Lecture 15

Association Rules

Giuseppe Manco

Readings:

Chapter 6, Han and Kamber

Chapter 14, Hastie , Tibshirani and Friedman

Ottimizzazioni

- DHP: Direct Hash and Pruning (Park, Chen and Yu, SIGMOD'95)
  - Ottimizza le prime fasi dell'algoritmo con una strategia look-ahead
    - Efficace per i 2-itemsets e i 3-itemsets
- Partitioning Algorithm (Savasere, Omiecinski and Navathe, VLDB'95)
  - Ottimizza le scansioni del database
    - 2 sole scansioni
  - Simile al boosting
    - Applica Apriori su partizioni che fittano in memoria
    - Ricombina i risultati confrontandoli con il database
- Sampling (Toivonen'96)
  - Randomizzato: applica Apriori su un campione
- Dynamic Itemset Counting (Brin et. al. SIGMOD'97)
  - Estende il concetto di regole associative

Oltre il supporto e la confidenza

- **Esempio**

	coffee	not coffee	sum(row)
tea	20	5	25
not tea	70	5	75
sum(col.)	90	10	100

- **{tea} ⇒ {coffee} ha supporto 20% e confidenza 80%**
- **Il supporto per {coffee} è 90%**
  - **Se una persona acquista tea è meno propenso ad acquistare caffé (del 10%)**
  - **C'è una correlazione negativa tra l'acquisto del tea e l'acquisto del coffee**
  - **Infatti, {~tea} ⇒ {coffee} ha una confidenza più alta (93%)**

Correlazione e interesse

- Due eventi sono indipendenti se $P(A \wedge B) = P(A)*P(B)$
  - Altrimenti sono correlati
- Interesse = $P(A \wedge B) / P(B)*P(A)$
- Esprime la misura della correlazione
  - = 1 $\Rightarrow$ A e B sono indipendenti
  - < 1 $\Rightarrow$ A e B correlati negativamente
  - > 1 $\Rightarrow$ A e B positivamente correlati
- Esempio
  - I(*tea*, *coffee*)=0.89
  - Negativamente correlati

Lift e interesse

- **Lift: misura alternativa all'interesse**
- **Per una regola $A \Rightarrow B$:**
  - **lift = P(B|A) / P(A)**
  - **NB:**
    - **P(A) = supporto relativo di A**
  - **P(B|A) confidenza della regola**

- **Interpretazione:**
  - **lift > 1, A e B correlati positivamente**
  - **lift < 1, correlati negativamente**
  - **lift = 1, independenti**

Estensioni

- Regole multidimensionali
- Regole quantitative
- Regole multilivello

Regole multidimensionali

- **Associazioni tra valori differenti di 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

- **nationality** = French ⇒ **income** = high [50%, 100%]
- **income** = high ⇒ **nationality** = French [50%, 75%]
- **age** = 50 ⇒ **nationality** = Italian [33%, 100%]

Unidimensionali, multidimensionali

- **Unidimensionali (intra-attributo)**
  - Eventi: A, B, C appartengono alla transazione T
  - Occorrenza degli eventi: transazioni
- **Multidimensionali (inter-attributo)**
  - Eventi: attributo A assume valore a, attributo B assume valore b, ecc.
  - Occorrenza degli eventi: tuple

Unidimensionali, multidimensionali

Multidimensionali → **unidimensionali**

<1, Italian, 50, low> → <1, {nat/Ita, age/50, inc/low}>

<2, French, 45, high> → <2, {nat/Fre, age/45, inc/high}>

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

<1, yes, yes, no, no> → <1, {a, b}>

<2, yes, no, yes, no> → <2, {a, c}>

Attributi numerici

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: discretizzazione**
  - **[Age: 30..39] and [Married: Yes]** ⇒ **[NumCars:2]** (s = 40%,c = 100%)

Attributi numerici e discretizzazione

- **Due alternative**
  1. **Discretizzazione statica**
  2. **Discretizzazione dinamica**
     - ⌘ **Effettuata dall'algoritmo**
     - ⌘ **Interazione stretta tra Apriori e il discretizzatore**

