Mining Rare Association Rules

Or Gottman (ID. 209339480) & Roee Oscar (ID. 318603339)

Submitted as final project report for the Tabular Data Science course, Bar Ilan University, Semester A, 2022

Abstract

In the fields of Machine Learning and Data mining, association rules represent relations between variables in large datasets. In any transaction with a variety of items, association rules determine how items are connected. They are mainly measured by two measures, support and confidence, and are generated by variety of algorithms, such as Apriori, which get thresholds for these measures as parameters. The main problem is setting the values for these thresholds, in order to get informative and meaningful rules. To overcome this problem and raise the chances of the generated association rules to be informative, we propose an association rules mining technique where we first cluster similar samples and then mine each cluster for its association rules independently. We try to show that doing so results in mining of more insightful association rules.

1 Problem Description

The Data Science pipeline element which we are trying to improve is **Association Rules Mining**. Association rules are a data mining technique that is widely used for learning and analyzing correlations among items in databases. They have two associated measures: **Support** and **Confidence**.[1]

Given a transaction set D and an association rule $r = A \Longrightarrow B$:

- **Support** is the percentage of transactions in D that contain both A and B, and is defined by:

$$support(r) := P(A \cup B) = \frac{\sum_{T \in D}[I_{A \cup B \in T}]}{|D|}$$

- **Confidence** is the percentage of transactions in D containing A that also contain B, and is defined by:

$$confidence(r) := P(B|A) = \frac{P(B \wedge A)}{P(A)} = \frac{support(A \cup B)}{support(A)}$$

The most frequent problem with association rules is the adjustment of threshold values for these measures. Lower values will tend to produce more rules, which will make it hard to determine which ones are beneficial, while higher values can prevent important and meaningful rules from being considered. High support and confidence values usually lead to producing rules that are not necessarily relevant - rules that are most likely formed by dominating item sets and tend to be more obvious. Our goal is to create a tool which will improve and automate the process of mining the most promising and insightful association rules from a given dataset.

2 Solution Overview

As explained, association rules mining algorithms usually generate a large number of rules that are not interesting or are already known. The task of finding new knowledge among the generated rules makes the association rules exploration a new challenge. One possible solution is raising the support and the confidence values, resulting in a generation of fewer rules. The problem with this approach is that as the support and confidence values come closer to 100%, the algorithm might not yield enough rules at all. Even worse - the rules mined might be trivial and misleading, and probably won't contain any interesting information that we couldn't know earlier. That is because rules with high support mean that their associations are shared with high percentage of the samples in the dataset, which means that they are probably trivial.

We will try to mine association rules in a little more sophisticated way. Mining association rules from the dataset might not take into account interesting rules that occur in different portions of it. Our solution is to first pre-process the dataset by clustering, and then mine each cluster for its own association rules, independently. This will enable each cluster to express their own associations, without interference from other clusters, that usually consist of different patterns.

We take care of two kinds of datasets - labeled and unlabeled. Clustering labeled datasets is done by splitting them into smaller datasets, according to the labels' values. Unlabeled datasets require a little bit of preparing. For them we use a clustering algorithm - k-Means.[2] To do so, the dataset has to contain only numerical values - so we first prepare the dataset by one-hot encoding the textual categorical columns, and by label encoding the textual ordinal columns.[4] At this phase, we have a labeled dataset that can be split into smaller datasets. Each cluster is then converted into a transactional form, and mined for its association rules using the Apriori algorithm.[3] We keep and return a list of all the rules mined that way.

3 Experimental Evaluation

- **Lift** measures how many times more often A and B occur together than expected if they were statistically independent, and is defined by:

$$lift(A \Longrightarrow B) = lift(B \Longrightarrow A) = \frac{conf(A \Longrightarrow B)}{supp(B)} = \frac{P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(A) \cdot P(B)}$$

  A lift value of 1 indicates independence between A and B.[1]

- **Conviction** compares the probability that A appears without B if they were dependent on the actual frequency of the appearance of X without Y. In contrast to lift, it is a directed measure - it also uses the information of the absence of the consequent ($\bar{B}$). Conviction is defined by:

$$conviction(A \Longrightarrow B) = \frac{1 - supp(B)}{1 - conf(A \Longrightarrow B)} = \frac{P(A) \cdot P(\bar{B})}{P(A \cap \bar{B})}$$

A Conviction value of 1 indicates independence between A and B. Rules that always hold get conviction value of inf .[1]

We ran experiments on labeled datasets and unlabeled datasets.

3.1 Labeled Datasets

- **House Prices** The House Prices Dataset contains information about 1460 houses in Ames, Iowa, with 80 different features, and each house has its sale price (its label) in the 'SalePrice' column.[5] This column is numerical, and in order to cluster the dataset we first bin this column into 5 bins.

