Frequent Pattern Mining

Man-Kwan Shan
Dept. of Computer Science
National Cheng-Chi Univ.

客戶有什麼購物行為？

Market-Basket Transactions

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

2

Fast Algorithms for Mining Association Rules

Rakesh Agrawal Ramakrishnan Srikant*

IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120

Abstract

We consider the problem of discovering association rules between items in a large database of sales transactions. We present two new algorithms for solving this problem that are fundamentally different from the known algorithms. Empirical evaluation shows that these algorithms outperform the known algorithms by factors ranging from three for small problems to more than an order of magnitude for large problems. We also show how the best features of the two proposed algorithms can be combined into a hybrid algorithm, called AprioriHybrid. Scale-up experiments show that AprioriHybrid scales linearly with the number of transactions. AprioriHybrid also has excellent scale-up properties with respect to the transaction size and the number of items in the database.

tires and auto accessories also get automotive services done. Finding all such rules is valuable for cross-marketing and attached mailing applications. Other applications include catalog design, add-on sales, store layout, and customer segmentation based on buying patterns. The databases involved in these applications are very large. It is imperative, therefore, to have fast algorithms for this task.

The following is a formal statement of the problem [4]: Let $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ be a set of literals, called items. Let $\mathcal{D}$ be a set of transactions, where each transaction T is a set of items such that $T \subseteq \mathcal{I}$. Associated with each transaction is a unique identifier, called its *TID*. We say that a transaction T *contains* X, a set of some items in $\mathcal{I}$, if $X \subseteq T$

VLDB, 1994.

18642, 25835, 31500 Citations
(2015, 2020, 2024 Google Scholar)

Dynamic Itemset Counting and Implication Rules for Market Basket Data

Sergey Brin * **Rajeev Motwani** **Jeffrey D. Ullman**

Department of Computer Science

Stanford University

{sergey,rajeev,ullman}@cs.stanford.edu

Shalom Tsur

R&D Division, Hitachi America Ltd.

tsur@hitachi.com

Abstract

We consider the problem of analyzing market-basket data and present several important contributions. First, we present a new algorithm for finding large itemsets which uses fewer passes over the data than classic algorithms, and yet uses fewer candidate itemsets than methods based on sampling. We investigate the idea of item reordering, which can improve the low-level efficiency of the algorithm. Second, we present a new way of generating "implication rules," which are normalized based on both the antecedent and the consequent and are truly implications (not simply a measure of co-occurrence), and we show how they produce more intuitive results than other methods. Finally, we show how different characteristics of real data, as opposed to synthetic data, can dramatically affect the performance of the system and the form of the results.

of web pages, and many more. We applied market-basket analysis to census data (see section 5).

In this paper, we address both performance and functionality issues of market-basket analysis. We improve performance over past methods by introducing a new algorithm for finding large itemsets (an important subproblem). We enhance functionality by introducing *implication rules* as an alternative to association rules (see below).

One very common formalization of this problem is finding *association rules* which are based on *support* and *confidence*. The support of an itemset (a set of items), I, is the fraction of transactions the itemset occurs in (is a subset of). An itemset is called *large* if its support exceeds a given threshold, σ. An association rule is written $I \rightarrow J$ where I and J are itemsets [1]. The *confidence* of this rule is the fraction of transactions containing I that also contain J. For the association rule, $I \rightarrow J$ to hold, $I \cup J$ must be large and the confidence of the rule must exceed a given confid-

ACM SIGMOD 1997

Beyond Market Baskets: Generalizing Association Rules to Correlations

Sergey Brin*
Department of Computer Science
Stanford University
Stanford, CA 94305
brin@cs.stanford.edu

Rajeev Motwani†
Department of Computer Science
Stanford University
Stanford, CA 94305
motwani@cs.stanford.edu

Craig Silverstein‡
Department of Computer Science
Stanford University
Stanford, CA 94305
csilvers@cs.stanford.edu

Abstract

One of the most well-studied problems in data mining is mining for association rules in market basket data. Association rules, whose significance is measured via support and confidence, are intended to identify rules of the type, "A customer purchasing item A often also purchases item B." Motivated by the goal of generalizing beyond market baskets and the association rules used with them, we develop the notion of mining rules that identify correlations (generalizing associations), and we consider both the absence and presence of items as a basis for generating rules. We propose measuring significance of associations via the chi-squared test for correlation from classical statistics. This leads to a measure that is upward closed in the itemset lattice, enabling us to reduce the mining problem to the search for a border between correlated and uncorrelated itemsets in the lattice. We develop pruning strategies and devise an efficient algorithm for the resulting problem. We demonstrate its effectiveness by testing it on census data and finding term dependence in a corpus of text documents, as well as on synthetic data.

setting, the base information consists of register transactions of retail stores. The goal is to discover buying patterns such as two or more items that are bought together often.[1] The market basket problem has received a great deal of attention in the recent past, partly due to its apparent utility and partly due to the research challenges it presents. The past research has emphasized techniques for improving the performance of algorithms for discovering association rules in large databases of sales information. There has also been some work on extending this paradigm to numeric and geometric data [11, 12].

While Piatetsky-Shapiro and Frawley [26] define an "association problem" as finding recurring patterns in data, much of the recent work on mining of large-scale databases has concerned the important special case of finding association rules. Association rules, whose significance is measured via support and confidence as explained below, are primarily intended to identify rules of the type, "A customer purchasing item X is likely to also purchase item Y." In general, the development of ideas has been closely linked to the notion of associations expressed via the customer preference example.

Dynamic Data Mining: Exploring Large Rule Spaces by Sampling

Sergey Brin and Lawrence Page

Department of Computer Science

Stanford University

{sergey,page}@cs.stanford.edu

February 23, 1998

Abstract

A great challenge for data mining techniques is the huge space of potential rules which can be generated. If there are tens of thousands of items, then potential rules involving three items number in the trillions. Traditional data mining techniques rely on downward-closed measures such as support to prune the space of rules. However, in many applications, such pruning techniques either do not sufficiently reduce the space of rules, or they are overly restrictive.

We propose a new solution to this problem, called Dynamic Data Mining (DDM). DDM foregoes the completeness offered by traditional techniques based on downward-closed measures in favor of the ability to drill deep into the space of rules and provide the user with a better view of the structure present in a data set.

Instead of a single determinstic run, DDM runs continuously, exploring more and more of the rule space. Instead of using a downward-closed measure such as support to guide its exploration, DDM uses a user-defined measure called *weight*, which is not restricted to be downward closed. The exploration is guided by a heuristic called the *Heavy Edge Property*.

The system incorporates user feedback by allowing *weight* to be redefined dynamically. We test the system on a particularly difficult data set – the word usage in a large subset of the World Wide Web. We find that Dynamic Data Mining is an effective tool for mining such difficult data sets.

Stanford Technical Report, 1999

Mining Optimized Gain Rules for Numeric Attributes

Sergey Brin
Stanford University, Palo Alto
sergey@cs.stanford.edu

Rajeev Rastogi
Bell Labs, Murray Hill
rastogi@lucent.com

Kyuseok Shim
Bell Labs, Murray Hill
shim@lucent.com

Abstract

Association rules are useful for determining correlations between attributes of a relation and have applications in marketing, financial and retail sectors. Furthermore, *optimized association rules* are an effective way to focus on the most interesting characteristics involving certain attributes. Optimized association rules are permitted to contain uninstantiated attributes and the problem is to determine instantiations such that either the support, confidence or gain of the rule is maximized.

In this paper, we generalize the optimized gain association rule problem by permitting rules to contain disjunctions over uninstantiated numeric attributes. Our generalized association rules enable us to extract more useful information about seasonal and local patterns involving the uninstantiated attribute. For rules containing a single numeric attribute, we present an algorithm with linear complexity for computing optimized gain rules. Furthermore, we propose bucketing technique that can result in a significant reduction in input size by coalescing contiguous values without sacrificing optimality. We also present an approximation algorithm based on *dynamic programming* for two numeric attributes. Using recent results on *binary space partitioning trees*, we show that the approximations are within a constant factor of the optimal optimized gain rules. Our experimental results for a single numeric attribute demonstrate that our algorithm scales up linearly with the attribute's domain size as well as the number of disjunctions.

among the underlying data and have applications in marketing, financial and retail sectors. In its most general form, an association rule can be viewed as being defined over attributes of a relation, and has the form $C_1 \rightarrow C_2$, where C_1 and C_2 are conjunctions of conditions, and each condition is either $A_i = v_i$ or $A_i \in [l_i, u_i]$ (v_i, l_i and u_i are values from the domain of the attribute A_i). Each rule has an associated *support* and *confidence*. Let the *support* of a condition C_i be the ratio of the number of tuples satisfying C_i and the number of tuples in the relation. The support of a rule of the form $C_1 \rightarrow C_2$ is then the same as the support of $C_1 \wedge C_2$, while its confidence is the ratio of the supports of conditions $C_1 \wedge C_2$ and C_1. The association rules problem is that of computing all association rules that satisfy user-specified minimum support and minimum confidence constraints, and efficient schemes for this can be found in [AS94, MTV94, PCY95, SON95, HF95, SA95, SA96].

For example, consider a relation in a telecom service provider database that contains call detail information. The attributes of the relation are date, time, src_city, src_country, dst_city, dst_country and duration. A single tuple in the relation thus captures information about the two endpoints of each call, as well as the temporal elements of the call. The association rule

University of Cincinnati

Date: 4/27/2018

I, Hung-An Kao, hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering.

It is entitled:

Quality Prediction Modeling for Multistage Manufacturing using Classification and Association Rule Mining Techniques

Student's name: **Hung-An Kao**

This work and its defense approved by:

Committee chair: Jay Lee, Ph.D.

Committee member: Edzel Lapira, Ph.D.

Committee member: Jing Shi, Ph.D.

Committee member: David Thompson, Ph.D.

30424

MATEC Web of Conferences **123**, 00029 (2017) DOI:10.1051/matecconf/201712300029
ICPMMT 2017

Quality prediction modeling for multistage manufacturing based on classification and association rule mining

Hung-An Kao[1,2,*], *Yan-Shou* Hsieh[1], *Cheng-Hui* Chen[1], and *Jay* Lee[2]

[1] Central Industry Research & Service Division (CID), Institute for Information Industry, Nantou, 540, Taiwan
[2] NSF I/UCRC for Intelligent Maintenance Systems (IMS), University of Cincinnati, Cincinnati, OH 45221, USA

Abstract. For manufacturing enterprises, product quality is a key factor to assess production capability and increase their core competence. To reduce external failure cost, many research and methodology have been introduced in order to improve process yield rate, such as TQC/TQM, Shewhart Cycle · Deming's 14 Points, etc. Nowadays, impressive progress has been made in process monitoring and industrial data analysis because of the Industry 4.0 trend. Industries start to utilize quality control (QC) methodology to lower inspection overhead and internal failure cost. Currently, the focus of QC is mostly in the inspection of single workstation and final product, however, for multistage manufacturing, many factors (like equipment, operators, parameters, etc.) can have cumulative and interactive effects to the final quality. When failure occurs, it is difficult to resume the original settings for cause analysis. To address these problems, this research proposes a combination of principal components analysis (PCA) with classification and association rule mining algorithms to extract features representing relationship of multiple workstations, predict final product quality, and analyze the root-cause of product defect. The method is demonstrated on a semiconductor data set.

Overview of Association Rules

Association Analysis

- Association Analysis:
  - finding association relationships
    among sets of items or objects
    in transactional or relational DB
  - example: market basket analysis (association rule mining)
- e.g.
  - bread ^ milk $\rightarrow$ butter
  - age(25~35) ^ income(45,000~60,000) $\rightarrow$ buys(Toyata)

Association Rule from Transaction DB

- Given a set of transactions,
  find rules that will predict the occurrence of items
  based on the occurrences of other items in the transaction

Market-Basket Transactions

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\}$

Association Rule from Relational DB

People

Record ID	Age	Married	Num Cars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

Min-Support = 40% = 2 records
Min-confidence = 50%

Rules	Support	Confidence
<Age:30..39> and <Married:Yes> => <NumCars:2>	40%	100%
<Age:20..29> => <NumCars: 1>	40%	66.7%

Rules

Formal Definitions

- $I = \{i_1, i_2, i_3 \ldots i_n\}$, the set of all items
- $T \subseteq I$, a transaction
- D, a set of T, transaction DB
- Association rule: $A \rightarrow B$, $A \subset I$, $B \subset I$, $A \cap B = \varnothing$

TID	Items
100	a, c, d
200	b, c, e
300	a, b, c, e
400	b e

- $I=\{a, b, c, d, e\}$
- $\{b, c\} \rightarrow \{e\}$

  $\{b, c\} \subset I$, $\{e\} \subset I$,

  $\{b, c\} \cap \{e\} = \varnothing$

Formal Definitions (cont.)

- support ($A \rightarrow B$) = $P(A \cup B)$
  - fraction of transactions that contain an itemset
  - support ({b, c}→{e}) = support ({b, c, e}) = 50%
- confidence($A \rightarrow B$)=$P(B|A)=P(A \cup B)/P(A)$
  - fraction of transactions containing A that also contain B
  - confidence({b, c}→{e}) = P({b, c, e})/P({b,c}) = 100%

TID	Items
100	a, c, d
200	b, c, e
300	a, b, c, e
400	b e

- {a}→{c}, support = 50%,confidence=100%
- {c} →{a}, support = 50%, confidence=66%

Formal Definitions (cont.)

- Given a set of transactions T,
  the goal of association rule mining is
  to find all rules having
  - support ≥ *minsup* threshold and
  - confidence ≥ *minconf* threshold

* Strong rule

An Example

Given

(1) minimum support 2/4,

(2) minimum confidence 2/3

TID	Items
100	a, c, d
200	b, c, e
300	a, b, c, e
400	b e

– Association rules

- {b} → {e}, support = 75%, confidence = 100%
- {e} → {b}, support = 75%, confidence = 100%
- {a} → {c}, support = 50%, confidence = 100%
- {c} → {a}, support = 50%, confidence = 66%
- {b, c} → {e}, support = 50%, confidence = 100%
- {e} → {b, c}, support = 50%, confidence = 66%
- {c, e} → {b}, support = 50%, confidence = 100%
- {b, e} → {c}, support = 50%, confidence = 66%
- …

- Given
  (1) minimum support 2/4,
  (2) minimum confidence 2/3

TID	Items
100	a, c, d
200	b, c, e
300	a, b, c, e
400	b e

how many association rules are generated ?

how to generate all strong rules ?

Basic Approach of Association Rule Mining

- Itemset: a set of items
- *k*-itemset: an itemset that contains *k* items
- Frequent itemset: itemset that satisfy minimum support threshold
- Process of association rule mining
  - step 1: find all frequent itemstes (considering support)
  - step 2: generate strong association rules from frequent itemsets (considering confidence)

An Example

Given

(1) minimum support 2/4,

(2) minimum confidence 2/3

TID	Items
100	a, c, d
200	b, c, e
300	a, b, c, e
400	b e

- frequent itemsets
  - {a}:2, {b}:3, {c}:3, {e}:3, {a,c}:2, {b,c}:2, {b,e}:3, {c,e}:2, {b,c,e}:2
- strong rules
  - {a}→{c}: 2/2, {c}→ {a}: 2/3 {a, c}
  - {b}→{c}: 2/3, {c}→ {b}: 2/3 {b, c}
  - {b}→{e}: 3/3, {e}→ {b}: 3/3 {b, e}
  - {c}→{e}: 2/3, {e}→ {c}: 2/3 {c, e}
  - {b,e} → {c}: 2/3, {b,c} →{e}: 2/2, {c,e}→{b}: 2/2
    {c} → {b,e}: 2/3, {e} →{b,c}: 2/3, {b}→{c,e}: 2/3 {b, c, e}

- Given
  minimum support 2/4,

TID	Items
100	a, c, d
200	b, c, e
300	a, b, c, e
400	b e

how to generate all frequent itemsets ?

Apriori Algorithm

Apriori Algorithm

TID	Items
100	a, c, d
200	b, c, e
300	a, b, c, e
400	b e

- Rationale: apriori property
  - all non-empty subsets of a frequent itemset must also be frequent
  - All the supersets of an infrequent itemset must also be infrequent

    e.g. {d} is not frequent, {a,d} is not frequent either.
  - a special case of anti-monotone property
    - if a set cannot pass a test, all its supersets will fail the same test

Apriori Algorithm (cont.)

- Algorithms for discovering frequent itemsets make multiple passes over the data.
- In the 1st pass, we count the support of 1-items and determine which of them are frequent.
- In each subsequent pass,
  - we start with a seed set of itemsets found to be frequent in the previous pass.
  - We use this seed set for generating new candidate itemsets & count the actual support for these candidate itemsets during the pass over the data
  - At the end of the pass, we determine which of the candidate itemsets are actually frequent and they become the seed for the next pass.
- This process continues until no new frequent itemsets are found.

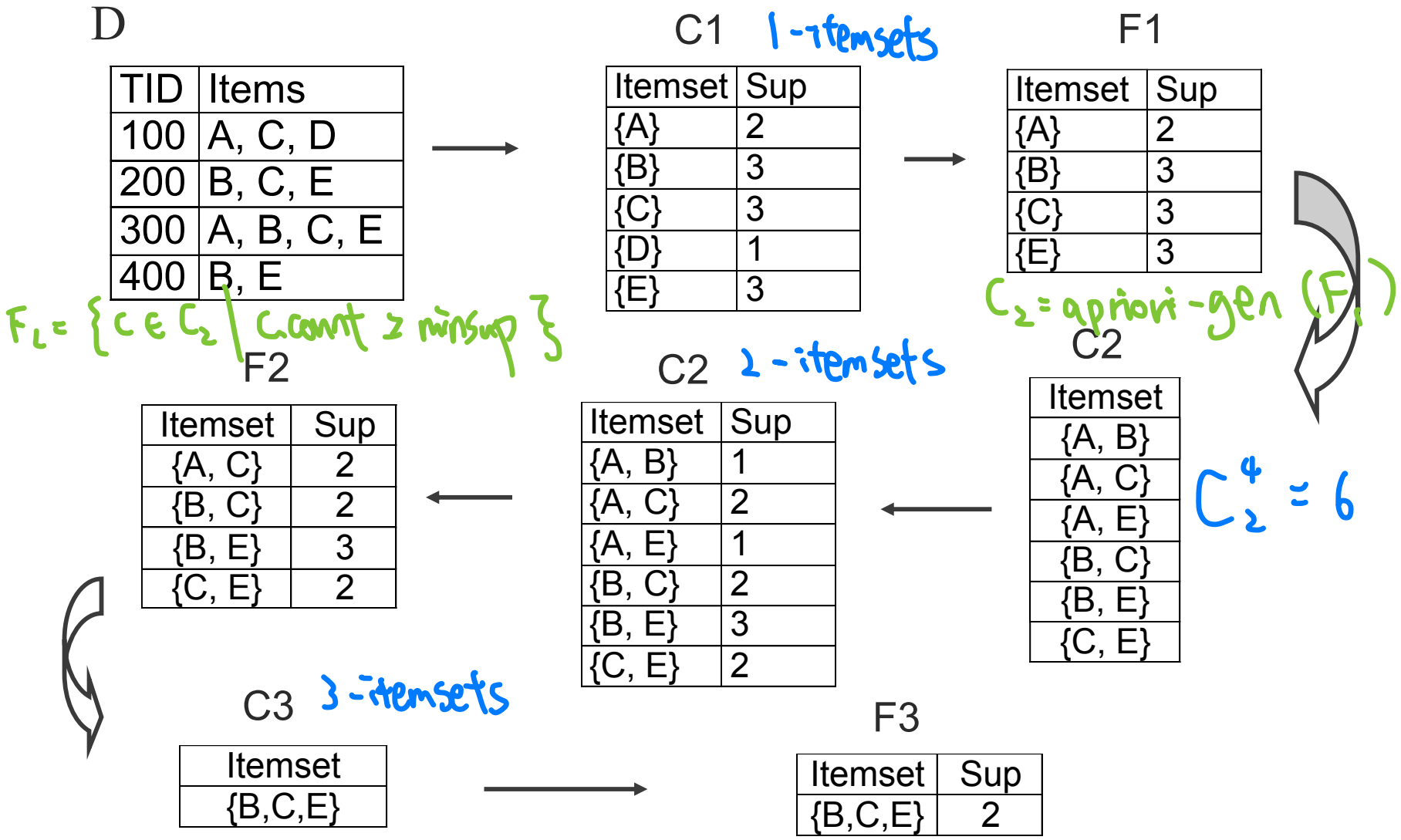
Apriori Algorithm (cont.)

