Regole Associative

Sistemi informativi per le Decisioni

Slide a cura di Prof. Claudio Sartori

Data Mining

- Estrazione (mining) di regolarità (pattern) da grandi quantità di dati.
  1. Analisi complesse
  2. Algoritmi efficienti e scalabili
- Si noti che:
  - I sistemi OLAP, di data o Information Retrieval non soddisfano necessariamente la proprietà 1.
  - Molti sistemi di apprendimento automatico (machine learning) e strumenti di analisi statistica non soddisfano necessariamente la 2.

Tipi di scoperta

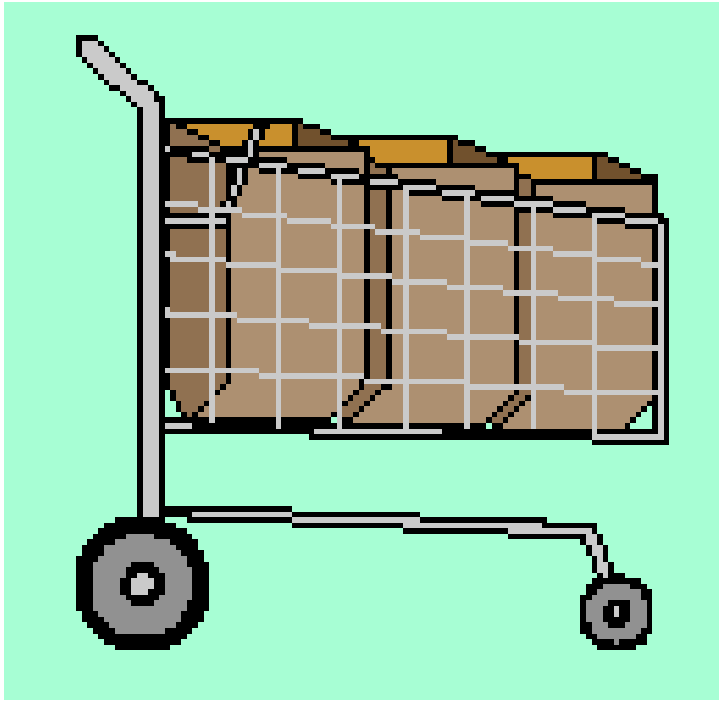
- Individuazione delle dipendenze
- Identificazione delle classi
- Descrizione delle classi
- Individuazione di outliers/eccezioni

Aree applicative

- Analisi di dati e supporto alle decisioni
  - Analisi e gestione dei mercati
  - Analisi e gestione del rischio
  - Scoperta e gestione delle frodi
- Altre applicazioni
  - Text mining e analisi del web
    - Documenti, news group, ...
  - Intelligent query answering

Scenario *“supermercato”*

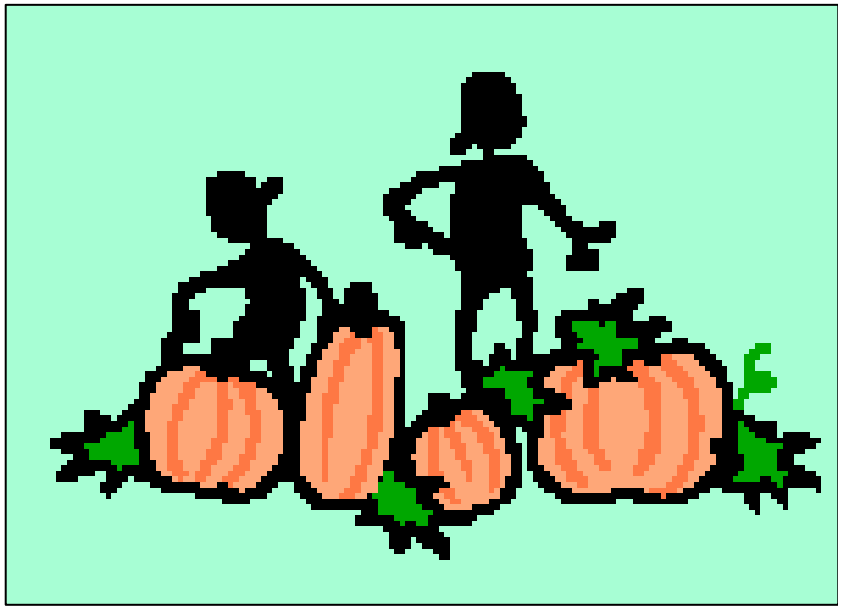
Una catena di supermercati, sulla base delle transazioni di vendite dei prodotti, vuole stabilire le correlazioni esistenti tra le varie vendite



