Soft Computing
Association Analysis I

Dr. Chun-Hao Chen

Outline

1、Motivation & Problem Definition

2、Association Mining & Properties

3、Advanced Rule Mining Algorithm

4、Related Research Directions

Why Association Mining

✓Extract useful information for managers making a better sales plan

Goods

https://zh-cht.gofreedownload.net/

How to arrange goods into supermarket?

https://zh-cht.gofreedownload.net/

Managers

Relationship between items

Supermarket

Source: https://www.supermarketnews.com/retail-financial/traditional-supermarkets-lose-share-playing-field-shifts

Transactions from Supermarket

- ✓What associations you can find from the following transaction
- ✓ Try to find them and figure out your reasons

TID	Items
D_1	(Dairyland chocolate milk), (Foremost chocolate milk), (Old Mills white bread),(Wonder white bread), (Present chocolate cookies),(Linton green tea beverage).
D_2	(Dairyland chocolate milk), (Foremost chocolate milk), (Dairyland plain milk), (Old Mills white bread), (Wonder wheat bread), (Present lemon cookies), (77 lemon cookies).
D_3	(Old Mills white bread), (Old Mills wheat bread), (77 chocolate cookies), (77 lemon cookies).
D_4	(Dairyland chocolate milk), (Old Mills white bread), (77 chocolate cookies).
D_5	(Old Mills white bread), (Wonder wheat bread).
D_6	(Dairyland chocolate milk), (Foremost chocolate milk), (Linton black tea beverage), (Nestle black tea beverage).

Specific Associations

✓e.g., {**Dairyland chocolate milk, Old Mills white bread**} is a specific association

TID	Items
D_1	(**Dairyland chocolate milk**), (Foremost chocolate milk), (**Old Mills white bread**),(Wonder white bread), (Present chocolate cookies),(Linton green tea beverage).
D_2	(**Dairyland chocolate milk**), (Foremost chocolate milk), (Dairyland plain milk), (**Old Mills white bread**), (Wonder wheat bread), (Present lemon cookies), (77 lemon cookies).
D_3	(Old Mills white bread), (Old Mills wheat bread), (77 chocolate cookies), (77 lemon cookies).
D_4	(**Dairyland chocolate milk**), (**Old Mills white bread**), (77 chocolate cookies).
D_5	(Old Mills white bread), (Wonder wheat bread).
D_6	(Dairyland chocolate milk), (Foremost chocolate milk), (Linton black tea beverage), (Nestle black tea beverage).

3 times out of 6

Generalized Associations

- ✓Focus on the four items: milk, bread, cookies, beverage
- ✓The six transactions can be transformed as follows

TID	Items
D_1	(* * milk), (* * milk), (* * bread),(* * bread), (* * cookies),(* * beverage).
D_2	(* * milk), (* * milk), (* * milk), (* * bread), (* * bread), (* * cookies), (* * cookies).
D_3	(* * bread), (* * bread), (* * cookies), (* * cookies).
D_4	(* * milk), (* * bread), (* * cookies).
D_5	(* * bread), (* * bread).
D_6	(* * milk), (* * milk), (* * beverage), (* * beverage).

{bread, cookies} ➔ 4 times out of 6

A Generalized Association

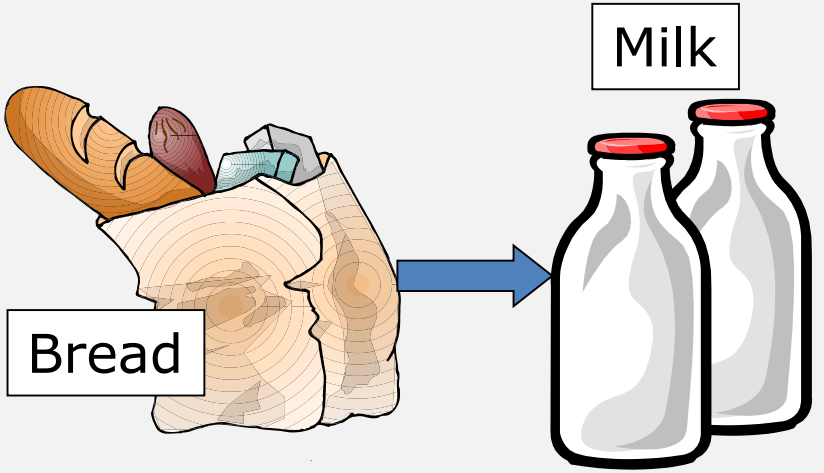
Association Rule Mining

✓Proposed by Agrawal *et al.* (*SIGMOD'93*)

TID	Items
T1	milk, bread, cookies, beverage
T2	milk, bread, cookies
T3	bread, cookies, beverage
T4	milk, bread
T5	milk, bread

minsup

minconf



IF bread is bought

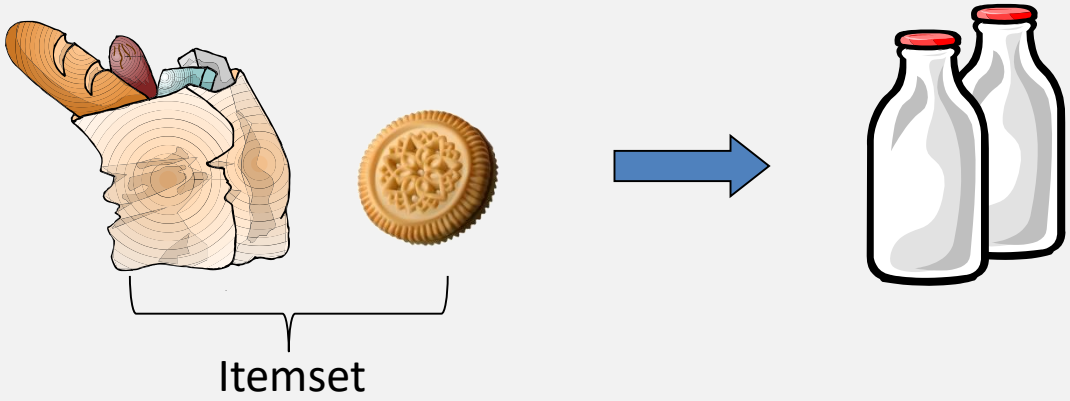
then milk is bought

Problem Definition

✓Given a set of transactions *D*

✓*Goal:*

- To find association rules A → B from the dataset *D* that reach the *minimum support* α and *minimum confidence* λ constraints
- Each of *A* and *B* is an itemsets (a set of items)



Apriori Algorithm

- ✓The well-known algorithm & Proposed by Agrawal et al.
- ✓Three main phases
  - ●Phase 1: Define the two threshold *minsup* and *minconf*
    - ✓e.g., *minsup* = 50%, *minconf* = 50%
  - ●Phase 2: Find large itemsets
  - ●Phase 3: Generate association rules

Example for Phase 2

Database

TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Scan Database →

C_1

Itemset	Sup.
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

C_2

Itemset
{A B}
{A C}
{A E}
{B C}
{B E}
{C E}

Scan Database →

C_2

Itemset	Sup.
{A B}	1
{A C}	2
{A E}	1
{B C}	2
{B E}	3
{C E}	2

C_3

Itemset
{B C E}

Scan Database →

C_3

Itemset	Sup.
{B C E}	2

Large itemsets

L_1

Itemset	Sup.
{A}	2
{B}	3
{C}	3
{E}	3

L_2

Itemset	Sup.
{A C}	2
{B C}	2
{B E}	3
{C E}	2

L_3

Itemset	Sup.
{B C E}	2

Example for Phase 3

Association rules	Confidence
IF BC THEN E	*S(BCE)/S(BC)=2/2*
IF BE THEN C	*S(BCE)/S(BE)=2/3*
IF CE THEN B	*S(BCE)/S(CE)=2/2*
IF B THEN CE	*S(BCE)/S(B)=2/3*
IF C THEN BE	*S(BCE)/S(C)=2/3*
IF E THEN BC	*S(BCE)/S(E)=2/3*
IF A THEN C	*S(AC)/S(A)=2/2*
IF C THEN A	*S(AC)/S(C)=2/3*
IF B THEN C	*S(BC)/S(B)=2/3*
IF C THEN B	*S(BC)/S(C)=2/3*
IF B THEN E	*S(BE)/S(B)=3/3*
IF E THEN B	*S(BE)/S(E)=3/3*
IF C THEN E	*S(CE)/S(C)=2/3*
IF E THEN C	*S(CE)/S(E)=2/3*

Step-by-Step

- ✓Input Database (4 Transactions)
- ✓Parameter setting
  - *minsup* = 50%
  - *minconf* = 50%

Trans. Database	
TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Generate Large Itemsets

✓Generate Candidate 1-itemsets (C1)

Database	
TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Scan Database

e.g.
Item A appears in
TID 100, 300
➔ count = 2

C_1	
Itemset	Count
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

Generate Large Itemsets (Cont.)

✓Generate Large 1-itemsets (L1)

- Check item's count larger than minimum support or not
- Total transactions = 4 → minimum count = 2 (= 0.5*4)

Database	
TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Scan Database

e.g.
Item A appears in TID 100, 300
→ count = 2

C_1	
Itemset	Count
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

Generate Large Itemsets (Cont.)

✓Generate Large 1-itemsets (L_1)

- Check item's count larger than minimum support or not
- Total transactions = 4 → minimum count = 2 (= 0.5*4)

C_1	
Itemset	Count
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

e.g.

Count(A) = 2 ≥ 2

➔ Itemset A is L1

L_1	
Itemset	Count
{A}	2
{B}	3
{C}	3
{E}	3

Generate Large Itemsets (Cont.)

✓Generate Candidate 2-itemsets (C_2) from large 1-itemsets (L_1)

L_1

Itemset	Count
{A}	2
{B}	3
{C}	3
{E}	3

e.g.

