Associative Rule Mining in Large Datasets using Neural Network Algorithm and Enhanced Apriori-Based Algorithm

Febin Issac, Yeshwant More

Febin Issac - Software Developer, SAP Labs India | Yeshwant More - Software Architect, SAP Labs India
#138, EPIP Zone, Whitefield, Bengaluru, Karnataka PIN-560066, India

Abstract—*Unsupervised Learning is a type of machine learning algorithm, which is used to draw conclusions out of unlabeled input data. Association Rule Mining(ARM) is one of the key unsupervised data mining method, which is used to find interesting associations in large data sets. In this paper, we would explain about data clustering by K-means followed by Association Rule Mining using Apriori algorithm in each cluster to obtain meaningful associations and then comparing it with the Self Organizing Map(SOM) clustering method. SOM makes use of neural networks for generating frequent item sets and association rules from transaction data. We would compare the accuracy and performance of the above clustering algorithms based on parameters governed by each. When this comparison is applied on various kinds of datasets, the usage of the right algorithm for each type of dataset can be determined that increases the correctness of the machine learning solution.*

Keywords

Apriori Algorithm
Association Rule Mining
K Means
Kohonen's Self Organizing Map
Machine Learning
Neural Network Algorithms
Self Organizing Map

I. INTRODUCTION

This paper explains how machine learning can help us draw conclusions out of unlabeled input data.We would demonstrate how data clustering by K-means followed by Association Rule Mining using Apriori algorithm can be compared with the Self Organizing Map method of clustering. These comparisons will be drawn based on accuracy and performance for various kinds of datasets which would help us determining the most suitable algorithm for each type of dataset.

II. DATASET PREPARATION

The dataset considered for this paper contains all transactions occurred for a non-online Retail Store. The company mainly sells unique all-occasion gifts. Many customers of this company are also wholesalers. Such a company's dataset must be preprocessed for each kind of clustering and algorithm.

The dataset is then loaded across various attributes that is a combination of both kinds of data needed for the K-means clustering and ARM using Apriori Algorithm. The same dataset can be loaded as input to SOM by choosing the attributes on which SOM clustering could be applied. The dataset is prepared asper the relevance in the Neural network scenario.

Ignoring or considering inconsistent and strange data needs to be carefully done to avoid loss of information. There can be some attributes that can be safely ignored whereas there can also be genuine data without complete transaction information of the attributes which needs to be considered by the data scientist.

Abbreviations and Acronyms

ARM – Association Rule Mining
SOM – Self Organizing Map

III. THEORY

To implement the ARM and to derive better results we have considered the following techniques in Machine Learning

A. *K-Means Clustering*

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and effortless way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the

nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding must be done between the same data set points and the nearest new centroid. A loop has been generated. Because of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words, centroids do not move any more.
Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)} - c_j\right\|^2 \qquad (1)$$

where $\left\|x_i^{(j)} - c_j\right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j, is an indicator of the distance of the n data points from their respective cluster centers.

The algorithm is composed of the following steps:

1. *Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.*
2. *Assign each object to the group that has the closest centroid.*
3. *When all objects have been assigned, recalculate the positions of the K centroids.*
4. *Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.*

B. *Apriori Algorithm*

Apriori is an unsupervised algorithm used for frequent item set mining. It generates associated rules from given data set and uses 'bottom-up' approach where frequently used subsets are extended one at a time and algorithm terminates when no further extension could be carried forward.

Apriori algorithm works on its two basic principles, first that if an itemset occurs frequently then all subset of itemset occurs frequently and other is that if an itemset occurs infrequently then all superset has infrequently occurrences.

It helps to reduce the number of possible interesting item sets and the minimum support level required by an algorithm is just the input and data set. It is one of the easiest to implement and can be parallelized easily. It makes the use of large item set properties though it also suffers from many inefficiencies which have resulted in the production of other algorithms. The algorithm needs to rescan dataset after each time of increasing the length of frequent item set resulting in reducing the speed. It is also expensive to calculate as it must examine entire database.

The algorithm is efficient for Market Basket analysis and helps to increase market sale by assisting customers during the purchase of the item. It is also applicable in the field of health care for detecting drug reactions. It analyses and produces association rule which identifies adverse drug effect through patient characteristic and medication.

Another one of the most popular application is Google Auto-complete in which the search engine suggests the other associated words according to your specified word. It is also used in Amazon recommendation system. Python implementation for Apriori is through *PyPi* and in R through *arules*.

