CS57300: Homework 5

Hasini, Urala Liyanage Dona Gunasinghe

Q1

(a)

Item sets of size 1 = 2002
Items sets of size 2 = $\binom{2002}{2}$
Item sets of size 3 = $\binom{2003}{3}$
Size of the item set space with item sets of sizes 1-3 = 2002 + $\binom{2002}{2}$ + $\binom{2002}{3}$
= 2002 + 2003001 + 1335334000
= 1337339003
Note: I did not count the empty set since in the specification (2nd bullet point in Q1) asks to consider item sets of sizes [1-3].

(b)

1. (i). Items considered by the algorithm and found to be frequent: 720
2. (ii). Items considered by the algorithm but found to be infrequent: 73423

(c)

Pruning Ratio = $\frac{\text{number of itemsets not considered to be a candidate}}{\text{size of the item set space}} * 100\%$
= $\frac{1337339003-74143}{1337339003}$ *100% = $\frac{1337264860}{1337339003}$ *100%
= 99.99%
Note: Algorithm for pruning candidate sets is found in the *generateCandidateItemSets* method of *apriori.py* file, in which I did the two level of pruning.

(d)

False alarm rate = $\frac{\text{candidate item sets found to be infrequent}}{\text{candidate item sets for which the support was explicitly counted}}$
= $\frac{73423}{74143}$ *100%
= 99.028%

(e)

Below I report the top 30 association rules that are discovered, ordered by confidence, along with the support of the corresponding item set that generated the rule (I assume that the support asked here is the support of the whole item set from which the rule was constructed, and not the individual support for antecedent and consequence.).