Running the Apriori algorithm on a dataset with 80 columns takes unreasonably long time, so as seen in class, we chose to focus only on the following 14 prominent columns: 'OverallQual', 'YearBuilt', 'YearRemodAdd', 'OverallQual', 'OverallCond', 'BldgType', 'LotArea', 'GrLivArea', 'FullBath', 'BedroomAbvGr', 'LotFrontage', 'TotalBsmtSF', 'SalePrice'.

We then separate the dataset into clusters and mine each cluster for its association rules using Apriori algorithm with support threshold of 0.5 and confidence threshold of 0.8. This way we got a total of 126 rules.

Compared to simply running the Apriori algorithm with the same thresholds on the unpartitioned dataset, we only get 1 association rule, which is BedroomAbvGr of 3 BldgType of 1Fam, which is kind of trivial.

To get the around the same number of rules from the unpartitioned dataset as the number of rules we mined from the clusters, the support / confidence thresholds (or both) need to be lowered. We had to lower the support threshold to 0.15, and the confidence threshold to 0.25. That way we mined 120 rules.

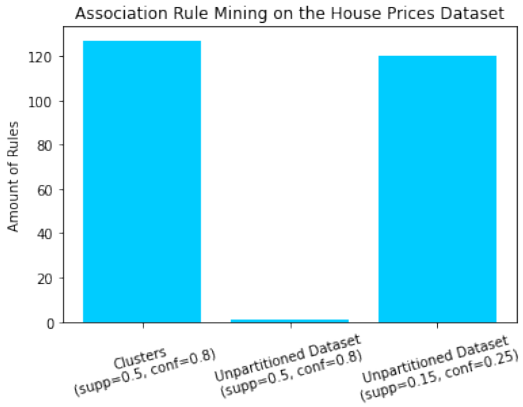

Figure 1: House Prices Dataset - Amount of Mined Rules per Approach

We then check the mined rules for their interestingness level. We count how many rules have lift and conviction values that aren't close to 1. Our assumption was that the rules we mined from clusters are more informative - so we expected them to have lift and conviction values under or above 1 - which mean negative correlation and strong dependancy, respectively.

We define an rule an interesting rule, when its interestingness measure is

not in the range of $[1-\epsilon, 1+\epsilon]$. We set ϵ to 0.5. When measuring by lift, we get that 45 out of the 126 rules mined from clusters, and 95 out of the 120 rules mined from the unpartitioned dataset are interesting. When measuring by conviction, we get the opposite - 112 rules from the clusters, and 102 from the unpartitioned dataset.

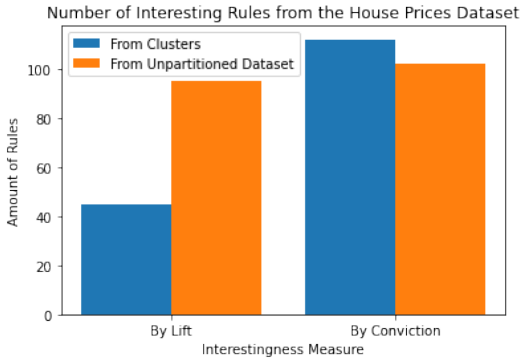

Figure 2: House Prices Dataset - Interesting Rules Summary

- **Adult Income** This dataset contains information about the annual income of adults. It has 16 columns and is divided into two classes.[6] We ran the same procedure on this dataset. Separating the data set into clusters and mining each one for its association rules with support threshold of 0.5 and confidence threshold of 0.8 resulted in 68 rules.

  Simply running the Apriori algorithm with the same thresholds on the unpartitioned dataset, resulted in 3 association rules only.

  We had to lower the support threshold to 0.29, and the confidence threshold to 0.4. That way we mined 65 rules.

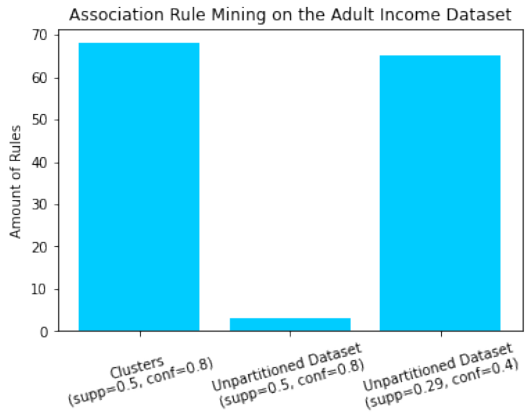

Figure 3: Adult Income Dataset - Amount of Mined Rules per Approach

When we tested the lift and conviction values, the unpartitioned dataset has more interesting rules when measuring by lift (28 vs 6), but less when measuring by conviction (46 vs 40):

