International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 13
October 2017

A Study on Soccer Transfers Data Using Market Basket Analysis

ANKITA SHARMA
Asst. Prof. St. Martin's Engineering College, Dulapally Road, Dhulapally, Near Kompally, Hyderabad, Telangana, India.

Abstract: *Although more than 20 years old, Market Basket Analysis (MBA) (or association rules mining) can still be a very useful technique to gain insights in large transactional data sets. The classical example is transactional data in a supermarket. For each customer we know what the individual products (items) are that he has put in his basket and bought. Other use cases for MBA could be web click data, log files, and even questionnaires. To perform MBA we need of course data, but we don't have real transactional data from a retailer that we can present here. So we are using soccer data instead. The data from eleven European soccer leagues starting from season. After some data wrangling we will able to generate a transactional data set suitable for market basket analysis. This analysis will fetch us results in the transformations of the players this helps us in finding the strengths and players dependencies for a team. n our research we used association rules between players and teams in the market basket. These associations show a variety transfers between the players. To show the dependence between the players and teams we used a Web plot.*

Keywords— Association Rule Mining , Apriori Algorithm, Market Basket Analysis.

1 INTRODUCTION

Market basket analysis is essentially the process of determining whether or not a relationship exists in your data between different discrete values. A reason for it being called "market basket" analysis is that it's generally applied to transactional data. A good example would be the products you put in your "basket" to purchase from the farmers' "market." Have you ever noticed that if you're craving a PB&J and you find the bread, you'll probably find the jelly nearby and then, nearby, you'll see the peanut butter? The reason for this is market basket analysis. Stores look at the data they've gathered from sales and see which products are more often purchased together and then design the store so those products are placed together. There is a method to their madness! Let's look at some of the key terms we'll use:

1.1 Rules:

A rule is simply a question about the association of products purchased together. There is a left-hand side and a right-hand side to each rule, and the rules are read from left to right. They're displayed as:

Apple Tart => Apple Croissant

Which would translate to "If you purchase an Apple Tart, what is the likelihood that you also purchase an Apple Croissant?"

1.2 Lift:

The lift is the how many times greater the association is between the sides than just mere chance. This really means that that if a customer purchases an Apple Tart then they are "lift" times more likely to purchase an Apple Croissant. So, if the lift is 2 and I also buy an Apple Tart, then I am twice as likely to also purchase an Apple Croissant.

1.3 Support:

The support is just the percentage of transactions that include the full rule. So, since we said that the rule was Apple Tart => Apple Croissant, then the support is just the percentage of transactions that included an Apple Tart and an Apple Croissant.

1.4 Confidence:

The confidence is a little trickier to define. The easiest way to consider confidence is to imagine all of the Apple Tarts purchased and then find the

percentage of those transactions that also included an Apple Croissant. So, mathematically, the confidence is saying, "Given an Apple Tart was purchased, what is the probability an Apple Croissant was also purchased (this is Bayes Theorem, which can be explained another day)?

Think of lift as the strength of the association or relationship whereas the support and confidence explains how much data actually supports this relationship.

2 RELATED WORKS

Our association analysis was performed using R and then visualized interactively in a R Stuido IDE. The package arules was used along with the corresponding commands for generating the rules, lift, support and confidence. R has an excellent suite of algorithms for market basket analysis in the arules package by Michael Hahsler and colleagues. It includes support for both the Apriori algorithm and the ECLAT (equivalence class transformation algorithm).

2.1 arules : Mining Association Rules and Frequent Itemsets: Arules, open source package available from The Comprehensive R Archive Network, is a powerful tool-set for mining associative rules in transactional databases. The most common use of arules package is market basket analysis in marketing and retail; though, there were successful attempts applying arules to medical problems, crime prevention, and book recommendations. Association mining is commonly used to make product recommendations by identifying products that are frequently bought together. But, if you are not careful, the rules can give misleading results in certain cases. Association mining is usually done on transactions data from a retail market or from an online e-commerce store. Since most transactions data is large, the apriori algorithm makes it easier to find these patterns or rules quickly. So, What is a rule?

A rule is a notation that represents which item/s is frequently bought with what item/s. It has an LHS and an RHS part and can be represented as follows:

itemset A => itemset B

This means, the item/s on the right were frequently purchased along with items on the left. How to measure the strength of a rule?

The apriori() generates the most relevent set of rules from a given transaction data. It also shows the support, confidence and lift of those rules. These three measure can be used to decide the relative strength of the rules. So what do these terms mean?

