Advances in Association Rule Mining: Sequential Analysis

- Mining sequential association rules
  - Key Concept: Frequent Sequences
  - Algorithms
- More on association analysis
- Summary

Mining Sequential Patterns

- A typical application example:
  Customers typically rent "Star Wars", then "Empire Strikes Back", and then "Return of the Jedi"
- Another example:
  HSI went "Up", then "Level", then "Up", and then "Down"
- Yet another example
  Customers read "Finance News", then "Headline News", and then "Entertainment News"

Objectives of Sequential Pattern Mining:

- Given a database *D* of customer transactions, the problem of mining sequential patterns is to find all sequential patterns with user-specified minimum support

What is sequential pattern mining about?

- Based on computing the *frequent sequences*

  (*vs* computing the *frequent itemsets* in non-sequential ARM)
- Each transaction consists of customer-id, transaction-time and the items purchased in the transaction and no customer has more than one transaction with the same transaction-time.
- Sequence consists of a list of itemset in temporal order and is denoted as

  $$<s1, s2, \ldots, sn>$$

  where si is the i-th itemset (not item!) in the sequence
- An illustrating example:

  Customers bought "Smart Phone & microSD card", then "Smart Watch & Selfie Stick", and then "Balance Wheel and Drone"

Sequential Pattern Mining: Overview

Transaction DB: D

CustomerID	Transaction Time	Items
1	Jun 25 93	30
1	Jun 30 93	90
2	Jun 10 93	10,20
2	Jun 15 93	30
2	Jun 20 93	40,60,70
3	Jun 25 93	30,50,70
4	Jun 25 93	30
4	Jun 30 93	40,70
4	July 25 93	90
5	Jun 12 93	90

Sequential version of *D*

CustomerID	Customer Sequence
1	<(30),(90)>
2	<(10 20),(30),(40 60 70)>
3	<(30 50 70)>
4	<(30),(40 70),(90)>
5	<(90)>

Customer sequence : all the transactions of a customer is a sequence ordered by increasing transaction time.

For a given minimum support, e.g. 25%, we want to find the frequent sequences:
{ <(30),(90)>, <(30),(40 70)> }
Interpretation of first sequence: There exist at least 2 customers (cf. minimum support 25%) buying item 30 in a transaction and then item 90 in a later transaction .
Thus, the support for a sequence is defined as the fraction of total customers (not transactions) who support this sequence.

Sequential Association Rule Mining Steps

1. Sort Phase

Convert D into a D' of customer sequences, i.e., the database D is sorted with customer-id as the major key and transaction-time as the minor key.

2. Frequent Itemset Phase

Find the set of all frequent/large itemset L (using the Apriori algorithm).

3. Transformation Phase

Transform each customer sequence into the frequent itemset representation, i.e., $<s_1, s_2, \ldots, s_n> \Rightarrow <l_1, l_3, \ldots, l_n>$ where $l_i \in L$.

4. Sequence Phase

Find the desired sequences using the set of frequent itemsets, using

4-1. AprioriAll or

4-2. AprioriSome or (pls refer to the original paper)

4-3. DynamicSome (pls refer to the original paper)

5. Maximal Phase (optional)

Find the maximal sequences among the set of frequent sequences.

for($k = n$; $k > 1$; k --)

  foreach k-sequence sk

    delete from S all subsequences of sk.

An Example

step 1

D'

CustomerID	Customer Sequence
1	<(30),(90)>
2	<(10 20),(30),(40 60 70)>
3	<(30 50 70)>
4	<(30),(40 70),(90)>
5	<(90)>

step 2

Minsup=25%

Freq. Itemsets†	Mapped to
(30)	1
(40)	2
(70)	3
(40 70)	4
(90)	5

† support is defined against customer (not transaction)!!

