



Relazione su Frequent Itemsets Mining

Fontolan Federico 854230

Rosada Fabio 851772

January 21, 2017

1 Introduzione

L'esercitazione che ci è stata assegnata prevedeva di implementare un algoritmo che prendesse in analisi dieci milioni di tweet e ritornasse i frequent itemsets; il nostro gruppo ha deciso di implementare un algoritmo random sampling perché la grande mole dei dati permette di raggiungere risultati soddisfacenti anche solo analizzandone una parte relativamente piccola come il 10% senza compromettere la validità dei dati essendo lo stesso campione estremamente grande.

Il core del nostro codice è stato scritto nella sua interezza in simultanea dai membri del gruppo, alternandosi nella scrittura come previsto nella tecnica del "pair programming". La revisione finale del codice con la riscrittura dei commenti e l'eliminazione delle righe superflue è stata affidata a Fabio Rosada, mentre la scrittura di questa relazione a Federico Fontolan.

2 Scelte Fatte nella Stesura del Codice

Il nostro codice all'avvio, dopo avere stampato i dettagli del settaggio corrente (sample, threshold e file di input), genera uno SparkContext e carica in memoria la lista di stopwords che servirà per la rimozione delle stesse: la peculiarità di queste parole è quella di essere estremamente ricorrenti, quindi ogni itemset che le contenga sarà di scarso interesse.

Successivamente leggiamo ed estraiamo un sample con la combinazione delle funzioni `textFile()` e `sample()` e generiamo una lista rappresentante il contenuto dei bucket: questa scelta permette di limitare il numero di letture del file che è la parte tempisticamente più onerosa permettendoci di eseguire l'intero programma in meno di dieci minuti. Nel file è comunque presente sotto forma di commento una versione funzionante del software che legge i file ogni volta questo sia necessario per eliminare lo spazio occupato da questa lista che però causa un abbattimento generale delle prestazioni.

La lettura dei tweet viene effettuata considerando i tweet come testo e non come json: questa scelta è dovuta alla lentezza nell'elaborazione del file con il secondo metodo riscontratasi nella fase di test. Finita questa fase viene eseguito un algoritmo apriori che continua a ricercare itemsets finché esiste almeno un itemset di rango inferiore: questo è dovuto al principio di monotonicità.

3 Risultati

NOTA: le funzioni per sinteticità mostrano i primi 20 risultati ordinati per frequenza, ma è possibile visualizzarli cambiando i valori di input nel codice.

```
#NOTA#  
tempo di esecuzione: 7.9 minuti  
  
### main.py ###  
  
File: /srv/2015-01-08_geo_en_it_10M.plain.json  
Threshold: 1000  
Sample size: 0.1  
Finita la lettura del basket...  
  
Finito il conteggio delle parole...
```

TOP 20 ABSOLUTE

1	love: 49010
2	just: 46679
3	like: 42163
4	follow: 30957
5	don: 30400
6	amp: 27490
7	good: 25804
8	know: 22369
9	lol: 21365
10	day: 21125
11	time: 21120
12	new: 20812
13	want: 19514
14	happy: 18860
15	today: 18384
16	got: 18317
17	need: 16991
18	people: 16686
19	make: 15972
20	life: 15216

Inizio calcolo rank: 2
lunghezza risultati: 48

TOP 20 ABSOLUTE

```
1 (u'follow', u'love'): 6696
2 (u'@harry_styles', u'follow'): 3482
3 (u'birthday', u'happy'): 3194
4 (u'@harry_styles', u'love'): 2753
5 (u'don', u'know'): 2635
6 (u'just', u'like'): 2128
7 (u'don', u'like'): 2068
8 (u'don', u'just'): 2032
9 (u'new', u'watch'): 2027
10 (u'video', u'watch'): 2009
11 (u'good', u'morning'): 1916
12 (u'feel', u'like'): 1841
13 (u'follow', u'please'): 1830
14 (u'dream', u'love'): 1818
15 (u'love', u'please'): 1702
16 (u'happy', u'love'): 1584
17 (u'love', u'make'): 1549
18 (u'don', u'want'): 1515
19 (u'just', u'want'): 1491
20 (u'love', u'nyou'): 1457
```

