

Simple Modification for an Apriori Algorithm with Combination Reduction and Iteration Limitation Technique

Adie Wahyudi Oktavia Gama^{a, 1, *}, Ni Made Widnyani^{b, 2}

^a Department of Information Technology, Universitas Pendidikan Nasional
Bedugul St Number 39, Denpasar, 80224, Indonesia

^b Department of Digital Business, Universitas Bali Internasional
Seroja St Jeruk Block Number 9, Denpasar, 80239, Indonesia

¹ gama.adiewahyudi@gmail.com *; ² nimadewidnyani90@gmail.com

* corresponding author

ARTICLE INFO

ABSTRACT

Article history:

Received 19 September 2020

Revised 13 October 2020

Accepted 03 November 2020

Published online 31 December 2020

Keywords:

Data Mining

Association Rules

Apriori Algorithm

Frequent Itemset

Apriori Optimization

Apriori algorithm is one of the methods with regard to association rules in data mining. This algorithm uses knowledge from an itemset previously formed with frequent occurrence frequencies to form the next itemset. An a priori algorithm generates a combination by iteration methods that are using repeated database scanning process, pairing one product with another product and then recording the number of occurrences of the combination with the minimum limit of support and confidence values. The a priori algorithm will slow down to an expanding database in the process of finding frequent itemset to form association rules. Modification techniques are needed to optimize the performance of a priori algorithms so as to get frequent itemset and to form association rules in a short time. Modifications in this study are obtained by using techniques combination reduction and iteration limitation. Testing is done by comparing the time and quality of the rules formed from the database scanning using a priori algorithms with and without modification. The results of the test show that the modified a priori algorithm tested with data samples of up to 500 transactions is proven to form rules faster with quality rules that are maintained.

This is an open access article under the CC BY-SA license
(<https://creativecommons.org/licenses/by-sa/4.0/>).

I. Introduction

Management information systems or systems that are related to transactions will produce data that continue to grow every time a process is carried out. Those data that continue to grow are not balanced with the acquisition of information for other decision support. The information produced is usually monotonous in the form of daily, weekly or annual reports. This phenomenon is often referred to as "rich information poor data", which means an increase in the amount of data is not comparable to the information obtained. This is due to a lack of analysis of the data stack.

Data mining is the solution to this phenomenon. Data mining is a method that applies data analysis and algorithms to create a specific identification of designs or model over the data [1]. Data mining will analyse large data to extract valuable new information or knowledge. One of data mining techniques that can be used to explore the relationship of new knowledge in the form of a combination of items hidden in the database is association analysis. The relationship can be represented in the form of association rules [2][3]. Association analysis measures the relationship between two or more hidden items in the database. The form of association rules is "If Antecedent Then Consequent", which means that the relationship strength of a product is determined when it is purchased with other products. The strength of the relationship of an associative rule can be measured by two parameters called support and confidence. Support value is the percentage of occurrence of a combination of items in the database. Confidence value or value of certainty reflects the strength of the relationships between items forming combinations in associative rules formed by the association.

Apriori algorithm is one of the forming rules of association in data mining. The initial research conducted by Agrawal in 1993 with the title "Mining Association Rules Between Sets of Items in Large Databases" was the beginning of the development of association methods using apriori algorithms [4]. In 1994, Agrawal and Srikant again conducted research with the association method entitled "Fast Algorithms for Mining Association Rules" [5]. The research was then focused on refining apriori algorithms that had been developed previously and from there apriori algorithm was known as one of the association rules forming algorithms. Apriori algorithm takes an iterative approach, that is, generating k -itemset that is used to form the next $(k + 1)$ -itemset. The principle of apriori algorithm is if an itemset often appears frequently, then all subsets of the itemset must also appear frequently in all transactions stored in a database [2].

In this algorithm candidate $(k + 1)$ -itemset is generated by combining two itemset on domain / size k . Candidates of $(k + 1)$ -itemset containing the frequency of subset that rarely appears or below the threshold will be trimmed and not used in determining association rules [2]. In accordance with association rules, apriori algorithms also use minimum support and minimum confidence to determine itemset rules which are suitable for use in decision making.

