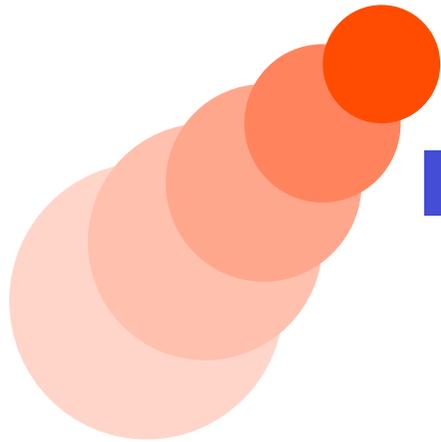


Data Warehousing, Data Mining & Business Intelligence



Regole Associative

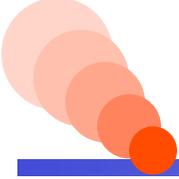
Paolo G. Franciosa

Dipartimento di Statistica, Probabilità e Statistiche Applicate

Università "La Sapienza"

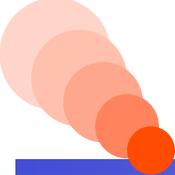
paolo.franciosa@uniroma1.it

Questo materiale deriva dalla traduzione e adattamento delle presentazioni pubblicate dal prof. Jiawei Han



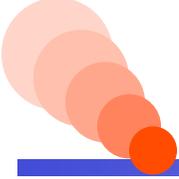
Scoperta di regole associative

- **Regole associative**
- Regole associative Booleane uni-dimensionali da database transazionali
- Regole associative multilivello da database transazionali
- Regole associative multidimensionali
- Dalle regole associative all'analisi di correlazione



Cosa sono le regole associative?

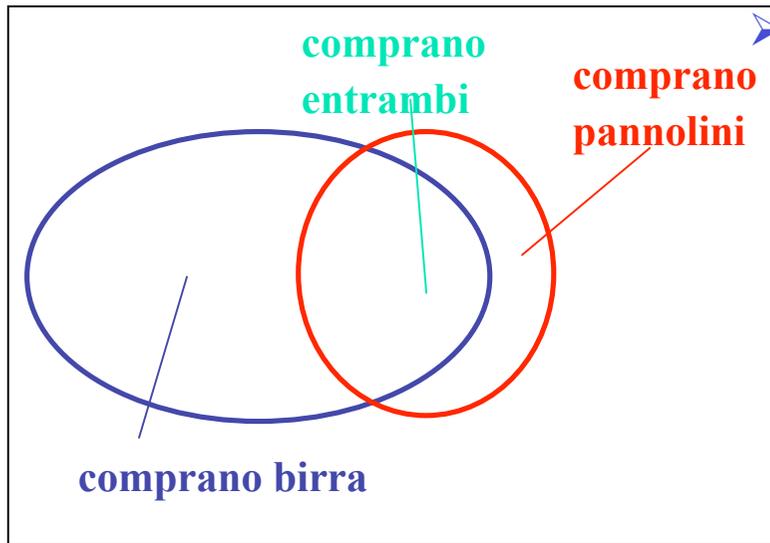
- Scoperta di regole associative:
 - trovare pattern frequenti sotto forma di relazioni causa-effetto
- Applicazioni:
 - market basket analysis, cross-marketing, progetto di cataloghi, loss-leader analysis, clustering, classificazione, etc.
- Formato regole: "Coda → Testa [supporto, confidenza]".
 - acquista(x, "pannolini") → acquista(x, "birra") [0.5%, 60%]
 - spec (x, "CS") & segue(x, "DB") → voto(x, "28..30") [1%, 75%]



Concetti di base

- Dati: (1) database di transazioni, (2) ogni transazione è un elenco di elementi (acquistati da un cliente)
- Trovare: tutte le regole che legano la presenza di un insieme di elementi alla presenza di un altro insieme
 - p.es., *il 98% delle persone che acquistano pneumatici e accessori per auto richiede anche i servizi*
- Applicazioni
 - * ⇒ *Servizi* (cosa dovremmo offrire per incrementare la richiesta di servizi?)
 - *Elettronica di consumo* ⇒ * (se vendiamo molta elettronica, **quali altri articoli** possiamo offrire?)
 - **Combinazioni di offerte** nella vendita diretta
 - Scoprire relazioni causa effetto per la diagnosi medica

Misure: supporto e confidenza



Trovare le regole $X \Rightarrow Y$ dati minima confidenza e supporto

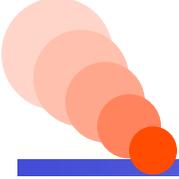
- **supporto, s** , probabilità che una transazione contenga $X \cup Y$
- **confidenza, c** , probabilità condizionata che una transazione contenente X contenga anche y

Transazione	Articoli acq.
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

supporto minimo 50%

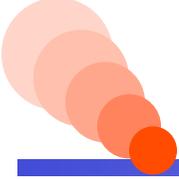
confidenza minima 50%:

- $A \Rightarrow C$ (50%, 66.6%)
- $C \Rightarrow A$ (50%, 100%)