Overview

The whole point of the algorithm (and data mining, in general) is to extract useful information from copious amounts of data. For example, the information that a customer who purchases a keyboard also tends to buy a mouse at the same time is acquired from the association rule below:
Support: The percentage of task-relevant data transactions for which the pattern is true.
Support(Keyboard=>Mouse)

$$= \frac{\text{No. of transactions containing both Keyboard and Mouse}}{\text{No. of total transactions}} \qquad (2)$$

Confidence: The measure of certainty or trustworthiness associated with each discovered pattern.
Confidence(Keyboard=>Mouse)

$$= \frac{\text{No. of transactions containing both Keyboard and Mouse}}{\text{No. of transactions containing (Keyboard)}} \qquad (3)$$

The algorithm aims to find the rules which satisfy both a minimum support threshold and a minimum confidence threshold (Strong Rules).

- Item: article in the basket.
- Itemset: a group of items purchased together in a single transaction.

How Apriori Works

1. Find all frequent item sets:
   - Get frequent items:
     - Items whose occurrence in database is greater than or equal to the minimum support threshold.
   - Get frequent item sets:
     - Generate candidates from frequent items.
     - Prune the results to find the frequent item sets.
2. Generate strong association rules from frequent item sets
   - Rules which satisfy the minimum support and minimum confidence threshold.

C. *Self Organized Maps – Kohonen's SOM*

In Kohonen's Self Organized Maps, relevant patterns are represented by 'neurons' (units). The formal counterpart of such a unit is a weight vector. When the training of the neural network is finished, after the net weights have been fitted to the given data, the resulting final weight vectors are called prototypes. Before starting the methodological considerations, some symbols must be introduced. Let n be the number of interesting product categories and m the unrestricted number of individual market baskets to be analyzed. Then an individual market basket can be defined by binary transaction vector t_j (t_{j1}, t_{j2}, ..., t_{jn}) with j € {1, ..., m}, where t_{jk} 1 if market basket j contains at least one item of product category k € {1, ..., n} and t_{jk} 0 otherwise.

In most applications the units of an SOM are organized on a two-dimensional grid where the individual positions reflect the interrelations between the respective units. In the following, each unit u_h(h € {1, ..., p}) is represented by a weight vector η_{ab} (η_{ab1}, ..., η_{abk}, ..., η_{abn}) where a and b refer to the position of the unit within a rectangular grid and $0 \leq \eta_{abk} \leq 1$ holds true.

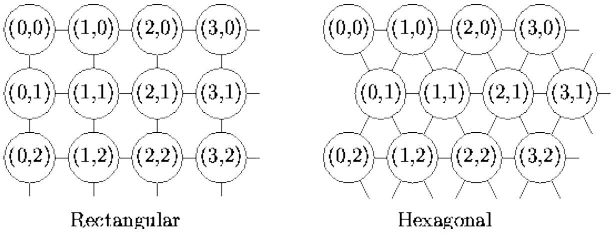


Figure 1 Types of SOMs

Applying the SOM approach to POS scanner data to identify existing purchase interdependences means, carrying out two different tasks simultaneously: finding an optimal set of prototypes representing similar market baskets and ensuring the optimal topological arrangement of these prototypes. In the resulting map similar weight vectors (representing similar market basket patterns) are located close together.

Training:

For all the iterations l € {1, ..., l_{max}} the distances between each weight vector η_{ab} and a randomly chosen input vector (market basket) t_j are computed according to

$dist\ (t_j ,\eta_{ab})=||\ t_j - \eta_{ab}\ ||^2 = \sum nk = 1(t_{jk} - \eta_{abk})^{2}$

to determine the winning unit u_{CaCb} with minimum distance. Then each weight vector must be updated as follows:

$\eta_{ab}(l + 1) = \eta_{ab}(l) + \alpha(l) \cdot nhf_{CaCb}\ (a, b) \cdot (t_j - \eta_{ab})$

where learning rate $\alpha(l) = \alpha(0)(1 - l/l_{max})$ is a decreasing function of iteration l. The extent of this adaptation can be controlled via neighborhood function

$$\mathrm{nhf}_{\mathrm{CaCb}}\,(\mathrm{a},\mathrm{b})\ \mathrm{e}\ =\ \exp\left(-\frac{(a-c_a)^2-(b-c_b)^2}{2\sigma(l)^2}\right)^{nt}$$

where $\alpha(l) = \alpha(0)(1 - l/l_{max})$ determines the scope of the neighborhood kernel. The whole procedure is repeated until the maximum number of iterations $l_{max}$is reached. The minimization of distances finally leads to the optimal prototype system {η_{CaCb} }.

To optimize the topological structure of the whole map the accumulated neighborhood distance will be minimized as well.