step 3

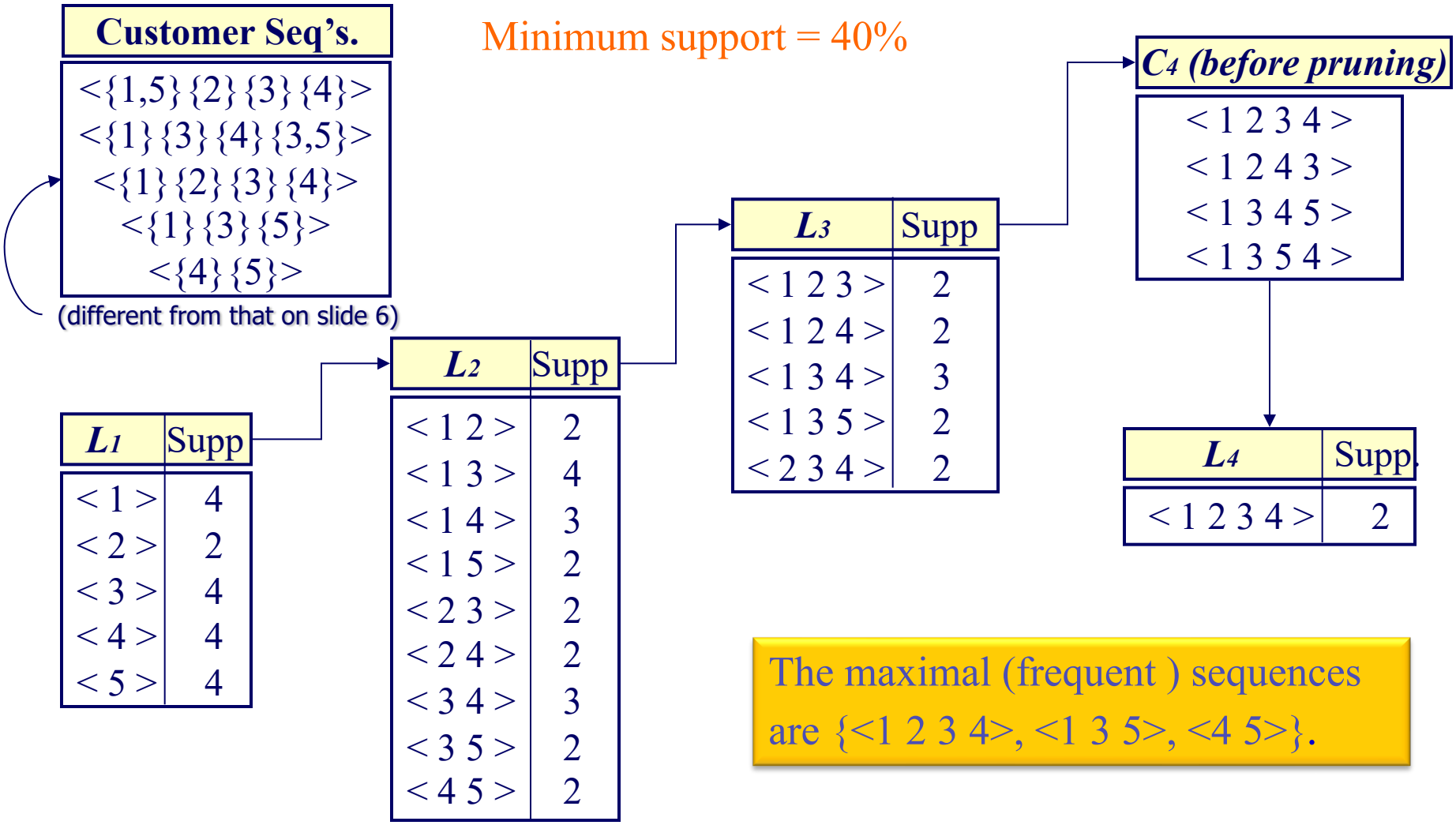
CID	Customer Sequence	Transformed Sequence	Mapping
1	<(30),(90)>	<{(30)}{(90)}>	<{1}{5}>
2	<(10 20),(30),(40 60 70)>	<{(30)}{(40),(70),(40 70)}>	<{1}{2,3,4}>
3	<(30 50 70)>	<{(30),(70)}>	<{1,3}>
4	<(30),(40 70),(90)>	<{(30)}{(40),(70),(40 70)}{(90)}>	<{1}{2,3,4}{5}>
5	<(90)>	<{(90)}>	<{5}>

AprioriAll Algorithm

step 4

```
Apriorialll()
{
  L1 = {frequent 1-itemsets};
  k = 2;              /* k represents the pass number.  */
  while (Lk-1 != ∅)
  {
     F = F ∪ Lk ;
     Ck = New candidates of size k generated from Lk-1 ;
     for each customer-sequence c ∈ D
        increment the count of all candidates in Ck that are contained in c ;
     Lk = All candidates in Ck with minimum support ;
     k++ ;
  }
  return ( F ) ;
}
```

An example for step 4: Sequence Phase



How to generate candidate sequences?

- The candidate generation steps are similar to those of non-sequential association rule mining. However, be mindful of the order!
- Step 1: self-joining L_{k-1}

  insert into $\boldsymbol{C_k}$

  select $\boldsymbol{p.itemset_1, p.itemset_2, \ldots, p.itemset_{k-1}, q.itemset_{k-1}}$

  from $\boldsymbol{L_{k-1}\ p, L_{k-1}\ q}$

  where $\boldsymbol{p.itemset_1=q.itemset_1, \ldots, p.itemset_{k-2}=q.itemset_{k-2}}$

  Step 2: pruning

  forall ***sequence c in*** $\boldsymbol{C_k}$ do

  forall ***(k-1)-subsequence s of c*** do

  if (*s is not in* L_{k-1}) **then delete** *c* **from** C_k

Example of Generating Candidates

- L_3={*abc, abd, acd, ace, bcd*}
- Self-joining: L_3*L_3
  - *abcd* and *abdc* from *abc* and *abd*
  - *acde* and *aced* from *acd* and *ace*
- Pruning:
  - *abdc* is removed because *adc/bdc* is not in L_3
  - *acde* is removed because *ade/cde* is not in L_3
  - *aced* is removed because *aed/ced* is not in L_3
- C_4={*abcd*}

Essential Definitions for Phase 5: Maximal Phase
- idea of maximal (frequent) sequence

Definition 1.

