Data Mining2 – Advanced Aspects and Applications

Fosca Giannotti and Mirco Nanni

Pisa KDD Lab, ISTI-CNR & Univ. Pisa

http://www-kdd.isti.cnr.it/

DIPARTIMENTO DI INFORMATICA - Università di Pisa

anno accademico 2013/2014

Data Mining
Association Analysis: Basic Concepts and Algorithms

Lecture Notes for Chapter 6

Introduction to Data Mining

by

Tan, Steinbach, Kumar

Association rules - module outline

- What are association rules (AR) and what are they used for:
  - The paradigmatic application: Market Basket Analysis
  - The single dimensional AR (intra-attribute)
- How to compute AR
  - Basic Apriori Algorithm and its optimizations
  - Multi-Dimension AR (inter-attribute)
  - Quantitative AR
  - Constrained AR
- How to reason on AR and how to evaluate their quality
  - Multiple-level AR
  - Interestingness
  - Correlation vs. Association

Association Rule Mining

- Given a set of transactions, find rules that will predict the occurrence of an item 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

Example of Association Rules

{Diaper} → {Beer},
{Milk, Bread} → {Eggs,Coke},
{Beer, Bread} → {Milk},

Implication means co-occurrence, not causality!

Definition: Frequent Itemset

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread,Diaper}) = 2
  - $\sigma(X) = |\{t_i | X$ contained in t_i and ti is a trasaction$\}|$
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

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

Definition: Association Rule

- **Association Rule**
  - An implication expression of the form X → Y, where X and Y are itemsets
  - Example:
    {Milk, Diaper} → {Beer}

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

- **Rule Evaluation Metrics**
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

Example:

$$\{\text{Milk}, \text{Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

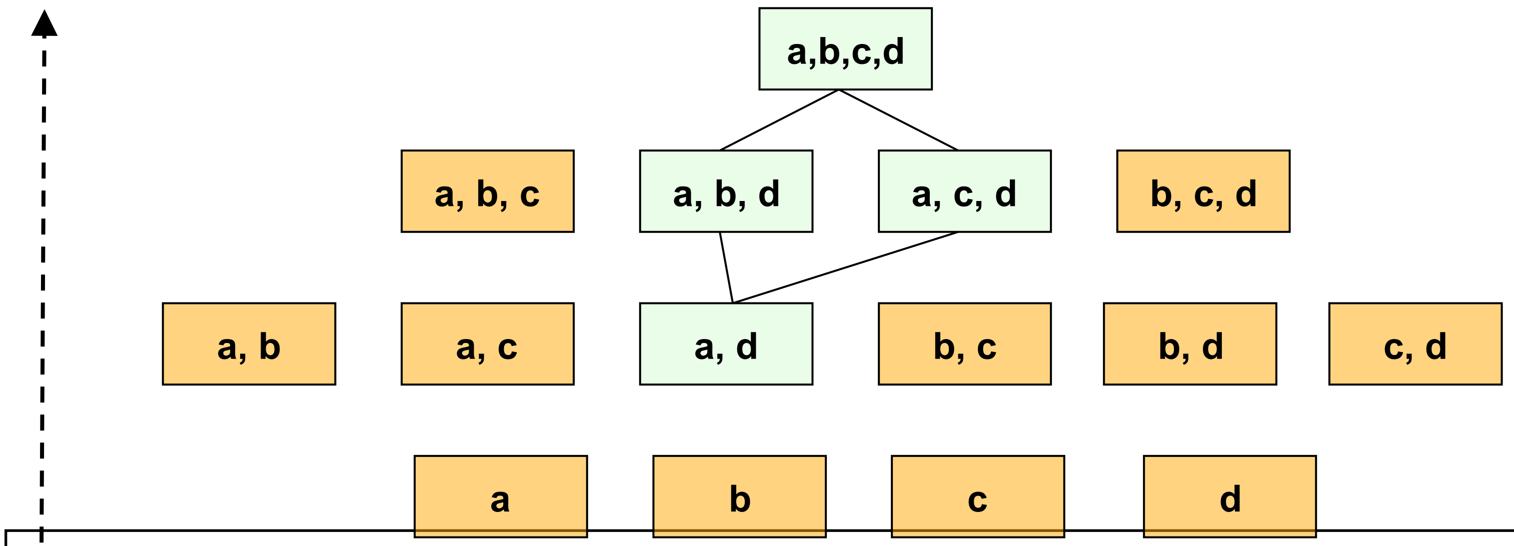
The Apriori Algorithm

- ***The classical Apriori algorithm [1994] exploits a nice property of frequency in order to prune the exponential search space of the problem:***

  "if an itemset is infrequent all its supersets will be infrequent as well"

- ***This property is known as "the antimonotonicity of frequency" (aka the "Apriori trick").***

- ***This property suggests a breadth-first level-wise computation.***



Apriori Execution Example *(min_sup = 2)*

Database TDB

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan TDB →

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

→

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan TDB ←

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

←

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan TDB →

L_3

itemset	sup
{2 3 5}	2

The Apriori Algorithm

- **Join Step**: C_k is generated by joining L_{k-1} with itself
- **Prune Step**: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
- <u>Pseudo-code</u>:

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
C_{k+1} = candidates generated from L_k;
for each transaction t in database do
increment the count of all candidates in C_{k+1} that are contained in t
L_{k+1} = candidates in C_{k+1} with min_support
end
return $\cup_k L_k$;

Generating Association Rules from Frequent Itemsets

- Only strong association rules are generated
- Frequent itemsets satisfy minimum support threshold
- Strong rules are those that satisfy minimun confidence threshold

$$\boxed{\frac{support(A \cup B)}{support(A)}}$$

```
For each frequent itemset, f, generate all non-empty subsets of
For every non-empty subset s of f do
    if support(f)/support(s) ≥ min_confidence then
        output rule s ==> (f-s)
end
```

