International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Artificial intelligence based Data Pruning using Regression for Smart Apriori Algorithm

Kamalesh Kumar
Computer Science & Engg. Dept
BBDITM, Lucknow, India
leekampat@gmail.com

Shekhar Srivastava
Computer Science & Engg. Dept
BBDITM, Lucknow, India
shekharsri@gmail.com

Vineet Agarwal
Computer Science & Engg. Dept
BBDITM, Lucknow, India
avineet242@gmail.com

Abstract: Artificial Intelligence commonly referred to as Knowledge Discovery in databases (KDD), is a significant area of study at the moment. Frequent pattern finding is a crucial Artificial Intelligence approach. This method focuses on identifying co-occurrence links between elements. Association rule mining is a current area of research for KDD, and numerous algorithms have been created in this area. The item-set associations are found using this approach. Efficiency has long been a source of debate about mining association regulations. Apriori is founded on the idea of mining different datasets for insightful patterns. Despite being a tried and true method, it nonetheless has several flaws. Due to the frequent development of candidate item-sets that are not required, it suffers from a lack of superfluous database scans when looking for frequent item-sets. Additionally, redundant sub-item sets are formed, and the programme searches the database again. This work makes a reduction in the production of sets that are redundant. The entire execution time is shortened as a result. Additionally, fewer transactions need to be scanned.

Key Words: Apriori, Artificial Intelligence, Pruning, Regression

1. Introduction

Rule of Association It is one of the methods used in Artificial Intelligence. The objective is to identify the products that are frequently bought together so that they can be positioned on the store's shelves properly. Cross-selling can also make use of this information. It is made up of if/then statements that are designed to uncover connections among data that might otherwise appear unconnected and is kept in warehouses or other repositories. A person is most likely to purchase insurance for a new car, for instance. Data is analysed to identify recurring patterns so that association rules can be formed. Association rules are used to forecast client behaviour. For market basket analysis, these are utilised. Web usage mining, intrusion detection, continuous production, and bioinformatics are more areas where association rules are used. The algorithms used in association rule mining are numerous. The age-old Apriori algorithm is employed in rule mining. The most popular approach for extracting frequent sets and locating association rules in transactional databases is the apriori algorithm. As long as there are existing item-sets in the database, it starts by identifying the most common single items before combining those things to create larger item-sets. As a result, it is known as a bottom-up method. In order to extract the association rules from a sizable database, frequent sets are generated. The primary goal of the Artificial Intelligence process is to extract information from a dataset and then transform it into a form that can be understood and utilised again.

2. Relevant Work

Parvinder The association rules were established by S. Sandhu et al using weight and utility attributes. They talked about how the traditional Apriori algorithm does not take into account the quantitative data related to the qualities, resulting in rules that are common but have less weight. Some of the traits that occur less frequently might really occur more frequently. These qualities could be significant from a business perspective. The association rules are amended to include the weight gains (W) and utility gains (U) for this purpose [1]. The characteristics' profit margins are increased by utility improvements. They created rules using the Apriori algorithm. Then, using W-gain and U-gain, UW score is computed for each association rule. The rules with a UW score higher than the predetermined threshold are chosen.

An method that makes use of the dataset comprising transactions in a transposed form was proposed by Gurudutta Verma et al. The dataset is transposed with the use of the parallel transposition method [2]. Diagonal elements in the dataset are unaffected by the transposition. The spots below the diagonal that are symmetrical are occupied by those above the diagonal, and vice versa. Lk-1 is then used to construct the candidate sets, after which pruning is carried out to identify the large item-sets. The experimental findings demonstrate how quick and effective the suggested algorithm is at identifying association rules. The I/O cost is reduced, and the support value is retrieved more quickly than with the Apriori technique.

Due to issues with the Apriori method, strong association rules' effectiveness is constrained. Some rules obtained have no use because there are only two threshold values: support and confidence. In order to optimise the Apriori algorithm, a third threshold value known as the interest measure is included [3]. Support and confidence served as the foundation for the interest measure's definition. There is also a minimum interest measure value, sometimes known as a threshold value.

