ENHANCED SLAP ALGORITHM TO FIND TOP FREQUENT ITEMSETS IN THE ONLINE RETAIL STORE

[a]V. Ravi kumar, [b]Dr. V. Purnachandra Rao.

[a]Associate Professor of Information Technology, TKREC, Hyderabad, Telangana, India
[b] Professor,SVIT, Hyderabad, Telangana, India.

[a]ravi.vanoj@gmail.com
[b]pcrao.vemuri@gmail.com

Abstract: Online shopping plays an important role from the past few decades. E-commerce websites to increase their profits by performing business analytics on the top utility items. Existing transactional databases implemented sequential patterns to find top utilities. More number of resources and computations are required to access these databases. In the proposed research, the model recognizes the frequent patterns by extending the Genetic Approach known as "Slap", a swarm optimization which finds the best solution for the frequent activities. Optimization technique in this research is associated with finding the top-k utilities with maximum accuracy. The major advantage of this approach lies in its colony formation in perfect shape by communicating with neighbours. The elements in the local search space continuously communicate with their agents and update their positions in global space. The implementation of genetic approaches with multi objective function helps the model to reduce the features and improve the accuracy. The proposed approach has obtained 97.1% accuracy.

Keywords: Transaction Databases, Sequential Mining, Salp Algorithm, Swarm Intelligence, Accuracy

1. Introduction

Eclat, which searches the subset tree in-depth first. The effectiveness of processing memory consumption and time use between the old methods and the suggested algorithm was the authors' main concern. The itemset's occurrences would be among the aspects affecting the algorithm's performance. A set of members of the class for a sub tree rooted at P frequent items formed by computing diffset for all unique pairings of itemsets and evaluating the support of the resulting itemsets serves as the input to the technique. With those item sets that are discovered to be common at the current level, a recursive process call is made. Until all frequent item sets have been listed, this process is repeated. Finding recurring patterns in huge databases is a crucial task.

The likelihood of purchasing subordinate products if a prime item is secured is predicted using two association rules data mining methods, namely Apriori and Eclat. The Eclat algorithm is quicker than the Apriori method because of its vertical approach. This algorithm's main goal is to calculate the support of the proposed item set using set intersection rather than creating subsets that don't already exist in the tree data structure. By altering the support value, the Eclat approach is run on the dataset. The scanning rate of Eclat is high. It is a clean set because the span of support and confidence lies between 0 and 1. To test the efficacy of the algorithms, reinforcement learning techniques like Thompson sampling and upper confidence limit can be used. Classification algorithms can be added to association rule algorithms to improve them.

In Eclat variations that employ tidset format in the initial looping and diffset in the subsequent looping. The volume of tidsets is among the key elements determining Eclat's operating time and memory consumption usage as intersecting tidsets is its primary function. Additional time and storage are required for larger datasets. Diffset has been demonstrated to significantly outperform traditional Eclat (tidset) in terms of performance and memory utilization, particularly in large databases. Diffset lacks its strengths over tidsets in sparse databases. Although the proposed technique performs moderately when it comes to time execution, it displays better patterns than Eclat-tidset. The algorithm performance would be influenced by the likelihood of itemset repetitions in each database.

2. Literature Review

In [1], Md. Rezaul Karim et al provided a method for Apache Spark-based MFP mining. Within the Spark framework, a paradigm is presented for converting a conventional single pipeline to an inter-cluster pipeline. To generate transaction value databases, two new algorithms named TV and iTV are developed. These algorithms use prime number-based data processing techniques in both a static and incremental environment. To keep the partly downward closure property and to decrease the state space and the set of candidate frequent itemsets, two efficient pruning approaches have been created. Maximal Frequent Pattern with Apache Spark is the name of the suggested algorithm (MFPAS). For the past few years, the Hadoop MapReduce architecture has been extensively employed to address some of these issues. SPARK takes a careless approach. Decentralized analytic jobs are managed using the SOMA platform6, which also supports the big data processing conditions needed for job execution.

