A project report on

Association Rule Mining

Submitted by
ASHOK SUTHAR (17MCMT17)
DHILBER M (17MCMI15)

Students of
M.TECH IN COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF COMPUTER & INFORMATION SCIENCE
UNIVERSITY OF HYDERABAD
2017-2019

Submitted to
Dr. V. RAVI
Professor
INSTITUTE FOR DEVELOPMENT AND RESEARCH IN BANKING TECHNOLOGY (IDRBT)

Association Rule Mining On Online Retail Data

Abstract

Association rules are if then rules generated by analyzing the existing data. Here we are working on Online Raatail data of a super market. We use 2 different algorithms namel (i)Apriori algorith and (ii)FP Growth algorithm which are explained in detail to generate association rules. Here we varies various parameters of algorithm like support, confidence, lift, leverage etc and the corresponding results (rules) are tabulated.

Introduction

Association rule Mining is a rule based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. Based on the concept of strong rules, Rakesh Agrawal, Tomasz and Arun Swami introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale(POS) systems in supermarkets.

For example, the rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, they are likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysisassociation rules are employed today in many application areas including Web usage mining, intrusion detection, continuous production, and bioinformatics. In contrast with sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Mining Association Rules is one of the most important application fields of Data Mining. Provided a set of customer transactions on items, the main intention is to determine correlations among the sales of items. Mining association rules, also known as market basket analysis, is one of the application fields of Data Mining. Think a market with a gathering of large amount of customer transactions. An association rule is X⇒Y, where X is referred as the antecedent and Y is referred as the consequent.

X and Y are sets of items and the rule represents that customers who purchase X are probable to purchase Y with probability %c where c is known as the confidence. Such a rule may be: “Eighty percent of people who purchase cigarettes also purchase matches". Such rules assists to respond questions of the variety “What is Coca Cola sold

with?" or if the users are intended in checking the dependency among two items A and B it is required to determine rules that have A in the consequent and B in the antecedent.

Problem statement

The given data set consist of various attributes all regarding online retail of a super market. Objective is to generate association rules, that helps organisation predict which all products are bought together, based on the given data. Application of apriori algorithm and fp growth algorithm is expected.

Advantages of Rule mining and Market Basket Analysis

Market Basket Analysis empowers marketing and sales organizations to make better, informed decisions about how and where to deploy their efforts and resources. Moreso, strategic action plans can be developed and deployed that align resources around these insights to increase sales and profitability.

The primary objective of Market Basket Analysis is to improve the effectiveness of marketing and sales tactics using customer data collected during the sales transaction. It can also be used to optimize and facilitate business operations particularly with regards to inventory control and channel optimization.

Leading organizations are applying Market Basket Analysis modeling in the financial services, insurance, retail, health care and information communication and technology industries to deploy and improve:

- Cross-Sell / Upsell - Existing customers present an opportunity to grow revenues through cross-sell and upsell strategies.
- Product Promotions and Placement - Determine which products are most likely to be purchased together and develop highly effective product promotion and placement strategies.
- Next-Best Offer - Identify the products or services your customers are most likely to be interested in for their next purchase by applying Market Basket Analysis in an operational setting.

Tools Used

- Weka
- Python

Techniques Used

- FP Growth Algorithm
- Apriori Algorithm

Variable Description

Var. #	Variable Name	Variable Type
1.	Invoice No	Numerical
2.	Stock Code	Numerical
3.	Description	Nominal
4.	Quantity	Numerical
5.	Invoice Date	Nominal
6.	Unit Price	Numerical
7.	Customer ID	Numerical
8.	Country	Nominal

Invoice No : It is the unique number for each individual transaction. (Numerical)

Stock Code : Unique code for each product available.(Numerical)

Description : Description of the product.(Nominal)

Quantity : no. Of particular product ordered (Numerical)

Invoice Date : Date of purchase(Nominal)

Unit price : (Numerical)

Customer ID : (Numerical)

Country : (Nominal)

Algorithm Description

Apriori Algorithm

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

Defenition :
Following the original definition by Agrawal the problem of association rule mining is defined as: Let I = {i1, i2, ..., in} be a set of n binary attributes called items. Let D = {t1, t2, ..., tn} be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form X→Y where X, Y ⊆ I and ∩ = ∅. The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively.

General Process :

Association rule generation is usually split up into two separate steps: 1. First, minimum support is applied to find all frequent itemsets in a database. 2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules. While the second step is straight forward, the first step needs more attention. Finding all frequent itemsets in a database is difficult since it involves searching all possible itemsets (item combinations). The set of possible itemsets is the power set over I and has size 2n − 1 (excluding the empty set which is not a valid itemset). Although the size of the powerset grows exponentially in the number of items n in I, efficient search is possible using the downward-closure property of support (also called anti-monotonicity) which guarantees that for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. Exploiting this property, efficient algorithms (e.g., Apriori and Eclat) can find all frequent itemsets.