A sequence $<a_1, a_2, \ldots, a_n>$ is *contained* in another sequence $<b_1, b_2, \ldots, b_m>$ if there exist integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}, \ldots, a_n \subseteq b_{i_n}$

- E.g. <(3), (4 5), (8)> *is contained in* <(7) ,(3 8),(9),(4 5 6),(8)>? **Yes**
- E.g. <(3), (5)> *is contained in* <(3 5)>? **No**
- Consider the examples:

  Customers rent "Star Wars" **(3)**, *then "Empire Strikes Back"* **(5)**

  Customers rent "Star Wars" and "Empire Strikes Back" **(3 5)**

Definition 2.

A sequence *s* is *maximal* (maximal sequence) if *s* is not contained in any other sequence.

Forming the Sequential Association Rules

- Again, the user specified *confidence* is used by the rule generation step to qualify the strength of the sequential association rules
- The rule generation step is rather simple:
  - For each frequent sequence S, divide the sequence into two nonempty sequential parts S_f and S_l and generate the rule R: $S_f \Rightarrow S_l$.
  - If R satisfies the minimum confidence, i.e.

    $conf(S_f \Rightarrow S_l)$ = support(S)/support(S_f) $\geq min\_conf$

    then R is a strong sequential association rule and should be output.
- E.g. <1 2 3 4> will form 1$\rightarrow$ 2 3 4, 1 2 $\rightarrow$ 3 4, 1 2 3 $\rightarrow$ 4

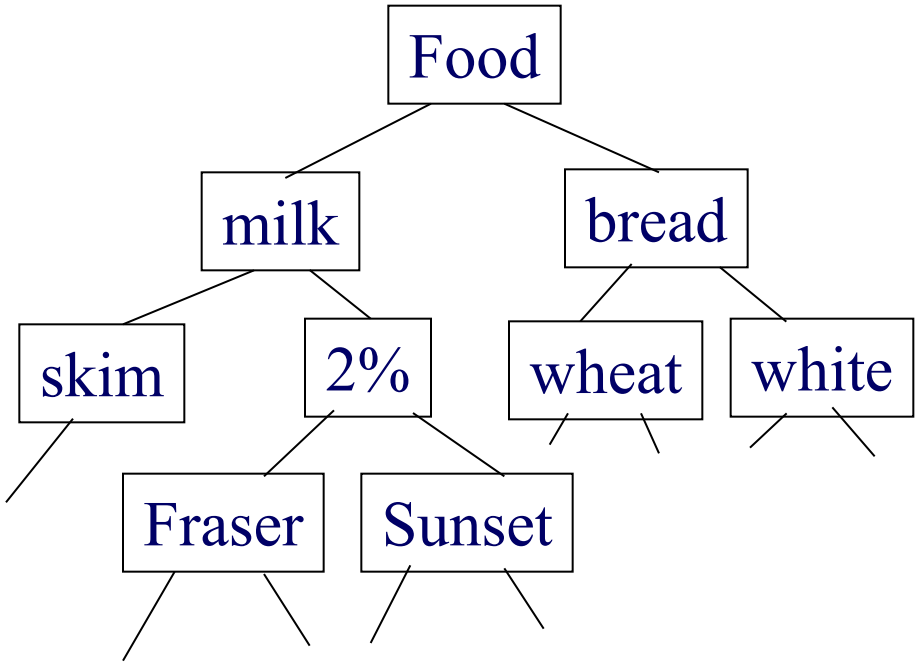
  Rules like 1 3 $\rightarrow$ 2 4, 1 2 4 $\rightarrow$ 3, etc. cannot be formed because the temporal order has been distorted!!

Advances in Association Rule Mining

- Mining sequential association rules
  - Key Concept: Frequent Sequences
  - Algorithms
- More on association analysis
- Summary

Multiple-Level/Generalized Association Rules

- Items often form hierarchy.
- Items at the lower level are expected to have lower support.
- Rules regarding itemsets at appropriate levels could be quite useful, e.g.,
  - 2% milk $\rightarrow$ wheat bread
  - 2% milk $\rightarrow$ bread
- Two methods, namely, *multilevel association rules* and *generalized association rules (GAR)* were introduced.

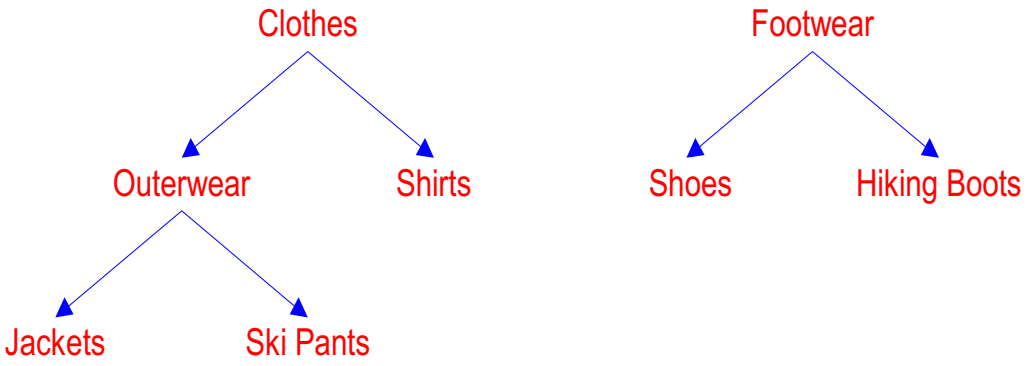


Multiple-Level/Generalized Association Rules: Redundancy Problem

- Some rules may be redundant due to "ancestor" relationships between items.
- Example
  - milk $\Rightarrow$ wheat bread [support = 8%, confidence = 70%]
  - 2% milk $\Rightarrow$ wheat bread [support = 2%, confidence = 72%]
- We say the first rule is an ancestor of the second rule. The second rule above is redundant.
- A rule is redundant if its support is close to the "expected" value, based on the rule's ancestor.