1-itemset is used to find 2-itemset, which is a combination of 2 items, for example, if buy Shirt then buy Long Pants. 2-itemset is then used to find 3-itemset which is a combination of 3 items, for example if you buy Shirts and buy pens then buy Long Pants and so on until there are no more k -itemset that can be found in the database transaction [6].

Apriori reasoning uses prior knowledge of an itemset with frequent occurrence frequencies. It uses an iterative approach where k -itemset is used to explore $(k + 1)$ -itemset [6]. Candidate $(k + 1)$ -itemset is generated from merging two itemset on domain k . Candidate $(k + 1)$ -itemset containing the frequency of subset that rarely appears or below the threshold will be trimmed and not used to form association rules [2].

There is a relatively huge amount of research on apriori algorithms [7][8][9][10][11][12][13]. Studies related to the application of apriori algorithms that are used as references in this study are as follows:

1. The application of apriori algorithms that had been previously developed without using optimization techniques to obtain more efficient association rules [14].
2. Improved the apriori algorithm by determining "set size" and "set size frequency". Set size is the number of items per transaction while the set size frequency is the number of transactions that have at least "set size" items. This set size and set size frequency are used to eliminate insignificant key candidates [15].
3. Optimization of apriori algorithms by reducing or pruning the number of candidates for frequent itemset candidates on itemset C_k [16].
4. Improved the apriori algorithm by reducing the number of transactions (transaction reduction) whose number of items transaction did not meet the specified limit value. Reducing these transactions has an impact on efficiency improvement when scanning databases [17].
5. The utilization of apriori algorithms to establish customer segmentation in the SMES sector [18].
6. Application of apriori algorithms to form associations in sales database [19][20][21].

The essence of all research on optimization of apriori algorithms is limiting the frequent itemset candidates that are generated by bypassing unwanted transactions so that it does not overtake or repeat database scanning excessively; so that it will produce better and faster association rules.

Apriori algorithm has the disadvantage that it is less efficient on a larger database. Its performance will slow down because it has to do a large database scanning with a large number of transactions. Iteration is done repeatedly to get the frequent itemset combination in forming the right association rules. Modification techniques are needed to optimize the performance of apriori algorithms so as to get frequent itemset to form association rules in a short time [22][23][24][25][26][27][28][29]. Modifications in this study are obtained through combining combination reduction and iteration limitation techniques.

II. Method

A. Association Analysis

The association method is often used to analyse the contents of a consumer shopping cart in a transaction process [30][31]. The association method is also known as the market basket analysis. A simple example of an association method application is an analysis of a product purchased at a clothing store. The results will be obtained in the analysis, for example, the degree of possibility of consumers buying Trousers and clothes together. The application of the association method in the example can later help the shop owner to arrange the placement of goods and the inventory, or to make a promotion by giving special discounts for the combination of items that are often purchased.

Association analysis can be explained as a process to explore association rules that meet minimum support and minimum confidence requirements, where support and confidence are explained as follows:

1. Analysis of high frequency patterns, at this stage, is looking for item combinations that meet the minimum requirements of the support value in the database. The support value of an item is obtained by the following formula:

$$\text{Support}(A) = \frac{\text{Lots of Purchase Item } A}{\text{Total Purchase}} \quad (1)$$

The support value of 2 items is explained by the formula below:

$$\text{Support } A, B = P(A \cap B) \quad (2)$$

$$\text{Support } A, B = \frac{\sum \text{Purchase Items } A \text{ and } B}{\sum \text{Purchase}} \quad (3)$$

2. Formation of Association rules is sought after all high frequency patterns have been found: those which meet the minimum requirements for confidence by calculating the confidence value of associative rules $A \rightarrow B$.