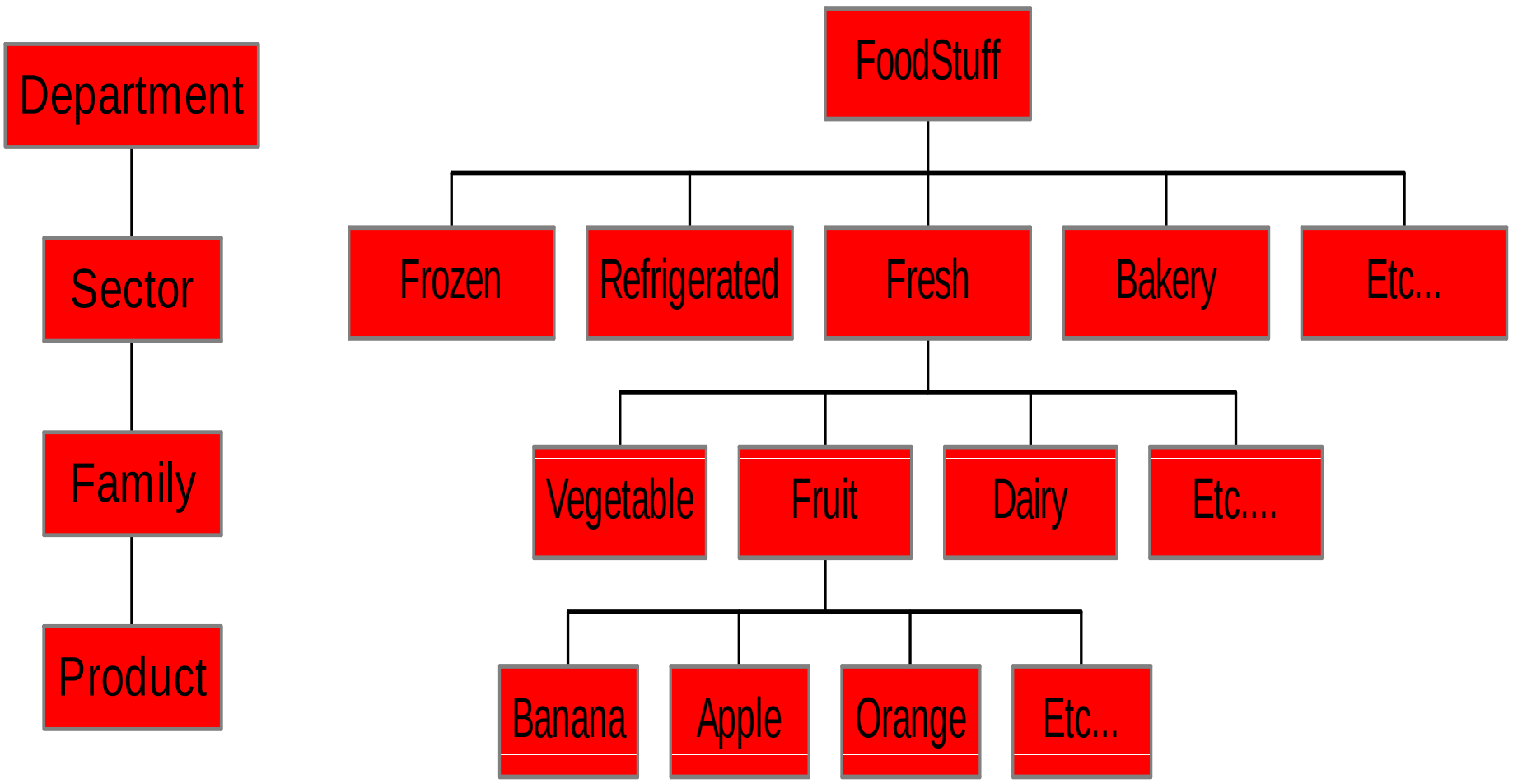
Regole e vincoli

- **Due tipi:**
  - **Di forma (Meta-regole)**
    - **P(x, y) ^ Q(x, w) → takes(x, "database systems").**
  - **Di Contenuto**
    - **sum(LHS) < 100 & min(LHS) > 20 & sum(RHS) > 1000**
- **Approcci**
  - **Genera le regole, seleziona quelle che soddisfano i vincoli**
  - **Constraint-pushing**
    - **Riprogetta l'algoritmo Apriori**