Mining Generalized Association Rules - Algorithm Basic (Agrawal 95)

- A straight forward method to mine generalized rules
- Only one additional step is needed:
  - add all the ancestors of each item in the original transactions T to T, and call it extended transaction T'
- Run any association rule mining algorithm (e.g. Apriori) on the extended transactions
- An example:



Original Transactions T

Transaction	Items bought
100	Shirt
200	Jacket, Hiking Boots
300	Ski Pants, Hiking Boots
400	Shoes
500	Shoes
600	Jacket

Algorithm Basic

- Form the extended transactions

Extended Transactions T'

Transaction	Items bought
100	Shirt, (Clothes)
200	Jacket, (Outerwear, Clothes,), Hiking Boots, (Footwear)
300	Ski Pants, (Outerwear, Clothes,) Hiking Boots, (Footwear)
400	Shoes, (Footwear)
500	Shoes, (Footwear)
600	Jacket, (Outerwear, Clothes)

- Find frequent itemsets (minsup=30%, minconf=60%)

Frequent Itemsets

Itemset	Support
{Jacket}	2
{Outerwear}	3
{Clothes}	4
{Shoes}	2
{Hiking Boots}	2
{Footwear}	4
{Outwear, Hiking Boots}	2
{Clothes, Hiking Boots}	2
{Outerwear, Footwear}	2
{Clothes, Footwear}	2

- Find the rules

Rules

Rule	Support	Conf.
{Outerwear}⇒{Hiking Boots}	33%	66%
{Outerwear}⇒{Footwear}	33%	66%
{Hiking Boots}⇒{Outerwear}	33%	100%
{Hiking Boots}⇒{Clothes}	33%	100%

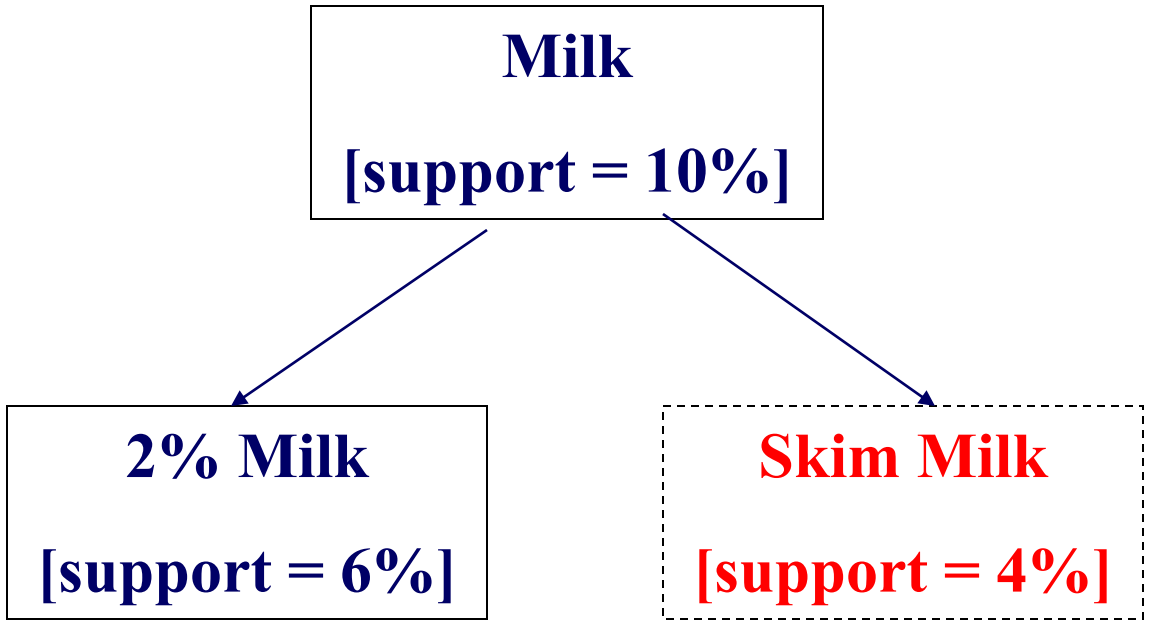
How to set minimum support? Uniform Support vs. Reduced Support

- Uniform Support: the same minimum support for all levels
  - **Pros:** One minimum support threshold. No need to examine itemsets containing any item whose ancestors do not have minimum support.
  - **Cons:** Lower level items do not occur as frequently. If support threshold
    - too high $\Rightarrow$ miss low level associations!
    - too low $\Rightarrow$ generate too many high level associations!!
- Reduced Support: reduced minimum support at lower levels
  - There are 4 search strategies:
    - Level-by-level independent
    - Level-cross filtering by k-itemset
    - Level-cross filtering by single item
    - Controlled level-cross filtering by single item