IF ever AND worst THEN isNegative, support: 0.0308, confidence: 0.993548387097

IF worst THEN isNegative, support: 0.0532, confidence: 0.992537313433

IF horrible THEN isNegative, support: 0.0368, confidence: 0.983957219251

IF rude THEN isNegative, support: 0.0488, confidence: 0.968253968254

IF terrible THEN isNegative, support: 0.0352, confidence: 0.967032967033

IF fantastic THEN isPositive, support: 0.0316, confidence: 0.923976608187

IF manager THEN isNegative, support: 0.0552, confidence: 0.923076923077

IF delicious THEN isPositive, support: 0.0678, confidence: 0.91869918699

IF excellent THEN isPositive, support: 0.0424, confidence: 0.917748917749

IF amazing THEN isPositive, support: 0.0608, confidence: 0.910179640719

IF waited THEN isNegative, support: 0.0306, confidence: 0.889534883721

IF madison THEN isPositive, support: 0.0454, confidence: 0.876447876448

IF perfect THEN isPositive, support: 0.0332, confidence: 0.869109947644

IF awesome THEN isPositive, support: 0.052, confidence: 0.860927152318

IF phone THEN isNegative, support: 0.0336, confidence: 0.857142857143

IF wonderful THEN isPositive, support: 0.0328, confidence: 0.854166666667

IF staff AND friendly THEN isPositive, support: 0.036, confidence: 0.85308056872

IF money THEN isNegative, support: 0.0542, confidence: 0.846875

IF asked THEN isNegative, support: 0.0782, confidence: 0.840860215054

IF later THEN isNegative, support: 0.0486, confidence: 0.840830449827

IF favorite THEN isPositive, support: 0.0496, confidence: 0.823920265781

IF friendly THEN isPositive, support: 0.0938, confidence: 0.822807017544

IF minutes THEN isNegative, support: 0.0854, confidence: 0.814885496183

IF customers THEN isNegative, support: 0.0372, confidence: 0.812227074236

IF finally THEN isNegative, support: 0.0448, confidence: 0.805755395683

IF 15 THEN isNegative, support: 0.0324, confidence: 0.79802955665

IF customer THEN isNegative, support: 0.0662, confidence: 0.79376498801

IF should THEN isNegative, support: 0.0678, confidence: 0.79020979021

IF love THEN isPositive, support: 0.0978, confidence: 0.786173633441

IF call THEN isNegative, support: 0.047, confidence: 0.785953177258

Discussion: Most of the association rules found above are interesting. All of them can be viewed as classification rules for reviews since the consequence is either isNegative or isPositive - which is related to the class label, although we did not explicitly looked for rules with class label being assigned as the consequence.
Also, most of them have correctly identified the key words in the reviews which affect the reviews to be positive or negative. For example, if a review contains pleasant words such as wonderful, perfect, fantastic etc, the rules predicts them as positive reviews and if a review contains unpleasant words such as horrible, rude, worst, waited etc, the rules predicts them as negative reviews. One can view them as obvious rules too.
Among them are couple of not-so obvious patterns too, such as IF call THEN isNegative, IF customers THEN isNegative and IF 15 THEN isNegative etc.
Results also reflect the effect of apriori principle on both the frequent item sets and rules. E.g: Rules pairs: (IF ever AND worst THEN isNegative, IF worst THEN isNegative) and (IF staff AND friendly THEN isPositive, IF friendly THEN isPositive) are among top 30 rules.

Q2.

(a)

Consider an arbitrary rule: IF A AND B THEN C which is created from an item set of size 3.
Let the binary features corresponding to A, B and C w.r.t 20 instances be:
A: [0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1]
B: [0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1]
C: [0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0]

When constructing the contingency table, we take (A AND B) = 1, (A AND B) = 0 as rows and C = 1 and C = 0 as columns.

Following is the contingency table with exact values for the above arbitrary rule (Above data is extracted from the results of the implementation on a subset of review data (20 reviews) which I used for testing my algorithm).

Table 1: Contingency Table

	C=1	C=0	
(A AND B) = 1	4	1	5
(A AND B) = 0	4	11	15
	8	12	20

Cell (1,1) corresponds to true positives, cell (1,2) corresponds to false positives, cell (2,1) corresponds to false negatives and cell (2,2) corresponds to true negatives.

(b)

χ^2 score $= \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$,
where i represents each cell in the contingency table, o_i is the observed value of the ith cell and e_i is the expected value of the ith cell.
Expected value is calculated assuming that antecedent and consequence are independent.
Below is how e_i is calculated:
Consider the cell (1,1) - corresponding to (A AND B) = 1 and C=1.
Expected value = p(C=1|(A AND B) = 1)*N, where N is the total number of instances.
= p(C=1).p(A AND B) = 1).N (due to independence assumption)

$= \frac{8}{20} \cdot \frac{5}{20} \cdot 20$

In general, expected value for a particular cell = $\frac{\text{row total*column total}}{total}$

(c)

Lets consider the first rule: *IF ever AND worst THEN isNegative*:
The cell counts are calculated from the support values which were computed for antecedent item set, consequence item set and the whole frequent item set during the frequent item set generation. The recorded support values for all frequent item sets and confidence values for all rules are submitted in a separate file called: "FullExperimentQ1_Final", along with the source code.

Table 2: Contingency Table (with observed values)

	isNegative=1	isNegative=0	
(ever AND worst) = 1	154	1	155
(ever AND worst) = 0	2346	2499	4845
	2500	2500	5000

Table 3: Contingency Table (with expected values)

	isNegative=1	isNegative=0
(ever AND worst) = 1	77.5	77.5
(ever AND worst) = 0	2422.5	2422.5

Chi squared score = $\sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$
= 155.85738539898134.
Associated p-value = 9.0956136187011771e-36.

(d)

Let us consider the contingency table for a generic rule (see contingency table below).
When we generalize the rule, total of each column will stay the same as we removes terms from the antecedent. Since the antecedent is a conjunction of two or more attributes (if it is a single attribute, then there wont be any further generalization), total of the row

Table 4: Contingency Table for a generic rule

	Consequence=1	Consequence=0	
Antecedent = 1	TP	FP	TP+FP
Antecedent = 0	FN	TN	TN+FN
	TP+FN	TN+FP	TP+FP+TN+FN

1 stays the same or goes up (increases) and total of row 2 stays the same or goes down (decreases), as we generalize.

(e)

When we specialize the rule, total of each column will still stay the same as we add terms to the antecedent. Since the antecedent can be a single attribute or a conjunction of two or more attributes, total of the row 1 stays the same or goes down (decreases) and total of row 2 stays the same or goes up (increases), as we specialize.

