Regole Associative

Gianluca Amato

Corso di Laurea in Economia Informatica
Università “G. D'Annunzio”di Chieti-Pescara

Regole Associative

Regole associative booleane mono-dimensionali

Metodi di analisi multi-livello

Regole associative multi-dimensionali

Database di transazioni

- Regole associative vengono spesso ricavate da particolari basi di dati che memorizzano transazioni.
- In questo contesto, una transazione è un elenco di oggetti.
  - Ad esempio, uno scontrino è una transazioni e i prodotti acquistati sono gli oggetti che fanno parte di una transazione.
- Ci sono due modi di vedere una transazione dal punto di vista di un database relazionale:
  - come un attributo multivalore (un valore per ogni oggetto).
    - anche se in realtà gli attributi multivalore non fanno parte della teoria dei DB relazionali...
  - com un insieme di attributi booleani, uno per ogni possibile oggetto. Il valore true indica che l'oggetto fa parte della transazione, false altrimenti.

Cos'è una regola associativa ?

- Una regola associativa è una regola che collega tra di loro
  - gli oggetti di una transazione, oppure
  - i valori degli attributi in un database relazionale.
- Le regole associative hanno la forma
  - Corpo -> Testa [supporto,confidenza]
- Regola associativa sulla transazione “compra”
  - compra(x,”pannolino”) -> compra(x,”birra”) [0.5%, 60%]
    - spesso abbreviata in: pannolino -> bitta [0.5%, 60%]
- Regola associativa sul database studenti
  - laurea(x,”informatica”) & sostenuto(x,”basi dati”) -> voto(x,30)[1%, 75%]

Cosa sono supporto e confidenza?

- Sia data la regola:
  - compra(x,"pannolino") $\rightarrow$ compra(x,"birra")
- Supporto: la percentuale di acquisti che comprendono sia i pannolini che la birra.
  - Prob(pannolino $\cap$ birra)
- Confidenza: tra gli acquisti che includono i pannolini, la percentuale di quelli che includono anche la birra.
  - Prob(birra | pannolino)
- Le regole associative sono considerati interessanti se superano dei valori di soglia sia per confidenza che per il supporto.

Esempio: calcolo supporto e confidenza

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

- Problema: trovare tutte le regole del tipo $X \wedge Y \rightarrow Z$ con
  - supporto e confidenza minimi 50%
- Risultati
  - $A \rightarrow C$ [sup: 50%, conf: 66.6%]
  - $C \rightarrow A$ [sup: 50%, conf: 100%]

Classificazione delle regole associative

- Regole booleane e quantitative: una regola è booleana se concerne solo la presenze o assenza di un certo oggetto in una transazione. Altrimenti è detta quantitativa.
  - Regola booleana mono-dimensionale:
    buys(x, “SQLServer”) ^ buys(x, “DMBook”) → buys(x, “DBMiner”) [0.2%, 60%]
  - Regola quantitativa multi-dimensionale:
    age(x, “30..39”) ^ income(x, “42..48K”) → buys(x, “PC”) [1%, 75%]
- Regole mono-dimensionali e multi-dimensionali: a seconda del numero di attributi coinvolti.
  - Vedi esempio di sopra
- Le regole booleane mono-dimensionali sono le più comuni: corrispondono alla market basket analysis.

Altra terminologia

- Si parla di analisi di associazioni a un singolo livello o a livelli multipli: a seconda se tutte le regole appartengono allo stesso livello di astrazione o no.
  - Una analisi multi-livello è in grado di trovare il seguente insieme di regole:
    - age(x,"30...39") => buys(x,"computer")
    - age(x,"30...39") => buys(x,"laptop computer")
- Ci sono varie possibili estensioni al concetto di regola di associazione. Ad esempio:
  - Analisi di correlazione
  - Analisi di maxpatterns e closed itemsets.

Regole Associative

Regole associative booleane mono-dimensionali

Metodi di analisi multi-livello

Regole associative multi-dimensionali

Frequent Itemsets (1)

- Presentiamo uno degli algoritmi fondamentali per l'analisi di associazioni mono-dimensionali booleane: Apriori.
- Si basa sul concetto di frequent itemset:
  - itemset: un insieme di oggetti che possono far parte delle nostre transazioni. Esempio: {pannolini, birra, nutella} è un itemset per gli acquisti in un supermercato.
  - k-itemset: un itemset con k elementi.
  - la frequenza di un itemset è il numero di transazioni in cui sono presenti tutti gli elementi di un itemset.
  - un frequent itemset è un itemset la cui frequenza supera la soglia minima di supporto.