Inizio calcolo rank: 3
lunghezza risultati: 15

TOP 20 ABSOLUTE

```
1 (u'@harry_styles', u'follow', u'love'): 2203
2 (u'new', u'video', u'watch'): 1694
3 (u'just', u'photo', u'posted'): 1355
4 (u'follow', u'love', u'please'): 1297
5 (u'@fifthharmony', u'prize', u'win'): 1267
6 (u'@fifthharmony', u'entering', u'prize'): 1221
7 (u'@fifthharmony', u'daily', u'prize'): 1220
8 (u'@fifthharmony', u'fanuary', u'prize'): 1217
9 (u'barometer', u'temperature', u'today'): 1197
10 (u'@fifthharmony', u'harmony', u'prize'): 1192
11 (u'@fifthharmony', u'fifth', u'prize'): 1191
12 (u'dream', u'follow', u'love'): 1181
13 (u'@themattspinosa', u'video', u'watch'): 1133
14 (u'barometer', u'humidity', u'temperature'): 1125
15 (u'barometer', u'rain', u'temperature'): 1061
```

Inizio calcolo rank: 4
lunghezza risultati: 7

TOP 20 ABSOLUTE

```
1 (u'@fifthharmony', u'daily', u'prize', u'win'): 1266
2 (u'@fifthharmony', u'entering', u'prize', u'win'): 1263
3 (u'@fifthharmony', u'january', u'prize', u'win'): 1259
4 (u'@fifthharmony', u'harmony', u'prize', u'win'): 1234
5 (u'@fifthharmony', u'fifth', u'prize', u'win'): 1233
6 (u'barometer', u'humidity', u'temperature', u'today'): 1191
7 (u'thematospinosa', u'new', u'video', u'watch'): 1045
```

Inizio calcolo rank: 5
lunghezza risultati: 6

TOP 20 ABSOLUTE

```
1 (u'@fifthharmony', u'daily', u'entering', u'prize', u'win'): 1266
2 (u'@fifthharmony', u'daily', u'january', u'prize', u'win'): 1216
3 (u'@fifthharmony', u'daily', u'harmony', u'prize', u'win'): 1193
4 (u'@fifthharmony', u'daily', u'fifth', u'prize', u'win'): 1192
5 (u'barometer', u'humidity', u'temperature', u'today', u'wind'): 1191
6 (u'barometer', u'humidity', u'rain', u'temperature', u'today'): 1191
```

Inizio calcolo rank: 6
lunghezza risultati: 4

TOP 20 ABSOLUTE

```
1 (u'@fifthharmony', u'daily', u'entering', u'harmony', u'prize', u'win'): 1239
2 (u'@fifthharmony', u'daily', u'fifth', u'harmony', u'prize', u'win'): 1191
3 (u'barometer', u'humidity', u'rain', u'temperature', u'today', u'wind'): 1191
4 (u'@fifthharmony', u'daily', u'january', u'harmony', u'prize', u'win'): 1189
```

Inizio calcolo rank: 7
lunghezza risultati: 2

TOP 20 ABSOLUTE

```
1 (u'@fifthharmony', u'daily', u'entering', u'fifth', u'harmony', u'prize', u'win'): 1238
2 (u'@fifthharmony', u'daily', u'entering', u'january', u'harmony', u'prize', u'win'): 1236
```

Inizio calcolo rank: 8
lunghezza risultati: 1

TOP 20 ABSOLUTE

```
1 (u'@fifthharmony', u'daily', u'entering', u'january', u'fifth', u'harmony', u'prize', u'win'): 1235
```

Inizio calcolo rank: 9
lunghezza risultati: 0
Non ci sono più risultati utili

Commenti Per questo esperimento è stato scelto un threshold molto basso (lo 0.001) per permettere di ottenere anche un buon numero di tuple di rango superiore al 2.

Nel corso di vari test sia con il file maggiore sia con quello minore abbiamo notato che alcune particolari tuple molto lunghe compaiono un numero molto elevato di volte: questo fenomeno succede se un tweet viene retwettato un numero di volte superiore al threshold.