Association Rules Mining

By Wilson Peguero Rosario

Data mining is used to locate beneficial information by finding anomalies, patterns, or correlations within data. Association rules mining uses "if-then" statements to show the most important relationships between data. Some real-world examples of its use include medical diagnosis, purchasing patterns, consumer website usages, or content recommendation engines.

To perform association rules analysis/mining, complete the following:

1. Access the "UCI Machine Learning Repository," located in the topic Resources. Note: There are about 120 data sets that are suitable for use in a clustering task. For this part of the exercise, you must choose two of these datasets, provided they include at least 10 attributes and 10,000 instances.
2. Ensure that the data sets are suitable for clustering using this method.
3. You may search for data in other repositories, such as Data.gov or Kaggle.

For your selected dataset, build a clustering model as follows:

1. Explain the dataset and the type of information you wish to extract. Recall that the dataset must consist of transactions of the form If {x1, x2, ..., xn} then {y1, y2, ..., yk}.
2. Explain the Apriori algorithm and how you will be using it in your analysis (list the steps, the intuition behind the mathematical representation, and address its assumptions).
3. Identify the appropriate software packages.
4. Preprocess the data, describe their characteristics, and visualize key characteristics like popular items and choices.
5. Build the clustering model by implementing the Apriori algorithm.
6. Run the model (make predictions).
7. Display clustering results (quantitative and visual).
8. Explain the meaning of each step in the context of the dataset.
9. Interpret results and adjust your clustering.
10. Validate the model, addressing support, confidence, lift, and conviction. Then, explain the results.

Prepare a comprehensive technical report as a markdown document or Jupyter notebook, including all code, code comments, all outputs, plots, and analysis. Make sure the project documentation contains a) problem statement, b) algorithm of the solution, c) analysis of the findings, and d) references.

Problem Statement

Data mining is a technique used for examining large data structures to find patterns, trends, and hidden insight that may not be found using simple statistical or query based techniques. This technique uses a multitude of algorithms to classify, segment, and correlate data to view any possible rules associated with the said data set. Using Data mining to discover any rules or conditions that determine whether an individual makes more than 50K is viable for informing the public on whether factors (such as sex, race, native-country, workclass, education, occupation, etc.).

The main goal of using this data set is to see if there are any patterns in society that impact how much one earns.

Solution

Before any Association Data Mining Learning can be done, one must first preprocess the data into a more readable format for computers.

In [ ]:

```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules

cols = [
    'age',
    'workclass',
    'fnlwght',
    'education',
    'education-num',
    'marital-status',
    'occupation',
    'relationship',
    'race',
    'sex',
    'capital-gain',
    'capital-loss',
    'hours-per-week',
    'native-country',
    'earning'
]

df1 = pd.read_csv('adult/adult.data', names=cols)
df2 = pd.read_csv('adult/adult.test', names=cols)

df__adult = pd.concat([df1, df2])
df__adult.head(10)
```

Out[ ]:

	age	workclass	fnlwght	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capi
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	
5	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	
6	49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black	Female	0	

	age	workclass	fnlwght	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capi
7	52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	
8	31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	14084	
9	42	Private	159449	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	5178	

Now that one has taken a look into the data set, one should be able to now create an itemset to develop the Association Rule Learning algorithm.

Before that, one must change the continuous numerical value to categorical.

In [ ]:
```
df__adult['age range'] = pd.cut(df__adult['age'], range(df__adult['age'].min(), df__adult['age'].m
df__adult['capital-gain range'] = pd.cut(df__adult['capital-gain'], range(df__adult['capital-gain'
df__adult['earning'] = df__adult['earning'].apply(lambda x: x.replace(".", ""))
# This is commented out as the capital-loss is all blank
# df__adult['capital-loss range'] = pd.cut(df__adult['capital-loss'], range(df__adult['capital-los
```

Now that the desired columns are in place, one is able to further develop the data set into an itemset by going from a pandas dataframe to a list.

In [ ]:
```
data = []
for index, row in df__adult.iterrows():
    data.append([str(row['age range']),str(row['education']),str(row['earning'])])
```

In [ ]:
```
a = TransactionEncoder()
a_data = a.fit(data).transform(data)
df = pd.DataFrame(a_data, columns=a.columns_)
df = df.replace(False, 0)
df = df.replace(True, 1)
df.head()
```

Out[ ]:

	10th	11th	12th	1st-4th	5th-6th	7th-8th	9th	<=50K	>50K	Assoc-acdm	Assoc-voc	Bachelors	Doctorate	HS-grad	Masters	Pr
0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	
1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	
2	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	
3	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	

Now let's apply the apriori algorithm.

In [ ]:
```
df = apriori(df, min_support = 0.2, use_colnames=True, verbose=1)
df.head()
```

```
Processing 12 combinations | Sampling itemset size 2
```

Out[ ]:

	support	itemsets
0	0.760718	(<=50K)
1	0.239282	(>50K)
2	0.323164	(HS-grad)
3	0.222718	(Some-college)
4	0.271918	(HS-grad, <=50K)

Now that one has the supports as well as the

In [ ]:
```
df_ar = association_rules(df, metric="confidence", min_threshold=0.5)
df_ar
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(HS-grad)	(<=50K)	0.323164	0.760718	0.271918	0.841422	1.106089	0.02608	1.508919

From the observation above, it seems that there is not any significant correlation in finance. The most one can obtain is that the probability of someone earning less than or equal to 50K given that they are HS-grad is 27.2%.