Midterm

Satwik Singh satwiks2

Question 1

1. It’s a set of descriptive statistics of a dataset, which includes minimum, first quartile(25%), median, third quartile(75%) and maximum.
2. Boxplots are box-like diagrams that measures the dispersion of the dataset. A box plot is constructed from five values: the minimum value, the first quartile, the median, the third quartile, and the maximum value. We use these values to compare how close other data values are to them. The ends of the box are at the first and third quartiles, with median marked with a line inside the box. The height of the box represents the inter-quartile range. Two extensions from first and third quartile separately to connect the minimum and maximum, with points outside of the outlier filter plotted individually.
3. If two distribution have the same five-number summaries and outliers, then the boxplots can be the same. Boxplot only measures the dispersion of a dataset not the actual distribution and two different distributions can have the same dispersion.
4. A quantile plot displays all the data and plots the quantile information. The quantile plot permits identification of any peculiarities of the shape of the sample distribution, which might be symmetrical or skewed to higher or lower values.
5. A q-q plot is a plot of the quantiles of the first data set against the quantiles of the second data set. It is a graphical technique for determining if two data sets come from populations with a common distribution.
6. A quantile plot only shows the distribution of one dataset but a quantile-quantile plot compares the distribution and plots two different dataset.

Question 2

```
print("\nQ2 ---")
contingency_dat =np.array([[100,400],[300,200]])
contingency_dat_2 =np.array([[100,400],[300,20000]])

stat,p,ddof,expected = chi2_contingency(contingency_dat)
print("chi_stat: " + str(stat))
print("ddof: " + str(ddof))
alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')

stat,p,ddof,expected = chi2_contingency(contingency_dat_2)
print("chi_stat: " + str(stat))
print("ddof: " + str(ddof))
alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
```

1. $E_{buybeer,buydiaper} = \frac{count(BuyBeer)*count(BuyDiaper)}{total} = \frac{500*400}{1000} = 200$
2. $E_{buybeer,nobuydiaper} = \frac{count(BuyBeer)*count(NotBuyDiaper)}{total} = \frac{500*600}{1000} = 300$
3. Chi sq statistic = 165.004
    1. Create expected values table using formula:
       expected val = (row total * column total) / grand total
    2. Then use the expected value table to calculate chi squared statistic using:
       $\chi^2 = \sum \frac{(observed-expected)^2}{expected}$
4. Using code the p value is close to 0 so we reject the null hypothesis at significance level 0.05.
    1. If we want to do it by hand we get df = 1 and alpha = 0.05
    2. using lookup table we get critical value = 3.841
    3. since critical value < calculated chi square value we reject the null hypothesis hence we conclude that they are dependent.
5. Chi sq statistic = 877.818. Using the same process as above.
6. Using the same proces we still reject the null hypothesis since our critical value is still less than

the calculated chi square statistic hence we conclude that they are dependent.

Question 3

a. Frequent patterns

1. Min sup = 0.6.
   The most frequent k-itemset with largest k = the 3-itemset {B,C,D} with rel_sup = 3/5 = 0.6.
   1. {A}: 4/5, {B}: 3/5, {C}: 4/5, {D}: 4/5
   2. {A,C}: 3/5, {A,D}: 3/5
   3. {B,C,D}: 3/5.
      Therefore, the frequent k-itemset for max k=3 is {B,C,D}.
2. Strong Association rules:
   min sup = 0.6, the only set of 3-itemset that passes is {B,C,D}. Therefore, all associations between these 3 items:
   1. For $buys(x, B) \wedge buys(x, C) \Rightarrow buys(x, D)$
      - the support is 3/5
      - the confidence is $\frac{sup(B,C,D)}{sup(B,C)} = 0.6/0.6 = 1$
      - the confidence > minconf.
        $buys(x, B) \wedge buys(x, C) \Rightarrow buys(x, D)$. [60%,100%]

