

Borrowing Patterns Monitoring in Library: Application of Apriori algorithm

¹Adewole K. S., ² Akintola A. G., ³ Mabayoje M. A., ⁴ Ogbomon G. A.

^{1,2,3,4}Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, P.M.B. 1515, Ilorin, Kwara-State, Nigeria.

¹adewole.ks@gmail.com, ²abimbolar@yahoo.com, ³[mmabayoje@yahoo\(gmail\).com](mailto:mmabayoje@yahoo(gmail).com),

Abstract

Data is a valuable tool for any institution, and with the world advancing in technology, data stored in database management systems are growing in different capacities in almost all organization. The opportunities of database management systems have been explored. However, many organizations have not been able to leverage these opportunities in gaining business intelligence from their repositories. This paper addresses the issue of knowledge discovery from large databases using association rule mining. Apriori algorithm is implemented to discover hidden knowledge from a library database. Data depicting nine (9) different books were used within forty-seven (47) unique transactions. Eighteen (18) unique transactions were generated from the database showing the borrowing pattern of library users. The frequencies of borrowing of books were obtained as well as the associations. The result shows that borrowing a particular book may leads to borrowing another book as revealed in the association between Data structure in C (DS) textbook and Programming in C (C) textbook. The discovered pattern can help librarians in restructuring their bookshelf arrangement, and for book recommendation system. This system can also help students to have good knowledge different related books.

KEYWORDS: Data Mining, Association Rule, Market Basket, Apriori Algorithm, Mining Support, Confidence.

Introduction

Libraries form a major part of any academic institution or organization. They are the basis of every educational institution and as a result should be well organized so that it can be used efficiently. The library management systems available today are designed to manage books, journals, manuscripts etc. They keep track of the books issued or returned and tell about the availability of the books in the libraries. From the transactional database of such library management system, the behavior of readers in terms of the books they borrow can be examined. For example, a set of books, web design and web development that appear frequently as well as together in a database is a pattern set. This patterns may reveal that if a reader borrows a literature textbook, he might also borrow a dictionary. From the history of the database, these borrowings occur sequentially, and this is known as frequent pattern. Informed decision can be made easily about book placement, ordering, lending,

and to find out if there are any successful books that have no significant related elements. Similar books can be found so that they can be placed in shelf near to each other.

A librarian must know the needs of library users and adapt to them. In the same vain, pattern analysis is one possible way to finding out which books can be put together. According to Pillai and Vyas (2011), market basket analysis gives retailer good information about related sales on group of goods basis. In other words, pattern analysis is a key feature to effective book arrangement in the libraries. Customers who buy bread often also buy several products related to bread such as milk, butter or egg. It makes sense that these items are placed side-by-side in a retail center so that customers can access them quickly. Such related group of goods also must be located side-by-side in order to remind customers of related items and to lead them through the center in a logical manner.

Market basket analysis is one of the data mining methods focusing on discovering purchasing patterns by extracting associations or co-occurrences from a store's transactional data. It determines the products, which are bought together and to reorganize the supermarket layout, and also to design promotional campaigns such that products purchase can be improved (Pillai and Vyas, 2011).

Furthermore, the market consumer behaviors need to be analyzed, which can be done through different data mining techniques. Data mining finds interesting patterns from databases such as association rules, correlations, sequences, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems.

Association rule mining finds interesting association or correlation relationships among a large set of data items. Association rules are derived from the frequent itemsets using support and confidence as threshold levels. The sets of items, which have minimum support is known as frequent itemset. The support of an itemset is defined as the proportion of transactions in the data set, which contain the itemset. Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern. Association rules derived depends on confidence. Frequent itemset generation is done using data mining algorithms like Apriori, K-Apriori, Eclat and FP-Growth algorithms (Nasreen, Azam, Shehzad, Naeem and Ghazanfar, 2014). Apriori algorithm is used in this paper due to it wide application in association rule mining. The remainder of this paper is organized as follow: Section 2 discusses the related work on association mining; Section 3 presents the methodology applied in this paper; Section 4 focuses on results and discussion; and Section 5 concludes the paper.

