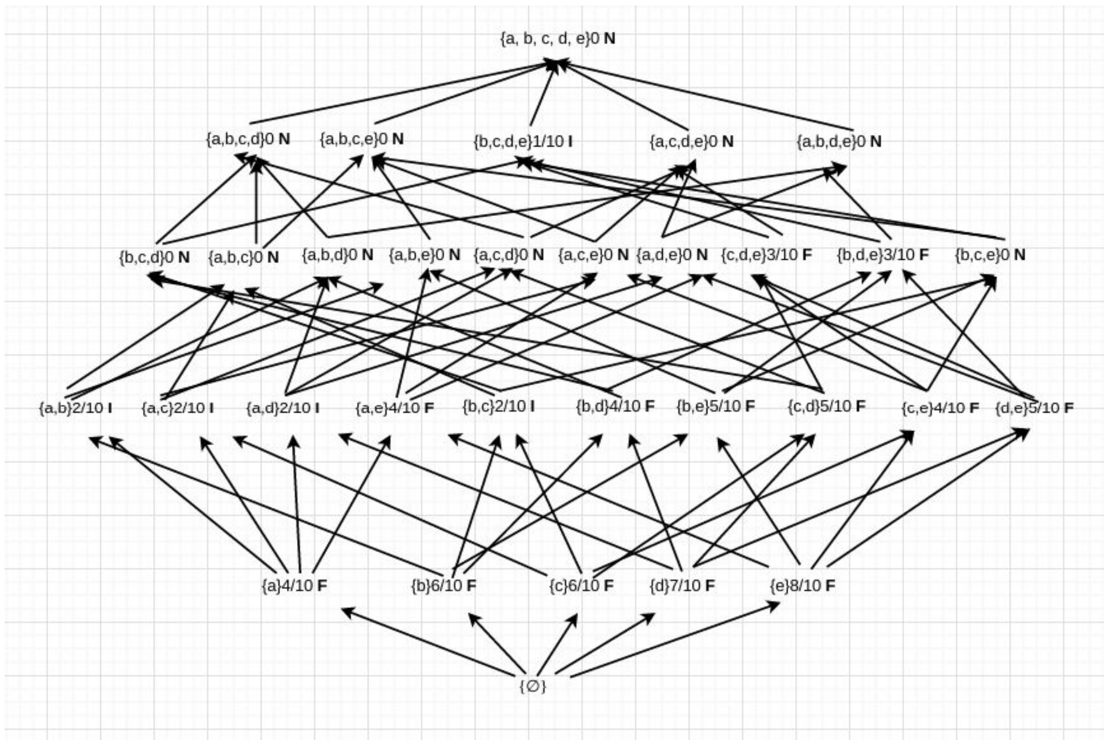
Jason Chen
Data Mining
Homework 5

NOTE: Using 1 late day

1.



i.

ii.

1. 13/31 or 41.94%
2. 13/31 ir 41.94%
3. 5/31 or 16.13%

iii.

1.

b -> c

	c	¬c	
b	.2	.4	.6
¬b	.4	0	.4
	.6	.4	1

a -> d

	d	¬d	
a	.2	.2	.4
¬a	.5	.1	.6
	.7	.3	1

b -> d

	d	¬d	
b	.4	.2	.6
¬b	.3	.1	.4
	.7	.3	1

e -> c

	c	¬c	
e	.4	.4	.8
¬e	.2	0	.2
	.6	.4	1

d -> e

	e	¬e	
d	.5	.2	.7
¬d	.3	0	.3
	.8	.2	1

2. b -> c
    Lift: .33 / 0.6 = 0.55
    Interest: 0.2 / 0.6 / 0.6 = 0.55
    Confidence: 0.2/0.6 = 0.33
    support : 0.2

a -> d

Lift: 0.5 / 0.7 = .71

Interest: 0.2 / 0.4 / 0.7 = .71

Confidence: 0.2 / 0.4 = 0.5

support : 0.2

b -> d

Lift: 0.33 / 0.7 = 0.47

Interest: 0.4 /0.6 /0.7 = .47

support : 0.4

Confidence: 0.4 / 0.6 = .33

e -> c

Lift: 0.5/0.6= 0.83

Interest: 0.4 / 0.8 / 0.6 = 0.83

Confidence: 0.4 / 0.8 = 0.5

support : 0.4

d -> e

Lift: 0.71 / 0.8 = 0.89

Interest: 0.5 / 0.7 / 0.8 = 0.89

Confidence: 0.5 / 0.7 = 0.71

support : 0.5

2.

