Association Rules. Apriori algorithm

Data mining lab 7. Part I

Association rules generation

- Step 1. Find all frequent itemsets Fi, 2<=i<=T,

  T -total number of items
- Step 2. Generate rules from the frequent itemsets

Tutorial exercise 1. Frequent itemsets

Find all frequent itemsets from the following data. Minsupport = 2.

Pizza toppings dataset:

Order ID	Extra cheese	Onions	Peppers	Mushrooms	Olives	Anchovy
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

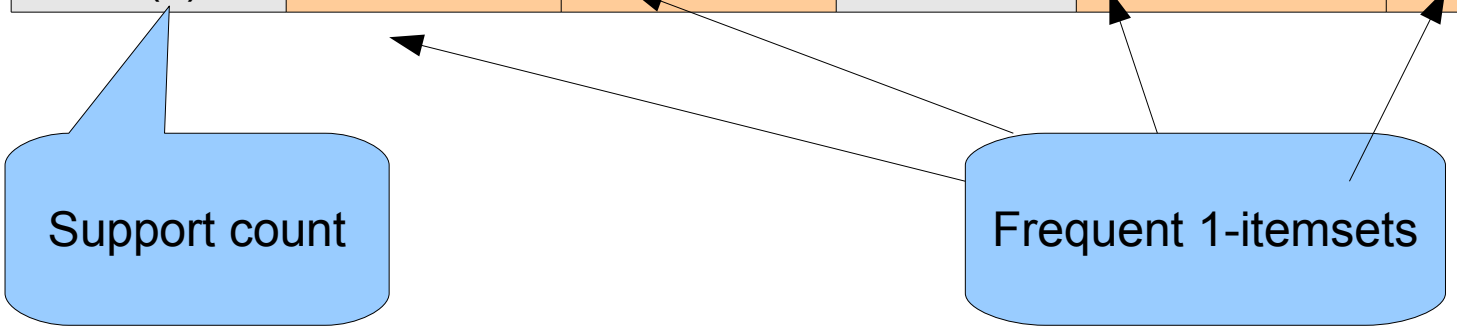
Binary data format

1. Replace item names by codes

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

2. Count 1-item frequent itemsets F1

	A	B	C	D	E	F
	1	1			1	
			1	1		
		1				1
	1			1		
	1	1		1	1	
	1	1		1		
Total (s):	4	4	1	4	2	1



3. Generate candidate 2-item frequent itemsets C2

	A	B	D	E
A				
B				
D				
E				

C2: {{A,B},{A,D},{A,E},{BD},{B,E},{D,E}}

4. Verify counts of C2 by counting

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

	A	B	D	E
A		3	3	2
B			2	2
D				~~1~~
E				

F2={{A,B},{A,D},{A,E},{B,D},{B,E})

5. Generate candidate 3-item frequent itemsets C3

F2={{A,B},{A,D},{A,E},{B,D},{B,E})

	A,B	A,D	A,E	B,D	B,E
A,B					
A,D					
A,E					
B,D					
B,E					

3-items frequent itemsets must share 3-2=1 item

C3={{A,B,D},{A,B,E},{A,D,E},{B,D,E}}

6. Prune C2 based on F2

F2={{A,B},{A,D},{A,E},{B,D},{B,E})

C3={{A,B,D},{A,B,E},{A,D,E},{B,D,E}}

Pruned, since subset {D,E} is infrequent

C3 after pruning={{A,B,D},{A,B,E}}

7. Verify 3-item itemsets counts – generate F3

Order ID	A	B	C	D	E	F
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

	A,B	A,D	A,E	B,D	B,E
A,B					
A,D					
A,E					
B,D					
B,E					

F3={{A,B,D},{A,B,E}}

8. Generate 4-item candidate frequent itemsets C4

The only possible 4-items itemset from F3={{A,B,D},{A,B,E}} is C4={A,B,D,E}

Prune C4 using F3

F3={{A,B,D},{A,B,E}}

{A,B,D,E} is pruned since its subset {B,D,E} is infrequent

9. Report frequent 2-3 itemsets:

F2={{A,B},{A,D},{A,E},{B,D},{B,E})

F3={{A,B,D},{A,B,E}}

Order ID	Extra cheese	Onions	Peppers	Mushrooms	Olives	Anchovy
1	1	1			1	
2			1	1		
3		1				1
4	1			1		
5	1	1		1	1	
6	1	1		1		

Customers buy following toppings together:
{extra cheese, onions, mushrooms}, {extra cheese, onions,olives} etc

10. Designing code for Apriori

General pseudocode

```
F1 = {frequent 1-item sets};
k = 2;
while( Fk-1 is not empty )
{
    Ck = Apriori_generate( Fk-1 );
    Ck = Apriori_prune( Ck );
    for all transactions t in T
        {Subset( Ck, t );}
    Fk = { c in Ck s.t. c.count >=minimum_support};
    k++;
}

Answer = union of all sets Fk;
```