Related Works

Apriori algorithm is a level-wise, breadth-first algorithm, which counts transactions (Annie and Kumar, 2011). Apriori algorithm uses prior knowledge of frequent itemset properties. Apriori uses an iterative approach known as a level-wise search, in which n -itemsets are used to explore $(n+1)$ -itemsets. To improve the efficiency of the level-wise

generation of frequent itemsets, the Apriori property insists that all non-empty subsets of a frequent itemset must also be frequent. This is made possible because of the anti-monotone property of support measure - the support for an itemset never exceeds the support for its subsets. A two-step process consists of join and prune actions are done iteratively.

Agrawal and Srikant (1997) proposed the Apriori algorithm to solve the problem of mining frequent itemsets. According to them, Apriori uses a candidate generation method, such that the frequent, k -itemset in one iteration can be used to construct candidate $(k+1)$ -itemsets for the next iteration. Apriori terminates its process when no new candidate itemsets can be generated. Apriori is an algorithm for mining frequent item sets for Boolean association rule. This name is based on the fact that the algorithm uses prior knowledge of frequent item set properties. Apriori search an iterative approached which is known as level-wise search, where k item set are used to explore $(k+1)$ item sets. In this algorithm two steps is used for iteration. The first step generates a set of candidate item sets and the second step is to count the occurrence of each candidate set in database (all infrequent item sets). Apriori uses two pruning technique, first on the bases of support count (should be greater than user specified support threshold) and second for an item set to be frequent , all its subset should be in last frequent item set. The iterations begin with size 2 itemsets and the size is incremented after each iteration (Sujata, Priyanka, Sachin & Kuthe, 2015).

Direct Hashing and Pruning (DHP) algorithm, proposed by Park, Chen and Yu (1997), improves the performance of Apriori. It uses a hash table to filter the infrequent candidate 2-itemsets and employs database trimming to lower the costs of database scanning. However, the aforementioned methods cannot avoid scanning the database many times to verify frequent itemsets. Jianying and Mojsilovic (2007) also presented an algorithm for frequent itemset mining that identifies high-utility item combinations. In contrast to the traditional association rule and frequent item mining techniques, the goal of the algorithm is to find segments of data, defined through combinations of few items (rules), which satisfy certain conditions as a group and maximize a predefined objective function. They formulated the task as an optimization problem, presented an efficient approximation to solve it through specialized partition trees, they called it High-Yield Partition Trees.

Unlike Apriori, the FP-growth method by Han, Pei, Yin, and Mao (2004) uses an FP-tree to store the frequency of information of the transaction database. Without candidate generation, FP-growth uses a recursive divide-and-conquer method and the database projection approach to find the frequent itemsets. However, the recursive mining process may decrease the mining performance and raise the memory requirement. Grahne and Zhu (2005) however introduces an FP-growth*, which uses an FP-array technique to reduce the need to traverse FP-trees. Nevertheless, it still has to generate conditional FP-trees for recursive mining. Their experimental results show that running time and memory consumption of FP-growth* is almost equal to that of FP-growth. Racz (2004) improves FP-growth algorithm and employed the tree structure to raise the mining performance. The FP-growth method of Han (2000) and Han et al. (2004) requires two database scans when

mining all frequent itemsets. The first scan of the database derives the set of frequent 1-itemsets, denoted as F1, and then inserts F1 into a header table in the decreasing order of items' supports, where this order list is denoted as List. The header table not only stores items and their supports, but also contains a pointer to a list that links all corresponding nodes of frequent 1-itemset in the FP-tree. In the second scan, the items of F1 in each transaction are inserted into an FP-tree in Loader as a path. Distinct paths that have the same prefix path are merged to construct the FP-tree for reducing memory space.

The FP-growth method is efficient, but it has a potential drawback. The major work of mining frequent itemsets in FP-growth is recursively constructing new conditional FP-trees and traversing paths on these trees. When the FP-tree becomes huge, the recursive mining on the tree may decrease the mining performance drastically. Thus, Park et al. (1997) proposed an implementation called nonordfp to improve the FP-growth algorithm. Nonordfp is based on a more compact structure called tree for constructing FP-tree. Each node in the tree contains a counter and a pointer to the parent. The nodes are stored in an array. Thus, the task of traversing paths becomes reading the array sequentially. Another major feature of Nonordfp is that the recursive mining processes can be replaced by building new tree structures, which avoids rebuilding each conditional pattern base. Experimental results performed demonstrated that nonordfp is very efficient. Ke-Chung, Liao and Chen (2011) revealed that nonordfp incurs a space problem, as it requires continuous memory space for the array structure.