In [2], Dingming Wu et al offer a solution based on the FP*-tree, a specially built index structure, and the corresponding bound-based algorithm. To rapidly determine the top-k maximum diverse frequent itemsets, the authors merged the maximality and variety requirements. The depth of the class tree is proportional to the width of each code. Each component of an item's code correlates to a cluster on the path that leads from the base to the component. The diversity score limit is set at 0 from the beginning. On a 3,040,715 transaction commercial data set, the proposed techniques are assessed. It uses the Item-Encoding algorithm to determine the diversity value of each detected maximal frequent item collection. This work proposes a simple approach for mining MDIFs that incorporates the cutting-edge FPMAX. And the recommended bound-based method performs noticeably better than the fundamental approach.

In [3], Cagatay Turkay et al introduce Progressive Data Science as a unique knowledge discovery paradigm in which progress is ingrained in each stage of the data science process. A combined effort from numerous research communities is necessary to ensure progress in such a novel paradigm. The authors provided a unified perspective through a reevaluation of the widely used and major KDD pipeline, reviewed the numerous issues raised by progressiveness, and then discussed some of the promising early efforts from various communities who are interested in progressive approaches. It is necessary to develop new techniques that can manage progressive computations because the majority of research so far has concentrated on batch processing settings. These methods suggest which examples to incorporate into a modeling framework to greatly enhance the precision and assurance of a learned model.

In [4], Murali Bharatham et al suggested the ARM method for obtaining unit rule matches in the collection of periodic data collection. The authors identify the non-obscure client raising style in the star bushel analysis. Additionally, considerations for the goods that would be safeguarded for customers should be made. mining for associations through social database-driven economic frameworks. Characteristic pillars were placed on various benches during Flat Partitioning. The entire database is divided into n sections in the first phase. Every wrapped database is individually stacked into necessary memory before the components of a neighborhood visit are located. Syndicate every locally visited element, and create a globally welcoming environment. For KDD, the dynamic inspection point is used to look for relationships within the object sets. On a single processor, the balanced Apriori calculation's delayed effects work satisfactorily. Working out parallel processors is necessary.

In [5], Bhukya Krishna and Geetanjali Amarawat provided a method for calculating the ongoing thing sets using the exchanges database's single output in the circle. Because small portions of the database can fit into primary storage with ease, the approach solves the memory issue for huge databases that cannot fit into primary memory. The method determines these itemsets' negative border, which is indicated by NBD (S). The set of candidate itemsets that did not meet minimal support constitutes the negative boundary. The acceptance for the candidate sets is counted using tidlists. The support for an itemset in a partition is calculated by dividing the primary key of a tidlists of the itemset by the entire quantity of transactions in the partition. There is a tid list created for each itemset. The model stores the details of each customer's items in a Customer Bit- vector display with a sliding window, which is based on the BitVector concept.

In [6], Dongyu Liu et al model multivariate ST data as tensors, and then offer a novel piecewise rank-one tensor breakdown algorithm that supports automatically dividing the data into homogenous divisions and identifying the latent features in each split for contrast and visual summarization. A quantitative assessment of how accurately the derived patterns visually match the actual data is optimized by the algorithm. With the help of TPFlow, analysts may gradually break the dataset into small subsets along several parameters using a directional and iterative approach. A subset of the data created during the partitioning process is represented by each node in the tree. The dataset is more complicated, making it more challenging to directly characterize the original data using rank-1 factors because the deviations persist even after multiple partitioning rounds. When the vector size grows, the decomposition process may lose effectiveness. Contextual information is not included in the proposed method.

Table 1: Comparative Mechanisms of the Existing Approaches

S.No	Author	Method	Merits	Demerits
1.	Md. Rezaul Karim	Apache Spark-based MFP mining	Hadoop MapReduce architecture is used.	Many pattern-matching rules
2.	Dingming Wu	FP*-tree	Includes cutting-edge FPMAX	High costs, not scalable, fits for smaller datasets.
3.	Cagatay Turkay	Progressive Data Science using KDD	provided a unified perspective through a reevaluation	Only for batch processing.
4.	Murali Bharatham	ARM method	FLAT partitioning	Not working using parallel processors as single processors delays.
5.	Bhukya Krishna	Improved Apriori algorithm	Splits huge datasets into small clusters and key is issued.	Slow computation, scans dataset many times.
6.	Dongyu Liu	novel piecewise rank-one tensor breakdown algorithm	Homogeneous splits of the data use a directional iterative approach so less computation time.	Not scalable, contextual information is not included.

