

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: (re: Disney+)
Customers typically watch “Avengers”, then “Avengers: Infinity War”, and then “Avengers: Endgame”
- Another example: (re: Hong Kong Stock Market)
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

$$\langle s_1, s_2, \dots, s_n \rangle$$

where s_i is the i -th itemset (**not item!**) in the sequence

- An illustrating example:

Customers bought “Smart Phone & Phone Case”, then “Smart Watch”, 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	$\langle(30),(90)\rangle$
2	$\langle(10\ 20),(30),(40\ 60\ 70)\rangle$
3	$\langle(30\ 50\ 70)\rangle$
4	$\langle(30),(40\ 70),(90)\rangle$
5	$\langle(90)\rangle$

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:
 $\{ \langle(30),(90)\rangle, \langle(30),(40\ 70)\rangle \}$

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., $\langle s_1, s_2, \dots, s_n \rangle \Rightarrow \langle l_1, l_3, \dots, l_n \rangle$ 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



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)>



Minsup=25%

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



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

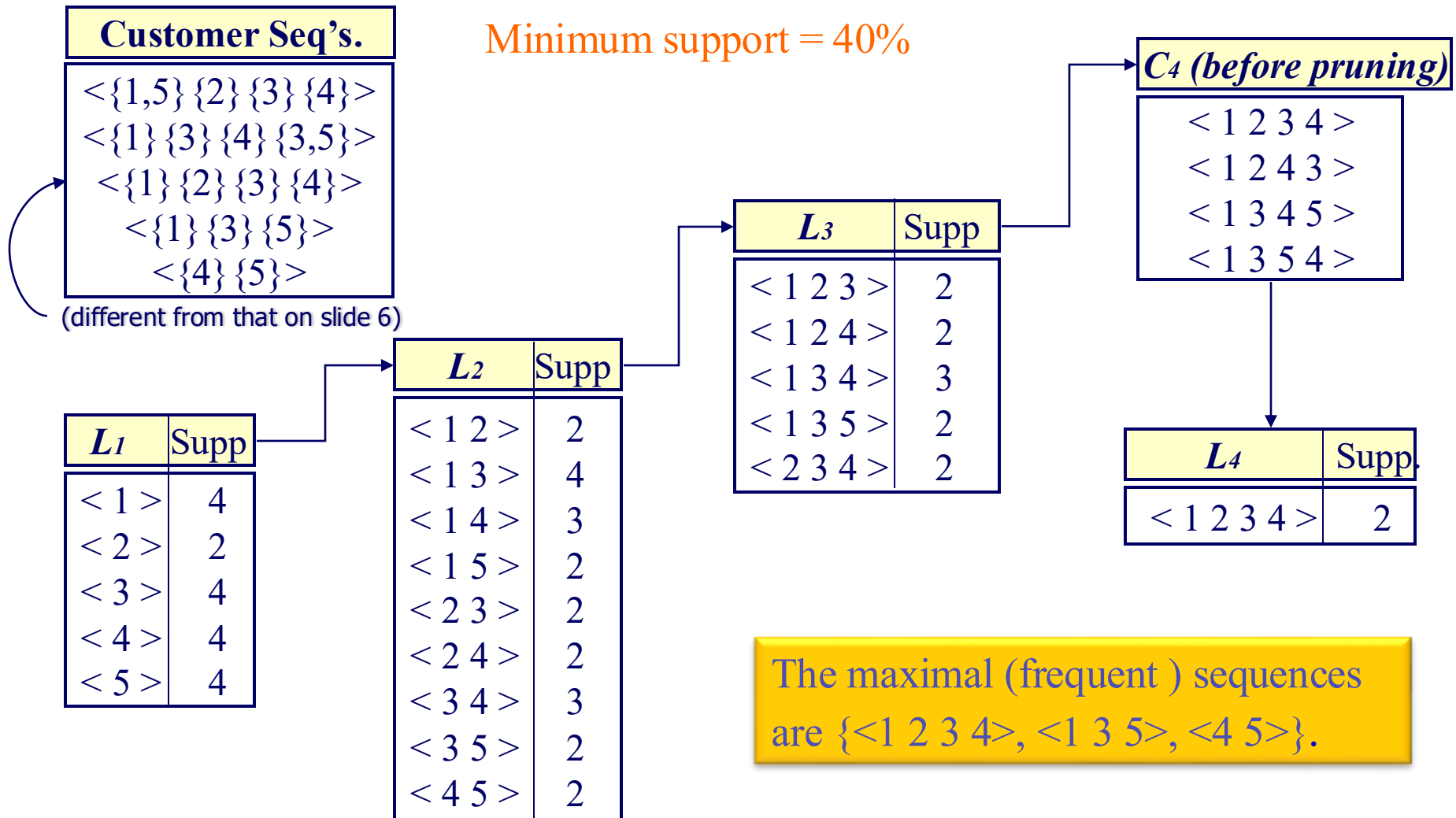
CID	Customer Sequence	Transformed Sequence	Mapping
1	<(30),(90)>	<{(30)} {(90)}>	<{1} {5}>
2	<(10 20),(30),(40 60 70)>	<{(30)} {(40),(70),(40 70)}>	<{1} {2,3,4}>
3	<(30 50 70)>	<{(30),(70)}>	<{1,3}>
4	<(30),(40 70),(90)>	<{(30)} {(40),(70),(40 70)} {(90)}>	<{1} {2,3,4} {5}>
5	<(90)>	<{(90)}>	<{5}>

