Data Mining Assignment - Pattern Mining

Nick Wils

March 16, 2022

Abstract

You can find the code of my project in the following link: github_link

1 Question 1: An association rule mining algorithm

To find the association rules I used an existing online implementation of the Apriori algorithm in python. I used the 'apyori' library and used an function to visualize the results [1].

1.1 Concepts of the Apriori algorithm

To calculate and use the apriori algorithm there are some general concept we need to understand.

1.1.1 Support

The support of an itemset is in how many transactions item A occurs. It is calculated by dividing the amount transactions with itemset A in by the amount of transactions in your data set.

$$\textbf{Support}(itemset) = \frac{\#\text{ transactions with itemset A}}{\#\text{ transactions}}$$

1.1.2 Confidence

Lets say you have a rule $A \rightarrow B$.Than the confidence can be defined as how much of the time itemset B occurs in the transaction if itemset A is in it. It is calculated by the amount of transactions with itemset A and B, divided by the amount of transactions with itemset A.

$$\textbf{Confidence}(A \rightarrow B) = \frac{\#\text{ transactions with itemset A and B}}{\#\text{ transactions with itemset A}}$$

1.1.3 Lift

Lets say you again have a rule $A \rightarrow B$. The lift shows how likely it is that itemset A and B are in the transaction instead of only itemset B

- If the lift is greater then 1, then the itemsets are more likely to be in the same transactions.
- If the lift is less then 1, the products are unlikely to be in the same transactions.
- If the lift is 1, there is no correlation between the itemsets.

The lift is calculated by divideing the cofidence of $A \rightarrow B$ by the support of B.

$$\textbf{Lift}(A \rightarrow B) = \frac{\text{Confidence}(A \rightarrow B)}{\text{Support(B)}}$$

1.2 The Apriori algorithm

The algorithm is based on the idea that all subsets of a frequent itemset must also be frequent. The algorithm needs two variables to work. The minimum support threshold and the minimum confidence threshold. Without the minimum support threshold all the possible rules will be generated, that are 3^N rules (with N as the amount of items). The minimum confidence threshold is there to limit the amount of rules that is generated from the frequent itemsets. The algorithm will first find all the itemsets that have a big enough support.

This will work as follows:

1. It will start by finding all itemsets of size 1, that means that they contain only 1 item.
2. It will calculate the support of these itemsets and only keep those that have a support greater than the minimum support threshold.
3. From this reduced list it will find all the combinations that lead to an itemset of size 2.
4. It will now search for the support of these new itemsets and again only keep those that have a support greater than the minimum support threshold.
5. It will repeat these last two steps with each time a bigger itemset, until it finds no new itemsets.

With all these frequent itemsets the algorithm can generate the association rules that have a confidence level higher then the minimum confidence threshold.

2 Question 2: Top 10 association rules in terms of support

I ran the Apriori algorithm with the following parameters:

- minimum support threshold = 0.0004, we want the top ten of this sow we have to get at least ten but not necessarily more.
- minimum confidence threshold = 0, we just want the top ten of the support, sow we don't want to filter on confidence.
- minimum lift threshold = 1.1, we only want rules with a positive correlation.
- minimum itemsize = 2, we want a rule sow we need a minimum itemsize of 2.

With these parameters I got the following top ten support rules (where in a rule $A \rightarrow B$, A is noted as Itemset left and B as Itemset right):

Rank	**Itemset left**	**Itemset right**	**Support**	**Confidence**
1	5677043	5697463	0.004187	0.329341
2	5814516	5814517	0.003730	0.875000
3	5809910	5809912	0.003730	0.162791
4	5809910	5809911	0.003578	0.156146
5	5886282	5909810	0.002284	0.352941
6	5649235	5649236	0.002284	0.352941
7	5751383	5751422	0.002284	0.288462
8	5886282	5892179	0.002131	0.329412
9	5649236	5649271	0.002055	0.143617
10	5886282	5900651	0.001979	0.305882

Table 1: Top ten support rules.

To get the amount of transactions where the rule occurs you can multiply the support with the total number of transactions which is 13137.

The length of each rule in the top ten support is 2, as you can see in 'Table 1'. This is no surprise as the support is simply the the amount of transactions in which this rule occurs and the minimum length is 2. I will explain this with an example:

Lets say you have a rule $A \rightarrow B$ that occurs in 20 transactions. If we make this rule bigger rule, lets say $A \rightarrow B, C$. That would mean that you add an extra restriction. Rule $A \rightarrow B$ still needs to hold and on top of that rule $A \rightarrow C$ needs to hold. This means that the amount of transactions where rule $A \rightarrow B, C$ occurs is upper-bound by 20.

So the only way that a bigger rule like $A \rightarrow B, C$ would be in this table is when both rules: $A \rightarrow B$ and $A \rightarrow c$ are on the table and $A \rightarrow B, C$ occurs more than other rules. Which are some pretty strong conditions.

3 Question 3: The 10 most interesting rules

To determine the 10 most interesting rules it is important to know what your goal is. Are you trying to advertise items to costumers? Or are you trying to find which items should be sold in a package deal? Or even are you trying to find out which items should not be sold together?