Large 1-itemset: A

➔ Generate three C_2

➔ {AB}, {AC}, {AE}

C_2

Itemset
{A B}
{A C}
{A E}
{B C}
{B E}
{C E}

Generate Large Itemsets (Cont.)

✓Generate Large 2-itemsets (L_2)

- Check itemset's count larger than minimum support or not
- minimum count = 2 (= 0.5*4)

C_2

Itemset
{A B}
{A C}
{A E}
{B C}
{B E}
{C E}

Scan Database →

C_2

Itemset	Count
{A B}	1
{A C}	2
{A E}	1
{B C}	2
{B E}	3
{C E}	2

e.g.
Count(AB) = 1 ≤ 2
➔ Delete {A B}

L_2

Itemset	Count
{A C}	2
{B C}	2
{B E}	3
{C E}	2

Generate Large Itemsets (Cont.)

✓Generate Candidate 3-itemsets (C_3) from L_2

L_2

Itemset	Count
{A C}	2
{B C}	2
{B E}	3
{C E}	2

e.g.
{BC} and {BE}
➔generate C_3
➔{BCE}

C_3

Itemset
{B C E}

Generate Large Itemsets (Cont.)

✓Generate Large 3-itemsets (L_3)

- Check itemset's count larger than minimum support or not
- minimum count = 2 (= 0.5*4)

C_3
Itemset
{B C E}

Scan Database →

C_3	
Itemset	Count
{B C E}	2

→

L_3	
Itemset	Count
{B C E}	2

e.g.
Count(BCE) = 2 ≤ 2
➔ {BCE} is L3

Database	
TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Generate Large Itemsets (Cont.)

✓All large itemsets

L_1

Itemset	Count
{A}	2
{B}	3
{C}	3
{E}	3

L_2

Itemset	Count
{A C}	2
{B C}	2
{B E}	3
{C E}	2

L_3

Itemset	Count
{B C E}	2

Generate Association Rules

- ✓Generate Candidate Association Rule
  - ✓E.g. Large 2-itemset BC
  - ✓Two candidate rules

1. B → C
2. C → B

L_1

Itemset	Count
{A}	2
{B}	3
{C}	3
{E}	3

L_2

Itemset	Count
{A C}	2
{B C}	2
{B E}	3
{C E}	2

L_3

Itemset	Count
{B C E}	2

Generate Association Rules (Cont.)

✓Calculate confidence value of each rule

1. B → C

Conf(B → C) = P(BC)/P(B)
= 2/3 = 0.66

2. C → B

Conf(C → B) = P(BC)/P(C)
= 2/3 = 0.66

L_1

Itemset	Count
{A}	2
{B}	3
{C}	3
{E}	3

L_2

Itemset	Count
{A C}	2
{B C}	2
{B E}	3
{C E}	2

L_3

Itemset	Count
{B C E}	2

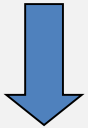
Generate Association Rules (Cont.)

✓Generate Association Rules

- Minimum Confidence = 50%

1. B → C

Conf(B → C) = P(BC)/P(B)
= 2/3 = 0.66



$Rule_1$: B → C, sup=0.5, conf=0.66

2. C → B

Conf(C → B) = P(BC)/P(C)
= 2/3 = 0.66

$Rule_2$: C → B, sup=0.5, conf=0.66

Generate Association Rules (Cont.)

✓After generating all association rules

Association rules	Confidence
IF BC THEN E	*S(BCE)/S(BC)=2/2*
IF BE THEN C	*S(BCE)/S(BE)=2/3*
IF CE THEN B	*S(BCE)/S(CE)=2/2*
IF B THEN CE	*S(BCE)/S(B)=2/3*
IF C THEN BE	*S(BCE)/S(C)=2/3*
IF E THEN BC	*S(BCE)/S(E)=2/3*
IF A THEN C	*S(AC)/S(A)=2/2*
IF C THEN A	*S(AC)/S(C)=2/3*
IF B THEN C	*S(BC)/S(B)=2/3*
IF C THEN B	*S(BC)/S(C)=2/3*
IF B THEN E	*S(BE)/S(B)=3/3*
IF E THEN B	*S(BE)/S(E)=3/3*
IF C THEN E	*S(CE)/S(C)=2/3*
IF E THEN C	*S(CE)/S(E)=2/3*

Discussion

L_2	
Itemset	Sup.
{A C}	2
{B C}	2
{B E}	3
{C E}	2

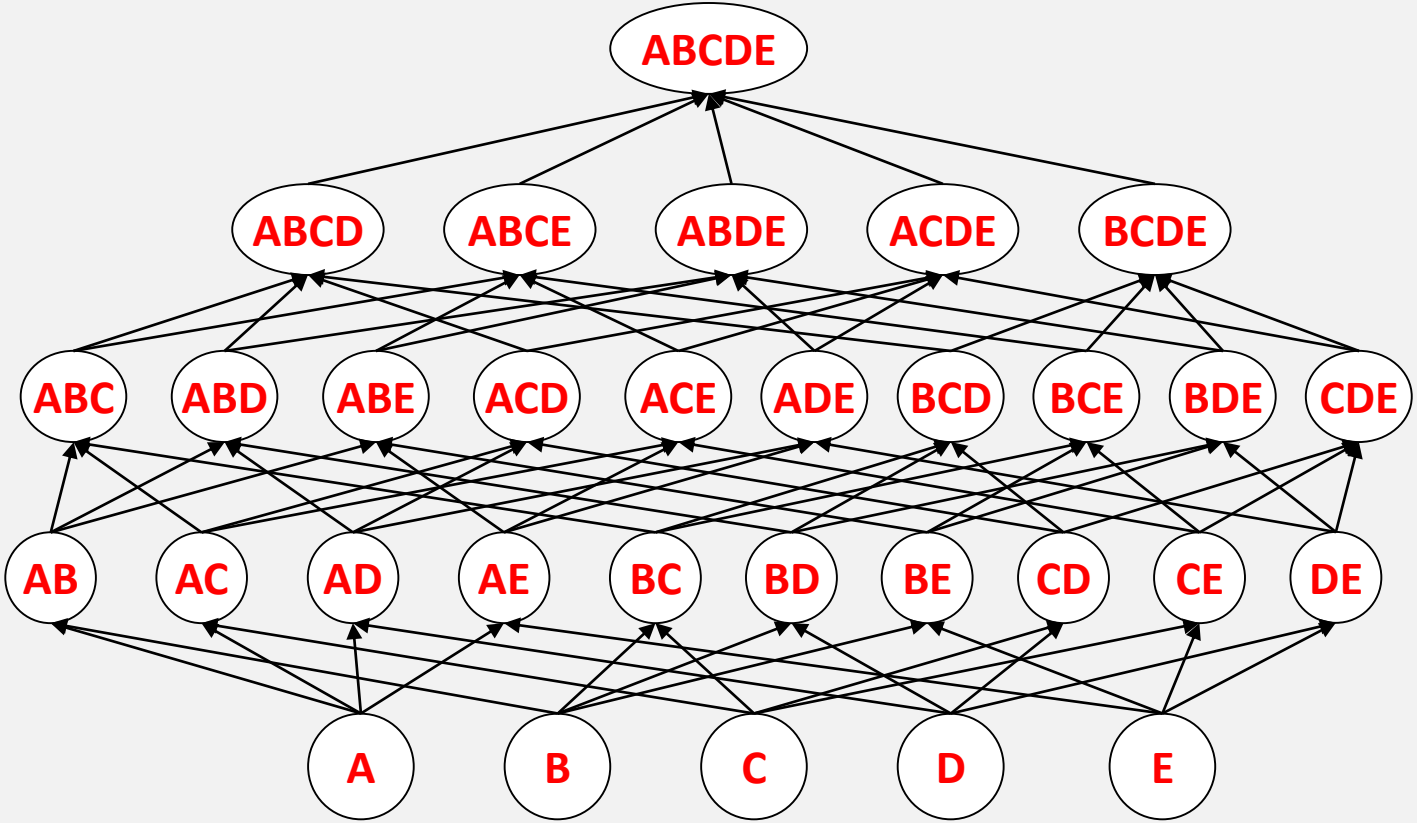
e.g.
{BC} and {BE}
➔generate C_3
➔{BCE}

C_3
Itemset
{B C E}

Why generate only a candidate 3-itemset?

