

Progetto di basi di dati

AREA ANALYTICS

Abitudini degli utenti tramite Association rule learning

Implementazione dell'algoritmo **Apriori**:

Nel file APRIORI.SQL, ci sono tutti i passaggi dell'algoritmo commentati in blocchi.

Le tabelle che servono per raccogliere i dati di transazioni e items, sono rispettivamente UTILIZZO e CANDIDATE1 create e popolate all'inizio dall'utente.

La stored procedure *APRIORI (IN supp double, IN conf double)* che prende in ingresso due double, che rappresentano le soglie di Supporto e di Confidenza da usare più avanti nel processo dell'algoritmo per ottenere le regole di associazione più forti. Evidenziamo adesso passo dopo passo con degli screenshots:

	Transactions	ItemSets
▶	1	device1
	2	device1, device3
	3	device3, device1, device4
	4	device1, device2
	5	device3, device4
	6	device3, device5, device1
	7	device2
	8	device3, device1
	9	device3, device5, device4
	10	device4, device3
	11	device2, device5
	12	device4, device5, device1
*	NULL	NULL

Inseriamo all'interno della tabella UTILIZZO, tutte le transazioni.

	Items	Supporto
▶	device1	7
	device2	3
	device3	7
	device4	5
	device5	4
*	NULL	NULL

inseriamo nella tabella CANDIDATE1 l'elenco degli Items che costituiscono le transazioni.

$I = \{ \text{device1, device2, device3, device4, device5} \}$

PASSO 1: $L_{k-1}; k = 2, \dots |I| \quad L_1$

```
1 select *
2 from candidate1;
```

Items	Supporto
device1	7
device2	3
device3	7
device4	5
device5	4
NULL	NULL

dall'insieme C_1 a sinistra, ottengo l'insieme di large itemset L_1 a destra, selezionando quelli items che hanno valore di supporto maggiore o pari al valore della soglia prestabilita all'inizio dell'algoritmo.



```
1 select *
2 from large1;
```

Items
device1
device2
device3
device4
device5
NULL

PASSO 2: $C_2 = L_1 \text{ CROSS JOIN } L_1$

```
1 select *
2 from large1;
```

Items
device1
device2
device3
device4
device5
NULL

Effettuo il cross join per ottenere tutte le combinazioni possibili tra gli items rimasti in L_1 .

Elimino i record che contengono più di una volta lo stesso item.

C_2 a destra .



```
1 select *
2 from candidate2;
```

Items1	Items2	Supporto
device1	device2	1
device1	device3	4
device1	device4	2
device1	device5	2
device2	device1	1
device2	device3	0
device2	device4	0
device2	device5	1
device3	device1	4
device3	device2	0
device3	device4	4
device3	device5	2
device4	device1	2
device4	device2	0
device4	device3	4
device4	device5	2
device5	device1	2
device5	device2	1
device5	device3	2
device5	device4	2
NULL	NULL	NULL

PASSO 3 & 4

Pruning : da C_2 selezioniamo gli itemset che hanno supporto maggiore o pari al valore del Supporto prestabilito per ottenere l'insieme L_2 di 2-itemsets.

Trasferisco in L_2 l'insieme di record rimasto in C_2 .

```
1 • select *
2   from Large2;
```

Result Grid | Filter Rows:

Items1	Items2
device1	device3
device1	device4
device1	device5
device3	device1
device3	device4
device3	device5
device4	device1
device4	device3
device4	device5
device5	device1
device5	device3
device5	device4
NULL	NULL

Poi si ripete dal passo1, finché non si ottiene un insieme L_k vuoto.

Effettuando diversi test di questo algoritmo con, abbiamo notato che:

- Diminuire la lunghezza massima (in termini di numero di items che costituiscono la regola) delle regole associative permette di ridurre il build time, di generare anche delle regole più semplici.
- Aumentare la soglia minima di supporto, ci permette di ridurre il build time, e di generare poche regole.
- Quando noi diamo un dataset piccolo, l'algoritmo potrebbe generare delle false regole, o meglio regole che non sono sempre vere, non capitano molto spesso.
- Per quanto riguarda lo studio delle abitudini degli utenti della smarthome, le possibilità sono tante:
 - Regole che stabiliscono quale gruppo di device uso molto spesso
 - Le impostazioni più usate su un dispositivo dato
 - Momento della giornata e quali device tende ad usare di più.