(f)

Following contingency table reports the cell counts for the best possible specialization (i.e: specialization of the rule that has highest accuracy.) for the rule reported in (c) above.

Table 5: Contingency Table (with observed values)

	isNegative=1	isNegative=0	
Antecedent = 1	2500	0	2500
Antecedent = 0	0	2500	2500
	2500	2500	5000

Accuracy of the best possible specialization $= \frac{\text{True Positives (TP) + True Negatives (TN)}}{\text{total number of instances}}$
$= 1$.

Accuracy of the initial rule $= \frac{154+2499}{5000}$
$= 2653/5000$
$= 0.5306$

(g)

In this question, I assume *further specialization* means specialization after the best possible specialization. As we discussed in (e), with further specialization, the total of row 1 can only go down or stays the same. Which means the count of cell (1,1) which corresponds to True Positives, could go down or stays the same. Also, total of row 2 can only go up or stays the same. However, count of cell (2,2) which corresponds to True Negatives could only go down or stays the same, as it is at the maximum in the best possible specialization and since the total of each columns stays the same.

Therefore, the accuracy (TP+TN)/total could only go down or stays the same with further specialization after the best possible specialization.

Once we reach a specialization of a rule (i.e: by adding terms to antecedent on a branch of rule lattice with same consequence) with maximum accuracy, and if its immediate next level of specialization decreases accuracy, further specialization of that rule also decreases accuracy. Hence we can prune all the specializations of a rule with maximum accuracy, immediately after the level which decreases its accuracy.

This is related to the apriori principle because accuracy of subsets (w.r.t further specialization) of a particular rule stays the same or is monotonically decreasing, once the accuracy has reached the maximum.

Q3.

(a)

Generating rules based on chi squared score as the interestingness measure is implemented in the source file named: "rulesChiSq.py".

(b)

Following are the newly found top 30 rules with chi squared score as the interestingness measure. I have reported support (of the corresponding frequent item set), chi squared score and p-val along with each rule.

IF ever AND isNegative THEN worst,
support: 0.0308, interestingness: 904.688173077, p-val: 9.39012060855e-199

IF worst THEN ever AND isNegative,
support: 0.0308, interestingness: 904.688173077, p-val: 9.39012060855e-199

IF friendly THEN staff AND isPositive,
support: 0.036, interestingness: 480.216803977, p-val: 1.91675953183e-106

IF staff AND isPositive THEN friendly,
support: 0.036, interestingness: 480.216803977, p-val: 1.91675953183e-106

IF worst THEN ever,
support: 0.031, interestingness: 463.589616337, p-val: 7.95674208946e-103

IF ever THEN worst,
support: 0.031, interestingness: 463.589616337, p-val: 7.95674208946e-103

IF worst AND isNegative THEN ever,
support: 0.0308, interestingness: 461.166007971, p-val: 2.68008449205e-102

IF ever THEN worst AND isNegative,
support: 0.0308, interestingness: 461.166007971, p-val: 2.68008449205e-102

IF staff THEN friendly,
support: 0.0422, interestingness: 299.358343229, p-val: 4.5453492572e-67

IF friendly THEN staff,
support: 0.0422, interestingness: 299.358343229, p-val: 4.5453492572e-67

IF isPositive THEN delicious,
support: 0.0678, interestingness: 279.373890694, p-val: 1.02810326593e-62

IF delicious THEN isPositive,
support: 0.0678, interestingness: 279.373890694, p-val: 1.02810326593e-62

IF isNegative THEN worst,
support: 0.0532, interestingness: 274.788357452, p-val: 1.02645730411e-61

IF worst THEN isNegative,
support: 0.0532, interestingness: 274.788357452, p-val: 1.02645730411e-61

IF staff THEN friendly AND isPositive,
support: 0.036, interestingness: 269.676623867, p-val: 1.33471279735e-60

IF friendly AND isPositive THEN staff,
support: 0.036, interestingness: 269.676623867, p-val: 1.33471279735e-60