Downward Closure Property

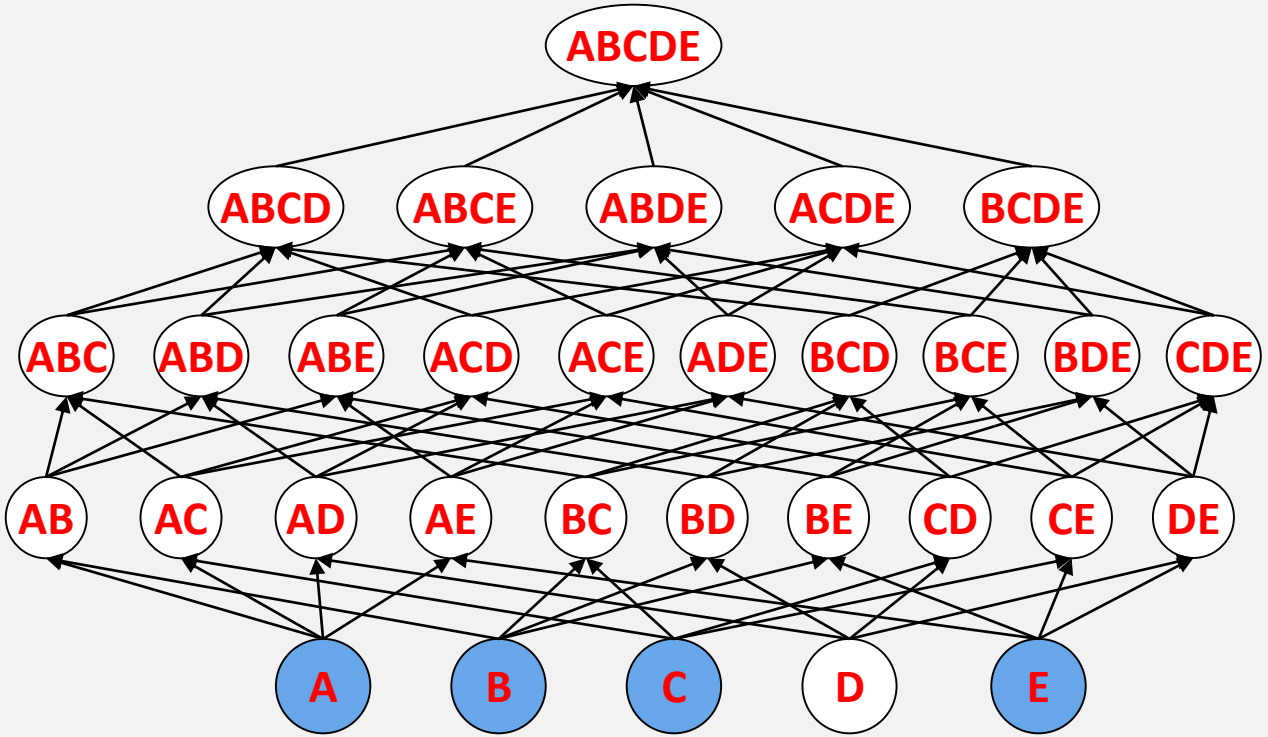
✓Subsets of a large *k*-itemset are also frequent.



Downward Closure Property (Cont.)

✓Continue Previous Example

- L_1: A, B, C and E

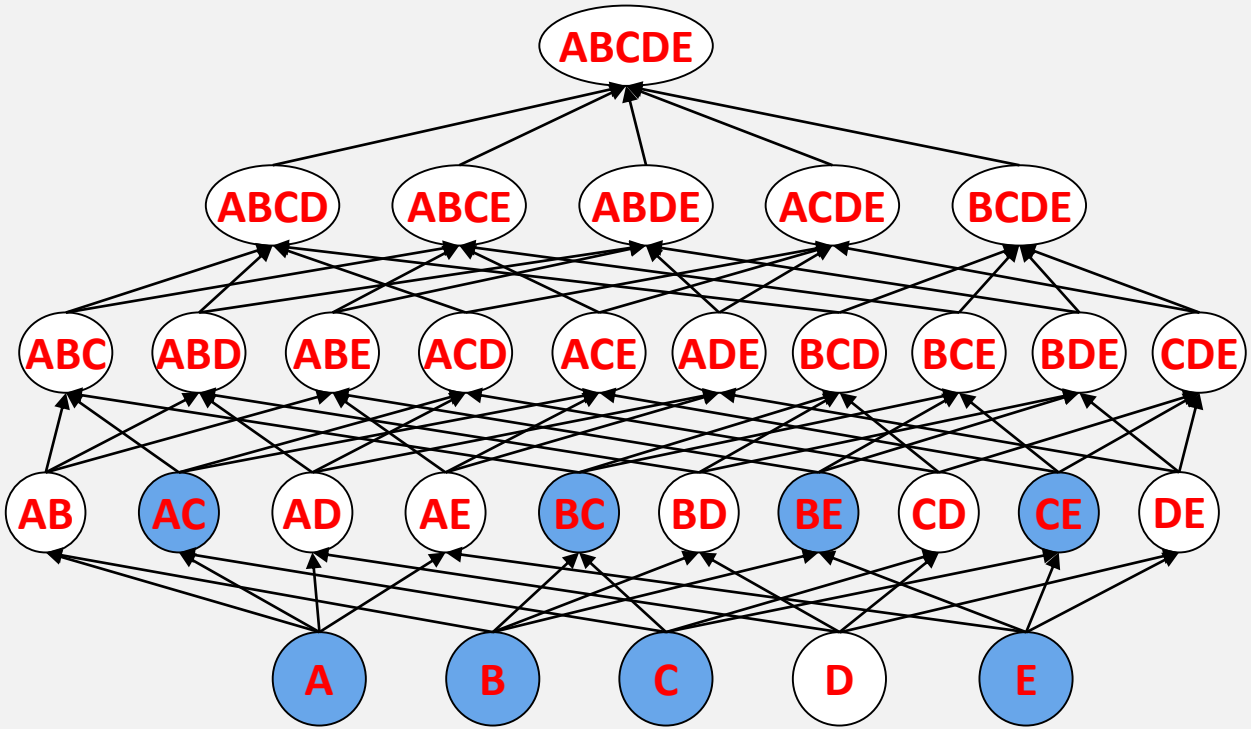


L_1	
Itemset	Count
{A}	2
{B}	3
{C}	3
{E}	3

Downward Closure Property (Cont.)

✓Continue Previous Example

- L_2: AC, BC, BE and CE

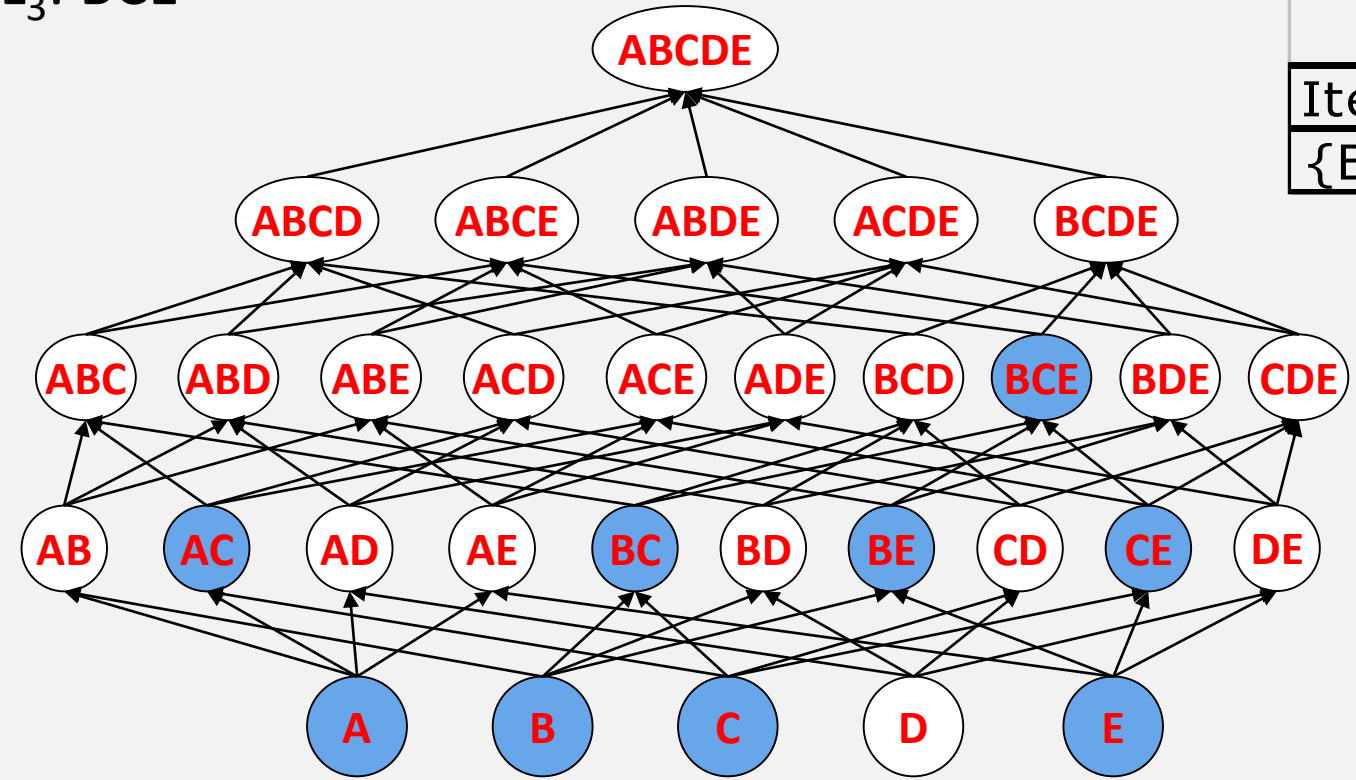


L_2	
Itemset	Count
{A C}	2
{B C}	2
{B E}	3
{C E}	2

Downward Closure Property (Cont.)

