

TROBLMI APRIORI

Apriori genera e testa gli itemsets. Il problema è che per generare gli itemset candidati ci vuole un grosso dispendio computazionale.

Per generare frequent pattern di dimensione 100, bisogna avere minimi $2^{100} - 1$ candidati. Troppi...

FP-GROWTH

Approccio più veloce, bastano solo due passi:

- 1) Crea una struttura dati detta FP-Tree
- 2) Ottieni i frequent itemset dal FP-Tree

COSTRUZIONE DEL FP-TREE

- 1) Scannerizzare il dataset e calcolare il supporto di ogni singolo oggetto.

Non considerare gli oggetti non frequenti.

Ordinare gli oggetti rimasti per ordine decrescente di supporto, così da condividere fra i vari nodi gli oggetti più frequenti.

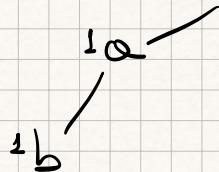
- 2) Mappare ogni transazione nell'albero, una per volta. Quando le transazioni si sovrappongono per certi oggetti, incrementare l'adeguato contatore. È utile mantenere un puntatore fra diverse istanze dello stesso oggetto nell'albero.

Esempio

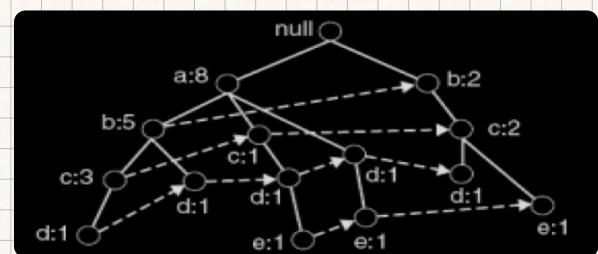
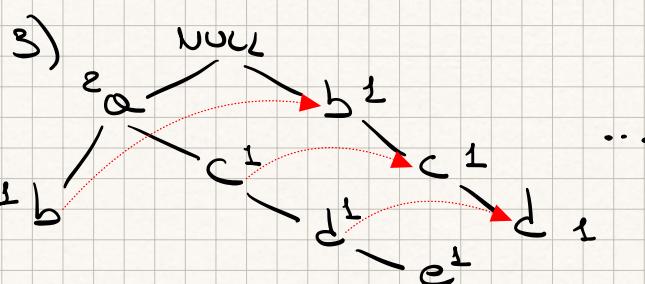
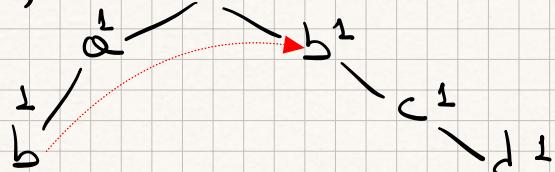
TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}

Vogliamo i supporti: a: 8, b: 7, c: 5, d: 5, e: 3.

1) null



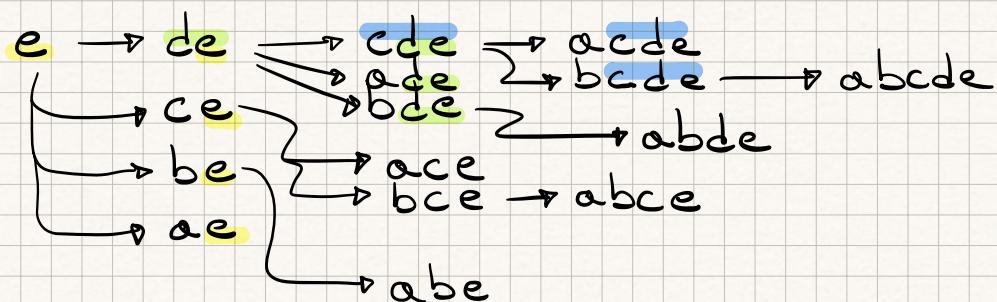
2) null



GENERAZIONE DEI FREQUENT PATTERN

Partendo dalle foglie per poi salire, utilizzando un approccio divide e conquista:

Per esempio, partendo da "e" nell'esempio di prima, elenchiamo prima tutti gli itemset frequenti che finiscono per "e" poi quelli che finiscono in "de", ecc...



