Institutionen för datavetenskap
Linköpings universitet

EXAM

732A61and TDDD41
Data Mining –
Clustering and Association Analysis

732A75 Advanced Data Mining

June 9, 2020, kl 8-12

Teachers: Patrick Lambrix, José M Pena

This is an individual exam. No help from others is allowed. No uploading or downloading solutions is allowed. No communication regarding the exam is allowed (except with the teachers mentioned above).

You are allowed to use the course literature, the course slides and your own notes.
Answers to the exam questions may be sent to Urkund.

Instructions:
1. Start each question at a new page.
2. Write at one side of a page.
3. Write clearly.
4. If you make assumptions about a question, that are not explicitly stated, you need to write these down. (These assumptions cannot change the exercise or question.)
5. Hand in via LISAM before 12:00. If you have problems handing in via LISAM, send your answers via e-mail to Patrick.lambrix@liu.se latest 12:05 and keep trying to upload afterwards. Handing in after this time will be considered as not handed in.

GOOD LUCK!

1. Clustering by partitioning (1+2+2=5p)

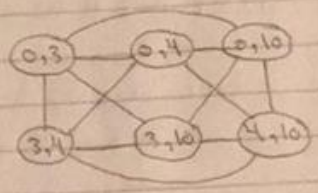
a. Given the data set {0, 3, 4, 10}. Assume we use Euclidean distance and k = 2. Draw the graph representation of the clustering problem.

b. Start at an arbitrary node and show one iteration of the PAM algorithm on the graph. Give all steps in the computation and show at what node that iteration ends.

c. Assume numlocal = 1 and maxneighbor = 2. Start at the same node as in question b and show one iteration of the CLARANS algorithm on the graph. Give all steps in the computation and show at what node that iteration ends.

Q.1 {0, 3, 4, 10} K = 2

a



each node represents a potential solution

b.

- Start at node {0,3} arbitrarily and assign data points to closest medoid and calculate the cost

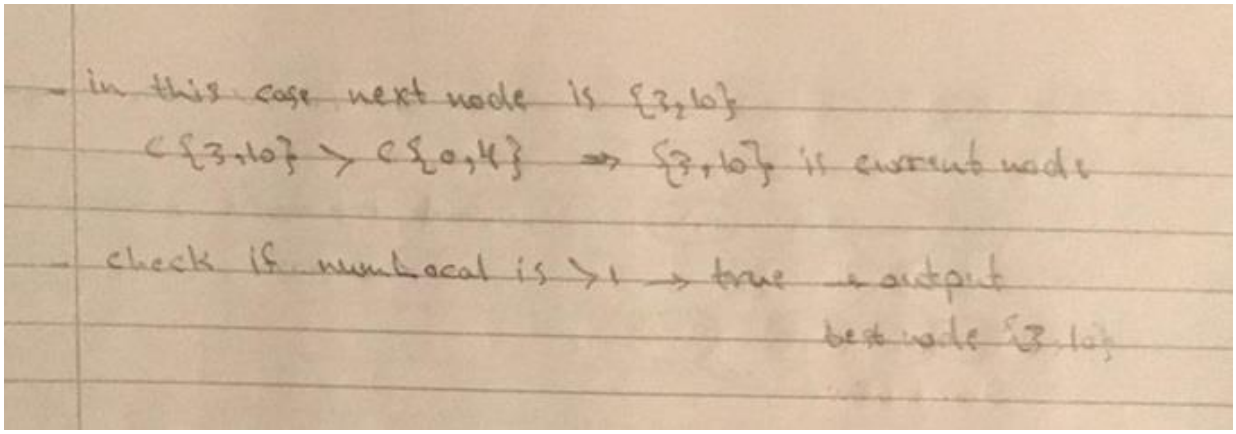
$= \sqrt{(0-0)^2} + \sqrt{(3-3)^2} + \sqrt{(3-4)^2} + \sqrt{(3-10)^2} = 8$