- level-wise approach
  - (k-1)-itemsets are used to explore k-itemsets
  - join $C_k = F_{k-1} \otimes F_{k-1} = \{A \otimes B | A, B \in F_{k-1}, |A \cap B| = k-2\}$
  - prune C_k by subset test
  - Generate F_k by scanning transaction DB

C : candidate itemset

F : frequent itemset

Apriori Algorithm: An Example



Algorithm Apriori

termination condition = no frequent itemsets can be generated ($F_k = \phi$)

F_1={frequent 1-itemsets};
for (k=2; $F_{k-1} \neq 0$; k++) do begin
    C_k=apriori-gen(F_{k-1}); //candidate generation
    for all transactions t ∈ D do begin //support counting
        C_t=subset(C_k, t)
        for all candidate c ∈C_t do
            c.count++;
    end
    F_k={c ∈C_k|c.count ≥ minsup} //frequent k-itemset
end
Answer=$\cup_k F_k$;

C_t: A set of all k-itemsets in C_k that are subsets of t

Candidate Generation

C_k=apriori-gen(F_{k-1});

join step

insert into C_k
select p.item$_1$, p.item$_2$…,p.item$_{k-1}$, q.item$_{k-1}$
from F_{k-1} p, F_{k-1} q
where p.item$_{1=}$q.item$_1$,…, p.item$_{k-2=}$q.item$_{k-2}$,
p.item$_{k-1}$ <q.item$_{k-1}$;

prune step (apriori property : removes any k-itemset that has an infrequent (k-1)-itemset)

for all itemsets c $\in$ C_k do
for all (k-1)-subsets s of c do
if (s $\notin$ F_{k-1}) then
delete c from C_k

Candidate Generation (cont.)

join step

insert into C_k
select p.item$_1$, p.item$_2$…,p.item$_{k-1}$, q.item$_{k-1}$
from F_{k-1} p, F_{k-1} q
where p.item$_1$ = q.item$_1$,…, p.item$_{k-2}$ = q.item$_{k-2}$,
p.item$_{k-1}$ < q.item$_{k-1}$;

→ the first (k-2) items in p and q are identical

the (k-1)th item of p < the (k-1)th item of q

Example.

insert into C_4
select p.item$_1$, p.item$_2$, p.item$_3$, q.item$_3$
from F_3 p, F_3 q
where p.item$_1$=q.item$_1$, p.item$_2$=q.item$_2$,
p.item$_3$ < q.item$_3$;

$p = \{A, B, C\},\ q = \{A, B, D\}$, p join q $= \{A, B, C, D\}$

p = {A, B, C}, q = {A, B, D}, p join q = {A, B, C, D}

given p = {A, B, C}, q = {B, C, D},
is it necessary to join p, q ?

Candidate Generation (cont.)

prune step

for all itemsets c $\in C_k$ do

for all (k-1)-subsets s of c do

if (s $\notin F_{k-1}$) then

delete c from C_k

Example

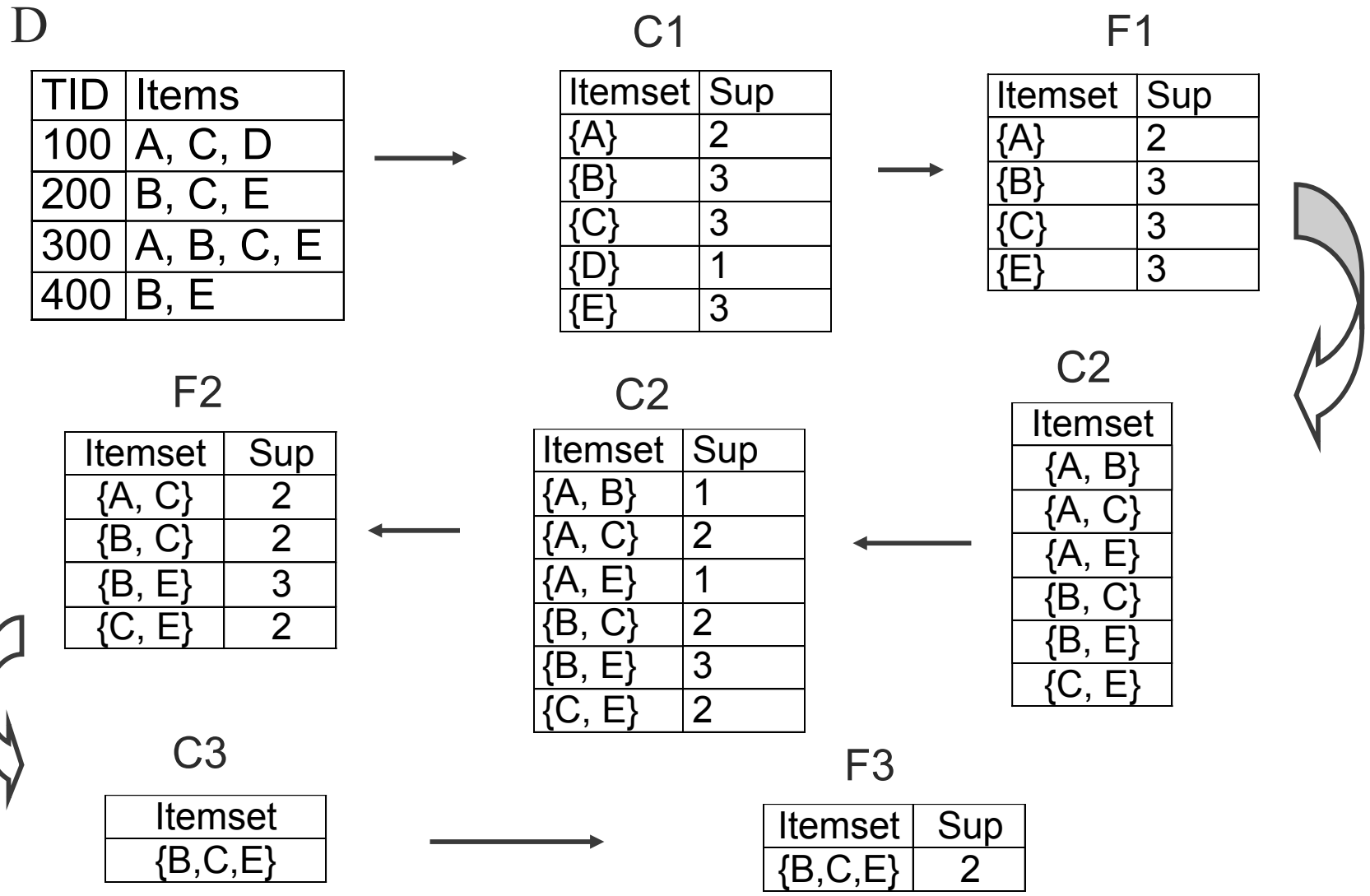
candidate *c* = {A, B, C, D}

check all 3-subsets *s* of *c*

{a, b, c}, {a, b, d}, {a, c, d}, {b, c, d}

if any of these four 3-subsets is infrequent in pass 3, prune *c*

Apriori Algorithm: An Example



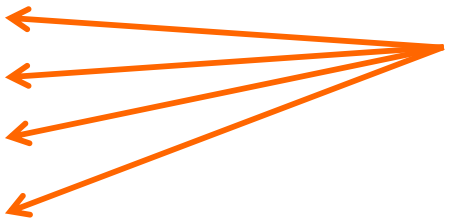
Algorithm Apriori

F_1={frequent 1-itemsets};
for (k=2; $F_{k-1} \neq 0$; k++) do begin
    C_k=apriori-gen(F_{k-1}); //candidate generation
    for all transactions t $\in$ D do begin //support counting
        C_t=subset(C_k, t)
        for all candidate c $\in C_t$ do
            c.count++;
    end
    F_k={c $\in C_k$|c.count $\geq$ minsup} //frequent k-itemset
end
Answer=$\cup_k F_k$;

Support Counting: Approach 1

D

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



C2

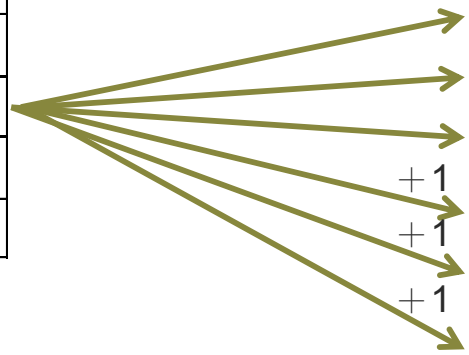
Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

Complexity=O(6 x 4)

Support Counting: Approach 2

D

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



C2

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

Complexity=O(4 x 6)

Which approach is faster ?

Approach 2 : avoid relocating transactions (optimized memory access)

there are many items in a transaction in real cases

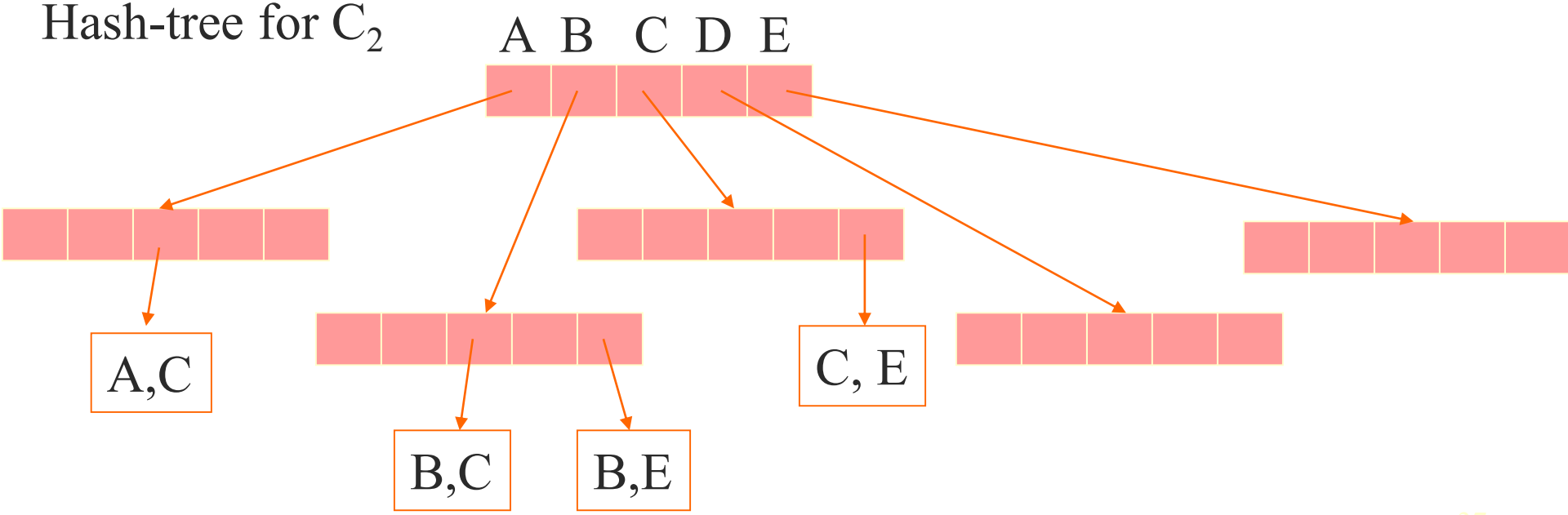
Subset Function

C_2

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

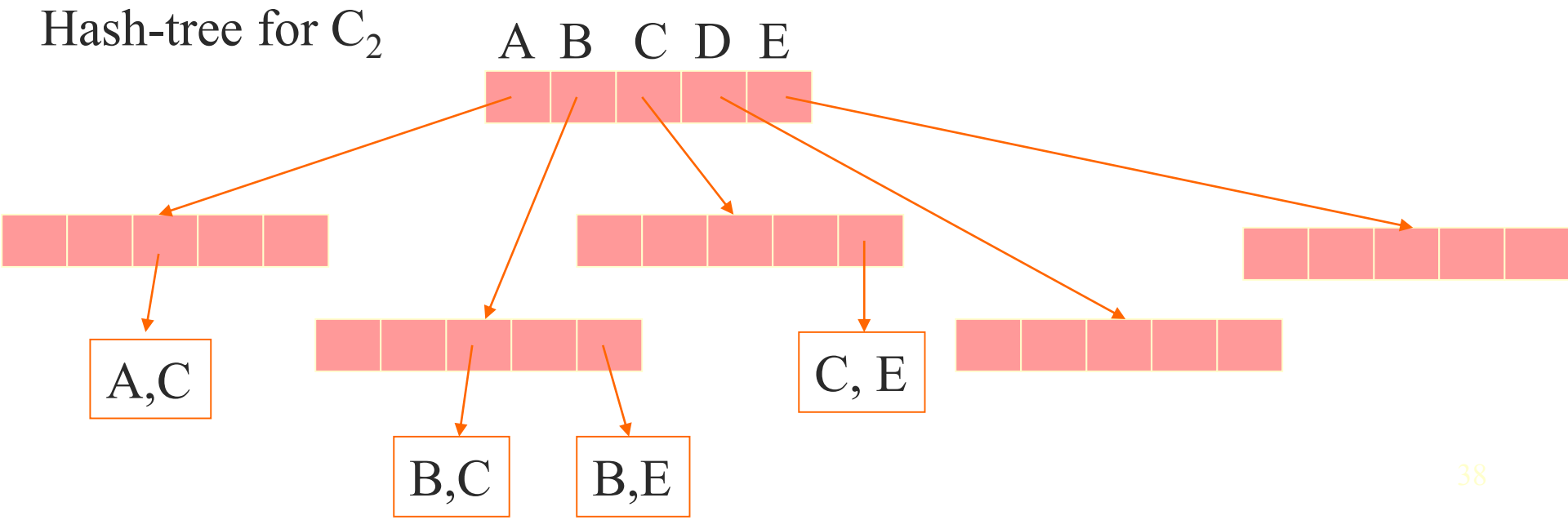
transaction: { B, C, E} ⟶ ♥ ♥ ♥ ⟶ {{B,C},{B,E},{C,E}}

Hash-tree for C_2



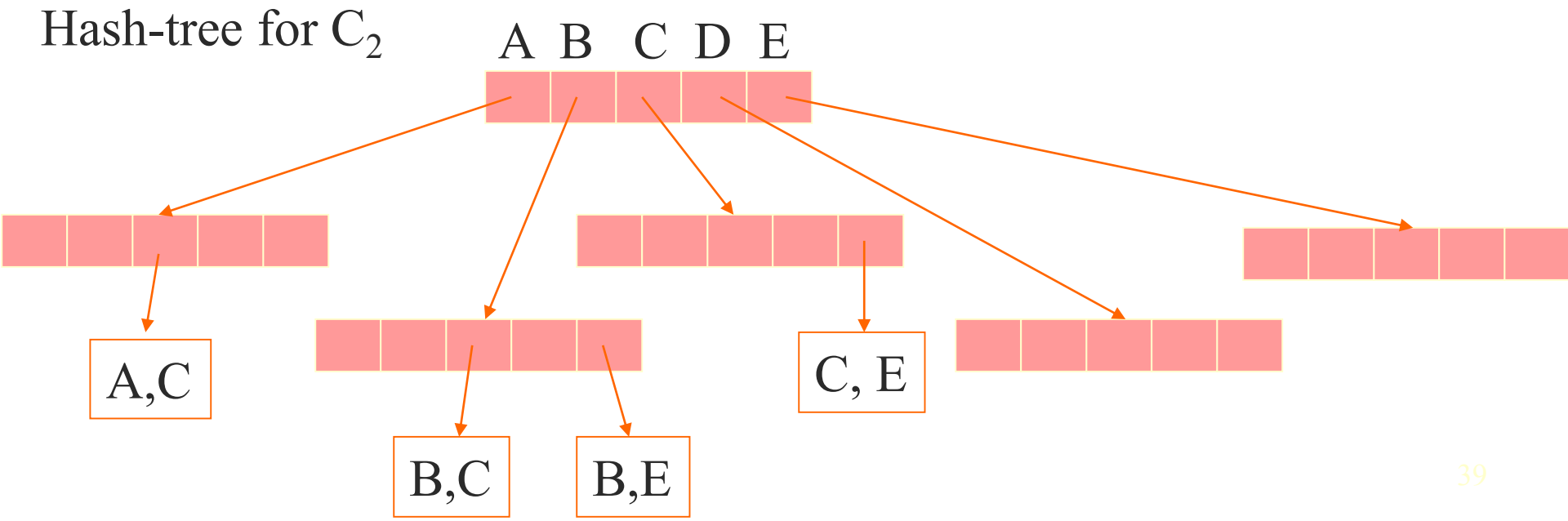
Hash Tree: Data Structure

- A node of the hash tree either contains a list of itemsets (a leaf node) or a hash table (an interior node).
- In an interior node each bucket of the hash table points to another node.
- The root of the hash tree is designed to be at depth 1.
- An interior node at depth *d* points to nodes at depth *d*+1.
- Itemsets are stored in the leaves.



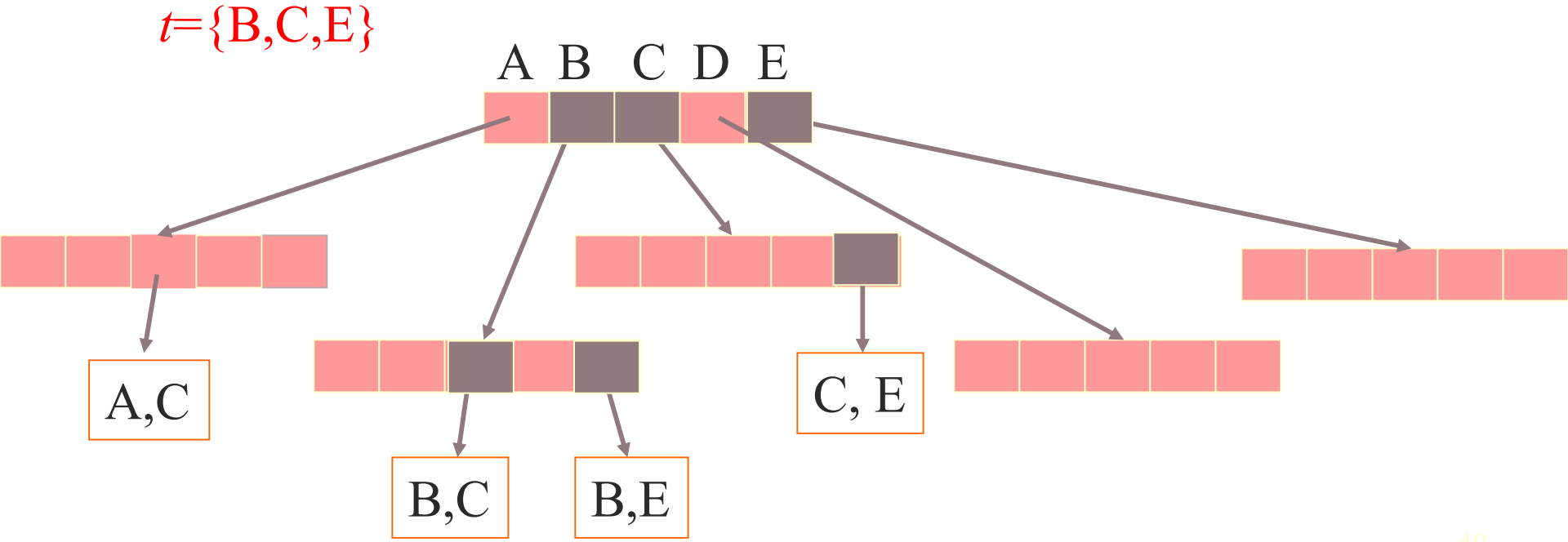
Hash Tree: Construction

- When we add an itemset *c*, we start from the root and go down the tree until we reach a leaf.
- At an interior node at depth *d*, we decide which branch to follow by applying a hash function to the *d*-th item of the itemset.
- All nodes are initially created as leaf nodes.
- When the number of itemsets in a leaf node exceeds a specified threshold, the leaf node is converted to an interior node.