TEST: CALL APRIORI(1, 50)

Prima riga:

device1 -> device3, device4 14,28

InvConfidenza rappresenta la confidenza della regola inversa a quella della riga.

device3, device4 -> device1 25

ID	Antecedente1	Conseguente1	Conseguente2	Confidenza	InvConfidenza
1	device1	device3	device4	14.285714285714285	25
2	device1	device3	device5	14.285714285714285	50
3	device1	device4	device3	14.285714285714285	25
4	device1	device4	device5	14.285714285714285	50
5	device1	device5	device3	14.285714285714285	50
6	device1	device5	device4	14.285714285714285	50
7	device3	device1	device4	14.285714285714285	50
8	device3	device1	device5	14.285714285714285	50
9	device3	device4	device1	14.285714285714285	50
10	device3	device4	device5	14.285714285714285	50
11	device3	device5	device1	14.285714285714285	50
12	device3	device5	device4	14.285714285714285	50
13	device4	device1	device3	20	25
14	device4	device1	device5	20	50
15	device4	device3	device1	20	25
16	device4	device3	device5	20	50
17	device4	device5	device1	20	50
18	device4	device5	device3	20	50
19	device5	device1	device3	25	25
20	device5	device1	device4	25	50
21	device5	device3	device1	25	25
22	device5	device3	device4	25	25
23	device5	device4	device1	25	50
24	device5	device4	device3	25	25

TEST: CALL APRIORI (2, 50)

Items	Supporto
device1	7
device2	3
device3	7
device4	5
device5	4
NULL	NULL

C₁



Items
device1
device2
device3
device4
device5
NULL

L₁



Items1	Items2	Supporto
device1	device2	1
device1	device3	4
device1	device4	2
device1	device5	2
device2	device1	1
device2	device3	0
device2	device4	0
device2	device5	1
device3	device1	4
device3	device2	0
device3	device4	4
device3	device5	2
device4	device1	2
device4	device2	0
device4	device3	4
device4	device5	2
device5	device1	2
device5	device2	1
device5	device3	2
device5	device4	2
NULL	NULL	NULL

C₂



Items1	Items2
device1	device3
device1	device4
device1	device5
device3	device1
device3	device4
device3	device5
device4	device1
device4	device3
device4	device5
device5	device1
device5	device3
device5	device4
NULL	NULL

L₂

```

1 • select *
2   from candidate3;

```

C₃

Items1	Items2	Items3	Supporto
device1	device3	device4	1
device1	device3	device5	1
device1	device4	device3	1
device1	device4	device5	1
device1	device5	device3	1
device1	device5	device4	1
device3	device1	device4	1
device3	device1	device5	1
device3	device4	device1	1
device3	device4	device5	1
device3	device5	device1	1
device3	device5	device4	1
device4	device1	device3	1
device4	device1	device5	1
device4	device3	device1	1
device4	device3	device5	1
device4	device5	device1	1

candidate3 2



A questo punto, **L₃** risulta vuoto perché nessun record ha supporto $\geq \text{MinSupp}$ quindi l'algoritmo si ferma, e considera **L₂** come large itemsets da qui pescare per generare tutte le regole associative.

```

1 • call apriori(2, 50);
2 • select *
3   from ARules2;
4   -- where Confidenza > __MinConf;
5

```

ID	Antecedente	Conseguente	Confidenza
1	device1	device3	57.14285714285714
2	device1	device4	28.57142857142857
3	device1	device5	28.57142857142857
4	device3	device1	57.14285714285714
5	device3	device4	57.14285714285714
6	device3	device5	28.57142857142857
7	device4	device1	40
8	device4	device3	80
9	device4	device5	40
10	device5	device1	50
11	device5	device3	50
12	device5	device4	50
NULL	NULL	NULL	NULL

Result 3 **ARules2 4**

TEST: CALL APRIORI (3, 50)

in questo caso, sono già stati selezionate le regole con Confidenza > MinConf

device1 -> device3

device4 -> device3: regole che si possono verificare andando a controllare la tabella delle transazioni. In particolare, si nota che nelle transazioni 3, 5, 9 e 10 device3 e device4 compaiono entrambi.

ID	Antecedente	Conseguente	Confidenza
1	device1	device3	57.14285714285714
2	device3	device1	57.14285714285714
3	device3	device4	57.14285714285714
4	device4	device3	80

	Transactions	ItemSets
▶	1	device 1
	2	device 1, device 3
	3	device 3, device 1, device 4
	4	device 1, device 2
	5	device 3, device 4
	6	device 3, device 5, device 1
	7	device 2
	8	device 3, device 1
	9	device 3, device 5, device 4
	10	device 4, device 3
	11	device 2, device 5
	12	device 4, device 5, device 1
•	NULL	NULL

TEST: CALL APRIORI (7, 50)

Caso particolare in cui, L_2 è vuoto perché tra i record di C_2 nessun ha supporto ≥ 7 .

A regola, per generare le regole associative, l'algoritmo dovrebbe considerare la tabella di large itemsets precedente quindi L_1 , però qui abbiamo degli itemsets costituiti solamente da un item e non è possibile generare nessuna regola con un item solo. L'algoritmo si ferma e non ci sono regole associative.

Items1	Items2	Supporto
device1	device3	4
device3	device1	4