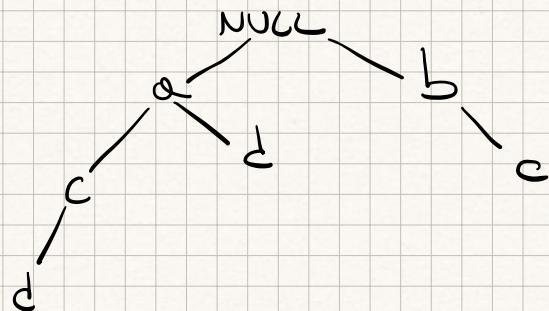
Da notare che gli oggetti dentro un itemset sono sempre in ordine di supporto. Non posso mai trovare, ad esempio "d bce", perché $\text{supp}(d) = 5$, mentre quello $\text{supp}(b) = 7$

CONDITIONAL FP-TREE

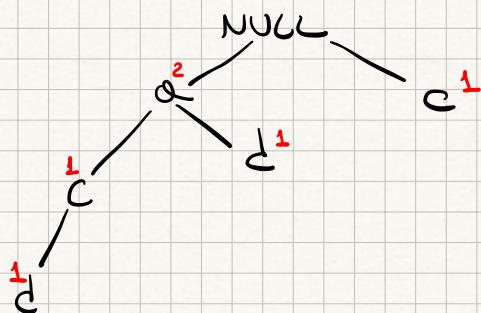
È un FP-Tree costituito tenendo conto di sole alcune transazioni, contenenti determinati oggetti saltati a priori:

TID	Items
1	{a,b}
2	{b,c,d}
3	{a,c,d,e}
4	{a,d,e}
5	{a,b,c}
6	{a,b,c,d}
7	{a}
8	{a,b,c}
9	{a,b,d}
10	{b,c,e}

Costruiamo il conditional FP-Tree per le transazioni contenenti 'e'.



Nelle slide fa un albero diverso:



Non considera la 'b', in 'bce', crede perché 'b' ha un supporto sotto al threshold = 2.

Osservio Libro

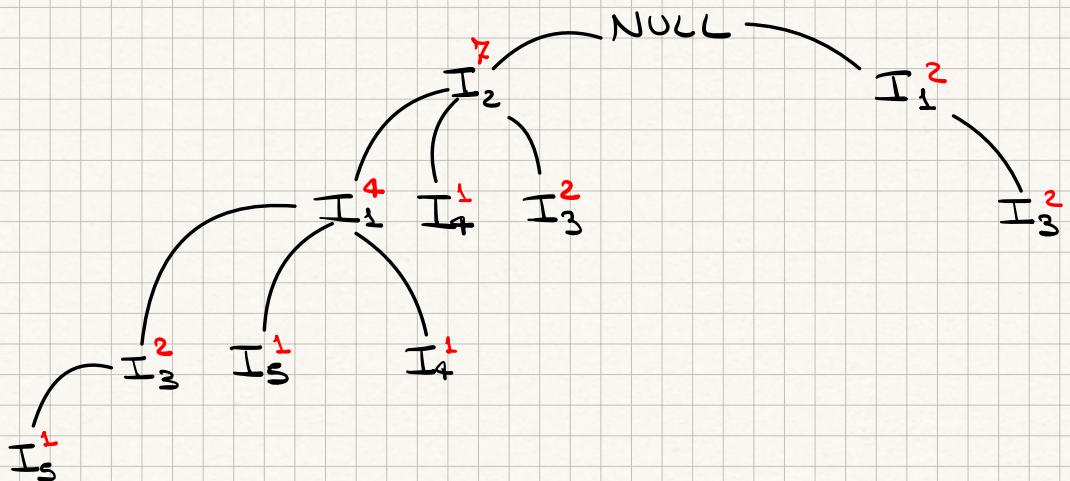
TID	List of item IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

1) Troviamo i set di 1-itemset frequenti considerando $\text{min_supp} = 2$.

$$\begin{array}{l} \{I_2\}: 7 \\ \{I_4\}: 2 \end{array} \quad \begin{array}{l} \{I_1\}: 6 \\ \{I_5\}: 2 \end{array} \quad \begin{array}{l} \{I_3\}: 6 \\ \{I_3\}: 2 \end{array}$$

// Primo scan di D.

