4. Data Mining

Solutions

- Exercise 1: Multi MinSup
  - M={3, 2, 5, 4, 6, 1}
  - Read transactions:

Transactions
1, 4, 6
1
1, 5, 6
1, 6
4, 6
1, 2, 3, 5
1, 2, 3, 5
6
1
1, 6

Item	MIS %
1	70
2	17
3	15
4	30
5	30
6	35

Item	Count	SUP %	MIS %
1	8	80	70
2	2	20	17
3	2	20	15
4	2	20	30
5	3	30	30
6	6	60	35

  - L={3, 2, 5, 4, 6, 1}
  - F_1={3, 2, 5, 6, 1}

F	Item	SUP %	MIS %
F1	3	20	15
	2	20	17
	5	30	30
	6	60	35
	1	80	70

4. Data Mining

Solutions

Item	Count	SUP %	MIS %	Transactions
1	8	80	70	1, 4, 6
2	2	20	17	1
3	2	20	15	1, 5, 6
4	2	20	30	1, 6
5	3	30	30	4, 6
6	6	60	35	1, 2, 3, 5
				1, 2, 3, 5
				6
				1
				1, 6

$\varphi = 20\%$

- L={3, 2, 5, 4, 6, 1}
- Candidate gen., K=2
  - {3, 2} : sup(2) = 20%
    20% > MIS(3) = 15 and
    |sup(3) – sup(2)| = |20 - 20| = 0 < φ = 20%
    so {3, 2} is a good candidate
  - {3, 5}: is a good candidate
  - {3, 4}: is a good candidate
  - {3, 6}: is NOT a good candidate (> φ)
  - {3, 1}: is NOT a good candidate (> φ)

4. Data Mining

Solutions

– L={3, 2, 5, 4, 6, 1}
  - {2, 5}: is a good candidate
  - {2, 4}: is a good candidate
  - {2, 6}: is NOT a good candidate (> φ)
  - {2, 1}: is NOT a good candidate (> φ)

$\varphi = 20\%$

Item	Count	SUP %	MIS %
1	8	80	70
2	2	20	17
3	2	20	15
4	2	20	30
5	3	30	30
6	6	60	35

4. Data Mining

Solutions

$\varphi = 20\%$

Transactions
1, 4, 6
1
1, 5, 6
1, 6
4, 6
1, 2, 3, 5
1, 2, 3, 5
6
1
1, 6

- L={3, 2, 5, 4, 6, 1}
  - {5, 4}: sup(4) = 20% < MIS(5) = 30% so {5, 4} is NOT a good candidate
  - {5, 6}: is NOT a good candidate
  - {5, 1}: is NOT a good candidate (> φ)
  - 4 can't be used as seed since sup(4) < MIS(4)
  - {6, 1}: is a good candidate
- C2={{3, 2}, {3, 5}, {3, 4}{2, 5}, {2, 4} {6, 1}}

Item	Count	SUP %	MIS %
1	8	80	70
2	2	20	17
3	2	20	15
4	2	20	30
5	3	30	30
6	6	60	35

4. Data Mining

Solutions

- C2={{3, 2}, {3, 5}, {3, 4}{2, 5}, {2, 4}, {6, 1}}
- Read Transactions to calculate F2
  - F2 ={{3, 2}, {3, 5}, {2, 5}, {6, 1}}

F	Item	SUP %	MIS %
F1	3	20	15
	2	20	17
	5	30	30
	6	60	35
	1	80	70
F2	{3, 2}	20	15
	{3, 5}	20	15
	{2, 5}	20	17
	{6, 1}	40	35