Frequent Itemsets (2)

- Per trovare le regole associative interessanti
  - Si determinano tutti gli itemset frequenti
  - Se X è un itemset frequente e X=X1∪ X2, allora X1 => X2 è una regola di associazione che supera la soglia minima di sopporto.
  - Se la regola supera anche la soglia di confidenza minima, allora è una regola interessante.

Esempio: itemsets e regole associative

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

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

Per la regola $A \Rightarrow C$:

support = $P(\{A \cap C\}) = 50\%$

confidence = $P(\{A \cap C\})/P(\{A\}) = 66.6\%$

Il principio Apriori

- Qualunque sottoinsieme di un itemset frequent è un itemset frequente.
- Si applica questo principio nel calcolo degli itemset frequenti:
  - In maniera iterativa, trovo tutti i k-itemset frequenti per k da 1 in poi.
  - Tutti i possibili k-itemset frequenti sono ottenuti dall'unione di due (k-1)-itemset frequenti.
    - Non è necessario controllare tutti i possibili sottoinsiemi di k elementi per sapere se sono frequenti.

L'algoritmo Apriori (1)

- Si basa su due passi fondamentali, eseguiti ripetutamente: Join e Prune.
  - Passo Join: partendo da L_{k-1}, l'insieme dei k-itemset frequenti, genero C_k, l'insieme dei k-itemset candidati.
    - Per controllare se $X \in C_k$ è frequente, posso fare una scansione della base di dati e contare in quante transazioni appare X.
    - L'operazione è costosa, si riduce prima il numero di candidati con il passo Prune.
  - Passo Prune: elimino da C_k tutti quegli itemset X per cui esiste un sottoinsieme di k elementi di X che non è in L_{k-1}.

L'algoritmo Apriori (2)

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

L_1 = {frequent items}; // ottenuti da una scansione del database

for (k = 1; L_k !=∅; k++) do begin

C_{k+1} = candidati generati da L_k con i passi Join e Prune

for each transaction t in database do

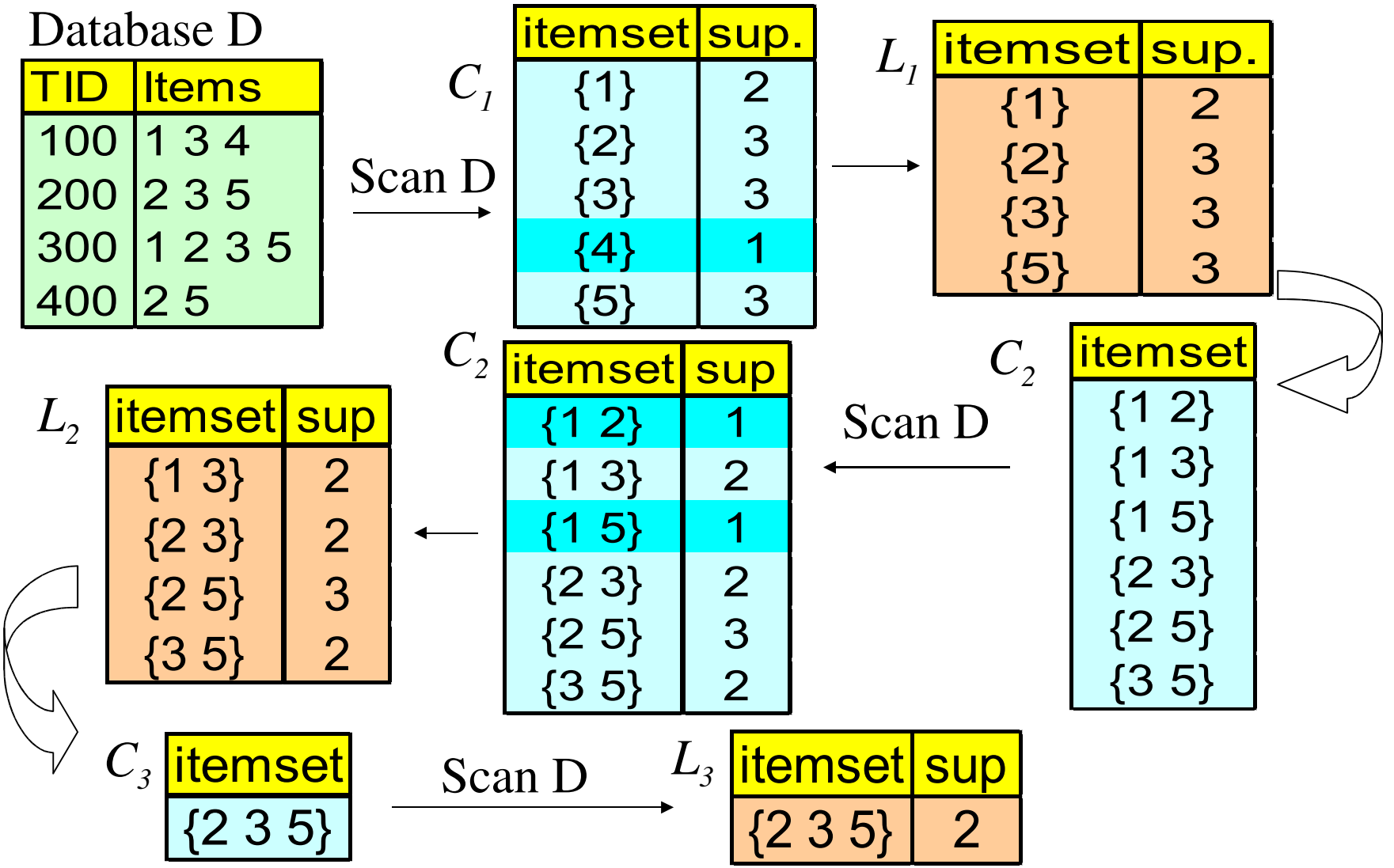
incrementa il conteggio di tutti i candidati in C_{k+1} che sono contenuti in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