The confidence value of $A \rightarrow B$ rules is obtained from the formula as following:

$$\text{Confidence} = P(B|A) = \frac{\sum \text{Purchase Items } A \text{ and } B}{\sum \text{Purchase Item } A} \quad (4)$$

The following is an example of clothing sales data. Each transaction data written as in the table 1. The sales data on Table 1 is translated into tabular forms 1-itemset as in the Table 2. The results of the translation will be used to form the next candidates $(k + 1)$ -itemset. A combination of 2-itemsets that might be obtained by pairing one product with another product from Table 2, then calculating the number of occurrences in each transaction by scanning the database. The result of the combination written as the Table 3.

Table 1. Sales table

Id	Date	Clothing Name
1	2017-08-01	Jacket, T-Shirt
2	2017-08-01	T-Shirt, Shirt, Trousers
3	2017-08-01	Shirt, Trousers
4	2017-08-01	Shirt, Shorts, Trousers
5	2017-08-01	Shirt, Trousers, Jacket, T-Shirt

Table 2. Description of transactions to form 1-itemset

No	Jacket	T-Shirt	Shirt	Trousers	Shorts
1	1	1	0	0	0
2	0	1	1	1	0
3	0	0	1	1	0
4	0	0	1	1	1
5	1	1	1	1	0
Total	2	3	4	4	1

Table 3 shown the prospective of 2-itemsets Candidates. If the threshold value (min_support) = 2 is obtained for the candidate on Table 3, frequent 2-itemset is as follows

F2 =
 Jacket, T-Shirt
 T-Shirt, Shirt
 T-Shirt, Trousers
 Shirt, Trousers

The frequent 3-itemsets candidates are formed in the same way. Similar method is used in pairing item one with other items to form a 3-itemset candidate as in the Table 3.

The threshold value (min_support) has been predetermined = 2. Therefore frequent 3-itemset from Table 4 is obtained as follows:

F3 = T-Shirt, Shirt, Trousers

If $(k + 1)$ -itemset that can be formed no longer exists, the support and confidence value for each frequent itemset combination is calculated. Association rules are formed based on selected frequent $(k + 1)$ -itemset.

The selected association rules are a rule that has a confidence value greater than or equal to the min_confidence value. The min_confidence value is 80%. The following Table 7 forms the association rules on Table 5 and Table 6.

The Rules of Final Association shown on Table 7 aims to choose the most suitable rules as a guide to improve decision making and marketing strategies. This stage produces output in the form of frequent itemset or rule with the highest multiplication of support and confidence value. This stage is the final conclusion of the apriori process which later explains that association rules with the strongest influence are rules that have the highest multiplication of support and confidence values.

Table 3. Prospective of 2-itemset candidates

Combination		Number
Jacket	T-Shirt	2
Jacket	Shirt	1
Jacket	Trousers	1
Jacket	Shorts	0
T-Shirt	Shirt	2
T-Shirt	Trousers	2
T-Shirt	Shorts	0
Shirt	Trousers	4
Shirt	Shorts	1
Trousers	Shorts	1

Table 4. Prospective of 3-itemset candidate

Combination			Number
Jacket	T-Shirt	Shirt	1
Jacket	T-Shirt	Trousers	1
Jacket	T-Shirt	Shorts	0
Jacket	Shirt	Trousers	1
Jacket	Shirt	Shorts	0
Jacket	Trousers	Shorts	0
T-Shirt	Shirt	Trousers	2
T-Shirt	Shirt	Shorts	0
T-Shirt	Trousers	Shorts	0
Shirt	Trousers	Short	1

Table 5. Prospective association rules of F2

IF Antecedent, then Consequent	Support	Confidence
If Jacket, then T-Shirt	2 / 5 = 40%	2 / 2 = 100%
If T-Shirt, then Jacket	2 / 5 = 40%	2 / 3 = 66.7%
If T-Shirt, then Shirt	2 / 5 = 40%	2 / 3 = 66.7%
If Shirt, then T-Shirt	2 / 5 = 40%	2 / 4 = 50%
If T-Shirt, then Trousers	2 / 5 = 40%	2 / 3 = 66.7%
If Trousers, then T-Shirt	2 / 5 = 40%	2 / 4 = 50%
If Shirt, then Trousers	4 / 5 = 80%	4 / 4 = 100%
If Trousers, then Shirt	4 / 5 = 80%	4 / 4 = 100%

