Data Mining Lab 5
Association Rules

Lavinia Pulcinella
Student ID: i6233926

October 25, 2020

1 Lab 5: Association Rules

Lavinia Pulcinella

Student ID: i6233926

1.1 Problem 1

1 Run the Apriori algorithm to generate all frequent itemsets from the 'T10I4D100K' data set at a support thresholds of 0.01%, 0.02% and 0.03%, and report the number of frequent itemsets so produced.

	0.01%	0.02%	0.03%
Reading T10I4D100K.dat	0.05s	0.05s	0.05s
Filtering, sorting and recoding items	0.00s	0.00s	0.00s
Sorting and reducing transactions	0.02s	0.02s	0.00s
building transaction tree	0.02s	0.01s	0.01s
checking subsets of size (. . .)	1.13s	0.73s	0.42s
writing < null >	0.04s	0.01s	0.01s
total time	**1.26s**	**0.82s**	**0.52s**

0.01 : checking subsets of size 1 2 3 4 5 6 7 8 9 10 11

0.02 : checking subsets of size 1 2 3 4 5 6 7 8 9 10

0.03 : checking subsets of size 1 2 3 4 5 6 7 8 9 10

Comment on the possible reason(s) for this difference in performance

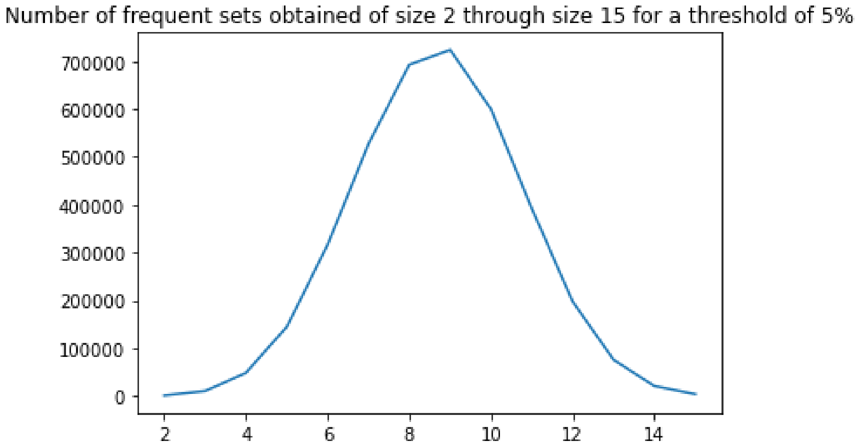
Answer: By highering the support threshold an higher number of itemsets is "ruled out" and thus the time required to run the apriori algorithm decreases as the threshold increases. Thus the time differences are due to the number of subsets that have to be checked for a given threshold as well as (at a smaller level) to the differences in output size.

2 Run Apriori (using the -ts option) on the 'mushroom' data set to generate frequent itemsets of sizes 2 through 15 at support thresholds of 5%, 10% and 20%.

Generate three plots, one for each threshold, showing the number of frequent sets obtained of size 2 through size 15. (To get the distribution of the number of frequent itemsets use option -Z).

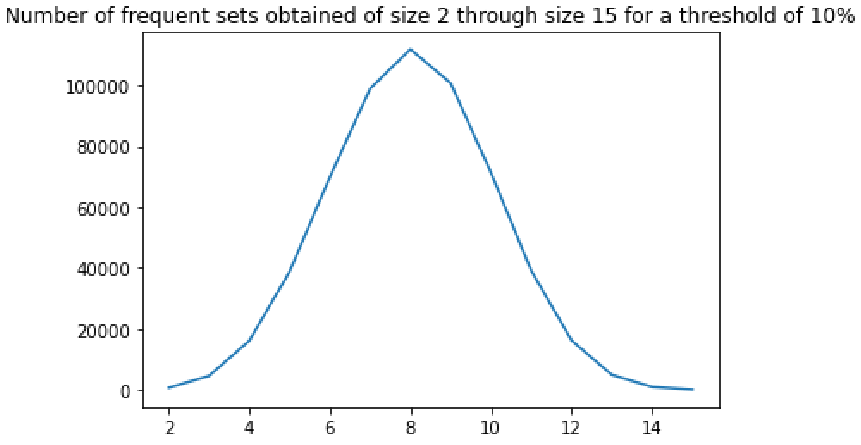
apriori -ts -s5 -m2 -n15 -Z mushroom.dat