Esempio: algoritmo Apriori



Pseduocodice per Join e Prune

- Supponiamo che gli item in L_k e nel database siano ordinati in maniera lessicografica.
- Join:

  insert into C_k

  select $p.item_1$, $p.item_2$, …, $p.item_{k-1}$, $q.item_{k-1}$

  from L_{k-1} p, L_{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:

  forall *itemsets c in* C_k do

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

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

Possibili ottimizzazioni

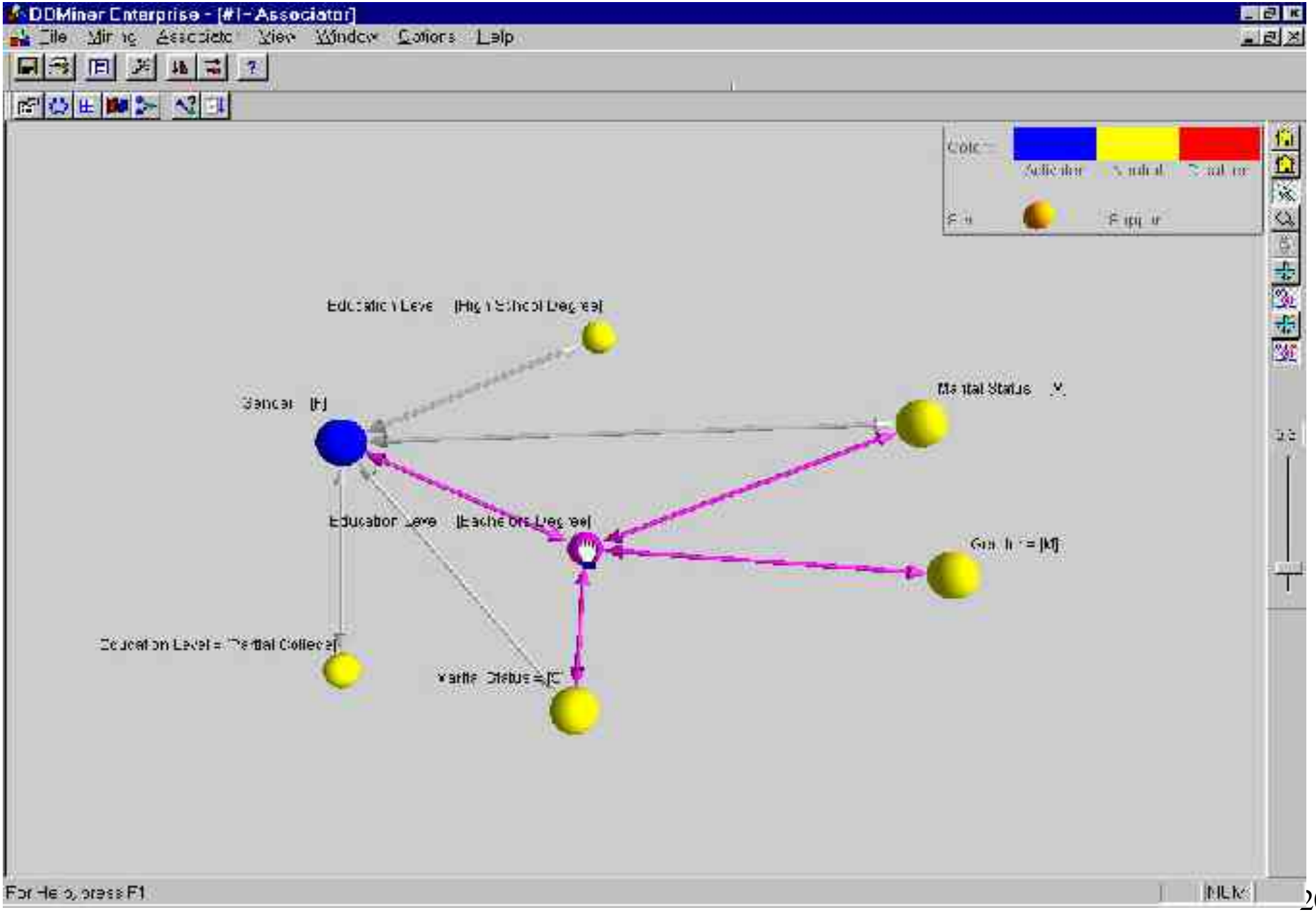
- Ne sono state studiate tante, vediamone due:
  - Transaction reduction: se una transazione non contiene nessun itemset in L_k non potrà contenere nessun itemset in L_{k+1} o successivi. Si può quindi eliminare e non considerare mai più.
  - Partizionamento: si divide il database in tante parti, ognuna delle quali può essere caricata in memoria. Calcolo gli itemset frequenti per le varie partizioni. Alla fine del calcolo gli itemset frequenti globali, tenendo conto che un item frequente per tutto il database è frequente per almeno una delle sue partizioni.

Presentazione delle regole associative (1)

	Body	Implies	Head	Supp (%)	Conf (%)	F	G	H	I
1	cust(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00'	28.48	40.1				
2	cust(x) = '500.00~1000.00'	==>	revenue(x) = '500.00~1000.00'	27.48	39.78				
3	cust(x) = '0.00~1000.00'	==>	order_qty(x) = '0.00~100.00'	64.17	84.01				
4	cust(x) = '0.00~1000.00'	==>	revenue(x) = '1000.00~500.00'	10.48	14.34				