✓Continue Previous Example

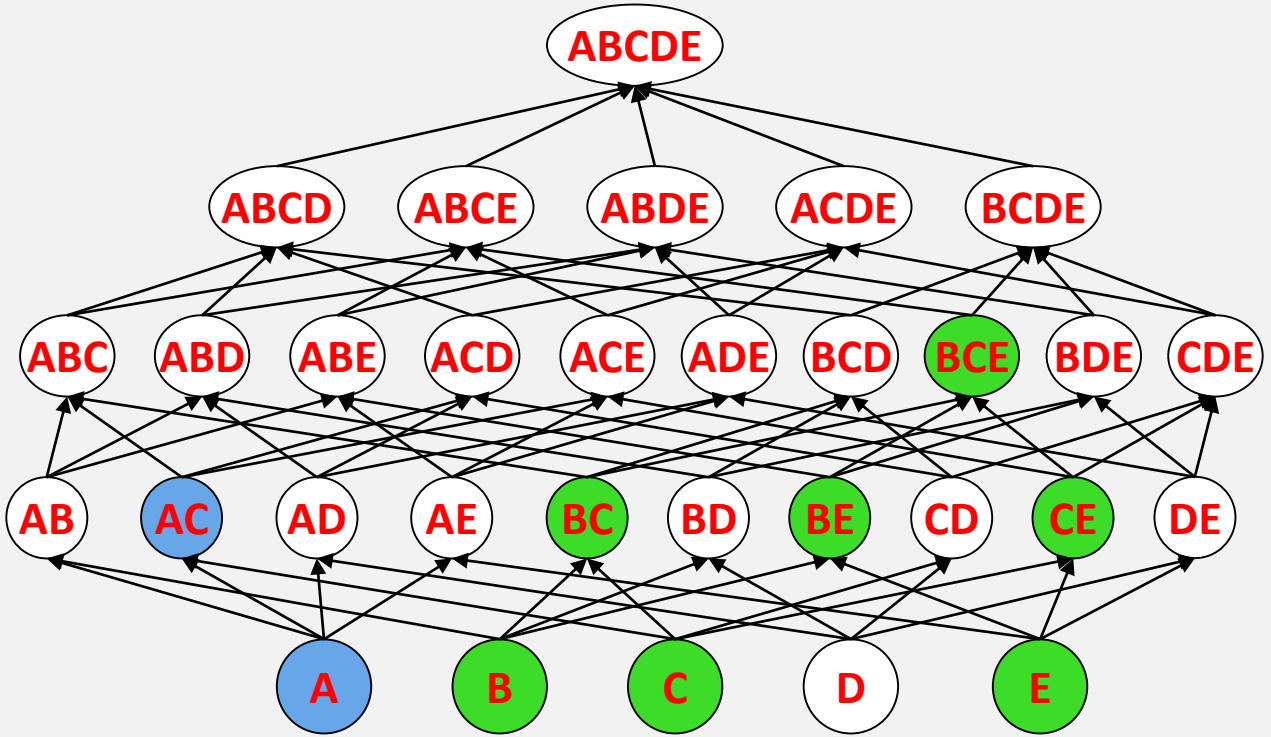
- L_3: BCE



C_3
Itemset
{B C E}

Downward Closure Property (Cont.)

- ✓BCE is a large 3-itemset
  - ●Then, BC, BE, CE, B, C, E are all large itemsets



Flowchart of Association Rule Mining

Apriori algorithm

Database	
TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Min. Sup.

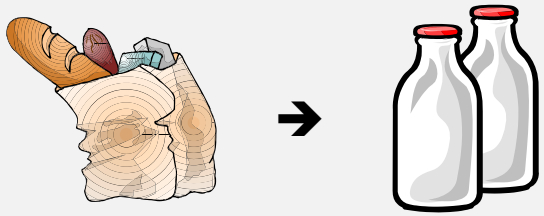
Min. Conf.

Generate Large Itemsets

Candidate Itemsets

Large Itemsets

Association Rule



Discussion

Do you find any problem of Apriori Algorithm?

Problem 1: Scan Database

Database	
TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

Scan Database →

C_1	
Itemset	Sup.
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

C_2
Itemset
{A B}
{A C}
{A E}
{B C}
{B E}
{C E}

Scan Database →

C_2	
Itemset	Sup.
{A B}	1
{A C}	2
{A E}	1
{B C}	2
{B E}	3
{C E}	2

C_3
Itemset
{B C E}

Scan Database →

C_3	
Itemset	Sup.
{B C E}	2

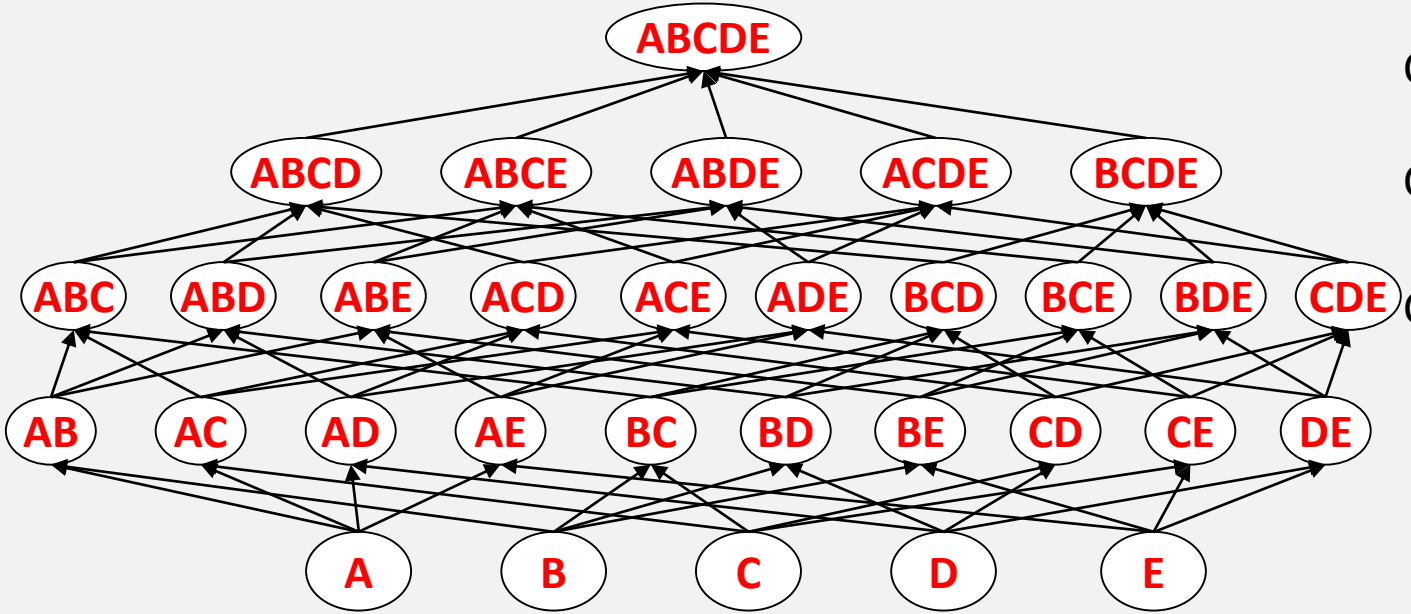
Large itemsets

L_1	
Itemset	Sup.
{A}	2
{B}	3
{C}	3
{E}	3

L_2	
Itemset	Sup.
{A C}	2
{B C}	2
{B E}	3
{C E}	2

L_3	
Itemset	Sup.
{B C E}	2

Problem 2: Candidate Itemsets



Candidate 5-itemsets: $C_5^5 = 1$

Candidate 4-itemsets: : $C_4^5 = 5$

Candidate 3-itemsets: $C_3^5 = 10$

Candidate 2-itemsets: $C_2^5 = 10$

Candidate 1-itemsets: $C_1^5 = 5$

$$C_n^k = \binom{n}{k} = \frac{P_n^k}{k!} = \frac{n!}{k!(n-k)!}$$

$C_2^{100} = 100*99/1*2 = 50*99$

$C_2^{1000} = 1000*999/1*2 = 500*999$

$C_2^{10000} = 10000*9999/1*2 = 5000*9999$

Main Problems of Apriori Algorithm

- ✓Scan the database several times
- ✓Time-consuming for generating candidate itemsets

Could We Avoid Those Problems?

✓Solution

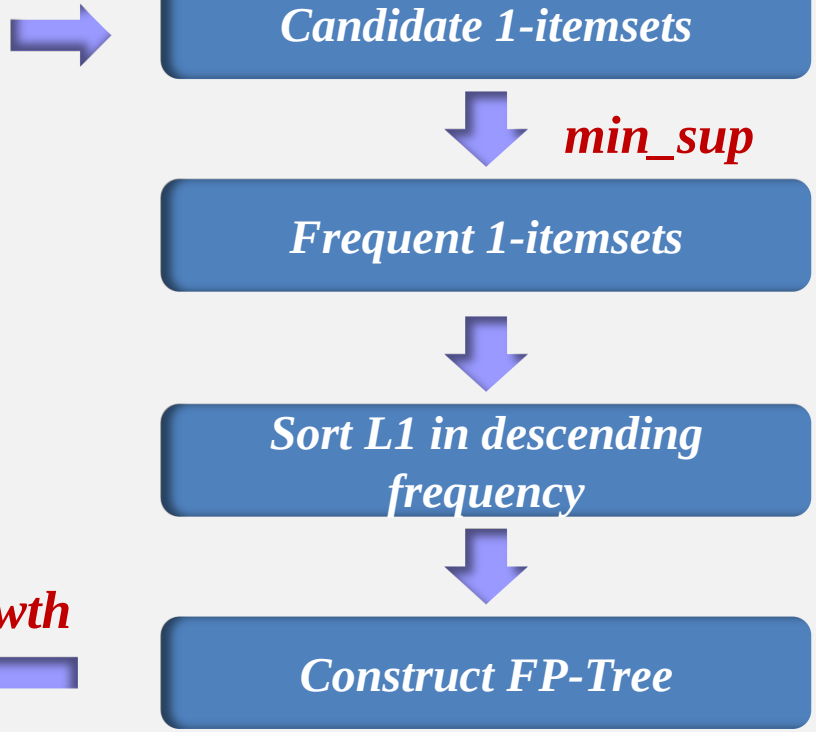
- Proposed by Han et al.
  - Jiawei Han, Jian Pei and Yiwen Yin, “Mining Frequent Patterns without Candidate Generation,” SIGMOD Conference, pp. 1-12, 2000

FP-Growth Algorithm

Flowchart of the FP-growth Algorithm

Transactional Database

Transaction ID	Items Purchased
10	A, B, C
20	A, C
30	A, D
40	B, E, F



FP-Growth Algorithm - Step-by-Step

- ✓Two Phases
  - ●Phase 1: Constructing FP-tree
  - ●Phase 2: Executing FP-Growth mining method
    - ✓Mining from FP-tree