Table 6. Candidate rules association of F3

IF Antecedent, then Consequent	Support	Confidence
If T-shirt and shirts, then Trousers	2/5 = 40%	2/2 = 100%
If T-Shirt and Trousers, then Shirt 2/5	= 40%	2/2 = 100%
If Shirt and Trousers, then T-Shirt 2/5	= 40%	2/4 = 50%

Table 7. Rules of Final Association

IF Antecedent, then Consequent	Support	Confidence	Support * Confidence
if buy Jacket, then buy T-Shirt	2 / 5 = 40%	2 / 2 = 100%	0.4
if buy Shirt, then buy Trousers	4 / 5 = 80%	4 / 4 = 100%	0.8
if buy Trousers, then buy Shirt	4 / 5 = 80%	4 / 4 = 100%	0.8
if buy T-Shirt and Shirt, then buy Trousers	2 / 5 = 40%	2 / 2 = 100%	0.4
if buy T-Shirt and Trousers, then buy Shirt	2 / 5 = 40%	2 / 2 = 100%	0.4

Apriori algorithm uses all items in the database transaction every time the process of the scanning database generates combinations. It is very timely inefficient, because the items that rarely appear are still used in forming combinations. Figure 1 shows the flowchart of apriori algorithm.

The explanation of the flowchart on Figure 1 can be described as follows:

1. Determining the minimum support and minimum confidence value using approximate values by trial and error. In this research, this has been determined for minimum support = 2 and minimum confidence = 80%
2. Apriori Algorithm using the iterative approach for k-itemset is generated to form the next (k + 1) -itemset.
3. (k + 1) -itemset candidates which are formed with frequencies that rarely appear in the database or below the threshold (min_support) will be eliminated and not used in determining association rules.
4. 1-itemset is formed by scanning a database and then the number of occurrences of each item on each transaction is counted.
5. Furthermore, the itemset is used to form 2-itemsets. Candidates for 2-itemset are formed by pairing one item with another item so that it forms a 2-itemset combination.
6. The value of 2-itemsets that have been formed is then calculated for its appearance on every transaction. The threshold (min_support) value is determined to eliminate candidates that are not frequent.
7. The support and confidence values of the 2-itemset that qualify are then calculated. 2-itemset whose support and confidence values are above or equal to min_support and min_confidence will be used to form association rules.
8. Then iteration is repeated by using formed 2-itemset to find 3-itemset and so on until there is no more frequent (k + 1)-items left.
9. After all association rules from frequent (k + 1) -itemset are formed, then the values of support and confidence are calculated. Multiplication results from the highest support and confidence values are the best association rules of all transactions in the database.