3. Proposed Methodology

The proposed genetic algorithm is basically divided into two phases: leading and following. The initial position is known as “Leader”. It updates the position of salp in every iteration as shown in equation (x)

$$Position_j^l = Current\_pos_j \pm 2 * e^{(\frac{4*t}{L})^2} * (upper\_bou_j - lower\_bou_j) * rand \text{ - (x)}$$

Where,

$Position^l_j$ denotes the leader movement in the direction of j^th^ dimension

$Current\_pos_j$ represents the current position of leader

Upper_bou represents the upper boundary of the particle

Lower_bou denotes the lower boundary of the particle

Rand denotes a random variable with a range of 0 to 1

t & L denotes displacement with respect to time

The exploration of domain search space is majorly dependent on the time and displacement. The position of leader slap is updated with respect to the sources available. The position of the follower slaps is updated using the equation (x)

$$Follower\_pos_j^i = \frac{1}{2} * (curr\_pos_j^i + curr\_pos_j^{i-1}) \text{ - (x)}$$

The proposed research to define a mathematical model integrates the slap algorithm with SVM approach by tuning the coefficient values of the non-linear kernel functions. The objective function of the algorithms tries to reduce the error rate along with maximization of accuracy. The pseudo code for computation of objective function is discussed in below section:

Pseudocode for ESLAP Algorithm:

Input: Online Retail Store Data, ORSD

Output: Accuracy & Top-K utilities

```
Begin:
1. Initialize slap populations (sp), lower (l) and upper (u) boundaries
2.  for i←0 to sp-1:
       fol_bin←binary_conversion(input[i], threshold,sp,len(ORSD))
       for j←0 to i:
          fit[i,j]←obj_function(input[i,j], class[i,j], opts)
          con←rand(0,1)
          if(i==0 & con>=0.5):
              update the position using equation (x)
          elif:
              update the follower position using equation (x)
3. Initialize the slap parameters
4. Define model using SVM non-linear kernel
5. acc[0], best_acc←auc_score(model.fit(X,y))
6. for i ←0 to n-1:
       acc[i]← auc_score(i, model.fit(X,y))
        if(acc[i]>best_acc[i]):
           best_acc←acc[i]
```

4. Results & Discussion

Figure 2 denotes the central tendency metrics over the products to analyze the data based on the quantities and customer ids. It is observed that most of the transactions has frequent customer patterns.

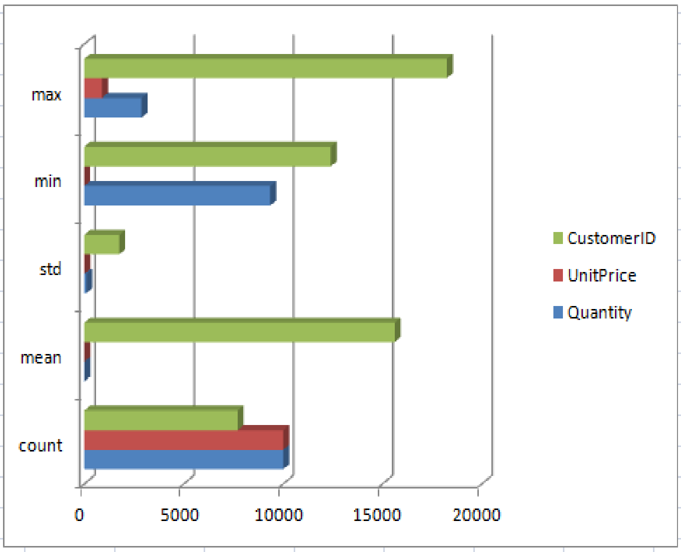
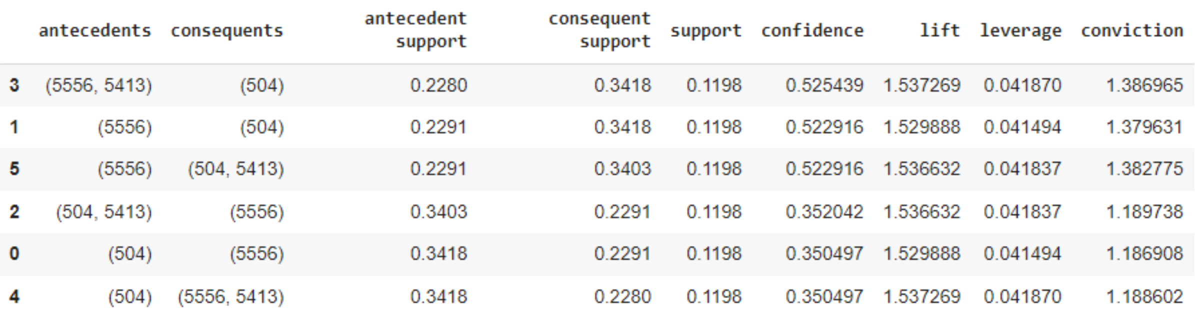