Transactions
1, 4, 6
1
1, 5, 6
1, 6
4, 6
1, 2, 3, 5
1, 2, 3, 5
6
1
1, 6

4. Data Mining

Solutions

- F2 ={{3, 2}, {3, 5}, {2, 5}, {6, 1}}; k = 3
- Join:
  - {3, 2, 5}: MIS(2) < MIS(5) and |sup(2) – sup(5)| = 10 < φ, so it can be joined
  - Nothing else can be joined
- Prune
  - {3, 2} and {3, 5} ∈ F2
  - Since {2, 5} ∈ F2 the head problem is avoided otherwise we should have recorded also sup({2, 5})
- C3 = {3, 2, 5}

Transactions
1, 4, 6
1
1, 5, 6
1, 6
4, 6
1, 2, 3, 5
1, 2, 3, 5
6
1
1, 6

Item	Count	SUP %	MIS %
1	8	80	70
2	2	20	17
3	2	20	15
4	2	20	30
5	3	30	30
6	6	60	35

4. Data Mining

Solutions

minconf = 60%

- Scan transactions, F3={3, 2, 5}
  - Sup({3, 2, 5}) = 20 % > MIS(3) = 15
- Step 2: rule generation from F3={3, 2, 5}
  - Non-empty subsets: {3, 2}, {3, 5}, {2, 5}, {3}, {2}, {5}
  - Possible rules derived from F_3:
    - {3, 2} ⟶ {5}, [sup = 20%, conf = 100%]
    - {3, 5} ⟶ {2}, [sup = 20%, conf = 100%]
    - {2, 5} ⟶ {3}, [sup = 20%, conf = 100%]
    - {3} ⟶ {2, 5}, [sup = 20%, conf = 100%]
    - {2} ⟶ {3, 5}, [sup = 20%, conf = 100%]
    - {5} ⟶ {3, 2}, [sup = 20%, conf = 67%]
  - All are valid since minconf = 60%

F	Item	SUP %	MIS %
F1	3	20	15
	2	20	17
	5	30	30
	6	60	35
	1	80	70
F2	{3, 2}	20	15
	{3, 5}	20	15
	{2, 5}	20	17
	{6, 1}	40	35
F3	{3, 2, 5}	20	15

4. Data Mining

Solutions

- Possible rules derived from F_2:
  - {3} ⟶ {2}, [sup = 20%, conf = 100%]
  - {2} ⟶ {3}, [sup = 20%, conf = 100%]
  - {3} ⟶ {5}, [sup = 20%, conf = 100%]
  - {5} ⟶ {3}, [sup = 20%, conf = 67%]
  - {2} ⟶ {5}, [sup = 20%, conf = 100%]
  - {5} ⟶ {2}, [sup = 20%, conf = 67%]
  - {6} ⟶ {1}, [sup = 40%, conf = 67%]
  - {1} ⟶ {6}, [sup = 40%, conf = 50%]
- Except {1} ⟶ {6}, all are valid

minconf = 60%

F	Item	SUP %	MIS %
F1	3	20	15
	2	20	17
	5	30	30
	6	60	35
	1	80	70
F2	{3, 2}	20	15
	{3, 5}	20	15
	{2, 5}	20	17
	{6, 1}	40	35
F3	{3, 2, 5}	20	15

4. Data Mining

Solutions

SID	Sequence
1	<(dc)b(ac)>
2	<bc(bac)>
3	<(ab)a>

- Exercise 2: GSP
  - Initial step
    - All singleton sequences are <a>, <b>, <c>, <d>
  - General step, k = 1
    - <d> can't form patterns so it can be left out

Cand	Support
<a>	3
<b>	3
<c>	2
<d>	1

4. Data Mining

Solutions

– General step, k = 1, generate length 2 candidates

- First generate 2 event candidates

	<a>	<b>	<c>
<a>	<aa>	<ab>	<ac>
<b>	<ba>	<bb>	<bc>
<c>	<ca>	<cb>	<cc>

- Then generate 1 sequence candidates, each event with 2 items

	<a>	<b>	<c>
<a>		<(ab)>	<(ac)>
<b>			<(bc)>
<c>			

4. Data Mining

Solutions

– k = 2, we have 12 2-length candidates

  - After the second table scan we remain with 7 2-patterns: <ba>, <bc>, <ca>, <cb>, <cc>, <(ab)>, <(ac)>