Let's consider the rule A => B in order to compute these metrics.

$$Support = \frac{Number\ of\ transactions\ with\ both\ A\ and\ B}{Total\ number\ of\ transactions} = P(A \cap B)$$

$$Confidence = \frac{Number\ of\ transactions\ with\ both\ A\ and\ B}{Total\ number\ of\ transactions\ with\ A} = \frac{P(A \cap B)}{P(A)}$$

$$ExpectedConfidence = \frac{Number\ of\ transactions\ with\ B}{Total\ number\ of\ transactions} = P(B)$$

$$Lift = \frac{Confidence}{Expected\ Confidence} = \frac{P(A \cap B)}{P(A).P(B)}$$

The directionality of the rule is lost when lift is used. That is, the lift of any rule, A => B and the rule B => A will be the same. See the calculation below:

A -> B

- Support: $P(A \cap B)$
- Confidence: $\frac{P(A \cap B)}{P(A)}$
- Expected Confidence: $P(B)$
- Lift: $\frac{Confidence}{Expected\ Confidence} = \frac{P(A \cap B)}{P(A).P(B)}$

B -> A

- Support: $P(A \cap B)$
- Confidence: $\frac{P(A \cap B)}{P(B)}$
- Expected Confidence: $P(B)$
- Lift: $\frac{Confidence}{Expected\ Confidence} = \frac{P(A \cap B)}{P(A).P(B)}$

3 Literature Survey

Numerous research works has been done in the field of association rule mining. The availability of huge amount of data has attracted many researchers for mining large data and extracts the knowledge from them. R. Srikant et al. in [2] given that the problems of minimum support (MinSup) and minimum confidence (MinConf) occurs during mapping in Boolean association rules.

The "MinSup" problem can be overcome by combining adjacent intervals/values. But, "MinConf" problem still persists; moreover information loss can be reduced by increasing number of intervals without encountering "MinSup" problem. By increasing number of intervals during parallel combining of adjacent intervals it introduces problem of execution time and generation of many rules. To reduce execution time problem, restrict the extent to which adjacent values/intervals may be combined by introducing maximum support parameter i.e. if their combined support value increases beyond maximum support value then stop combining of intervals. They introduced measure of partial completeness which measures the information lost due to partitioning and also gives information regarding whether to partition or not and if partition is to be done then number of partitions.

Direct application of this technique may generate many similar rules. This problem can be solved by interest measure i.e. greater than expected value used to identify interesting rules. They gave an algorithm for such association rule mining based on Apriori algorithm for finding Boolean association rules. The most basic algorithm for finding the frequent itemsets is the Apriori algorithm [6]. The algorithm works in two major steps join and prune. Each Itemset is considered as candidate 1-itemset. The frequent itemsets that satisfy the support are combined to obtain the candidate set. The candidate set is pruned on the basis of the antimonotone property of the support measure of Apriori which states that "support for an itemset never exceeds the support for its subsets".

The pruning is also guided by the Apriori principle "If an itemset is frequent, then all of its subsets must also be frequent." The database is scanned for all itemset to find whether it is frequent or not. Here, algorithm extends candidate generation procedure of Apriori to add pruning using interest measure. Finally, the frequent itemset are obtained and the algorithm terminates when the entire items are combined. For handling large databases a new approach for compact tree structure is done for compressing the original transaction tree. The compact transaction tree (CT tree) [7] has two parts head and body. The head part contains the item name and frequency count of the item and the body part contains the frequency count of occurrence of item in the transaction. The CT tree construction is done after arranging the item in lexicographical order.

The mining procedure is the modification of the basic Apriori algorithm for finding the rule. The head of the tree is read, skipping the initial scan of the database. The count of the item is done on the basis of occurrence of item. The CT-Apriori algorithm finds frequent patterns from the compact databases. This approach reduces amount of storage space and running time of the algorithm. However, the Apriori candidate generation may require some storage space. Another basic algorithm developed by Han et al. in [8] is the frequent pattern tree growth algorithm (FP-Tree). The FP-tree structure is for storing compressed and vital information about frequent patterns and develops an efficient FP-growth mining algorithm for mining complete frequent patterns. This has an advantage over the Apriori algorithm as it reduces space and time complexity. The algorithm requires two database scans; one is for constructing and ordering frequent patterns and second is for tree branches building. First the frequent item sets are obtained by comparing the support count with the user defined support value. The frequent items are sorted in decreasing order for construction of the FP tree. The sub frequent conditional patterns are extracted from the FP tree for each itemset and the mining is done. Efficiency of mining is achieved as (a) The FP tree construction reduces the repeated scanning of the database and thus time complexity reduces. (b) Mining is done by FP-Growth which does not produce costly large number of candidate set generation. This reduces the space for storage of large candidate sets. However, the FP tree constructed for large data items require large storage space. (c) A divide and conquer method is used to decompose the mining task into a smaller one for mining limited patterns in conditional databases, which reduces the search space. Tannu Arora et al. in [9] proposed Dynamic-FP approach which uses combine approach of Dynamic itemset counting (DIC) and fp tree. DIC algorithm is an extension to Apriori algorithm used to reduce number of scans on the dataset. In this approach, itemsets are dynamically added and deleted as transactions are read. It depends on the fact that for an itemset to be frequent all of its subsets must also be frequent, so these itemsets can only be examined whose subsets are all frequent. Both Apriori and

