Getting Started with ECLAT Algorithm in Association Rule Mining

December 22, 2021

Topics: Machine Learning

The goal in data mining is to find rules that predict an item based on the occurrence of other items in the transactions.

Generally, to achieve this, we have two approaches;

1. Generating the frequent itemsets. Here, we generate all itemsets with greater support than the adopted minimum support.
2. Rule Generation. We generate a frequent high set from the frequent itemsets.

Prerequisites:

To follow along with this article, the learner needs to be comfortable working with the following libraries in Python: Scikit-learn, Pandas, and Numpy.

Introduction to ECLAT Algorithm

As we mentioned before, the main idea in the association rule is to discover valid information and knowledge from a large dataset. Several algorithms have been developed over the years that make this activity as successful as possible. The major algorithm used includes:

3. FP Growth Algorithm

The first algorithm to be introduced in the data mining domain was the Apriori algorithm. However, this algorithm had some limitations in discovering frequent itemsets. Its limitations created a need for a more efficient algorithm.

Later, the Eclat algorithm was introduced to deal with the weakness of the Apriori algorithm. Between these two algorithms, we have significant differences:

- Unlike the Apriori algorithm, which is applicable with *horizontal dataset*, the Eclat algorithm is applicable only with a dataset in *vertical dataset* format.
- In the Eclat algorithm, only the *support* and confidence is counted as confidence. As in the case of Apriori, it is not computed. Here, the *Support* is nothing but the number of times an item is in a database.

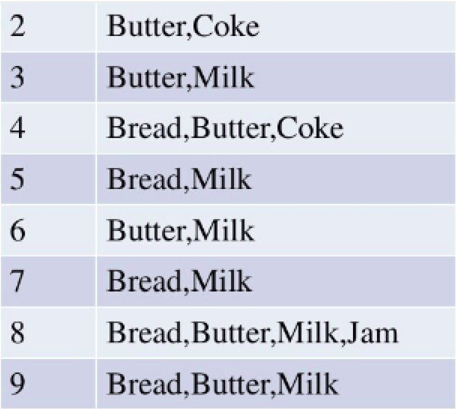
At each stage of the generated database, the Eclat algorithm uses the current generated dataset to learn frequent itemset, unlike the Apriori which scans the original database repeatedly. Since the Eclat scans over the database once, it is much faster than the Apriori algorithm.

However, this doesn't mean the Apriori algorithm is worse. On the contrary, when dealing with a larger dataset, Apriori tends to shine best. Thus, the Eclat algorithm works better with small and medium datasets.

The key takeaway here is that Eclat works well with the vertical data format. Since most datasets are in the horizontal format, to apply the Eclat algorithm, we first have to convert them to vertical format.

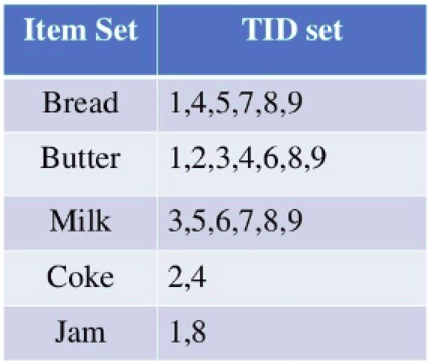
Below are examples of Horizontal and verticle data formats.

Horizontal data format:



2	Butter,Coke
3	Butter,Milk
4	Bread,Butter,Coke
5	Bread,Milk
6	Butter,Milk
7	Bread,Milk
8	Bread,Butter,Milk,Jam
9	Bread,Butter,Milk

Verticle data format:



Item Set	TID set
Bread	1,4,5,7,8,9
Butter	1,2,3,4,6,8,9
Milk	3,5,6,7,8,9
Coke	2,4
Jam	1,8