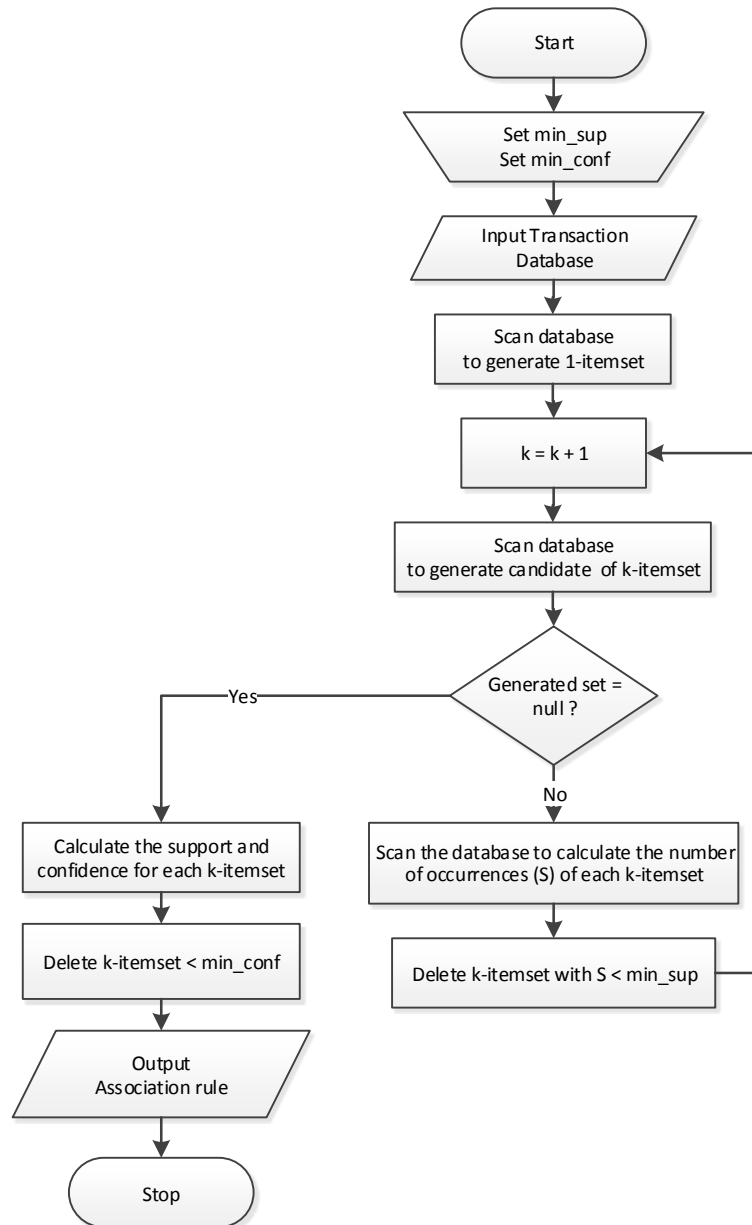


Fig. 1. Flowchart of Apriori Algorithm [11]

Apriori algorithm has the disadvantage that it is less efficient on a larger database. The apriori algorithm performance will slow down because it has to perform an extensive database scanning with a large number of transactions and repeated iterations to get the combination of frequent itemset so that it forms the right association rules. These weaknesses can be overcome by applying modification techniques on the formation of candidates of the frequent itemset combination.

B. Combination Reduction

The modified algorithm in this study employs methods of reduction combination or different generated reduction combinations. Combination reduction handles frequent itemset or a combination of the results of the previous scanning database to form the next itemset candidate. The generated combination then contains frequent itemset from the results of previous scanning database. The combination formed by this method is certainly fewer than combinations that are formed by apriori method without modification and have more opportunities to become frequent itemset because the combination used to form the next itemset is a frequent itemset. Apriori method without modification consumes more time because of repeated scanning to generate all combinations without regard to the previous frequent itemset.

1) Specifying Items that are used to Generate Combinations (1-Items)

Finding 1-itemset has to be completed before generating a possible combination that appears. The 1-itemsets must qualify the minimum support emergence that will be used to form combinations in the search for frequent itemset. 1-itemset is searched by scanning the database and accumulating the number of occurrences of each item in all transactions. Items whose occurrence values are less than the minimum support are not used in determining the combination of $(k + 1)$ -itemset while items that are qualified will be used as a combination pair in forming the next itemsets.

2) Generating Itemset Combinations based on Previous Frequent Itemset

After obtaining frequent itemset from 2-itemset resulting from the initiate database scanning, then the combination 3-itemset candidate is generated by simply pairing frequent itemset from 2-itemset with other items that meet the minimum support. 3-combination itemset candidates that do not contain frequent itemset from 2-itemset and items unqualified for the minimum support do not need to be generated. This will result in considerable time saving, low computing and avoiding the memory allocation to run out.