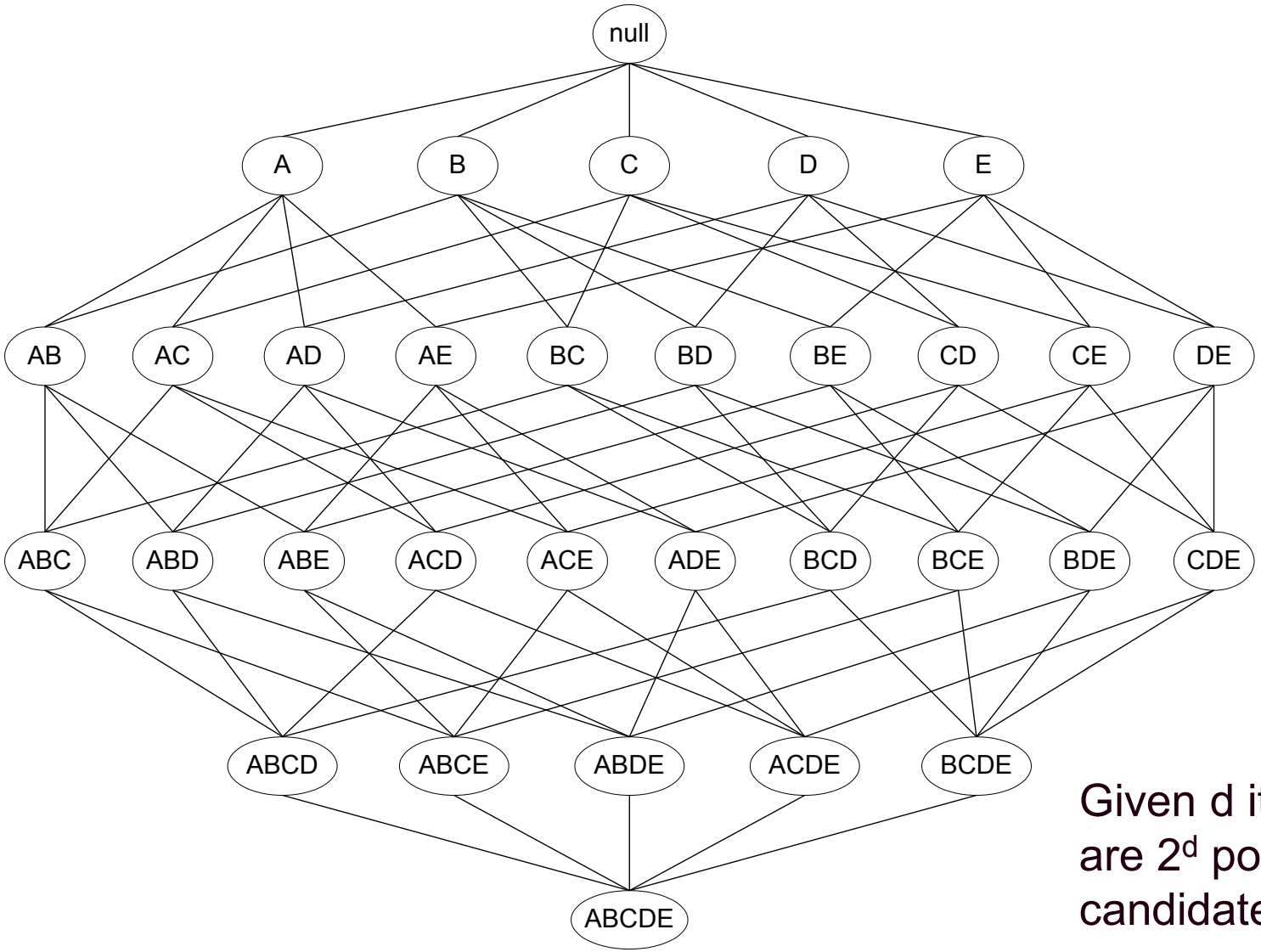
Hash Tree: Search for subsets

- For the root, hash on every item in *t*
- If in the interior node and reach this node by hashing item *i*, hash on each item that comes after *i* in *t* and recursively do.
- If in leaf, add the corresponding itemset into answer set.



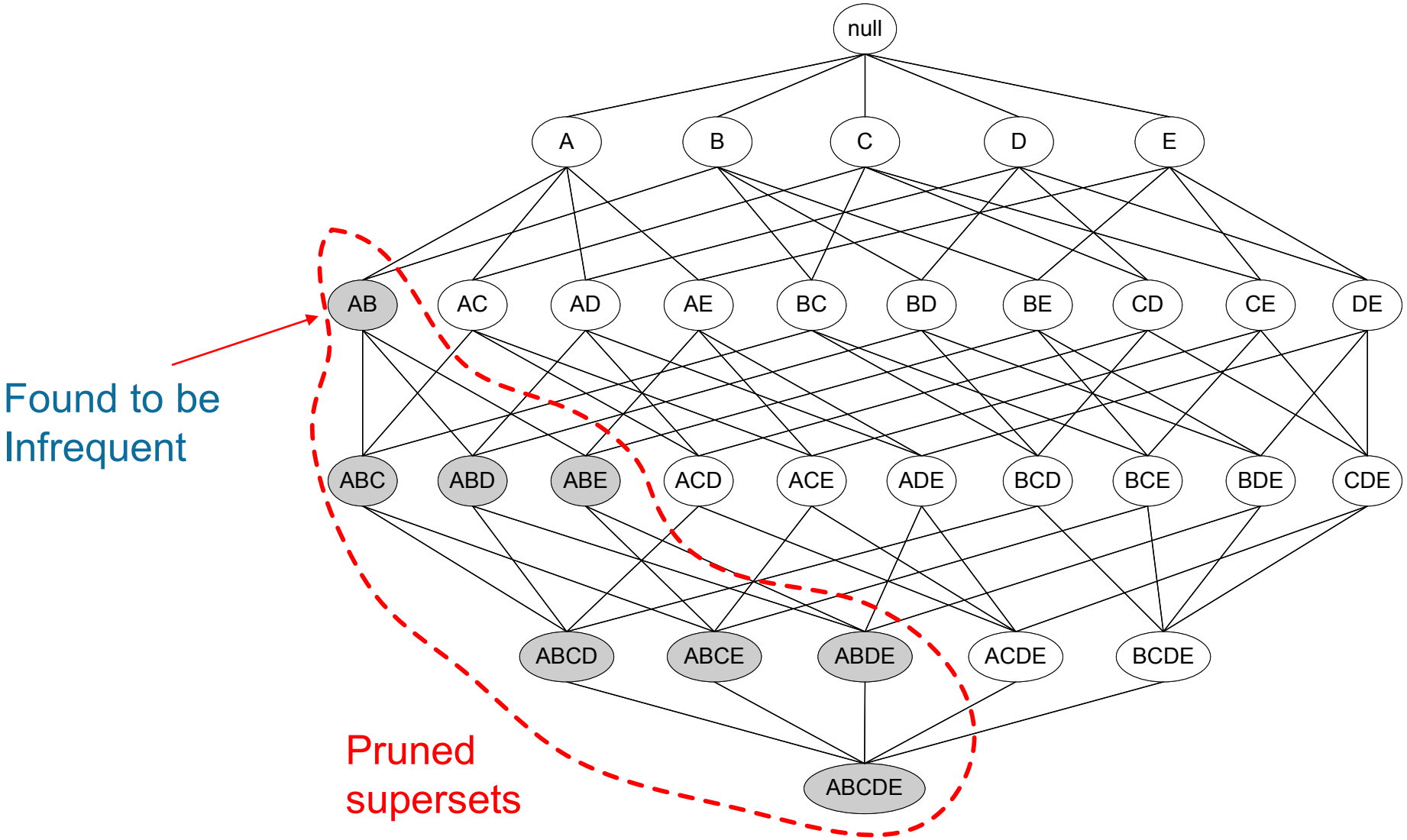
Rationale of Apriori ?

Frequent Itemset Generation (Lattice)



Given d items, there are 2^d possible candidate itemsets

Rationale of Apriori Algorithm



Improvement of Apriori Algorithm

Improvement of Apriori Algorithm

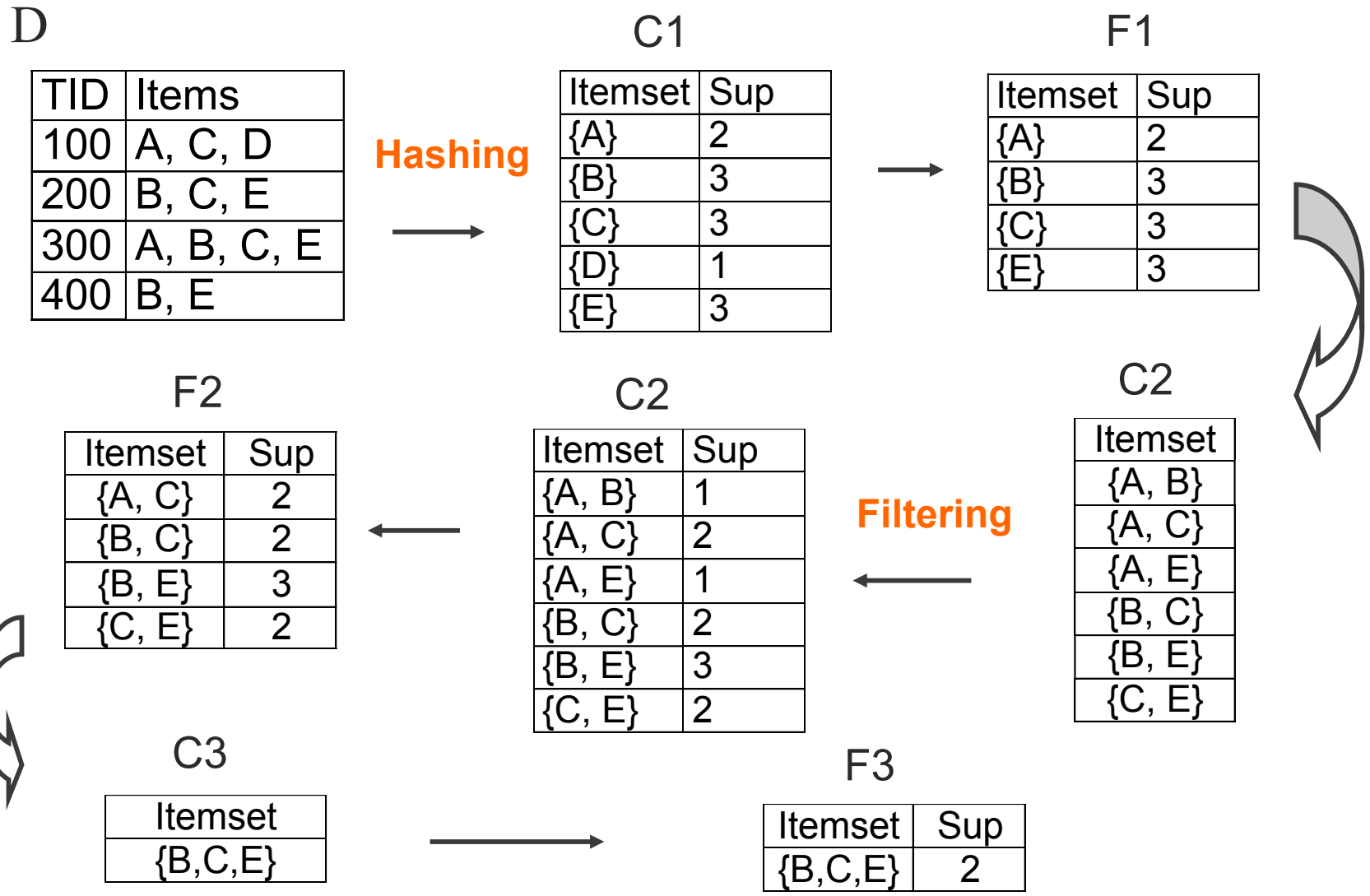
- Hashing (DHP)
- Scan reduction (DHP)
- Transaction reduction(DHP)
- Partitioning
- sampling
- FP-Trees

DHP(Direct Hashing & Pruning)

- Observation of performance in frequent itemset mining
  - initial candidate set generation is the key issue to improve
  - amount of transaction data must be scanned
- Major features of DHP
  - efficient generation for frequent itemsets
  - effective reduction on transaction database size
  - option of reducing #(database scan) required.

- An Effective Hash Based Algorithm for Mining Association Rules, J. S. Park, M. S. Chen, P. Yu, ACM SIGMOD 1995.

Apriori Algorithm: An Example



Efficient Generation of Frequent Itemsets

- Using hashing to filter out unqualified candidates

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

{A,C}, {A,D}, {C,D}
{B,C}, {B,E},{C,E}
{A,B},{A,C},{A,E},{B,C},{B,E},{C,E}
{B,E}

C1

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

($F_1 = \{\{A\},\{B\},\{C\},\{E\}\}$)

$F_1 * F_1$ self-join

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

C_2

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

{C,E}
{C,E}
{A,D}

{A,E}

{B,C}
{B,C}

{B,E}
{B,E}
{B,E}

{A,B}

{A,C}
{C,D}
{A,C}

H_2	3	1	2	0	3	1	3

h({x,y})=((ord(x)*10+ord(y)) mod 7

e.g. $h\{B,C\} = (ord(B) * 10 + ord(C)) \bmod 7 = (2*10+3) \bmod 7 = 2$

|Items| = 1000

|Transactions| = 100,000

|Average Items| = 10

Min. sup. = 0.75%

	Apriori	DHP		
	number	number	D_k	$\|D_k\|$
L_1	760	760	6.54MB	100,000
C_2	288,420	318	6.54MB	100,000
L_2	211	211		
C_3	220	220	0.51MB	20,047
L_3	204	204		
C_4	229	229	0.25MB	8,343
L_4	227	227		
C_5	180	180	0.16MB	4,919
L_5	180	180		
C_6	94	94	0.10MB	2,459
L_6	94	94		
C_7	29	29	0.06MB	1,254
L_7	29	29		
C_8	4	4	0.05MB	1,085
L_8	4	4		
total time	43.36	13.57		

Effective Reduction on Transaction Database Size

- Pruning unqualified items or transactions

D_2

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

{A,C}
{B,C}, {B,E},{C,E}
{A,C},{B,C},{B,E},{C,E}
{B,E}

D_3

TID	Items
200	B, C, E
300	B, C, E

an item in transaction *t* can be trimmed if it does not appear in at least *k* of the candidate *k*-itemsets in *t*

k=2

F_2

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

Partitioning Approach

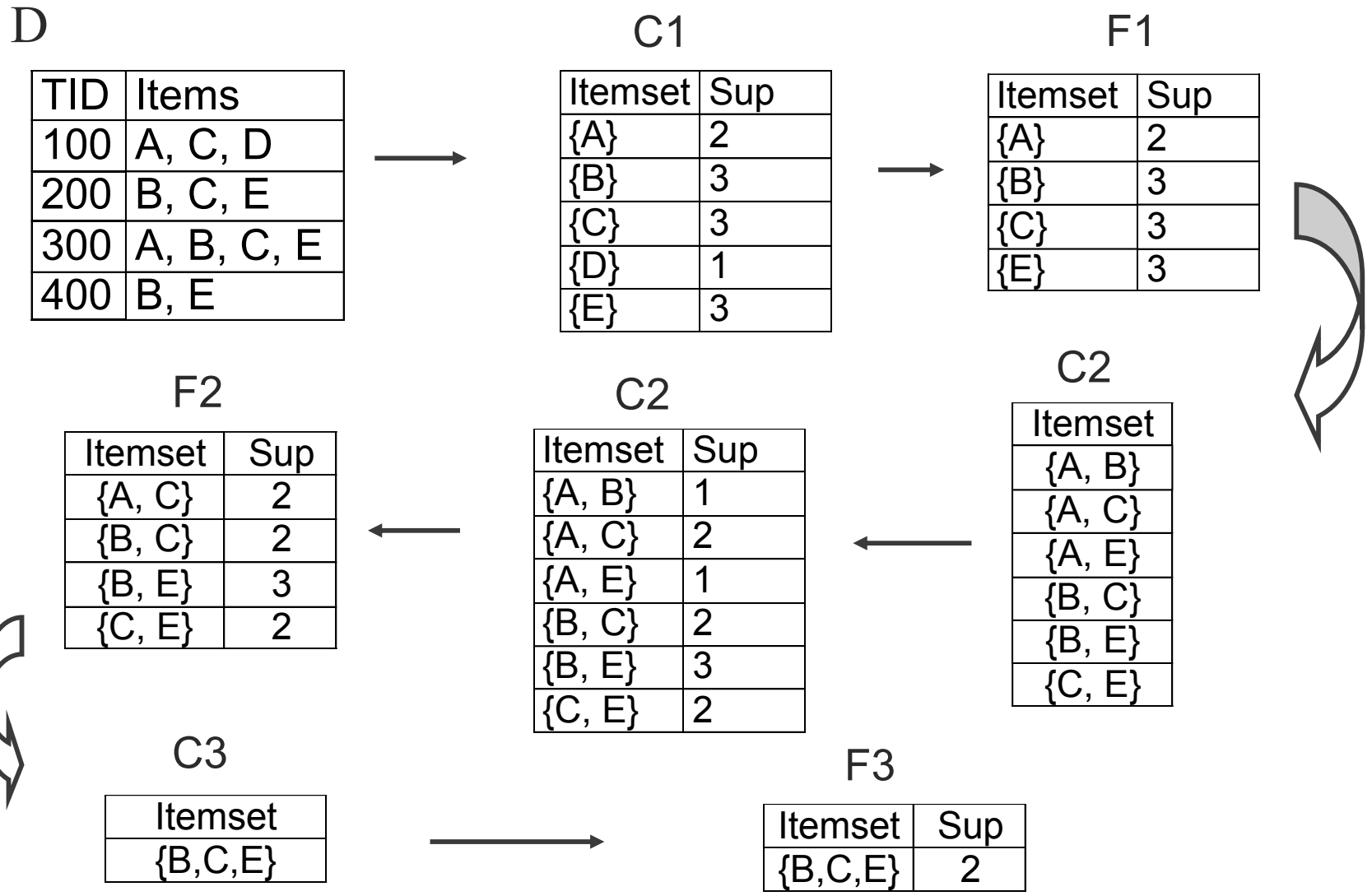
suitable for parallel & distributed computing

Partitioning

memory-resident partitions

- Transaction DB is divided into equal-sized partitions
- Partition size is chosen to be resident in main memory
- Observation: any potential frequent itemset appears as a frequent itemset in at least one of the partitions.
- Two phases scanning
  - First scan: generates a set of all potentially frequent itemsets
    - Each partition generates the local frequent itemsets
  - Second scan: actual support is measured
    - Collection of local frequent itemset = global candidate itemset
    - Global frequent itemsets are found by scan DB

Apriori Algorithm: An Example



Frequent Pattern Tree

FP-Tree Approach

FP-Tree Approach

- Motivation
  - Mining in main memory to reduce #(DB scans)
  - Without candidate generation
  - More frequently occurring items will have better chances of sharing item than less frequently occurring items

FP-Growth (cont.)