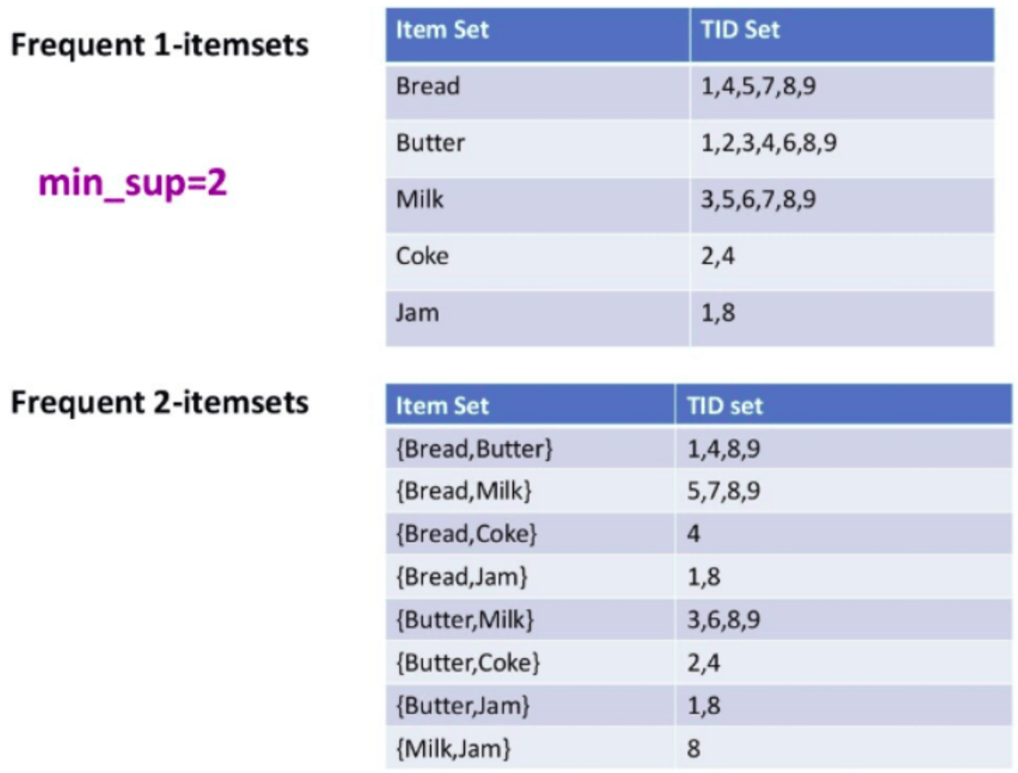
We obtain a verticle data format by making a list of transactions to which each particular item is found.

Let’s look at the steps in the Eclat algorithm.

Eclat Algorithm

1. Get tidlist for each item in the database. Here, we scan the entire database. The tidlist of item {a} is the list of transactions in which item {a} is contained.
2. Intersect the tidlist of item {a} with the tidlist of item {b} and generate a new transaction list whose elements are transactions in which both items {a} and {b} are found.
3. Repeat step 1 on {a}-conditional to other items in the database.

should obtain an output similar to the one below:



Frequent 1-itemsets

min_sup=2

Item Set	TID Set
Bread	1,4,5,7,8,9
Butter	1,2,3,4,6,8,9
Milk	3,5,6,7,8,9
Coke	2,4
Jam	1,8

Frequent 2-itemsets

Item Set	TID set
{Bread,Butter}	1,4,8,9
{Bread,Milk}	5,7,8,9
{Bread,Coke}	4
{Bread,Jam}	1,8
{Butter,Milk}	3,6,8,9
{Butter,Coke}	2,4
{Butter,Jam}	1,8
{Milk,Jam}	8

Advantages

1. Eclat algorithm has low memory requirements compared to Apriori as it uses the Depth-First Search approach.
2. The Eclat algorithm does not repeatedly scan the data to discover frequent itemsets, thus, is generally faster than the Apriori algorithm.
3. Eclat algorithm outdoes the Apriori algorithm provided the dataset is not too big.
4. Eclat algorithm scans only the currently generated dataset that is scanned in the Eclat algorithm. This is unlike in Apriori where the original dataset is scanned at each stage.

Disadvantages

If the tidlist is too large, the Eclat algorithm may run out of memory.

Let’s proceed and implement this algorithm in python.

To have the best rules, we wull adopt the Apriori algorithm in our implementation. To get started, we need to import the necessary libraries for this session.

```
# The first thing is to install this package
!pip install apyori
```

Data Preprocessing

Importing the libraries

The libraries we will work with for this session are as follows:

```
import numpy as np # to deal with numeric data
import pandas as pd # to deal with dataframe
```

Dataset importation

```
data = pd.read_csv('/content/drive/MyDrive/Market_Basket_Optimisation.csv', h
transact_list = [] # create an empty list to store transactions
for i in range(0, 7501):
  transact_list.append([str(data.values[i,j]) for j in range(0, 20)]) # add t
```

Eclat Algorithm

Since we adopted the Apriori algorithm, we need to generate rules as follows:

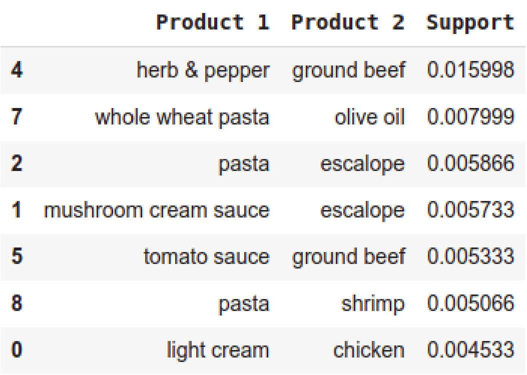
```
from apyori import apriori # import the apriori library
rules = apriori(transactions = transact_list, min_support = 0.003, min_confi

# list of results coming from the apriori model
rslt = list(rules)
```

Organizing the above output in a pandas dataframe

```
def inspect(rslt): # function to organize the output
    left_handSide         = [tuple(result[2][0][0])[0] for result in rslt] #
    right_handSide         = [tuple(result[2][0][1])[0] for result in rslt] #
    supports    = [result[1] for result in rslt] # get the supports
    return list(zip(left_handSide,right_handSide, supports)) # zip the above
rslt_DataFrame = pd.DataFrame(inspect(rslt), columns = ['Product 1', 'Product
```

Now that our output is organized in a pandas dataframe, we can have a look at the first seven supports as follows:

```
rslt_DataFrame.nlargest(n = 7, columns = 'Support') # printing the first 7 su
```



	Product 1	Product 2	Support
4	herb & pepper	ground beef	0.015998
7	whole wheat pasta	olive oil	0.007999
2	pasta	escalope	0.005866
1	mushroom cream sauce	escalope	0.005733
5	tomato sauce	ground beef	0.005333
8	pasta	shrimp	0.005066
0	light cream	chicken	0.004533

The rule (herb & pepper) has the highest support from the output above. The second rule with the highest support is the (whole wheat pasta with olive oil) and so on. This table means that the first rule is the most important. Therefore, for the seller to maximize their sales and profit, they should adopt the first rule.

The link to the source code is provided here.

Conclusion

We mentioned the top three algorithms mainly used in this field to discover knowledge from the data: Apriori, Eclat, and FP growth algorithms. First, we saw the limitation of the Apriori algorithm, and later, we were able to discuss how the Eclat algorithm solves this.

We also discussed how these two algorithms differ before talking about the pros and cons of the Eclat algorithm. Later, we implemented the Eclat algorithm by adapting the Apriori algorithm to improve accuracy.

Peer Review Contributions by: Willies Ogola

Did you find this article helpful?

5

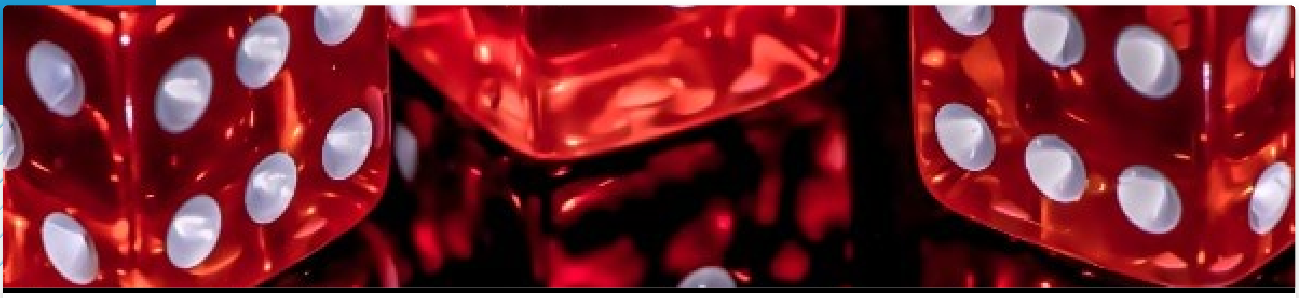
0 Comments

The EngEd community is subject to Section's moderation policy.

Be the first to comment...

L

Similar Articles



Machine Learning

Machine Learning

Getting Started with Kernel PCA in Python

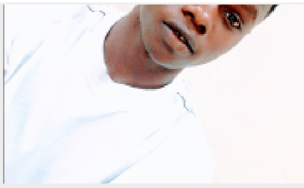
Read More →

Machine Learning, Languages

System of Linear Equations using Gaussian Elimination Algorithm

Read More →

EngEd Author Bio



Jackson Munyai

Jackson Munyai is a second-year Computer Science student at Chuka University, Kenya. Jackson is mainly interested in A.I. and data science. He likes visiting new places, playing video games, and hanging out with friends.

View author's full profile →

varnish cache logo cloud native computing foundation logo the linux foundation logo lf edge logo

Section Privacy Policy Terms of Service