For example; shorts Item in the Table 8 above are removed. This is because the occurrence values less than minimum support = 2. After going through the process of forming a combination with apriori algorithm, frequent 2-itemset is obtained from the scanning database, namely:

F2 =

1. Jacket, T-Shirt
2. T-Shirt, Shirt
3. T-Shirt, Trousers
4. Shirt, Trousers

Results from frequent 2-itemset then is used to make 3-itemset candidate. Iteration is obtained through similar previous method by pairing just a combination that includes frequent 2-itemset only with one other item that meets the minimum support. The 3-itemset candidates are obtained as in the Table 9 which illustrates that combinations are generated only for those that contain frequent itemset of 2-itemset paired with other items with minimal support qualification. Consequently, the unqualified items have been removed. This combination reduction will reduce computation in forming combinations so that it saves time and accelerates the apriori algorithm to find association rules.

Iteration in apriori algorithms is not limited until all combinations of generated itemset appear in transaction data; in which in this case are as many as the number of items contained in the transaction. The application of iteration limitation here is to limit the repetition of the scanning database in generating a combination of $(k + 1)$ -itemset. It is obtained by using the mode formula to find out how many items are often purchased in one transaction that often occurs.

Table 8. Items that Meet the Minimum Support

No	Jacket	T-Shirt	Shirt	Trousers
1	1	1	0	0
2	0	1	1	1
3	0	0	1	1
4	0	0	1	1
5	1	1	1	1
Number	2	3	4	4

Table 9. Prospective 3-Itemset Candidates

Combination			Amount
Jackets	T-Shirt	Shirts	1
Jackets	T-shirts	Trousers	1
Jackets	Shirts	Trousers	1
T-Shirt	Shirt	Trousers	2

Table 10. Transaction Data with Set Size for Iteration Limitation

TId	Transaction	Name Item Name	Set Size
1	2013-06-10	Jacket, T-Shirt	2
2	2013-06-10	T-Shirt, Shirt, Trousers	3
3	2013-06-10	Shirt, Trousers	2
4	2013-06-10	Shirt, Shorts, T-Shirt, Trousers	4
5	2013-06-10	Shirt, Trousers, Jacket, T-Shirt	4

In the example, 100 transaction samples and 25 items that meet the minimum support are exercised, and the number of transactions that often appear are as many as 2 items in the transaction. Most consumers buy 2 items in one transaction. This can be used as an iterative delimiter where up to 2 frequent itemset in accordance with the habits are often done by consumers and this process faster and more efficient.

Based on the transaction data on Table 10, the set size of transactions that often appear is set size = 2 and set size = 4, then the value used is the largest value: $k = 4$ because the possibility of getting the best association rules becomes greater. Iteration to search $(k + 1)$ -itemset with apriori algorithm will be halted until no more frequent itemset and the maximal limit of iteration is $k \leq 4$.

III. Results and Discussions

This study was conducted to determine the results of the comparison of apriori algorithm without modification and with a modified apriori algorithm. Modifications of apriori algorithm are expected to be faster in generating association rules so that they are more time-efficient.

Ratio results are measured in terms of required time between apriori algorithms without modification with the apriori algorithm that have been modified. Both algorithms are exercised with several database samples with the number of transactions that continue to grow and each experiment's required time is calculated until it establishes the association rule. The algorithm's required time is obtained from the algorithm's expiry time calculation. This aims to obtain the algorithm of time-reduced association rules which is executed in accordance with the formula as follows:

$$\text{\$reqtime} = \text{\$t_starttime} - \text{\$t_endtime} \quad (5)$$

The results of the required time comparison of the algorithm can be recorded in Table 11 which shows comparison on data sample 400 and 500 in Apriori without modification is failed because the database server is error (time out). The memory bandwidth cannot accommodate the large iteration of data. The result graphs of measurement in terms of time and number of transactions which shown in Figure 2:

Graph of research time comparison on Figure 2 shows that the apriori algorithm that has been modified is more time efficient in order to obtain association rules. The horizontal data show the number of transactions while the vertical data shows the required time to get the association rules. The red lines represent the development of the results of apriori with modification, while the blue lines represent the development of the results of the apriori without modification. Apriori without modification represented by a blue line shows a sharp increase, meaning that the more data increases, the higher the computation in the combination formation process and the more time needed to obtain

Table 11. Comparison of apriori algorithm time with multiple sample transaction

No.	Sample Transaction	Required time (in microseconds)	
		Apriori	Apriori Modification
1	100	6.16	0.81
2	200	144.90	5.16
3	300	942.87	18.08
4	400	Failed	36.71
5	500	Failed	75.64



Fig. 2. Time comparison chart of apriori algorithms

frequent itemset. The red line shows an increase that is not too sharp and tends to be flat, meaning that even though the transaction data continues to increase the required time is proportional to the increase in transaction data.