[57]:
```
import matplotlib.pyplot as plt
x=[2, 3, 4, 5, 6, 7, 8, 9, 10 ,11, 12, 13,14,15]
y = [1329 , 10618, 48226, 144928, 315873, 527176, 692740, 723735, 600196,␣
 ↪391578, 197889 , 75624 , 21041 , 4000]
plt.plot(x, y)
plt.title('Number of frequent sets obtained of size 2 through size 15 for a␣
 ↪threshold of 5%')
plt.show()
```



apriori -ts -s10 -m2 -n15 -Z mushroom.dat

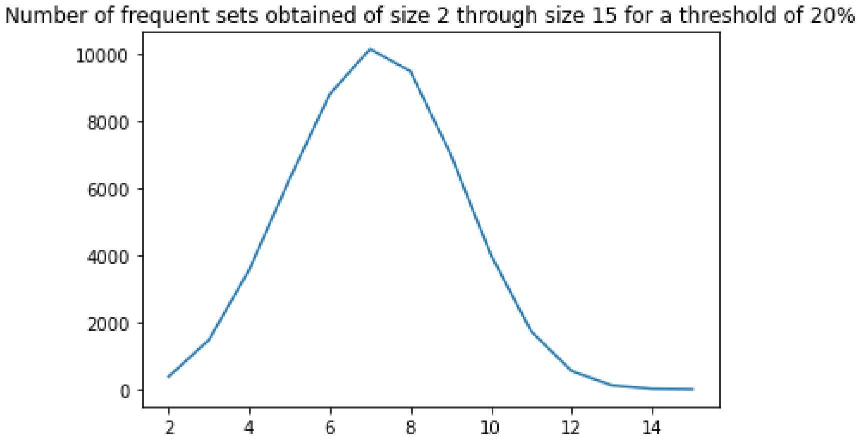
[58]:
```
x=[2, 3, 4, 5, 6, 7, 8, 9, 10 ,11, 12, 13,14,15]
y = [763, 4593, 16150, 38800, 69835, 98846, 111786, 100660, 71342, 39171, 16292,␣
 ↪4956,  1039, 134]
plt.plot(x, y)
plt.title('Number of frequent sets obtained of size 2 through size 15 for a␣
 ↪threshold of 10%')
plt.show()
```



apriori -ts -s20 -m2 -n15 -Z mushroom.dat

[59]:

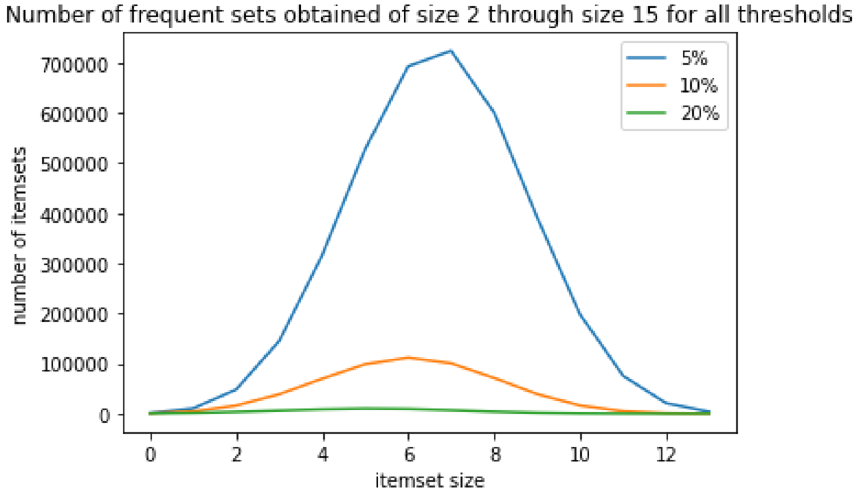
```
x=[2, 3, 4, 5, 6, 7, 8, 9, 10 ,11, 12, 13,14,15]
y = [376, 1472, 3559,  6267 , 8802, 10151, 9488, 7010, 4004, 1729, 546, 119, 16,␣
 ↪ 1]
plt.plot(x, y)
plt.title('Number of frequent sets obtained of size 2 through size 15 for a␣
 ↪threshold of 20%')
plt.show()
```



[61]:

```
itemsets = pd.read_csv('plot.csv', sep=';', header = 1)
itemsets.plot(y=['5%', '10%', '20%'])
plt.xlabel('itemset size')
plt.ylabel('number of itemsets')
plt.title('Number of frequent sets obtained of size 2 through size 15 for all␣
 ↪thresholds')
```

```
plt.show()
```



Comment on the general trends illustrated by the plots, and comment on the reason(s) for these trends.