scoperta di regole di associazione

Scenario *“user/customer profiling”*

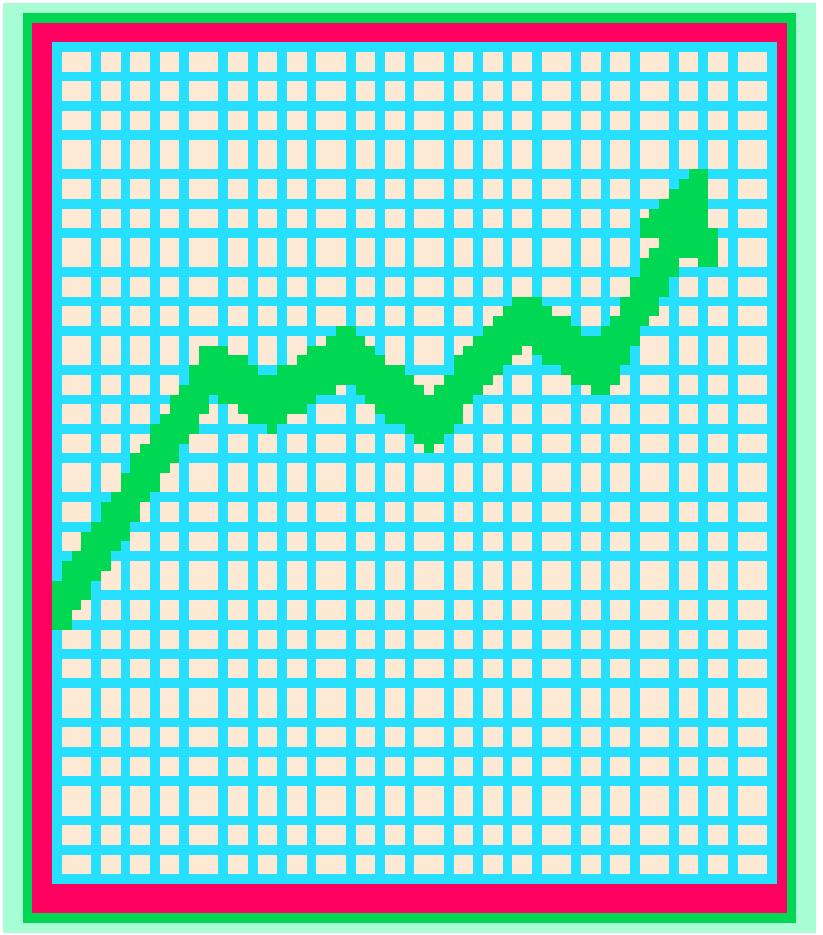
Un’azienda che vende i suoi prodotti via web è interessata a “personalizzare” i contatti postali con offerte e novità inviate periodicamente ai clienti, cercando di individuare dei “profili” (comportamenti) comuni, o perlomeno simili



individuazione di cluster

Scenario *“investimenti finanziari”*

Una società che opera
in ambito finanziario vuole
confrontare l’andamento
di un titolo quotato in borsa
con quello di altre
compagnie, per avere
informazioni utili a fini
di investimento



analisi di serie temporali

Analisi e gestione dei mercati

- Sorgenti di dati
  - Transazioni di carte di credito, carte fedeltà, buoni sconto, lamentele clienti
  - Studi su abitudini dei consumatori
- Individuazione di gruppi di clienti target
  - Trovare gruppi (cluster) che modellano clienti di caratteristiche uniformi
- Determinare pattern di acquisto
  - Es.: conversione di un conto in banca da singolo a co-intestato ➔matrimonio
- Analisi cross-market
  - Associazioni tra vendite di prodotti
  - Predizioni basate sull'associazione

Regole Associative - argomenti

- Cosa sono le regole associative (AR)
  e per cosa sono usate
  - L'applicazione paradigmatica: Market Basket Analysis (MBA)
  - AR mono-dimensionali (intra-attribute)
- Come calcolare le AR
  - Algoritmo Apriori di base e sue ottimizzazioni
  - AR multi-dimensionali (inter-attribute)
  - Quantitative AR
- Come ragionare sulle AR e come valutarne la qualità
  - Multiple-level AR
  - Significatività (interestingness)
  - Correlazione vs. Associazione

Scoperta di regole associative

- Esempio: *market basket analysis*
  - scenario supermercato
  - codici a barre, fidelity card
  - informatizzazione casse vendita
  - enormi quantità di dati sulle vendite disponibili in forma elettronica

Scoperta di regole associative - concetti base

- Transazione
  - insieme di elementi (item) acquistati congiuntamente (quello che si trova in un carrello della spesa)
- Regola Associativa
  - dato un insieme di item I e un insieme di transazioni D, una regola associativa del tipo

    $X \Rightarrow Y$ (X implica Y) (con $X,Y \subset I$ e $X \cap Y = \varnothing$)

    è un'implicazione

  - *chi compra X compra anche Y*

Concetti base (ii)

- Supporto di una regola
  - la regola $X \Rightarrow Y$ ha supporto *s* se una frazione pari a *s* delle transazioni contengono tutti gli item in $X \cup Y$

    es.: (il 40% delle transazioni natalizie include panettone e champagne)
- Confidenza di una regola
  - la regola $X \Rightarrow Y$ ha confidenza *c* se una frazione pari a *c* delle transazioni in cui compare X contiene Y

    es.: (a Natale, l'80% delle persone che comprano champagne comprano anche il panettone)
- Problema:
  - determinare tutte le regole associative che abbiano supporto almeno pari a MINSUPP e confidenza almeno pari a MINCONF

Esempio

- La regola A $\Rightarrow$ C ha
  - Supporto pari al 50%, perché {A C} compare in 2 transazioni su 4
  - Confidenza pari al 66%, perché su 3 transazioni in cui compare A, in due compare anche C
- La regola C $\Rightarrow$ A ha
  - Supporto pari al 50%
  - Confidenza pari al 100%

Transaction ID	*Items*
100	A B C
200	A C
300	A D
400	B E F

Decomposizione del problema

R. Agrawal, T. Imielinski, Arun Swami: “Mining Association Rules between Sets of Itemsets in Large Databases”. ACM SIGMOD Conference, 1993.