Caratteristiche delle regole associative

- Booleane vs. quantitative (tipo di valori)
 - $\text{acquista}(x, \text{"SQLServer"}) \ \& \ \text{acquista}(x, \text{"DMBook"}) \rightarrow \text{acquista}(x, \text{"DBMiner"})$ [0.2%, 60%]
 - $\text{età}(x, \text{"30..39"}) \ \& \ \text{reddito}(x, \text{"42k..48k"}) \rightarrow \text{acquista}(x, \text{"PC"})$ [1%, 75%]
- Dimensione singola o multipla
 - cardinalità degli insiemi testa e coda
- Livello di dettaglio
 - $\text{acquista}(x, \text{"superalcolici"})$ oppure $\text{acquista}(x, \text{"brandy"})$?
- Estensioni
 - correlazione, analisi di causalità
 - non sempre implicate dalle regole associative
 - pattern massimali e insiemi chiusi



Scoperta di regole associative

- Regole associative
- Regole associative Booleane uni-dimensionali da database transazionali
- Regole associative multilivello da database transazionali
- Regole associative multidimensionali
- Dalle regole associative all'analisi di correlazione

Ricerca di regole associative

2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. supporto 50%
Min. confidenza 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

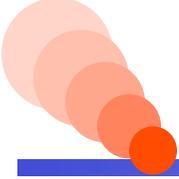
Regola $A \Rightarrow C$:

supporto = $\text{supporto}(\{A \cup C\}) = 50\%$

confidenza = $\text{supporto}(\{A \cup C\}) / \text{supporto}(\{A\}) = 66.6\%$

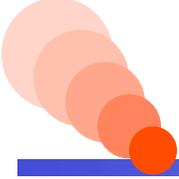
Algoritmo **Apriori**:

Ogni sottoinsieme di un insieme frequente
deve essere frequente



Ricerca di insiemi frequenti: Apriori

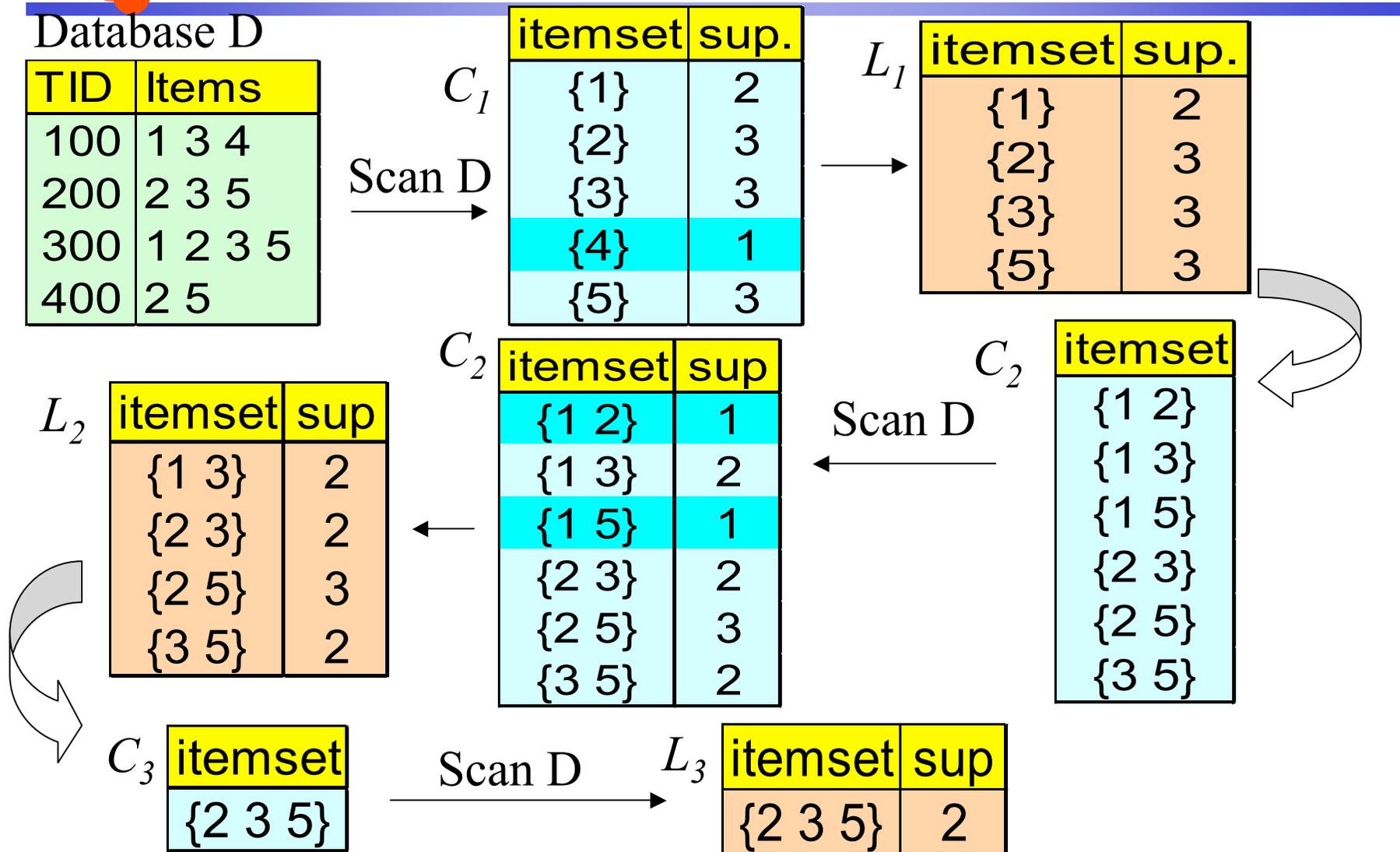
- *Insiemi frequenti*: insiemi che hanno frequenza superiore alla soglia di supporto minimo
 - ogni sottoinsieme di un insieme frequente deve essere frequente
 - se $\{AB\}$ è un insieme frequente, sia $\{A\}$ che $\{B\}$ devono essere frequenti
 - trova iterativamente gli insiemi frequenti di cardinalità da 1 a k (k -insiemi)
- Usa gli insiemi frequenti per generare regole associative