Rule Generation

- Given a frequent itemset L, find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement
  – If {A,B,C,D} is a frequent itemset, candidate rules:

$ABC \rightarrow D$,	$ABD \rightarrow C$,	$ACD \rightarrow B$,	$BCD \rightarrow A$,
$A \rightarrow BCD$,	$B \rightarrow ACD$,	$C \rightarrow ABD$,	$D \rightarrow ABC$
$AB \rightarrow CD$,	$AC \rightarrow BD$,	$AD \rightarrow BC$,	$BC \rightarrow AD$,
$BD \rightarrow AC$,	$CD \rightarrow AB$,		

- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

Multidimensional AR

Associations between values of different attributes :

CID	nationality	age	income
1	Italian	50	low
2	French	40	high
3	French	30	high
4	Italian	50	medium
5	Italian	45	high
6	French	35	high

RULES:

nationality = French $\Rightarrow$ income = high [50%, 100%]

income = high $\Rightarrow$ nationality = French [50%, 75%]

age = 50 $\Rightarrow$ nationality = Italian [33%, 100%]

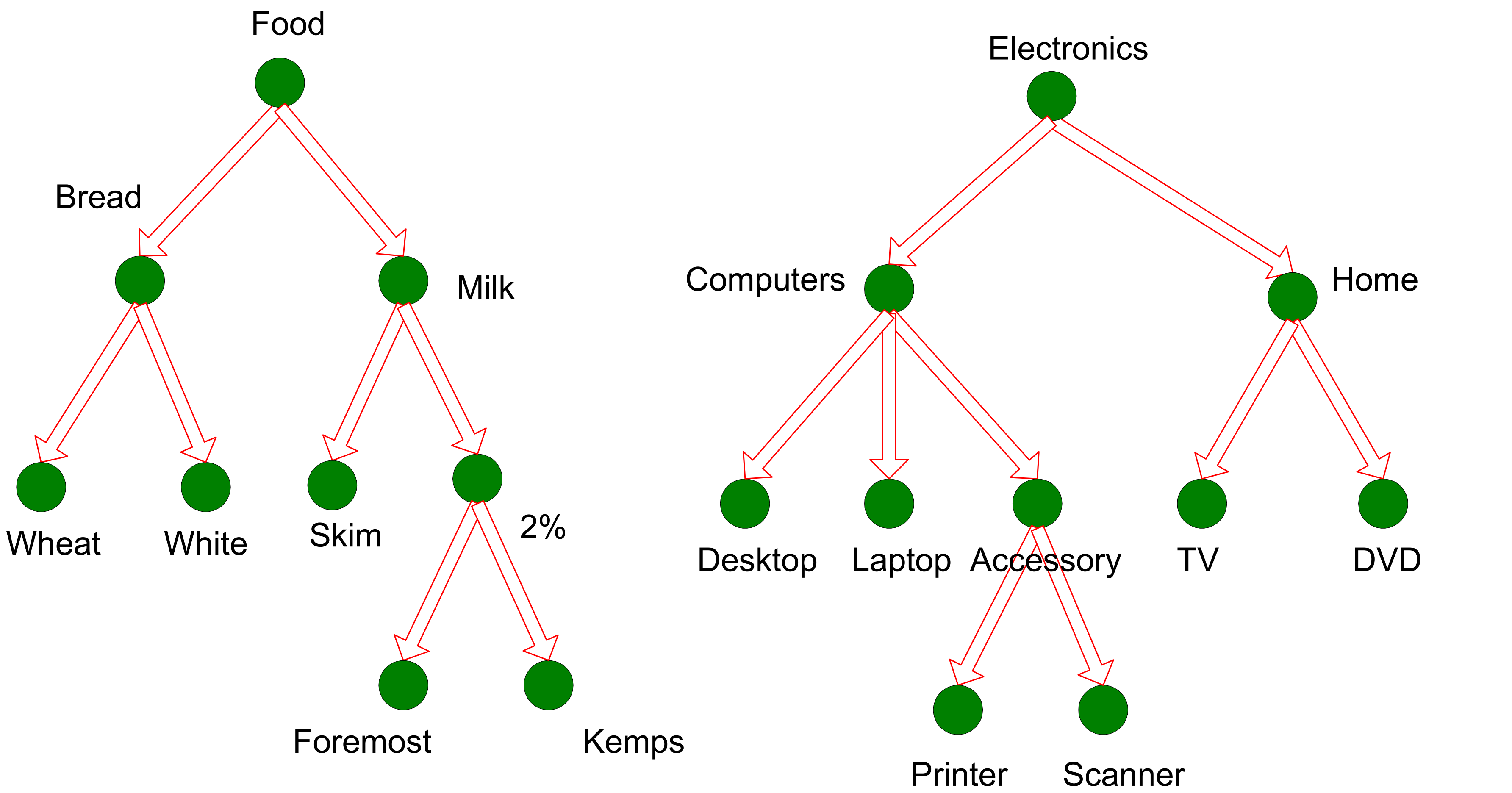
Discretization of quantitative attributes

Solution: each value is replaced by the interval to which it belongs.
height: 0-150cm, 151-170cm, 171-180cm, >180cm
weight: 0-40kg, 41-60kg, 60-80kg, >80kg
income: 0-10ML, 11-20ML, 20-25ML, 25-30ML, >30ML

CID	height	weight	income
1	151-171	60-80	>30
2	171-180	60-80	20-25
3	171-180	60-80	25-30
4	151-170	60-80	25-30

Problem: the discretization may be useless (see weight).