Uniform Support

Multi-level mining with uniform support

Level 1
min_sup = 5%

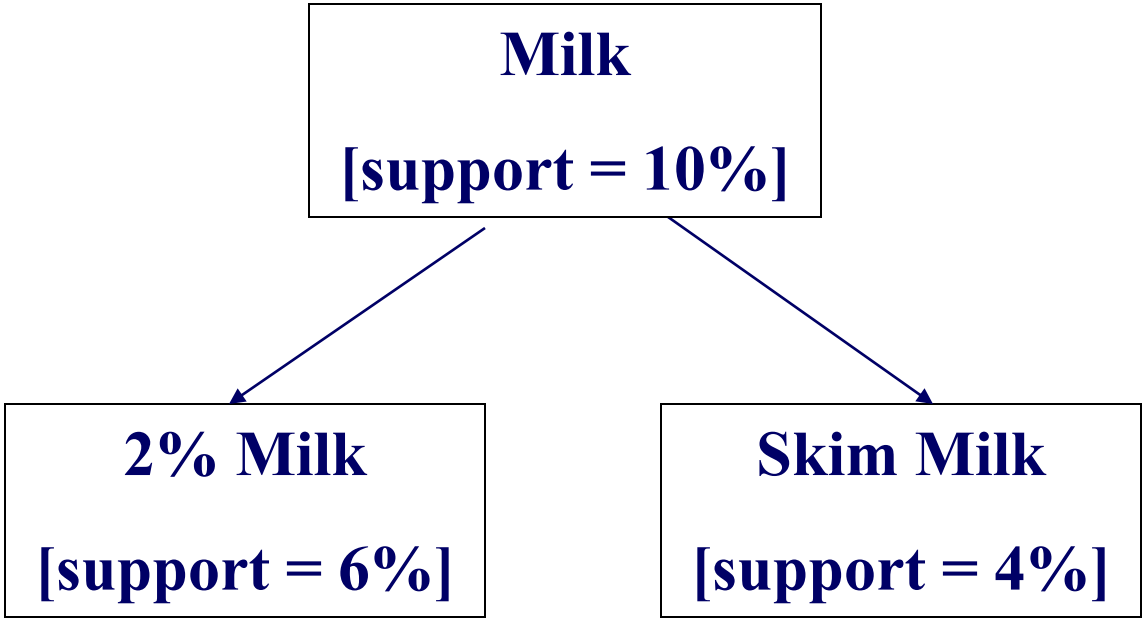


Level 2
min_sup = 5%

Reduced Support

Multi-level mining with reduced support

Level 1
min_sup = 5%



Level 2
min_sup = 3%

Multi-Dimensional Association

- Single-dimensional rules:
  buys(X, "milk") $\Rightarrow$ buys(X, "bread")
- Multi-dimensional rules: $\geq$2 dimensions or predicates
  - Inter-dimension association rules (*no repeated predicates*)
    age(X,"19-25") $\wedge$ occupation(X,"student") $\Rightarrow$ buys(X,"coke")
  - hybrid-dimension association rules (*repeated predicates*)
    age(X,"19-25") $\wedge$ buys(X, "popcorn") $\Rightarrow$ buys(X, "coke")
- Categorical Attributes
  - finite number of possible values, no ordering among values
- Quantitative Attributes
  - numeric, implicit ordering among values

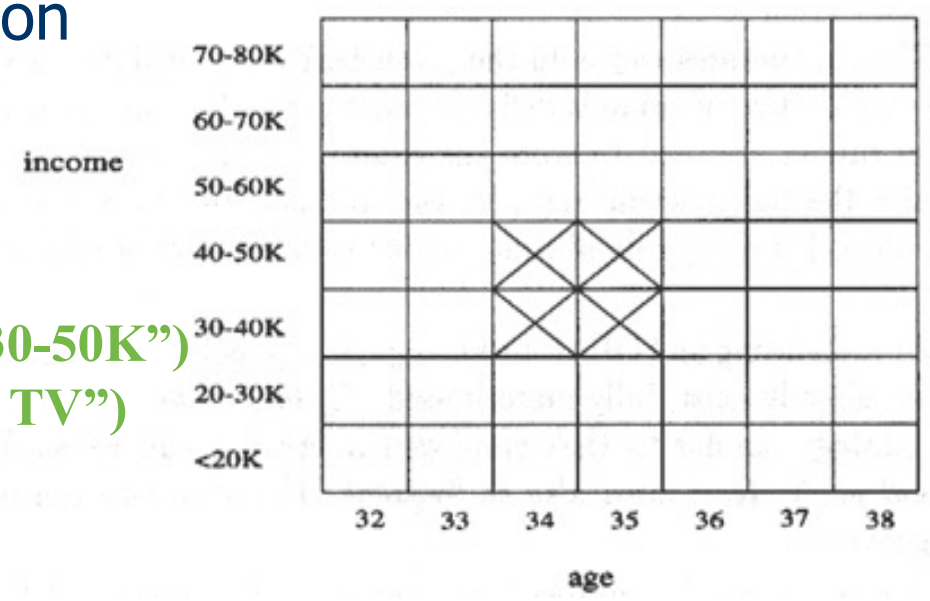
Mining Quantitative Associations

- Techniques can be categorized by how numerical attributes, such as age or salary are treated
  - Static discretization based on predefined concept hierarchies
  - Dynamic discretization based on data distribution
  - Clustering: Distance-based association
    - one dimensional clustering then association