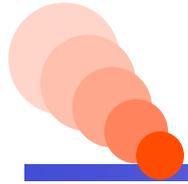


L'algoritmo Apriori

- **Join Step:** C_k è generato unendo L_{k-1} con sè stesso
- **Prune Step:** i (k-1)-insiemi non frequenti non possono essere sottoinsiemi di insiemi frequenti
- Pseudo-codice:
 - C_k : insiemi candidati di taglia k
 - L_k : insiemi frequenti di taglia k
 - $L_1 = \{\text{elementi frequenti}\};$
 - for** ($k = 1; L_k \neq \emptyset; k++$) **do**
 - C_{k+1} = candidati generati da L_k ;
 - for each** transazione t in database **do**
 - incrementa la frequenza dei candidati in C_{k+1} contenuti in t
 - L_{k+1} = candidati in C_{k+1} che raggiungono il supporto minimo
 - return** $\bigcup_k L_k$;

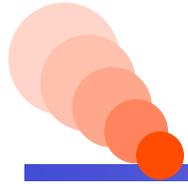
Algoritmo Apriori — Esempio





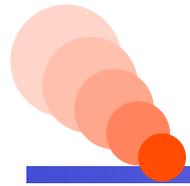
Generazione dei candidati

- Step 1: self-joining L_{k-1}
 - forall **insiemi X, Y in L_{k-1} con $k-2$ elementi comuni**
 - do **inserisci la loro unione in C_k**
- Step 2: pruning
 - forall **insiemi c in C_k do**
 - forall **$(k-1)$ -sottoinsiemi s di c do**
 - if **$(s$ is not in $L_{k-1})$ then delete c from C_k**



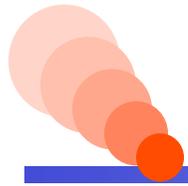
Esempio di generazione dei candidati

- $L_3 = \{abc, abd, acd, ace, bcd\}$
- Self-joining: $L_3 * L_3$
 - $abcd$ da abc e abd
 - $abce$ da abc e ace
 - $acde$ da acd e ace
- Pruning:
 - $abce$ può essere eliminato poiché abe non è in L_3
 - $acde$ può essere eliminato poiché ade non è in L_3
- $C_4 = \{abcd\}$



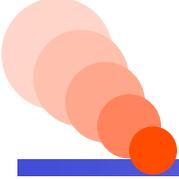
Migliorare l'efficienza di Apriori

- **Rimozione di transazioni:** una transazione che non contiene k -insiemi frequenti non sarà utile nelle scansioni successive
- **Partizionamento:** partiziono l'insieme di transazioni T in T_1, T_2, \dots, T_m : un insieme potenzialmente frequente in T deve essere frequente in almeno un T_i
- **Campionamento:** mining su un sottoinsieme di transazioni abbassando la soglia di supporto. Poi verifico se il supporto totale supera la soglia



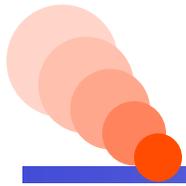
Apriori Performance Bottlenecks

- Il nucleo dell'algoritmo Apriori:
 - usare $(k - 1)$ -insiemi frequenti per generare k -insiemi frequenti **candidati**
 - scan del DB per conteggiare il supporto dei candidati
- Bottleneck: **generazione dei candidati**
 - numero enorme di candidati:
 - 10^4 (1)-insiemi frequenti generano $5 \cdot 10^7$ (2)-insiemi candidati
 - per trovare un 100-insieme frequente devo generare $2^{100} \approx 10^{30}$ candidati.
 - Scansioni ripetute:
 - per trovare gli (n) -insiemi servono n scansioni



Un algoritmo più efficiente

- Non generare l'insieme dei candidati
- Comprimere le transazioni in una struttura gerarchica: Frequent-Pattern tree (FP-tree), ordinata in base alla frequenza degli elementi
- Mining sul FP-tree