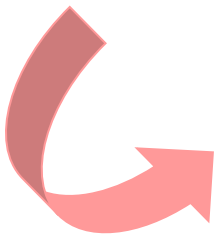
- Frequent pattern Growth
- Divide-and-conquer strategy
- Algorithm
  - Phase 1: Construct FP-Tree (frequent-pattern tree)
  - Phase 2: FP-Growth (frequent pattern growth)
    - Divide FP-tree into conditional FP-tree (conditional DB), each associated with one frequent item
    - Mine each such DB separately

FP-Trees Construction

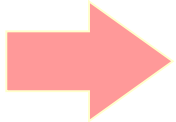
- Step 1: Find frequent 1-item, sorted items in frequency descending order by scanning DB

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

Given minimum support count 3



a	3
b	3
c	4
f	4
m	3
p	3



f	4
c	4
a	3
b	3
m	3
p	3

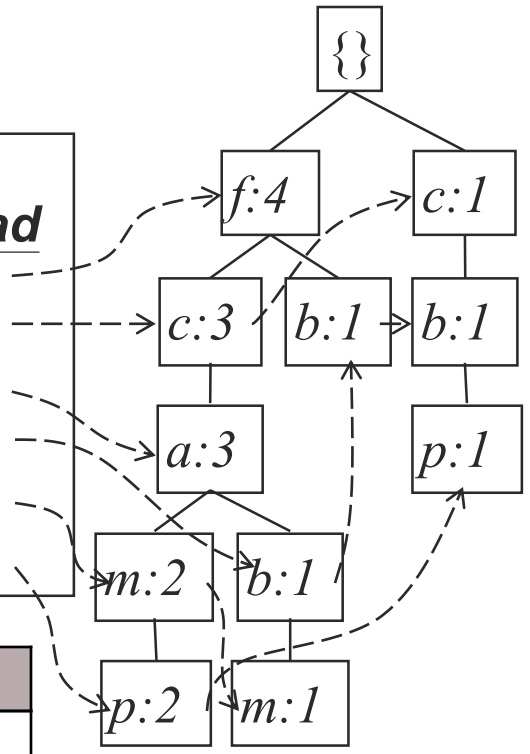
FP-Trees Construction (cont.)

Step 2: Scan DB and construct the FP-tree

f	4
c	4
a	3
b	3
m	3
p	3

Header

Item	*frequency*	*head*
f	*4*	
c	*4*	
a	*3*	
b	*3*	
m	*3*	
p	*3*	



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

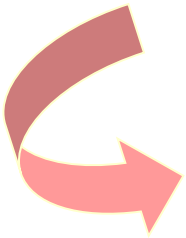
* FP-Tree is a compressed representation of the database

FP-Trees Construction (cont.)

Step 2: Scan DB and construct the FP-tree

Step 2.1: Scan DB & transform each transaction into an ordered set by pruning infrequent items and ordering frequent items according support counts

f	4
c	4
a	3
b	3
m	3
p	3



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

FP-Trees Construction (cont.)

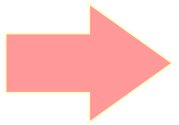
Step 2: Scan DB and construct the FP-tree

Step 2.2: Construct the FP-tree

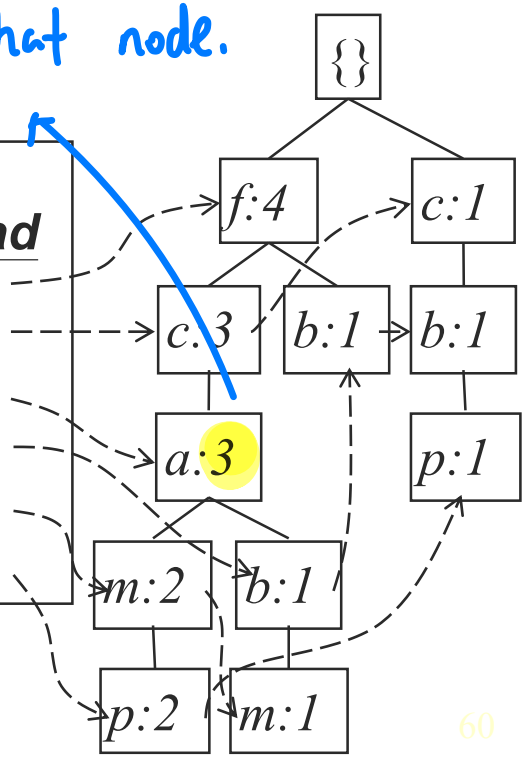
by aggregating the ordered set with common prefix

The support count of a node represents the #(transactions) that contain the item in the path leading up to that node.

Ordered
{f, c, a, m, p}
{f, c, a, b, m}
{f, b}
{c, b, p}
{f, c, a, m, p}



Header		
Item	*frequency*	*head*
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Prefix 字首, 前綴
Suffix 字尾, 後綴

NCCU

Prefix: N
NC
NCC
NCCU

Suffix: U
CU
CCU
NCCU

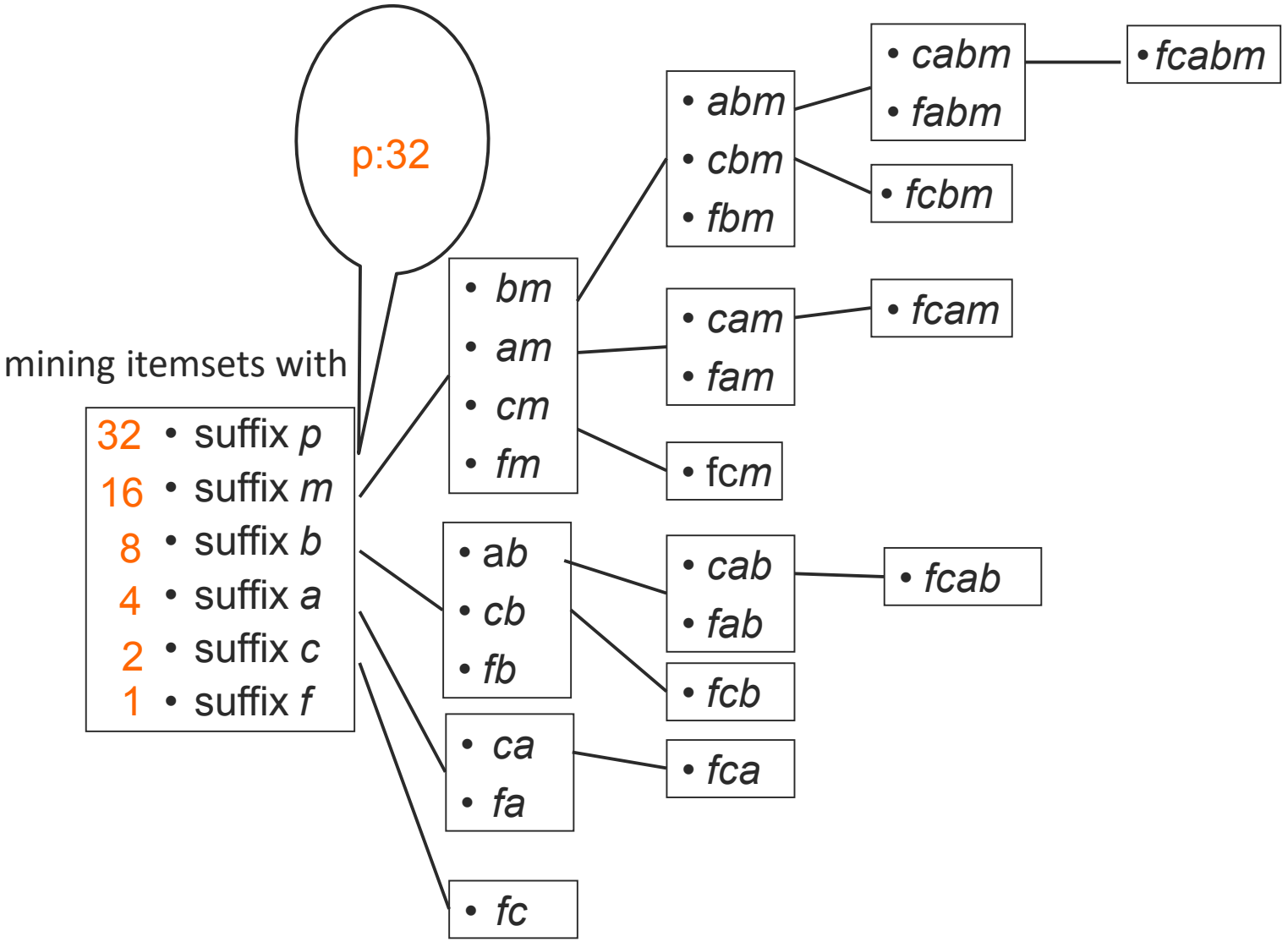
Rationale of FP-Growth

- All the frequent itemsets can be divided into
  - Itemsets with suffix *p*: {*p*}, {*c, p*}
  - Itemsets with suffix *m*: {*f, c, a, m*}, {*f, c, m*}, {*f, a, m*}, {*c, a, m*}, {*a, m*}, {*c, m*}, {*f, m*}, {*m*}
  - Itemsets with suffix *b*: {*b*}
  - Itemsets with suffix *a*: {*f, c, a*}, {*f,a*}, {*c,a*}, [*a*}
  - Itemsets with suffix *c*: {*f,c*}, {*c*}
  - Itemsets with suffix *f*: {*f*}
- All the frequent itemsets with suffix *m* can be divided into
  - Itemsets with suffix *bm*:
  - Itemsets with suffix *am*: {*f, c, a, m*}, {*f, a, m*}, {*c, a, m*}, {*a, m*}
  - Itemsets with suffix *cm*: {*c, m*}, {*f, c, m*},
  - Itemsets with suffix *fm*: {*f, m*}

* suffix pm has already been mined !

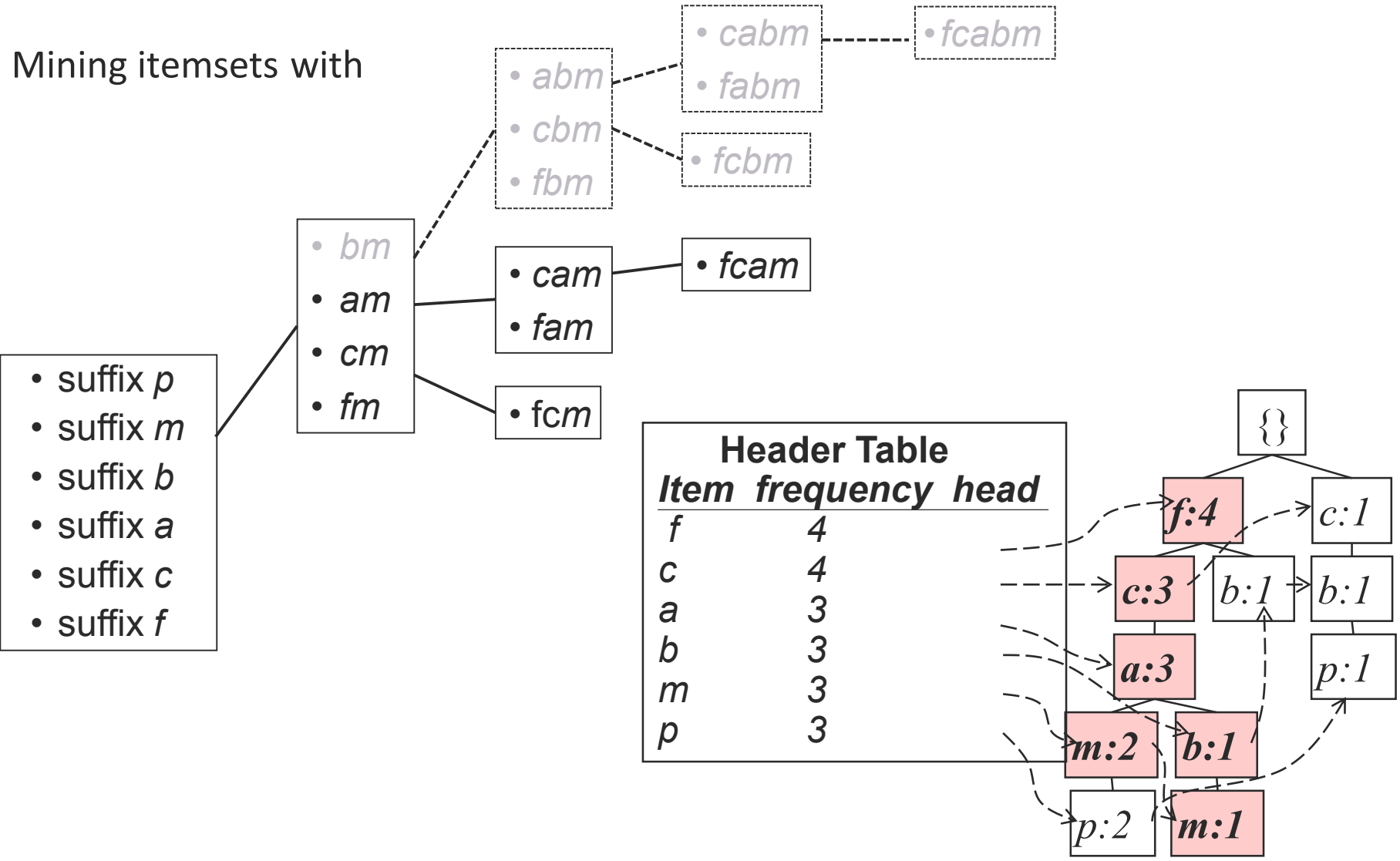
Rationale of FP-Growth (cont.)

- Mining all the frequent itemsets



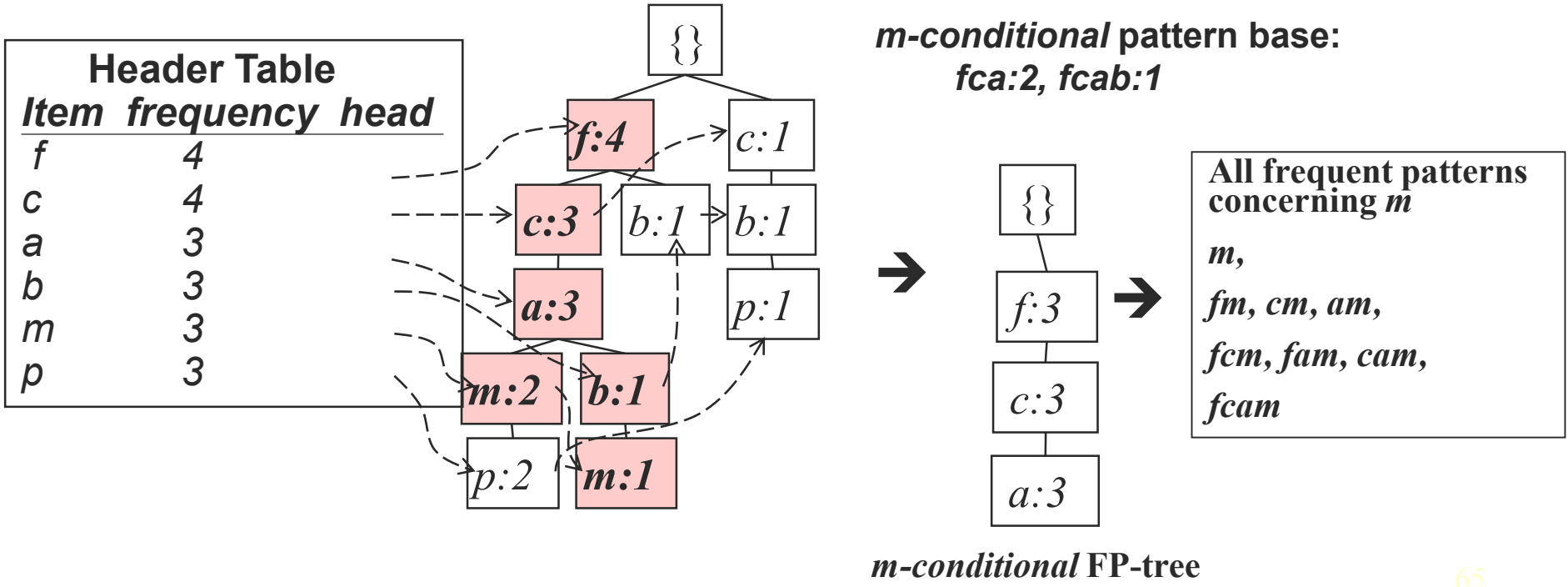
Rationale of FP-Growth (cont.)

- Mining all the frequent itemsets

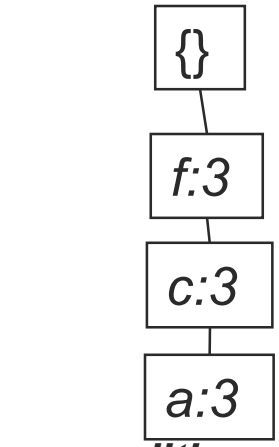


Construct Conditional FP-tree

- For each pattern-base
  - Accumulate the count for each item in the base
  - Construct the conditional FP-tree for the frequent items of the pattern base

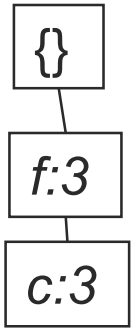


Recursively Mining Conditional FP-tree



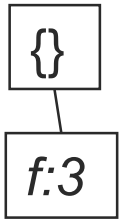
***m-conditional* FP-tree**

Cond. pattern base of “am”: (*fc:*3)



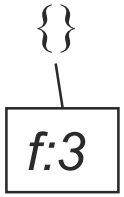
***am-conditional* FP-tree**

Cond. pattern base of “cam”: (*f*:3)



***cam-conditional* FP-tree**

Cond. pattern base of “cm”: (*f*:3)



***cm-conditional* FP-tree**

FP-Growth Overview

- Start from each frequent 1-pattern (initial suffix pattern), construct conditional FP-tree
- Mining recursively on such tree.
- Pattern growth is achieved by the concatenation of suffix pattern with frequent patterns generated from conditional FP-tree
- If the conditional FP-tree has a single path *P*, the complete set of frequent pattern of *T* can be generated by enumeration of all the combinations of the sub-paths of *P*

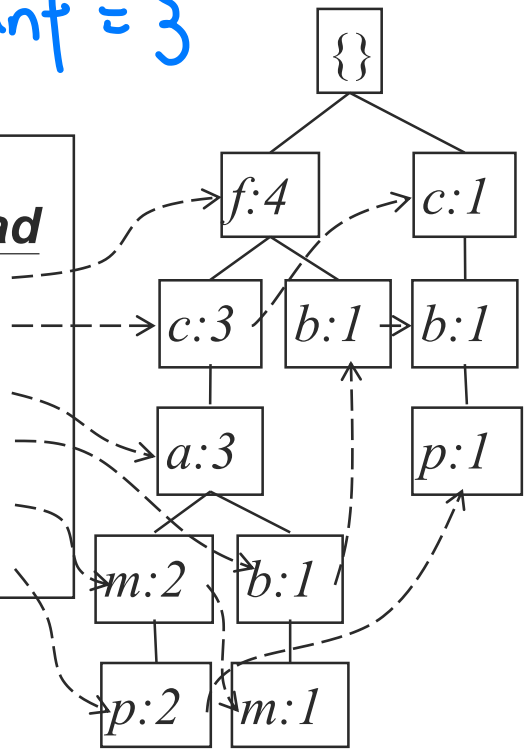
FP-Growth Overview (cont.)

minimum support count = 3

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

Header

Item	*frequency*	*head*
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Item	Cond. DB	Cond. FP-Tree	Frequent patterns
f			*f:4*
c	*f:3*	*f:3*	C:4 *fc:3*
a	*fc:3*	*fc:3*	*a:3, fca:3, fa:3, ca:3*
b	*fca:1, f:1, c:1*		*b:3*
m	*fca:2, fcab:1*	*fca:3*	*m:3, fm:3, cm:3, am:3, fcm:3, fam:3, cam:3,fcam:3*
p	*fcam:2, cb:1*	*c:3*	*p:3, cp:3*

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

Quantitative Association Rules

Quantitative Association Rules

- Association rule mining from

  relational data (attributed data)

  with quantitative attributes (numerical interval attributes)

Record ID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

Rule	Support	Confidence
<Age:30..39> and <Married:Yes> => <NumCars:2>	40%	100%
<Age:20..29> => <NumCars: 1>	40%	66.7%

如何運用 Transaction DB 的 Association Rule Mine Relational DB的 Quantitative Association Rule ?