Quantitative Association Rules

- Proposed by Lent, Swami and Widom ICDE'97
- Numeric attributes are *dynamically* discretized
  - Such that the confidence or compactness of the rules mined is maximized
- 2-D quantitative association rules: $A_{quan1} \wedge A_{quan2} \Rightarrow A_{cat}$
- Cluster *adjacent* association rules to form general rules using a 2-D grid
- Example

age(X, "34-35") ∧ income(X, "30-50K") ⇒ buys(X, "high resolution TV")



Spatial Association Analysis

- Spatial association rule: $A \Rightarrow B$ [s%, c%]
  - A and B are sets of spatial or nonspatial predicates
    - Topological relations: *intersects, overlaps, disjoint,* etc.
    - Spatial orientations: *left_of, west_of, under,* etc.
    - Distance information: *close_to, within_distance,* etc.
  - s% is the support and c% is the confidence of the rule
- Examples

is_a(x, large_town) ^ intersect(x, highway) ⇒

adjacent_to(x, water) [7%, 85%]

is_a(x, large_town) ^adjacent_to(x, georgia_strait) ⇒

close_to(x, u.s.a.) [1%, 78%]

After ALL…

What is the main contribution of association rule mining?

- Association rule mining is basically a statistical technique, basically not too much intelligence is embedded!
- However, it offers a feasible (or efficient DB) solution to very large database applications!
  - It makes good use of the limited RAM to keep track of the supposedly huge number of candidate itemsets
- Basically, the main contribution of association rule mining is that it is easy and flexible to apply.
  - It can be applied to virtually all applications (or whatever database you have)!!

How association rule mining is applied?

- What is *a transaction* in your application?
- What is *an item* in your application?
- What is *a customer* in your application? (for sequential association rule mining)

As a data scientist, you need to be creative enough for all these!!

Mining Associations in Image Data

- Basic idea
  - one image = one transaction
  - one image feature (e.g. a color) = an item
  - A sequence of fashion design images = a customer
- An example:
  - *is(Jacket color, orange) ^ is(T-shirt color, white) ⇒ is_for(fashion, teenages) [20%, 85%]*
- Yet another example (a bit different):
  - *has(Jacket color, orange) ^ has(Jacket color, white) ⇒ is(Sales volume, better) [20%, 85%]*
- Special features:
  - Need # of occurrences besides Boolean existence (as in Apriori)
  - Need spatial relationships
    - Blue on top of white squared object is associated with brown bottom

Mining the SARS Data

A tra

A transaction

of Buildings with Confirmed SARS Patients by District and Da

A transaction

District	Date of posting onto DH website																		
	12/4	13/4	14/4	15/4	16/4	17/4	18/4	19/4	20/4	21/4	22/4	23/4	24/4	25/4	26/4	27/4	28/4	29/4	30/4
HONG KONG ISLAND																			
CENTRAL AND WESTERN	1	0	0	0	1	1	2	2	2	1	1	1	0	0	0	0	0	0	0
WAN CHAI	2	1	1	0	0	0	0	1	1	2	2	2	2	2	1	0	0	1	1
EASTERN	7	1			11	9	9	8	7	7	8	8	5	3	4	4	5	5	5
SOUTHERN	0				0	1	1	1	1	1	1	1	0	0	0	0	0	0	0
KOWLOON																			
YAU TSIM MONG	7	1			6	5	4	2	2	1	1	4	5	4	3	2	2	3	3
SHAM SHUI PO	8	12			8	9	11	12	8	7	5	5	7	7	5	6	7	7	6
KOWLOON CITY	9	10	11	13	13	11	7	4	5	4	2	4	4	4	5	2	1	1	2
WONG TAI SIN	8	9	9	10	9	8	6	6	5	7	6	8	10	11	10	9	8	8	6
KWUN TONG	40	37	29	25	25	24	21	17	16	16	16	16	14						
NEW TERRITORIES WEST																			
KWAI TSING	15	15	16	17	17	15	11	15	14	13	9	12	14	13	11	10	10	12	12
TSUEN WAN	4	3	5	4	4	6	6	4	3	2	3	4	5	4	4	3	3	3	3
TUEN MUN	12	10	9	7	10	6	4	3	3	1	1	3	3	2	1	3	4	4	4
YUEN LONG	2	2	2	2	5	6	6	7	6	6	6	7	6	6	6	7	8	9	9
NEW TERRITORIES EAST																			
NORTH	2				4	9	8	8	8	8	7	7	4	3	2	3	4	3	4
TAI PO	19	1			33	31	27	22	24	21	24	21	23	28	29	25	21	15	16
SHA TIN	22		22	16	19	19	20	19	18	18	18	20	19	15	12	11	9	8	4
SAI KUNG		8	10	7	7	7	8	7	6	7	6	5	5	4	4	3	2	1	1
ISLANDS	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0