i. Assuming each attribute is a binary, **2,047** possible frequent itemsets
But if we are considering all attributes, since attributes like city can take on many different values.
The possible frequent itemsets would be: $\mathbf{2^{188} - 1}$ possible frequent itemsets

ii.
$$R = \sum_{k=1}^{d-1} \left[\binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right]$$
= 3^11-2^12+1
173,052 possible association rules
Source:http://michael.hahsler.net/SMU/EMIS7331/slides/chap6_basic_association_analysis.pdf

iii. Support will have the larger impact on the efficiency of the apriori association rule algorithm simply because support tends to weed out infrequent items not meeting the minimum support, reducing the size of the frequent itemsets rapidly. This is called the rare item problem and may become an issue with accuracy of the association rules produced but will undoubtedly make the algorithm run faster since there are less frequent itemsets to iterate over.

3. Algorithm successfully finds association rules
4.
   i.
      1. ({noiseLevel$average,priceRange$2}->goodForGroups$1) support: 0.333511834636 confidence: 0.954044117647
      2. ({alcohol$full_bar,attire$casual}->goodForGroups$1) support: 0.344114812038 confidence: 0.971575446024
      3. ({attire$casual,state$AZ}->goodForGroups$1) support: 0.371425511406 confidence: 0.905483028721
      4. ({alcohol$full_bar}->goodForGroups$1) support: 0.375816643461 confidence: 0.969872857933
      5. ({state$AZ}->goodForGroups$1) support: 0.378279961444 confidence: 0.906338208879
      6. ({attire$casual,priceRange$1}->goodForGroups$1) support: 0.382885295063 confidence: 0.860823501084
      7. ({priceRange$1}->goodForGroups$1) support: 0.383420798972 confidence: 0.86078384227
      8. ({alcohol$none,attire$casual}->goodForGroups$1) support: 0.38449180679 confidence: 0.838001867414
      9. ({alcohol$none}->goodForGroups$1) support: 0.385777016172 confidence: 0.838064215914
      10. ({attire$casual,noiseLevel$average,waiterService$}->goodForGroups$1) support: 0.392524365428 confidence: 0.926440849343
      11. ({noiseLevel$average,waiterService$}->goodForGroups$1) support: 0.405590660812 confidence: 0.92818627451
      12. ({attire$casual,caters$}->goodForGroups$1) support: 0.43290136018 confidence: 0.913652802893
      13. ({caters$}->goodForGroups$1) support: 0.441897825854 confidence: 0.914247728784
      14. ({attire$casual,priceRange$2}->goodForGroups$1) support: 0.456142229838 confidence: 0.940591872792
      15. ({priceRange$2}->goodForGroups$1) support: 0.460104958766 confidence: 0.940455341506
      16. ({attire$casual,waiterService$}->goodForGroups$1) support: 0.552211631145 confidence: 0.911276069282
      17. ({waiterService$}->goodForGroups$1) support: 0.57470279533 confidence: 0.912740261949

18. ({attire$casual,noiseLevel$average}->goodForGroups$1) support: 0.612723572882 confidence: 0.919036144578
19. ({noiseLevel$average}->goodForGroups$1) support: 0.631466209703 confidence: 0.920387137059
20. ({attire$casual}->goodForGroups$1) support: 0.871907464924 confidence: 0.903551609323

ii. Support is usually good at including interesting rules, but sometimes may leave out some interesting rules due to the rare item problem, where significant items are left out because they are infrequent. Confidence can sometimes be misleading because although confidence is high, the confidence of a rule with the inverse of the consequent variable could be even higher (e.g p(A|B) = .6, p(A|¬B) = .9)

iii.

Lift refers to the association between the two variables. Having lift<1 means the variables are negatively association and lift>1 means the variables are positively association. Utilizing lift instead of confidence seems to put more weight on larger itemsets, as we can see the itemsets with the highest lift values association larger sized itemsets with the consequent item (e.g alcohol$full_bar attire$casual noiseLevel$average->goodForGroups$1)

1. {noiseLevel$average}->goodForGroups$1:
   lift: 1.01688021521 support: 0.631466209703
2. {attire$casual,noiseLevel$average,waiterService$}->goodForGroups$1:
   lift: 1.02356859665 support: 0.392524365428
3. {noiseLevel$average,waiterService$}->goodForGroups$1:
   lift: 1.02549701161 support: 0.405590660812
4. {attire$casual,stars$3.5}->goodForGroups$1:
   lift: 1.02893464886 support: 0.274392203063
5. {stars$3.5}->goodForGroups$1:
   lift: 1.03017256944 support: 0.283709971083
6. {attire$casual,caters$,noiseLevel$average}->goodForGroups$1:
   lift: 1.03193503837 support: 0.313805290779
7. {caters$,noiseLevel$average}->goodForGroups$1:
   lift: 1.03273214901 support: 0.319053229089
8. {priceRange$2}->goodForGroups$1:
   lift: 1.03905236347 support: 0.460104958766
9. {attire$casual,priceRange$2}->goodForGroups$1:
   lift: 1.03920320864 support: 0.456142229838
10. {priceRange$2,waiterService$}->goodForGroups$1:
    lift: 1.04517832209 support: 0.30395201885
11. {attire$casual,priceRange$2,waiterService$}->goodForGroups$1:
    lift: 1.04537974258 support: 0.301274499304