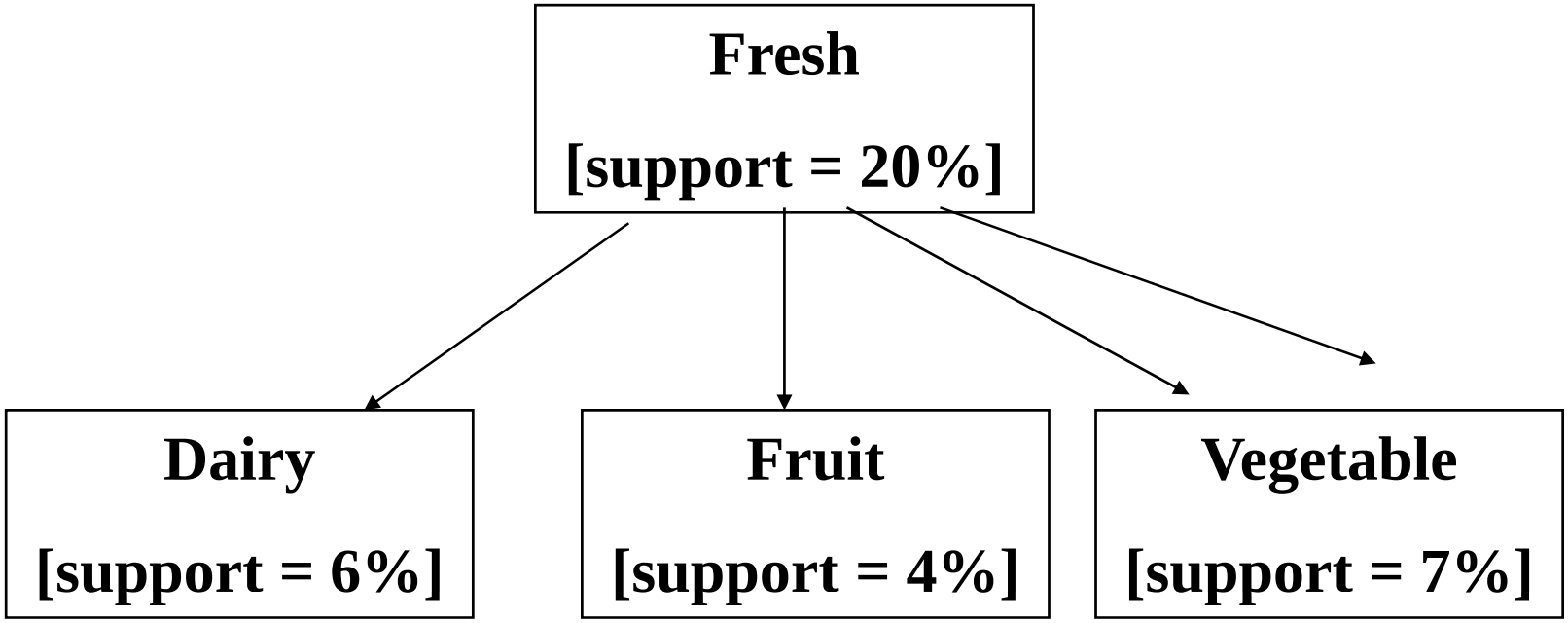
Regole multilivello

- **Regole su più livelli**
  - Prodotto, marca contro prodotto
    - Pasta barilla -> pasta
  - I livelli primitivi sono poco informativi
    - Supporto alto = poche regole
    - Supporto basso=troppe regole
- **Approccio**
  - Ragioniamo a più livelli di astrazione
  - Background knowledge: gerarchia di concetti
- **Regole multilivello**
  - Combinano le associazioni con gerarchie di concetto

Gerarchia di concetto



Multilevel AR



Fresh ⇒ Bakery [20%, 60%]
Dairy ⇒ Bread [6%, 50%]

