Data Mining and Warehousing [CS-2376]
Prof. Anirban Mondal
Spring 2021, Ashoka University

Association Rule Mining Assignment

Rishabh Goswami, Mohammed Zahir Ali, Shreyash Naman

Association Rule Mining is a method for finding meaningful relationships between itemsets in large datasets. For our implementation of Association Rule Mining, we have chosen the Apriori Algorithm. The Apriori property states that any subset of a frequently occurring itemset must also be frequent, i.e. if an itemset {I1, I2} is frequent, its constituent items {I1} and {I2} (1-itemsets) must be frequent as well.

Following are the steps in our algorithm:
Step 1: Import the data from a csv file and store it into a dataframe in the form of a list.

Step 2: Loop through all the items in the list, while counting the frequency of occurrence for each item. A dictionary of all unique items is created, which forms our first candidate set (C1). This is essentially a dictionary of all 1-itemsets.

Step 3: Check the frequency of each item in C1, and select only those items which occur frequently enough to meet our minimum support criteria. This step is called pruning because items that are not frequent are eliminated. These selected items are stored in a dictionary L1. L1 contains all frequent 1-itemsets.

Step 4: We then generate Candidate Set C2, which is a list of all 2-itemsets. Following apriori property, we know that 1-itemsets that were pruned in *Step 3* would also not meet the support criteria when included in 2-itemsets. So we only use 1-itemsets in L1 to create C2. L1 is added to itself to create all possible combinations of 2-itemsets.
For example: L1: {(1), (2), (3)} ; C2: {(1,2), (2,3), (1,3)}

Step 5: We then evaluate the itemsets in C2 against the dataframe, to see which itemsets meet the minimum support criteria. Itemsets that are not frequent enough are rejected, and only

items that meet the support frequency are selected and put into another dictionary L2. Thus, C2 is pruned to get L2, and the latter is a dictionary that contains all frequent 2-itemsets.

Step 6: In a similar process, we generate C3 (Candidate Set of 3-itemsets) from L2, by adding items in L2 with themselves.
Each C(k) is generated by adding L(k-1).

Step 7: C3 is pruned to get L3, which is a dictionary containing all frequent 3-itemsets. Each C(k) is thus pruned to get L(k).

Step 8: This continues iteratively until we get a null C(k), where the algorithm terminates.

Step 9: For each frequent itemset (I) obtained, we generate all non-empty subsets (S) of the itemset. We then generate the association rules S -> (I - S), with Confidence = Support(I)/Support(S). To get only the relevant association rules, this is then pruned again, wherein all rules that do not meet the minimum specified confidence criteria are eliminated.

Notes

We tested our algorithm with the accompanying dataset '*store_data.csv*' which has 7501 transactions. The algorithm might need adjustment depending on the format and type of dataset provided.

Acknowledgements

We gratefully acknowledge the authors of the following resources that we used as reference:

"Apriori Algorithm In Data Mining With Examples" , January 22, 2020. *Datamining365*. https://www.datamining365.com/2020/01/apriori-algorithm.html

Malogie Mohammad, Arkok Bassam. " AN IMPROVED APRIORI ALGORITHM FOR ASSOCIATION RULES." February, 2014. *International Journal on Natural Language Computing.*
https://arxiv.org/ftp/arxiv/papers/1403/1403.3948.pdf

Vaish, Piush. “Apriori Algorithm” January, 2019.
https://adataanalyst.com/machine-learning/apriori-algorithm-python-3-0/

“Apriori-Implementation-in-Python.” February, 2018. *Github*.
http://ethen8181.github.io/machine-learning/association_rule/apriori.html

Manoj, Harsha. “Underrated Machine Learning Algorithms — APRIORI.”
January, 2020. *Towards Data Science*.
https://towardsdatascience.com/underrated-machine-learning-algorithms-apriori-1b1d7a8b7c

“Apriori-Implementation-in-Python.” April, 2018. *Github*.
https://github.com/Titan-BT-7274/Apriori-Implementation-in-Python

“Association Rule Mining Python.” September, 2016. *Github*.
https://github.com/nalinaksh/Association-Rule-Mining-Python