12. {attire$casual,noiseLevel$average,priceRange$2}->goodForGroups$1:
    lift: 1.05381566469 support: 0.331798222127
13. {noiseLevel$average,priceRange$2}->goodForGroups$1:
    lift: 1.05406578233 support: 0.333511834636
14. {alcohol$full_bar}->goodForGroups$1:
    lift: 1.07155400243 support: 0.375816643461
15. {alcohol$full_bar,attire$casual}->goodForGroups$1:
    lift: 1.07343508928 support: 0.344114812038
16. {alcohol$full_bar,waiterService$}->goodForGroups$1:
    lift: 1.07424819652 support: 0.259505194388
17. {alcohol$full_bar,priceRange$2}->goodForGroups$1:
    lift: 1.0761638106 support: 0.281353753882
18. {alcohol$full_bar,attire$casual,priceRange$2}->goodForGroups$1:
    lift: 1.07659564246 support: 0.277605226518
19. {alcohol$full_bar,noiseLevel$average}->goodForGroups$1:
    lift: 1.07755967224 support: 0.270750776481
20. {alcohol$full_bar,attire$casual,noiseLevel$average}->goodForGroups$1:
    lift: 1.07783654043 support: 0.252222341223

5. **NOTE:** "Total Itemsets" refers to the total **frequent** itemsets discovered

   i.

```
→ homework5 git:(master) ✗ python association-rules.py yelp4.csv .25 .75
frequent itemsets size 1 - 13
frequent itemsets size 2 - 37
frequent itemsets size 3 - 36
frequent itemsets size 4 - 11

Total itemsets: 97
Total Association rules with single-variables in the consequent: 101
```

   ii. minSup = 0.1:

```
→ homework5 git:(master) ✗ python association-rules.py yelp4.csv .1 .75
frequent itemsets size 1 - 19
frequent itemsets size 2 - 92
frequent itemsets size 3 - 173
frequent itemsets size 4 - 145
frequent itemsets size 5 - 51
frequent itemsets size 6 - 4

Total itemsets: 484
Total Association rules with single-variables in the consequent: 576
```

minSup = 0.3:

```
→ homework5 git:(master) ✗ python association-rules.py yelp4.csv .3 .75
frequent itemsets size 1 - 12
frequent itemsets size 2 - 23
frequent itemsets size 3 - 17
frequent itemsets size 4 - 4

Total itemsets: 56
Total Association rules with single-variables in the consequent: 55
```

minSup = 0.5:

```
→ homework5 git:(master) ✗ python association-rules.py yelp4.csv .5 .75
frequent itemsets size 1 - 4
frequent itemsets size 2 - 5
frequent itemsets size 3 - 2

Total itemsets: 11
Total Association rules with single-variables in the consequent: 10
```

iii.

minConfidence = 0.4:

```
→ homework5 git:(master) ✗ python association-rules.py yelp4.csv .25 .4
frequent itemsets size 1 - 13
frequent itemsets size 2 - 37
frequent itemsets size 3 - 36
frequent itemsets size 4 - 11

Total itemsets: 97
Total Association rules with single-variables in the consequent: 216
```

minConfidence = 0.6:

```
→ homework5 git:(master) ✗ python association-rules.py yelp4.csv .25 .6
frequent itemsets size 1 - 13
frequent itemsets size 2 - 37
frequent itemsets size 3 - 36
frequent itemsets size 4 - 11

Total itemsets: 97
Total Association rules with single-variables in the consequent: 155
```

minConfidence = 0.8:

```
→ homework5 git:(master) ✗ python association-rules.py yelp4.csv .25 .8
frequent itemsets size 1 - 13
frequent itemsets size 2 - 37
frequent itemsets size 3 - 36
frequent itemsets size 4 - 11

Total itemsets: 97
Total Association rules with single-variables in the consequent: 93
```

iv. The findings from these evaluations supports my point that support has a greater impact on the efficiency of the algorithm. As support increases, the total itemsets drastically decreases. This causes the total number of association rules to also decrease drastically. The reason for such a decrease in frequent itemsets is because the probability of an itemset needs to be much higher in order to meet the support threshold.