- Il primo lavoro sulla scoperta di regole associative ha mostrato come il problema si possa decomporre in due fasi, sfruttando l'osservazione che

$$c(X \Rightarrow Y) = s(X \cup Y)/s(X)$$

dove s è il supporto di un insieme, ovvero la frazione di transazioni che lo contiene

Decomposizione (continua)

- Scoperta dei “large itemsets”
  - si determinano tutti gli insiemi di item che hanno supporto almeno pari a MINSUPP
- Generazione delle regole
  - se L è un “large itemset”, allora anche ogni sottoinsieme X di L lo è, con supporto $s(X) \geq s(L)$, e quindi è fornito in output dalla fase precedente

Regole Associative - argomenti

- Cosa sono le regole associative (AR)
  e per cosa sono usate
  - L'applicazione paradigmatica: Market Basket Analysis (MBA)
  - AR mono-dimensionali (intra-attribute)
- Come calcolare le AR
  - Algoritmo Apriori di base e sue ottimizzazioni
  - AR multi-dimensionali (inter-attribute)
  - Quantitative AR
- Come ragionare sulle AR e come valutarne la qualità
  - Multiple-level AR
  - Significatività (interestingness)
  - Correlazione vs. Associazione

Algoritmo Apriori Formulazione base

- Trova gli insiemi frequenti di oggetti (FI=frequent itemset):
  - devono soddisfare il vincolo sul supporto
  - un sottoinsieme di un FI è a sua volta un FI
    - se {A,B} è frequente anche {A} e {B} lo sono, con supporto maggiore
  - iterativamente trova i FI con cardinalità da 1 a K
- usa i FI per generare le regole associative

La proprietà Apriori

- Se B è frequente e A $\subseteq$ B allora anche A è frequente
  - ogni transazione che contiene B contiene anche A, quindi supp.(A) $\geq$ supp.(B))
- Conseguenza: se A non è frequente, non è necessario generare e valutare gli insiemi che lo includono
- Esempio:

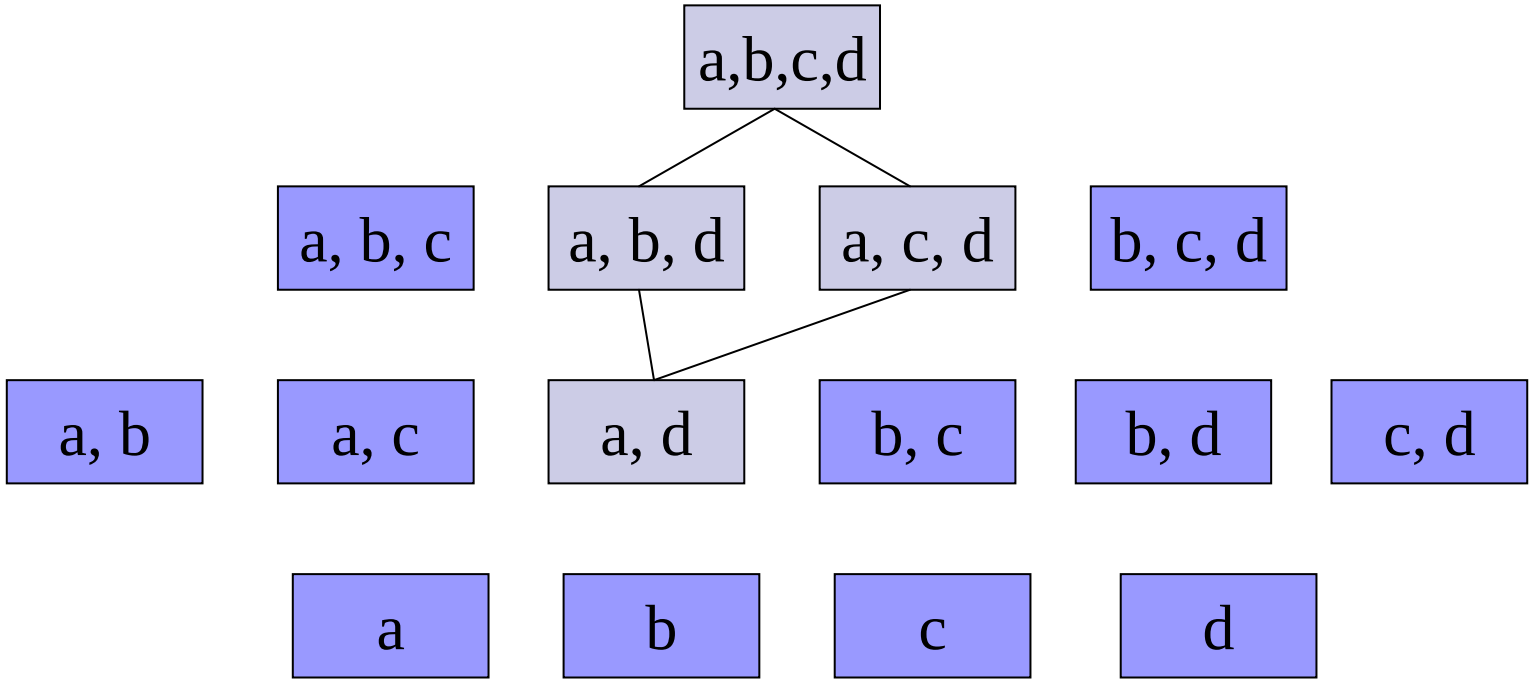
  <1, {a, b}> <2, {a} >

  <3, {a, b, c}> <4, {a, b, d}>

  con MINSUPP = 30%.
  - Allora {c} non è frequente e non occorre valutare:

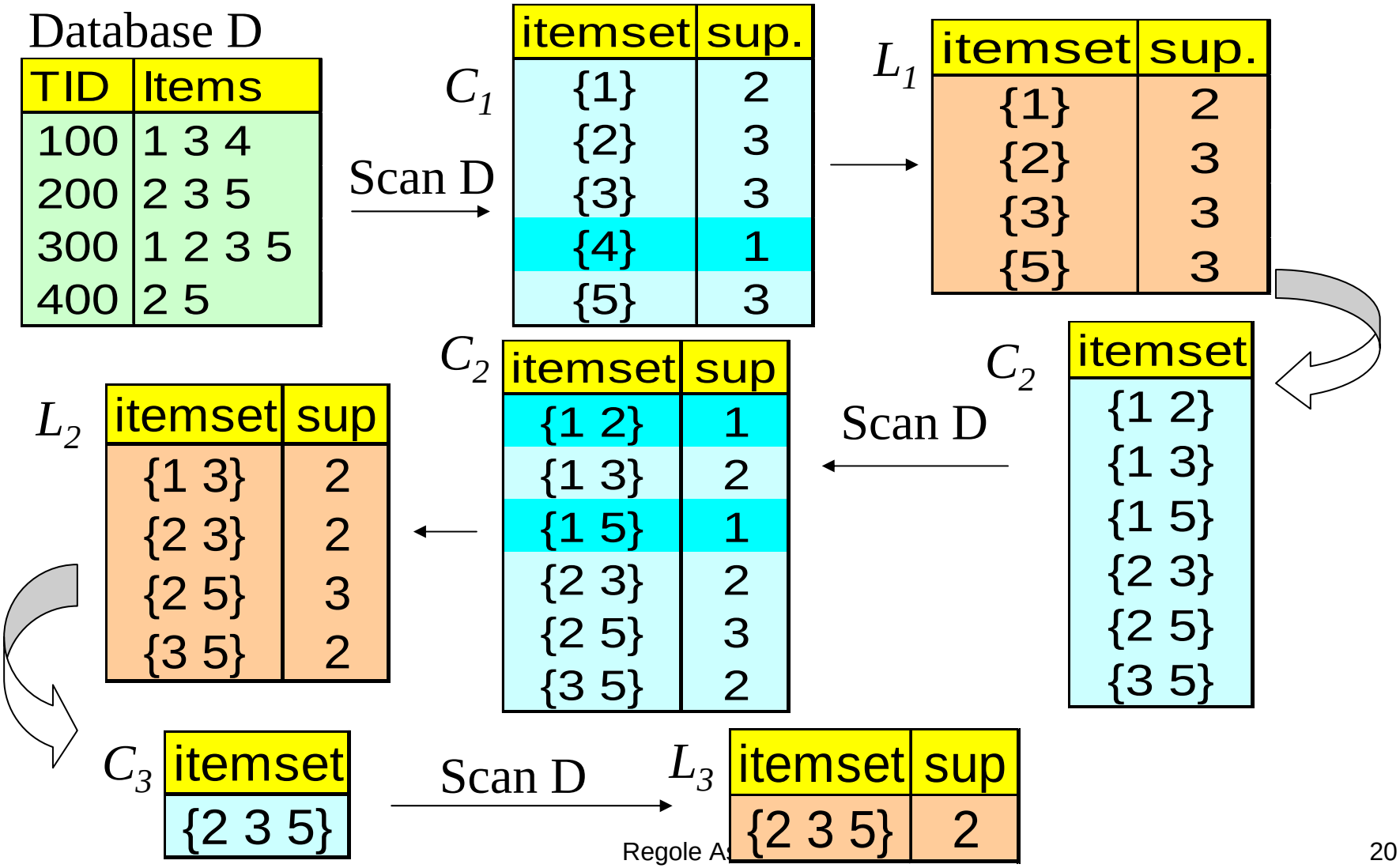
    {c, a}, {c, b}, {c, d}, {c, a, b}, {c, a, d}, {c, b, d}