The results of several trials with several transaction samples show that the quality of association rules obtained by apriori modification algorithms is no different from unmodified algorithms. The association rules obtained from the apriori algorithm without modification with the modified association algorithm are the same in several attempts. The results of the experiment show that there is no quality degradation from the established association rules.

IV. Conclusion

Apriori algorithm is suitable to be applied to transactions in large database to find frequent itemset. Association rules that result from frequent itemset can then be used for improving decisions in organizing item displays, arranging inventory or promotion strategies with the example of applying discounts for combination items that often appear in transactions according to the established association rules. Apriori performance that slows down in larger databases can be optimized by using modification method. Apriori algorithm that has been modified with combination reduction and iteration limitation techniques has proven to be more efficient in terms of time than the performance of unmodified algorithms in generating association rules. The quality of the resulting rules is also unchanged, in other words the results obtained are similar between the apriori algorithm without modification and the modified apriori algorithm.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

- [1] U. Fayyad, G. P. Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI Mag.*, vol. 17, no. 3, pp. 37–54, 1996.
- [2] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. United States of America: Pearson Addison-Wesley, 2006.
- [3] J. Pamungkas and Y. Handrianto, "Association Rules for Product Sales Data Analysis Using The Apriori Algorithm," *Sink. Junal Penelit. Tek. Inform.*, vol. 5, no. 1, p. 84, 2020.