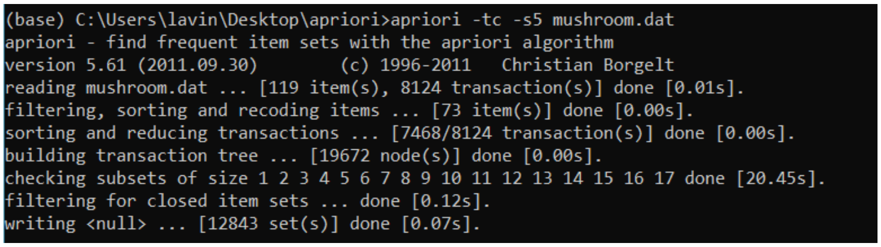
Also comment on how the plots vary between the three thresholds.

Answer: For all three thresholds, the plot of the number of frequent itemsets as a function of the itemset's size have a bell-shaped curve. However, by plotting them toghether we see that for a lower threshold (of 5%) the bell-shaped curve is much more evident since it contains (on average) a much higher number of itemsets per itmeset size. In general, as the support threshold increases a smaller number of itemsets is found.

3 Use the Apriori algorithm to generate closed (using the -tc option) and maximal (using the -tm option) frequent itemsets from the 'T10I4D100K' and 'mushroom' data sets. Use a support threshold of 5% for the 'mushroom' data set and 0.01% for the 'T10I4D100K' data set.

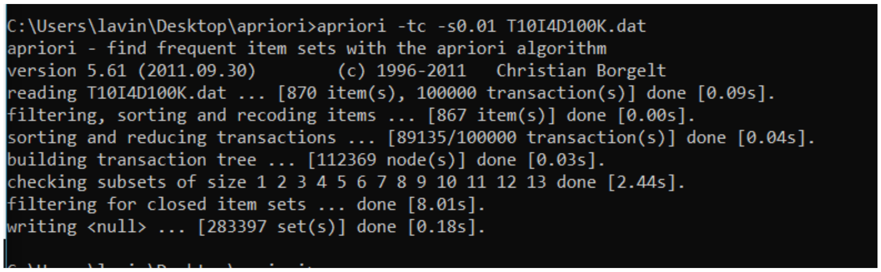
- tc closed itemsets

For mashroom s 5% : apriori -tc -s5 mushroom.dat



```
(base) C:\Users\lavin\Desktop\apriori>apriori -tc -s5 mushroom.dat
apriori - find frequent item sets with the apriori algorithm
version 5.61 (2011.09.30)        (c) 1996-2011   Christian Borgelt
reading mushroom.dat ... [119 item(s), 8124 transaction(s)] done [0.01s].
filtering, sorting and recoding items ... [73 item(s)] done [0.00s].
sorting and reducing transactions ... [7468/8124 transaction(s)] done [0.00s].
building transaction tree ... [19672 node(s)] done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 done [20.45s].
filtering for closed item sets ... done [0.12s].
writing <null> ... [12843 set(s)] done [0.07s].
```

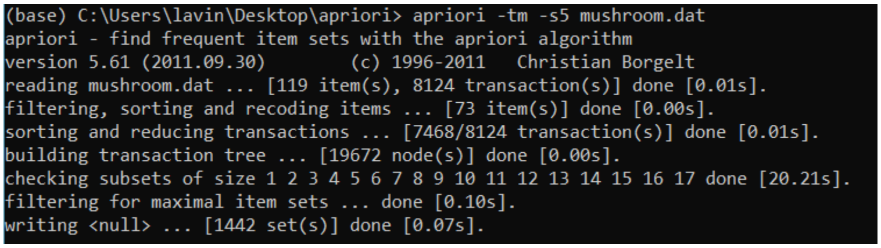
For 'T10I4D100K' s 0.01% : apriori -tc -s0.01 T10I4D100K.dat



```
C:\Users\lavin\Desktop\apriori>apriori -tc -s0.01 T10I4D100K.dat
apriori - find frequent item sets with the apriori algorithm
version 5.61 (2011.09.30)        (c) 1996-2011   Christian Borgelt
reading T10I4D100K.dat ... [870 item(s), 100000 transaction(s)] done [0.09s].
filtering, sorting and recoding items ... [867 item(s)] done [0.00s].
sorting and reducing transactions ... [89135/100000 transaction(s)] done [0.04s].
building transaction tree ... [112369 node(s)] done [0.03s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 11 12 13 done [2.44s].
filtering for closed item sets ... done [8.01s].
writing <null> ... [283397 set(s)] done [0.18s].
```