11. Generating 1-frequent itemset (in *map_reduce* framework). Map

Local test

```
def fun_map_1(e,params):
    delimiter=params['delimiter']
    count_map={}

    for row in e.split('\n'):
        transactions=row.split(delimiter)
        for i in range (0,len(transactions)):
            if transactions[str(i)]==1:
                count_map[str(i)]=1


    return [ (i,count) for i,count in count_map.iteritems()]
```

Map returns pairs (item id,1)

12. Generating 1-frequent itemset (in a *map_reduce* framework). Reduce

Local test

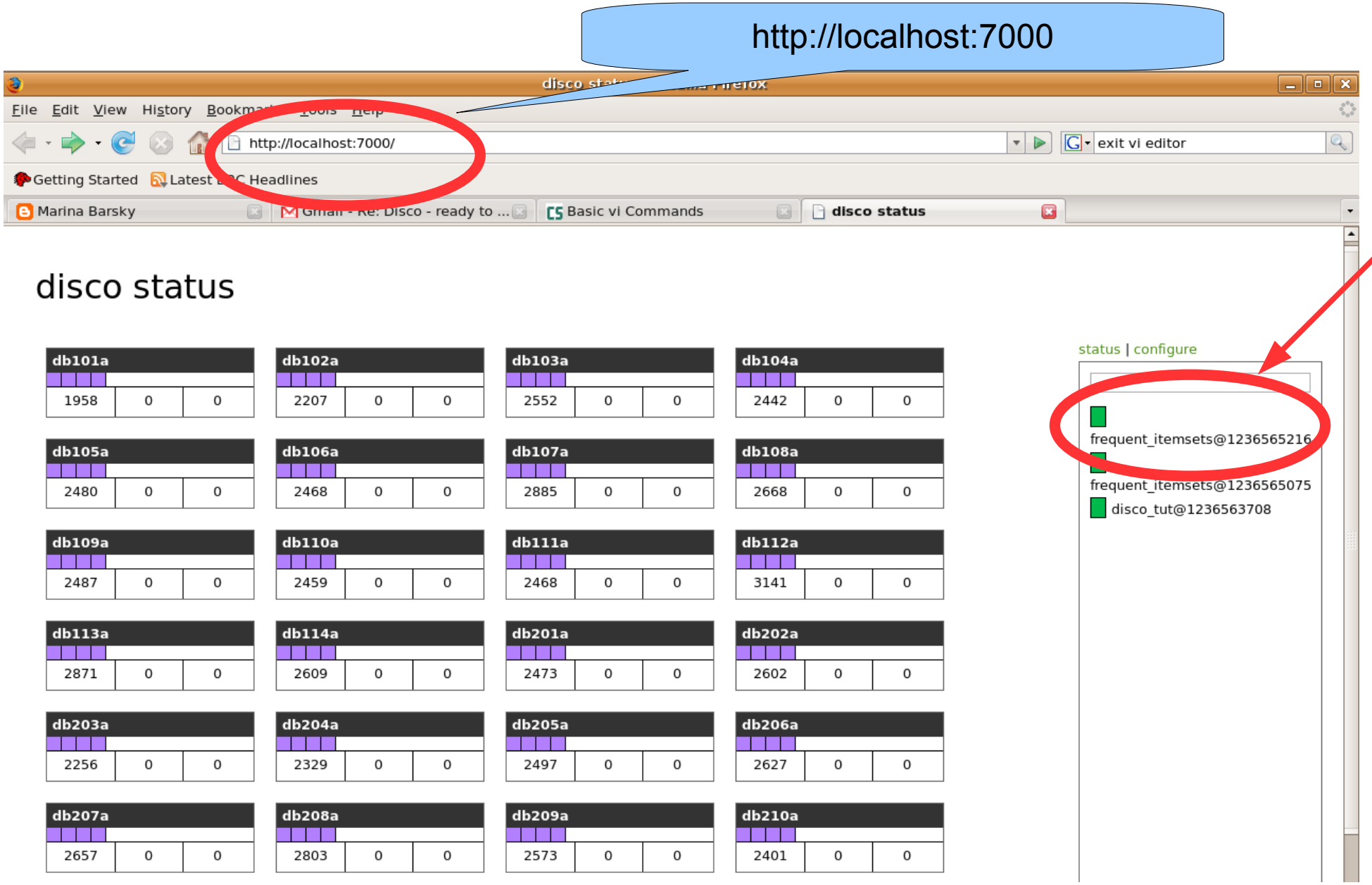
```
def fun_reduce1(iter, out, params):
    count_dict = {}
    min_supp=params['min_support']
    for i, count in iter:
       if i not in count_dict:
           count_dict[i]=int(count)
       else:
           count_dict[i]+=int(count)

    for i, total in count_dict.iteritems():
       if total>=min_supp
           out.add(i, total)
```