DIC are based on candidate generation. Dynamic – FP takes advantage of both FP-tree and DIC. Compared to Apriori, DIC generates half database scans. Means DIC is used to reduce number of scans on the dataset. By using combine approach we can mine the frequent itemsets dynamically without any candidate generation. It is better than FP-tree also and as compared to previous algorithm it gives better performance.

4 COMPARATIVE ANALYSIS OF SOCCER DATA

The data contains around 25.000 matches from eleven European soccer leagues starting from season 2008/2009 until season 2015/2016. After some data wrangling we were able to generate a transactional data set suitable for market basket analysis. The data structure is very simple, some records are given in the figure below:

player	season	club
Jose Darado	2011/2012	Real Betis
Jose Darado	2013/2014	Villareal
Jose Darado	2015/2016	Rayo Vallecano
Ruben Perez	2010/2011	Deportivo
Ruben Perez	2011/2012	Getafe
Ruben Perez	2012/2013	Real Betis
Ruben Perez	2013/2014	Elche
Ruben Perez	2014/2015	Granada

So we do not have customers but soccer players, and we do not have products but soccer clubs. In total, my soccer transactional data set contains around 18.000 records. Obviously, these records do not only include the multi-million transfers covered in the media, but also all the transfers of players nobody has ever heard of.

In R you we can use the arules package for MBA / association rules mining. Alternatively, when the order of the transactions is important, like my soccer transfers, you should use the arulesSequences package. After running the algorithm we got some interesting results. The figure below shows the most frequently occurring transfers between clubs:

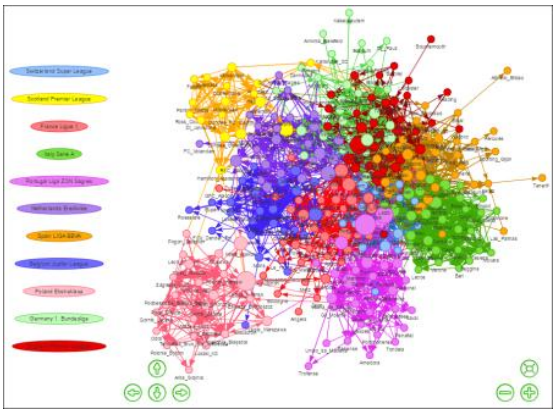
Show 25 entries Search:

from	to	support_perc	Ntransctions
Fiorentina	Genoa	0.1138 %	12
Kortrijk	Gent	0.1043 %	11
Catania	Chievo	0.0949 %	10
Polonia_Warsaw	Zaglebie_Lubin	0.0854 %	9
Young_Boys	Thun	0.0854 %	9
Standard_Liege	St_Truiden	0.0854 %	9
Atalanta	Sassuolo	0.0854 %	9
Fiorentina	Sampdoria	0.0854 %	9

So in this data set the most frequently occurring transfer is from Fiorentina to Genoa (12 transfers in total).

Network graph visualization:

All the rules that we get from the association rules mining form a network graph. The individual soccer clubs are the nodes of the graph and each rule "from ==> to" is an edge of the graph. In R, network graphs can be visualized nicely by means of the visNetwork package. The network is shown in the picture below.



The different colors represent the different soccer leagues. There are eleven leagues in this data, there are more leagues in Europe, but in this data we see that the Polish league is quite isolated from the rest. Almost blended in each other are the German, Spanish, English and French leagues. Less connected are the Scottish and Portuguese leagues, but also in the big English Premier and German leagues you will find less connected clubs like Bournemouth, Reading or Arminia Bielefeld.

The size of a node in the above graph represents it's betweenness centrality, it is an indicator of a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node. In R betweenness measures can be calculated with the igraph package. The most central clubs in the transfers of players are Sporting CP, Lechia

Gdansk, Sunderland, FC Porto and PSV Eindhoven.

Virtual Items:

An old trick among marketeers is to use virtual items in a market basket analysis. Besides the 'physical' items that a customer has in his basket, a marketeer can add extra virtual items in the basket. These could be for example customer characteristic like age-class, sex, but also things like day of week, region etc. The transactional data with virtual items might look like:

customer	time	item
ABCD	1	apples
ABCD	2	beer
ABCD	3	cheese
ABCD	4	MALE
ABCD	5	WEDNESDAY
ABCD	6	[18:23]
XYZ	1	bread
XYZ	2	eggs
XYZ	3	MALE
XYZ	4	MONDAY
XYZ	5	[65:100]

If you run a MBA on the transactional data with virtual items, interesting rules might appear. For example:

{Chocolate, Female ==> Eggs}

{Chocolate, Male ==> Apples}

{Beer, Friday, Male, Age[18:23] ==> sausages}.