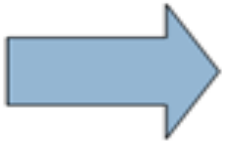
Phase 1: Construction of FP-Tree

- ✓Three steps
  - ●Scan DB to find L1
  - ●Sort L1 in descending frequency
  - ●Scan DB again
    - ✓Construct FP-tree

Step 1

- Scanning DB to find L1
- Example: *min_sup=60%*5=3 trans*

TID	Item bought
100	a, c, d, f, g, i, m, p
200	a, b, c, f, l, m, o
300	b, f, h, j, o, w
400	b, c, k, p, s
500	a, c, e, f, l, m, n, p



item	frequency	item	frequency
a	3	j	1
b	3	k	1
c	4	l	2
d	1	m	3
e	1	o	2
f	4	p	3
g	1	s	1
h	1	w	1
i	1		

L1 = {a, b, c, f, m, p}

Step 2

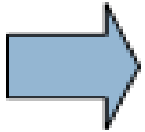
- Sort L1 in descending frequency
  - L1 = {a:3, b:3, c:4, f:4, m:3, p:3}

  L1' = {f:4, c:4, a:3, b:3, m:3, p:3}

min_sup = 3

- Sort DB

TID	Item bought
100	a, c, d, f, g, i, m, p
200	a, b, c, f, l, m, o
300	b, f, h, j, o, w
400	b, c, k, p, s
500	a, c, e, f, l, m, n, p

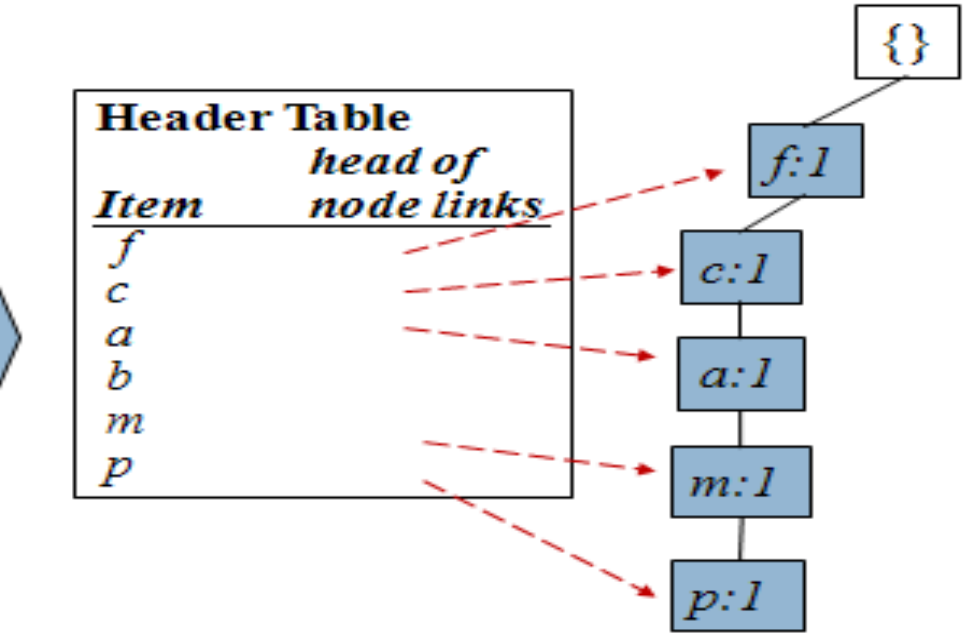


TID	Frequent items
100	f, c, a, m, p
200	f, c, a, b, m
300	f, b
400	c, b, p
500	f, c, a, m, p

Step 3

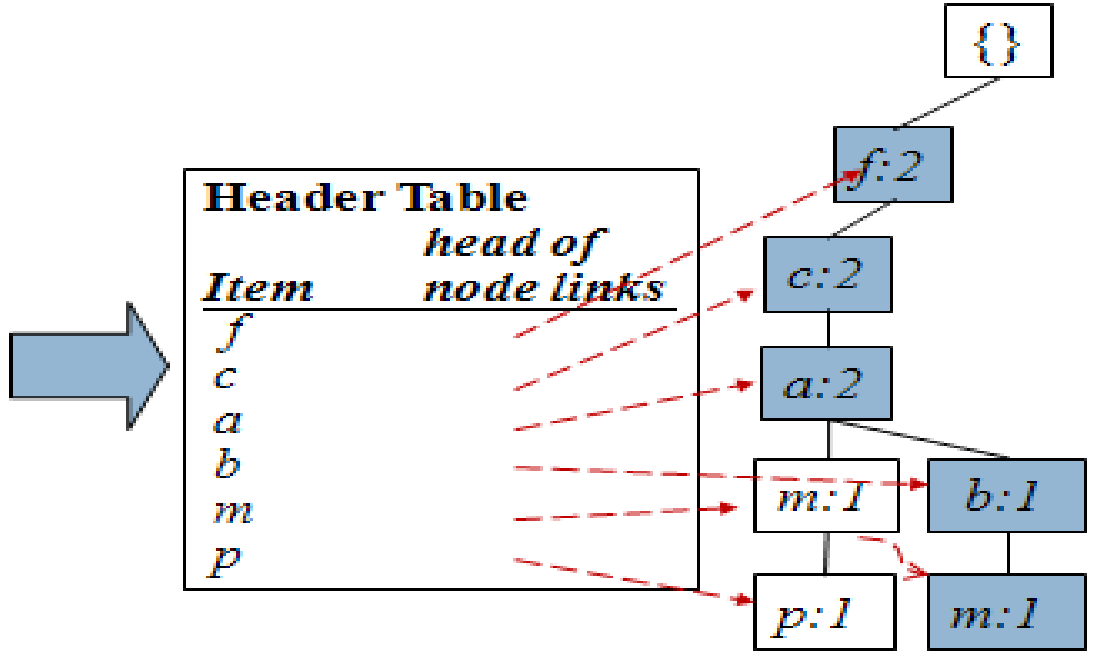
- Scanning DB again
  - Construct FP-tree

TID	Frequent items
100	f, c, a, m, p
200	f, c, a, b, m
300	f, b
400	c, b, p
500	f, c, a, m, p



Step 3 (Cont.)

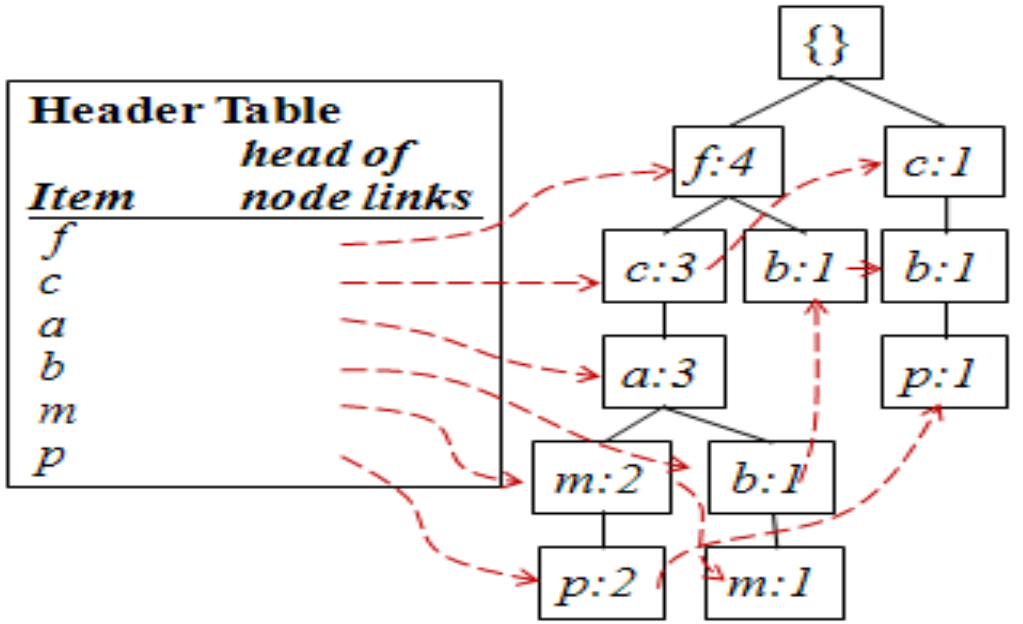
TID	Frequent items
100	f, c, a, m, p
200	f, c, a, b, m
300	f, b
400	c, b, p
500	f, c, a, m, p



Step 3 (Cont.)