Oltre l'Apriori

- Forza: generate-and-test
  - Concettualmente semplice
  - Adeguato per dataset sparsi
    - transazioni
- debolezza: generate-and-test!!!
  - Insiemi densi
  - La fase di generazione porta a troppi candidati
    - 10^4 1-itemsets frequenti generano 10^7 2-itemsets candidati
    - Per scoprire i patterns di lunghezza100
      - esempio: $\{a_1, a_2, \ldots, a_{100}\}$
    - Dobbiamo generare $2^{100} \approx 10^{30}$ candidati

FP-Growth: evitare la generazione di candidati

- **Proiezione e compressione del database**
  - **Proiettiamo il database sui suoi patterns frequenti**
  - **Comprimiamo il database in una struttura compatta**
    - **Frequent-Pattern tree (FP-tree)**
    - **Rappresentazione condensata ma completa**
    - **Non abbiamo bisogno di generare candidati**

FP-Tree

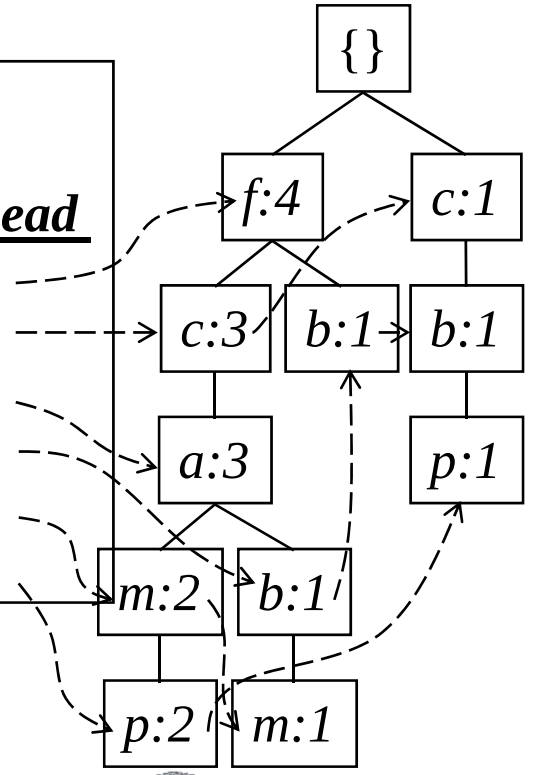
TID	*Items*	*items frequenti*
100	{*f, a, c, d, g, i, m, p*}	{*f, c, a, m, p*}
200	{*a, b, c, f, l, m, o*}	{*f, c, a, b, m*}
300	{*b, f, h, j, o, w*}	{*f, b*}
400	{*b, c, k, s, p*}	{*c, b, p*}
500	{*a, f, c, e, l, p, m, n*}	{*f, c, a, m, p*}