- calculate the total swapping cost TC_{ih} for each pair of non-selected object h and selected object i

	cluster1	cluster2	
$TC_{0,4}$	{0}	{3,4,10}	= 7 − 8 = −1
$TC_{0,10}$	{0,3,4}	{10}	= 7 − 8 = −1
$TC_{3,4}$	{0,3}	{4,10}	= 9 − 3 = 1
$TC_{3,10}$	{0,3,4}	{10}	= 4 − 3 = −4

PAM will choose the node with the minimum swapping cost in this case {3,4} and anew iteration starts

c. - start at {0,3}
- select random neighbor {0,4}
- check cost differential
  if C{0,3} > C{0,4} → {0,4} current node
  else increase the neighbor counter
  select another random neighbor
  * in this case the condition is true
- re-set the neighbor counter to 1 and select a new node
- When the neighbor counter reach 2, it will check if we reached numLocal → if true → output


- in this case next node is {3,10}
c{3,10} > c{0,4} => {3,10} is current node
- check if numbocal is >1 -> true , output
best node {3,10}

2. Hierarchical clustering (2+2+1=5p)

a. Show the different steps of the Agglomerative Hierarchical Clustering algorithm using the dissimilarity matrix below and *single* link clustering. Give partial results after each step.

	1	2	3	4	5
1	0				
2	5	0			
3	9	10	0		
4	3	2	6	0	
5	7	1	4	8	0

b. For the ROCK algorithm:

Assume the similarity matrix below. Assume that cluster C1 contains the data objects A, B and E. Assume that cluster C2 contains the data objects C and D. What is Link(C1,C2)? Show your computations.

	A	B	C	D	E
A	1				
B	0.9	1			
C	0.8	0.7	1		
D	0.1	0.2	0.7	1	
E	0.2	0	0.3	0.4	1

c. In BIRCH, given a cluster with the data points (1,2), (1,3) and (2,2), what is its cluster feature vector?:

Q.2

a.

	1	2	3	4	5
1	0				
2	5	0			
3	9	10	0		
4	3	2	6	0	
5	7	1	4	8	0

	1	{2,5}	3	4
1	0			
{2,5}	5	0		
3	9	4	0	
4	3	2	6	0

- take the most similar pair and merge them together

$d_{(2,5),1} = \min(d_{2,1}, d_{5,1}) = \min(5, 7) = 5$

$d_{(2,5),3} = \min(d_{2,3}, d_{5,3}) = \min(10, 4) = 4$

$d_{(2,5),4} = \min(d_{2,4}, d_{5,4}) = \min(6, 3) = 3$

	1	{2,5,4}	3
1	0		
{2,5,4}	3	0	
3	9	4	0

$d_{(2,5,4),1} = \min(d_{(2,5),1}, d_{4,1}) = \min(5, 3) = 3$

$d_{(2,5,4),3} = \min(d_{(2,5),3}, d_{4,3}) = \min(4, 6) = 4$

	{2,5,4,1}	3
{2,5,4,1}	0	
3	3	0

$d_{(2,5,4,1),3} = \min(d_{(2,5,4),3}, d_{1,3}) = \min(3, 9) = 3$

b. Link is the number of common neighbors between C_1 and C_2

- for C_1:
- Neighbors of A:

$sim(A,A) = 1 \geq 0.6$, $Sim(A,B) = 0.9 \geq 0.6$

$Sim(A,C) = 0.8 \geq 0.6$

∴ neighbors of A are: A, B, C

- Neighbors of B: B, A, C
- " " Σ: Σ

- for C_2:
- neighbors of C: A, B, C, D
" " D: C, D

$$Link(C_1, C_2) = \frac{C_1 \cap C_2}{C_1 \cup C_2} = \frac{3}{5} = 0.6$$

c. $CF = (N, LS, SS)$

where:

N = number of data points = 3

$LS = \sum_{i=1}^{N} \vec{X}_i = (4, 7)$

$SS = \sum_{i=1}^{N} X_i^2 = (6, 17)$

$\Rightarrow CF = (3, (4, 7), (6, 17))$

3. Density-based clustering (2p)

For the following statements say whether they are true or false.
If a statement is true, then prove it. (Observe that an example is not a proof.)
If a statement is false, then give a counterexample.