Despite the comprehensibility and simplicity, Apriori algorithm suffers from some weaknesses. The main restraint is waste of time to hold a vast number of candidate sets with much frequent itemsets, low minimum support or large itemsets. For example, if there are 104 frequent 1-itemsets, it need to generate more than 107 candidates into 2-length which in turn will be tested and accumulate. Furthermore, to detect frequent pattern in size 100 v1, v2... v100, it have to generate 2100 candidate itemsets that yield on costly and wasting of time of candidate generation. So, it will check for many sets from candidate itemsets, also it will scan database many times repeatedly for finding candidate itemsets. Apriori will be very low and inefficiency when memory capacity is limited with large number of transactions (Rao & Gupta, 2012).

Xin-hua, Ya-qiong & Qing-ling(2010)in their paper took grades of computer cultural foundation course as example. The weighted association rules algorithm was used to analyze grades of college-wide examination course in the paper, giving the model and mining method of mixed weighted association rules on grade database. Compared to directly apply the Apriori algorithm, more valuable correlations were obtained between the chapters, chapters and scores, colleges and chapters at the same threshold values. Thus, it is more helpful to guide teachers for their teaching and for the improvement of teaching quality. Apriori algorithm is not based on hardware implementation. However, research in hardware implementations of related data mining algorithms has been done (Estlick, Leeser, Szymanski & Theiler, 2001). In (Estlick et. al., 2001) the k-means clustering algorithm in implemented as an example of a special fabric in the form of a cellular array connected to a host processor. K-means clustering is data mining and that

groups together elements 2 based on a distance measure. This distance can be an actual measure of Euclidean distance. Item is a set is randomly assigned to a cluster, the centers of the clusters are computed, and then elements are added and removed from clusters to more efficiently move them closer to the centers of the clusters. This is based on the Apriori algorithm both are dependent on efficient set additions and computations performed on all elements of those sets. In the overall structure of apriori algorithm, the structure of the computation is also significantly different, as the system requires the use of glob.

In market basket analysis, medical diagnosis/ research, website navigation analysis, homeland security association rule mining plays very important role. As compare to the conventional algorithm, frequent item will take less time. It is considered in data mining to have key ideas of reducing time. Jayshree and Leena (2013) proposed an Apriori algorithm based on Bottom up approach and Support matrix to identify frequent item set. The algorithm replaces arbitrary user defined minimum support with functional model based on Standard Deviation. In proposed algorithm, Minimum support value was calculated based upon Standard Deviation value of support counts of all transactions. This approach makes this algorithm more comfortable for non-expert in data mining. The algorithm uses Bottom up Approach to find the frequent item set from largest frequent Item set to smallest frequent item set which help in mining long pattern easily. This algorithm works in 2 phases, Support Matrix Generation and Bottom Up approach to mine frequent items set based upon calculated minimum support. The proposed algorithm replaces the user-defined minimum threshold with standard deviation based functional model and it may lead to either too many or too few rules which can negatively impact the performance of entire model.

An Apriori algorithm was also proposed through reducing the time consumed in transactions scanning for candidate itemsets by reducing the number of transactions to be scanned. Whenever the k of k -itemset increases, the gap between the improved Apriori and the original Apriori increases from view of time consumed, and whenever the value of minimum support increases, the gap between the improved Apriori and the original Apriori decreases from view of time consumed. The time consumed to generate candidate support count in the improved Apriori is less than the time consumed in the original Apriori. Time consuming in improved Apriori in each value of minimum support is less than it in the original Apriori, and the difference increases more and more as the value of minimum support decreases. It showed that the improved Apriori reduce the time consuming by 84.09% from the original Apriori where the minimum support is 0.02, and by 56.02% in 0.10. As the value of minimum support increase the rate is decreased also. The average of reducing time rate in the improved Apriori is 68.39% (Mohammed & Bassam, 2014).