min_support = 0.5

1. Troviamo gli 1-itemsets frequenti
2. Ordiniamoli per ordine discendente
3. Costruiamo l'FP-Tree dal database

Header Table

Item	*Frequenza*	*head*
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

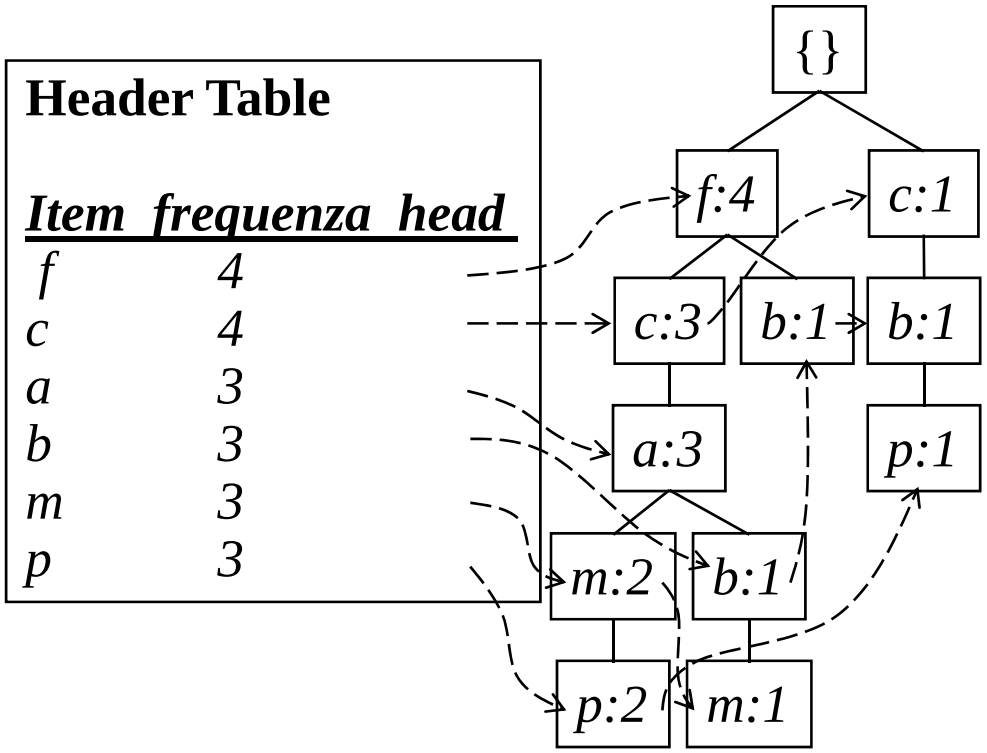


Utilizzare l'FP-Tree

- Idea: Frequent pattern growth
  - Sviluppa ricorsivamente i patterns frequenti tramite partizionamento
- Metodo
  - Per ogni item frequente, costruisci la conditional pattern-base, e il conditional FP-tree