Multi-level Association Rules



Multilevel AR

- Is difficult to find interesting patterns at a too primitive level
  - high support = too few rules
  - low support = too many rules, most uninteresting
- Approach: reason at suitable level of abstraction
- A common form of background knowledge is that an attribute may be generalized or specialized according to a hierarchy of concepts
- Dimensions and levels can be efficiently encoded in transactions
- Multilevel Association Rules : rules which combine associations with hierarchy of concepts

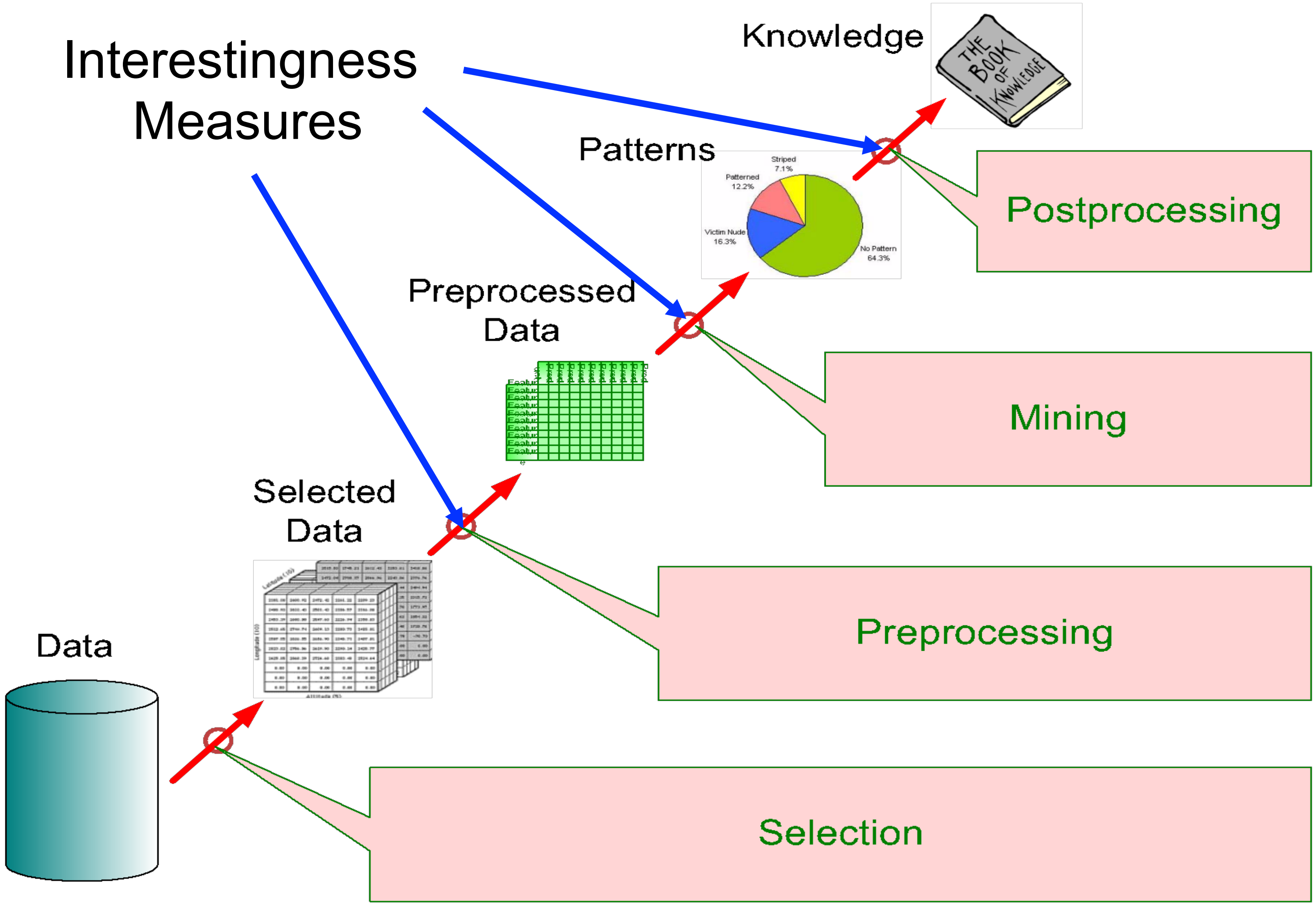
Pattern Evaluation

- Association rule algorithms tend to produce too many rules
  - many of them are uninteresting or redundant
  - Redundant if {A,B,C} → {D} and {A,B} → {D} have same support & confidence

- Interestingness measures can be used to prune/ rank the derived patterns

- In the original formulation of association rules, support & confidence are the only measures used

Application of Interestingness Measure



Computing Interestingness Measure

- Given a rule $X \rightarrow Y$, information needed to compute rule interestingness can be obtained from a contingency table

Contingency table for $X \rightarrow Y$

	Y	$\overline{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\overline{X}$	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	$\|T\|$

f_{11}: support of X and Y
f_{10}: support of X and $\overline{Y}$
f_{01}: support of $\overline{X}$ and Y
f_{00}: support of $\overline{X}$ and $\overline{Y}$

Used to define various measures

- support, confidence, lift, Gini, J-measure, etc.