Reduce adds counts for each item and adds them to the output list if the count is at least min_supp

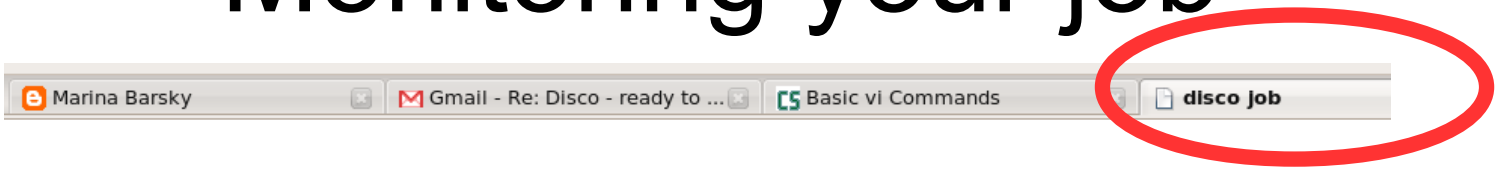
Disco session

```
mgbarsky@mgbarsky-pc:~$ ssh -L 7000:db101a:7000
dbssh1.cs.uvic.ca

mgbarsky@dbssh1.cs.uvic.ca's password:

[mgbarsky@db101a ~]$ wget
http://webhome.cs.uvic.ca/~mgbarsky/inputs/marketbasket.csv

[mgbarsky@db101a ~]$ wget
http://webhome.cs.uvic.ca/~mgbarsky/dmlabs/ficount.py

[mgbarsky@db101a ~]$ sudo distrfiles marketbasket.csv  200

Password:

[mgbarsky@db101a ~]$ sudo discorun ficount.py `cat
marketbasket.csv.chunks` > result_data

[  OK  ]
```

Monitoring the cluster



Monitoring your job



job:
frequent_itemsets@1236565216

[delete job records] [delete all job data]

Job info

Job is ready

started: 2009/03/08 19:20:17

	Waiting	Running	Done	Failed
Map	0	0	10	0
Reduce	0	0	5	0

Current nodes

Map Inputs

disco://db204a/mgbarsky.part_market
disco://db108a/mgbarsky.part_market
disco://db109a/mgbarsky.part_market
disco://db205a/mgbarsky.part_market
disco://db201a/mgbarsky.part_market

Results

dir://db201a/reduce/frequent_itemset
dir://db301a/reduce/frequent_itemset
dir://db107a/reduce/frequent_itemset
dir://db207a/reduce/frequent_itemset
dir://db309a/reduce/frequent_itemset

Filter: show

2009/03/08 19:20:17 master
READY
2009/03/08 19:20:17 master
Reduce phase done
2009/03/08 19:20:17 master
Received results from reduce:4 @ db201a.
2009/03/08 19:20:17 db201a
[reduce:4] Worker done
2009/03/08 19:20:17 db201a
[reduce:4] Reduce done

Results

```
Starting Disco job

Job 1 done. Result

216 36

217 92

66 30

215 56

212 63

213 40

210 85

211 10

214 15

218 34
```

13. To do

- The output of fun_reduce1 is a list of tuples
- Lexicographically order list by item id
- Combine 1-itemsets into 2-itemsets – generate C2 in form of the dictionary C2={(item1,item2),0}
- No pruning since each set has only 2 subsets and both are frequent
- Implement fun_map2 which takes the same input and C2 as a parameter. It counts for each transaction the number of co-occurrence of item1 and item2.
- The output of fun_map2 is reduced into frequent 2-item itemset F2

Output- F2. Min_support=60

('110', '238') 64	Tangerines, Creamy Peanut Butter
('132', '16') 66	Chicken Soup, Vanilla Ice Cream
('132', '141') 75	Chicken Soup, Blueberry Waffles
('141', '4') 67	Blueberry Waffles, Frozen Chicken Wings
('141', '175') 65	Blueberry Waffles, Microwave Popcorn
('124', '141') 70	Bologna, Blueberry Waffles
('124', '132') 71	Bologna, Chicken Soup
('132', '238') 61	Chicken Soup, Creamy Peanut Butter
('124', '16') 61	Bologna, Vanilla Ice Cream
('141', '16') 70	Blueberry Waffles, Vanilla Ice Cream
('132', '175') 61	Chicken Soup, Microwave Popcorn
('201', '4') 66	Frozen Peas, Frozen Chicken Wings
('132', '4') 62	Chicken Soup, Frozen Chicken Wings

Setting min_support high we possible miss all the interesting associations