- If p and q are density connected wrt eps and MinPts, then q is density reachable from q wrt eps and MinPts.
- If p is density reachable from q wrt eps and MinPts, then q is density reachable from p wrt eps and MinPts.
- If p is density reachable from q wrt eps and MinPts, then p and q are density connected wrt eps and MinPts
- If p is directly density reachable from q wrt eps and MinPts, then p and q are density connected wrt eps and MinPts.

Q.3

1- False

for q to be DR from p there need to be a chain of points $p_1, \ldots, p_n$ where $p_1 = q$, $p_n = p$ such that p_{i+1} is DDR from p_i

but if p and q are only density connected there is a point (o) that p and q are only DR from

2- false

because p need to be core to be DR from q

3- True

if p is DR from q → there exist a point (o) that both p and q are DR from → p and q are density connected

4- True

if p is DDR from q → p is DR from q

there exist a common point (o) which both p & q are DR from

4. Different types of data and their distance measures (2+2=4p)

a. What is the distance between Item K and Item L? (no normalization needed)

	A	B	C	D	E	F	G
Item K	(35,20,4)	(2,5,1)	Y	N	Y	N	77
Item L	(40,20,4)	(1,2,1)	Y	Y	Y	N	no-value-available

Attribute A is interval-based and Euclidean distance is used.
Attribute B is interval-based and Manhattan distance is used.
Attributes C and D are binary symmetric variables.
Attributes E and F are binary asymmetric variables.
Attribute G is interval-based.

b. Assume we have categorical data. One method to define a distance between two data objects is (p-m)/p where p is the total number of categorical variables and m is the number of categorical variables for which there is a match between the objects. A second method is to introduce a new asymmetric binary variable for each of the possible values for each of the categorical variables. Give a formula for the distance between two objects in the second method in terms of p and m (where p and m have the same meaning as above; i.e. p is the number of categorical variables - not the number of introduced binary variables, and m is the number of matches in the categorical variables). Show how you obtained the formula.

Q4 a. $A = \sqrt{(35-40)^2 + (20-20)^2 + (4-4)^2} = 5$

$B = |2-1| + |5-2| + |1-1| = 4$

$C = 0$

$D = 1$

$E = 0$

$F = 0$

$G = 0$

$$d(k,L) = \frac{(1)(5) + (1)(4) + (1)(0) + (1)(1) + (1)(0) + (0)(0) + (0)(0)}{5}$$

$$= \frac{10}{5} = 2$$

b. $sim = 1 - \frac{p-m}{p} = \frac{p}{p} - \frac{p-m}{p} = \frac{m}{p}$

$$A \begin{matrix} 1 & 0 \\ \frac{m}{p} & \frac{p-m}{p} \\ 0 & \frac{p-m}{p} \end{matrix} \quad B \begin{matrix} 1 & 0 \\ \frac{m}{p} & \frac{p-m}{p} \\ 0 & 0 \end{matrix}$$

$$d(A,B) = \frac{\frac{p-m}{p} + \frac{p-m}{p}}{\frac{m}{p} + 2\left(\frac{p-m}{p}\right)} = \frac{\frac{2(p-m)}{p}}{\frac{2p-m}{p}} = \frac{2p-2m}{2p-m}$$

5. Apriori algoritm (3+2+2=7p)

a. Run the Apriori algorithm on the following transactional database with minimum support equal to two transactions. Explain step by step the execution.

Transaction id	Items
1	A
2	B,C,D
3	B,D
4	B,C,E
5	C,D
6	C,E
7	F
8	A
9	B,C,D
10	B,D
11	B,C,E
12	C,D
13	C,E

b. Run the Apriori algorithm on the previous transactional database with minimum support equal to two transactions, and the following additional constraint: Find the frequent itemsets that contain the itemset {B, C}. Explain step by step the execution. Make clear when and how the constraint is used. Incorporate the constraint into the algorithm, i.e. do not simply run the algorithm and afterwards consider the constraint.

c. Run the Apriori algorithm on the previous transactional database with minimum support equal to two transactions, and the following additional constraint: Find the frequent itemsets that do NOT contain the itemset {B, C, D}. Explain step by step the execution. Make clear when and how the constraint is used. Incorporate the constraint into the algorithm, i.e. do not simply run the algorithm and afterwards consider the constraint.