Statistical-based Measures

- Measures that take into account statistical dependence

$$Lift = \frac{P(Y \mid X)}{P(Y)}$$

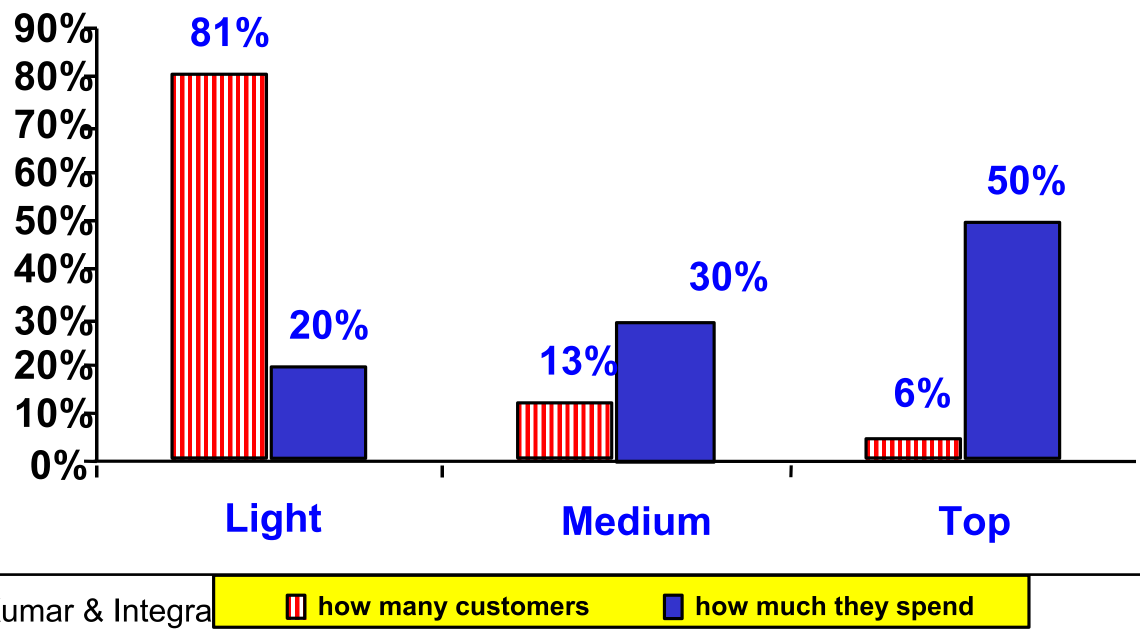
$$Interest = \frac{P(X,Y)}{P(X)P(Y)}$$

$$PS = P(X,Y) - P(X)P(Y)$$

$$\phi - coefficient = \frac{P(X,Y) - P(X)P(Y)}{\sqrt{P(X)[1 - P(X)]P(Y)[1 - P(Y)]}}$$

Conclusion (Market basket Analysis)

- MBA is a key factor of success in the competition of supermarket retailers.
- Knowledge of customers and their purchasing behavior brings potentially huge added value.



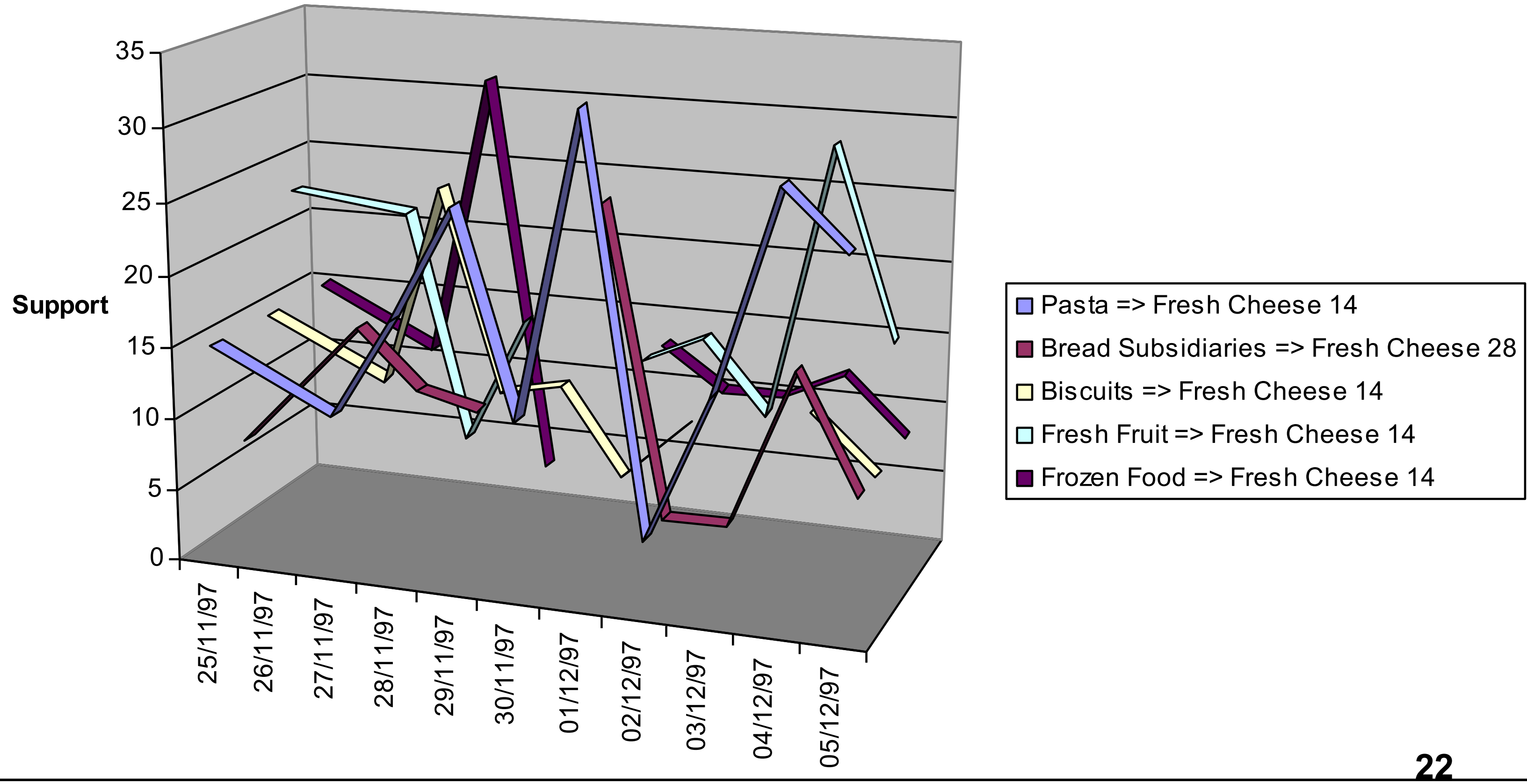
Which tools for market basket analysis?