Apriori - Esempio



{a,d} non è frequente, quindi i 3-insiemi {a,b,d}, {a,c,d}
e il 4-insieme {a,b,c,d}, non sono generati

L'algoritmo Apriori - Esempio



L'algoritmo Apriori

- Join Step: C_k è generato congiungendo L_{k-1} con se stesso
- Prune Step: Ogni (k-1)-itemset che non è frequente non può essere sottoinsieme di un k-itemset frequente
- Pseudo-code:

C_k: itemset candidato di dimensione k
L_k : itemset frequente di dimensione k

L_1 = {item frequenti};
for (k = 1; L_k !=∅; k++) do begin
  C_{k+1} = candidati generati da L_k;
  for each transazione t in database do
    incrementa il conteggio di candidati in C_{k+1} che sono contenuti in t
  L_{k+1} = candidati in C_{k+1} con min_support
  end
return $\cup_k L_k$;

Come generare i candidati?

- Si suppongano gli item in L_{k-1} ordinati
- Step 1: self-joining L_{k-1}

  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$_{k-1}$
- Step 2: pruning

  for all itemsets *c* in C_k do

  for all (*k*-1)-subsets *s* of *c* do

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

Generazione di candidati - esempio

- L_3={abc, abd, acd, ace, bcd}
- Self-joining: L_3*L_3
  - abcd da abc e abd
  - acde da acd e ace
  - altri?
- Pruning:
  - acde è eliminato perché ade non è in L_3
- C_4={abcd}

Come contare il supporto per i candidati?

- Perché è un problema?
  - il numero totale di candidati può essere enorme
  - una transazione può contenere molti candidati

- Metodo:
  - Gli itemsets candidati sono immagazzinati in un hash-tree
  - Leaf node di hash-tree contiene una lista di itemset e conteggi
  - Interior node contiene una hash table
  - Subset function: trova tutti i candidati contenuti in una transazione

Metodi per migliorare l'efficienza di Apriori

- DHP: Direct Hash and Pruning (Park, Chen and Yu, SIGMOD '95)
- Partitioning Algorithm (Savasere, Omiecinski, and Navathe, VLDB '95)
- Sampling (Toivonen, '96)
- Dynamic Itemset Counting (Brin et al., SIGMOD '97)

Metodi per migliorare l'efficienza di Apriori

- L'efficienza può essere migliorata:
  - riducendo la dimensione della base di dati da considerare nei passaggi successivi
  - riducendo il numero di candidati da considerare, usando tecniche di indirizzamento e partizionamento
  - riducendo il numero di "scan" dell'intera base di dati
- Transaction reduction: una transazione che non contiene nessun k-itemset frequente può essere trascurata nei passaggi successivi
- Dynamic itemset counting: aggiungi un nuovo itemset candidato durante la scansione, sulla base dei dati analizzati fino a quel momento
- Campionamento

Efficienza: Hash-based itemset counting

- Ridurre il numero dei candidati in C_k
- Anziché memorizzare i membri di C_k separatamente, si raccolgono in bucket tramite una funzione di hash
  - ad ogni inserimento si incrementa un conteggio relativo al bucket
  - l'uso di bucket e hash è molto veloce
  - al termine basta eliminare i bucket con un conteggio inferiore al supporto minimo, poiché a maggior ragione conterrà soltanto candidati con supporto inferiore al minimo
- Un *k*-itemset appartenente a un hash-bucket sotto la soglia non può essere frequente
- possibile riduzione sostanziale dei candidati, particolarmente per *k*=2

Efficienza: Partitioning

- Un itemset potenzialmente frequente nel DB
  deve essere frequente in almeno una porzione del DB
- Fase 1:
  - si divide l'insieme delle transazioni in n partizioni,
    in modo tale che ciascuna partizione stia in memoria
  - il supporto minimo, moltiplicato per il numero di transazioni
    in una partizione, fornisce il supporto minimo in quella partizione
  - si individuano gli itemset frequenti "localmente alla partizione"
- Fase 2: i soli itemset che sono frequenti
  in almeno una partizione sono combinati per valutare
  se sono frequenti anche globalmente
- si effettuano solo due scan completi, uno per fase

Generare Regole Associative dai Frequent Itemsets

- sono generate unicamente regole "forti"
- i FI soddisfano una soglia di minimo supporto
- le regole forti soddisfano una soglia di minima confidenza

- conf(A $\Rightarrow$ B) = Pr(B | A) = $\frac{support(A \cup B)}{support(A)}$

for each frequent itemset, **f**, generate all non-empty subsets of **f**
for every non-empty subset **s** of **f** **do**
    if supp(**f**)/supp(**s**) $\geq$ min_conf **then**
        output rule **s** $\Rightarrow$ **(f-s)**
end

Regole Associative Multidimensionali

Associazioni tra valori di diversi attributi :

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%]