Support :

The support supp(X) of an itemset X is defined as the proportion of transactions in the data set which contain the itemset.

supp(X)= no. of transactions which contain the itemset X / total no. of transactions

Confidence :

The confidence of a rule indicates the probability of both the antecedent and the consequent appearing in the same transaction. Confidence is the conditional probability of the consequent given the antecedent. For example, cereal might appear in 50 transactions; 40 of the 50 might also include milk. The rule confidence would be:

```
cereal implies milk with 80% confidence
```

Confidence is the ratio of the rule support to the number of transactions that include the antecedent.

Confidence(X) = no.of transactions containing both A and B / no.of transactions containing A

Lift:

A third measure is needed to evaluate the quality of the rule. Lift indicates the strength of a rule over the random co-occurrence of the antecedent and the consequent, given their individual support. It provides information about the improvement, the increase in probability of the consequent given the antecedent. Lift is defined as follows.

Lift(X) = (Rule Support) /(Support(Antecedent) * Support(Consequent))

FP Growth Algorithm

The FP-Growth Algorithm, proposed by Han in, is an efficient and scalable method for mining the complete set of frequent patterns by pattern fragment growth, using an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree).

In his study, Han proved that his method outperforms other popular methods for mining frequent patterns, e.g. the Apriori Algorithm and the TreeProjection. In some later works it was proved that FP-Growth has better performance than other methods, including Eclat and Relim. The popularity and efficiency of FP-Growth Algorithm contributes with many studies that propose variations to improve his performance

The FP-Growth Algorithm is an alternative way to find frequent itemsets without using candidate generations, thus improving performance. For so much it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the itemset association information.

In simple words, this algorithm works as follows: first it compresses the input database creating an FP-tree instance to represent frequent items. After this first step it divides the compressed database into a set of conditional databases, each one associated with one frequent pattern. Finally, each such database is mined separately. Using this strategy, the FP-Growth reduces the search costs looking for short patterns recursively and then concatenating them in the long frequent patterns, offering good selectivity.

In large databases, it’s not possible to hold the FP-tree in the main memory. A strategy to cope with this problem is to firstly partition the database into a set of smaller databases (called projected databases), and then construct an FP-tree from each of these smaller databases.

Results

(1)

* FP Growth Algorithm
* Minimum Support : .02
* Minimum Confidence : 0.5

SL. no	Association Rules	Confidence	Lift	Leverage	Conviction
1.	[ALARM.CLOCK.BAKELIKE.ORANGE_binarized=1]: 31 ==> [ALARM.CLOCK.BAKELIKE.GREEN_binarized=1]: 26	0.84	14.24	0.02	4.86
2.	[ALARM.CLOCK.BAKELIKE.RED_binarized=1]: 52 ==> [ALARM.CLOCK.BAKELIKE.GREEN_binarized=1]: 39	0.75	12.74	0.04	3.5
3.	[HAND.WARMER.OWL.DESIGN_binarized=1, HAND.WARMER.RED.RETROSPOT_binarized=1]: 30 ==> [HAND.WARMER.SCOTTY.DOG.DESIGN_binarized=1]: 22	0.69	10.97	0.02	3.11
4.	[FELTCRAFT.CUSHION.BUTTERFLY_binarized=1]: 39 ==> [FELTCRAFT.CUSHION.RABBIT_binarized=1]: 27	0.69	16.52	0.03	2.87
5.	[ALARM.CLOCK.BAKELIKE.IVORY_binarized=1]: 32 ==> [ALARM.CLOCK.BAKELIKE.RED_binarized=1]: 22	0.69	13.25	0.02	2.76
6.	[HAND.WARMER.SCOTTY.DOG.DESIGN_binarized=1, HAND.WARMER.RED.RETROSPOT_binarized=1]: 32 ==> [HAND.WARMER.OWL.DESIGN_binarized=1]: 22	0.69	10.93	0.02	2.73
7.	[ALARM.CLOCK.BAKELIKE.PINK_binarized=1]: 35 ==> [ALARM.CLOCK.BAKELIKE.GREEN_binarized=1]: 24	0.69	11.65	0.02	2.74
8.	[ALARM.CLOCK.BAKELIKE.GREEN_binarized=1]: 59 ==> [ALARM.CLOCK.BAKELIKE.RED_binarized=1]: 39	0.66	12.74	0.04	2.66
9.	[ALARM.CLOCK.BAKELIKE.IVORY_binarized=1]: 32 ==> [ALARM.CLOCK.BAKELIKE.GREEN_binarized=1]: 21	0.66	11.15	0.02	2.51
10.	[FELTCRAFT.CUSHION.RABBIT_binarized=1]: 42 ==> [FELTCRAFT.CUSHION.OWL_binarized=1]: 27	0.64	14.98	0.03	2.51

(2)