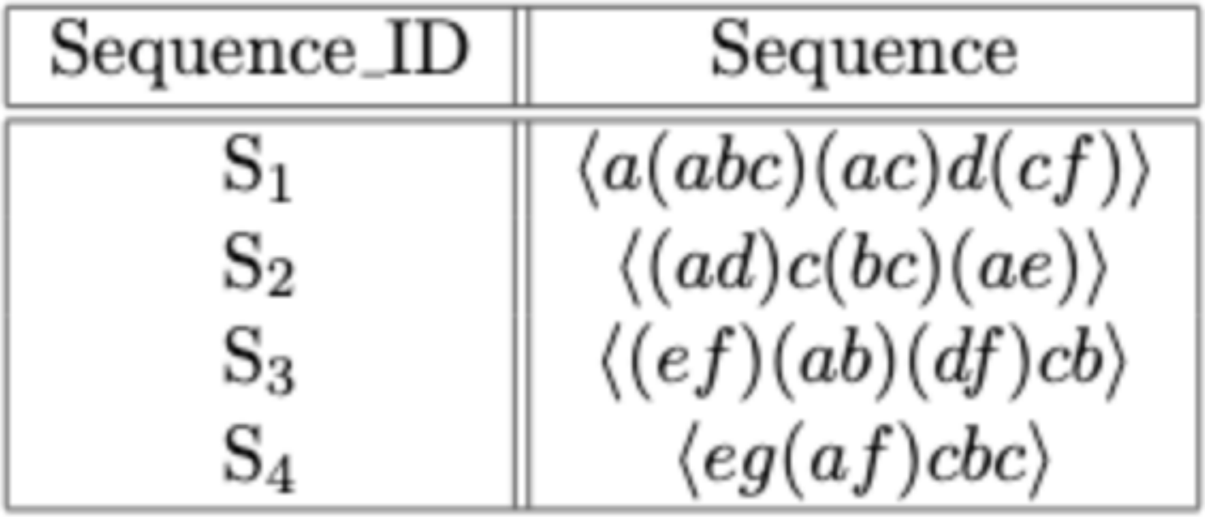
   2. For $buys(x, C) \wedge buys(x, D) \Rightarrow buys(x, B)$
      - the support is 3/5
      - the confidence is $\frac{sup(B,C,D)}{sup(C,D)} = 0.6/0.6 = 1$
      - the confidence > minconf.
        $buys(x, C) \wedge buys(x, D) \Rightarrow buys(x, B)$. [60%,100%]

   3. For $buys(x, D) \wedge buys(x, B) \Rightarrow buys(x, C))$
      - the support is 3/5
      - the confidence is $\frac{sup(B,C,D)}{sup(D,B)} = 0.6/0.6 = 1$
      - the confidence > minconf.
        $buys(x, D) \wedge buys(x, B) \Rightarrow buys(x, C)$. [60%,100%]

b. Constraint types

1. The average price of all the items in each pattern is greater than $50.
   - $c_1 : avg(S.price) > \$50$
   - This is a convertible constraint. It can be mined using FP-growth.
   - All the frequent items are listed in price descending order
   - A pattern as well as its conditional pattern base can be pruned, if the average price of items in the pattern is less than $50 and we've reached items priced less than $50 so the consequent items can't bring the average back above.
2. The sum of the price of all the items with profit over $5 in each patternis at least $200.
   - $c_1 : profit(S.price) > \$5, c_2 : sum(S.price) > \$200$
   - Constraint c1 is succinct and c2 is monotonic and data antimonotonic. It can be mined using FP-growth.
   - Put all items with profits more than $5 at the end of the list of frequent items so that they are pushed first in mining and we can easily prune.
   - Derive new FP-trees for frequent items with profit > $5 and mine recursively .
   - Once a pattern has sum at least $200, we can stop the recursion as it will only go up.
   - Patterns and conditional patterns can be pruned if sum of frequent items in the pattern is < $200 (data antimonotonocity).