Q.5 minSup = 2

a. - first we order the items in alphabetical order, in this case it is already sorted

- Generate candidate list size 1, and calculate the support

C_1	Sup	L_1	output
A	2	A	
B	6	B	A, B, C, D, E
C	8	C	
D	6	D	
E	4	E	
F	1		

F will not go to output because it did not satisfy minSup

C_2	Sup	L_2	output
AB	0	BC	
AC	0	BD	BC, BD, BE, CD, CE
AD	0	BE	
AE	0	CD	
BC	4	CE	
BD	4		
BE	2		
CD	4		
CE	4		
DE	0		

items AB, AC, AD, AE and CE will not go to output because they did not satisfy minSup

-

C_3	Sup	L_3	output
BCD	2	BCD	BCD, BCE
BCE	2	BCE	

* items BDE and CDE are not in C_3 because their subset DE are not in L_2, i.e. subset checking

C_4	Sup	L_4	output
BCDE	0	—	—

- stop no more items

6. - first we check if the constraint is monotone or anti monotone in this case it is monotone since if it is true for A it will be true for B where $A \subseteq B$

- items already ordered

C_1	Sup	const.	L_1	output
A	2	F	A	
B	6	F	B	--
C	3	F	C	
D	6	F	D	
E	4	F	E	
F	1	F		

* because the const. is monotone we need to keep the items that did not satisfy const. but will not go to output

C_2	Sup	const.	L_2	output
AB	0	F	BC	
AC	0	F	BD	BC
AD	0	F	BE	
AE	0	F	CD	
BC	4	T	CE	
BD	4	F		
BE	2	F		
CD	4	F		
CE	4	F		
DE	0	F		

C_3	Sup	Const.	L_3	output
BCD	2	T	BCD	
BCE	2	T	BCE	{BCD, BCE}

* items BDE and CDE did not pass subset checking

C_4	Sup	Const.	L_4	output
BCDE	0	T	-	-

- no items → stop

c

- check if the const. is monotone or antimonotone
in this case it is antimonotone since if it is true
for A it is true for B where $B \subseteq A$

- items already ordered

C_1	Sup	Const.1	Const.2	L_1 output
A	2	F	T	A
B	6	F	T	B
C	8	F	T	C
D	6	F	T	D
E	4	F	T	E
F	1	F	T	

- C_2 | sup | const.1 | const.2 | L_2 | output

AB	0	F	T	BC	
AC	0	F	T	BD	CE
AC	0	F	T	BE	
AE	0	F	T	CD	
BC	4	T	T	CE	
BD	4	F	T		
BE	2	F	T		
CD	4	F	T		
CE	4	F	T		
DE	0	F	T		

- C_3 | sup | const.1 | const.2 | L_3 | output

BCD	2	T	F	BCE	BCE
BCE	2	T	T		

* items BDE and CDE are not in C_2 because they did not pass subset checking

- no more items → stop

6. FP grow algorithm (4p)

Run the FP grow algorithm on the following transactional database with minimum support equal to one transaction. Explain step by step the execution.

Transaction id	Items
1	C, B, A
2	D, C, A
3	B, A
4	B, A
5	D, A
6	D, A

Q.6 1 C, B, A

2 D, C, A minSup = 1

3 B, A

4 B, A

5 D, A

6 D, A

- we count support for each item

A	B	C	D
6	3	2	3

- we take out items that do not satisfy minsup, in this case all items satisfy minsup

- sort the items in support descending order

A	B	D	C
6	3	3	2

1 A, B, C

2 A, D, C

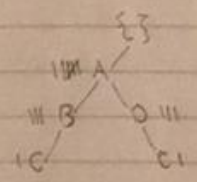
3 A, B

4 A, B

5 A, D

6 A, D

- we create FP tree by using a recursive insert-tree function that increase the count of each item whenever the item exist in database