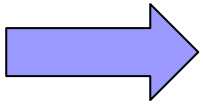
Confronto mono vs multi-dimensionale

- **Mono-dimensionale (Intra-attributo)**
  - gli eventi sono: *items A, B e C appartengono alla stessa transazione*
  - verificarsi di eventi: *transazioni*
- **Multi-dimensionale (Inter-attributo)**
  - gli eventi sono : *l'attributo A assume valore* a, *l'attributo B assume valore* b *e l'attributo C assume valore* c
  - verificarsi di eventi: *tuple*

Confronto mono vs multi-dimensionale

Multi-dimensionale

<1, Italian, 50, low>
<2, French, 45, high>

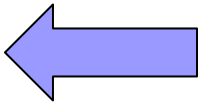


Mono-dimensionale

<1, {nat/Ita, age/50, inc/low}>
<2, {nat/Fre, age/45, inc/high}>

Schema: <ID, a?, b?, c?, d?>

<1, yes, yes, no, no>
<2, yes, no, yes, no>



<1, {a, b}>
<2, {a, c}>

Attributi Quantitativi

- Attributi quantitativi (es. età, reddito)
- Attributi categorici (es. colore di un'automobile)

CID	**height**	**weight**	**income**
1	168	75,4	30,5
2	175	80,0	20,3
3	174	70,3	25,8
4	170	65,2	27,0

- Problema: troppi valori distinti
- Soluzione: trasformare valori quantitativi in categorici con la discretizzazione

Regole associative quantitative

CID	Age	Married	NumCars
1	23	No	1
2	25	Yes	1
3	29	No	0
4	34	Yes	2
5	38	Yes	2

[Age: 30…39] and [Married: Yes] $\Rightarrow$ [NumCars:2]

- supporto = 40%
- confidenza = 100%

Discretizzazione di attributi quantitativi

- **Soluzione**: ogni valore è sostituito dall'intervallo a cui appartiene

  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

- Problema: la discretizzazione può essere inutile (**weight**).

Binning

- il dominio di valori di un attributo quantitativo viene partizionato in intervalli (bin)
- ogni tupla appartiene a un bin se il valore dell'attributo sotto esame appartiene all'intervallo associato al bin
  - equi-ampiezza
    - generalmente insoddisfacente
  - equi-profondità
    - ogni bin ha approssimativamente lo stesso numero di tuple assegnate
  - basato sull'omogeneità
    - in modo da massimizzare l'omogeneità delle tuple che appartengono allo stesso bin
    - potrebbe essere eseguito con un clustering mono-dimensionale