IF friendly THEN isPositive,
support: 0.0938, interestingness: 268.155716605, p-val: 2.8633061958e-60

IF isPositive THEN friendly,
support: 0.0938, interestingness: 268.155716605, p-val: 2.8633061958e-60

IF sure THEN make,
support: 0.034, interestingness: 255.893587795, p-val: 1.34785413214e-57

IF make THEN sure,
support: 0.034, interestingness: 255.893587795, p-val: 1.34785413214e-57

IF isPositive THEN amazing,
support: 0.0608, interestingness: 240.868455973, p-val: 2.54299060693e-54

IF amazing THEN isPositive,
support: 0.0608, interestingness: 240.868455973, p-val: 2.54299060693e-54

IF isNegative THEN asked,
support: 0.0782, interestingness: 238.263920997, p-val: 9.40272753719e-54

IF asked THEN isNegative,

support: 0.0782, interestingness: 238.263920997, p-val: 9.40272753719e-54

IF isNegative THEN rude,
support: 0.0488, interestingness: 232.746285821, p-val: 1.50119997306e-52

IF rude THEN isNegative,
support: 0.0488, interestingness: 232.746285821, p-val: 1.50119997306e-52

IF isPositive THEN love,
support: 0.0978, interestingness: 232.704005265, p-val: 1.53341170007e-52

IF love THEN isPositive,
support: 0.0978, interestingness: 232.704005265, p-val: 1.53341170007e-52

IF minutes THEN isNegative,
support: 0.0854, interestingness: 232.154186194, p-val: 2.02096972914e-52

IF isNegative THEN minutes,
support: 0.0854, interestingness: 232.154186194, p-val: 2.02096972914e-52

Discussion: In this case, we have less number of rules involving different features, because for almost every rule, the reverse of the same rule is also among the top 30. Since the chi-squared score is used as the interestingness measure, the rules generated from the same frequent item set with antecedent and consequence interchanged, get the same level of interestingness score. Although it is interesting to see that there is a significant overlap between the top 30 rules generated in the question 1 and here, some of the rules which were among the top 30 rules in the previous case are missing here - e.g: IF ever AND worst THEN isNegative. Overall, it seems that chi squared score is not a very good measure for interestingness.

(c)

Multiple comparison problem in the context of association rules algorithms is: when the number of statistical tests that we perform increases, the number of instances which will have p-values less than the significance threshold (say 0.05) purely by chance also increases, even if all null hypotheses that we test are really true. That means number of false positives increases.

In association rule mining algorithms, if we use a statistical measure as the measure of interestingness, we decide if a rule is interesting if the p-value related to its statistical measure is less than the significance threshold.

However, since we are testing large number of rules for significance, there is a high chance

that we get a high false positive rate (i.e: concluding that a rule is statistically significant (interesting), although it is not and although the null hypotheses is true).
In order to avoid that we can apply Bonferroni correction by dividing the significance threshold by the number of total tests that we perform. Since we perform one significance test for each rule, we can divide it by the number of total rules that we consider.
But then again, there can be false negatives (i.e: we might miss some interesting rules which are actually significant, but not passed the new significance threshold). Therefore, selecting the number to divide the significance threshold should be done with care, as the Bonferroni correction assumes that the individual tests are independent from each other.

(d)

1. Compute new significance threshold $= \frac{\text{old significance threshold}}{number of rules}$
$= \frac{0.05}{688}$
2. Interesting rules = rules whose p-value corresponding to chi squared score is less than equal to the new significance threshold.

(e)

The algorithm with chi squared score as the interestingness measure and **without** Bonferroni correction, found 606 rules to be significant and 82 rules to be non-significant.
The algorithm **with** Bonferroni correction, found 486 rules to be significant and 202 rules to be non-significant. Therefore, when Bonferroni correction is applied, the number of times the null hypothesis gets rejected among all the statistical tests, gets reduced.
However, we can not clearly say how many of them are false negatives (i.e: actually statistically significant, but considered to be non-significant) because we divided the significance threshold by total number of rules (i.e: number of statistical tests that we perform) which might not be independent from one another.
The top 30 rules remained the same as it was in (b).