- Association rule are needed but insufficient

- Market analysts ask for business rules:
  - Is supermarket assortment adequate for the company's target class of customers?
  - Is a promotional campaign effective in establishing a desired purchasing habit?

Business rules: temporal reasoning on AR

- Which rules are established by a promotion?
- How do rules change along time?



Sequential Pattern Mining

Sequential Pattern Mining

Lecture Notes for Chapter 7

Introduction to Data Mining

by

Tan, Steinbach, Kumar

Sequential Patterns- module outline

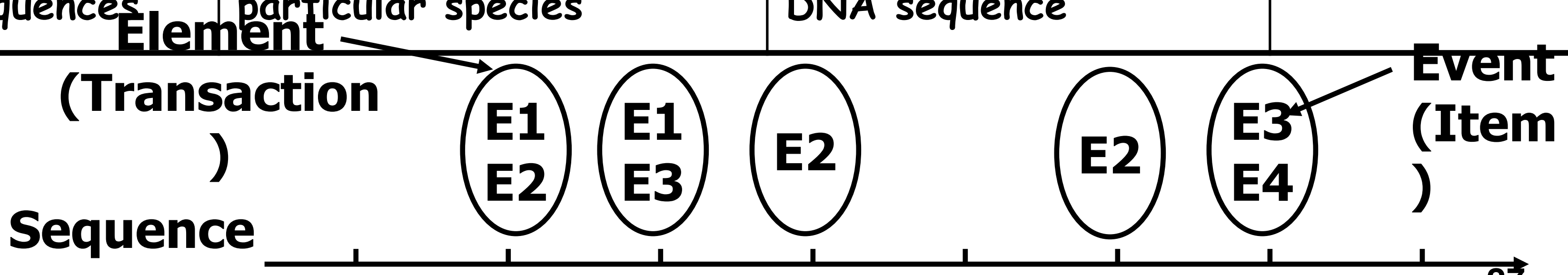
- What are Sequential Patterns(SP) and what are they used for
- From Itemset to sequences
- Formal Definiton
- Computing Sequential Patterns
- Timing Constraints

Sequential / Navigational Patterns

- Sequential patterns add an extra dimension to frequent itemsets and association rules - time.
  - Items can appear before, after, or at the same time as each other.
  - General form: "x% of the time, when A appears in a transaction, B appears within z transactions."
    - note that other items may appear between A and B, so sequential patterns do not necessarily imply consecutive appearances of items (in terms of time)
- Examples
  - Renting "Star Wars", then "Empire Strikes Back", then "Return of the Jedi" in that order
  - Collection of ordered events within an interval
  - Most sequential pattern discovery algorithms are based on extensions of the Apriori algorithm for discovering itemsets
- Navigational Patterns
  - they can be viewed as a special form of sequential patterns which capture navigational patterns among users of a site
  - in this case a session is a consecutive sequence of pageview references for a user over a specified period of time

Examples of Sequence Data

Sequence Database	Sequence	Element (Transaction)	Event (Item)
Customer	Purchase history of a given customer	A set of items bought by a customer at time t	Books, diary products, CDs, etc
Web Data	Browsing activity of a particular Web visitor	A collection of files viewed by a Web visitor after a single mouse click	Home page, index page, contact info, etc
Event data	History of events generated by a given sensor	Events triggered by a sensor at time t	Types of alarms generated by sensors
Genome sequences	DNA sequence of a particular species	An element of the DNA sequence	Bases A,T,G,C



From Itemset to sequences

- Goal: customize, personalize the offerts according the personal history of any client
- Analysis: to study the temporal buying behaviour
- " 5% of clients first has bought X, then Y then Z"
- Requirements: to keep trace of the history for the clients (nome, fidelity cards, carte di credito, bancomat, e-mail, codice fiscale)
- Domanins: vendite al dettaglio, vendite per corrispondenza, vendite su internet, vendite di prodotti finanziari/bancari, analisi mediche

Transaction with Client Identifier (Pseudo)

Intra-Transaction (Association Rules) …
Inter-Transaction (Sequential Patterns)

items $\{ i_1, \ldots, i_k \}$

Clients $\{ c_1, \ldots, c_m \}$

Transaztion $t \subseteq \{ i_1, \ldots, i_k \}$

Client trasactions

$T = \{ (c_1, date_1, t_1), \ldots, (c_n, date_n, t_n) \}$

Date may be replaced with a progressive number

CRM & SP

Conceptual Model

Cliente	Data	Trans
3	10/09/1999	{10}
2	10/09/1999	{10, 20}
5	12/09/1999	{90}
2	15/09/1999	{30}
2	20/09/1999	{40,60,70}
1	25/09/1999	{30}
3	25/09/1999	{30,50,70}
4	25/09/1999	{30}
4	30/09/1999	{40,70}
1	30/09/1999	{90}
4	25/10/1999	{90}