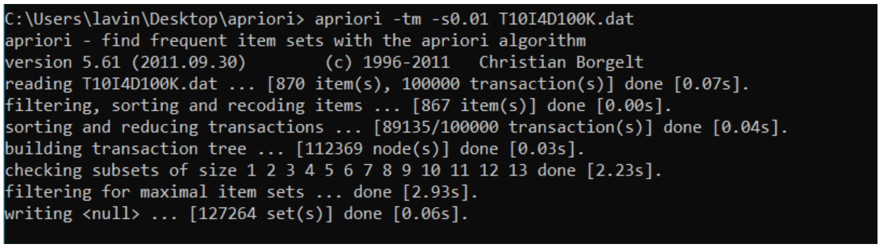
- tm maximal itemsets

For mashroom s 5% : apriori -tm -s5 mushroom.dat



```
(base) C:\Users\lavin\Desktop\apriori> apriori -tm -s5 mushroom.dat
apriori - find frequent item sets with the apriori algorithm
version 5.61 (2011.09.30)        (c) 1996-2011   Christian Borgelt
reading mushroom.dat ... [119 item(s), 8124 transaction(s)] done [0.01s].
filtering, sorting and recoding items ... [73 item(s)] done [0.00s].
sorting and reducing transactions ... [7468/8124 transaction(s)] done [0.01s].
building transaction tree ... [19672 node(s)] done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 done [20.21s].
filtering for maximal item sets ... done [0.10s].
writing <null> ... [1442 set(s)] done [0.07s].
```

For ‘T10I4D100K’ s 0.01% : apriori -tm -s0.01 T10I4D100K.dat



```
C:\Users\lavin\Desktop\apriori> apriori -tm -s0.01 T10I4D100K.dat
apriori - find frequent item sets with the apriori algorithm
version 5.61 (2011.09.30)        (c) 1996-2011   Christian Borgelt
reading T10I4D100K.dat ... [870 item(s), 100000 transaction(s)] done [0.07s].
filtering, sorting and recoding items ... [867 item(s)] done [0.00s].
sorting and reducing transactions ... [89135/100000 transaction(s)] done [0.04s].
building transaction tree ... [112369 node(s)] done [0.03s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 11 12 13 done [2.23s].
filtering for maximal item sets ... done [2.93s].
writing <null> ... [127264 set(s)] done [0.06s].
```

In summary:

Data set	-tc (closed)	-tm (maximal)	-ts (frequent)
T10I4D100K	283 397	127 264	411 365
mushroom	12 843	1 442	3 755 511

- **Compare the total number of closed and maximal frequent itemsets obtained for each dataset individually.**

  Answer: In general a closed itemset is set of items which is as large as it can possibly be without losing any transactions while a maximal frequent one is a frequent itemset which is not contained in another frequent itemset. Thus, unlike closed itemsets, maximal itemsets do not imply anything about transactions.

  In practice, the maximal itemsets are contained in the closed itemsets. That is why in the above table we see a smaller number of sets included in the maximal itesmsets than for closed itemsets.

- **How do these numbers compare with the number of frequent itemsets obtained from these data sets using the same threshold?**

  Answer: A frequent itemset is simply a set of items occurring a certain percentage of the time. Thus they can be considered as the general set including all sets of items. Thus both closed and maximal itemsets are included in the frequent itemsets.