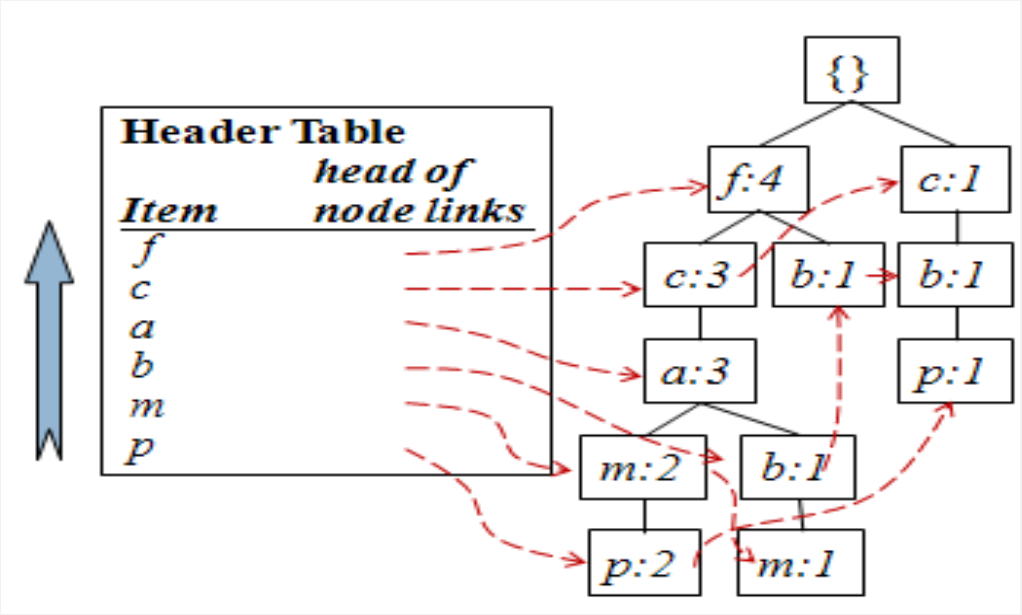
✓After reading all transactions

TID	Frequent items
100	f, c, a, m, p
200	f, c, a, b, m
300	f, b
400	c, b, p
500	f, c, a, m, p



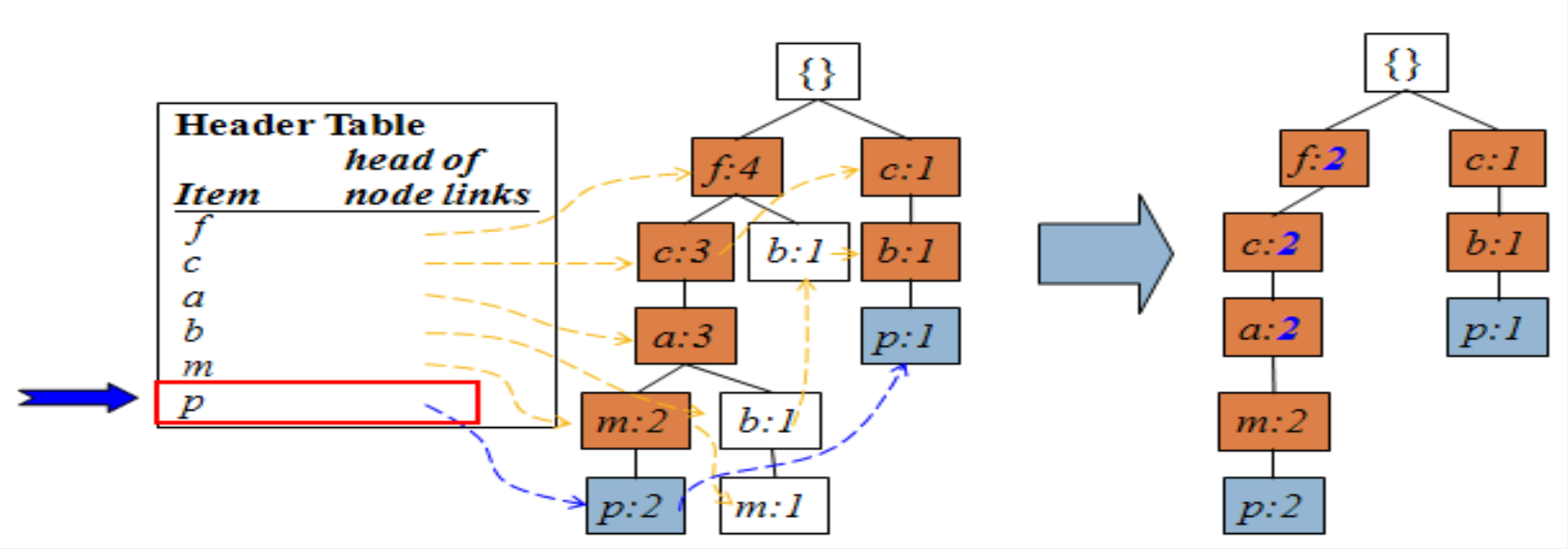
Phase 2: FP-Growth

- ✓Mining frequent patterns from FP-tree
- ✓Processing frequent items
  - ●One by one
  - ●Bottom up

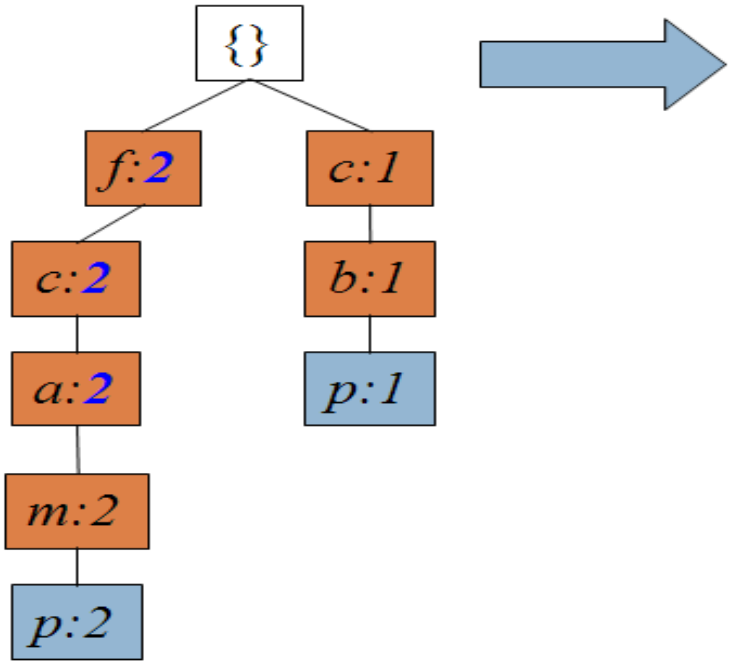


- ✓Each item
  - ●Generating its own conditional FP-tree

Example for *p*



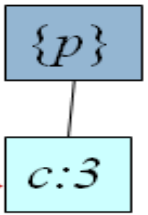
Example for *p* (Cont.)



- P's conditional pattern base
  - {(f:2, c:2, a:2, m:2), (c:1, b:1)}
  - → **{(c:3, f:2, a:2, m:2, b:1)}**

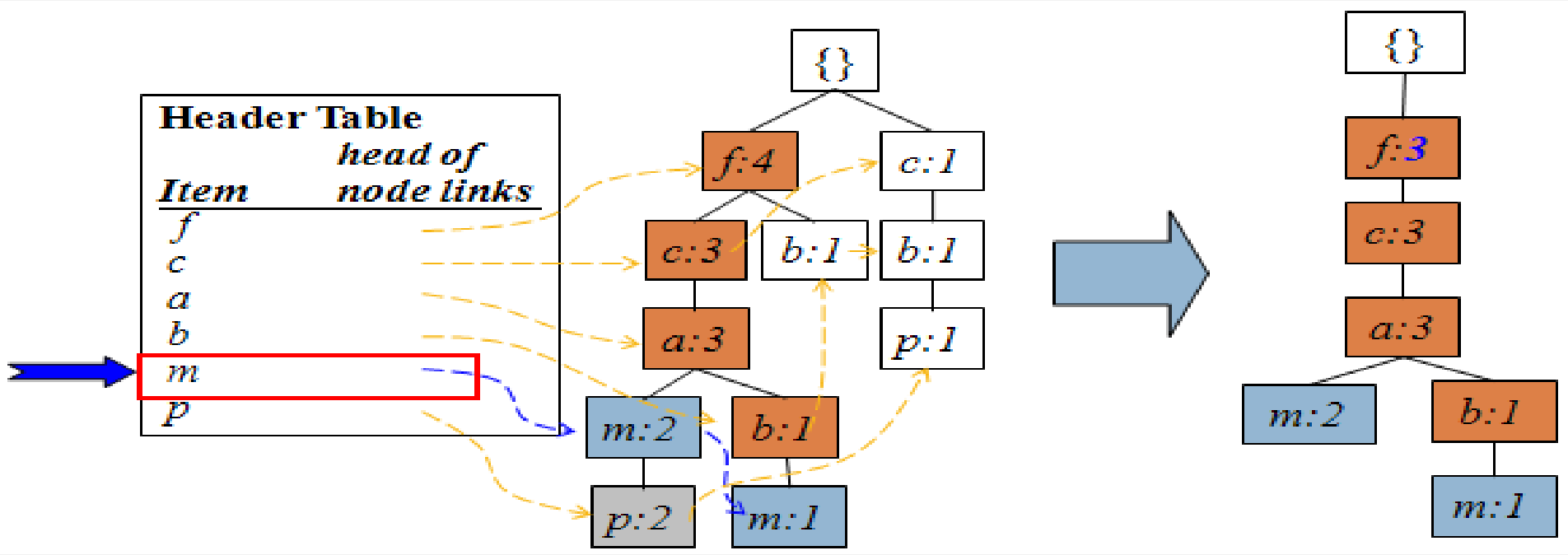
  min_sup = 3
- Constructing p's conditional FP-tree:

Header Table	
Item	*head of node links*
c	

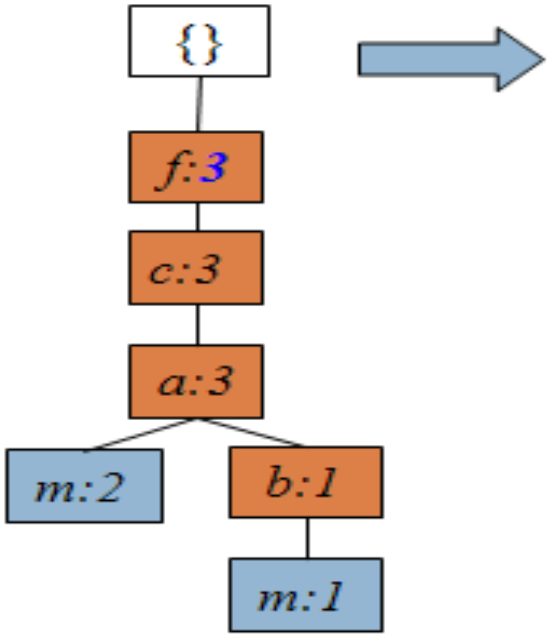


- Frequent patterns with p
  - **{p:3, cp:3}**

Example for *m*



Example for *m* (Cont.)