Virtual items that we can add to our soccer transactional data are: age-class, four classes: 1: players younger than 25, 2: [25, 29), 3: [29, 33) and 4: the players that are 33 or older. Preferred foot, two classes: left or right. Height class, four classes: 1; players smaller than 178 cm, 2: [178, 183), 3: [183, 186), and 4: players taller than 186 cm.

After running the algorithm again the results allow us to find out the frequently occuring transfers of let-footers. we can see 4 left footers that transferred from Roma to Sampdoria.


Show 25 entries Search:

sequence	Ntraasfers	support_perc
left	All	All
<{Virt_foot_left},{Roma},{Sampdoria}>	4	0.0362 %
<{Virt_foot_left},{Kortrijk},{Gent}>	4	0.0362 %
<{Virt_foot_left},{Polonia_Warsaw},{Zaglebie_Lubin}>	3	0.0271 %
<{Virt_foot_left},{FC_Zuerich},{Young_Boys}>	3	0.0271 %
<{Virt_foot_left},{Bayern_Munich},{Wolfsburg}>	3	0.0271 %
<{Virt_foot_left},{Feyenoord},{Willem_II}>	3	0.0271 %
<{Virt_foot_left},{ADO_Den_Haag},{Willem_II}>	3	0.0271 %
<{Virt_foot_left},{Manchester_City},{Sunderland}>	3	0.0271 %
<{Virt_foot_left},{AZ_Alkmaar},{PSV_Eindhoven}>	3	0.0271 %
<{Virt_foot_left},{Polonia_Bytom},{Podbeskidzie_Bielsko_Biala}>	3	0.0271 %

5 CONCLUSION

When we have transactional data, even as small as the soccer transfers, market basket analysis is definitely one of the techniques we should try to get some first insights.

6 REFERENCES

[1] J. Han and M. Kamber, "Data mining: Concepts and techniques (2nd edition)," Morgan Kaufman Publishes, 2006.

[2] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," In Proc. Conf. Management Data ACM SIGMOD, pp. 1–12, 1996.

[3] Farah Khan and Dr. Divakar Singh, "Knowledge Discovery on Agricultural Dataset Using Association Rule Mining," International Journal of Emerging Technology and Advanced Engineering, pp. 925 – 930, 2014.

[4] Akash Rajak and Mahendra Kumar Gupta", Association Rule Mining: Applications in Various Areas," International Conference on Data Management Ghaziabad, pp. 3-7, 2008.

[5] T. Karthikeyan and N. Ravikumar, "A Survey on Association Rule Mining," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), pp. 5223-5227, 2014.

[6] Agrawal R., Imielimski T. and Swami A., "Mining Association Rules between sets of items in large database," Proceedings of ACM

SIGMOD International Conference Management of Data, pp. 207- 216, 1993.

[7] Qian Wan and Aijun An, "Compact transaction database for efficient frequent pattern mining," IEEE International Conference on Granular Computing, pp. 652 - 659, 2005.

[8] Jiawei Han, Jian Pei, and Yiwen Yin, "Mining frequent patterns without candidate generation," International conference on management of data (ACM SIGMOD), pp. 1–12, 2000.

[9] Tannu Arora and Rahul Yadav, "Improved Association Mining Algorithm for Large Dataset," IJCEM International Journal of Computational Engineering and Management, pp. 36-38, 2011.

[10] C.I. Ezeife, "Mining Incremental Association rules with Generalized FP-tree," Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence, Springer, pp. 147-160, 2002.

[11] A. Vedula Venkateswara Rao and B. Eedala Rambabu, "Association rule mining using FP tree as Directed Acyclic Graph," International Conference on Advances in Engineering, Science and Management (ICAESM), pp. 202 - 207, 2012.

[12] Ashish Mangalampalli and Vikram Pudi, "Fuzzy Association Rule Mining Algorithm for Fast and Efficient Performance on Very Large Datasets," International Conference on Fuzzy Systems, IEEE, pp. 1163 – 1168, 2009.

[13] R. Prabamanieswari, "A Combined Approach for Mining Fuzzy Frequent Itemset," International Journal of Computer Applications (IJCA), International Seminar on Computer Vision, pp. 1–5, 2013.

[14] C.M. Kuok, A. W.-C. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," ACM SIGMOD, pp. 41–46, 1998.