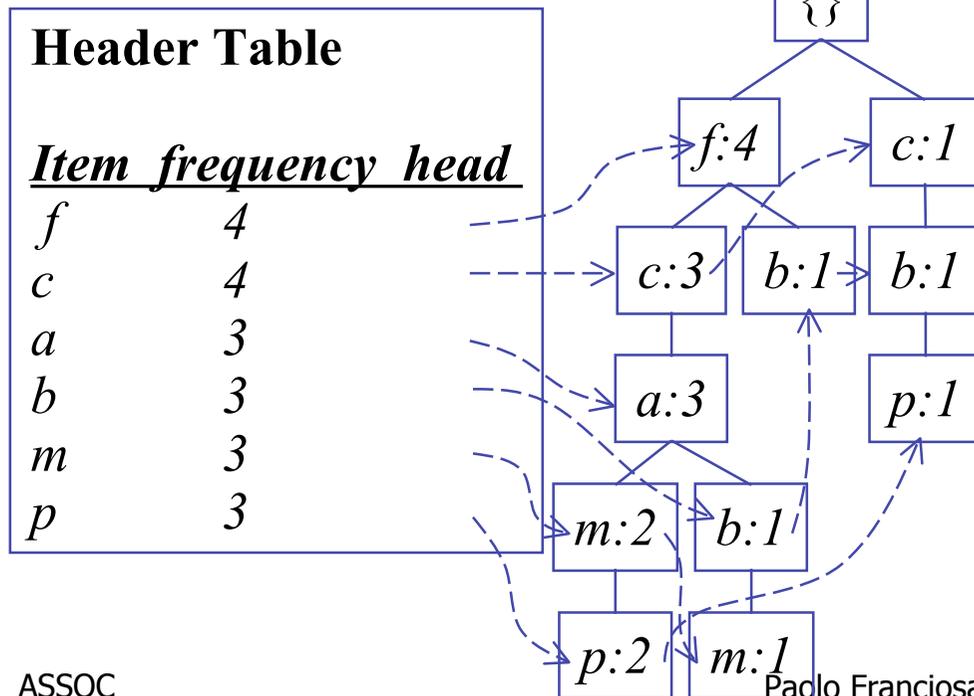
Esempio di FP-tree

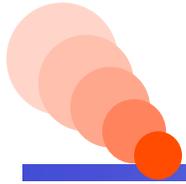
<u>TID</u>	<u>Items bought</u>	<u>(ordered) frequent items</u>
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

$min_support = 0.5$

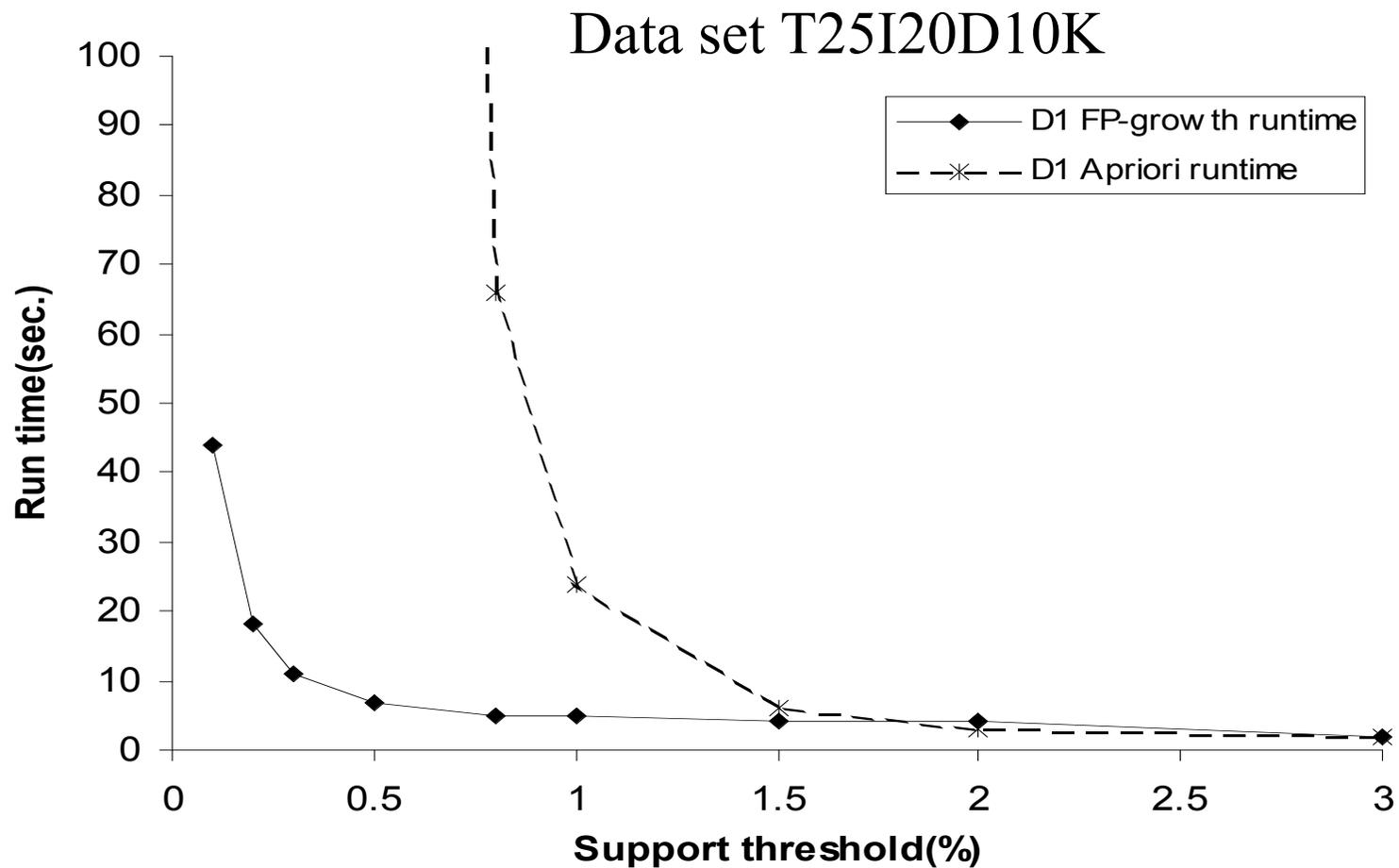
Passi:

1. Ricerca degli elementi frequenti
2. Ordinamento in base alla frequenza
3. Costruzione del FP-tree





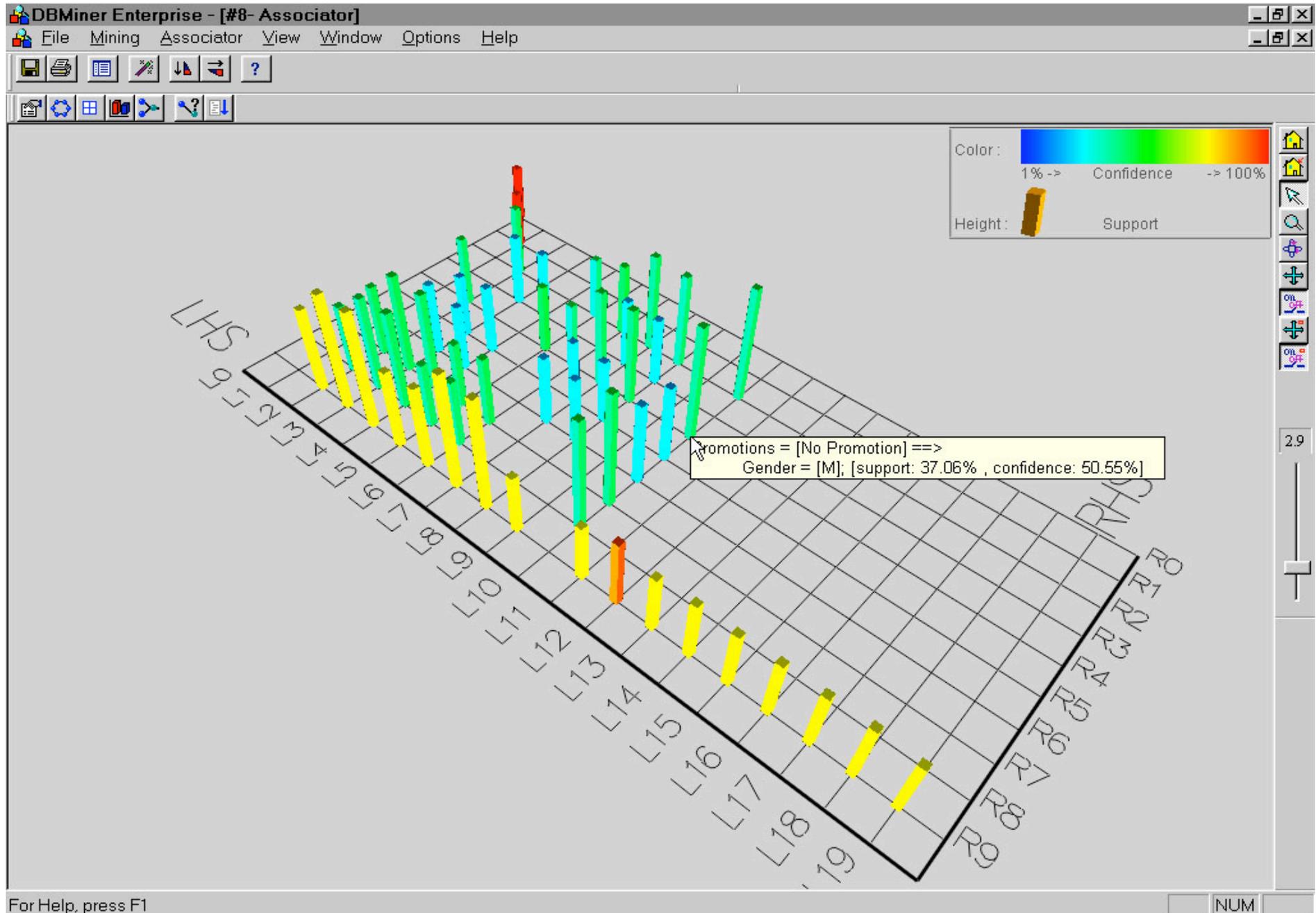
Efficienza dell'algoritmo si FP-tree

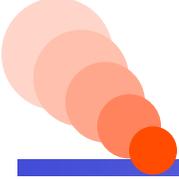


Presentazione di regole associative

	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				

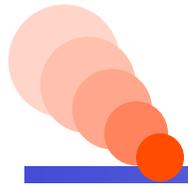
Visualizzazione di regole associative





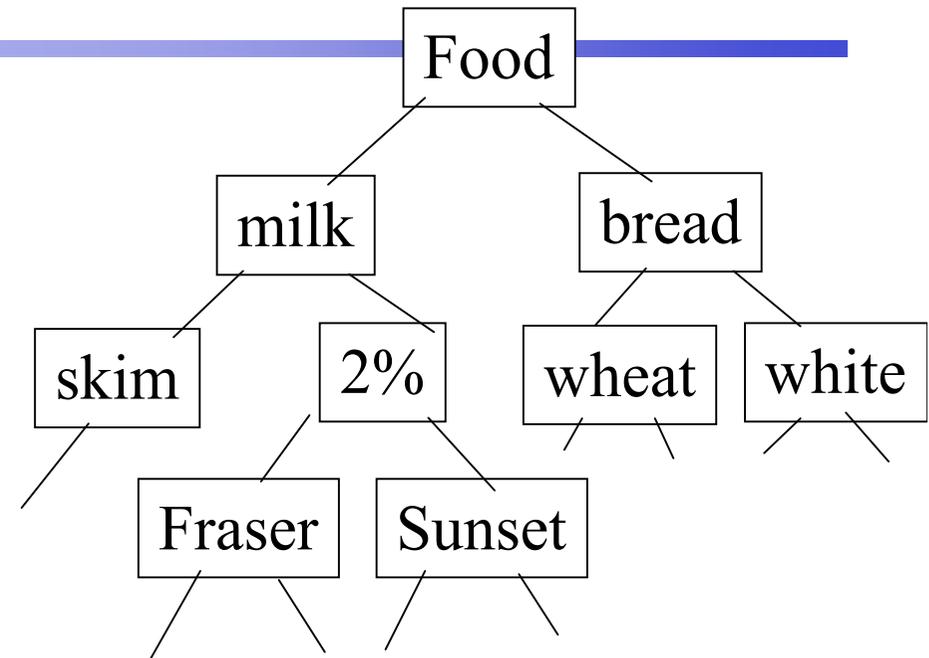
Scoperta di regole associative

- Regole associative
- Regole associative Booleane uni-dimensionali da database transazionali
- **Regole associative multilivello da database transazionali**
- Regole associative multidimensionali
- Dalle regole associative all'analisi di correlazione

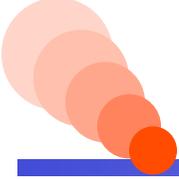


Regole associative multilivello

- Sono definite gerarchie di concetti
- Elementi in basso nelle gerarchie hanno supporti inferiori