Methodology

Association rule mining links one or more attributes of a dataset with another attributes, to discover hidden and significant relationship between the attributes, producing an if-then

statement concerning attribute values in the form of rules. According to Abdusalam, Adewole, Akintola & Hambali (2014), an association rule states that if a user is pick at random and find out that he/she selected some item (that is, bought some products), it can be assured by indicate the quantity by a percentage that he/she also bought some other products. Agrawal, Imielinki & Swanmi (1993), 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 {smartphone, memory card} = { sim card } found in the sales data of a supermarket would indicate that if a customer buys smartphone and memory card together, he or she is likely to also buy Sim card. Such information can be used as the basis for decisions about marketing activities such as promotional pricing or product placements. In addition to the above example from market basket analysis, association rules are employed today in many application areas including Web usage mining, intrusion detection and bio-informatics etc.

Association rule mining is one of the most popular data mining approaches. It is used to discover interesting relationships between variables in databases. According to Agrawal et al (1993), an association rule explains a close correlation between items in a database in the form of $x \Rightarrow y$ where x and y are sets of Itemset I (x and $y \subset I$) and $x \cap y = \Phi$. $I = I_1, I_2, \dots, I_m$ is an Itemset of m distinct attributes. The rule indicated x implies y whereby x is called antecedent and y is called consequent. There are two importance thresholds for the measurement of association rule mining, they are minimum support and minimum confidence (Kanyarat et al., 2014). The support of a rule $x \Rightarrow y$ is the probability of the Itemset $\{x, y\}$ that means the relevance of the rule and the confidence of a rule $x \Rightarrow y$ is the conditional probability of y given x ; that indicate the accuracy of the rule over the set of transactions: $D = \{d_1, d_2, \dots, d_n\}$ each $d_i \subset I$ Therefore, confidence is an important measure of the association rules to indicate the strength rules. If the confidence of the association rule $x \Rightarrow y$ is 80%, it means that 80% of the transactions that contain x also contain y (Kanyarat et al., 2014).

Structure of tables

Table 1 consist of a list of the books in the database and with their respective BookID. This are samples to demonstrate the operation of frequent itemset mining. This book may in large collection depending on the dataset under consideration.

Table 1: A list of books and BookID

Book Title	BookID
Programming in C	C
Programming in c++	C++
SQL and PL/SQL	DB1
Database Systems	DB2
Data structures through C	DS
Software Engineering	SW
.NET Framework	.NET

Table 2 consist of books borrowed in different transaction according to transaction ID, which is auto incremented. This shows the different combinations recorded from borrowing pattern of library users. It is important to note that, these transactions are supplied to the apriori algorithm during the experimental stage. The results obtained are discussed in section 4.

Table 2 Transaction Table

TID	Item set
1.	DS,C,.NET,DB1,DB2
2.	.NET,DB1,C,DS,DB2
3.	SW,C++,C,.NET,DB1
4.	SW,C,DS,DsII
5.	.NET,SW,DB1,C,DS1
6.	C,.NET,DB1,SW,DS,DB2
7.	DB2,DS,C,C++,DB1
8.	DB2,DB1,SW II
9.	DB1,C,C++,DS,.NET
10.	.NET,DB1,DB2,SW

Table 3 is the BorrowedBooks table, this table selected the different books a user borrowed per transaction.

Table 3: Borrowed Books

TID	Item-set
1.	C,C++,DB1
2.	DS,C,SW,.NET
3.	.NET,DB1,C
4.	DB2,DB1,SW,DS
5.	DB1,DS,.NET,C,C++

Apriori algorithm

Agrawal et al (1997), proposed the well-known algorithm, Apriori, to mine large itemsets to find out the association rules among items. This algorithm employs a level-wise approach, which iteratively generates candidate k -itemsets from previously found frequent $(k - 1)$ -itemsets, and then checks the supports of candidates to form frequent k -itemsets. The algorithm scans multiple passes over the database. The efficiency and correctness of the level-wise generation of frequent itemsets are based on an important property called the Apriori property. The algorithm in first pass counts item occurrences to find the set of frequent 1-itemsets, denoted as L_1 . A subsequent pass, say pass k , consists of two steps:

the join and prune steps. In the join step, a set of candidate k -itemsets (denoted as C_k) is generated by joining the frequent itemsets L_{k-1} found in the $(k - 1)$ th pass with itself.