    - Ripeti ricorsivamente il processo sul nuovo FP-tree condizionale
  - Se l'FP-tree risultante è vuoto o contiene un solo cammino
    - Tutti i sottoinsiemi del cammino sono itemsets frequenti

Conditional pattern base

- Parti dalla header table
- Attraversa l' FP-tree seguento i links che collegano il pattern *p*
- Accumula tutti i prefissi di p



***Conditional* pattern bases**

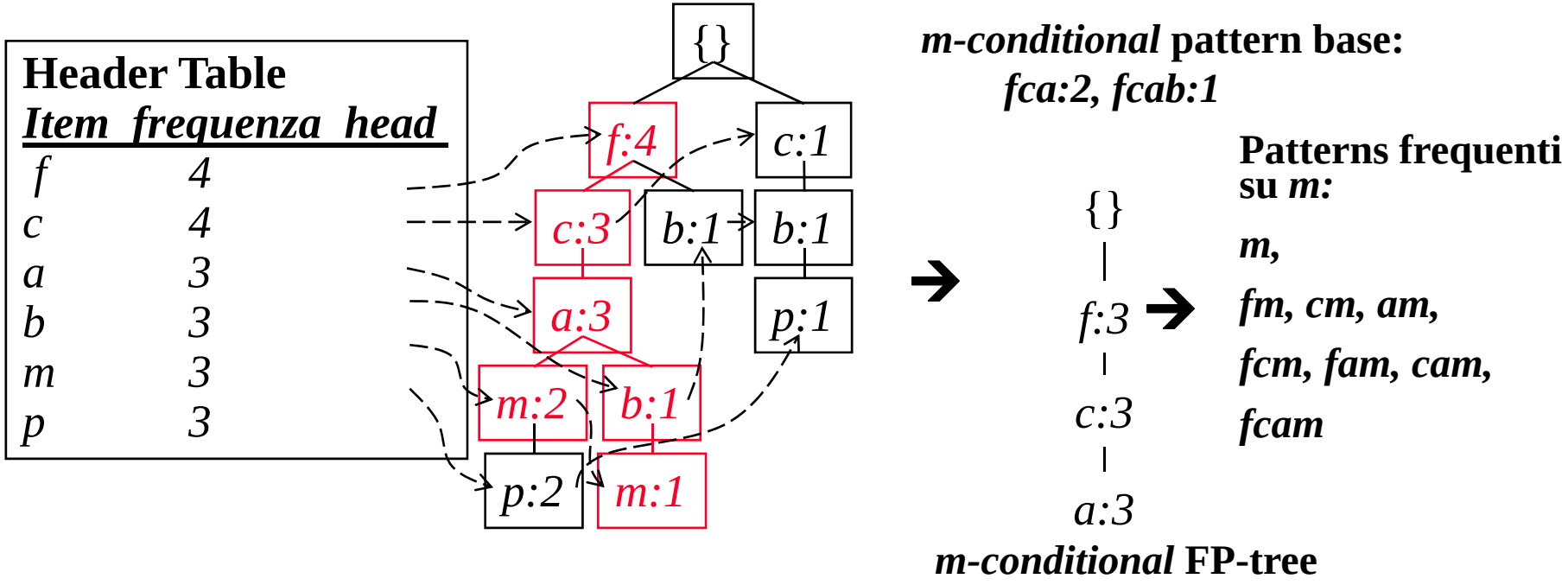
item	*cond. pattern base*
c	***f:3***
a	***fc:3***
b	***fca:1, f:1, c:1***
m	***fca:2, fcab:1***
p	***fcam:2, cb:1***