Figure 2: Descriptive Statistics of Item sets



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
3	(5556, 5413)	(504)	0.2280	0.3418	0.1198	0.525439	1.537269	0.041870	1.386965
1	(5556)	(504)	0.2291	0.3418	0.1198	0.522916	1.529888	0.041494	1.379631
5	(5556)	(504, 5413)	0.2291	0.3403	0.1198	0.522916	1.536632	0.041837	1.382775
2	(504, 5413)	(5556)	0.3403	0.2291	0.1198	0.352042	1.536632	0.041837	1.189738
0	(504)	(5556)	0.3418	0.2291	0.1198	0.350497	1.529888	0.041494	1.186908
4	(504)	(5556, 5413)	0.3418	0.2280	0.1198	0.350497	1.537269	0.041870	1.188602

Figure 3: Support & Confidence values for Zinc Product

In Figure 2, the model projects the rules that try to search the frequency using the description "Zinc". The equations are presented below:

Lift:

The difference between the actual and expected confidence in an association rule is known as the lift value.

$$\textbf{Lift (Zinc => Zinc is a famous mineral)} = \frac{\textbf{Confidence (}\text{Zinc => Zinc is a famous mineral)}}{\textbf{P}\text{ (Zinc is a famous mineral)}}$$

$$= \frac{\textbf{P (}\text{Zinc }\textbf{U}\text{ Zinc is a famous mineral}\textbf{)}}{\textbf{P}\text{ (Zinc is a famous mineral) }\textbf{P}\text{(Zinc)}}$$

Confidence:

Confidence is the possibility that when a person purchases item A, he or she is likely to also purchase item B.

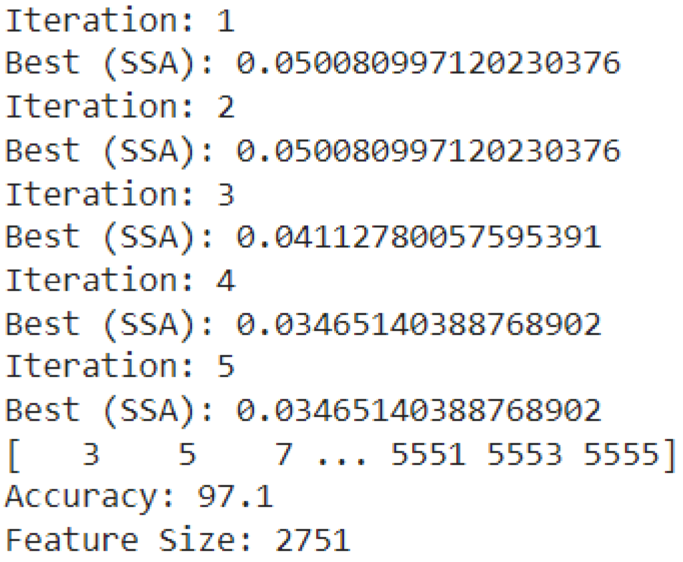
$$\textbf{Confidence(Zinc-> Zinc is a famous mineral)} = \frac{\textbf{Support(}\text{Zinc }\textbf{U}\text{ Zinc is a famous mineral}\textbf{)}}{\textbf{Support}\text{(Zinc)}}$$

Support:

The proportion of groups that contain each of the things stated in an association rule is known as its support.