Apriori algorithm is an influential algorithm for mining frequent itemsets from large repository of data for Boolean association rules. It extracts the frequent item sets from the candidate itemsets. Support for the frequent itemsets must be greater than minimum support that defined by user. Generally, there are two main steps to implement a priori algorithm. Firstly, determine the frequent item sets. Secondly, generate the association rules from the frequent item sets that fulfil the requirement of minimum confidence. A priori algorithm contains a number of passes over the database. During pass k , the algorithm finds the set of frequent item sets L_k of length k that satisfies the minimum support requirement. The algorithm terminates when L_k is empty. A pruning step eliminates any candidate, which has a smaller subset. This algorithm is described as follows (Kanyarat et al., 2014):

- 1) $L_1 = \{\text{large 1-itemsets}\};$
- 2) **for** ($k = 2; L_{k-1} \neq \emptyset; k++$) **do begin**
- 3) $C_k = \text{apriori-gen}(L_{k-1});$ // New candidates
- 4) **forall** transactions $t \in \mathcal{D}$ **do begin**
- 5) $C_t = \text{subset}(C_k, t);$ // Candidates contained in t
- 6) **forall** candidates $c \in C_t$ **do**
- 7) $c.\text{count}++;$
- 8) **end**
- 9) $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$
- 10) **end**
- 11) $\text{Answer} = \bigcup_k L_k;$

Discussion of Findings

The data mining software was developed using Java programming language. Figure 1 shows the Main Menu to access the interface where different transactions can be made.

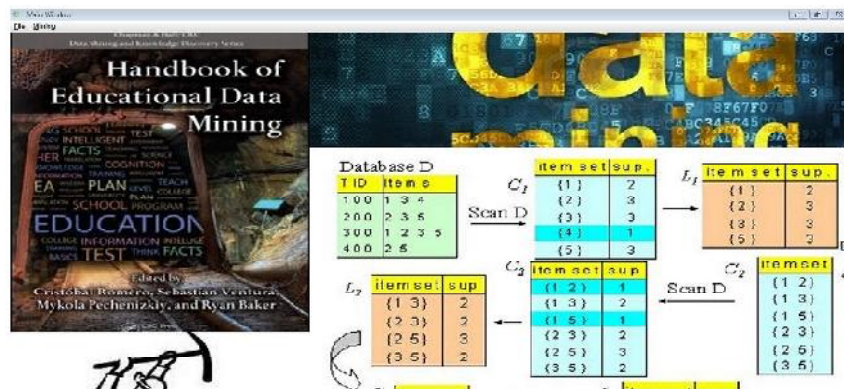


Figure 1: Main Menu

Figure 2 is an interface consisting of the “generate frequent item” and “generate association rule” buttons as well as a textfield; the textfield allow the input of the minimum support value, which is an integer. On supply of the minimum support value and clicking of the generate button, a list of frequent items that meet the minimum support value are displayed in the Item-set column and the serial number which is auto generated are displayed in the Serial column

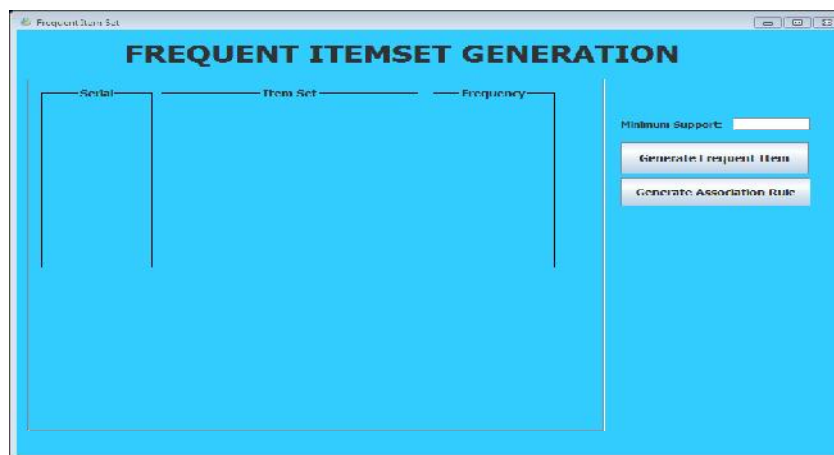


Figure 2 Interface Design

Figure 3 is a transaction interface showing frequent itemset generation after supplying the minimum support of value 4, as shown above and clicking on Generate Frequent Item button.