I conditional pattern bases sono sufficienti

- **Completezza**
  - Per ogni item frequente *p*, tutti i possibili itemsets frequenti che contengono p possono essere ottenuti dai nodi puntati da p nell'header
- **Correttezza**
  - Basta considerare i prefissi e associare ai prefissi la frequenza del nodo contenente p

FP-tree condizionali

- **Per ogni pattern-base**
  - Accumula il count per ogni itemAccumulate the count for each item in the base
  - Costruisci l'FP-tree per gli items frequenti nel pattern-base



FP-tree condizionali

Item	Conditional pattern-base	FP-tree condizionale
p	{(fcam:2), (cb:1)}	{(c:3)}|p
m	{(fca:2), (fcab:1)}	{(f:3, c:3, a:3)}|m
b	{(fca:1), (f:1), (c:1)}	Empty
a	{(fc:3)}	{(f:3, c:3)}|a
c	{(f:3)}	{(f:3)}|c
f	∅	∅

Ricorsione

Cond. pattern base di “am”: (fc:3)

***am-conditional* FP-tree**

Cond. pattern base di “cm”: (f:3)

***cm-conditional* FP-tree**

***m-conditional* FP-tree**

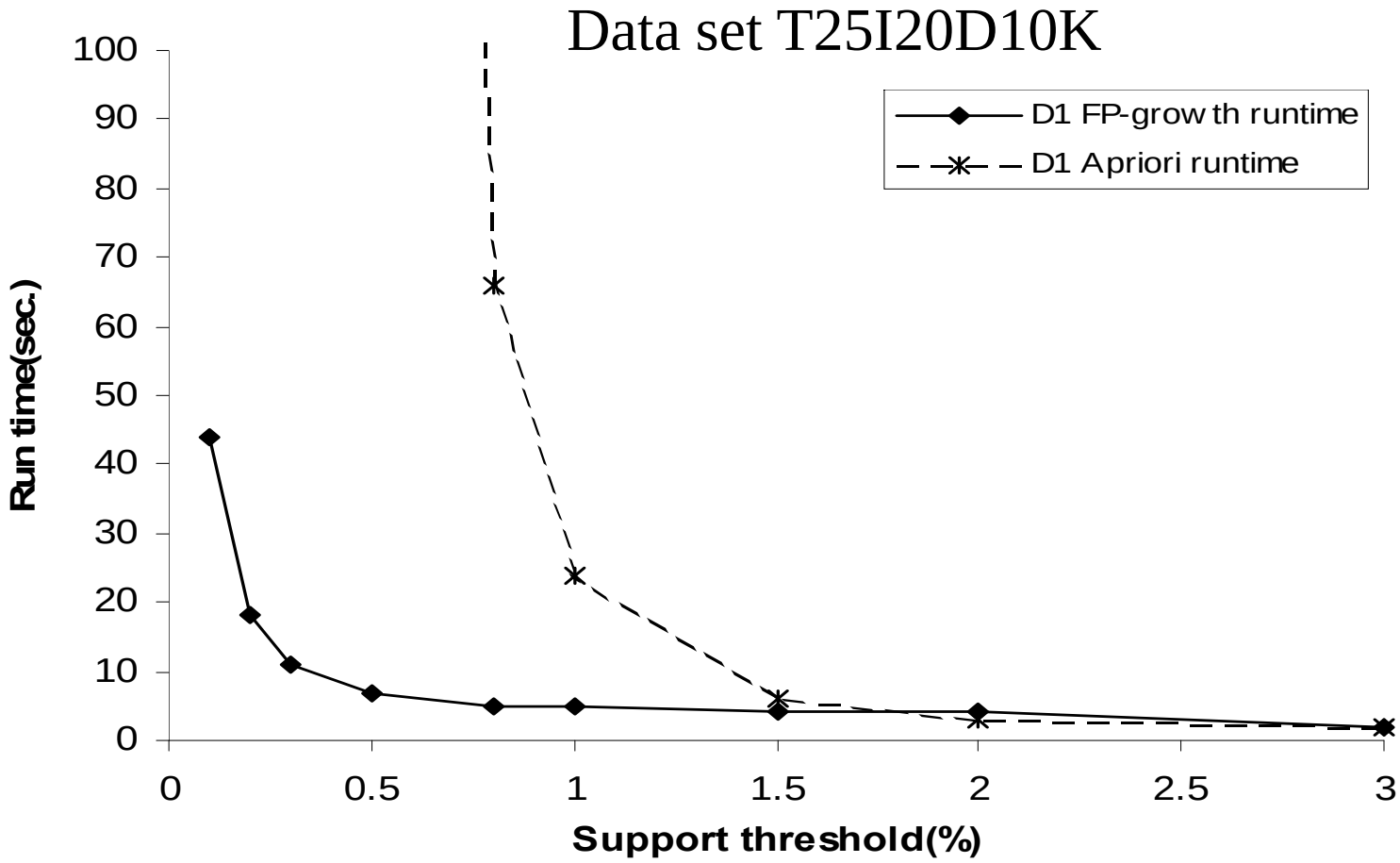
Cond. pattern base di “cam”: (f:3)