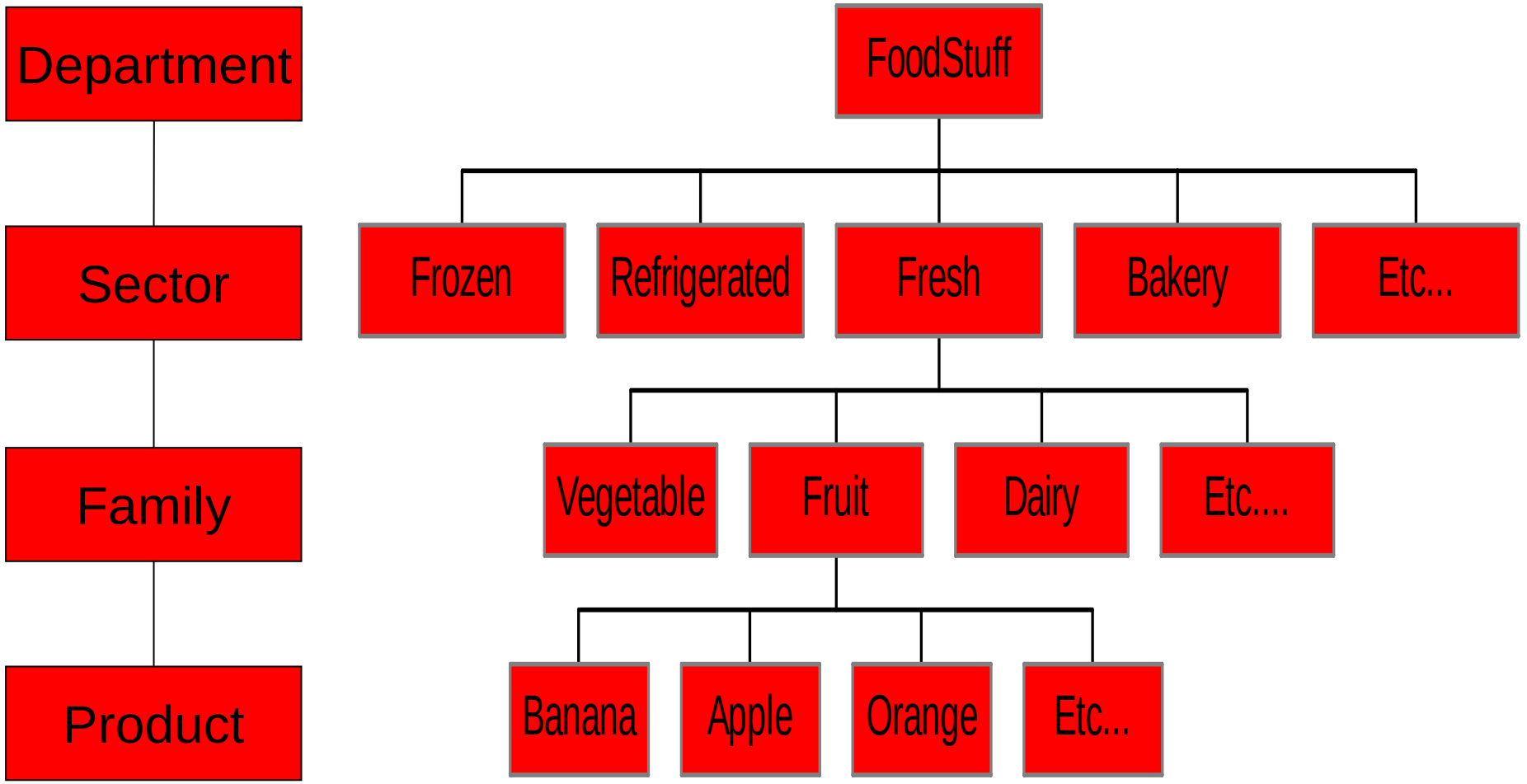
Regole Associative - argomenti

- Cosa sono le regole associative (AR)
  e per cosa sono usate
  - L'applicazione paradigmatica: Market Basket Analysis (MBA)
  - AR mono-dimensionali (intra-attribute)
- Come calcolare le AR
  - Algoritmo Apriori di base e sue ottimizzazioni
  - AR multi-dimensionali (inter-attribute)
  - Quantitative AR
- Come ragionare sulle AR e come valutarne la qualità
  - Multiple-level AR
  - Significatività (interestingness)
  - Correlazione vs. Associazione

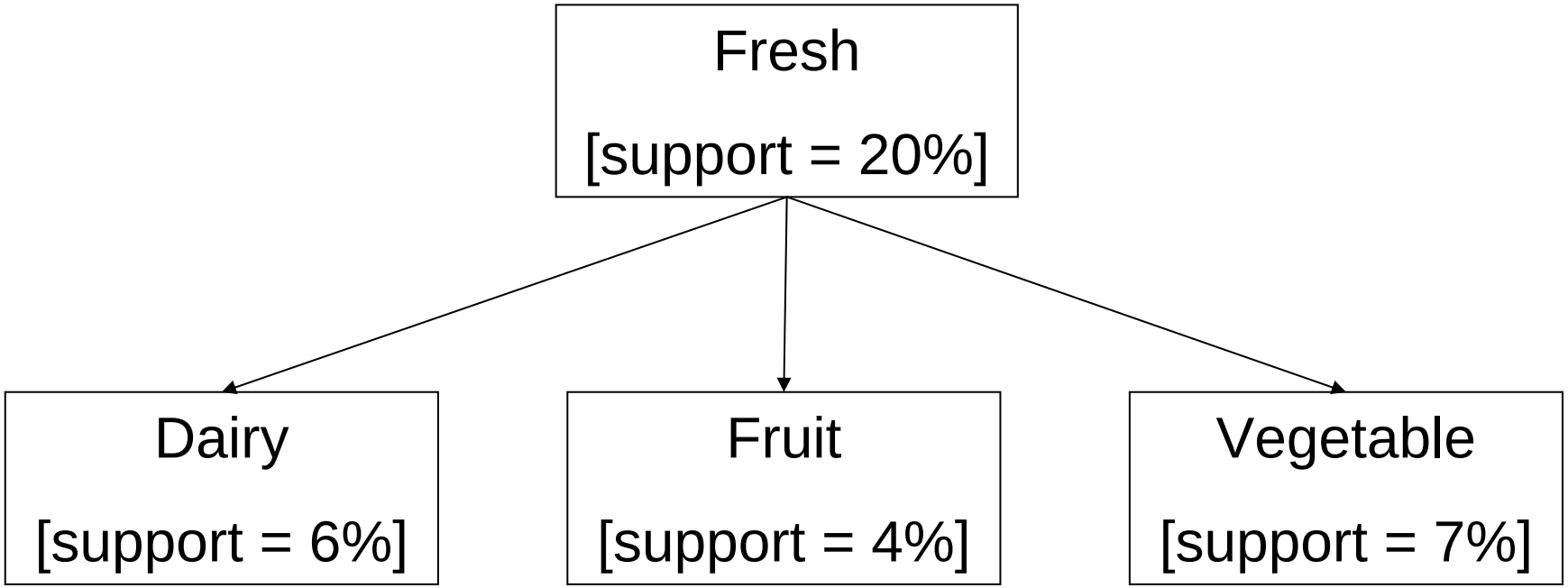
Multilevel AR

- è difficile trovare pattern interessanti
  a un livello troppo primitivo
  - supporto elevato = troppo poche regole
  - supporto basso = troppe regole, per lo più non interessanti
- approccio: ragionare a un adeguato livello di astrazione
- una forma comune di conoscenza di background è
  che un attributo può essere generalizzato o specializzato
  secondo una gerarchia di concetti
- le dimensioni e i livelli possono essere codificati efficientemente nelle transazioni
- Regole Associative Multilivello: combinano
  le associazioni con gerarchie di concetti