According to the suggested methodology, the association rule is only interesting if the interest measure is larger than or equal to the minimum interest value; otherwise, it is meaningless. According to the results of the experiments, needless rules or errors were eliminated when the interest measure was added. As a result, the algorithm is improved. They presented a paradigm in [4] that calls for splitting the task into two phases. By fixing issues like duplicate data or faulty data in the first phase, the consistency of the database is improved. The second part involves identifying the common data patterns. This method has the benefit of not scanning the entire database for common data patterns; instead, the patterns are looked for.

A better approach that involves only twice scanning the full database is explained in [3]. There are two steps in this strategy: The database is separated into several sections first. The frequent sets of each division are recorded once the database has been examined once. The second scan determines each set's support count and, in the end, identifies globally frequent item-sets. Second, when the dataset is divided, the potential itemsets are included. The candidate sets are included at the beginning so that they can be chosen prior to database scanning. This method is effective since it saves storage space and cuts down on the number of candidate item-sets. In [7], they suggest a new, improved Apriori method that makes use of a database that has been compressed by removing transactions that don't contain items the user isn't interested in. This decreases the size of the database, which in turn shortens the process's time because fewer scans are required. When features need to be classified and chosen, this method is helpful.

3. Proposed Approach

The Apriori algorithm is the foundation of the suggested methodology. The major goal is to enhance the Apriori algorithm, which is accomplished by using various data trimming techniques. Calculating the support value of each item does not need scanning the original dataset each time. Regression method is utilised throughout the scanning.

A type of regression analysis called linear regression models data using a least squares function, which depends on one or more independent variables and is a linear combination of the model's parameters. The model function in simple linear regression is a straight line. The outcomes of data fitting are statistically analysed. In a model for linear regression, the dependent variable, 0E, is defined as a linear combination of the parameters (but need not be linear in the independent variables). For instance, there is only one independent variable in simple linear regression when modelling N data points: Two parameters, 0 and 1, are used.

On to a straight line regression:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad (i = 1, \dots, N).$$

More general linear regression models represent the relationship between a continuous response y and a continuous or categorical predictor x in the form:

$$y = \beta_1 f_1(x) + \beta_2 f_2(x) + \ \dots + \beta_p f_p(x) + \varepsilon.$$

The response is modeled as a linear combination of (not necessarily linear) functions of the predictor, plus a random error ε. The expressions $f_j(x)\ (j = 1, \dots, p)$ are the terms of

the model. The $\beta_j\ (j = 1, \dots, p)$ are the unknown coefficients. Errors ε are assumed to be uncorrelated and distributed with mean 0 and constant (but unknown) variance.

Given n independent observations $(x_1, y_1), \dots, (x_n, y_n)$ of the predictor x and the response y, the linear regression model becomes an nxp system of equations:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} f_1(x_1) & \cdots & f_p(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_n) & \cdots & f_p(x_n) \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix},$$

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix},$$

Where,

$$X = \begin{pmatrix} f_1(x_1) & \cdots & f_p(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_n) & \cdots & f_p(x_n) \end{pmatrix},$$

$$\beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix},$$

$$\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}.$$

Here, X is the *design matrix* of the system. The columns of X are the terms of the modelevaluated at the predictors. To fit the model to the data, the system must be solved for the p coefficient values in $\beta^T = (\beta_1, \dots, \beta_p)$.

The function uses a reliable fitting technique that is sensitive to significant changes in very tiny portions of the data. Each data point is given a weight in order for this to work. Iteratively reweighted least squares is a method for automatically and repeatedly weighting data. Each point is given equal weight in the initial iteration, and the model coefficients are calculated using ordinary least squares. Weights are recalculated in succeeding iterations so that points furthest from model predictions in the preceding iteration are assigned a lower weight. Then, weighted least squares are used to recompute the model coefficients. The procedure is repeated until the coefficient estimates' values converge within a given tolerance. The support value is determined by adding up the big data set that is created by intersecting each item's transactions. As a result, more time is saved and the Apriori algorithm is more effective. The threshold value is altered at each phase. It is determined by summing the support values of each item in a candidate item set and dividing the result by the sum of all the item's unique transactions.