Logic Model

Data	Cliente	Articolo
10/09/1999	3	10
10/09/1999	2	10
10/09/1999	2	20
12/09/1999	5	90
15/09/1999	2	30
20/09/1999	2	40
20/09/1999	2	60
20/09/1999	2	70
25/09/1999	1	30
25/09/1999	3	30
25/09/1999	3	30
25/09/1999	3	70
25/09/1999	4	30
30/09/1999	4	40
30/09/1999	4	70
30/09/1999	1	90
25/10/1999	4	90

Sequence data from MB

Insieme di transazioni cliente

$T = \{ (data_1, c_1, t_1), \ldots, (data_n, c_n, t_n) \}$

Sequenza di transazioni per cliente c

$$seq(c) = \langle t_1, \ldots, t_i \ldots, t_n \rangle$$

ordinate per data

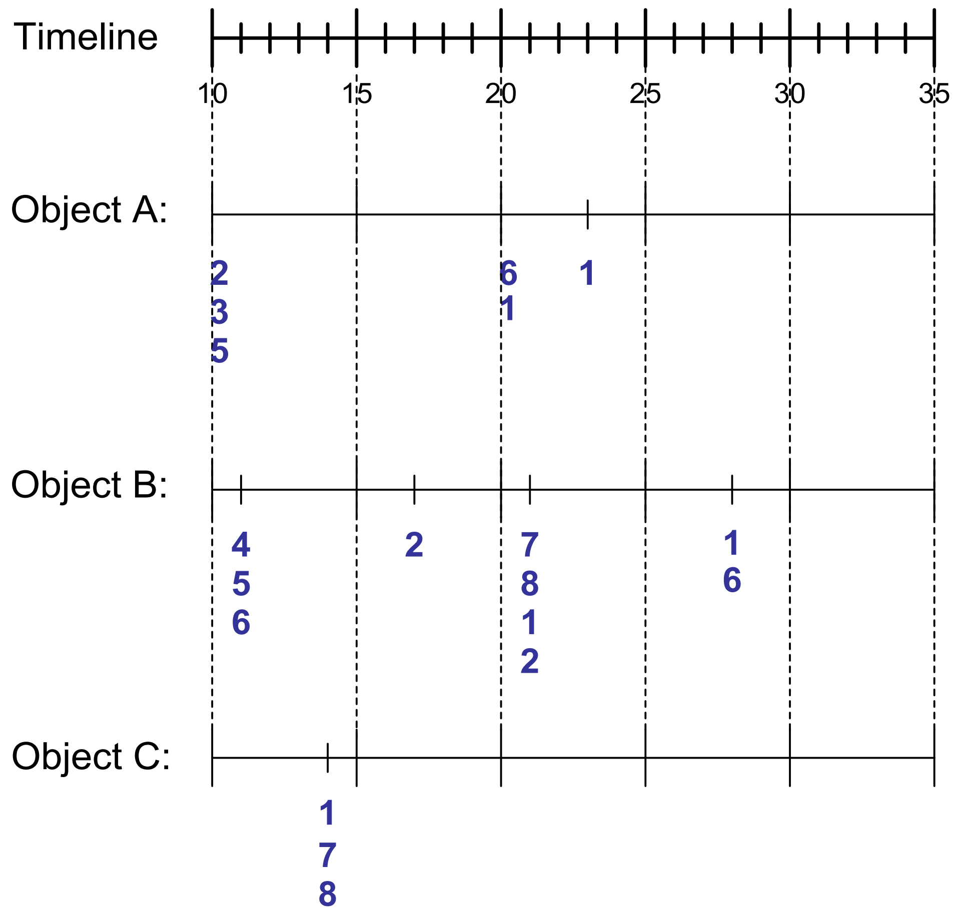
Cliente	Sequenza
1	< {30},{90} >
2	< {10, 20}, {30}, {40,60,70}>
3	<{10}, {30,50,70}>
4	< {30}, {40,70}, {90} >
5	<{90}>

Libro	Titolo
10	Star Wars Episode I
20	La fondazione e l'impero
30	La seconda fondazione
40	Database systems
50	Algoritmi + Strutture Dati =
60	L'insostenibile leggerezza
70	Immortalita'
90	I buchi neri

Sequence Data

Sequence Database:

Object	Timestamp	Events
A	10	2, 3, 5
A	20	6, 1
A	23	1
B	11	4, 5, 6
B	17	2
B	21	7, 8, 1, 2
B	28	1, 6
C	14	1, 8, 7



Sequences & Supports (intuition)

$\langle I_1, I_2, \ldots, I_n \rangle$ is contained in $\langle J_1, J_2, \ldots, J_m \rangle$
If there exist $h_1 < \ldots < h_n$ such that

$$I_1 \subseteq J_{h1}, \ldots, I_n \subseteq J_{hn}$$

< {30}, {90} > is contained in < {30}, {40,70}, {90} >
< {30}, {40,70} > is contained in < {10,20}, {30}, {40,50,60,70} >
and in < {30}, {40,70}, {90} >

$$\text{Support}(s) = \frac{|\{ c \mid s \text{ contained in } seq(c) \}|}{\text{number of clients}}$$

Support(< {20}, {70} >) = 40%
Supporto(< {90} >) = 60%

Formal Definition of a Sequence

- A sequence is an ordered list of elements (transactions)

$$s = \langle e_1 e_2 e_3 \dots \rangle$$

  - Each element contains a collection of events (items)

$$e_i = \{i_1, i_2, \dots, i_k\}$$

  - Each element is attributed to a specific time or location
- Length of a sequence, $|s|$, is given by the number of elements of the sequence
- A k-sequence is a sequence that contains k events (items)

Examples of Sequence

- Web sequence:

  < {Homepage} {Electronics} {Digital Cameras} {Canon Digital Camera} {Shopping Cart} {Order Confirmation} {Return to Shopping} >