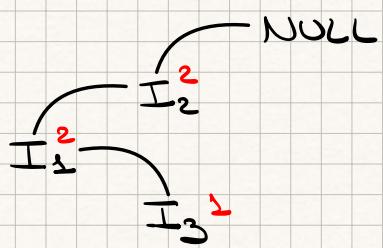
Costruiamo ora il FP-Tree: // Secondo scan di D



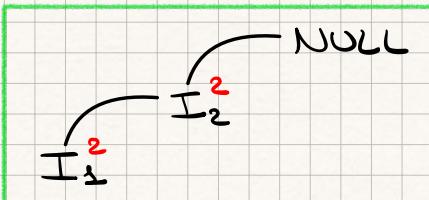
Ora bisogna costruire i frequent pattern, per farlo bisogna costruire i conditional FP-Tree portando da I_5 . I_5 appare in due ramo diversi dell'albero:

$$\begin{array}{l} \{I_2, I_1, I_5\} \\ \{I_2, I_1, I_3, I_5\} \end{array}$$

Considerando solo questi due, e in particolare i prefissi rispetto a I_5 ($\{I_2, I_1\}$ e $\{I_2, I_1, I_3\}$) costruiamo il conditional FP-Tree per il suffisso I_5 :



Questo verrebbe senza considerare il min supp = 2 che ci costringe a non includere I_3 che ha supp = 1



Questo single path FP-tree genera le seguenti 3 combinazioni di frequent pattern:

$\{I_2, I_5\}$ | $\{I_3, I_5\}$ | $\{I_2, I_3, I_5\}$

Ripetendo questo procedimento per gli altri suffissi arriviamo al seguente risultato:

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	$\{\{I_2, I_1: 1\}, \{I_2, I_1, I_3: 1\}\}$	$\langle I_2: 2, I_1: 2 \rangle$	$\{I_2, I_5: 2\}, \{I_1, I_5: 2\}, \{I_2, I_1, I_5: 2\}$
I4	$\{\{I_2, I_1: 1\}, \{I_2: 1\}\}$	$\langle I_2: 2 \rangle$	$\{I_2, I_4: 2\}$
I3	$\{\{I_2, I_1: 2\}, \{I_2: 2\}, \{I_1: 2\}\}$	$\langle I_2: 4, I_1: 2 \rangle, \langle I_1: 2 \rangle$	$\{I_2, I_3: 4\}, \{I_1, I_3: 4\}, \{I_2, I_1, I_3: 2\}$
I1	$\{\{I_2: 4\}\}$	$\langle I_2: 4 \rangle$	$\{I_2, I_1: 4\}$

PRO DI FP-GROWTH

- 1) Riduce sensibilmente il costo di ricerca degli itemset frequenti, riducendolo a una ricerca concatenando i suffissi di portanza dell'albero.
- 2) Bisogna scannerizzare il DB solo due volte.
Una prima volta per calcolare i supporti e ordinare gli item per supporto (Non ordinarli del DB direttamente, ma in ordine generale)
Una seconda volta per costruire l'FP-Tree.
- 3) FP-Tree di fatti è un modo per comprimere il DB
- 4) Non genera candidati ricorsivamente, solo un albero.

CONTRE DI FP-GROWTH

- 1) L'FP-Tree potrebbe non entrare nella memoria RAM
- 2) La costruzione di FP-Tree è comunque costosa.

Per grossi DB è irrealistico costruire un FP-Tree che sta in RAM. Una soluzione può essere quella di usare l'algoritmo per ogni proiezione del DB precedentemente diviso in più parti.

VERTICAL DATA FORMAT

Il concetto è rappresentare i dati in modo invertito, o verticale, elencando per ogni item l'elenco delle transazioni che lo contengono.

$$X : T_1, T_2, T_5, T_{10}$$

da qui le proprietà di questa notazione:

- 1) $t(X) = t(Y)$: X e Y appaiono sempre insieme
- 2) $t(X) \subset t(Y)$: X appare sempre con Y , e non il contrario.

DIFFSET

δ il set che contiene le transazioni non in comune fra gli item sotto analisi:

$$t(X) = \{T_1, T_2, T_3\}$$

$$t(Y) = \{T_1, T_3\}$$

$$\text{DIFFSET}(X, Y) = \{T_2\}$$

Esempio:

itemset	TID_set	itemset	TID_set
I1	{T100, T400, T500, T700, T800, T900}	{I1, I2}	{T100, T400, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}	{I1, I3}	{T500, T700, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}	{I1, I4}	{T400}
I4	{T200, T400}	{I1, I5}	{T100, T800}
I5	{T100, T800}	{I2, I3}	{T300, T600, T800, T900}
		{I2, I4}	{T200, T400}
		{I2, I5}	{T100, T800}
		{I3, I5}	{T800}