An item

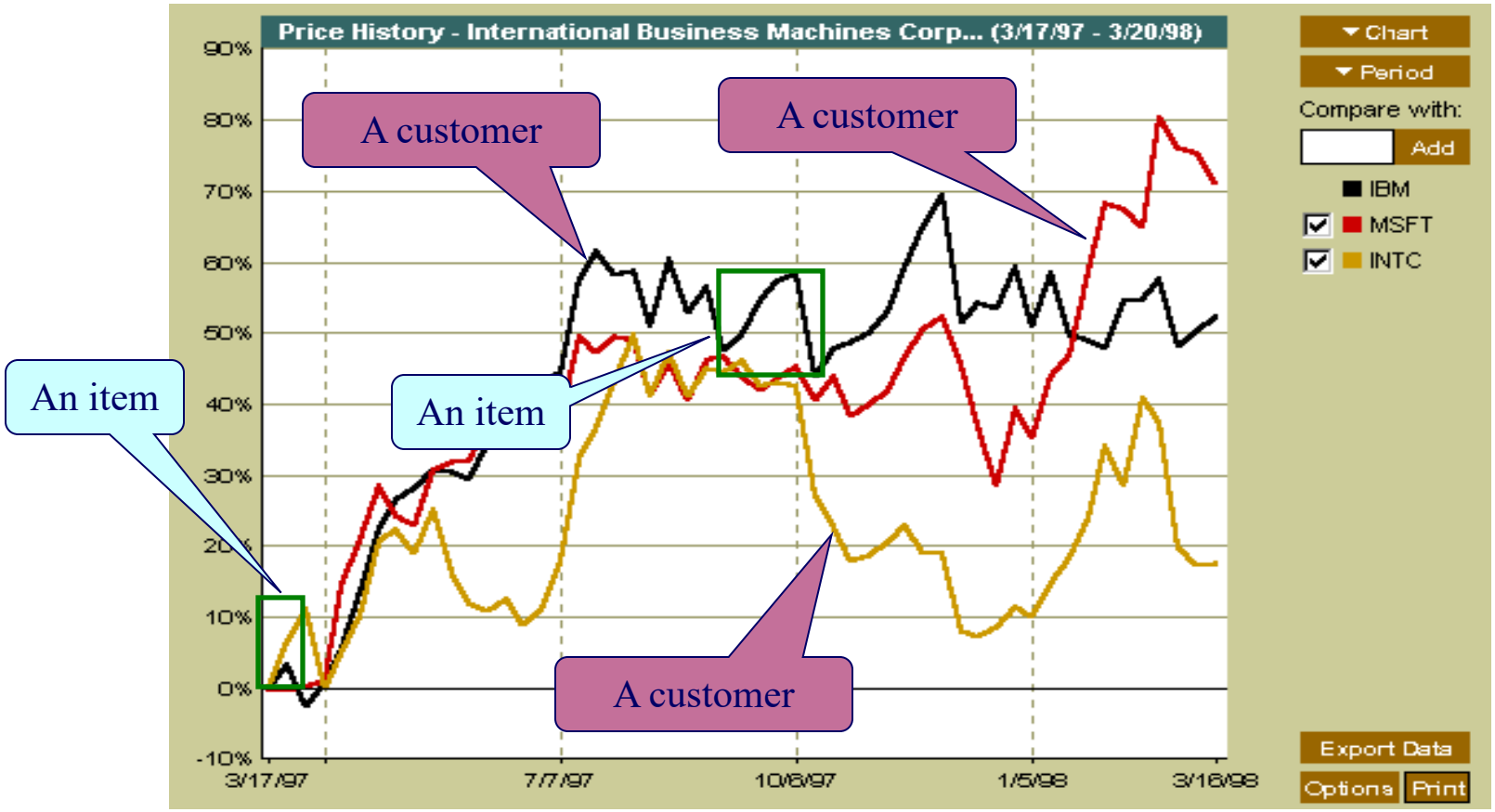
An item

A customer

An item

Mining Time-Series and Sequence Data

Time-series plot



Stock Time Series Data Mining

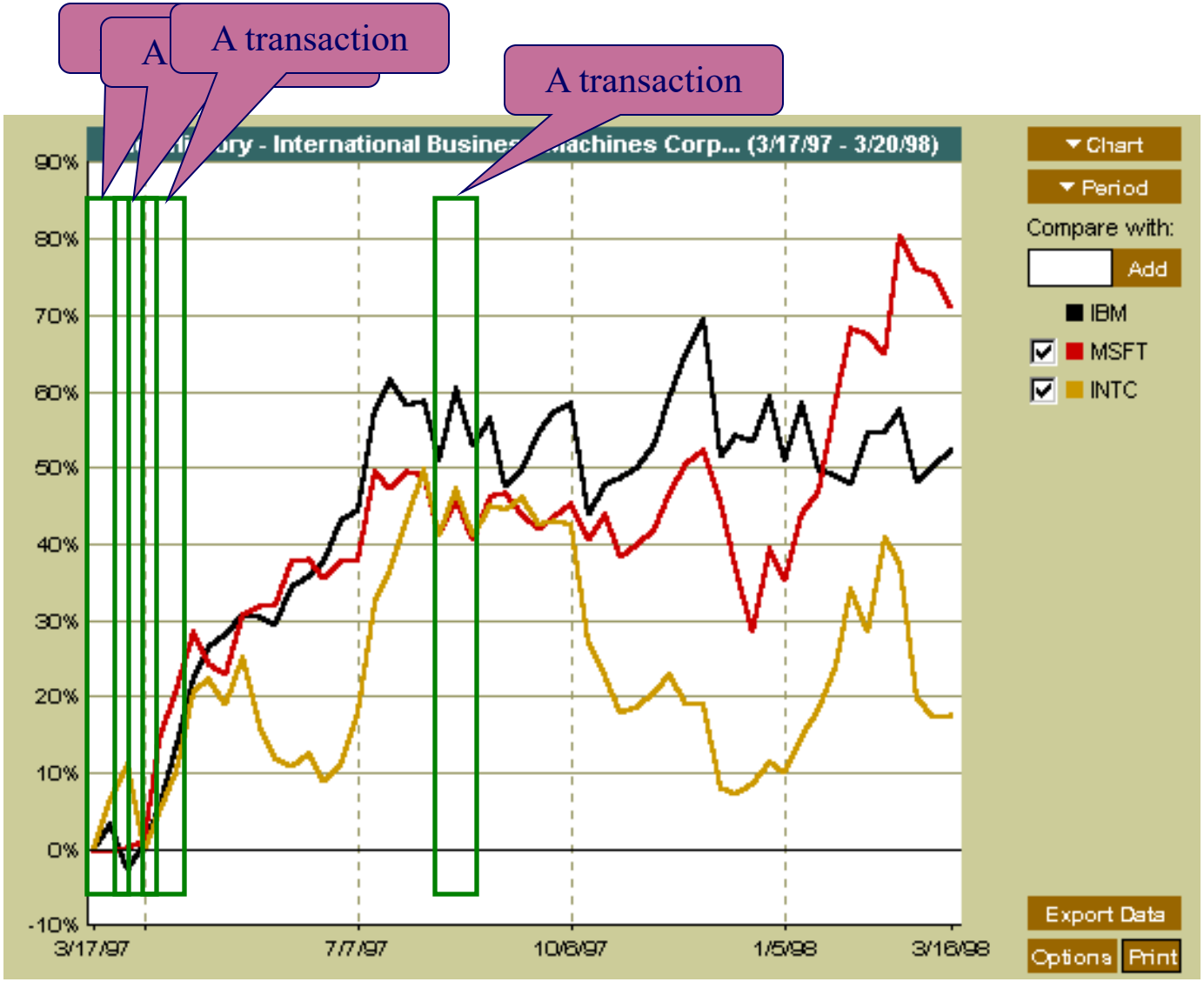
- There exist at least two types of stock data mining
  - Intra-stock data mining
  - Inter-stock data mining
- Intra-stock data mining (as elaborated)
  - Each stock is treated as a customer (sequential association analysis) and
  - Each price movement is treated as a transaction/item
  - i.e. MSFT:Go-up, Go-down, Go-up, Go-up, …, etc.
  - Hence, frequent sequence like "Go-up, Go-up, and then Go-down" can be mined.

Stock Time Series Data Mining (cont.)

- Inter-stock data mining
  - Each time window is treated as a transaction
  - Using non-sequential association rules
  - The behavior of different stocks within a time window will form the list of items