- Sequence of initiating events causing the nuclear accident at 3-mile Island:
  (http://stellar-one.com/nuclear/staff_reports/summary_SOE_the_initiating_event.htm)

  < {clogged resin} {outlet valve closure} {loss of feedwater} {condenser polisher outlet valve shut} {booster pumps trip} {main waterpump trips} {main turbine trips} {reactor pressure increases}>

- Sequence of books checked out at a library:

  <{Fellowship of the Ring} {The Two Towers} {Return of the King}>

Formal Definition of a Subsequence

- A sequence $\langle a_1\, a_2 \dots a_n \rangle$ is contained in another sequence $\langle b_1\, b_2 \dots b_m \rangle$ $(m \geq n)$ if there exist integers $i_1 < i_2 < \dots < i_n$ such that $a_1 \subseteq b_{i1}$, $a_2 \subseteq b_{i1}$, …, $a_n \subseteq b_{in}$

Data sequence	Subsequence	Contain?
< {2,4} {3,5,6} {8} >	< {2} {3,5} >	Yes
< {1,2} {3,4} >	< {1} {2} >	No
< {2,4} {2,4} {2,5} >	< {2} {4} >	Yes

- The support of a subsequence w is defined as the fraction of data sequences that contain w
- A *sequential pattern* is a frequent subsequence (i.e., a subsequence whose support is ≥ *minsup*)

Sequential Pattern Mining: Definition

- Given:
  - a database of sequences
  - a user-specified minimum support threshold, *minsup*
- Task:
  - Find all subsequences with support ≥ *minsup*

Sequential Pattern Mining: Challenge

- Given a sequence: <{a b} {c d e} {f} {g h i}>
  - Examples of subsequences:

    <{a} {c d} {f} {g} >, < {c d e} >, < {b} {g} >, etc.

- How many k-subsequences can be extracted from a given n-sequence?

<{a b} {c d e} {f} {g h i}> n = 9

k=4: y _ _ y y _ _ _ y

<{a} {d e} {i}>

Answer :

$$\binom{n}{k} = \binom{9}{4} = 126$$

Sequential Pattern Mining: Example

Object	Timestamp	Events
A	1	1,2,4
A	2	2,3
A	3	5
B	1	1,2
B	2	2,3,4
C	1	1, 2
C	2	2,3,4
C	3	2,4,5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

***Minsup* = 50%**

Examples of Frequent Subsequences:

< {1,2} > s=60%
< {2,3} > s=60%
< {2,4}> s=80%
< {3} {5}> s=80%
< {1} {2} > s=80%
< {2} {2} > s=60%
< {1} {2,3} > s=60%
< {2} {2,3} > s=60%
< {1,2} {2,3} > s=60%

Extracting Sequential Patterns

- Given n events: $i_1, i_2, i_3, \ldots, i_n$
- Candidate 1-subsequences:

  $\langle\{i_1\}\rangle, \langle\{i_2\}\rangle, \langle\{i_3\}\rangle, \ldots, \langle\{i_n\}\rangle$
- Candidate 2-subsequences:

  $\langle\{i_1, i_2\}\rangle, \langle\{i_1, i_3\}\rangle, \ldots, \langle\{i_1\} \{i_1\}\rangle, \langle\{i_1\} \{i_2\}\rangle, \ldots, \langle\{i_{n-1}\} \{i_n\}\rangle$
- Candidate 3-subsequences:

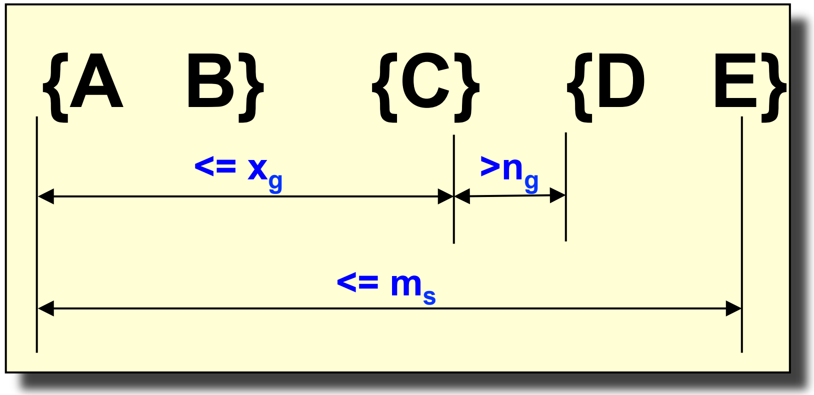
  $\langle\{i_1, i_2, i_3\}\rangle, \langle\{i_1, i_2, i_4\}\rangle, \ldots, \langle\{i_1, i_2\} \{i_1\}\rangle, \langle\{i_1, i_2\} \{i_2\}\rangle, \ldots,$
  $\langle\{i_1\} \{i_1, i_2\}\rangle, \langle\{i_1\} \{i_1, i_3\}\rangle, \ldots, \langle\{i_1\} \{i_1\} \{i_1\}\rangle, \langle\{i_1\} \{i_1\} \{i_2\}\rangle, \ldots$

Generalized Sequential Pattern (GSP)

- Step 1:
  - Make the first pass over the sequence database D to yield all the 1-element frequent sequences
- Step 2:

  Repeat until no new frequent sequences are found
  - Candidate Generation:
    - Merge pairs of frequent subsequences found in the (k-1)*th* pass to generate candidate sequences that contain k items
  - Candidate Pruning:
    - Prune candidate *k*-sequences that contain infrequent (*k-1)*-subsequences
  - Support Counting:
    - Make a new pass over the sequence database D to find the support for these candidate sequences
  - Candidate Elimination:
    - Eliminate candidate *k*-sequences whose actual support is less than *minsup*