* FP growth Algorithm
* Minimum suopport : 0.01
* Minimum Confidence : 0.8

SL.no	Association Rules	Confidence	Lift	Leverage	Conviction
1.	[HERB.MARKER.THYME_binarized=1]: 11 ==> [HERB.MARKER.PARSLEY_binarized=1]: 11	1	91.09	0.01	10.88
2.	[HERB.MARKER.PARSLEY_binarized=1]: 11 ==> [HERB.MARKER.THYME_binarized=1]: 11	1	91.09	0.01	10.88
3.	[HERB.MARKER.THYME_binarized=1]: 11 ==> [HERB.MARKER.BASIL_binarized=1]: 11	1	91.09	0.01	10.88
4.	[HERB.MARKER.BASIL_binarized=1]: 11 ==> [HERB.MARKER.THYME_binarized=1]: 11	1	91.09	0.01	10.88
5.	[HERB.MARKER.PARSLEY_binarized=1]: 11 ==> [HERB.MARKER.BASIL_binarized=1]: 11	1	91.09	0.01	10.88
6.	[HERB.MARKER.BASIL_binarized=1]: 11 ==> [HERB.MARKER.PARSLEY_binarized=1]: 11	1	91.09	0.01	10.88
7.	[ALARM.CLOCK.BAKELIKE.PINK_binarized=1, ALARM.CLOCK.BAKELIKE.ORANGE_binarized=1]: 15 ==> [ALARM.CLOCK.BAKELIKE.GREEN_binarized=1]: 15	1	16.98	0.01	10.88
8.	[HERB.MARKER.THYME_binarized=1]: 11 ==> [HERB.MARKER.PARSLEY_binarized=1, HERB.MARKER.BASIL_binarized=1]: 11	1	91.09	0.01	10.88
9.	[HERB.MARKER.PARSLEY_binarized=1]: 11 ==> [HERB.MARKER.THYME_binarized=1, HERB.MARKER.BASIL_binarized=1]: 11	1	91.09	0.01	10.88
10.	[HERB.MARKER.THYME_binarized=1, HERB.MARKER.PARSLEY_binarized=1]: 11 ==> [HERB.MARKER.BASIL_binarized=1]: 11	1	91.09	0.01	10.88

(3)

* FP Growth Algorithm
* Min Support : 0.015
* Min confidence : 0.9

SL.no	Association Rules	Confidence	Lift	Leverage	Conviction
1.	[ALARM.CLOCK.BAKELIKE.RED_binarized=1, ALARM.CLOCK.BAKELIKE.ORANGE_binarized=1]: 20 ==> [ALARM.CLOCK.BAKELIKE.GREEN_binarized=1]: 19	0.95	16.13	0.02	9.41
2.	[HAND.WARMER.RED.POLKA.DOT_binarized=1]: 18 ==> [HAND.WARMER.UNION.JACK_binarized=1]: 17	0.94	11.98	0.02	8.29

(5)

* Apriori Algorithm
* Min Support : 0.1
* Min Confidence : 0.91

SL.no	Association Rules	Confidence	Lift	Leverage	Conviction
1.	Quantity=1 4995 ==> Country=United_Kingdom 4921	0.99	1.01	0.01	4.84
2.	Quantity=3 1278 ==> Country=United_Kingdom 1235	0.97	1.04	0	2.11
3.	Quantity=2 2926 ==> Country=United_Kingdom 2818	0.96	1.04	0.01	1.95
4.	UnitPrice=2.1 1528 ==> Country=United_Kingdom 1449	0.95	1.02	0	1.39
5.	UnitPrice=2.95 1904 ==> Country=United_Kingdom 1793	0.94	1.02	0	1.24
6.	UnitPrice=1.25 2459 ==> Country=United_Kingdom 2299	0.93	1.01	0	1.11
7.	UnitPrice=3.75 1429 ==> Country=United_Kingdom 1330	0.93	1.0	0	1.04
8.	Quantity=4 1439 ==> Country=United_Kingdom 1335	0.93	1.0	0	1
9.	UnitPrice=0.85 1629 ==> Country=United_Kingdom 1489	0.91	0.99	0	0.84
10.	Quantity=6 1941 ==> Country=United_Kingdom 1769	0.91	0.98	0	0.82

(6)

* Apriori Algorithm
* Min Support : 0.1
* Min Confidence : 0.9

SL.no	Association Rules	Confidence	Lift	Leverage	Conviction
1.	Quantity=1 4995 ==> Country=United_Kingdom 4921	.99	1.06	0.01	4.84
2.	Quantity=2 2926 ==> Country=United_Kingdom 2818	.96	1.04	0.01	1.95
3.	UnitPrice=2.1 1528 ==> Country=United_Kingdom 1449	0.95	1.02	0	1.39
4.	UnitPrice=2.95 1904 ==> Country=United_Kingdom 1793	0.94	1.02	0	1.24
5.	UnitPrice=1.25 2459 ==> Country=United_Kingdom 2299	0.93	1.01	0	1.11
6.	UnitPrice=0.85 1629 ==> Country=United_Kingdom 1489	0.91	0.99	0	0.84
7.	Quantity=6 1941 ==> Country=United_Kingdom 1769	0.91	0.98	0	0.82
8.	UnitPrice=1.65 1629 ==> Country=United_Kingdom 1478	0.91	0.98	0	0.78