- Regole utili possono riguardare valori a diversi livelli

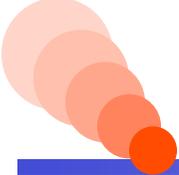


■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
■ ■	■ ■ ■ ■ , ■ ■ ■ ■ , ■ ■ ■ ■ , ■ ■ ■ ■ ■ ■
■ ■ ■ ■	■ ■ ■ ■ , ■ ■ ■ ■ , ■ ■ ■ ■ , ■ ■ ■ ■ ■ ■
■ ■ ■ ■	■ ■ ■ ■ , ■ ■ ■ ■ , ■ ■ ■ ■ , ■ ■ ■ ■ ■ ■
■ ■ ■ ■	■ ■ ■ ■ , ■ ■ ■ ■ ■ ■
■ ■ ■ ■	■ ■ ■ ■ , ■ ■ ■ ■ , ■ ■ ■ ■ , ■ ■ ■ ■ ■ ■ , ■ ■ ■ ■ ■ ■



Ricerca di associazioni multilivello

- Approccio top-down:
 - cercare regole **strong** a livelli elevati:
milk → bread [20%, 60%]
 - cercare raffinamenti a livelli inferiori, ev. con supporti inferiori:
2% milk → wheat bread [6%, 50%]
- Regole a profondità diversa sui diversi concetti
 - livelli incrociati:
2% milk → *Wonder* wheat bread
 - uso di percorsi alternativi nelle gerarchie:
2% milk → *Wonder* bread



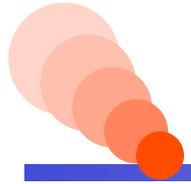
Associazioni multilivello: supporto uniforme o supporto ridotto?