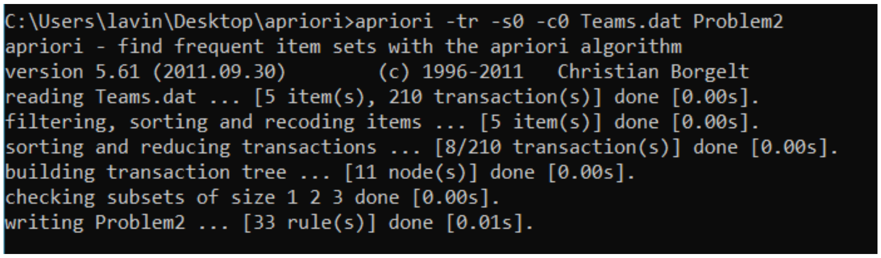
- **What relationship among closed, maximal and frequent itemsets is revealed by this comparison?**

  Answer: In general, frequent itemsets are of the largest size, closed itemsets the second largest and maximal ones the smaller size.

1.2 Problem 2

This question uses the data set 'Teams' that is based on a team A's performance against two teams B and C. The data set is in the file Teams.dat. The data set contains the following items:

1. Team A plays against Team B,
2. Team A plays against Team C,
3. Game is played at Team A's home,
4. Game is played away from Team A's home,
5. Game is won by Team A



```
C:\Users\lavin\Desktop\apriori>apriori -tr -s0 -c0 Teams.dat Problem2
apriori - find frequent item sets with the apriori algorithm
version 5.61 (2011.09.30)        (c) 1996-2011   Christian Borgelt
reading Teams.dat ... [5 item(s), 210 transaction(s)] done [0.00s].
filtering, sorting and recoding items ... [5 item(s)] done [0.00s].
sorting and reducing transactions ... [8/210 transaction(s)] done [0.00s].
building transaction tree ... [11 node(s)] done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing Problem2 ... [33 rule(s)] done [0.01s].
```

[38]:

```
import pandas as pd
problem2 = pd.read_csv("Problem2.txt", header = None, sep = "\t", names =␣
 ↪['rules'])
problem2
```

[38]:

```
                  rules
0    3 <-  (100.0, 33.3)
1    1 <-  (100.0, 47.6)
2    5 <-  (100.0, 42.9)
3    2 <-  (100.0, 52.4)
4    4 <-  (100.0, 66.7)
5    1 <- 3 (33.3, 14.3)
6    3 <- 1 (47.6, 10.0)
7    5 <- 3 (33.3, 67.1)
8    3 <- 5 (42.9, 52.2)
```

```
9     2 <- 3 (33.3, 85.7)
10    3 <- 2 (52.4, 54.5)
11    5 <- 1 (47.6, 40.0)
12    1 <- 5 (42.9, 44.4)
13    4 <- 1 (47.6, 90.0)
14    1 <- 4 (66.7, 64.3)
15    2 <- 5 (42.9, 55.6)
16    5 <- 2 (52.4, 45.5)
17    4 <- 5 (42.9, 47.8)
18    5 <- 4 (66.7, 30.7)
19    4 <- 2 (52.4, 45.5)
20    2 <- 4 (66.7, 35.7)
21   5 <- 3 1 (4.8, 70.0)
22  1 <- 3 5 (22.4, 14.9)
23  3 <- 1 5 (19.0, 17.5)
24  2 <- 3 5 (22.4, 85.1)
25  5 <- 3 2 (28.6, 66.7)
26  3 <- 5 2 (23.8, 80.0)
27  4 <- 1 5 (19.0, 82.5)
28  5 <- 1 4 (42.9, 36.7)
29  1 <- 5 4 (20.5, 76.7)
30  4 <- 5 2 (23.8, 20.0)
31  2 <- 5 4 (20.5, 23.3)
32  5 <- 2 4 (23.8, 20.0)
```

[43]:

```python
freqitemset = pd.read_csv("P2Freq.txt", header=None, names = ["Frequent Itemsets"])
freqitemset
```

[43]:

```
   Frequent Itemsets
0           3 (33.3)
1          3 1 (4.8)
2        3 1 5 (3.3)
3         3 5 (22.4)
4       3 5 2 (19.0)
5         3 2 (28.6)
6           1 (47.6)
7         1 5 (19.0)
8       1 5 4 (15.7)
9         1 4 (42.9)
10          5 (42.9)
11        5 2 (23.8)
12       5 2 4 (4.8)
13        5 4 (20.5)
14          2 (52.4)
15        2 4 (23.8)
16          4 (66.7)
```

In general we know that

$conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} = \frac{P(X,Y)}{P(X)} = P(Y|X)$