AprioriAll Algorithm



```
AprioriAll()
{
   $L_1 = \{\text{frequent 1-itemsets}\};$ 
   $k = 2;$           /*  $k$  represents the pass number. */
  while ( $L_{k-1} \neq \emptyset$ )
  {
     $F = F \cup L_k;$ 
     $C_k =$  New candidates of size  $k$  generated from  $L_{k-1}$ ;
    for each customer-sequence  $c \in D$ 
      increment the count of all candidates in  $C_k$  that are contained in  $c$ ;
     $L_k =$  All candidates in  $C_k$  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 C_k

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

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

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

Step 2: pruning

forall **sequence c in 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 $\langle a_1, a_2, \dots, a_n \rangle$ is *contained* in another sequence $\langle b_1, b_2, \dots, b_m \rangle$ if there exist integers $i_1 < i_2 < \dots < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}, \dots, a_n \subseteq b_{i_n}$

- E.g. $\langle (3), (4\ 5), (8) \rangle$ is contained in $\langle (7), (3\ 8), (9), (4\ 5\ 6), (8) \rangle$?

Yes

- E.g. $\langle (3), (5) \rangle$ is contained in $\langle (3\ 5) \rangle$? **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

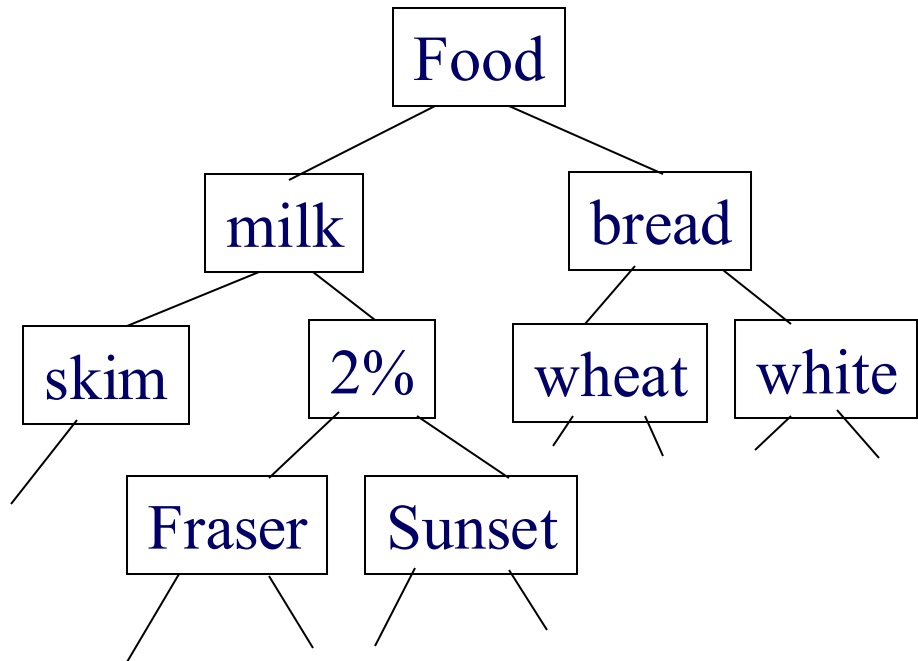
- Again, the user specified *confidence* is used by the rule generation step to qualify the strength of the **sequential association rules**
- The rule generation step is rather simple:
 - For each frequent sequence S , divide the sequence into two nonempty sequential parts S_f and S_l and generate the rule $R: S_f \Rightarrow S_l$.
 - If R satisfies the minimum confidence, i.e.
$$\text{conf}(S_f \Rightarrow S_l) = \text{support}(S) / \text{support}(S_f) \geq \text{min_conf}$$
then R is a strong sequential association rule and should be output.