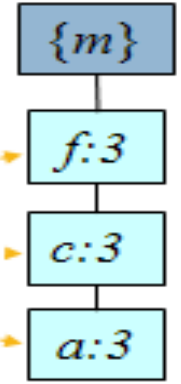
min_sup = 3

- m's conditional pattern base
  - {(f:2, c:2, a:2), (f:1, c:1, a:1, b:1)}

  → **{(f:3, c:3, a:3, b:1)}**

- Constructing m's conditional FP-tree:

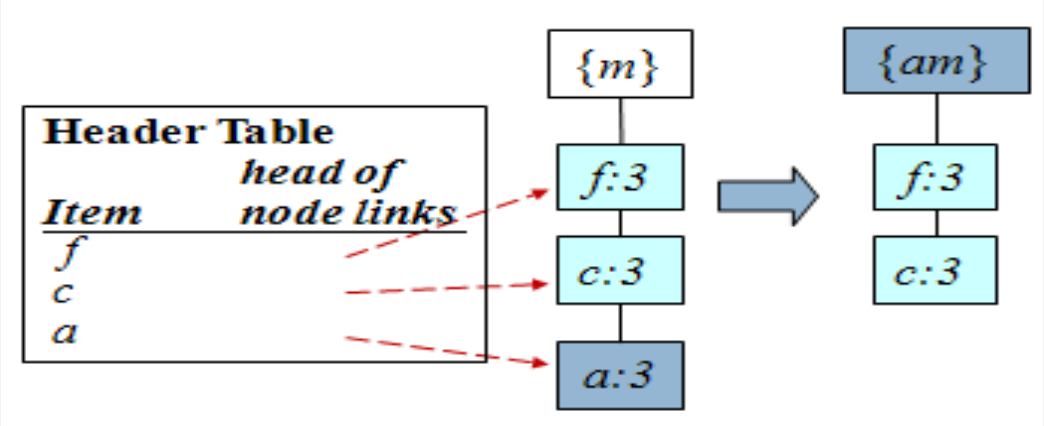
Header Table	
Item	*head of node links*
f	
c	
a	



- Frequent patterns with m
  - {m:3, am:3, cm:3, fm:3}

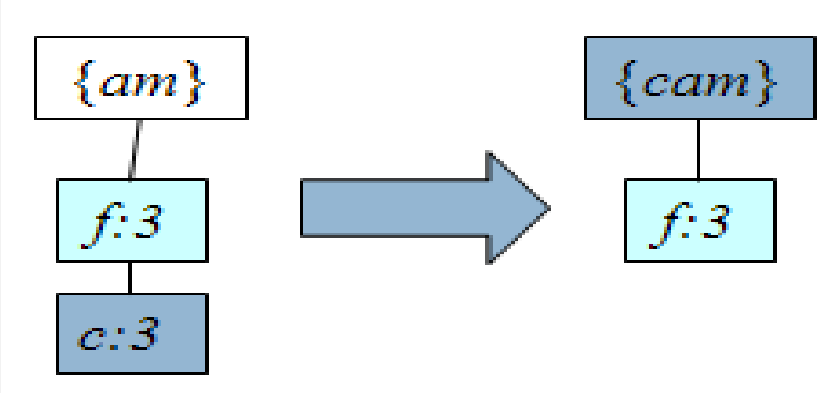
Example for *am*

- ✓Frequent patterns with *m*
  - {m:3, am:3, cm:3, fm:3}
- ✓Recursively constructing conditional FP-tree
  - am, cm, fm
  - Bottom up
- ✓*am* first
  - Prefix
    - ✓(f:3, c:3)
  - Large items
    - ✓f:3, c:3
  - Conditional FP tree for *am*
  - Large itemsets with *am*
    - ✓cam:3, fam:3



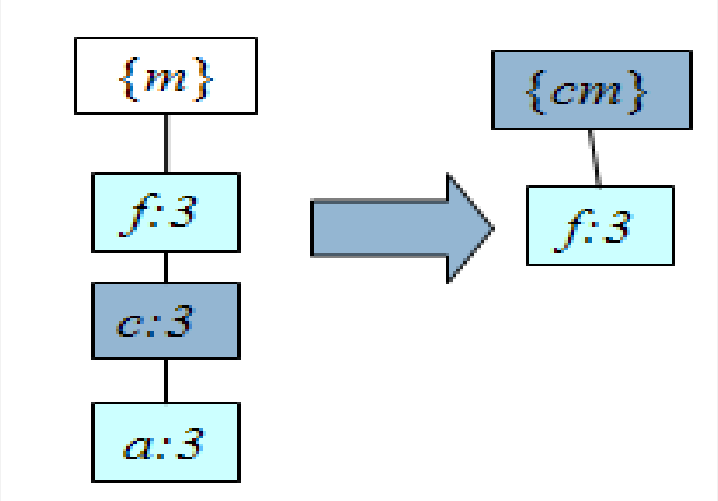
Example for *cam* and *fam*

- ✓Frequent patterns with *am*
  - {am:3, cam:3, fam:3}
- ✓Recursively constructing conditional FP-tree
  - cam, fam
  - Bottom up
- ✓cam first
  - fcam:3
- ✓fam next
  - null



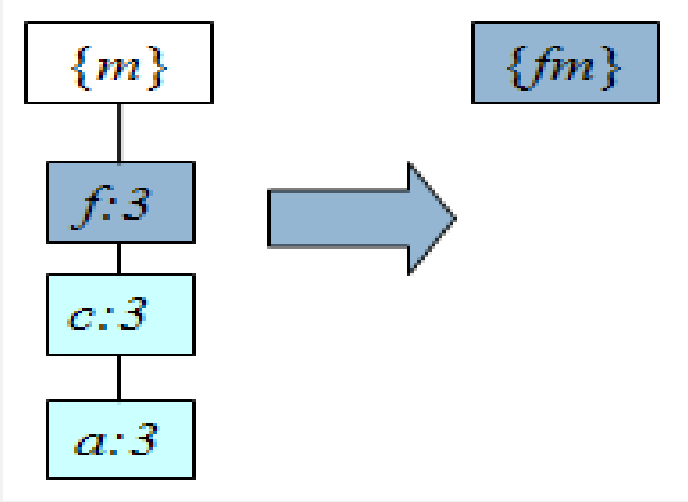
Example for *cm*

- ✓Frequent patterns with *m*
  - ●{m:3, am:3, cm:3, fm:3}
- ✓*cm*
  - ●Prefix
    - ✓(f:3)
  - ●Large items
    - ✓f:3
  - ●Conditional FP tree for *cm*
  - ●Large itemsets with *cm*
    - ✓fcm:3



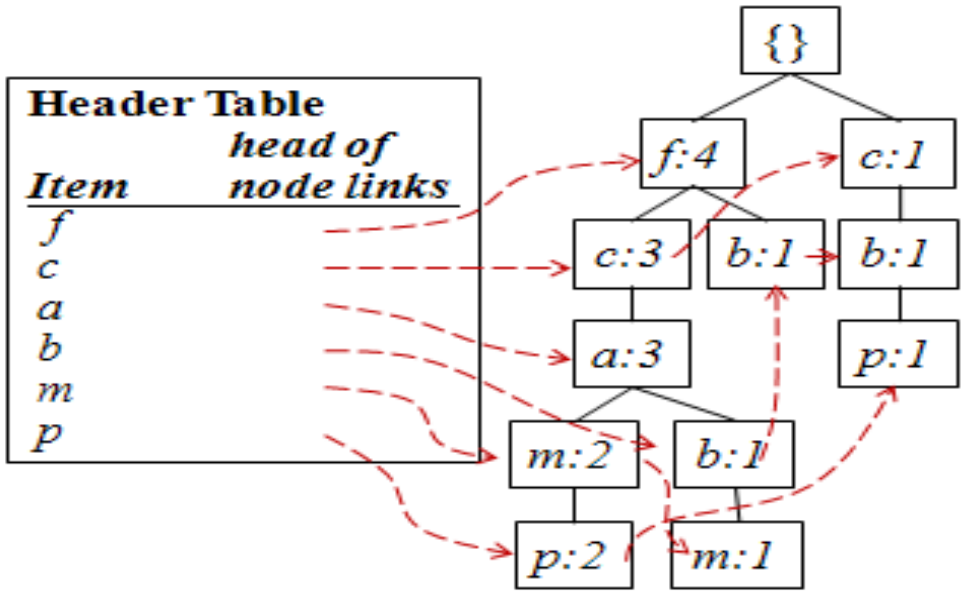
Example for *fm*

- ✓Frequent patterns with *m*
  - ●{m:3, am:3, cm:3, fm:3}
- ✓fm
  - ●Null



- ✓All large itemsets with *m*
  - ●m:3, fm:3, cm:3, am:3, fcm:3, fam:3, cam:3, fcam:3

Result



Item	frequent itemsets
f	f:4
c	c:4, fc:3
a	a:3, ca:3, fa:3, fca:3
b	b:3
m	m:3, fm:3, cm:3, am:3, fcm:3, fam:3, cam:3, fcam:3
p	p:3, cp:3

Summary

- ✓**A Novel data structure → Frequent-Patten tree (FP-tree)**
  - Compressing database to store frequent patterns
  - Saving costly scanning database
- ✓**FP-growth mining method**
  - Efficient mining frequent patterns from FP-tree
- ✓**Without candidate generation**
  - Faster than the Apriori algorithm
- ✓ **Scan database only twice**

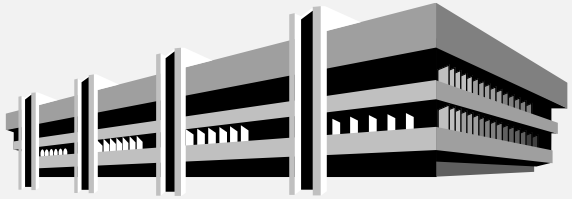
Related Research Direction

Binary Association Rule

Could you extend the association rule to other types of association rule?