***cam-conditional* FP-tree**

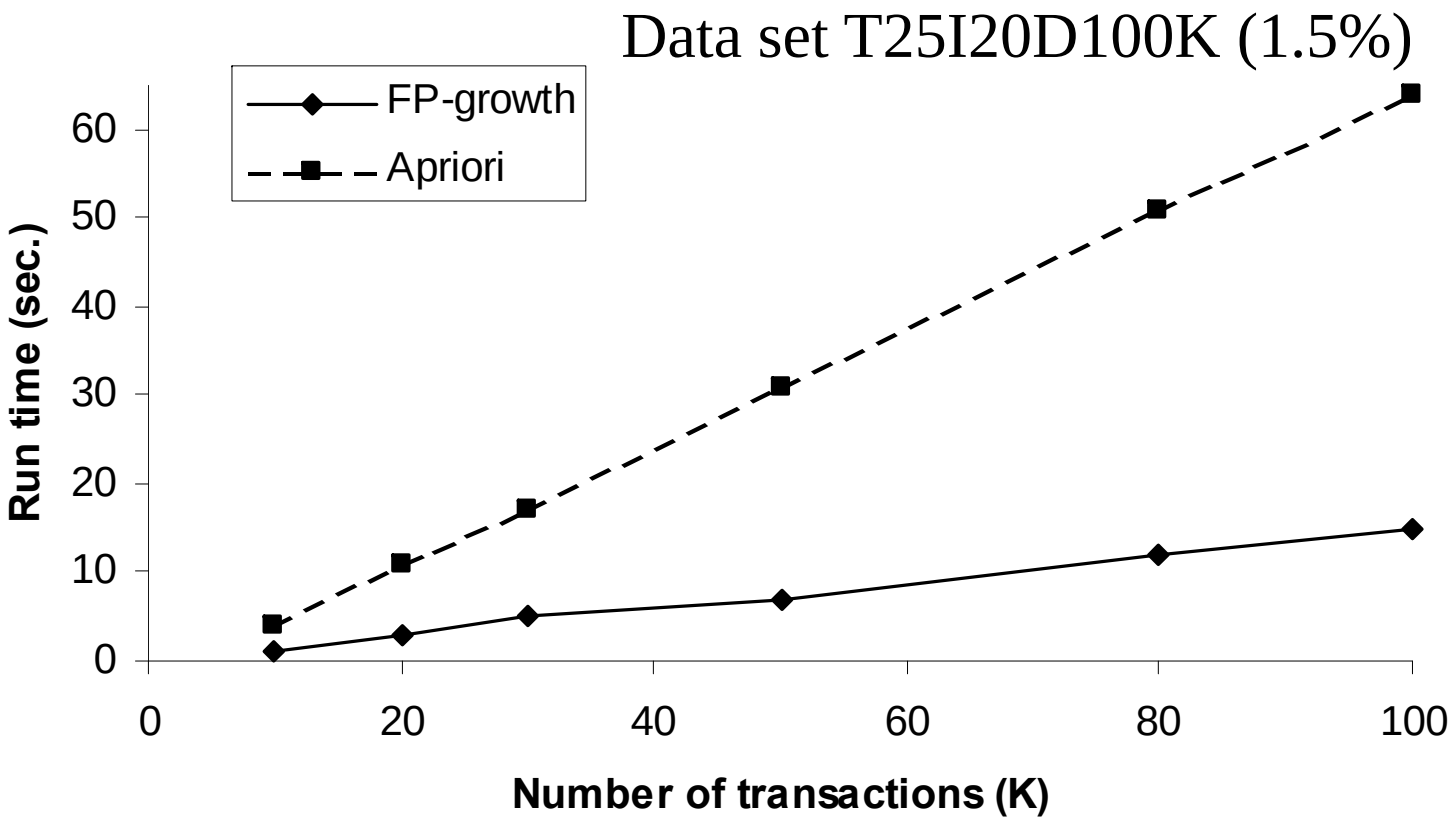
Correttezza

- **Teorema**
  - **Se α è frequente in D, B è il conditional pattern base di α, e β è un itemset di B**
    - **allora $\alpha \cup \beta$ è frequente in D se e solo se β è frequente in B.**
- **Esempio: "*abcdef* " è frequente se e solo se**
  - **"*abcd* " è frequente**
  - **"*ef* " è frequente nel conditional pattern-base di *abcd*'s**

FP-Growth vs. Apriori [1]



FP-Growth vs. Apriori [2]



Perché FP-Growth è più efficiente?

- **Dataset densi**
  - Evita la generazione di candidati
  - Su dataset densi, la proiezione del dataset sull'FP-tree è molto vantaggiosa
    - Compressione alta
  - Non c'è bisogno di fare molti scan
- **Dove funziona**
  - Dati biologici
  - Regole multidimensionali
    - Regole categoriche
- **Dove non funziona**
  - Sessioni Web
  - Dataset sparsi in generale
    - La compressione sull'FP-Tree è molto bassa

Associazioni e Weka

- Il formato ARFF
  - Estremamente inefficiente
  - Gli attributi rappresentano gli items
    - Rappresentazione sparsa
- Progetto
  - Estendiamo Weka per supportare il formato transazionale
  - Implementiamo FP-Growth

Estensioni

- **Ridondanza**
  - Relazione di inclusione: dati $A \subseteq B$ frequenti, quando è utile mantenere entrambi
- **Itemsets massimali**
  - A supporti bassi troppi itemsets frequenti
    - Gli itemsets sottoinsiemi di altri itemsets sono ridondanti
- **Itemsets chiusi**
  - Se supp(A) = supp(B)
    - A è ridondante
  - A è chiuso se
    - Non esiste B per cui supp(A) = supp(B)
  - Test del Chi-quadro per valutare la differenza di supporto
    - supp(A) ≠ supp(B) a meno del test del chi-quadro
- **Regole associative per la classificazione**
  - Conseguente fissato
  - Accuratezza predittiva