- E.g. $\langle 1\ 2\ 3\ 4 \rangle$ 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 → wheat bread
 - 2% milk → bread
- Two methods, namely, *multilevel association rules* and *generalized association rules (GAR)* were introduced.

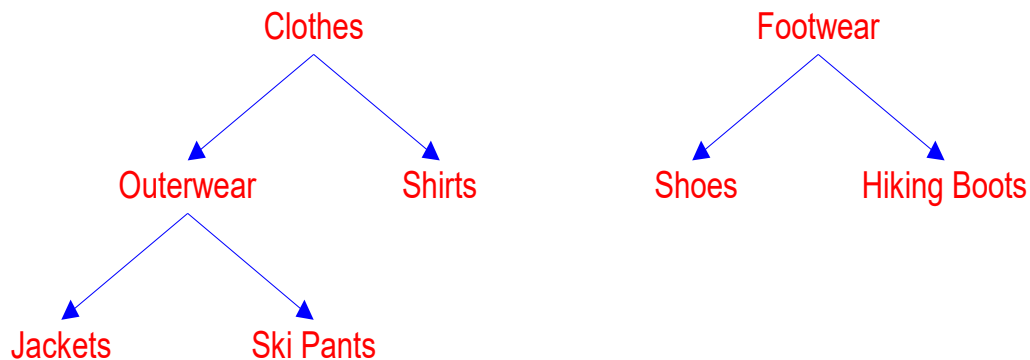


Multiple-Level/Generalized Association Rules: Redundancy Problem

- 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} \Rightarrow {Hiking Boots}	33%	66%
{Outerwear} \Rightarrow {Footwear}	33%	66%
{Hiking Boots} \Rightarrow {Outerwear}	33%	100%
{Hiking Boots} \Rightarrow {Clothes}	33%	100%

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

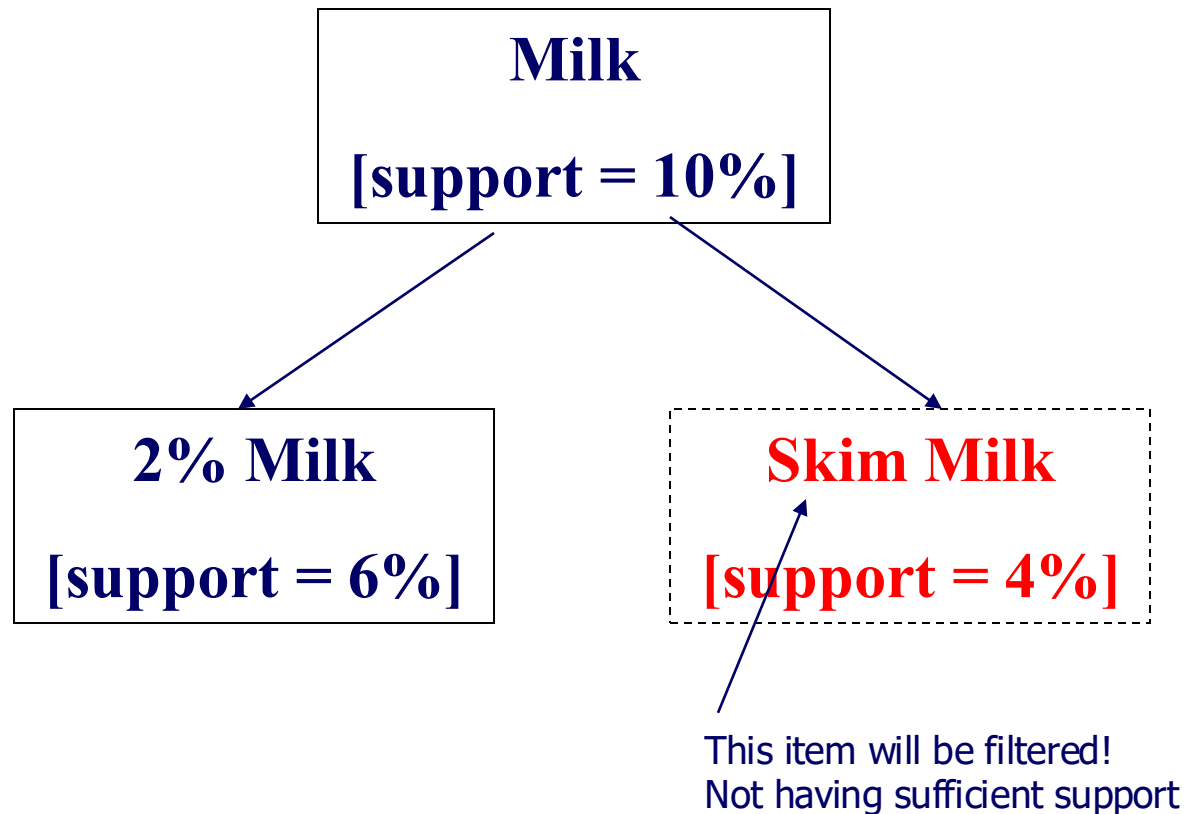
- Uniform Support: the same minimum support for all levels
 - **Pros:** One minimum support threshold. No need to examine itemsets containing any item whose ancestors do not have minimum support.