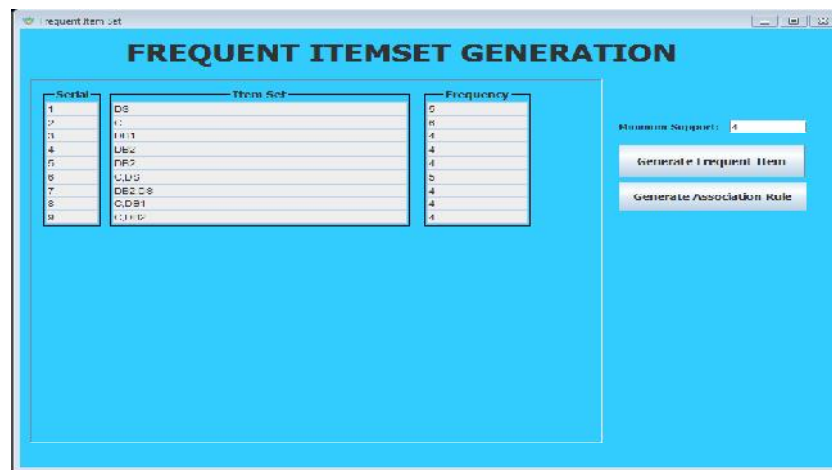


Figure 3: Transaction Interface



Figure 4 Output

Figure 4 shows the association according to the frequent itemset generated by the minimum support of 4 and the click of the generate association rule button. The result as shown above are the serial number of the items, association rules, rule support and rule confidence column. This interface also consists of minimum rule count 18, which is the total number of rules generated, and minimum confidence bar. To interpret this result, let us consider the item at serial number 1. This association rule generated indicates that, if a library user borrows the book with DC code, there is a possibility that the same user will borrow book with C code. Similar interpretation applies to other association rules. Association rules generated by Apriori algorithm can be pruned by defining a threshold on

the rule confidence. For instance, if we are interested in rule with ≥ 0.6 confidence, the implication of this is that rule with serial 11, 12, 14, 15, and 17 will be omitted. This association reveals the users strong interest, and it provides an insight into the reader borrowing patterns. This pattern can be used by librarian to re-organize the bookshelf, and to make an informed decision about users reading habit.

Generally, the issue of knowledge discovery from large databases using association rule mining was swotted. Apriori algorithm was implemented to discover hidden knowledge from a library database. Data depicting nine (9) different books were used within forty-seven (47) unique transactions. Eighteen (18) unique transactions were generated from the database showing the borrowing pattern of library users. The frequencies of borrowing of books were obtained as well as the associations. The result shows that borrowing a particular book may leads to borrowing another book as revealed in the association between Data structure in C (DS) textbook and Programming in C (C) textbook.

Conclusion

By means of pattern analysis of readers, the readers groups with different interests, hobbies and reading habits are acquired. Then, the lending information in different groups will help in performing association rules mining to acquire correlation information of various books, and to develop personalized books recommendation system. In addition, the implementation process of data mining technology is used to design a model for personalized information recommendation system. This paper highlights the processes involve in association mining by presenting a simple but important implementation of Apriori algorithm in studying readers borrowing patterns. This helps us to gain insight from the readers borrowing patterns, and to provide better way of organizing bookshelf. It is found that the correlation of borrowed books resources forms book recommendation pattern with different borrowing preferences, reading habits, and so on.

Recommendations

In this paper, traditional Apriori algorithm was used to discover interesting patterns in readers borrowing habits. This work can be extended by applying a more refined association mining algorithm to remove the need for candidate itemsets generation. This will improve the processing speed of the proposed approach in this paper. Therefore, algorithm such as AprioriDP and FP-growth are good candidates for this objective. In addition, there is a need to implement these algorithms on large datasets collection to provide us with better understanding of readers borrowing habit in a university library, and at the same time compare the performance of the various association mining algorithms in pattern recognitions.