5	cust(x) = '0.00~1000.00'	==>	region(x) = 'United States'	22.56	32.34				
6	cust(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'	28.48	34.54				
8	order_qty(x) = '0.00~100.00'	==>	cost(x) = '1000.00~2000.00'	12.91	15.37				
9	order_qty(x) = '0.00~100.00'	==>	region(x) = 'United States'	25.9	31.48				
10	order_qty(x) = '0.00~100.00'	==>	cost(x) = '0.00~1000.00'	64.17	77.95				
11	order_qty(x) = '0.00~100.00'	==>	product_line(x) = 'Tents'	13.52	16.43				
12	order_qty(x) = '0.00~100.00'	==>	revenue(x) = '500.00~1000.00'	10.57	23.38				
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	27.95				
16	revenue(x) = '0.00~500.00'	==>	cost(x) = '0.00~1000.00'	28.48	100				
17	revenue(x) = '0.00~500.00'	==>	order_qty(x) = '0.00~100.00'	28.48	100				
18	revenue(x) = '1000.00~1500.00'	==>	cost(x) = '0.00~1000.00'	10.48	96.78				
19	revenue(x) = '500.00~1000.00'	==>	cost(x) = '0.00~1000.00'	27.48	100				
20	revenue(x) = '500.00~1000.00'	==>	order_qty(x) = '0.00~100.00'	24.37	96.14				
21									
22									
23	cust(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.48	40.1				
24	cust(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.48	40.1				
25	cust(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	24.37	27.95				
26	cust(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	24.37	27.95				
27	cust(x) = '0.00~1000.00' AND order_qty(x) = '0.00~100.00'	==>	revenue(x) = '500.00~1000.00'	10.57	33.23				