Algorithm Used

Step-1 : Find all frequent itemsets
Step-2 : SA Get frequent items
Step-3 : Items whose occurrence in database is greater than or equal to the min.support threshold.
Step -4: Get frequent itemsets
Step -5: Generate candidates from frequent items.
Step -6: Prune the results to find the frequent itemsets using Linear Regression.
Step- 7: Goto Step-4 till outliers are removed.
Step -8: Generate strong association rules from frequent itemsets
Step-9: A Rules which satisfy the min.support and min.confidence threshold.

4. Results and Analysis

The outcomes of the suggested strategy are shown in this section, followed by a comparison of the modified algorithm with the original. The fundamental goal of the suggested methodology has been to do away with the shortcomings of the traditional Apriori algorithm. The following two parameters have been worked on in order to do this:

- A decrease in the quantity of transactions that need to be scanned
- Shortening of the execution time

The Number of Transactions Has Decreased

Regression approach used in conjunction with pruning helps to reduce the amount of transactions that need to be scanned. Each data point is given a weight in order for this to work. Iteratively reweighted least squares is a method for automatically and repeatedly weighting data. Each point is given equal weight in the initial iteration, and the model coefficients are calculated using ordinary least squares. Weights are recalculated in succeeding iterations so that points furthest from model predictions in the preceding iteration are assigned a lower weight. Then, weighted least squares are used to recompute the model coefficients. The support value is given by this intersection. In terms of the quantity of transactions, the improved method is compared to the Apriori algorithm. The total number of transactions to be scanned in Apriori is equal to the product of the total number of transactions and the number of passes, however in improved Apriori, transactions are decreased at each pass. It is equal to the transactions in the initial dataset only in the first iteration. It decreases in the next passages. The results of testing both methods on datasets of various sizes are shown in table 1 and graphically illustrated in figure.

Table 1 Comparison of Number of Transactions

	Apriori	Improved	Reduction (%)
500	250	151	39.6
1000	500	330	34
2000	1000	602	39.8
5000	2500	1578	36.88

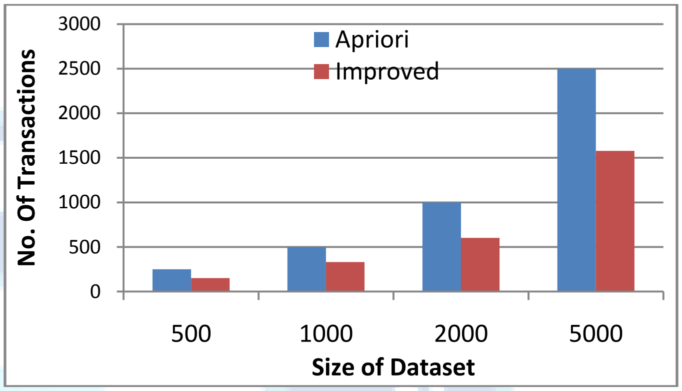

Figure 1: Comparison of Number of Transactions

Reduction in the Time of Execution

Two steps were taken to shorten the execution time: first, the support value was updated at each pass; and second, as was already mentioned, the number of transactions was decreased. The table below lists the outcomes of the shorter execution time and the efficiency attained. The time is displayed in seconds. These outcomes come from testing the Apriori method and the enhanced approach on datasets of various sizes.

Table 2 Comparison of Execution Time in Milliseconds

	Apriori	Improved	Efficiency (%)
500	228	150	34.21
1000	352	227	35.51
2000	388	232	40.21
5000	754	459	39.12

The graphical representation of the faster execution time than the Apriori algorithm is shown in Figure 2. The chart shows that the enhanced algorithm performs better than Apriori in terms of execution time.

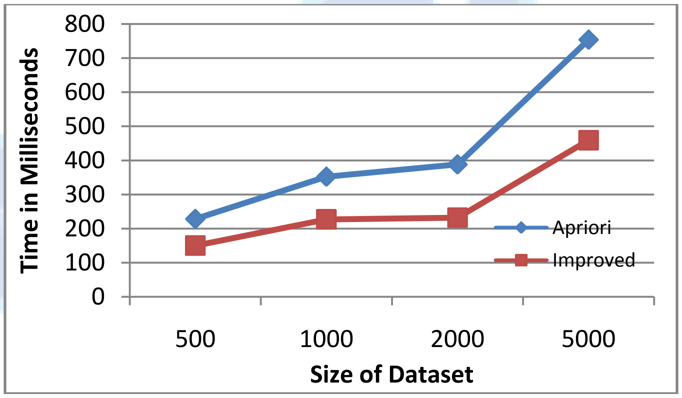

Figure 2: Comparison of Time of Execution (in milliseconds)