- [4] R. Agrawal, "Mining Association Rules between Sets of Items in Large Databases," in *Proceeding of the 1993 ACM SIGMOD Conference Washington DC, USA, 1993*, pp. 1–10.
- [5] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *Proceeding 20th VLDB Conf. Santiago, Chile., 1994*.
- [6] J. Han and M. Kamber, *Data Mining: Concepts and Techniques Second Edition*. United States of America: Elsevier Inc., 2006.
- [7] L. F. Panjaitan, Y. Handrianto, and A. Nurhadi, "Apriori Algorithm On Car Rental Analysis With The Most Popular Brands," *Sink. Junal Penelit. Tek. Inform.*, vol. 4, no. 2, p. 47, 2020.
- [8] E. Irfiani, "Application of Apriori Algorithms to Determine Associations in Outdoor Sports Equipment Stores," *Sink. Junal Penelit. Tek. Inform.*, vol. 3, no. 2, p. 218, 2019.
- [9] G. Danon, M. Schneider, M. Last, M. Litvak, and A. Kandel, "An Apriori-like algorithm for Extracting Fuzzy Association Rules between Keyphrases in Text Documents," *Cs.Bgu.Ac.II*, 2006.
- [10] Luthfiah and K. Ditha Tania, "K-Means and apriori algorithm for pharmaceutical care medicine (case study: Eye hospital of South Sumatera Province)," in *Journal of Physics: Conference Series*, 2019, pp. 1–7.
- [11] A. Ezhilvathani and K. Raja, "Implementation of Parallel Apriori Algorithm on Hadoop Cluster," *Int. J. Comput. Sci. Mob. Comput.*, vol. 2, no. 4, pp. 513–516, 2013.
- [12] N. A. Harun, M. Makhtar, A. A. Aziz, Z. A. Zakaria, F. S. Abdullah, and J. A. Jusoh, "The application of Apriori algorithm in predicting flood areas," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 3, pp. 763–769, 2017.
- [13] N. Badal and S. Tripathi, "Frequent Data Itemset Mining Using VS _ Apriori Algorithms," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 4, pp. 1111–1118, 2010.
- [14] J. Suresh and T. Ramanjaneyulu, "Mining Frequent Itemsets Using Apriori Algorithm," *Int. J. Comput. Trends Technol.*, vol. 4, no. 4, pp. 760–764, 2013.
- [15] S. A. Abaya, "Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation," *Int. J. Sci. Eng. Res.*, vol. 3, no. 7, pp. 1–4, 2012.
- [16] J. Yabing, "Research of an Improved Apriori Algorithm in Data Mining Association Rules," *Int. J. Comput. Commun. Eng.*, vol. 2, no. 1, pp. 25–27, 2013.
- [17] J. Singh, H. Ram, and J. S. Sodhi, "Improving Efficiency of Apriori Algorithm Using Transaction Reduction," *Int. J. Sci. Res. Publ.*, vol. 3, no. 1, pp. 1–4, 2013.
- [18] J. Silva, N. Varela, L. A. B. López, and R. H. R. Millán, "Association rules extraction for customer segmentation in the SMES sector using the apriori algorithm," in *Procedia Computer Science*, 2019, pp. 1207–1212.
- [19] A. W. O. Gama, I. K. G. D. Putra, and I. P. A. Bayupati, "Implementasi Algoritma Apriori untuk Menemukan Frequent Itemset dalam Keranjang Belanja," *Teknologi Elektro*, vol. 15, no. 2, pp. 27–32, 2016.
- [20] K. K. Widiartha, D. Putu, and D. Kumala, "Shopping Cart Analysis System in Product Layout Management with Apriori Algorithm," *Int. J. Appl. Comput. Sci. Inform. Eng.*, vol. 1, no. 2, pp. 53–64, 2019.
- [21] K. S. Raju, A. D. Devi, and D. D. D. Suribabu, "Mining Frequent Item Sets Using Apriori Algorithm on Shopping Dataset," *Mukth Shabd J.*, vol. 9, no. 5, pp. 6309–6320, 2020.
- [22] B. Patel, V. K. Chaudhari, R. K. Karan, and Y. . Rana, "Optimization of Association Rule Mining Apriori Algorithm Using ACO," *Int. J. Soft Comput. Eng.*, vol. 1, no. 1, pp. 24–26, 2011.
- [23] M. F. Akas, A. G. M. Zaman, and A. Khan, "Combined item sets generation using modified apriori algorithm," in *ACM International Conference Proceeding Series*, 2020, pp. 4–6.
- [24] H. Yu, J. Wen, H. Wang, and J. Li, "An improved Apriori algorithm based on the Boolean matrix and Hadoop," *Procedia Eng.*, vol. 15, pp. 1827–1831, 2011.
- [25] Z. Jie and W. Gang, "Intelligence Data Mining Based on Improved Apriori Algorithm," *J. Comput.*, vol. 14, no. 1, pp. 52–62, 2019.
- [26] X. Liu, Y. Zhao, and M. Sun, "An Improved Apriori Algorithm Based on an Evolution-Communication Tissue-Like P System with Promoters and Inhibitors," *Discret. Dyn. Nat. Soc.*, vol. 2017, 2017.
- [27] R. Sun and Y. Li, "Applying Prefixed-Itemset and Compression Matrix to Optimize the MapReduce-based Apriori Algorithm on Hadoop," in *ACM International Conference Proceeding Series*, 2020, pp. 89–93.
- [28] X. Yuan, "An improved Apriori algorithm for mining association rules," in *AIP Conference Proceedings*, 2017, pp. 1–6.
- [29] D. T. Larose, *An introduction to data mining*, vol. 134. Canada: John Wiley & Sons, Inc, 2005.
- [30] Y. Kurnia, Y. Isharianto, Y. C. Giap, A. Hermawan, and Riki, "Study of application of data mining market basket analysis for knowing sales pattern (association of items) at the O! Fish restaurant using apriori algorithm," in *Journal of Physics: Conference Series*, 2019, pp. 1–6.
- [31] J. R. Delos Arcos and A. A. Hernandez, "Analyzing online transaction data using association rule mining: Misumi philippines market basket analysis," in *ACM International Conference Proceeding Series*, 2019, pp. 45–49.