity.
- **Utility**: the score at the final exam.

Cost-effective patterns found in learning session 6

Utility	Pattern	trade-off	Average Cost	Support
1	$\langle Study\_Es\_6\_1, Study\_Es\_6\_1, Study\_Es\_6\_1 \rangle$	48.0	57.6	5
2	$\langle Study\_Es\_6\_1, Study\_Es\_6\_1, Study\_Es\_6\_3 \rangle$	15.0	33.0	5
4	$\langle Study\_Es\_6\_1, Study\_Es\_6\_2, Study\_Es\_6\_2 \rangle$	7.0	32.8	6
5	$\langle Study\_Es\_6\_1, Study\_Es\_6\_1 \rangle$	5.1	27.6	9
6	$\langle Study\_Es\_6\_1, Study\_Es\_6\_1, Deeds\_Es\_6\_1 \rangle$	6.0	40.5	6
7	$\langle Study\_Es\_6\_2, Study\_Es\_6\_2 \rangle$	2.9	20.7	11
8	$\langle Study\_Es\_6\_2, Study\_Es\_6\_2, Deeds\_Es\_6\_2 \rangle$	3.6	31.3	6
9	$\langle Study\_Es\_6\_1 \rangle$	1.2	11.0	20
10	$\langle Study\_Es\_6\_1, Deeds\_Es\_6\_2 \rangle$	2.1	21	13
11	$\langle Study\_Es\_6\_2, Study\_Es\_6\_3 \rangle$	1.56	18.2	16
12	$\langle Study\_Es\_6\_2 \rangle$	0.69	8.9	25
13	$\langle Study\_Es\_6\_3 \rangle$	0.64	8.52	25
14	$\langle Deeds\_Es\_6\_2 \rangle$	0.62	9.1	28
15	$\langle Study\_Es\_6\_2, Deeds\_Es\_6\_2, Study\_Es\_6\_3 \rangle$	1.7	27.0	10
16	$\langle FSM\_Es\_6\_1, FSM\_Es\_6\_1, Deeds\_Es\_6\_2, Study\_Es\_6\_3 \rangle$	3.9	64.2	5
17	$\langle Deeds\_Es\_6\_2, Study\_Es\_6\_3 \rangle$	0.89	15.6	16
18	$\langle Study\_Es\_6\_3, Study\_Es\_6\_3 \rangle$	1.0	18.8	9
20	$\langle Deeds\_Es\_6\_1, tudy\_Es\_6\_3, Study\_Es\_6\_3 \rangle$	1.6	32.7	7
21	$\langle FSM\_Es\_6\_3, Study\_Es\_6\_3, Study\_Es\_6\_3 \rangle$	4.5	94.8	6
23	$\langle Deeds\_Es\_6\_2, Study\_Es\_6\_3, Study\_Es\_6\_3 \rangle$	1.2	27.0	6
24	$\langle FSM\_Es\_6\_1, Deeds\_Es\_6\_1, Study\_Es\_6\_3, Study\_Es\_6\_3 \rangle$	3.6	86.3	6
28	$\langle Deeds\_Es\_6\_1, Deeds\_Es\_6\_2, Study\_Es\_6\_3, Study\_Es\_6\_3 \rangle$	1.35	38.0	5