Sheet1

Presentazione delle regole associative (2)



Regole Associative

Regole associative booleane mono-dimensionali

Metodi di analisi multi-livello

Regole associative multi-dimensionali

Regole associative a più livelli

- Per molte applicazioni, è difficile trovare delle associazioni forti tra oggetti a un livello primitivo di dettaglio.
  - Occorre considerare regole associativi a tutti i livelli della gerarchia di concetti.
- Approccio top-down:
  - Si calcolano gli itemset a partire dal livello di astrazione più generico;
    - latte -> pane [20%,60%]
  - Si procede via via verso livello sempre più specifici.
    - latte scremato -> pane integrale [6%, 50%]

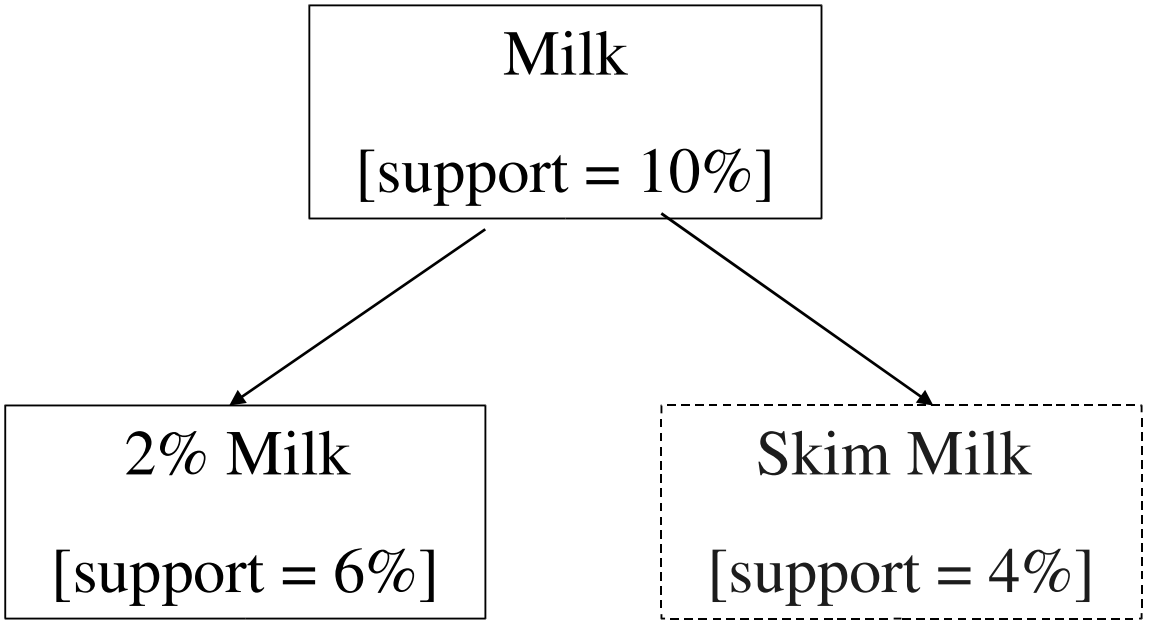
Supporto uniformo e supporto ridotto

- Supporto uniforme: lo stesso livello di supporto minimo è usato a tutti i livelli di astrazione.
  - Efficiente! Se un itemset a un certo livello di astrazione non è frequente, non dobbiamo esaminarlo per nessuno dei livelli di dettaglio maggiori.
  - È difficile scegliere i livello di supporto appropriato:
    - Troppo basso: genera troppe regole associative
    - Troppo alto: si rischia di mancare del tutto regole associative ad alto livello di dettaglio.
- Supporto ridotto: valori di supporto minimo diversi per ogni livello.