Quantitative Association Rules

- Approach: transform into transaction data
  - Step 1: Discretization numerical attributes
  - Step 2: transform each distinct attribute-value pair into an item

Record ID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

Age	Married
20..29:1	Yes:1
30..39:2	No:2

Example

Min-Support=40%=2 records
Min-confidence=50%

Discretization of Age and Married

Age	Married
20..29:1	Yes:1
30..39:2	No:2

People

Record ID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

After Mapping Attributes

Record ID	Age	Married	NumCars
100	1	2	1
200	1	1	1
300	1	2	0
400	2	1	2
500	2	1	2

Itemset
{A1, M2, N1}
{A1, M1, N1}
{A1, M2, N0}
{A2, M1, N2}
{A2, M1, N2}

Rules: Sample

Rule	Support	Confidence
<Age:30..39> and <Married:Yes> => <NumCars:2>	40%	100%
<Age:20..29> => <NumCars: 1>	40%	66.7%
<Married:Yes> => <Numcars:2>	40%	66.7%

Discretization of Numerical Attributes

- Binning methods
  - Equi-width: the interval size of each bin is the same
  - Equi-depth: each bin has approximately the same number of tuples assigned to it.
- Distance-based method by clustering techniques: group neighboring tuples into the same bin based on distance measures

Price($)	Equi-width (width $10)	Equi-depth (depth 2)	Distance-based
7	[0,10]	[7,20]	[7,7]
20	[11,20]	[22,50]	[20,22]
22	[21,30]	[51,53]	[50,53]
50	[31,40]		
51	[41,50]		
53	[51,60]		

Discretization 會影響 Quantitative Association Rule 的產生嗎？ Yes

Example

Min-Support=40%=2 records
Min-confidence=50%

Discretization of Age and Married

Age	Married
20..24:1	Yes:1
25..29:2	No:2
30..34:3	
35..39:4	

People

Record ID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

After Mapping Attributes

Record ID	Age	Married	NumCars
100	1	2	1
200	2	1	1
300	2	2	0
400	3	1	2
500	4	1	2

Itemset
{A1, M2, N1}
{A2, M1, N1}
{A2, M2, N0}
{A3, M1, N2}
{A4, M1, N2}

Rules: Sample

Rule	Support	Confidence
<Married:Yes> => <Numcars:2>	40%	66.7%

Closed Association Rules

Maximal Frequent Itemset

- Large number of frequent itemsets

  (especially when the support threshold is **low**) and

  a huge number of association rules

  e.g. Given 2 transactions

1. {a1, a2, …, a50}
2. {a1, a2, ..., a50, ..., a100}

   with minimum support 50%, minimum confidence 50%

   ➔ 2^{100}-1 frequent itemsets

- Maximal frequent itemset

  A frequent itemset *X* is maximal

  if there exists no itemset set *X'* such that

  X' is a proper superset of *X*.

  (i.e. none of its immediate supersets is frequent)

Maximal Frequnet Itemset

- An itemset is maximal frequent
  if none of its immediate supersets is frequent

TID	Items
1	A, B
2	B, C, D
3	A, B, C, D
4	A, B, D
5	A, B, C, D

Itemset	Support
A	4
B	5
C	3
D	4
A, B	4
A, C	2
A, D	3
B, C	3
B, D	4
C, D	3
A, B, C	2
A, B, D	3
A, C, D	2
B, C, D	3
A, B, C, D	2

Closed Frequent Itemset

- Maximal frequent itemsets do not contain the support information of their subsets.
- **Maximal frequent itemset**

  A frequent itemset *X* is a maximal itemset

  if there exists no itemset set *X'* such that

  X' is a proper superset of *X*.
- **Closed frequent closed itemset**

  A frequent itemset *X* is a **closed itemset**

  if there exists no itemset *X'* such that

  – (1) *X'* is a proper superset of *X*.

  – (2) every transaction containing *X* also contains *X'*.

Closed Itemset

- An itemset is closed if
  none of its immediate supersets has the same support as the itemset

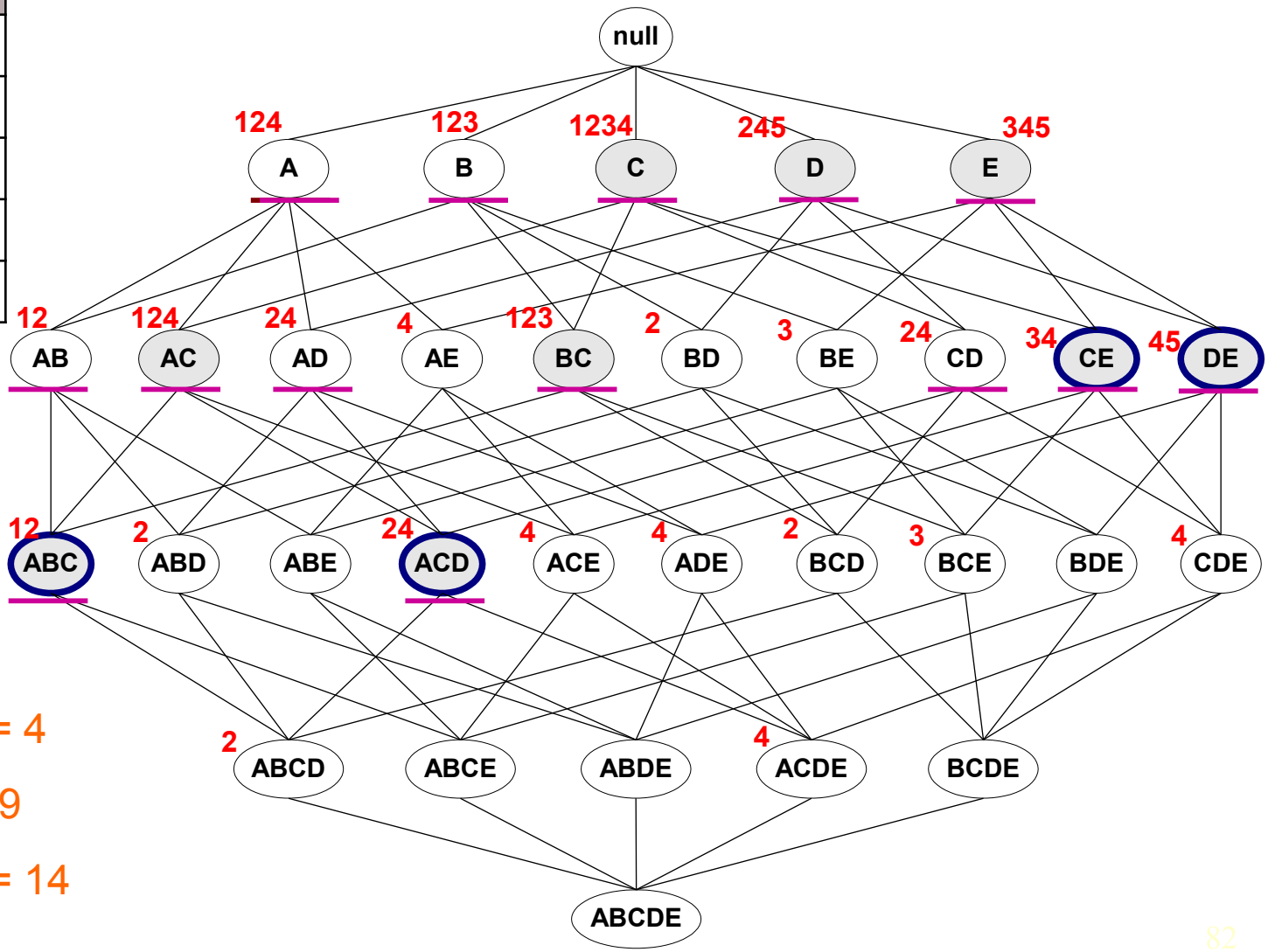
TID	Items
1	A, B
2	B, C, D
3	A, B, C, D
4	A, B, D
5	A, B, C, D

Itemset	Support
A	4
B	5
C	3
D	4
A, B	4
A, C	2
A, D	3
B, C	3
B, D	4
C, D	3
A, B, C	2
A, B, D	3
A, C, D	2
B, C, D	3
A, B, C, D	2

Maximal vs. Closed Itemsets

TID	Items
1	A, B, C
2	A, B, C, D
3	B, C, E
4	A, C, D, E
5	D, E

Minimum Support = 2

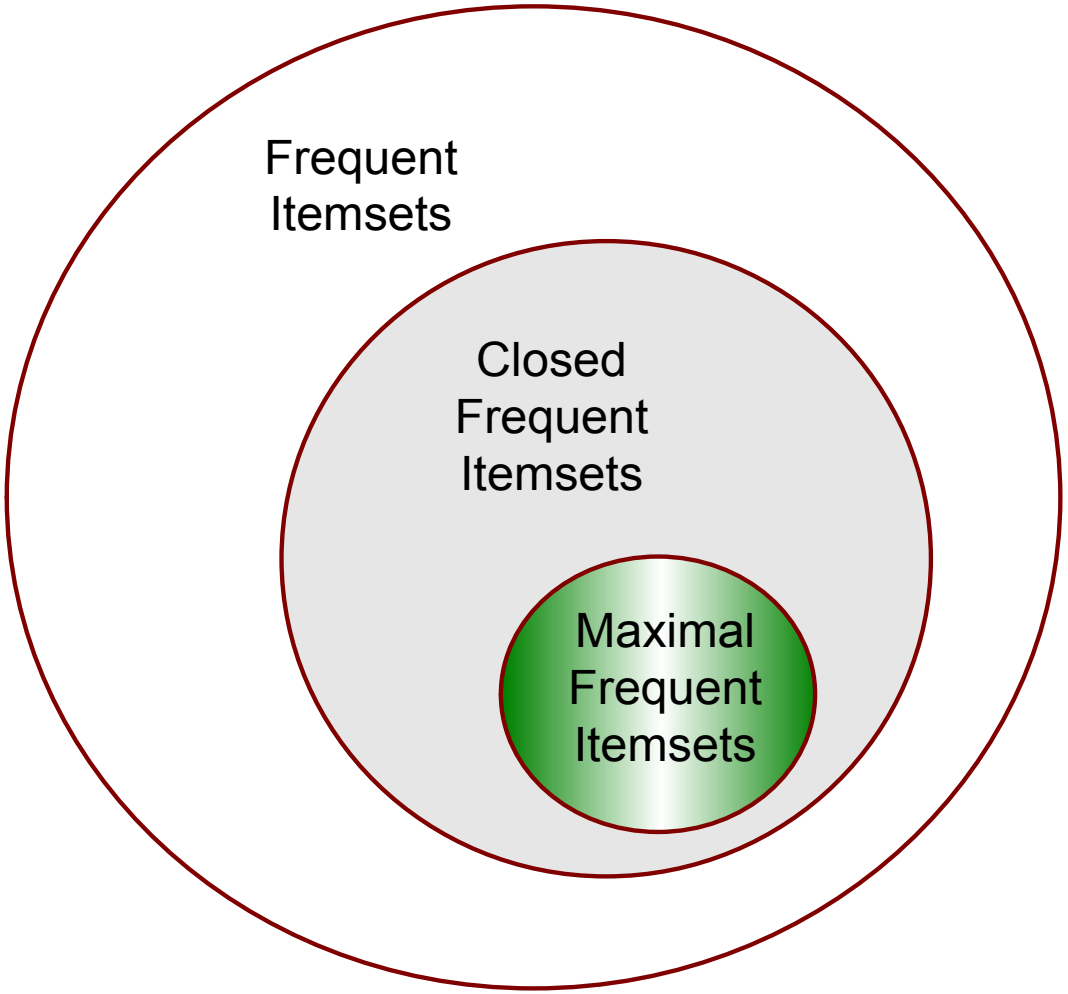


Maximal = 4

Closed = 9

#Frequent = 14

Maximal vs. Closed Itemsets



Closed Association Rules

- Association rule on frequent closed itemsets:

Rule $X \Rightarrow Y$ is an association rule on frequent closed itemsets if

  - (1) both X and $X \cup Y$ are frequent closed itemsets.
  - (2) there does not exist frequent closed itemset Z such that $X \subset Z \subset (X \cup Y)$.
  - (3) the confidence of the rule passes the given confident threshold.

Closed Association Rules (cont.)

TID	Items
100	a,c,d,e,f
200	a,b,e
300	c,e,f
400	a,c,d,f
500	c,e,f

Given minimum support 2

Closed Frequent Itemsets:

{a, c, d, f}, {c, e, f}, {a, e}, {c, f}, {a}, {e}

Given minimum confidence 50%,

Closet association rule

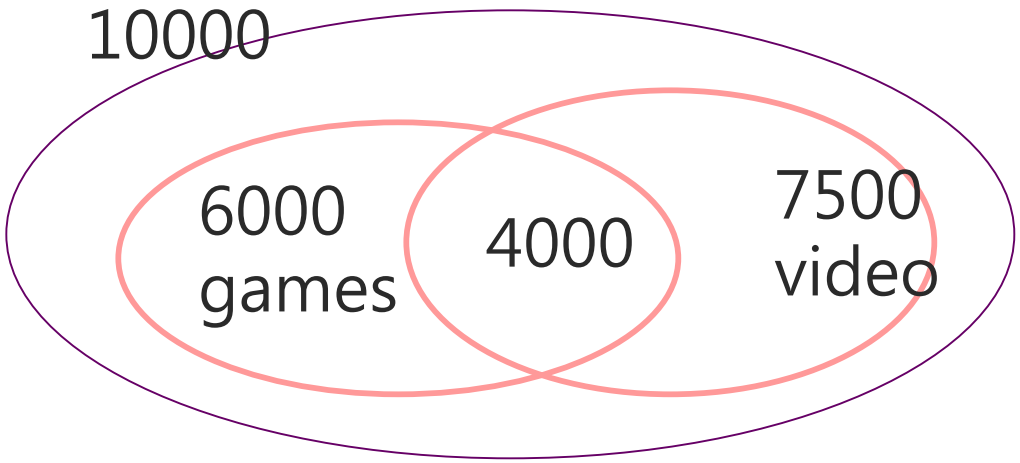
{c, f} $\Rightarrow$ {a, d} (2,50%), {a} $\Rightarrow$ {c, d, f} (2,67%),

{e} $\Rightarrow$ {c, f} (3,75%), {c, f} $\Rightarrow$ {e} (3,75%),

{e} $\Rightarrow$ {a} (2,50%), {a} $\Rightarrow$ {e} (2,67%)

From Association Mining to Correlation Analysis

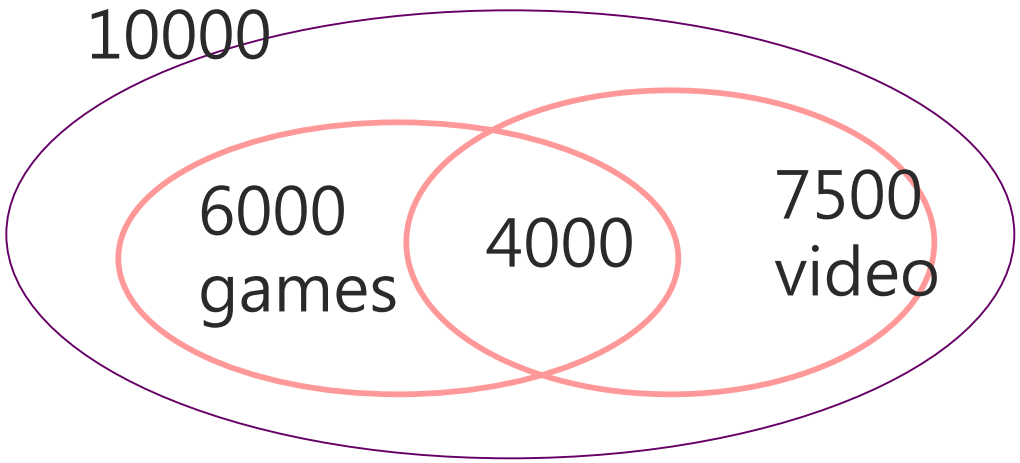
Strong Rules & Interesting



- Games $\rightarrow$ Videos,

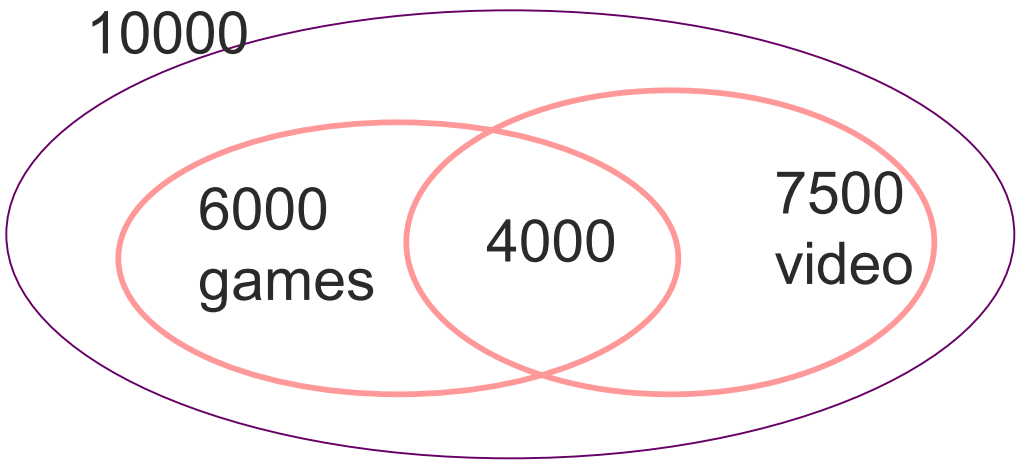
  support = 4000/10000 = 40%, confidence = 4000/6000 = 66%
- Prob(Videos) = 7500/10000 = 75%
- In fact, games & videos are negatively associated
- Purchase of games actually decrease the likelihood of purchasing videos