  - One can use the candlestick method to characterize the behavior, e.g. open-high-close-low (OHCL), open-low-close-low (OLCL), etc.
  - We will have:

	MSFT	IBM	INTC
TID#1:	OHCL	OLCL	OLCH
TID#2:	OLCL	OHCH	OHCL

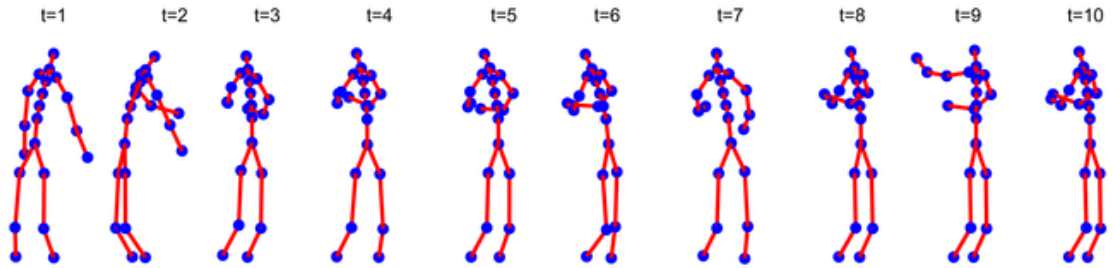
Inter-stock data mining



General time series data mining

Examples:

- Stock time series -> power consumption time series
  - Intra-building vs Inter-building TSDM
- Stock time series -> weather time series
  - Intra-location vs Inter-location TSDM
- Stock time series -> Motion sensing time series
  - Intra-object vs Inter-object TSDM



Summary

- Association rule mining
  - probably the most significant contribution from the database community in KDD
  - A large number of papers have been published
- Many interesting issues have been explored
- Interesting research directions
  - Association analysis in other types of data: spatial data, multimedia data, time series data, etc.