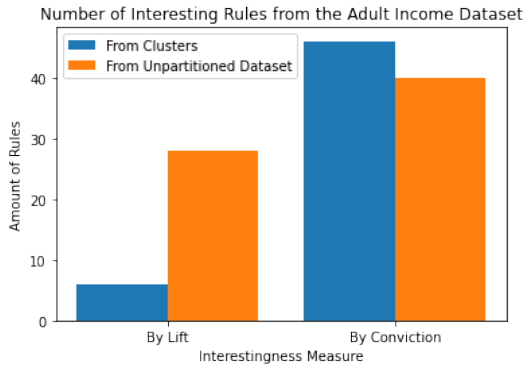

Figure 4: Adult Income Dataset - Interesting Rules Summary

3.2 Unlabeled Datasets

- **Customers Dataset** This dataset contains various information taken by a Telecommunication company about their customers such as age, region, martial status, etc.[7] Some of the customers are labeled, but not enough. We remove the labels and treat this dataset as an unlabeled one. We run k-Means to cluster the dataset and get the following results:

Figure 5: Customers Dataset - Amount of Mined Rules per Approach

Similar thing happens here. when using thresholds of 0.5 and 0.8 for support and confidence, respectively, we get a total of 76 rules from the clusters, and 0 rules from the unpartitioned dataset.

to get around the same number of rules from the unpartitioned dataset, we lowered the support threshold to 0.16 and the confidence threshold to 0.5. This way we got 73 rules.

When we tested the lift and conviction values, the unpartitioned dataset has way more interesting rules when measuring by lift (39 vs 3), but less when measuring by conviction (65 vs 76):

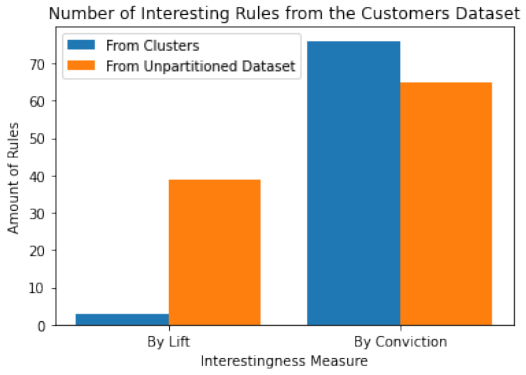

Figure 6: Customers Dataset - Interesting Rules Summary

4 Related Work

We came across an paper about clustering transactional data.[8] We then thought about using clustering methods in the process of association rules mining, with hope to find hidden rules that are hard to mine. We saw in another paper [9] that research has been carried out on the use of clustering algorithms to prepare the database, before extracting the association rules. So, We chose to try clustering datasets, convert each cluster into transactional form, then mine each cluster for its own association rules independently.

5 Conclusion

We have presented a more clever way of mining association rules from a data set. Our technique tries to find hidden association rules that are hard to detect when analyzing a given data set as a whole.

We suspect that measuring the lift and conviction values, is not the best way to measure the interestingness of rules for our purpose. That may be because the lift and conviction calculations for each rule depend on its support and confidence values, and doesn't do justice with the rare rules we mined from each cluster independently.

For future work we will try to check the interestingness level of our rules in another way - we will classify the test set and label it using a classifier, such as KNN. Then, we will randomly remove one cell from every sample, and see how well the rules mined earlier from the specific cluster (on the train data) restore the missing value, in comparison to the rules mined from the whole dataset.

References

[1] Michael Hahsler. *A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules* Link: `https://mhahsler.github.io/arules/docs/measures`

[2] MacQueen J. (1967). *Some methods for classification and analysis of multivariate observation*. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pp. 281-297.

[3] R. Agrawal and R. Srikant, *Fast Algorithms for Mining Association Rules in Large Databases*, Proc 20th Int'l Conf. Very large Data Bases, pp. 478-499, Sept 1994.

[4] Alakh Sethi, *One-Hot Encoding vs. Label Encoding using Scikit-Learn.*

[5] House Prices Dataset, Kaggle. Link: `https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data`

[6] Adult Income Dataset, Kaggle. Link: `https://www.kaggle.com/wenruliu/adult-income-dataset`

[7] Pratham Tripathi, Customer Classification Dataset, Kaggle. Link: `https://www.kaggle.com/datasets/prathamtripathi/customersegmentation`

[8] Mahmoud A. Mahdi, Samir E. Abdelrahman, Reem Bahgat and Ismail A. Ismail. *F-Tree: an algorithm for Clustering Transactional Data Using Frequency Tree*

[9] Renan de Padua, Exuperio Ledo Silva Junior, Laes Pessine do Carmo, Veronica Oliveira de Carvalho and Solange Oliveira Rezende, *Preprocessing data sets for association rules using community detection and clustering: a comparative study*, XIII Encontro Nacional de Intelig^encia Artificial e Computational, pp. 553-564, 2016.