IV. Implementation

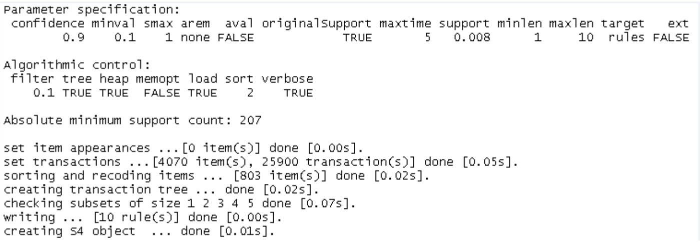
A. Implementation using Enhanced Apriori with K-means clustering:

We followed the mentioned steps to implement Rule Mining

1. Take the prepared dataset and apply K means clustering to the same and identify various clusters using the attributes Time of Purchase (To use the time of purchase data in the input dataset, we must derive the month of purchase), Country of purchase and Total value of purchase. We can enhance the clustering more efficiently, if we can get more info about the transactions or the customer who did the transaction.
2. Separate the datasets according to the cluster information
3. Apply the Apriori algorithm for association rule mining in each cluster (By doing this, we can achieve more crisp information regarding the association pertaining to each cluster. Clusters can be formed based on the country, customer or Time of purchase)
4. Collect and display association rules with high support and confidence for each cluster.

Apriori output in one of the cluster formed via K-means will be as follows

> basket_rules <- apriori (txn, parameter = list (sup = 0.008, conf = 0.9,target="rules"))

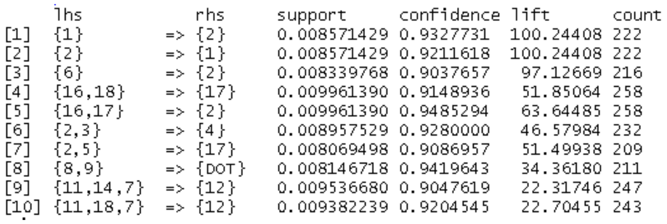


```
Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.9    0.1    1 none FALSE            TRUE       5   0.008      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 207

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[4070 item(s), 25900 transaction(s)] done [0.05s].
sorting and recoding items ... [803 item(s)] done [0.02s].
creating transaction tree ... done [0.02s].
checking subsets of size 1 2 3 4 5 done [0.07s].
writing ... [10 rule(s)] done [0.00s].
creating S4 object  ... done [0.01s].
```

Figure 2 Screen Snapafter Apriori function in R

> basket_rules
set of 10 rules

> inspect(basket_rules)



```
     lhs            rhs     support     confidence lift      count
[1]  {1}         => {2}     0.008571429 0.9327731  100.24408 222
[2]  {2}         => {1}     0.008571429 0.9211618  100.24408 222
[3]  {6}         => {2}     0.008339768 0.9037657   97.12669 216
[4]  {16,18}     => {17}    0.009961390 0.9148936   51.85064 258
[5]  {16,17}     => {2}     0.009961390 0.9485294   63.64485 258
[6]  {2,3}       => {4}     0.008957529 0.9280000   46.57984 232
[7]  {2,5}       => {17}    0.008069498 0.9086957   51.49938 209
[8]  {8,9}       => {DOT}   0.008146718 0.9419643   34.36180 211
[9]  {11,14,7}   => {12}    0.009536680 0.9047619   22.31746 247
[10] {11,18,7}   => {12}    0.009382239 0.9204545   22.70455 243
```

Figure 3Output of:>inspect (basket_rules)

However, it is clear that going through all the 5668 rules manually is not a viable option.We used the different visualization techniques implemented in arulesViz package in R.

> plot(rules, measure=c("support", "lift"), shading="confidence",)

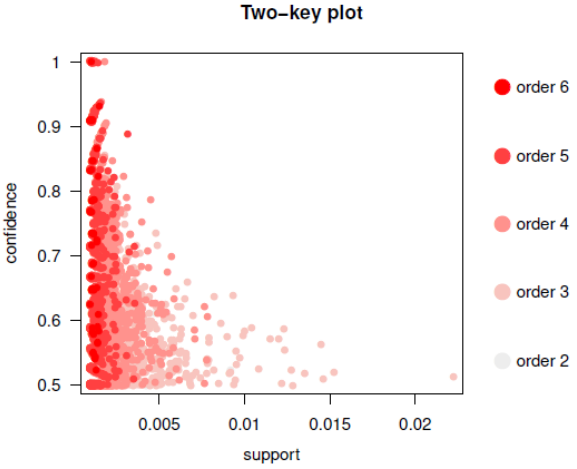


Figure 4 Output of: >plot (rules, measure=c ("support", "lift"), shading="confidence",)