References

- Abdusalam S., Adewole. K., Akintola A. & Hambali M. (2014).Data Mining in Market Basket Transaction: An Association Rule Mining Approach. *International Journal of Applied Information Systems (IJAIS)*. 7(10). 15-20.
- Agrawal R & Srikant R (1997).Mining Generalized Association Rules. *Future Generation Computer Systems*, 13(2--3)
- Agrawal, R., Imielinski, T. & Swami, A.(1993): Mining Association Rules between Sets of Items in Large Databases. *Proceedings of the 1993 ACM SIGMOD Conference on Management of data*. 207-216.
- Annie L. & Kumar A. (2012).Market Basket Analysis for a Supermarket based on Frequent Itemset Mining. *International journal of Computer Issues*, 5(3), 257-264
- Estlick, M., Leeser, M., Szymanski, J. and Theiler, J. (2001). Algorithmic Transformations in the Implementation of K-means Clustering on Reconfigurable Hardware. In *Proceedings of the Ninth Annual IEEE Symposium on Field Programmable Custom Computing Machines (FCCM 01)*, p 103-110
- Grahne, G., & Zhu, J. (2005). Fast algorithms for frequent itemset mining using FP-trees. *Knowledge and Data Engineering, IEEE Transactions*. 17(10), 1347-1362.
- Han J, Pei J (2000) Mining frequent patterns by pattern-growth: methodology and implications. *SIGKDD Explorations Newsletter* 2(2), 14–20
- Han, J., Pei, J., Yin, Y. & Mao, R. (2004). Mining Frequent Pattern without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8(1) 53- 87.
- Ke-Chung, L., Liao, I. & Chen. Z. (2011).An improved frequent pattern growth method for mining association rules. *Expert Systems with Applications* 38. 5154–5161
- Jianying, H. & Mojsilovic, A. (2007). High-utility pattern mining: A method for discovery of highutility item sets. *Pattern Recognition*, 40(11), 3317-3324.
- Mohammed, A. & Bassam, A. (2014). An Improved Apriori Algorithm for Association Rules. *International Journal on Natural Language Computing (IJNLC)* 3(1), 21-29
- Nasreen S., Azam M. A., Shehzad K., Naeem U. & Ghazanfar M. A. (2014).Frequent Pattern Mining Algorithms for Finding Associated Frequent Patterns for Data Streams: A Survey. *The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks*, Elsevier 37, 109-116
- Park-Chen-Yu Park, J., Chen M.& Yu P. (1997). Using a hash-based method with transaction trimming for mining association rules. *IEEE Transactions on Knowledge and Data Engineering*. 9(5) 813-824
- Pillai J. & Vyas O.(2011). Overview of Itemset Utility Mining and its Applications. *International Journal of Computer Applications*. 5(11) 9-13.
- Racz. B. (2004): nonordfp: An FP-growth variation without rebuilding the FP-tree. In *Proceedings of ICDM Workshop on Frequent Itemset Mining Implementations*. volume 126 of *CEUR Workshop Proceedings*, Brighton, UK.
- Rao, S. & Gupta, R. (2012): Implementing Improved Algorithm Over APRIORI Data Mining

Association Rule Algorithm. International Journal of Computer Science and Technolog. 489-493.

Sujata, S., Priyanka, J., Sachin, C. and Kuthe, A. (2015). Apriori Algorithm Using Data Mining. International Journal of Computing and Technology (IJCAT), 2(3), 107-110.

Xin-hua Z., Ya-qiong D. and Qing-ling Z. (2010). The Analysis on Course Grade of College-wide Examination Based on Mixed Weighted Association Rules Mining Algorithm. In proceedings of the 14thIEEE International Conference on Computer Application and System Modeling (ICCASM), , 530 - 533

Authors' Biography



Akintola Abimbola Ganiyat is a Lecturer in the Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, Ilorin. She obtained a Bachelor degree (B.Sc.) in Computer Science from the Bowen University Iwo, Osun State, Nigeria in 2007, where she graduated with a Second Class Honours (Upper Division). She holds a Master's degree (M.Sc.) in Computer Science from the University of Ilorin in 2011. She is currently pursuing a doctoral degree (Ph.D.) in Computer Science in University of Ilorin. Her research interests areas include Biometrics, Data Mining and Software Engineering. She can be reached through akintola.ag@unilorin.edu.ng