5. Conclusion

We conclude that the enhanced Apriori algorithm is a useful algorithm for reducing the number of transactions after putting the suggested technique into practise. To increase efficiency, the work is now done on transactions rather than objects, and to do this, the dataset is pruned by regression. This shortens the time needed to scan the dataset, which further shortens the total amount of time. At each pass, the minimal support value is also determined, removing any generated sets that are not necessary. Despite being straightforward, the algorithm does more precise trimming. On a single processor, the modified Apriori algorithm produces acceptable results. The application of the enhanced Apriori algorithm to the data dispersed among parallel processors may be included in future study. In that circumstance, different outcomes are anticipated. Using this method, we also want to see if the number of database scans per CPU decreases.

References

[1]. P. Sandhu, D. Dhaliwal, S. Panda, A. Bisht, “An Improvement in Apriori algorithm Using Profit And Quantity”, Second International Conference on Computer and Network Technology, 2010, pp. 3-7.

[2]. Y. Jia, G. Xia, H. Fan, Q. Zhang, X. Li, “An Improved Apriori Algorithm Based on Association Analysis”, Networking and Distributed Computing (ICNDC), Third International Conference, IEEE, 2012, pp. 208-211.

[3]. J. Hu, “The Analysis on Apriori Algorithm Based on Interest Measure” ,proceedings of the International Conference on Control Engineering and Communication Technology, IEEE Computer Society, 2012, pp. 1010-1012.

[4]. C.Yadav et. al, “An Approach to Improve Apriori Algorithm Based On Association rule Mining”, 4th ICCCNT ,2013, pp. 1-9.

[5]. D. Ping, G. Yongping, “A New Improvement of Apriori Algorithm for Mining Association Rules”, International Conference on Computer Application and System Modelling (ICCASM), 2010, pp. V2-529.

[6]. Y. Zhou, W. Wan, J. Liu, L. Cai, “Mining Association Rules Based on an Improved Apriori Algorithm”, International Journal of Computer Applications Volume 112 – No 4, February 2010, pp. 414-418.

[7]. V. Vaithiyanathan, K. Rajeswari, R. Phalnikar3 , S. Tonge, “Improved Apriori algorithm based on Selection Criterion”, Computational Intelligence & Computing Research (ICCIC), IEEE International Conference ,2012, pp. 1-4.

[8]. Y. Liu et. al, “Application and improvement discussion about Apriori algorithm of association rules mining in cases mining of influenza treated by contemporary famous old Chinese medicine”, Bioinformatics and Biomedicine Workshops (BIBMW), IEEE International Conference, 2012, pp. 316-322.

[9]. A. Srivastava ,et. al. " An Approach for Personalization on Web Based Systems", International Journal of Research and Development in Applied Science and Engineering, Volume 5, Issue 1, March 2014.

[10] Jiawei Han • Hong Cheng • Dong Xin • Xifeng Yan “Frequent pattern mining: current status and future Directions” Data Min Knowl Disc (2007) 15:55–86 DOI 10.1007/s10618-006-0059-1.

[11] K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cube.SIGMOD Rec., 28:359–370, 1999.

[12] Mehdi Adda, Lei Wu, Sharon White(2012), Yi Feng ” pattern detection with rare item-set mining” International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.1, No.1, August 2012.

[13] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In VLDB, pages 487–499, 1994.

[14] R. Agrawal, T. Imieinski, and A. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, Proceedings of the ACM SIGMOD International Conference on the Management of Data, pages 207– 216, Washington DC, 1993. ACM Press.

[15] W. Shi, F.K. Ngok, and D.R. Zusman. Cell density regulates cellular reversal frequency in Myxococcus xanthus. In Proceedings of the National Academy of Sciences of the United States of America, volume 93(9), pages 4142–4146,1996.

[16] X. Dong, Z. Niu, X. Shi, X. Zhang, and D. Zhu. Mining both positive and negative association rules from frequent and infrequent itemsets. In ADMA, pages 122–133,2007.

[17] X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. ACM Transactions on Information Systems, 22(3):381–405, 2004.