Supporto uniforme

Level 1
min_sup = 5%

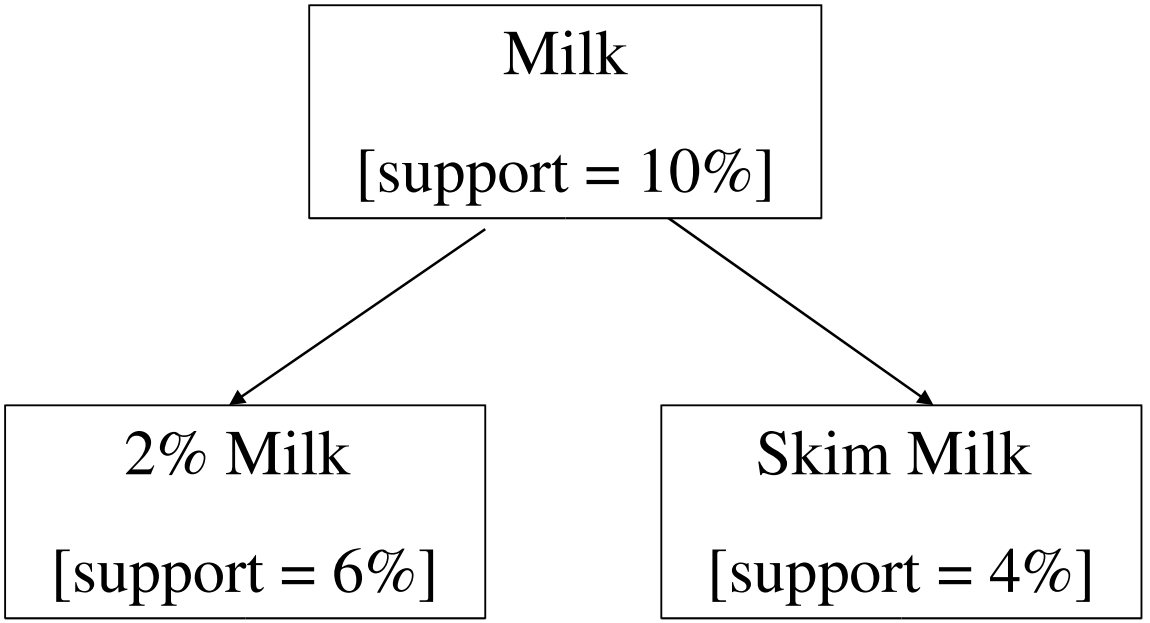
Level 2
min_sup = 5%



Supporto ridotto

Level 1
min_sup = 5%

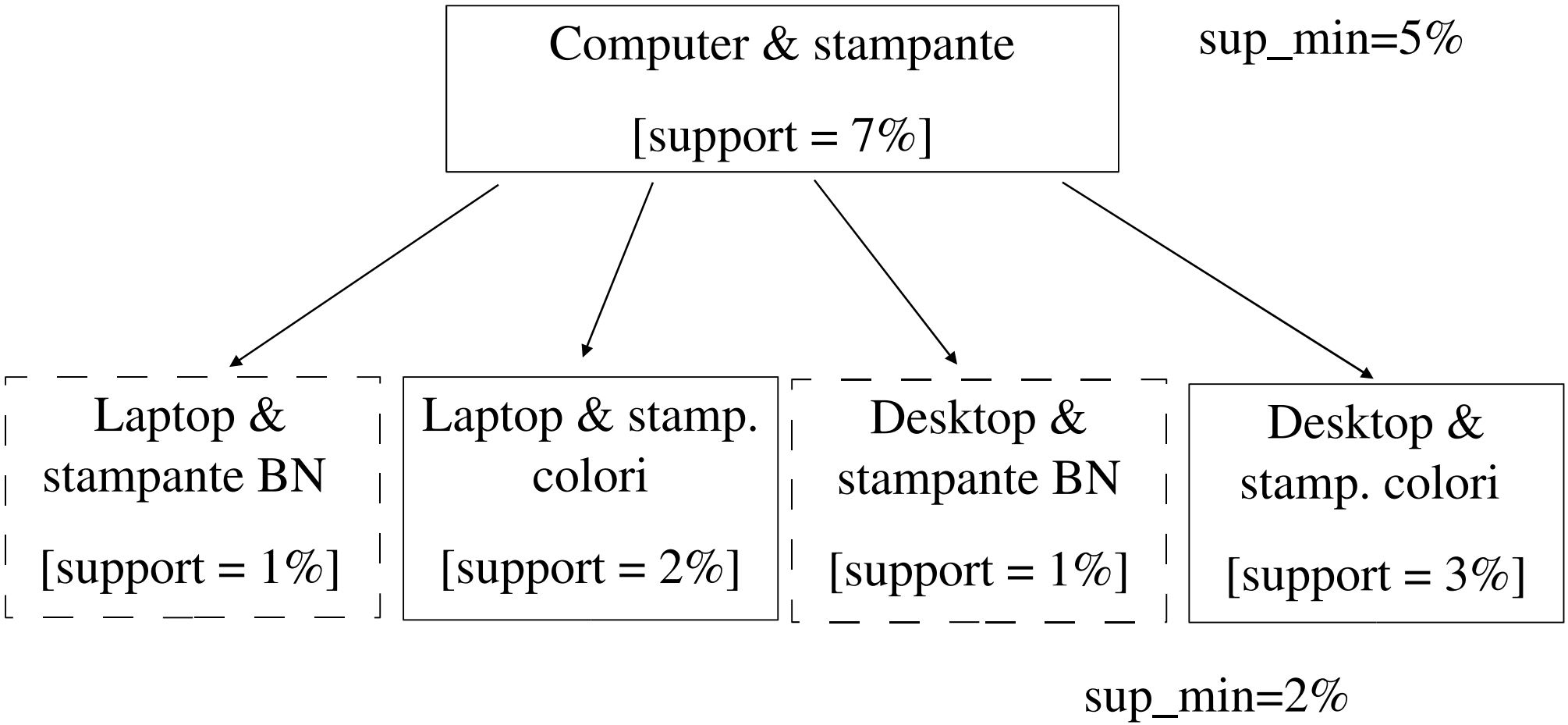
Level 2
min_sup = 3%



Strategie per il metodo di supporto ridotto (1)

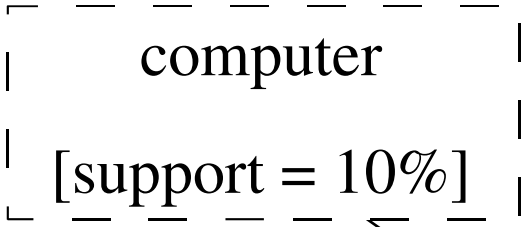
- Livelli indipendenti: gli itemset a differenti livelli di astrazione sono calcolati in maniera indipendente tra di loro.
  - Strategia completa: trova tutti gli itemset frequenti