➤ SUPPORTO UNIFORME

- **vantaggi:** soglia unica – se un valore su un certo livello non supera la soglia possiamo potare la ricerca sui livelli più dettagliati
- **svantaggi:** il supporto decresce naturalmente all'aumentare del dettaglio; se la soglia di supporto è
 - **troppo alta** ⇒ perdo associazioni utili più dettagliate
 - **troppo bassa** ⇒ genero molte regole meno dettagliate

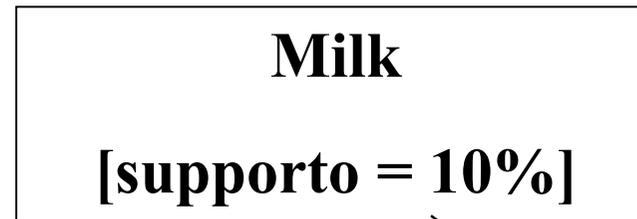
➤ SUPPORTO RIDOTTO

- 4 strategie di ricerca:
 - indipendente sui vari livelli
 - filtro k -insiemi sui vari livelli
 - filtro elementi singoli sui vari livelli
 - mista



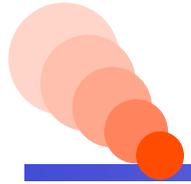
Supporto uniforme

livello 1
min_sup = 5%



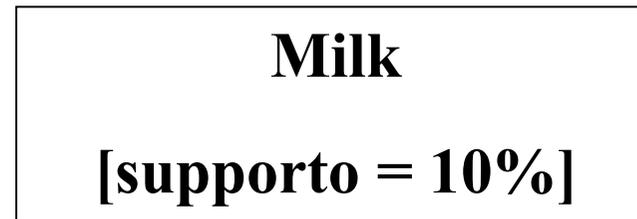
livello 2
min_sup = 5%





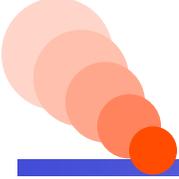
Supporto ridotto

livello 1
min_sup = 5%



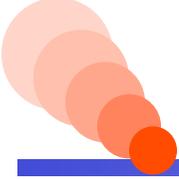
livello 2
min_sup = 3%





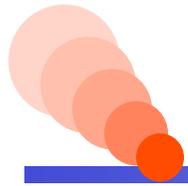
Associazioni multilivello: ridondanza

- Le relazioni ai livelli più elevati potrebbero spiegare (includere) alcune relazioni ai livelli inferiori
- Esempio:
 - milk \Rightarrow wheat bread [8%, 70%]
 - 2% milk \Rightarrow wheat bread [2%, 72%]
- la prima regola è un **ancestor** della seconda
- la seconda regola è ridondante se il suo supporto è vicino a quello "atteso" in base alla regola ancestor, p.es. se $P(2\% \text{ milk}) \sim P(\text{milk}) / 4$



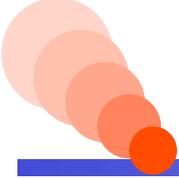
Scoperta di regole associative

- Regole associative
- Regole associative Booleane uni-dimensionali da database transazionali
- Regole associative multilivello da database transazionali
- **Regole associative multidimensionali**
- Dalle regole associative all'analisi di correlazione



Regole associative multidimensionali

- Regole monodimensionali:
 - acquista(X, "milk") \Rightarrow acquista(X, "bread")
- regole multidimensionali: più di due dimensioni o predicati
 - regole inter-dimensionali (*predicati/dimensioni diverse*)
età(X, "19-25") & occupazione(X, "student") \Rightarrow acquista(X, "coke")
 - regole ibride (*predicati/dimensioni ripetuti*)
età(X, "19-25") & acquista(X, "popcorn") \Rightarrow acquista(X, "coke")
- Attributi categorici
 - numero finito di valori, nessun ordinamento significativo
- Attributi quantitativi
 - numerici, ordinamento implicito sempre presente



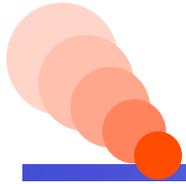
Tecniche per regole multidimensionali

- Si cercano regole su “predicati”
 - esempio: {età, occupazione, acquista} è un insieme di 3 predicati.
 - come trattiamo dati numerici?

- 1. Discretizzazione statica
 - discretizzazione definita a priori

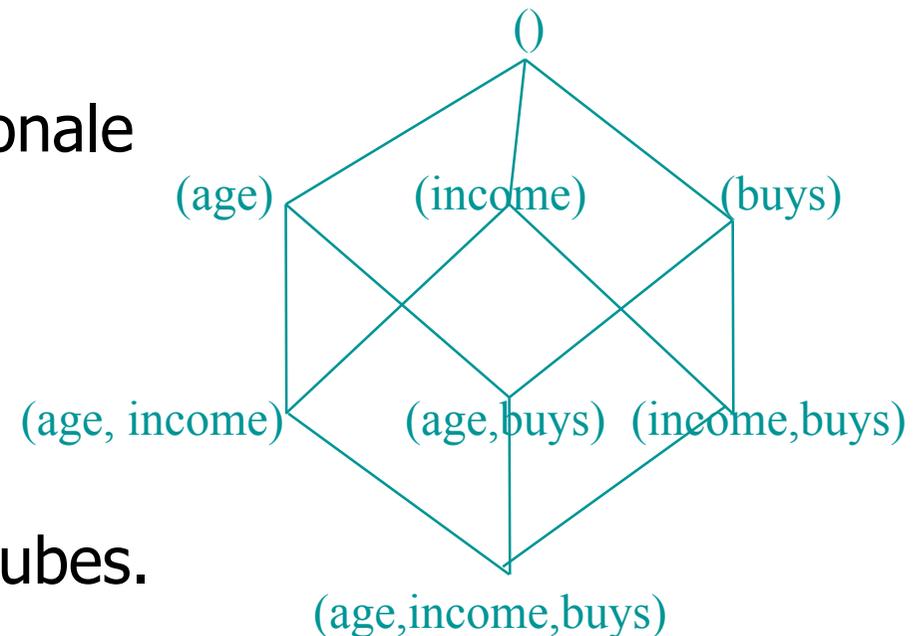
- 2. Discretizzazione basata sulla distribuzione (p.es. ARCS)
 - binning a frequenza costante, binning a massima separazione