Timing Constraints (I)



x_g: max-gap

n_g: min-gap

m_s: maximum span

$x_g = 2$, $n_g = 0$, $m_s = 4$

Data sequence	Subsequence	Contain?
< {2,4} {3,5,6} {4,7} {4,5} {8} >	< {6} {5} >	Yes
< {1} {2} {3} {4} {5}>	< {1} {4} >	No
< {1} {2,3} {3,4} {4,5}>	< {2} {3} {5} >	Yes
< {1,2} {3} {2,3} {3,4} {2,4} {4,5}>	< {1,2} {5} >	No

Time constraints (2)

- Sliding Windows (transazione contenuta in più transazioni)

$\langle I_1, I_2, \ldots, I_n \rangle$ è contenuta in $\langle J_1, J_2, \ldots, J_m \rangle$

se esistono $h_1 < u_1 < \ldots < h_n < u_n$ per cui

$$I_1 \subseteq \cup_{k = h1..u1} J_k, \ldots, I_n \subseteq \cup_{k = hn..un} J_k$$

$$\text{transaction-time}(J_{ui}) - \text{transaction-time}(J_{hi}) < \text{window-size} \text{ per } i = 1..n$$

< {30}, {40,70} > è contenuta in < {30}, {40}, {70} >

se transaction-time({70}) - transaction-time({40}) < window-size

- Time Constraints (limite di tempo tra due transazioni)

$\langle I_1, I_2, \ldots, I_n \rangle$ è contenuta in $\langle J_1, J_2, \ldots, J_m \rangle$

se esistono $h_1 < \ldots < h_n$ per cui

$$I_1 \subseteq J_{h1}, \ldots, I_n \subseteq J_{hn}$$

$$\text{mingap} < \text{transaction-time}(J_{hi}) - \text{transaction-time}(J_{hi-1}) < \text{maxgap}$$

Sequences & Supports

$\langle I_1, I_2, \ldots, I_n \rangle$ is contained in $\langle J_1, J_2, \ldots, J_m \rangle$
If there exist $h_1 < \ldots < h_n$ such that

$$I_1 \subseteq J_{h1}, \ldots, I_n \subseteq J_{hn}$$

< {30}, {90} > is contained in < {30}, {40,70}, {90} >
< {30}, {40,70} > is contained in < {10,20}, {30}, {40,50,60,70} >
and in < {30}, {40,70}, {90} >

$$\text{Support}(s) = \frac{|\{ c \mid s \text{ contained in } seq(c) \}|}{\text{number of clients}}$$

Support(< {20}, {70} >) = 40%
Supporto(< {90} >) = 60%

Sequential Patterns

Given MinSupport and a set of sequences

S = { s | Support(s) >= MinSupport }

A sequence in S is a *Sequential Pattern if is not contained in any other sequence of S*

MinSupport = 40%

< {30}, {90} > is a sequantial pattern

Supporto< {30} >) = 80% is not a sequantial pattern as it is contained in < {30}, {90} >

MinSupporto = 50%

< {30}, {90} > non è in S

< {30} > è un pattern sequenziale

Altre Generalizzazioni

- Sliding Windows (transazione contenuta in più transazioni)

$\langle I_1, I_2, \ldots, I_n \rangle$ è contenuta in $\langle J_1, J_2, \ldots, J_m \rangle$

se esistono $h_1 < u_1 < \ldots < h_n < u_n$ per cui

$$I_1 \subseteq \cup_{k = h1..u1} J_k , \ldots, I_n \subseteq \cup_{k = hn..un} J_k$$

transaction-time(J_{ui}) - transaction-time(J_{hi}) < window-size per i = 1..n

< {30}, {40,70} > è contenuta in < {30}, {40}, {70} >

se transaction-time({70}) - transaction-time({40}) < window-size

- Time Constraints (limite di tempo tra due transazioni)

$\langle I_1, I_2, \ldots, I_n \rangle$ è contenuta in $\langle J_1, J_2, \ldots, J_m \rangle$

se esistono $h_1 < \ldots < h_n$ per cui

$$I_1 \subseteq J_{h1} , \ldots, I_n \subseteq J_{hn}$$

mingap < transaction-time(J_{hi}) - transaction-time(J_{hi-1}) < maxgap per i = 2..n

Sequential Pattern Mining:
Cases and Parameters

- Duration of a time sequence T
  - Sequential pattern mining can then be confined to the data within a specified duration
  - Ex. Subsequence corresponding to the year of 1999
  - Ex. Partitioned sequences, such as every year, or every week after stock crashes, or every two weeks before and after a volcano eruption

- Event folding window w
  - If $w = T$, time-insensitive frequent patterns are found
  - If $w = 0$ (no event sequence folding), sequential patterns are found where each event occurs at a distinct time instant
  - If $0 < w < T$, sequences occurring within the same period w are folded in the analysis

Sequential Pattern Mining:
Cases and Parameters

- Time interval, int, between events in the discovered pattern
  - int = 0: no interval gap is allowed, i.e., only strictly consecutive sequences are found
    - Ex. "Find frequent patterns occurring in consecutive weeks"
  - min_int ≤ int ≤ max_int: find patterns that are separated by at least min_int but at most max_int
    - Ex. "If a person rents movie A, it is likely she will rent movie B within 30 days" (int ≤ 30)
  - int = c ≠ 0: find patterns carrying an exact interval
    - Ex. "Every time when Dow Jones drops more than 5%, what will happen exactly two days later?" (int = 2)

Aspetti Computazionali

- Mail Order: Clothes
  - 16.000 items
  - 2.900.000 transazioni
  - 214.000 clienti
  - 10 anni
  - Algoritmo GSP (Shrikant e Agrawal) su IBM RS/6000 250