Problems of Association Rule

- ✓In real world applications
  - ●Quantitative value may exist in transaction



Supermarket

TID	Items
T1	(milk, 6); **(bread, 4); (cookies, 7), (beverage, 7).**
T2	(milk, 7); **(bread, 7); (cookies, 12).**
T3	**(bread, 8); (cookies, 12); (beverage, 6).**
T4	(milk, 2); **(bread, 3).**
T5	(milk, 3); **(bread, 8).**

Quantitative value

First problem

➔ How to handle Quantitative Transactions?

Quantitative Association Rules

- ✓Proposed by Srikant *et al.* (SIGMOD'96)
  - ●Extend from Apriori Algorithm

TID	Items
T1	**(Age: 50, Car, 5)**
T2	**(Age: 24, Car, 1)**
T3	**(Age: 28, Car, 1)**
T4	**(Age: 22, Car, 0)**
T5	**(Age: 35, Car, 4)**

TID	Items
T1	**(Age: [50, 59], Car: [3, 5])**
T2	**(Age: [20, 29], Car: [0, 1])**
T3	**(Age: [20, 29], Car: [0, 1])**
T4	**(Age: [20, 29], Car: [0, 1])**
T5	**(Age: [30, 39], Car: [3, 5])**

IF Age is [20,… 29], then Number of Car is [0, 1]. Sup. = 60%, Conf. = 66.6%

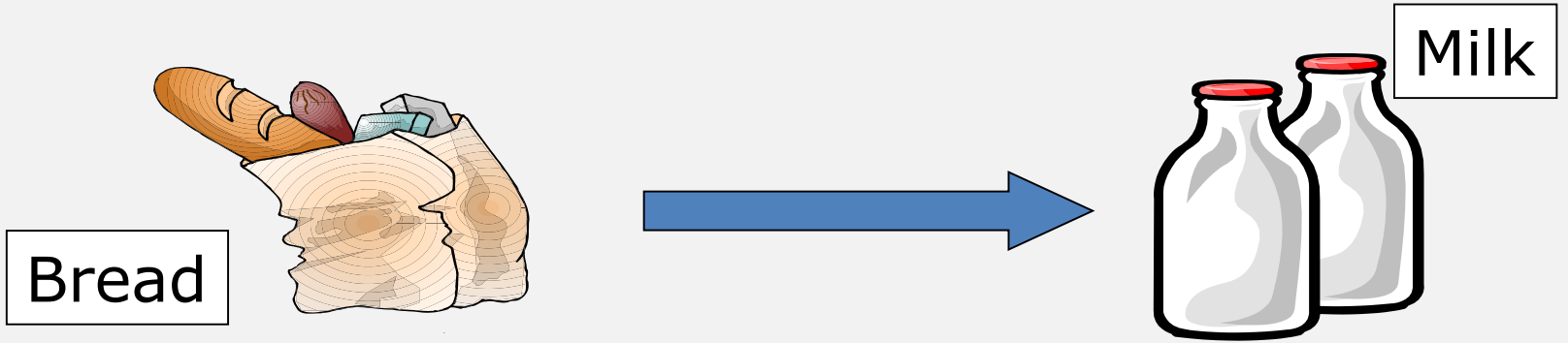
Age: 25

Car

Fuzzy Association Rules

✓Proposed by Kuok *et al.* (*SIGMOD Record*, 1998)

●Another approach to handle quantitative value



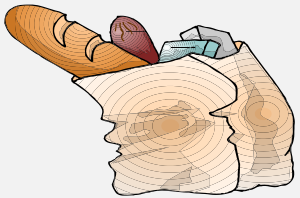
IF middle amount of bread is bought
Then high amount of milk is bought

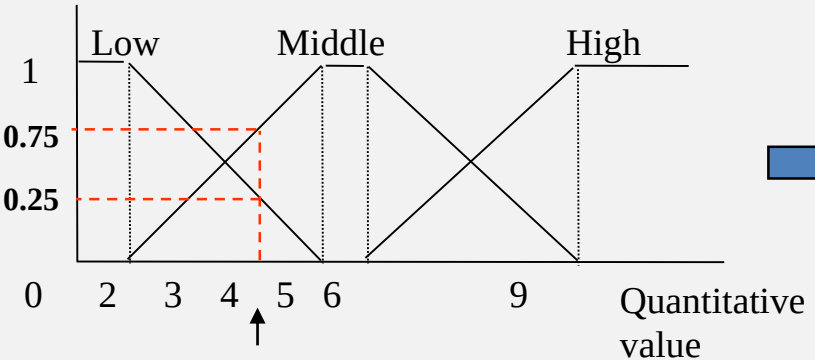
Linguistic Term

Fuzzy Association Rules (Cont.)

- ✓How to handle quantitative data?
- ✓e.g.

(Bread, 4)





$$\frac{0.25}{Bread.Low} + \frac{0.75}{Bread.Middle}$$

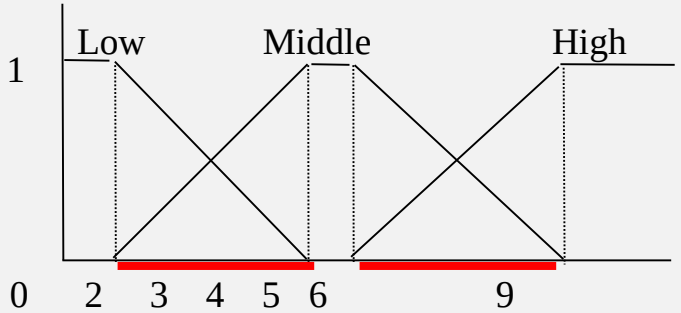
Benefits of Fuzzy Association Rules

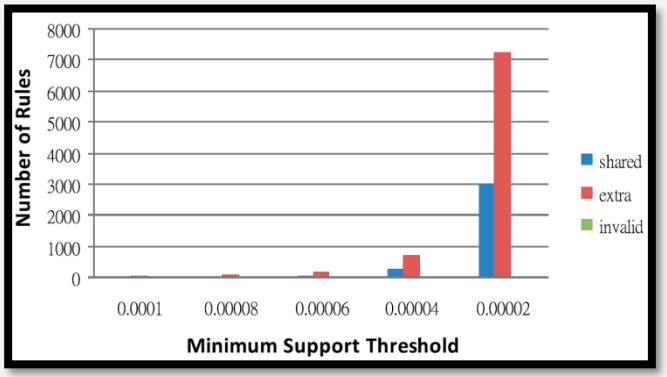
- ✓Three Advantages (Kuok et al.)
  - ✓Understandable to Human
  - ✓Handling Quantitative Value Well (Shape boundary problem)
  - ✓Deriving Extra Information

(Bread, 4)

Bread.Middle
Bread.High

Membership functions





Problems of Association Rule (Cont.)

✓Second problem

- ●A minimum support for all item is not appropriate

✓e.g.

Supermarket

Pan

High Utility Item

Price: 1,000
Sales: 50
Gain: 50,000

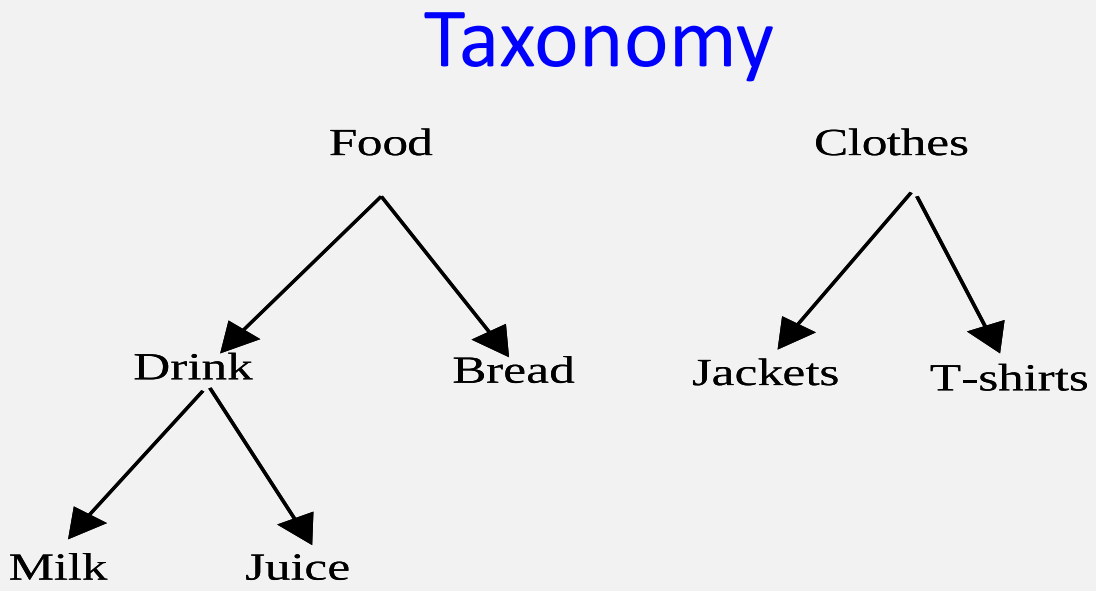
Milk

Price: 20
Sales: 200
Gain: 4,000

Different Items ➔ Different Minimum Supports

Problems of Association Rule (Cont.)

- ✓Only the terminal items can appear in transaction data
- ✓However, items may have taxonomy (as follows)



Conclusion

- ✓ Association Rule Mining algorithm
  - Binary association rule (Apriori)
  - An efficient rule mining algorithm (FP-growth)
  - Property of the Apriori algorithm
  - Related problems of Apriori algorithm and possible solutions