Apriori Algorithm and Rule Generation

Group member: Fanzi Xiao Shaoming Xu Haowei Zhou

Apriori Algorithm implementation detail

We implement the Apriori algorithm by these steps.

First, we get data and label from the file, then preprocess the data in place.

Second, we get all the length 1 itemsets from the data, and used them to generate the length 1 frequent and the infrequent itemsets.

Then we repeat the followed steps until no frequent itemsets are generated.

1. generate length (k+1) candidate itemsets from length k frequent itemsets.
2. prune the length (k+1) candidate itemsets by using the length (k) infrequent
3. Record the support for each length (k+1) candidate by counting the number of them in the original data. Separate them into frequent or the infrequent list.

Association rules implementation detail

The key part is wrote in function rule_gen() where we use the BFS algorithm to achieve the Association rules algorithm.

First, make a queue by deque(), and append the given itemset in queue. The queue saves the head of the rules.

Then repeat the followed steps until the length of queue is zero.

1. Pop a head in queue
2. Divide the the support of given itemset with the head to get the confidence.
3. If the confidence is not smaller than the minimum confidence
    a. Save the rule in the result
    b. When the length of head is not less than 1, generate the new heads by the head, and append them in the queue. Suppose the length of new heads is k, then the head length is k+1.

The answer of the queries in requirement 1

```
Support is set to be 30.0%
number of length-1 frequent itemsets: 194
number of length-2 frequent itemsets: 5323
number of length-3 frequent itemsets: 5251
number of length-4 frequent itemsets: 1463
number of length-5 frequent itemsets: 388
number of length-6 frequent itemsets: 61
number of length-7 frequent itemsets: 3
number of all lengths frequent itemsets: 12683


Support is set to be 40.0%
number of length-1 frequent itemsets: 167
number of length-2 frequent itemsets: 753
number of length-3 frequent itemsets: 149
number of length-4 frequent itemsets: 7
number of length-5 frequent itemsets: 1
number of all lengths frequent itemsets: 1077
```

```
Support is set to be 50.0%
number of length-1 frequent itemsets: 109
number of length-2 frequent itemsets: 63
number of length-3 frequent itemsets: 2
number of all lengths frequent itemsets: 174


Support is set to be 60.0%
number of length-1 frequent itemsets: 34
number of length-2 frequent itemsets: 2
number of all lengths frequent itemsets: 36


Support is set to be 70.0%
number of length-1 frequent itemsets: 7
number of all lengths frequent itemsets: 7
```

The answer of the queries in requirement 2

Support: 0.5

Confidence: 0.7

Template 1

result11 = template1("RULE", "ANY", ['G59_Up'], 0.7,record50, fq_list50)
count11 = 26

result12 = template1("RULE", "NONE", ['G59_Up'], 0.7,record50, fq_list50)
count12 = 91

result13 = template1("RULE", 1, ['G59_Up', 'G10_Down'], 0.7,record50, fq_list50)
count13 = 39

result14 = template1("HEAD", "ANY", ['G59_Up'], 0.7,record50, fq_list50)
count14 = 9

result15 = template1("HEAD", "NONE", ['G59_Up'], 0.7,record50, fq_list50)
count15 = 108

result16 = template1("HEAD", "1", ['G59_Up', 'G10_Down'], 0.7,record50, fq_list50)
count16 = 17

result17 = template1("BODY", "ANY", ['G59_Up'], 0.7,record50, fq_list50)
count17 = 17

result18 = template1("BODY", "NONE", ['G59_Up'], 0.7,record50, fq_list50)
count18 = 100

result19 = template1("BODY", "1", ['G59_Up', 'G10_Down'], 0.7,record50, fq_list50)
count19 = 24

template 2

```
result21 = template2("RULE", 3, 0.7, record50, fq_list50)
count21 = 9

result22 = template2("HEAD", 2, 0.7, record50, fq_list50)
count22 = 6

result23 = template2("BODY", 1, 0.7, record50, fq_list50)
count23 = 117
```

template 3

```
result31 = template3("1or1", "HEAD", "ANY", ['G10_Down'], "BODY", 1, ['G59_Up'],
0.7 ,record50, fq_list50)
count31 = 24

result32 = template3("1and1", "HEAD", "ANY", ['G10_Down'], "BODY", 1, ['G59_Up'],
0.7 ,record50, fq_list50)
count32 = 1

result33 = template3("1or2", "HEAD", "ANY", ['G10_Down'], "BODY", 2, 0.7 ,record50,
fq_list50)
count33 = 11

result34 = template3("1and2", "HEAD", "ANY", ['G10_Down'], "BODY", 2, 0.7 ,record50,
fq_list50)
count34 = 0

result35 = template3("2or2", "HEAD", 1, "BODY", 2, 0.7 ,record50, fq_list50)
count35 = 117

result36 = template3("2and2", "HEAD", 1, "BODY", 2, 0.7 ,record50, fq_list50)
count36 = 3
```