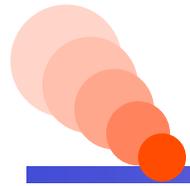
- 3. Regole associative basate sulla distanza (ARCS)
 - tecniche di clustering applicate alle tuple di supporto



Discretizzazione statica

- Valori rimpiazzati da intervalli (gerarchie?)
- Applicabile p.es. su data cubes
- Celle del cuboide n-dimensionale corrispondono all'AND di predicati
- Maggior efficienza su data cubes.





Regole associative quantitative

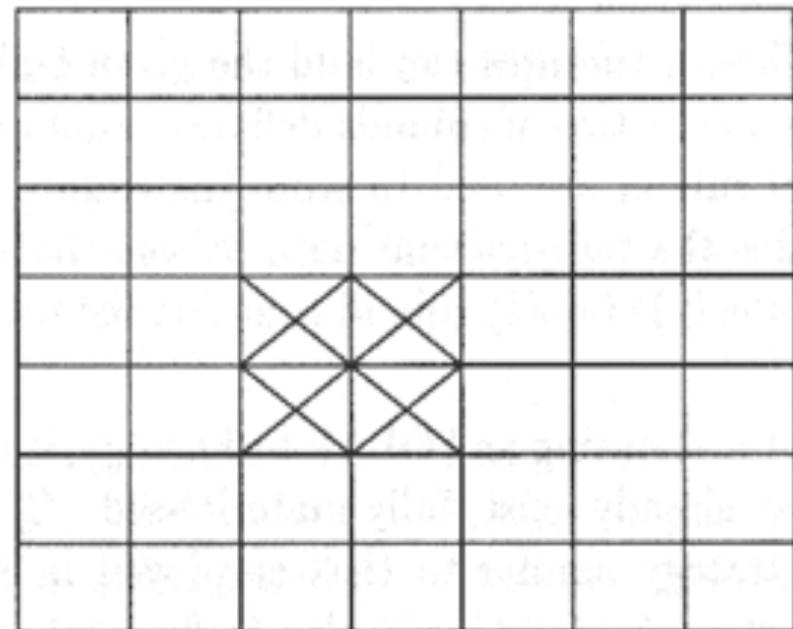
- Discretizzazione dinamica
 - in modo da massimizzare confidenza e compattezza delle regole
- p.es. regole quantitative 2-D: $A_{\text{quan1}} \wedge A_{\text{quan2}} \Rightarrow A_{\text{cat}}$
- Clusterizza (unisce) regole "adiacenti" su una griglia 2-D

- Esempio:

$\text{età}(X, "34-35") \wedge \text{reddito}(X, "30K - 50K") \Rightarrow \text{acquista}(X, "high resolution TV")$

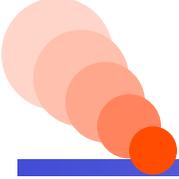
income

70-80K
60-70K
50-60K
40-50K
30-40K
20-30K
<20K



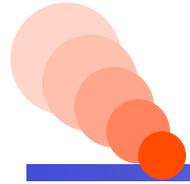
32 33 34 35 36 37 38

age



Scoperta di regole associative

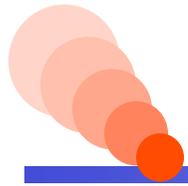
- Regole associative
- Regole associative Booleane uni-dimensionali da database transazionali
- Regole associative multilivello da database transazionali
- Regole associative multidimensionali
- **Dalle regole associative all'analisi di correlazione**



Misure di interesse

- Misure oggettive
 - ① *supporto*
 - ② *confidenza*

- Misure soggettive
 - ① *inattesa*
 - ② *utile*



Critiche a supporto e confidenza

- Esempio "classico": 5000 studenti
 - *giocabasket* ⇒ *cereali* [40%, 66.7%]
 - ⇒ *cereali* 75%
 - *giocabasket* ⇒ *non cereali* [20%, 33.3%]
più interessante, nonostante abbia supporto e confidenza minori

	g. basket	non g. basket	
cereali	2000	1750	3750
non cereali	1000	250	1250
	3000	2000	5000

Critiche a supporto e confidenza

- Esempio 2:
 - X e Y: correlate positivamente
 - X e Z, correlate negativamente

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

lift = 2

- dobbiamo misurare la dipendenza

$$corr_{A,B} = \frac{P(A \cup B)}{P(A)P(B)}$$

	Supporto	Confidenza
X=>Y	25%	50%
X=>Z	37,50%	75%

lift = 0.857

- $P(B|A)/P(B)$: **lift** di $A \Rightarrow B$