3-Itemsets in Vertical Data Format	
itemset	TID_set
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

CORRELATIONS LIFT

Il LIFT misura quante volte in più, rispetto a ciò che si ci aspetta, X e Y si verificano insieme, considerando X e Y statisticamente indipendenti:

$$C_{lift}(X \rightarrow Y) = C_{lift}(Y \rightarrow X) = \frac{Conf(X \rightarrow Y)}{Supp(Y)} = \frac{Conf(Y \rightarrow X)}{Supp(X)}$$
$$= \frac{P(X \cup Y)}{P(X)P(Y)}$$

ALL CONFIDANCES $[\emptyset, 1]$

$$\text{all.conf}(A, B) = \frac{\sup(A \cup B)}{\max\{\sup(A), \sup(B)\}} = \min\{P(A|B), P(B|A)\}$$

MAX CONFIDANCES $[\emptyset, 1]$

$$\text{max-conf}(A, B) = \max\{P(A|B), P(B|A)\}$$

KULCZYNSKI $[\emptyset, 1]$

$$\text{Kulc}(A, B) = \frac{1}{2} P(A|B) + \frac{1}{2} P(B|A)$$

COSINE $[\emptyset, 1]$

$$\text{cosine}(A, B) = \sqrt{P(B|A) \cdot P(A|B)}$$

Analizziamo varie situazioni basandoci sulla tabella che segue

2 × 2 Contingency Table for Two Items

	$milk$	\overline{milk}	Σ_{row}
$coffee$	mc	\overline{mc}	c
\overline{coffee}	$m\bar{c}$	$\overline{m\bar{c}}$	\bar{c}
Σ_{col}	m	\overline{m}	Σ

1)

$$NMC = 10\ 000$$

$$\bar{N}C = 1000$$

$$NC = 1000$$

$$\bar{NC} = 100\ 000$$

 \Rightarrow

$$\chi^2 = 90557$$

$$Lift = 0.91$$

$$all_conf = 0.91$$

$$max_conf = 0.91$$

$$Kulczynski = 0.91$$

$$Cosine = 0.91$$

$$\chi^2 = \emptyset$$

$$NMC = 10\ 000$$

$$\bar{N}C = 1000$$

$$NC = 1000$$

$$\bar{NC} = 100$$

 \Rightarrow

$$Lift = 1$$

$$all_conf = 0.91$$

$$max_conf = 0.91$$

$$Kulczynski = 0.91$$

$$Cosine = 0.91$$

Nel secondo caso abbiamo diminuito sensibilmente i valori delle TRANSAZIONI NUCLEUS, che condizionano sia χ^2 che Lift.

$$\sum_i^n \sum_j^m \frac{O_{ij} - E_{ij}}{E_{ij}} \Rightarrow E_{ij} = \frac{Count(i) \times Count(j)}{n}$$

$$\frac{NMC - \frac{N \cdot C}{n}}{\frac{N \cdot C}{n}} + \frac{\bar{N}C - \frac{\bar{N} \cdot C}{n}}{\frac{\bar{N} \cdot C}{n}} + \frac{NC - \frac{N \cdot \bar{C}}{n}}{\frac{N \cdot \bar{C}}{n}} + \frac{\bar{NC} - \frac{\bar{N} \cdot \bar{C}}{n}}{\frac{\bar{N} \cdot \bar{C}}{n}}$$

Abbiamo sempre il totale di transazioni n che condiziona il calcolo.

Vale lo stesso per il Lift

$$\text{Lift}(A \Rightarrow B) = \frac{P(A \cup B)}{P(A) \cdot P(B)} = \frac{\frac{N_{AB}}{N}}{\frac{N_A}{N} \cdot \frac{N_B}{N}} = \frac{N_{AB}}{\frac{N_A N_B}{N}}$$

Per quanto riguarda Kulczyński:

$$\begin{aligned} \frac{1}{2} P(A|B) + \frac{1}{2} P(B|A) &= \frac{1}{2} \left(\frac{P(AB)}{P(B)} + \frac{P(BA)}{P(A)} \right) \\ &= \frac{1}{2} \left(\frac{\frac{N_{AB}}{N}}{\frac{N_B}{N}} + \frac{\frac{N_{BA}}{N}}{\frac{N_A}{N}} \right) = \frac{1}{2} \left(\frac{N_{AB}}{N_B} + \frac{N_{BA}}{N_A} \right) \end{aligned}$$

Non abbiamo più n a condizionare il calcolo.