SID	Sequence
1	<(dc)b(ac)>
2	<bc(bac)>
3	<(ab)a>

Candidate	Support	SIDs
<aa>	1	3
<ab>	0	-
<ac>	0	-
<ba>	3	1, 2, 3
<bb>	1	2
<bc>	2	1, 2
<ca>	2	1, 2
<cb>	2	1, 2
<cc>	2	1, 2
<(ab)>	2	2, 3
<(ac)>	2	1, 2
<(bc)>	1	2

4. Data Mining

Solutions

– Generalization:
  - Join
    – Joining k-1 elements together to obtain k-length candidates
    – Idea by join is that two sequences, s1 and s2 can be joined if after dropping the first item from s1 and the last item from s2, we obtain the same sequence
    – E.g.:
      » <bc> and <ca> can be joined since by dropping b from <bc> and a from <ca> we obtain <c>. The joined result is <bca>
      » <ba> and <(ab)> can also be joined and we obtaine <b(ab)>
  - Prune
    – Is similar to the apriori algorithm
    – <bca> passes pruning only if <bc>, <ba> and <ca> ∈ F2
    – <b(ab)> passes pruning only if <ba>, <bb> and <(ab)> ∈ F2

4. Data Mining

Solutions

- k = 2, generate length 3 candidates
  - <ba>, <bc>, <ca>, <cb>, <cc>, <(ab)>, <(ac)>

	<ba>	<bc>	<ca>	<cb>	<cc>	<(ab)>	<(ac)>
<ba>	-	-	-	-	-	<b(ab)>	<b(ac)>
<bc>	-	-	<bca>	<bcb>	<bcc>		
<ca>	-	-	-	-	-	<c(ab)>	<c(ac)>
<cb>	<cba>	<cbc>	-	-	-	-	-
<cc>	-	-	<cca>	<ccb>	-	-	-
<(ab)>	<(ab)a>	<(ab)c>	-	-	-	-	-
<(ac)>	-	-	<(ac)a>	<(ac)b>	<(ac)c>	-	-

  - Now perform pruning
    - <bc>, <ba> and <ca> $\in F_2$ so <bca> is a good candidate
    - <bcb> is not, because <bb> $\notin F_2$
    - ...
  - After pruning
    - C_3=<b(ac)>, <bca>, <bcc>, <c(ab)>, <c(ac)>, <cba>, <cbc>, <cca>, <ccb>

4. Data Mining

Solutions

– k = 3, we have 9 3-length candidates

- C_3=<b(ac)>, <bca>, <bcc>, <c(ab)>, <c(ac)>, <cba>, <cbc>, <cca>, <ccb>
- After table scan
  F_3 = <b(ac)>, <c(ac)>

Candidate	Support	SIDs
<b(ac)>	2	1, 2
<bca>	1	2
<bcc>	0	-
<c(ab)>	1	2
<c(ac)>	2	1, 2
<cba>	1	1
<cbc>	1	1
<cca>	0	-
<ccb>	0	-

SID	Sequence
1	<(dc)b(ac)>
2	<bc(bac)>
3	<(ab)a>

4. Data Mining

Solutions

– Build C_4 from F_3= <b(ac)>, <c(ac)>

	<b(ac)>	<c(ac)>
<b(ac)>	-	-
<c(ac)>	-	-

- We can't build any 4 length candidate so we remain with <b(ac)>, <c(ac)> as 3-patterns

4. Data Mining

Solutions

- Exercise 3: MA(4)

Data	MA(4)
4,38	
4,19	
4,65	
6,40	4,905
6,26	5,375
13,51	7,705
4,19	7,59
8,41	8,0925
6,50	8,1525
8,43	6,8825
9,87	8,3025
9,56	8,59
6,57	8,6075
9,03	8,7575
10,18	8,835