- construct conditional database where X-conditional database consists of all the prefix paths leading to X in the FP tree

A -

B A: III

D A: III

C AB: 1, AD: 1

- A conditional database

empty

- B-conditional database

1. A
2. A
3. A

* sort in support descending order and prune items that do not satisfy support, in this case no change

* output the frequent 1-itemsets by adding B as a suffix

output | AB |

* construct FP tree and the conditional database

{} item conditional database

III A/ A -

* re-start the process, no more items within B conditional database

- D-conditional database

1. A
2. A
3. A

* After finding frequent 1-itemsets and sorting the transactions in support descending order, we have

1 A
2 A
3 A

* output the frequent 1-itemsets by add D as a suffix

output {AD}

* construct FP tree and the conditional database

{} item conditional database
A A -

* re-start the process, no more items within D conditional database

- C-conditional database
1 A, B
2 A, D

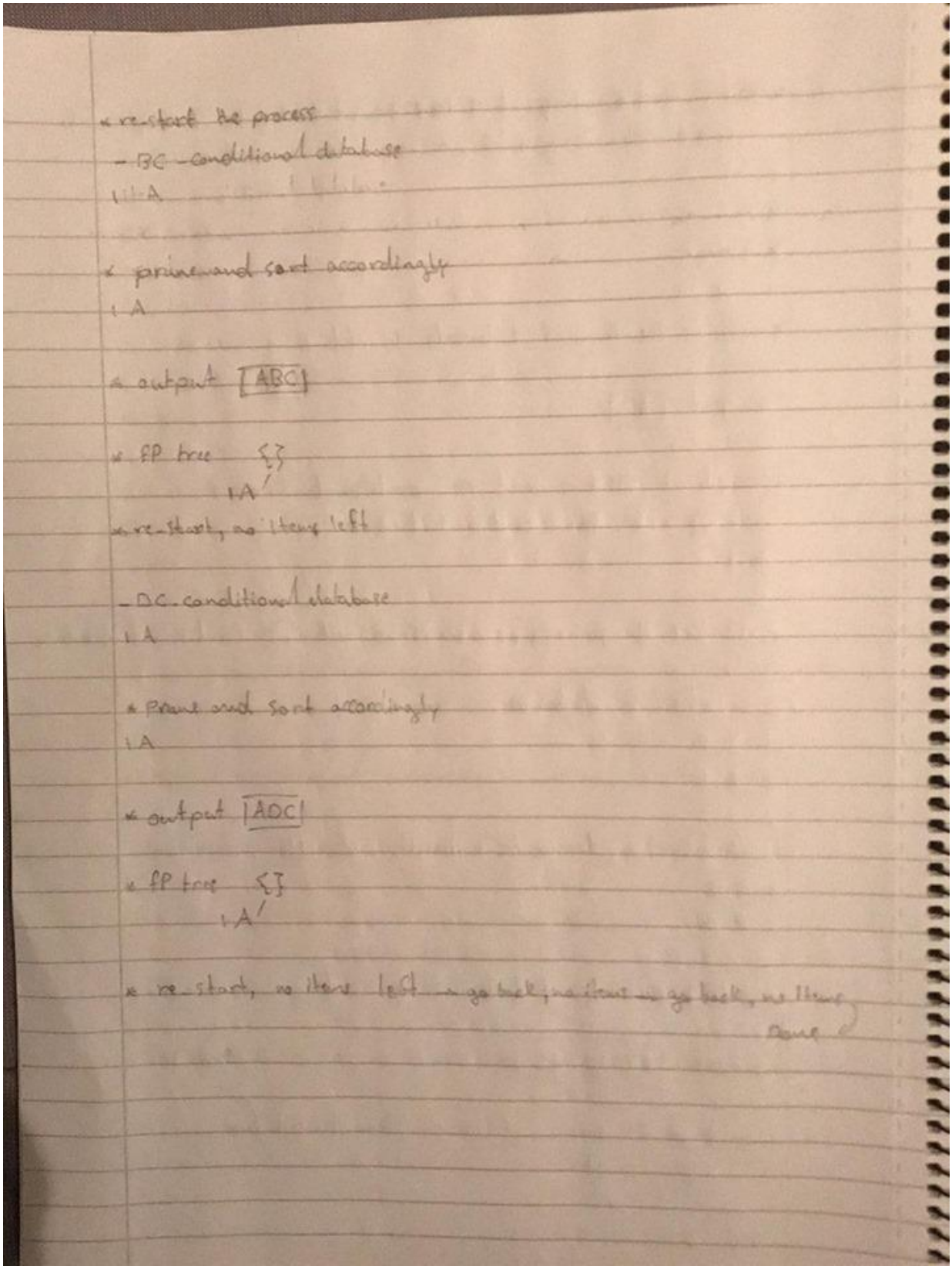
- After finding frequent 1-itemsets and sorting accordingly, we have
1 A, B
2 A, D