- Filtraggio incrociato: un k-itemset a livello i viene considerato un candidato solo se al livello i-1 è frequente.
  - Strategia efficiente: ma può non trovare tutti gli itemset frequenti.
- Filtraggio incrociato sul singolo item: stessa cosa del filtraggio incrociato, ma il filtraggio avviene solo a livello di 1-itemset.
  - Compromesso tra i due

Filtraggio incrociato



Filtraggio incrociato sul singolo item

Level 1
min_sup = 12%



Level 2
min_sup = 3%

Laptop
(non esaminato)

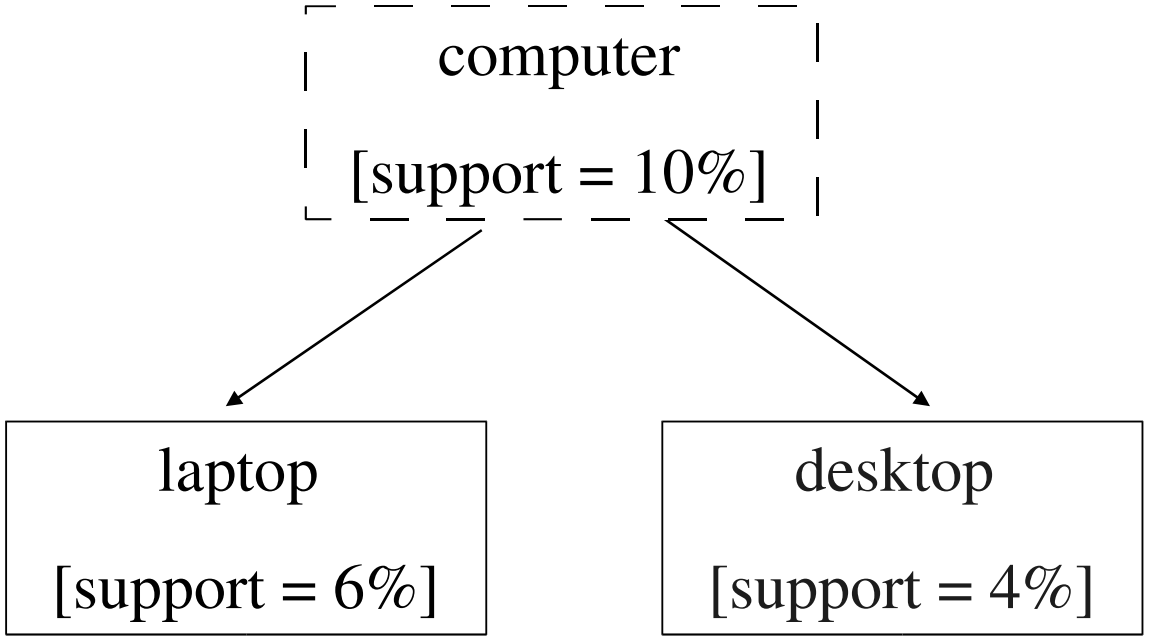
Desktop
(non esaminato)

