Association Rule Mining

Aakash Hemadri <17pw01@psgtech.ac.in>

Objective

- Goal is to understand the usage of the association rule mining and it's implementation.
- Association mining can be used to predict probable `if this; then rule` behaviour.
- Usually applied in large markets to place items that one might buy if you purchase an associated item. This maximises profit and enhances the experience of the customer when they can find everything at ease.

Dataset

Member_number	Date	itemDescription
unique identity of a customer	Date of purchase	Name of item

Pre-processing

- Grouped data by Member_number
- Cleaned transaction list for grouped data.

Reasoning

- Associations can be discerned based transactions across dates by each unique customer.
- Apriori algorithm can be applied on each set of transactions to retrieve probable `if this; then rules`, their confidence, support and lift.

Rule of Mining

Parameters considered: - Support - Confidence - Lift

Choice of Algorithm

- Apriori Algorithm is chosen due to our transaction data. Step followed :down:
  - Set minimum support & confidence
  - Extract all subsets having a higher value of support than minimum threshold
  - Select all rules from subsets with confidence value higher than the minimum threshold
  - Order the rules by descending order of lift.

Time complexity

- Apriori Algorithm can be slow. The algorithm scans the data too many times. Usage of this algorithm assumes that the data is permanently in the memory.
- The limitation are that both the time and space complexity of this algorithm are very high:O(2^|x|), thus exponential, where |Dx| is the horizontal width (the total number of items) present in the database.

Recommendation

Intepretation of the results of the apriori rules could be used to inform placement of products in store, while I'd suggest to avoid blindly following high confidence rules due to possible correctness & completeness issues of the input data, further iterations that encompasses the full catalogue of the store across a full year of transactions, across seasons and changes in spending habits of consumer could yield further benefits.

Source

Jupyter Notebook

Python Export

```
# To add a new cell, type '# %%'
# To add a new markdown cell, type '# %% [markdown]'
# %% [markdown]
# # Associtaion rule mining
# %% [markdown]
# ## Dataset
# ##### https://www.kaggle.com/heeraldedhia/groceries-dataset

# %%
import pandas as pd
data = pd.read_csv('groceries.csv')
data.itemDescription.describe()

# %% [markdown]
# ## Preprocessing

# %%
total_tuples = len(data)
unique_tuples = data['Member_number'].unique()
grouped_tuples = data.groupby(data['Member_number'])
print("Total: " + str(total_tuples),
        "\nUnique customers: " + str(len(unique_tuples)),
        "\nCustomers after grouping: " + str(len(grouped_tuples)))
dataset=grouped_tuples
grouped_tuples.head(5)

# %% [markdown]
# ## Transactions

# %%
transactions = [list(dataset.get_group(unique_tuple)['itemDescription'])
                for unique_tuple in unique_tuples]
# set(map(tuple, transactions_duplicates))
clean_transactions = []
for transaction in transactions:
    clean_transactions.append(set(transaction))
clean_transactions[1]


# %%
## Association Rule Mining using Apriori Algorithm

from apyori import apriori
association_rules = list(apriori(clean_transactions,min_support=0.07,
                        min_confidence=0.4, min_lift=1, min_length=2))
for rule in association_rules:
    if len(list(rule[0])) > 1:
        print(f'Rule: {list(rule[0])[0]} => {list(rule[0])[1]}')
```

```
        print(f'## Support: {str(rule[1])}')
        print(f'## Confidence: {str(rule[2][0][2])}')
        print(f'## Lift: {str(rule[2][0][3])}\n')
#         print('')


# %%
```

References

Kaggle