Visualization and Interpretability

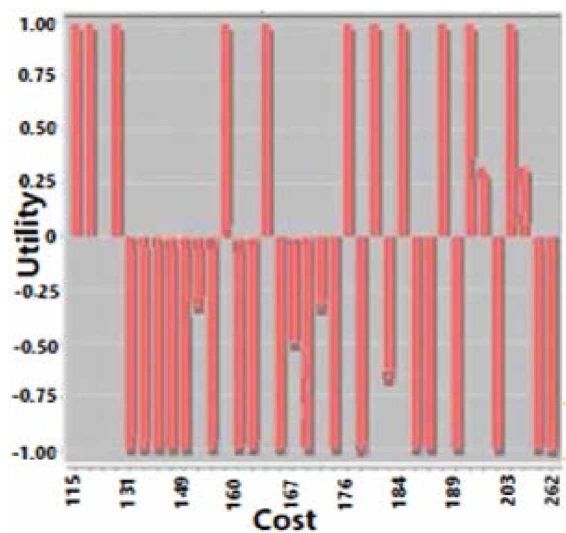
<e₁,e₆>



$cor(<e_1,e_6>)=0.210$

positive correlation

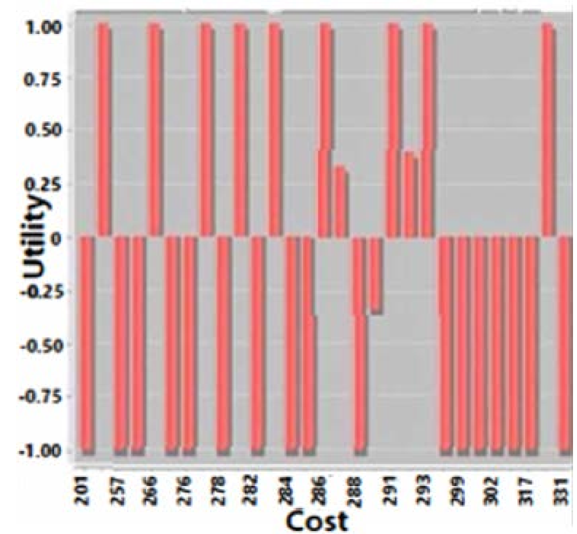
$<e_4,e_5>$



$cor(<e_4,e_5>)=-0.109$

negative correlation

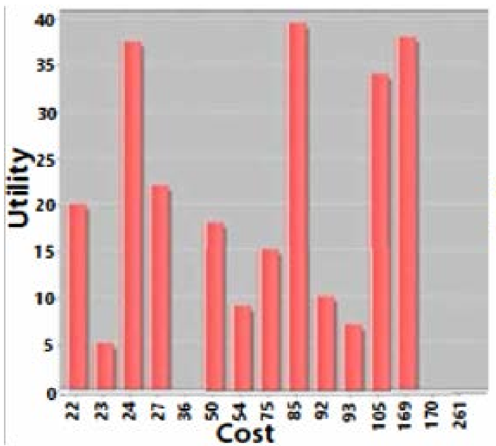
$<e_2,e_3>$



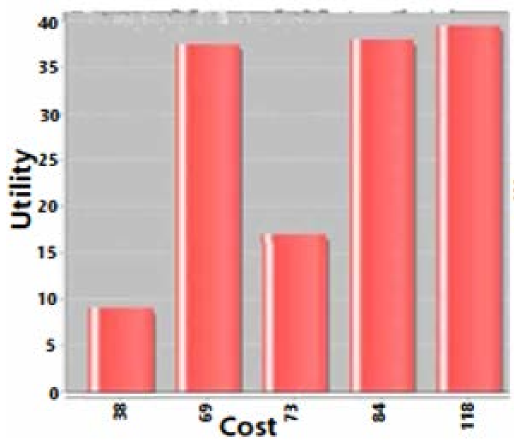
$cor(<e_2,e_3>)=0.001$

no correlation

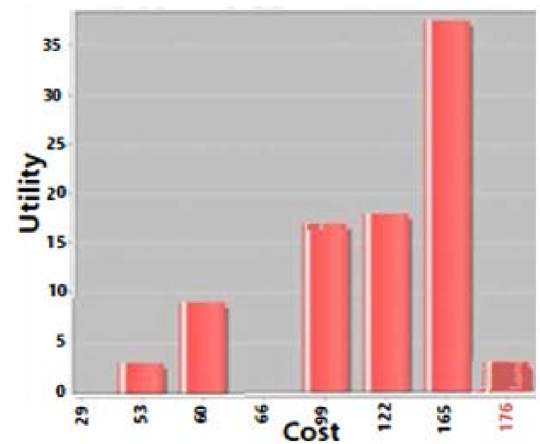
<Study_Es_6_2>
<Deeds _Es_6_2>
<Study_Es_6 _3>



<Deeds_Es_6 _1>
<Deeds_Es_6_2>
<Study_Es_6_3><Study_Es_6_3>



<FSM_Es_6_3>
<Study _Es_6_3><Study_Es_6 _3>



tr(<Study_Es_6_2> <Deeds _Es_6_2><Study_Es_6 _3>)=**1.74**,
cost / utility=15 / 27

tr(<Deeds_Es_6 _1><Deeds_Es_6_2><Study_Es_6_3><Study_Es_6_3>)=**1.35**,
cost / utility=21 / 28

tr(<FSM_Es_6_3> <Study _Es_6_3><Study_Es_6 _3>)=**4.5**,
cost / utility=21 / 94.8

Conclusion

- We proposed to mine **cost-effective patterns**.
- We defined two versions of the problem, for two real-life scenarios.
- We defined efficient algorithms based on a novel **AMSC lower-bound** and **upper-bound** on the utility, to discover patterns efficiently.
- Patterns found in e-learning show that useful patterns can be found having a low cost and a high utility.
- Can help to understand how to use learning material efficiently.

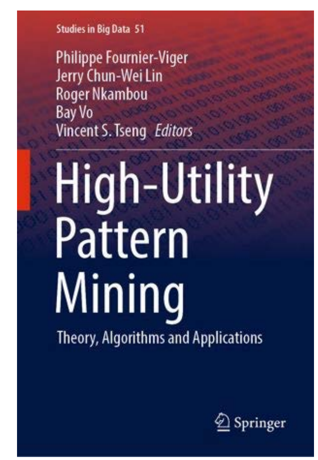
Future Work

Sid	Personal Information	<Activity : cost>	Utility
S_1	<male, college, CS, python,…>	<(a:4)(b:2)(e:4)(c:4)(d:5)>	90
S_2	<female, doctor, Math, java, … >	<(b:3)(c:2)(f:1)(d:1)(e:2)>	80
S_3	<male, senior, CS, C++, …>	<(a:2)(f:2)(e:1)(c:3)(d:5)>	70
S_4	<female, senior, Engineer,C, …>	<(a:2)(b:2)(c:1)(f:2)>	60

Take users' personal information into consideration, giving more reasonable recommendations for different group of people.

e.g. If we need to recommend some courses to learn machine learning, for users who adapt at python, courses related with python should be recommended with priority; for users who are not in CS related major, basic and advance courses should be recommended in order.

Thank you. Questions?



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