Correlation Analysis



- Corr(A,B) = P(A & B) / (P(A) P(B))
- E.g. Corr(games, videos)=0.4/(0.6*0.75)=0.89
  - Corr(A, B)=1, A & B are independent
  - Corr(A, B)<1, occurrence of A is negatively correlated with B
  - Corr(A, B)>1, occurrence of A is positively correlated with B

Contingency Table



	Game	Game’	Total
Video	4000	3500	7500
Video’	2000	500	2500
Total	6000	4000	10000

Solution

- A correlation rule A→B is measured
  not only by its support & confidence
  but also by the correlation between A and B

- Approach
  - Lift(A, B) = P(A∪B)/P(A)P(B)
    = Conf(A→B)/Sup(B)

A → B

support	P(A∪B)
confidence	P(A∪B) / P(A)
lift	P(A∪B) / P(A)P(B)

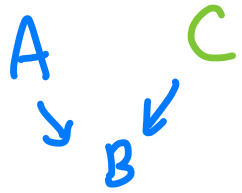
A和B

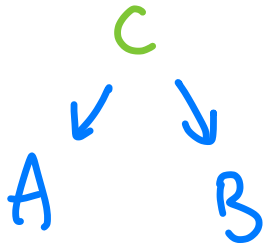
有相關

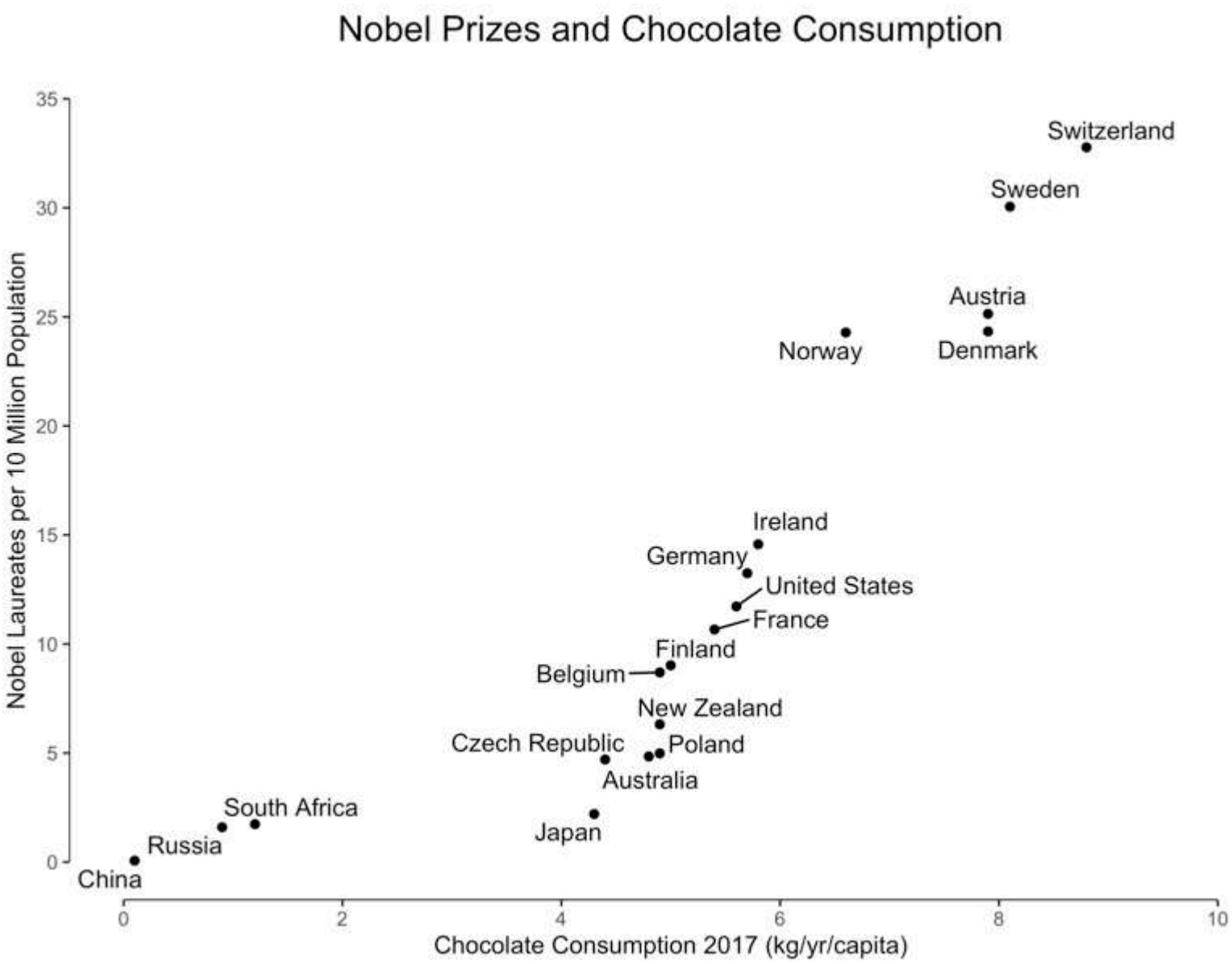
Correlation

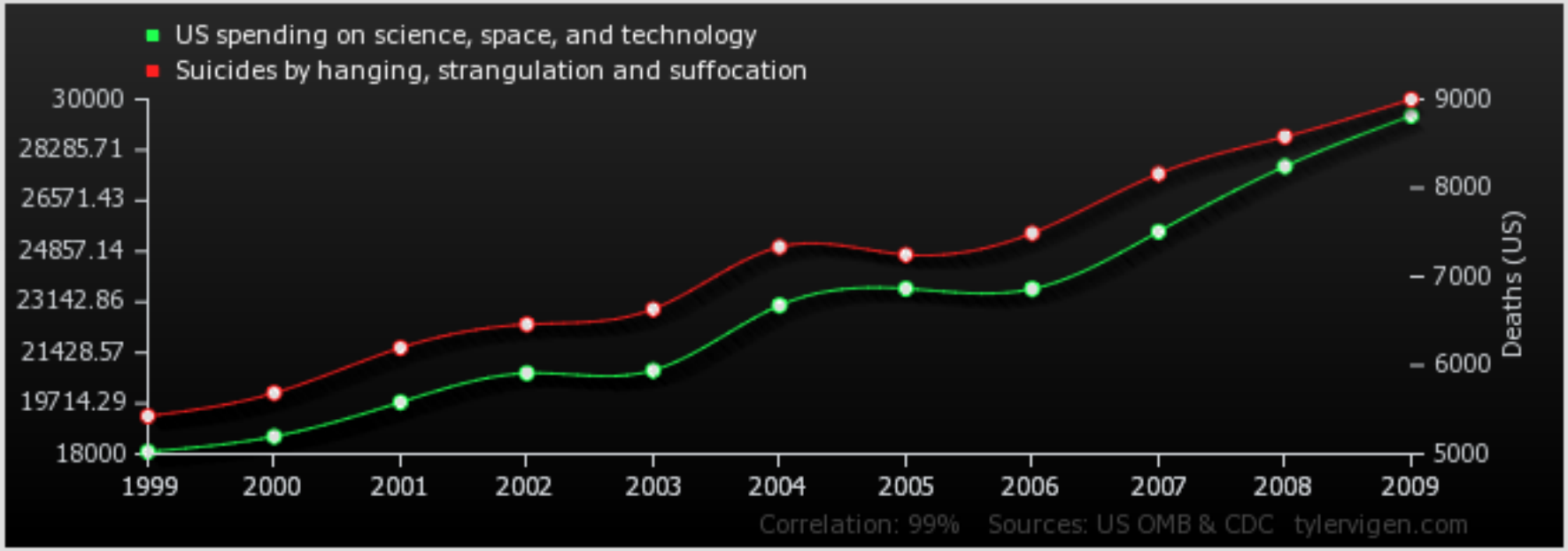
邏輯上的5種可能

- A造成B
- B造成A (因果倒置)
- A造成B，但是，C也會造成B (一果多因)
- C造成A，同時，C也會造成B (隱藏變因)
- 純屬巧合





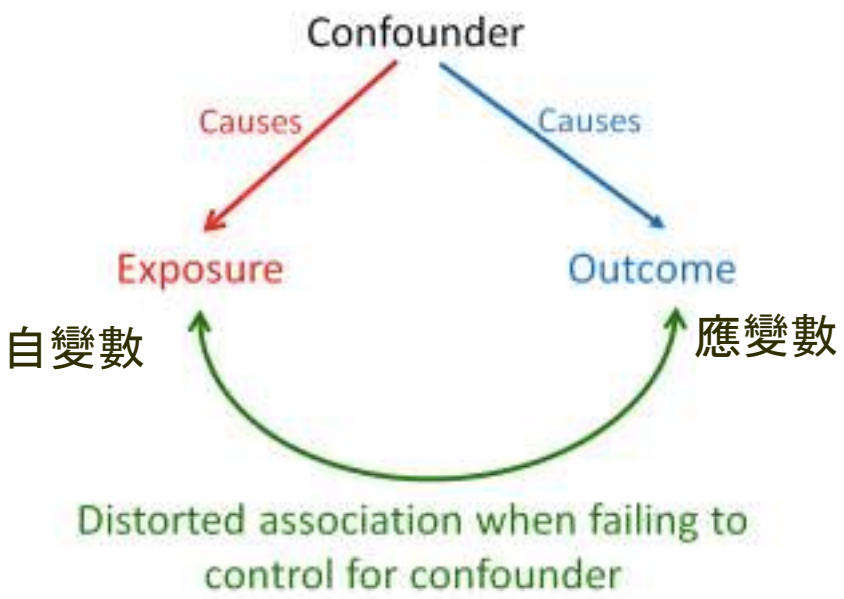

Nobel Prizes and Chocolate Consumption
Nobel Laureates per 10 Million Population
Chocolate Consumption 2017 (kg/yr/capita)
0
5
10
15
20
25
30
35
0
2
4
6
8
10
Switzerland
Sweden
Austria
Norway
Denmark
Ireland
Germany
United States
France
Finland
Belgium
New Zealand
Czech Republic
Poland
Australia
South Africa
Japan
Russia
China



	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009
US spending on science, space, and technology *Millions of todays dollars (US OMB)*	18,079	18,594	19,753	20,734	20,831	23,029	23,597	23,584	25,525	27,731	29,449
Suicides by hanging, strangulation and suffocation *Deaths (US) (CDC)*	5,427	5,688	6,198	6,462	6,635	7,336	7,248	7,491	8,161	8,578	9,000

Correlation: 0.992082

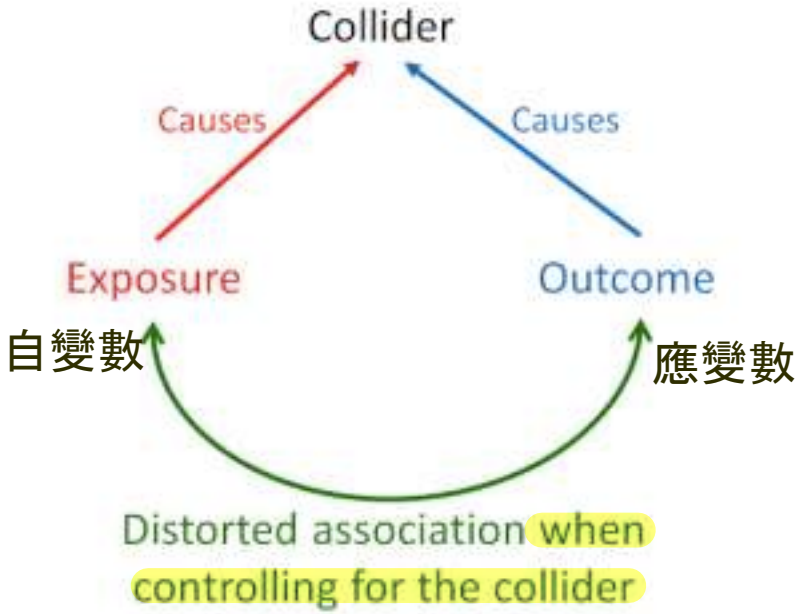
干擾因子



Latent Variable

隱藏變因

對撞因子



Berkson's Paradox

一果多因

棉花糖實驗

- 史丹福大學沃爾特·米歇爾博士1966年到1970年代早期在kindergarten 進行的有關自制力的一系列心理學經典實驗
- 在實驗中，小孩可以選擇

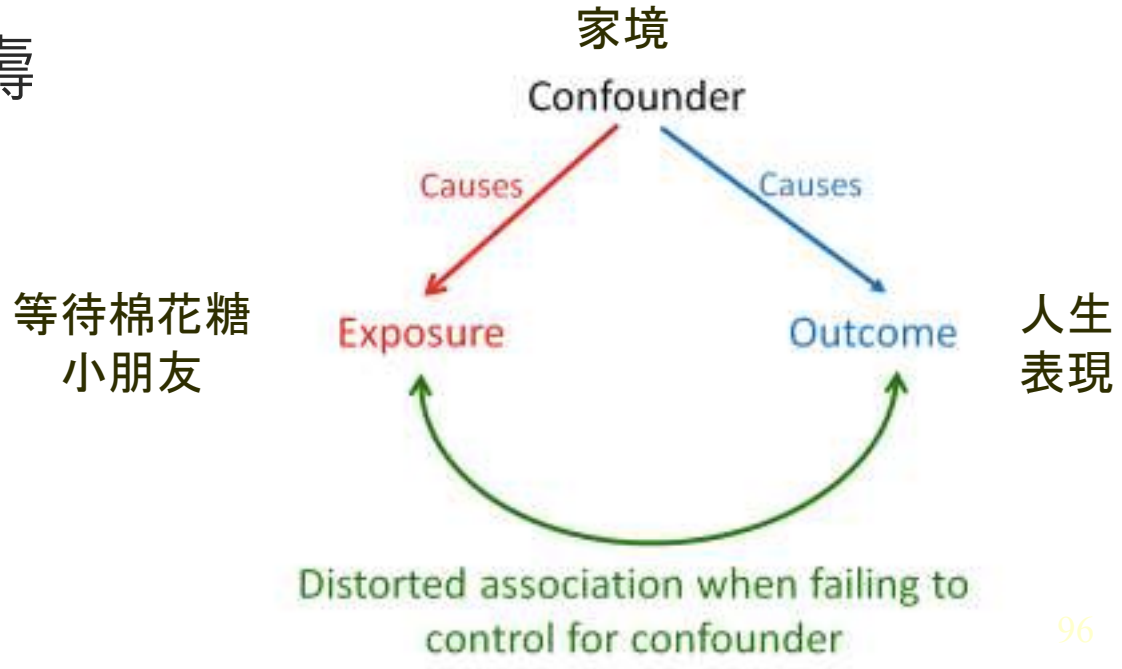
(1) 立即得到 1 份獎勵（棉花糖、餅乾、巧克力等），or

(2) 等待15分鐘，得到 2 份獎勵

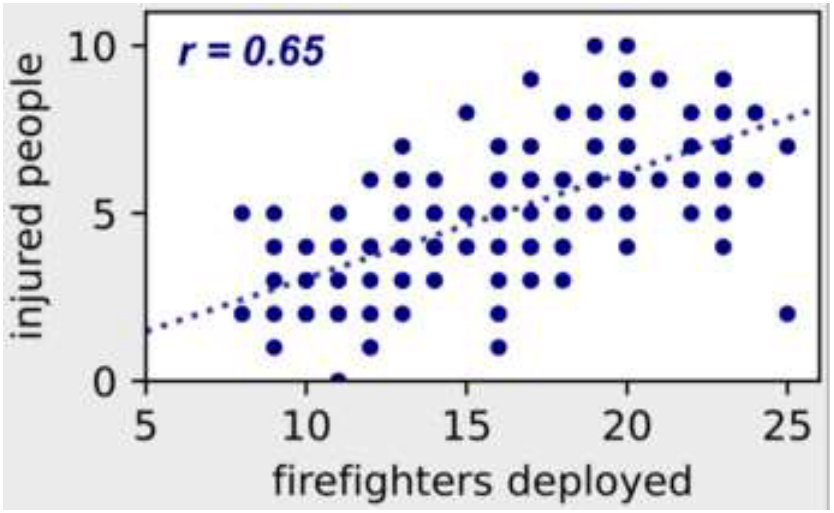
- 實驗結果：選擇等待的小孩，長大後有更好的SAT成績、教育成就、BMI等

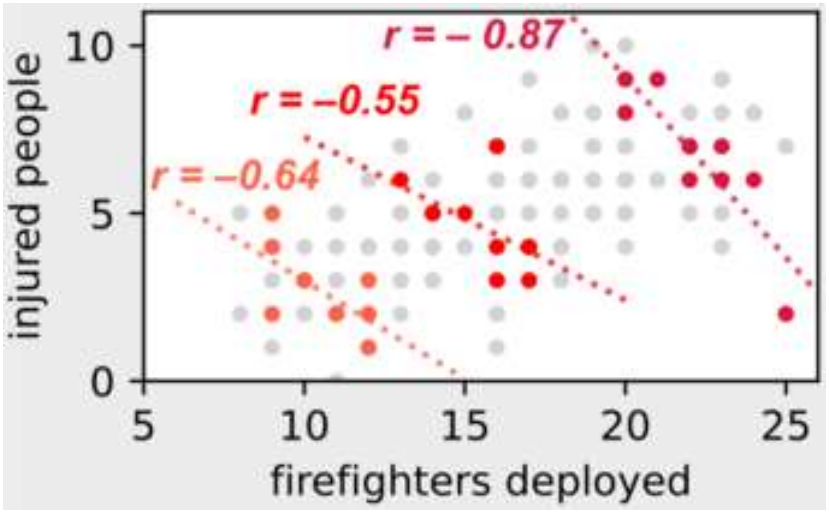
Confounder

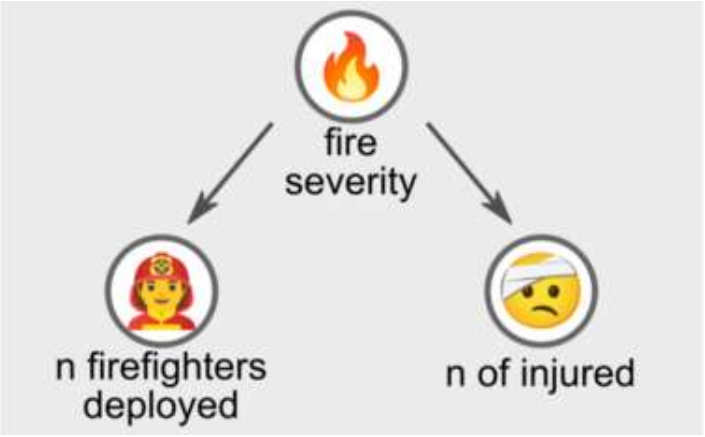
- 選擇等待棉花糖的小朋友(i.e 延遲滿足能力)的小朋友有更好的人生表現。
- 冰淇淋銷售量越高，溺水死亡人數越多
- 哈佛畢業生薪水比它校畢業生高
- 睡前喝紅酒比較長壽



Positive association between Number of firefighters and Number of injured !?!