 - **Cons:** Lower level items do not occur as frequently. If support threshold is
 - too high \Rightarrow miss low level associations!
 - too low \Rightarrow generate too many high level associations!!
- Reduced Support: reduced minimum support at lower levels
 - There are many strategies to do so to fit individual application.

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:

$\text{buys}(X, \text{"milk"}) \Rightarrow \text{buys}(X, \text{"bread"})$

- Multi-dimensional rules: ≥ 2 dimensions or predicates

- Inter-dimension association rules (*no repeated predicates*)

$\text{age}(X, \text{"19-25"}) \wedge \text{occupation}(X, \text{"student"}) \Rightarrow \text{buys}(X, \text{"coke"})$

- hybrid-dimension association rules (*repeated predicates*)

$\text{age}(X, \text{"19-25"}) \wedge \text{buys}(X, \text{"popcorn"}) \Rightarrow \text{buys}(X, \text{"coke"})$

- Categorical Attributes

- finite number of possible values, no ordering among values

- Quantitative Attributes

- numeric, implicit ordering among values

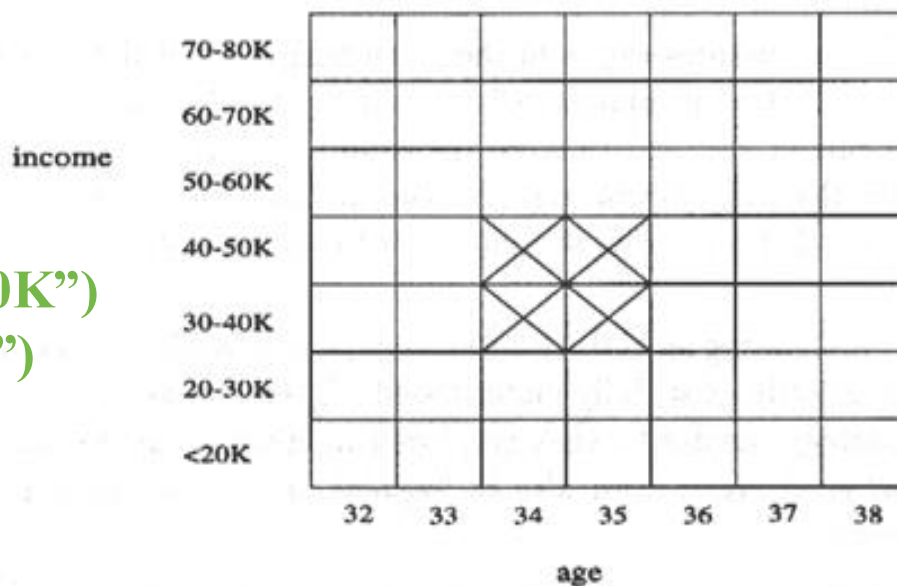
Mining Quantitative Associations

- 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

- Numeric attributes are *dynamically* discretized
 - Such that the confidence or compactness of the rules mined is maximized
- 2-D quantitative association rules: $A_{\text{quan1}} \wedge A_{\text{quan2}} \Rightarrow A_{\text{cat}}$
- Cluster *adjacent* association
 - Rules to form general rules using a 2-D grid
- Example

$\text{age}(X, \text{"34-35"}) \wedge \text{income}(X, \text{"30-50K"})$
 $\Rightarrow \text{buys}(X, \text{"high resolution TV"})$



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

$is_a(x, large_town) \wedge intersect(x, highway) \Rightarrow$
 $adjacent_to(x, water) [7\%, 85\%]$

$is_a(x, large_town) \wedge adjacent_to(x, georgia_strait) \Rightarrow$