Strategie per il metodo di supporto ridotto (1)

- Filtraggio incrociato controllato sul singolo item: per ogni livello abbiamo due soglie:
  - Soglia di supporto minimo
  - Soglia di passaggio di livello
- Gli itemset frequenti sono quelli la cui frequenza supera il supporto minimo.
- Gli itemset che vengoo considerati per il livello successivo di dettaglio sono quelli la cui frequenza supera la soglia di passaggio.

Filtraggio incrociato controllato

Level 1
min_sup = 12%
level_passage=10%



Level 2
min_sup = 3%

Cross-level association rules

- Finora abbiamo considerato regole i cui i vari item hanno tutti lo stesso livello di dettaglio.
- Cross-level association rules: quando i livelli di dettaglio dei vari item differiscono.
  - Computer -> stampante BN [sup: 4%, conf. 30%]

Associazioni multi-livello ridondanti

- Alcune regole possono essere ridondanti a causa di regole più astratte.
- Esempio:
  - R1) Latte => pane integrale [sup: 8%, conf: 70%]
  - R2) Latte scremato => pane integrale [sup: 2%, conf: 72%]
- Diciamo che la regola R1 è antenato della R2: la R1 si ottiene da R2 rimpiazzando qualche oggetto con un oggetto a un livello di astrazione superiore.
- Supponiamo che ¼ delle vendite di latte è per latte scremato
  - Supporto atteso "Latte scremato => pane integrale" è 1/4*8%=2%
  - La confidenza attesa è il 70%
  - Seconda regola ridondante!!

Regole Associative

Regole associative booleane mono-dimensionali

Metodi di analisi multi-livello

Regole associative multi-dimensionali

Regole associative multi-dimensionali (1)

- Regole associative mono-dimensionali si estraggono da database transazionali.
- Regole associative multi-dimensionali:
  - Regole inter-dimensionali (ogni predicato occorre una sola volta):
    - age(X,”19--25”) & occupation(X,”student”) -> buys(X,”coke”);
    - estratti da database relazionali
  - Regole ibride (predicati ripetuti)
    - age(X,”19--25”) & buys(X,”pop corn”) -> buys(X,”coke”)

Regole associative multi-dimensionali (2)

- Regole multi-dimensionali booleane
  - Si usa un algoritmo simile ad Apriori
  - Invece di considerare item-sets si considerano predicate-sets
    - {age (X,"20-29"), occupation(X,"student")} è un 2-predicate set.
- Regole multi-dimensionali quantitative.
  - Gli attributi quantitativi vanno discretizzati. Ci sono due approcci:
  - discretizzazione statica, eseguita prima di applicare l'algoritmo di ricerca delle regole associative
    - una volta discretizzati, si applica un algoritmo standard come Apriori.
  - discretizzazione dinamica, eseguita in parallelo alla ricerca di regole associative.

Discretizzazione statica e dinamica

- La discretizzazione dinamica produce regole “migliori”, tipicamente con un supporto e confidenza più elevati.
- Supponiamo di discretizzare age negli intervalli “1-18”, “19-36”, “37-48”,...
- Applicando Apriori possiamo ottenere regole del tipo

  age(X,”1-18”) & buys(X,”computer”) -> buys(X,”joystick”)
  age(X,”19-36”) & buys(X,”computer”) -> buys(X,”joystick”)

- Un algoritmo che usa una discretizzazione dinamica è in grado di ottenere:

  age(X,”12-28”) & buys(X,”computer”) -> buys(X,”joystick”)

  con confidenza e supporto maggiori.
  - l'intervallo “12-28” è generato ad-hoc per migliorare le prestazioni della regola associativa.