Gerarchia di concetti



Multilevel AR



Fresh ⇒ Bakery [20%, 60%]
Dairy ⇒ Bread [6%, 50%]
Fruit ⇒ Bread [1%, 50%] is not valid

Supporto e confidenza di AR Multilevel

- Dal particolare al generale:
  il supporto delle regole aumenta
  (nuove regole possono divenire valide)
- Dal generale al particolare
  il supporto delle regole diminuisce
  (regole possono divenire non valide,
  il loro supporto scende sotto la soglia)
- come viene influenzata la confidenza?
- un po' di pazienza

Ragionare con AR multilivello

- Livello troppo basso ⇒ troppe regole troppo primitive

  Esempio: Mela Melinda ⇒ Colgate dentifricio

  È una curiosità, non un comportamento significativo

- Livello troppo alto ⇒ regole non interessanti

  Esempio: Alimentari ⇒ Generi vari

- Ridondanza ⇒ una regola può essere ridondante a causa di relazioni di “antenato”
  - una regola è ridondante se il suo supporto è vicino al valore atteso in base all'antenato della regola

Ragionare con AR multilivello (ii)

- Esempio (latte ha 4 sottoclassi)
  1. latte $\Rightarrow$ pane [support = 8%, confidence = 70%]
  2. latte magro $\Rightarrow$ pane, [support = 2%, confidence = 72%]
- la regola 1. è antenata della regola 2., in quanto ha come premessa un antenato (nella gerarchia dei concetti) della premessa di 2.
- se i discendenti del latte influenzassero in modo uniforme l'acquisto di pane, dovremmo aspettarci per la regola discendente la stessa confidenza
- in questo caso la regola discendente ha confidenza molto vicina a quella della regola "antenata", quindi può considerarsi ridondante, ovvero non aggiunge nuova conoscenza

Mining AR Multilivello

- Calcola itemset frequenti ad ogni livello di concetti, a partire da quelli più alti, fino a che non sono trovati altri itemset frequenti
- Per ogni livello usare Apriori
- Approccio top-down, un approfondimento successivo
  - trova regole forti ad alto livello:
    fresco $\Rightarrow$ panetteria [20%, 60%].
  - quindi trova regole più deboli a basso livello:
    frutta $\Rightarrow$ pane [6%, 50%].
- variazioni
  - a livelli incrociati:
    frutta $\Rightarrow$ pane nero
  - gerarchie alternative multiple:
    frutta $\Rightarrow$ pane Mulino Bianco

Associazione contro Correlazione

- Esempio
  - Stiamo analizzando transazioni di vendita di materiale elettronico di consumo
  - Su 10.000 transazioni, 6000 contengono giochi da computer, 7500 contengono materiale video
  - Un'analisi di associazione con supporto minimo 30% e confidenza minima 60% mette in evidenza la regola

    compra(X, "giochi computer") $\Rightarrow$
    compra(X, "video") [s=40%,c=66%]

    Sembrerebbe una buona regola, ma un'analisi più attenta mette in evidenza che in realtà l'acquisto di giochi computer ha un'influenza negativa verso l'acquisto di video, che in generale copre il 75% delle transazioni

La correlazione

- Strumento statistico utilizzabile per controllare i risultati della ricerca di regole associative
- Due itemset sono indipendenti se la probabilità della loro unione è uguale al prodotto delle singole probabilità

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

- correlazione >1 significa influenza positiva
- correlazione <1 significa influenza negativa

Tavole di contingenza

	giochi	$\neg$ *giochi*	$\sum$ *righe*
video	4000	3500	7500
$\neg$ *video*	2000	500	2500
$\sum$ *colonne*	6000	4000	10000

Calcolo correlazioni

- Le tavole di contingenza sono la base per il calcolo delle correlazioni
- Le tavole di contingenza sono un diretto prodotto del *data cube* (*OLAP*)
- Un uso estensivo di questo strumento sarebbe troppo oneroso dal punto di vista del calcolo, e darebbe ulteriori problemi di analisi della significatività statistica
- Attualmente si preferisce usare questa tecnica come filtraggio successivo delle regole calcolate con lo strumento di supporto/confidenza

Utilità delle regole associative

- Trova tutte le regole che hanno “noccioline” nel conseguente
  - possono essere usate per capire quali prodotti il supermercato deve comprare per favorire la vendita di noccioline
- Trova tutte le regole che hanno “noccioline” nell'antecedente
  - può prevedere quali prodotti possono subire una riduzione delle vendite se il supermercato decide di non vendere più noccioline
- Trova tutte le regole che hanno “noccioline” nell'antecedente e “birra” nel conseguente
  - può servire per capire quali altri prodotti oltre alle noccioline servono per favorire la vendita di birra
- Trova tutte le regole che riguardano item delle corsie 10 e 11
  - possono essere usate ai fini di una migliore organizzazione dei prodotti nelle corsie
- Trova le regole più “interessanti” per le “noccioline”
  - ad esempio le regole con maggiore confidenza e/o supporto