Simpson's Paradox

- Example:
  - Accept rate of male: 35/80=44%
  - Accept rate of female: 20/60=33%

	Male	Female	Total
Accept	35	20	55
Reject	45	40	95
Total	80	60	140

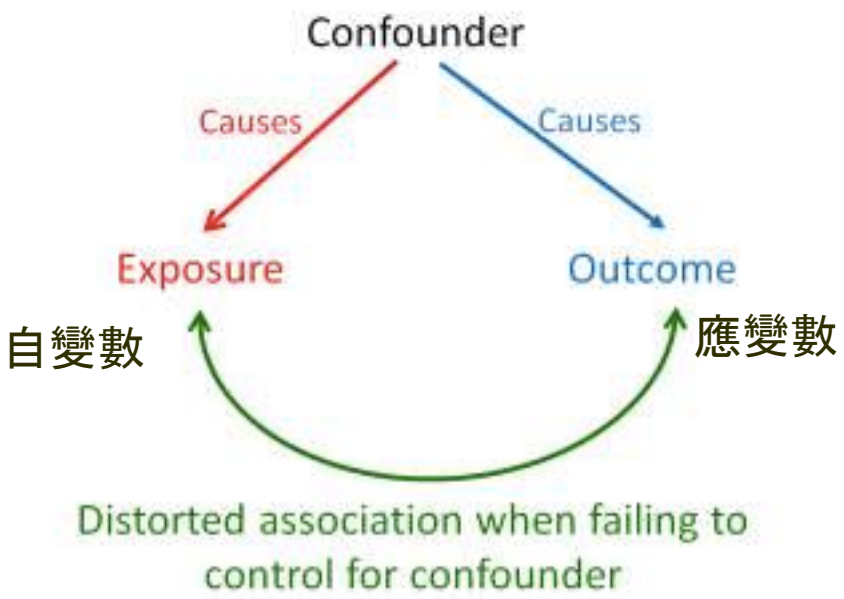
Simpson's Paradox (cont.)

- Example: Why? lurking variable

	Male	Female	Total
Accept	35	20	55
Reject	45	40	95
Total	80	60	140

	Electric Engineering		Foreign Language	
	Male	Female	Male	Female
Accept	30	10	5	10
Reject	30	10	15	30
Total	60	20	20	40

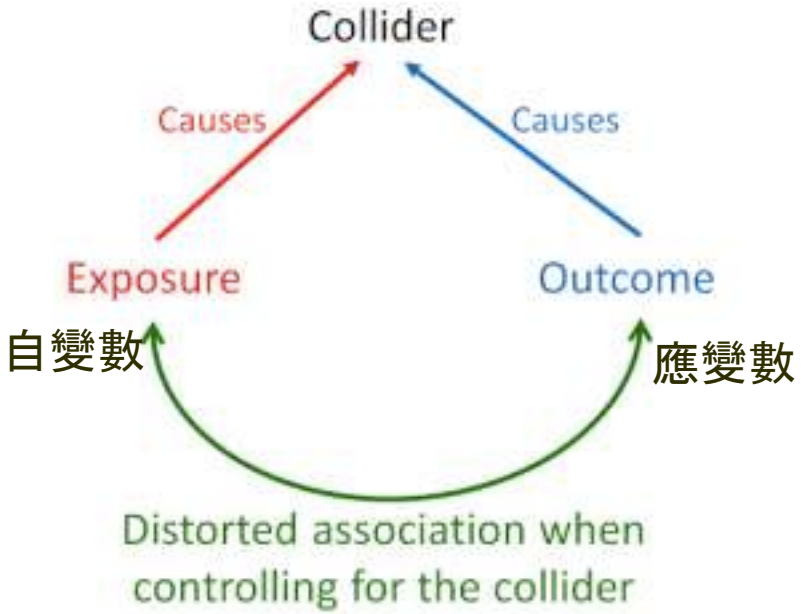
干擾因子



Latent Variable

隱藏變因

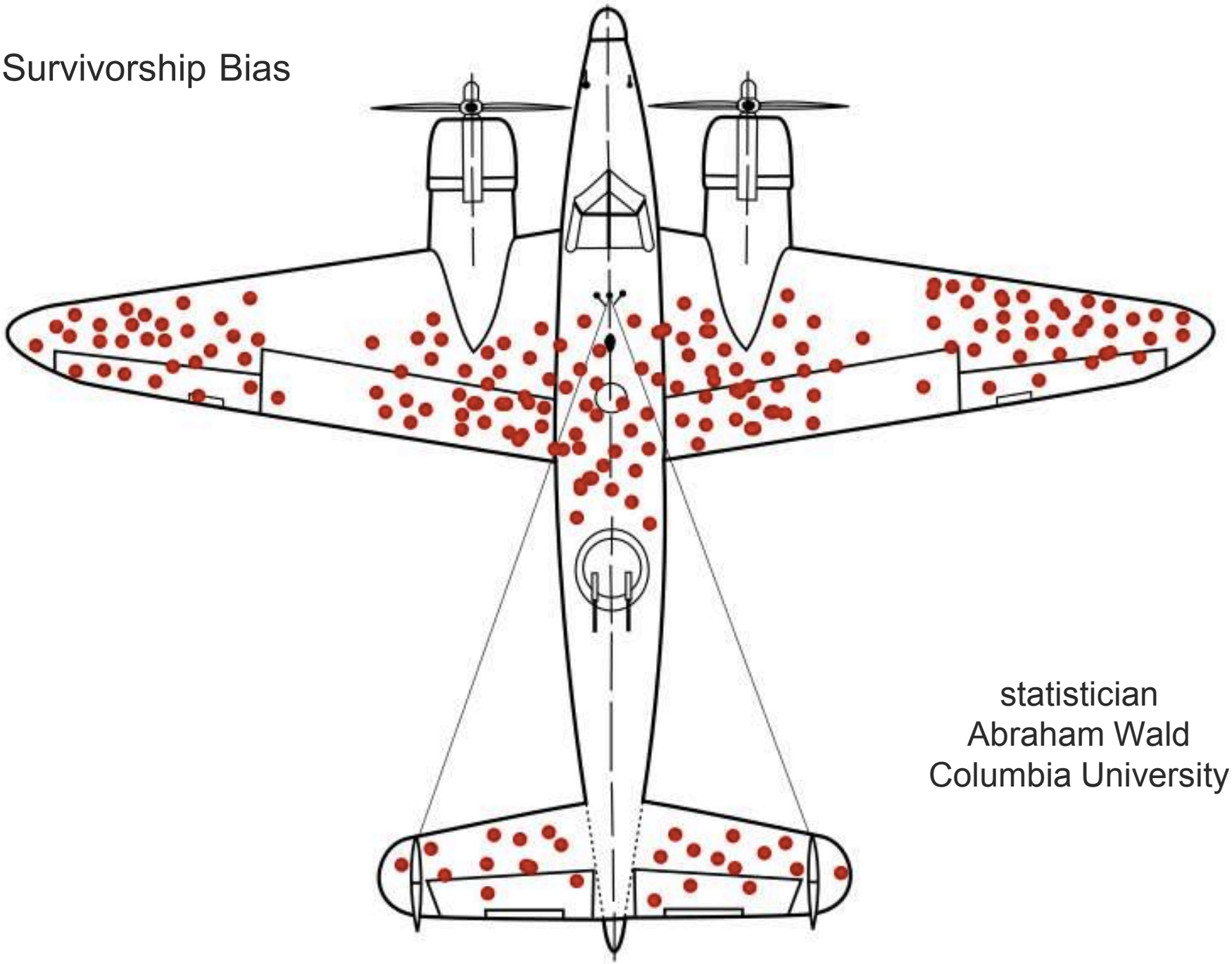
對撞因子



Berkson's Paradox

一果多因

Survivorship Bias



statistician
Abraham Wald
Columbia University

Negative association between COVID-19 severity and smoking cigarettes !?!

European Commission

JRC Publications Repository

Home Search Help

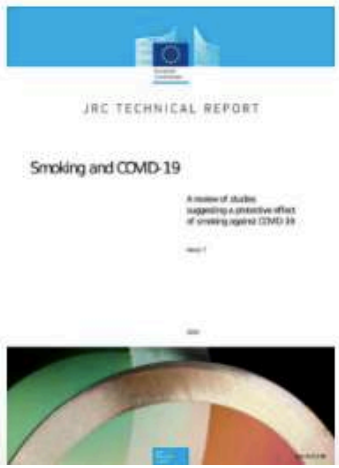
European Commission > JRC > JRC Publications Repository >

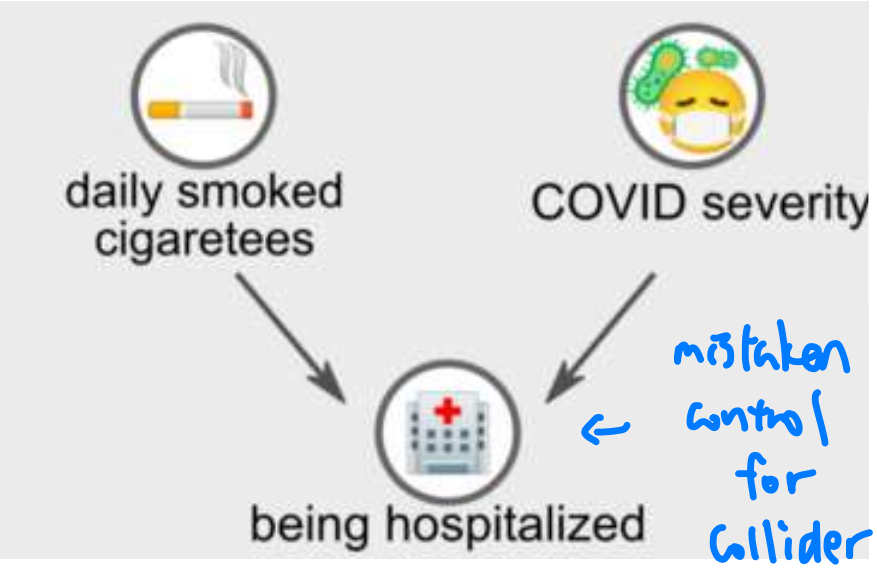
Smoking and COVID-19 - A review of studies suggesting a protective effect of smoking against COVID-19

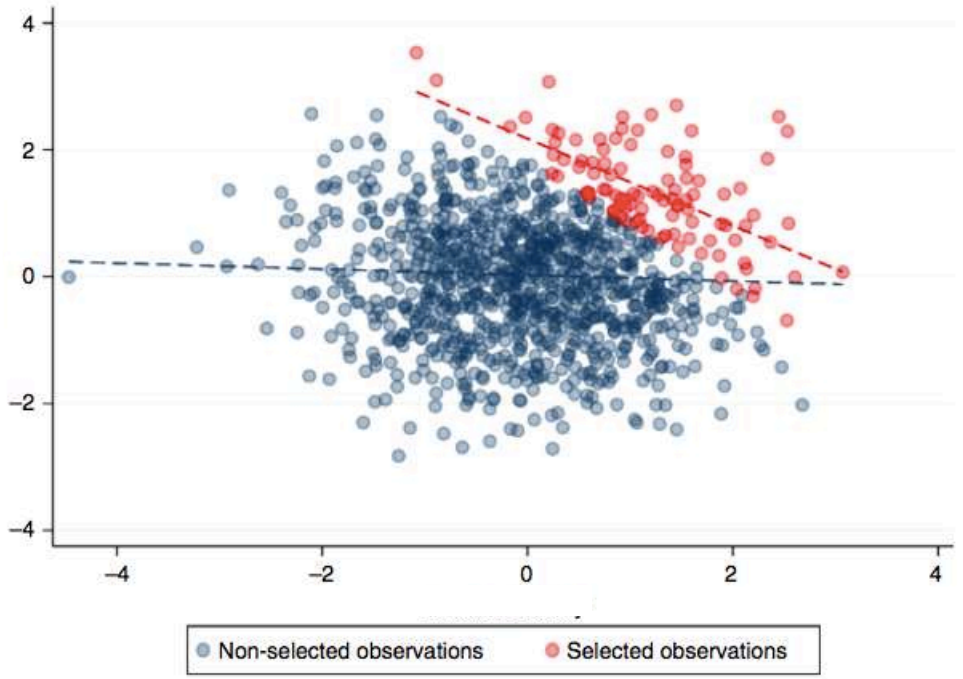
2020 Technical reports Health and consumer protection

Smoking and COVID-19 - A review of studies suggesting a protective effect of smoking against COVID-19

Abstract: The risk factors for contracting symptomatic COVID-19 are not yet fully understood, age and certain underlying health conditions are considered to be detrimental in this respect. Case studies revealed an astonishingly low number of current smokers among patients suffering from symptomatic COVID-19 compared to the general population, leading to the conclusion that smoking/nicotine uptake might have a preventive effect. This is difficult to understand seeing that studies found an increased expression of the angiotensin-converting enzyme (ACE-2) in smokers, the entrance gate of the coronavirus into human cells. Consequently, the use of the proportion of smokers in the general population as a reference for deriving prevalence ratios to study the association of smoking with COVID-19 disease outcomes may be inappropriate. Prevalence data for smoking and comorbidities (hypertension, diabetes mellitus, and chronic obstructive pulmonary disease) reported in 25 studies, which partially identified a potentially beneficial effect of smoking/nicotine intake, were re-analysed to investigate the relationship between COVID-19 mortality and national smoking prevalence taking account of







- Having high severity of COVID-19 increases chances of being hospitalized.
- Smoking several cigarettes a day is a major risk factor for a variety of diseases, which increase the chances of being hospitalized.
- if a hospital patient has lower COVID-19 severity, they have higher chances of smoking cigarettes! Indeed, they must have some disease different from COVID-19 (e.g. heart attacks, cancer, diabetes) to justify their hospitalization, and this disease may very well be caused by their smoking cigarettes.

* Collider bias undermines our understanding of COVID-19 disease risk and severity, Nature Communications, 2020.


A和B
有相關
邏輯上的5種可能
A造成B
B造成A (因果倒置)
A造成B，但是，C也會造成B (一果多因)
C造成A，同時，C也會造成B (隱藏變因)
純屬巧合

DATA SCIENCE

When Correlation is Better than Causation

A heuristic approach for using correlations to inform decisions

BRITTANY DAVIS

11 AUG 2021 • 7 MIN READ

Other Forms of Association Rules

Mining Association Rule in Temporal DB

- Temporal Databases
- e.g. To discover if some diseases are likely to cause some other diseases (complication) with minimum support =3

➔ the timing constraint

= the tuples whose duration overlap with each other

➔ {B, A∪C} (i.e.{R1,R2}, {R3,R4}, {R5,R6})

	Patient ID	Disease	Start	End
R1	1	A	1	3
R2	1	B	2	7
R3	1	C	9	12
R4	1	B	10	15
R5	2	A	3	6
R6	2	B	5	7

Mining Inter-Transaction Association Rules

- Intra-transaction association rules:

  e.g. When the prices of Apple and Google go up, at 80% of probability the price of Facebook goes up on the same day

- Inter-transaction association rules:

  e.g. If the price of Apple and Google go up, Facebook will most likely (80% probability) go up the next day and then drop four days later.