* output the frequent 1-itemsets by adding C as a suffix

output {AC, BC, DC}

* construct FP tree and the conditional database

{} item
A:2 A -
B:1 D:1 B A:1
D A:1


* re-start the process
- BC - conditional database
| A
* prune and sort accordingly
| A
* output [ABC]
* FP tree {}
| A'
* re-start, no items left
- DC - conditional database
| A
* Prune and sort accordingly
| A
* output [ADC]
* FP tree {}
| A'
* re-start, no items left - go back, no items - go back, no items
Done

7. Rule generation (4p)

Apply the rule generation algorithm to the frequent itemset {A, B, C} on the database below in order to produce association rules with confidence greater or equal than 50 %. Explain step by step the execution.

Transaction id	Items
1	C, B, A
2	D, C, A
3	A, B
4	A, B
5	A, D
6	A, D

d. 1 C, B, A frequent Itemset: ABC
2 D, C, A minConf: 0.5
3 A, B
4 A, B
5 A, D
6 A, D

- generules (ABC, ABC, 0.5)
X → Y
* if X doesn't result in a rule with minConf so neither does any subset of X, so we start with the largest possible antecedent to take advantage of this rule
A = {AB, AC, CB} antecedent subset of size 2

- AB → C
$conf(AB \rightarrow C) = \frac{sup(ABC)}{sup(AB)} = \frac{1}{3} < 0.5$

* does not generate rule we stop and don't do recursive calls to gen rules for subsets of AB

- AC → B
$conf(AC \rightarrow B) = \frac{sup(ABC)}{sup(AC)} = \frac{1}{2} = 0.5$ → [AC → B] output

we call genrule(ABC, AC, 0.5) to check for subsets of size 1 as antecedents to the rule

- A → CB {A, C}
$conf(A \rightarrow CB) = \frac{sup(ABC)}{sup(A)} = \frac{1}{6} < 0.5$

* does not generate rule so we don't call genrule

- C → AB
$conf(C \rightarrow AB) = \frac{sup(ABC)}{sup(C)} = \frac{1}{2} = 0.5$ → [C → AB] output
* C does not have supsets so we stop and go back to antecedent subset size 2

- $BC \rightarrow A$

$Conf(BC \rightarrow A) = \frac{Sup(ABC)}{Sup(BC)} = \frac{1}{1} \geq 0.5 \rightarrow$ $BC \rightarrow A$ output

* we call generate $(ABC, BC, 0.5)$ to look in antecedent subset size 1 of item $BC \rightarrow \{B, C\}$

- $B \rightarrow CA$

$Conf(B \rightarrow CA) = \frac{Sup(ABC)}{Sup(B)} = \frac{1}{3} < 0.5$

* does not generate a rule, we stop and go to the next item in the subset

- $C \rightarrow AB$

$Conf(C \rightarrow AB) = \frac{Sup(ABC)}{Sup(C)} = \frac{1}{2} = 0.5 \rightarrow$ $C \rightarrow AB$ output

C does not have subsets and all larger subsets have been searched so we stop