Question 4



Sequence_ID	Sequence
S_1	$\langle a(abc)(ac)d(cf)\rangle$
S_2	$\langle (ad)c(bc)(ae)\rangle$
S_3	$\langle (ef)(ab)(df)cb\rangle$
S_4	$\langle eg(af)cbc\rangle$

Minsup = 3

1. ⟨a⟩ prefix, projected database:
    1. $\langle a \rangle$: $\langle (abc)(ac)d(cf) \rangle$, $\langle (\_d)c(bc)(ae) \rangle$, $\langle (\_b)(df)cb \rangle$, $\langle (\_f)cbc \rangle$
       Here the support comes out to be $\{\langle a \rangle : 2, \langle b \rangle : 4, \langle \_b \rangle : 2, \langle c \rangle : 4, \langle d \rangle : 2, \langle f \rangle : 2\}$ so we only take b and c forward in the pattern.
    2. $\langle ab \rangle$: $\langle (\_c)(ac)d(cf) \rangle, \langle (\_c)(ae) \rangle, \langle c \rangle$
       Here the support comes out to be $\{\langle a \rangle : 2, \langle \_c \rangle : 2, \langle c \rangle : 3, \langle d \rangle : 1, \langle f \rangle : 1\}$ so we only take c forward in the pattern.
    3. $\langle ac \rangle$: $\langle (ac)d(cf) \rangle, \langle (bc)(ae) \rangle, \langle bc \rangle$
       Here only c has support more than 3.
       So the patterns are: $\langle a \rangle, \langle ab \rangle, \langle ac \rangle, \langle abc \rangle, \langle acc \rangle$

2. ⟨c⟩ prefix, projected database:
    1. $\langle c \rangle$: $\langle (ac)d(cf) \rangle, \langle (bc)(ae) \rangle, \langle b \rangle, \langle bc \rangle$
       Here the support comes out to be $\{\langle a \rangle : 2, \langle b \rangle : 3, \langle c \rangle : 3, \langle d \rangle : 1, \langle e \rangle : 1, \langle f \rangle : 1\}$ so we only take b and c forward.
    2. $\langle cb \rangle$: $\langle (\_c)(ae) \rangle, \langle c \rangle$
       Here the support comes out to be $\{\langle a \rangle : 1, \langle c \rangle : 2, \langle \_c \rangle : 1, \langle e \rangle : 1\}$ so no more generation.
    3. $\langle cc \rangle$: $\langle d(cf) \rangle, \langle (ae) \rangle$
       Here the support comes out to be $\{\langle a \rangle : 1, \langle c \rangle : 1, \langle d \rangle : 1, \langle e \rangle : 1, \langle f \rangle : 1\}$ so no more generation.
       So the patterns are: $\langle c \rangle, \langle cb \rangle, \langle cc \rangle$

3. ⟨d⟩ prefix, projected database:
    1. $\langle d \rangle$: $\langle (cf) \rangle, \langle c(bc)(ae) \rangle, \langle (f)cb \rangle$, only c has support of 3
    2. $\langle dc \rangle$: $\langle (bc)(ae) \rangle, \langle b \rangle$ no more patterns.
       So the patterns are: $\langle d \rangle, \langle dc \rangle$

4. ⟨b⟩ prefix, projected database:
    1. $\langle b \rangle$: $\langle (\_c)(ac)d(cf) \rangle, \langle (\_c)(ae) \rangle, \langle (df)cb \rangle, \langle c \rangle$ only c has support of 3
    2. $\langle bc \rangle$: $\langle d(cf) \rangle, \langle b \rangle$ no more patterns.
       So the patterns are: $\langle b \rangle, \langle bc \rangle$

5. ⟨e⟩ prefix, projected database:

   1. $\langle e \rangle$: $\langle (\_f)(ab)(df)cb \rangle, \langle g(af)cbc \rangle$ no more patterns.
      So the patterns are: $\langle e \rangle$

6. ⟨f⟩ prefix, projected database:
   1. $\langle f \rangle$: $\langle (ab)(df)cb \rangle, \langle cbc \rangle$ no more patterns.
      So the patterns are: $\langle f \rangle$

7. No, since every prefix has been accounted for except g which didn't pass the minsup requirement in the first pass itself.