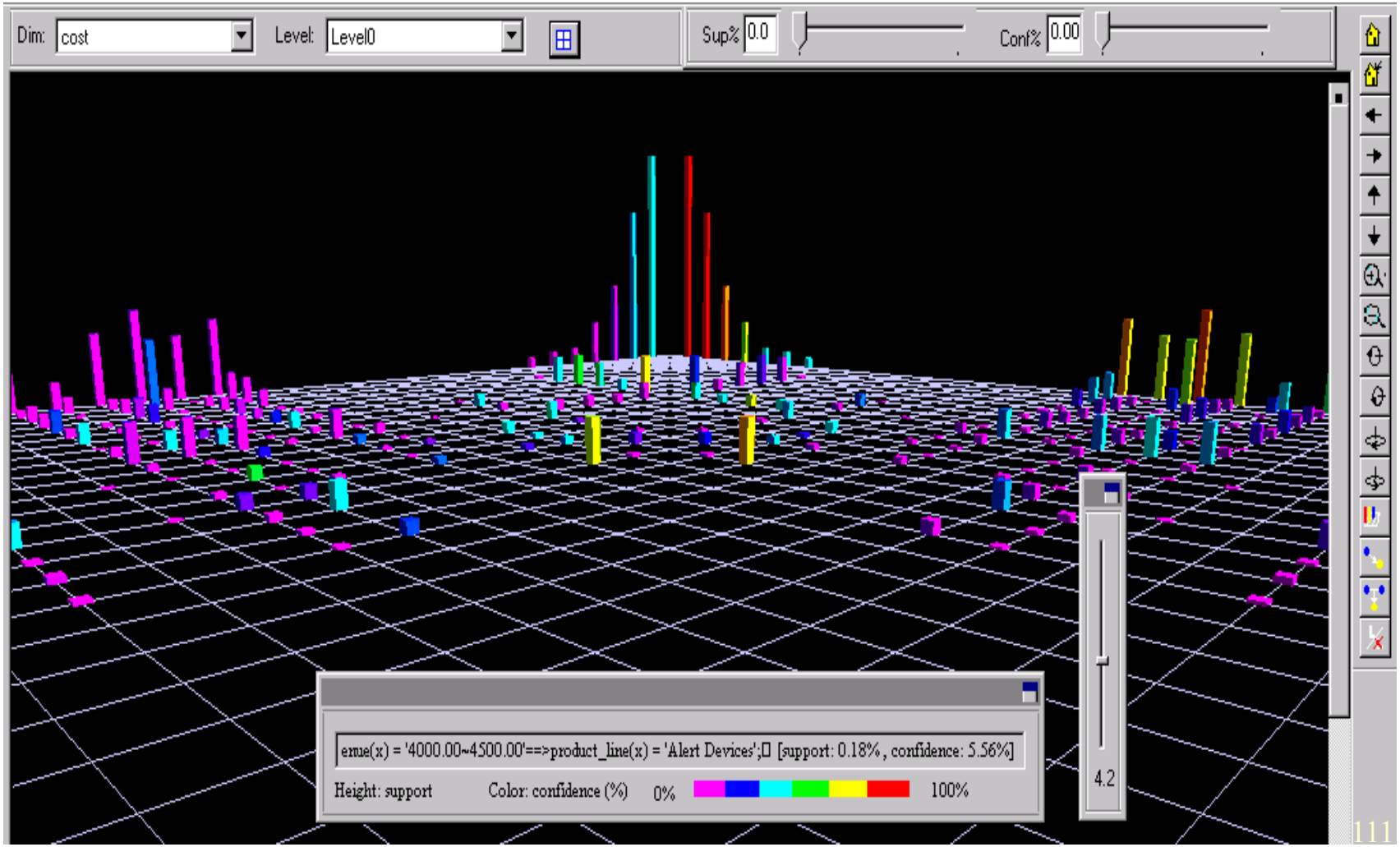
Visualization of Association Rules

Presentation of Association Rules

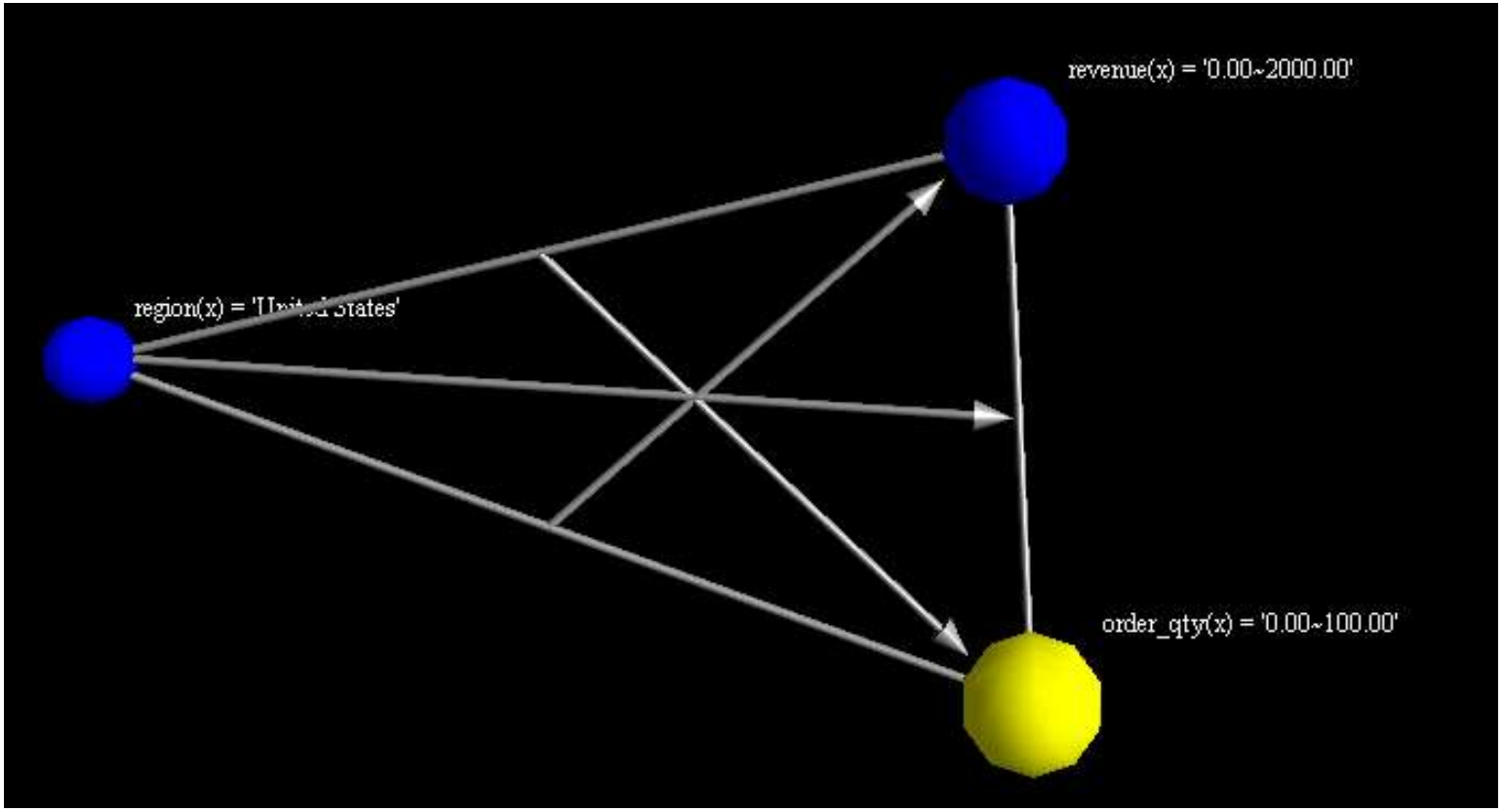
	Body	Implies	Head	Supp (%)	Conf (%)	F	G	H	I
1	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00'	28.45	40.4				
2	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00'	20.46	29.05				
3	cost(x) = '0.00~1000.00'	==>	order_qty(x) = '0.00~100.00'	59.17	84.04				
4	cost(x) = '0.00~1000.00'	==>	revenue(x) = '1000.00~1500.00'	10.45	14.84				
5	cost(x) = '0.00~1000.00'	==>	region(x) = 'United States'	22.56	32.04				
6	cost(x) = '1000.00~2000.00'	==>	order_qty(x) = '0.00~100.00'	12.91	69.34				
7	order_qty(x) = '0.00~100.00'	==>	revenue(x) = '0.00~500.00'	28.45	34.54				
8	order_qty(x) = '0.00~100.00'	==>	cost(x) = '1000.00~2000.00'	12.91	15.67				
9	order_qty(x) = '0.00~100.00'	==>	region(x) = 'United States'	25.9	31.45				
10	order_qty(x) = '0.00~100.00'	==>	cost(x) = '0.00~1000.00'	59.17	71.86				
11	order_qty(x) = '0.00~100.00'	==>	product_line(x) = 'Tents'	13.52	16.42				
12	order_qty(x) = '0.00~100.00'	==>	revenue(x) = '500.00~1000.00'	19.67	23.88				
13	product_line(x) = 'Tents'	==>	order_qty(x) = '0.00~100.00'	13.52	98.72				
14	region(x) = 'United States'	==>	order_qty(x) = '0.00~100.00'	25.9	81.94				
15	region(x) = 'United States'	==>	cost(x) = '0.00~1000.00'	22.56	71.39				
16	revenue(x) = '0.00~500.00'	==>	cost(x) = '0.00~1000.00'	28.45	100				
17	revenue(x) = '0.00~500.00'	==>	order_qty(x) = '0.00~100.00'	28.45	100				
18	revenue(x) = '1000.00~1500.00'	==>	cost(x) = '0.00~1000.00'	10.45	96.75				
19	revenue(x) = '500.00~1000.00'	==>	cost(x) = '0.00~1000.00'	20.46	100				
20	revenue(x) = '500.00~1000.00'	==>	order_qty(x) = '0.00~100.00'	19.67	96.14				
21									
22									
23	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.45	40.4				
24	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.45	40.4				
25	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	19.67	27.93				
26	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	19.67	27.93				
27	cost(x) = '0.00~1000.00' AND order_qty(x) = '0.00~100.00'	==>	revenue(x) = '500.00~1000.00'	19.67	33.23				

Sheet1

Association Rule Visualization



Association Rule Visualization



Other Issue of Association Rules Mining

- Infrequent pattern mining: rare patterns
- Negative association rules

  {Coca Cola} $\rightarrow$ ~ {Pepsi Cola}
- Incremental and interactive association rule mining
  - incorporate database updates without having to mine the entire database again from scratch
- Distributed association rule mining (parallel)
- Privacy preserving

How to discover the following frequent patterns using Apriori approach ?

- Infrequent patterns: rare patterns, anomaly pattern
- Negative association rules

  {Coca Cola} $\rightarrow$ ~ {Pepsi Cola}
- Frequent sequences
- Frequent graphs
- Repeating patterns
- Periodical patterns

Frequent Sequence

Sequence Database

TID	Items
100	ACD
200	EBC
300	AEBC
400	BE

Frequent Sequence

sequence	Sup
AC	2
BC	2
EB	2
EC	2
EBC	2

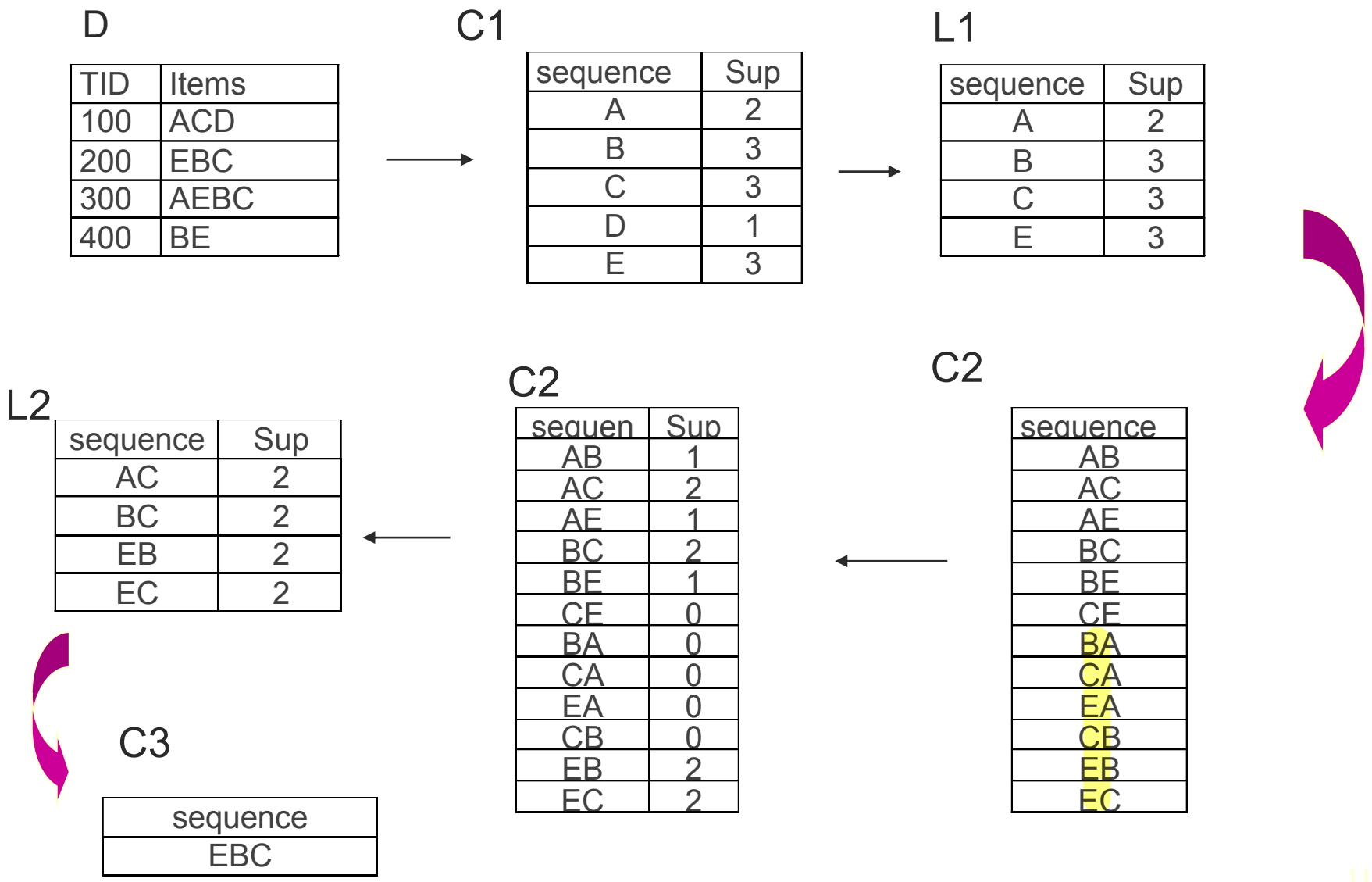
Sequence Database

TID	Items
100	CAD
200	BEC
300	AEBC
400	BE

Frequent Sequence

sequence	Sup
BC	2
BE	2
EC	2

Apriori Algorithm for Frequent Sequence: An Example



Sequential Pattern Mining

- Sequential pattern
  - Subsequence of itemsets that appears frequently over a set of itemset sequences

transaction ID customer ID

TID	CID	Item
T10	100	C
T20	200	A, B
T30	300	C, E, G
T40	400	C
T50	100	H
T60	200	C
T70	400	D, G
T80	200	D, F, G
T90	500	H
T100	400	H

raw transactions

sequences

CID	Items
100	<(C)(H)>
200	<(A, B)(C)(D, F, G)>
300	<(C, E, G)>
400	<(C)(D, G)(H)>
500	<(H)>

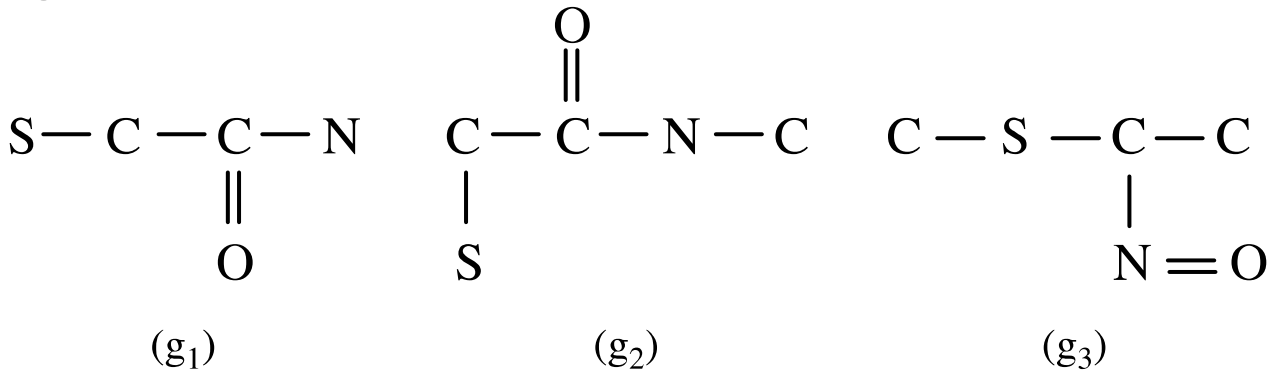
Sequential pattern >25%
<(C)(H)> <(C)(D, G)>

Summary

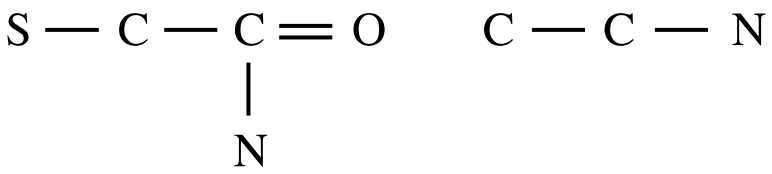
- Frequent Patterns: appear in a data frequently
  - Frequent itemset mining (from a set of itemsets)
  - Association rule mining (from a set of itemsets)
  - Frequent sequence mining (from a set of sequences)
  - Sequential pattern mining (from a set of sequences of itemset)
  - Frequent graph mining (from a set of graphs)
  - Periodical pattern mining (from a sequence of events)
  - Repeating pattern mining (from a sequence of events)

Frequent Graph Mining

- Frequent graph
  - sub-graph that appears frequently over a set of graphs



(g_1) (g_2) (g_3)



frequency: 2 frequency: 3

Periodical Pattern

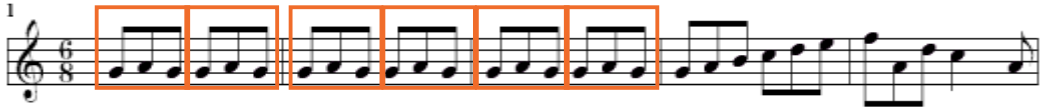
- Find the partial periodical pattern from a sequence of events.

AQCXDBQCFDCQCADY

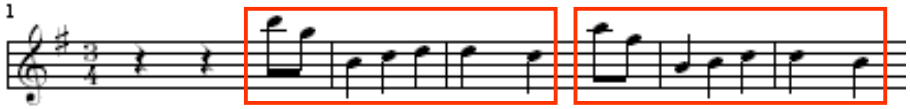
*QC*D

Repeating Patterns

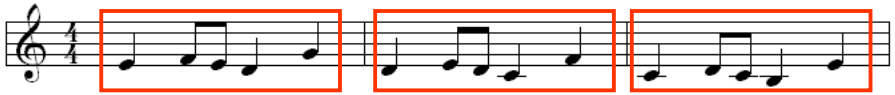
- Motif discovery: variation of repeating pattern



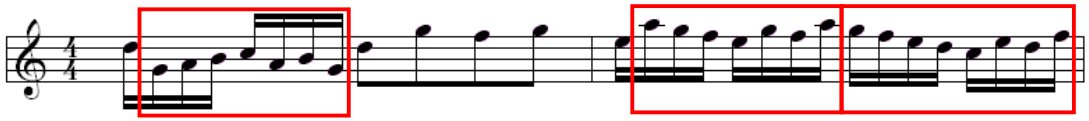
Exact repeat



Interval repeat

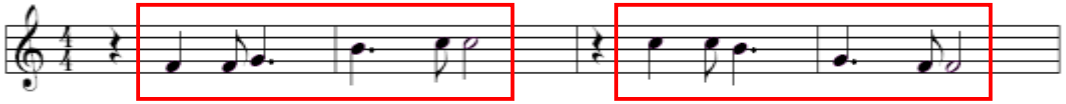


Sequence

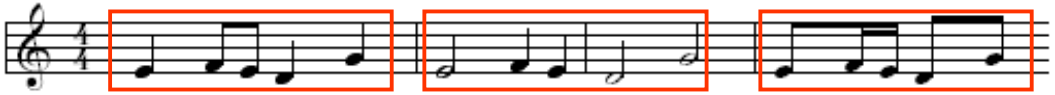


Contrary motion

1 1 1 -2 1, -2 -1 -1 -1 2 -1 2



Retrograde



Augmentation or Diminution

Summary (cont.)

- Approaches of Frequent Itemset Mining
  - Apriori Algorithm
  - DHP
  - Partitioned Approach
  - FP-Tree
- Quantitative Association Rule from Relational FB
- Maximal, Closed Itemset
- Interesting Measure
- Variations of Association Rules

LINE貼圖 關聯規則分析

EKONOMIPRISET 2021
THE PRIZE IN ECONOMIC SCIENCES 2021

KUNGL. VETENSKAPS-AKADEMIEN
THE ROYAL SWEDISH ACADEMY OF SCIENCES


Photo: UC Berkeley

David Card, USA

”för hans empiriska bidrag till arbetsmarknadsekonomi”

“for his empirical contributions to labour economics”


Photo: Creative Commons Wiki

Joshua D. Angrist, USA


Photo: Stanford Graduate School of Business

Guido W. Imbens, USA

”för deras metodologiska bidrag till analysen av kausala samband”

“for their methodological contributions to the analysis of causal relationships”

#nobelprize

THE NOBEL PRIZE

Causal Inference

2021諾貝爾經濟學獎，為何是一場靜悄悄的革命？

如何判斷一場革命有沒有成功？(端傳媒 評論 By Ye Wang 2021/10/14)

- 「革命勝利了！」在今年諾貝爾經濟學獎得主公布之後，推特上一眾社會科學學者異口同聲地發出了感慨。
- 他們口中的革命，是發軔於統計學，並逐漸擴散到社會科學各個領域，由因果推斷（causal inference）方法驅動，悄然間改變了實證研究基本面貌的「可置信性革命（credibility revolution）」。
- 如今，當你翻開一篇社會科學中的實證論文，有很大概率會發現如下字眼的身影：「識別策略（identification strategy）」、「內生性（endogeneity）」、「準隨機分配（quasi-random assignment）」，亦或「自然實驗（natural experiment）」。雖然含義略有不同，但它們都體現了同樣的思想：想要論證從X到Y的因果關係，我們必須要依賴X獨立於Y發生的隨機變動。這一變動可以來自研究者的人為干預，即真正的對照實驗，也可以源於出乎意料的外生政策或事件衝擊。在後一種情況中，研究者無法控制隨機分配的過程，只能觀測到最終的結果，就彷彿是在自然中恰好撞見了一場由第三方執行完畢的對照實驗。因此這種情況得名「準實驗」或「自然實驗」。

Causal Inference (cont.)

- 一個經典的例子是今年諾獎得主Angrist於1990年發表於《美國經濟評論》（American Economic Review）的論文。他感興趣的問題是，服兵役會給個體未來的收入帶來怎樣的改變。顯然，直接對比有無參軍經歷者當下的工資水平，得到的估計並不準確。因為具備某些特質（比如身體強壯或服從紀律）的個體參軍意願更高，而這些特質又會影響他們在勞動力市場上的表現。因此，我們很難知道，工資水平的差異究竟完全是由兵役導致，還是源自個體在其他方面的差異。這些會對因果識別產生干擾的差異，被統計學家們形象地稱為「混淆變量（confounder）」。

- 如果我們可以開展一場實驗，隨機地決定每名被試需不需要參軍，那自然就可以排除混淆變量的干擾。只不過，這樣的實驗若由學者執行，必然違反倫理，不具備可行性。Angrist獨闢蹊徑，考察了七十年代初美國政府在越戰期間進行的軍事動員。當時的美國國防部出於公平性的考慮，採用了抽籤的方式來決定每名適齡男性是否要應徵入伍。Angrist的分析發現，被抽中的越戰老兵跟未參戰的同齡人相比，在八十年代的收入要低15%。由於抽籤的隨機性，這一歷史事件相當於是由政府實施的大規模實驗，因此上述數字可以被視作對服兵役和收入水平之間因果關係的可信估計。