> sel <- plot (rules, measure=c ("support", "lift"), shading="confidence", interactive=TRUE)

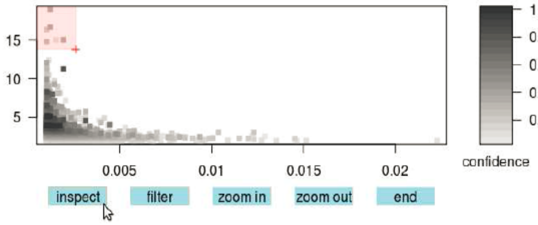


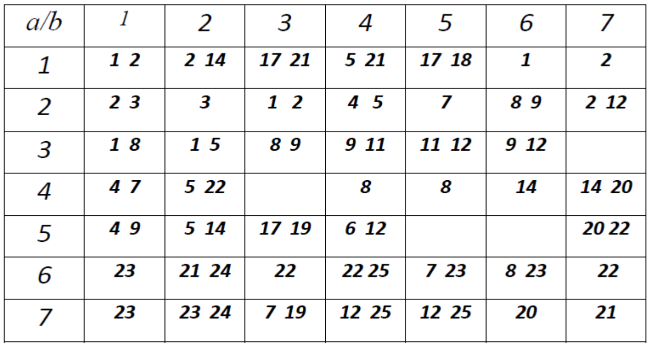
Figure 5Outputof:> sel <- plot (rules, measure=c ("support", "lift"), shading="confidence", interactive=TRUE)

The selected section in Red represents the Association Rules with high confidence.

B. *Implementation using Kohonen's SOM*

We followed the mentioned steps to implement Rule Mining

1. Read the dataset into a data frame.
2. Create a matrix in which each element in the matrixwill be an individual market basket.
3. Create a SOM grid with 7 nodes along X and Y axis (7 X 7 SOM layer) (Here we are using rectangular topology for the grid). Also, the right number of nodes will give an ideal information of the data representation
4. Run the SOM modelling using the derived matrix and the grid.



a/b	1	2	3	4	5	6	7
1	1 2	2 14	17 21	5 21	17 18	1	2
2	2 3	3	1 2	4 5	7	8 9	2 12
3	1 8	1 5	8 9	9 11	11 12	9 12	
4	4 7	5 22		8	8	14	14 20
5	4 9	5 14	17 19	6 12			20 22
6	23	21 24	22	22 25	7 23	8 23	22
7	23	23 24	7 19	12 25	12 25	20	21

Figure 6 Output SOM Matrix

Checking several possible alternatives,a 7*7 layer was found to produce the best results with respect to heterogeneity and simplicity of the prototype system as well as the interpretability in content. Each of the 49 fields of the map represents one unit or prototype. To make interpretations easier, however, only those product categories with weights greater than 0.8 have been displayed. (Product IDs were given integer values for better understanding.) Now the product categories which clubbed together represents the products frequently bought together.

V. Conclusion

The purpose of market basket analysis is to determine what products customers purchase together.By targeting customers who are already known to be likely buyers, the effectiveness of marketing can be significantly increased. For the same dataset two Association Rule Mining algorithms were applied (Apriori and SOM) and meaningful rules were generated where SOM does the clustering within itself according to the features of the transaction. You could continue to fine tune the user controlled parameters (minimum support, confidence, improvement and learning factor) until obtaining the results you want.

VI. References

[1] Andrew Moore: "K-means and Hierarchical Clustering – Tutorial Slides"
http://www-2.cs.cmu.edu/~awm/tutorials/kmeans.html

[2] Apriori Algorithm – Classical algorithm for data mining
https://www.techleer.com/articles/155-apriori-algorithm-classical-algorithm-for-data-mining/

[3] Data Mining Using Neural Networks
https://researchbank.rmit.edu.au/eserv/rmit:9493/Rahman.pdf

[4] Self-Organizing Map (SOM)
http://users.ics.aalto.fi/jhollmen/dippa/node9.html

[5] Visualization Association Rules: Introduction to the R-extension Package arulesViz
https://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf

[6] Market Basket Analysis/Association Rule Mining using R package – arules
https://prdeepakbabu.wordpress.com/2010/11/13/market-basket-analysisassociation-rule-mining-using-r-package-arules/

[7] Market basket analysis with neural gas networks and self-organising maps
https://link.springer.com/content/pdf/10.1057%2Fpalgrave.jt.5740092.pdf