3.0.1 First interesting rules

I chose to find items with that should be sold together. To do this tried to find a good balance between a high confidence and a high level.

- A high confidence tells us that there are more transactions with A and B than there are with only A.
- A high level tells us that there are more transactions with A and B than there are with only B

. With carefully raising the minimum confidence threshold and sorting on level in descending order.

- minimum support threshold = 0.0004, its still important that the items are sold frequently enough.
- minimum confidence threshold = 0.7, its important that the items are more bought together than that only itemset A was bought.
- minimum lift threshold = 1.1, this is the parameter i will maximize, sow it doesn't matter how low this is.
- minimum itemsize = 2, we want a rule sow we need a minimum itemsize of 2.

With these parameters I got 38 rules, from which you can see the top 10 in 'Table 2'.

Rank	**Itemset left**	**Itemset right**	**Support**	**Confidence**	**Level**
1	5883311	5848309	0.000457	0.857143	1126.028571
2	5899427	5899430	0.000457	0.857143	1023.662338
3	5827356	5827357	0.000913	0.923077	1010.538462
4	5853036	5853038	0.000533	0.700000	919.590000
5	5808664	5808663	0.000533	0.777778	851.472222
6	5738962	5738965	0.000533	0.700000	835.990909
7	5784998	5784997	0.000609	0.727273	682.441558
8	5757450	5757469	0.000457	0.750000	656.850000
9	5855094	5855093	0.000761	0.769231	631.586538
10	5851710	5851708	0.000457	0.750000	615.796875

Table 2: Top ten level rules.

3.0.2 Second interesting rules

After trying some variables i found that there are some rules with 100 percent confidence. This means that itemset A was never bought without itemset B. Which is of course very interesting to know if you want to advertise itemset A.

To get 10 rules with confidence level of 100 percent I had to lower the minimum support a bit. The rules with the highest support among all the rules with confidence level 100 percent, are the most interesting.

To get this I used the following parameters:

- minimum support threshold = 0.0004, its still important that the items are sold frequently enough.
- minimum confidence threshold = 0.7, its important that the items are more bought together than that only itemset A was bought.
- minimum lift threshold = 1.1, this is the parameter i will maximize, sow it doesn't matter how low this is.
- minimum itemsize = 2, we want a rule sow we need a minimum itemsize of 2.

With these parameters I got 26 rules, from which you can see the top 10 in 'Table 3'.

Rank	**Itemset left**	**Itemset right**	**Support**	**Confidence**	**Level**
1	5814516	5814517	0.001446	1.0	230.473684
2	5827357	5827356	0.000913	1.0	1010.538462
3	5707826	5692527	0.000685	1.0	110.394958
4	5805492	5805493	0.000457	1.0	597.136364
5	5678118	5677420	0.000381	1.0	1642.125000
6	5814515	5814518	0.000381	1.0	547.375000
7	5880203	5880205	0.000381	1.0	2627.400000
8	5910964	5910963	0.000381	1.0	1459.666667
9	5814515	5814517	0.000304	1.0	230.473684
10	5814516	5814517	0.000304	1.0	230.473684

Table 3: Top ten level rules.

It is important to note that the support of most of these rules is very low, which can explain the extreme confidence level.

4 Question 4: Additional information

Now we have all the information, we can chance the transactions to give us more interesting rules. First i changed the way how we made the baskets. I split the product_ids in two parts, the purchased an viewed products. I added a prefix to separate them. Instead of grouping by user, I chose to group by session id. As this a user might by products multiple times on different dates. Items that are added or removed from the shopping cart are less interesting, as this information is already visible in viewed and purchased products. I also added the brands and categories to the transactions. I grouped them by user as this would give an insight in the person.

5 Question 5: Additional information results

The rules i would deem most interesting are still those with a high level and confidence. As those give a very high correlation. I used the following parameters

- minimum support threshold = 0.0004
- minimum confidence threshold = 0.3
- minimum lift threshold = 1.1
- minimum itemsize = 2

To get the following results:

Rank	Itemset left	Itemset right	Support	Confidence	Level
1	purchase-5819894	purchase-5823667	0.000473	0.821429	1080.267375
2	purchase-5814516	purchase-5814517	0.000925	0.882353	810.083241
3	purchase-5677043	purchase-5697463	0.000966	0.321918	237.336343
4	view-5900651	view-5886282	0.000473	0.403509	178.493939
5	purchase-5751422	purchase-5751383	0.000514	0.304878	174.530129
6	purchase-5809911	purchase-5809910	0.000760	0.787234	160.276240
7	purchase-5809912	purchase-5809910	0.000740	0.620690	126.368778
8	1487580013858390233	1487580013841613016	0.000801	0.619048	69.087702
9	1487580013539623112	1487580013950664926	0.000843	0.312977	25.213829
10	1783999067156644376	1487580008246412266	0.003206	0.543554	23.784887

Table 4: Top ten level rules additional information.

References

[1] Naibei, Benjamin (2021) *Getting started with Apriori algorithm in Python*, Section, `https://www.section.io/engineering-education/apriori-algorithm-in-python/`.

[2] Leslie Lamport (1994) *LaTeX: a document preparation system*, Addison Wesley, Massachusetts, 2nd ed.