The above rules are of the form $Y \leftarrow X(support, confidence)$

The confidence here looks like it is $P(Y|X) \times 100$ (see below)

a. *Find the probability of Team A winning against Team B (Pr[Won_By_A | Team_B]). Also find the probability of Team A winning against Team C (Pr[Won_By_A | Team_C]). Compare these probabilities.*

- $P(A_{win}|B) = conf(B \rightarrow A_{win})$ which according to the encoding of the Teams dataset corresponds to $conf(1 \rightarrow 5)$: **5 ← 1 (47.6, 40.0)**

  By computing only frequent itemsets we can obtain the following probability:

  $conf(1 \rightarrow 5) = P(A_{win}|B) = \frac{supp(1 \cup 5)}{supp(1)} = \frac{19}{47.6} = 0.4$

- $P(A_{win}|C) = conf(C \rightarrow A_{win})$ which according to the encoding of the Teams dataset corresponds to $conf(2 \rightarrow 5)$: ** 5 ← 2 (52.4, 45.5)**

  Similarly:

  $conf(2 \rightarrow 5) = P(A_{win}|C) = \frac{supp(2 \cup 5)}{supp(2)} = \frac{23.8}{52.4} = 0.45$

Thus by just looking at the confidence values of the above rules, where the confidence value refers to how ofthen item in Y appears in "transactions" that contain X, we can see that A has higher chances to win against team C since P(A win | C) has a confidence of 45.5 (which is higher than P(A win | B)s one which is 40).

Thus, a higher confidence value is also a consequence of the fact that the event of A winning *appears more times* when team A plays against team C rather than team B.

b. *Compare the probabilities of Team A winning against Team B and C at a home venue (Pr[Won_by_A | Team_B,Home] and Pr[Won_by_A | Team_C,Home] respectively). State the rules that you based your comparison upon*

- $P(A_{win}|B, Home) = conf(B, Home \rightarrow A_{win}) = conf(1, 3 \rightarrow 5)$:

  5 ← 3 , 1 (4.8, 70.0)

- $P(A_{win}|C, Home) = conf(C, Home \rightarrow A_{win}) = conf(2, 3 \rightarrow 5)$:

  5 ← 3 , 2 (28.6, 66.7)

By a similar reasoning to point a. in this case the confidence value is higher for A winning against team B while being in a home venue (for A). In particular:

$P(A_{win}|B, Home) = P(5|1, 3) = 0.7$

$P(A_{win}|C, Home) = P(5|2, 3) = 0.66$

c. *Compare the probabilities of Team A winning against Team B and C when the games are played away from A's home (Pr[Won_by_A | Team_B,Away] and Pr[Won_by_A | Team_C,Away] respectively). State the rules that you based your comparison upon.*

- $P(Awin|B, Away) = conf(B, Away \rightarrow Awin) = conf(1, 4 \rightarrow 5)$:

5 ← 1 4 (42.9, 36.7)

- $P(Awin|C, Away) = conf(C, Away \rightarrow Awin) = conf(2, 4- \rightarrow 5)$:

  5 ← 2 4 (23.8, 20.0)

Confidence value is higher for A winning against B while being away from a home venue. In particular

$P(A_{win}|B, Away) = P(5|1, 4) = 0.367$

$P(A_{win}|C, Away) = P(5|2, 4) = 0.20$

d. *Are the results in (b) and (c) consistent with those in (a)? Explain why or why not.*

According to the law of total probability the results obtained in (b) and (c) are consistent with those in (a). In particular :

$P(A_{win}|B) = P(A_{win}|B, Home) \times P(Home|B) + P(A_{win}|B, Away) \times P(Away|B)$

where

$P(Home|B) = P(3|1) = \frac{4.8}{47.6} = 0.1$ & $P(Away|B) = P(4|1) = \frac{42.9}{47.6} = 0.9$

Thus:

$(0.7 \times 0.1) + (0.367 \times 0.9) = 0.4 = P(A_{win}|B)$

$P(A_{win}|C) = P(A_{win}|C, Home) \times P(Home|C) + P(A_{win}|C, Away) \times P(Away|C)$

where

$P(Home|C) = P(3|2) = \frac{28.6}{52.4} = 0.55$ & $P(Away|C) = P(4|2) = \frac{23.8}{52.4} = 0.45$

Thus,

$(0.66 \times 0.55) + (0.2 \times 0.45) = 0.45 = P(A_{win}|C)$