 $close_to(x, u.s.a.) [1\%, 78\%]$

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 is association rule mining 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:

- $is(\text{Jacket color, orange}) \wedge is(\text{T-shirt color, white}) \Rightarrow is_for(\text{fashion, teenages}) [20\%, 85\%]$

■ Yet another example (a bit different):

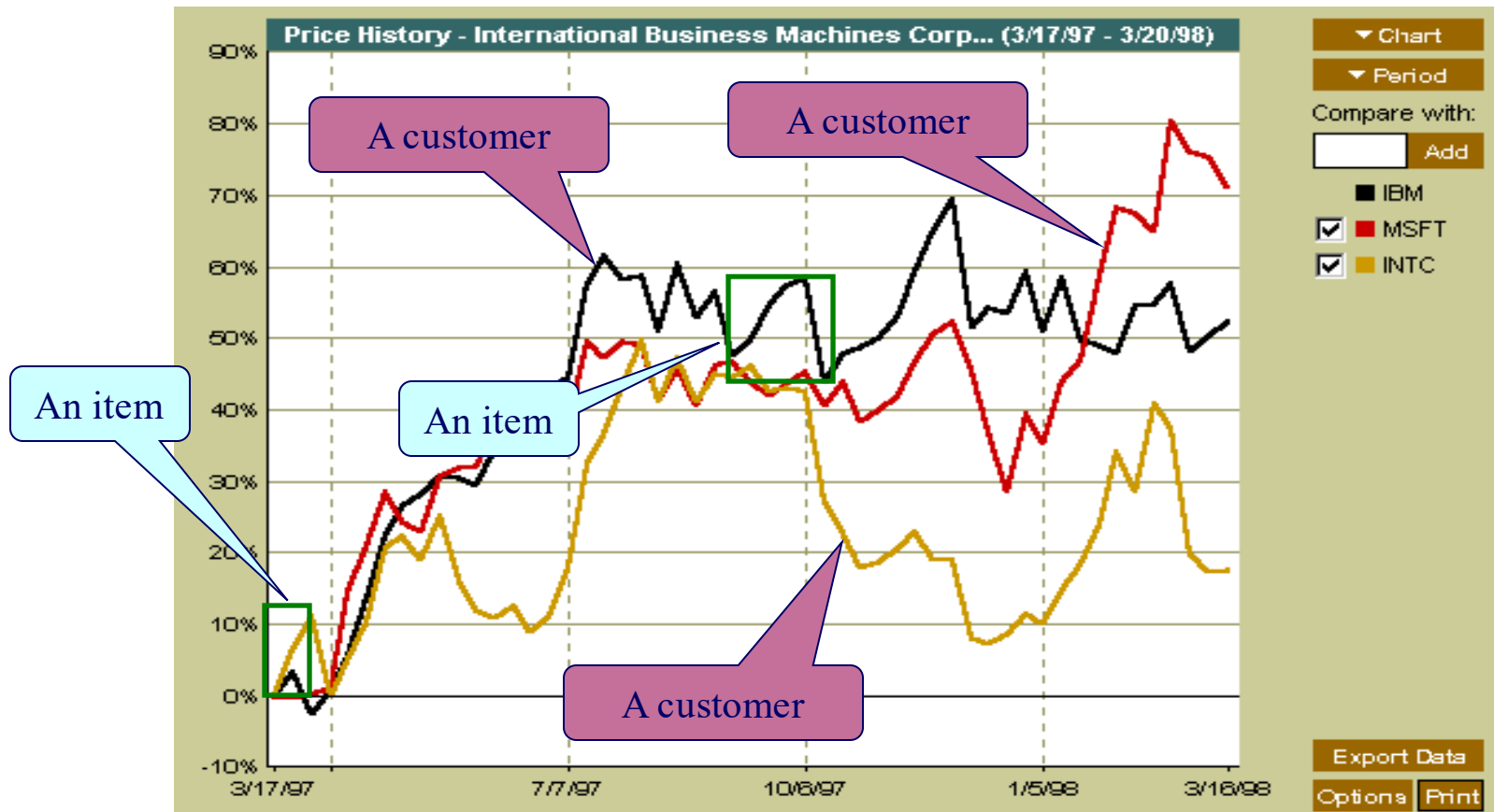
- $has(\text{Jacket color, orange}) \wedge has(\text{Jacket color, white}) \Rightarrow is(\text{Sales volume, better}) [20\%, 85\%]$

■ 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 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.

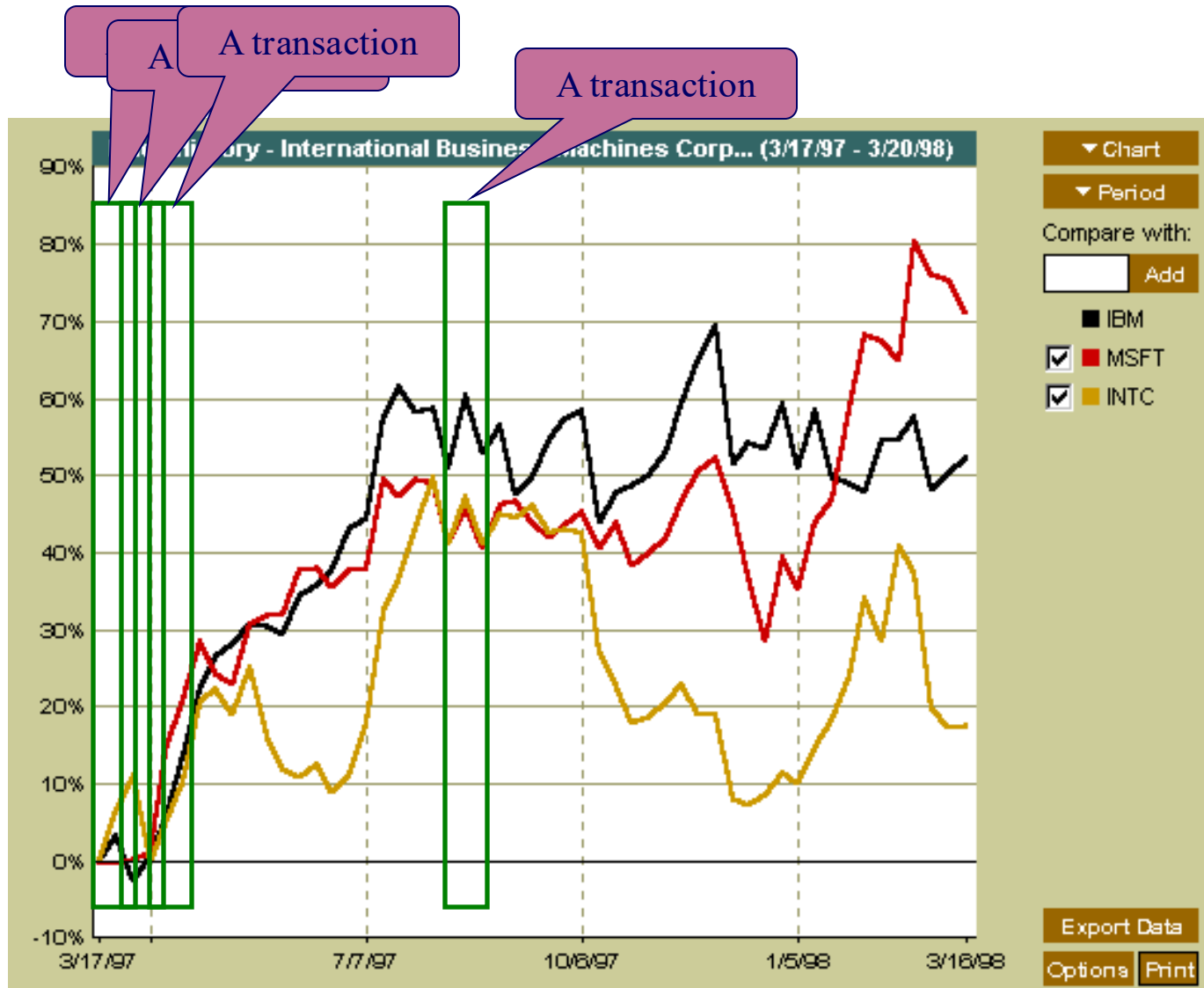
Stock Time Series Data Mining (cont.)

- 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

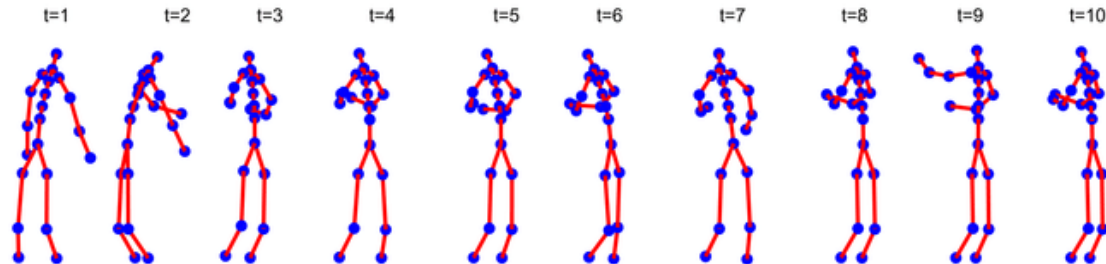
Inter-stock data mining



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.