All of the groupings that were taken into account are used to generate the percentage value.

$$\text{Support (Zinc)} = \frac{\text{Number of transactions in which Zinc appears}}{\text{Total number of transactions}}$$



```
Iteration: 1
Best (SSA): 0.05008099712023037б
Iteration: 2
Best (SSA): 0.050080997120230376
Iteration: 3
Best (SSA): 0.04112780057595391
Iteration: 4
Best (SSA): 0.03465140388768902
Iteration: 5
Best (SSA): 0.03465140388768902
[   3    5    7 ... 5551 5553 5555]
Accuracy: 97.1
Feature Size: 2751
```

Figure 4: Accuracy Metrics based on Feature Set

Figure 4 denotes the best accuracy obtained with the specified number of iterations. The accuracy of the model is improved and number of features is reduced.

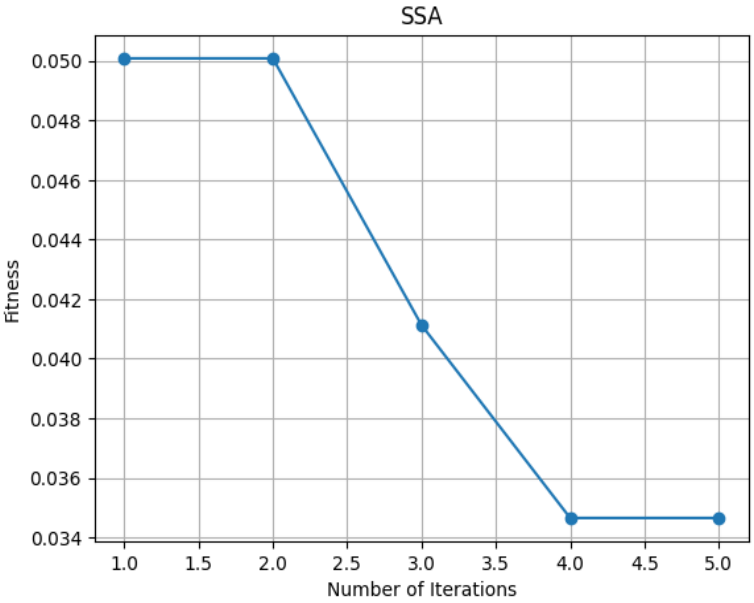


Figure 5: Fitness Computation based on Number of Iterations

Figure 5 denotes the fitness function generated at each iteration by denoting X-axis with number of iteration and Y-axis with fitness values.

5. Conclusion

The identification of patterns for finding the frequency of items using the genetic approaches has more impact than the machine learning. The model initially utilized the concept of encoding units to convert the categorical data into numerical so that any learning algorithm can train the model easily. Later it visualizes the statistical representation of the every column to identify the relation in terms of central tendency. When the model is converted into to numerical the number of attributes are increased and it has to identify the commonness using the genetic approach known as "SLAP", which +2.9% better than the existing approaches. It also evaluated by the other metrics associated with sequence mining like lift, confidence, and support.

References

Karim, M. R., Cochez, M., Beyan, O. D., Ahmed, C. F., & Decker, S. (2018). Mining maximal frequent patterns in transactional databases and dynamic data streams: A spark-based approach. Information Sciences, 432, 278–300. doi:10.1016/j.ins.2017.11.064

Wu, D., Luo, D., Jensen, C. S., & Huang, J. Z. (2019). Efficiently Mining Maximal Diverse Frequent Itemsets. Prosody, Phonology and Phonetics, 191–207. doi:10.1007/978-3-030-18579-4_12

Turkay, C., Pezzotti, N., Binnig, C., Strobelt, H., Hammer, B., Keim, D. A., … Rusu, F. (2018). Progressive Data Science: Potential and Challenges. doi:10.48550/ARXIV.1812.08032

M. Bharatham and Nagalakshmi, "ARM using Apriori with pretend Temporal database portioned Slant in weka tool," 2018 3rd International Conference on Contemporary Computing and Informatics (IC3I), 2018, pp. 155-159, doi: 10.1109/IC3I44769.2018.9007278.

Krishna, B., & Amarawat, G. (2018). Data Mining in Frequent Pattern Matching Using Improved Apriori Algorithm. Emerging Technologies in Data Mining and Information Security, 699–709. doi:10.1007/978-981-13-1498-8_61

Liu, D., Xu, P., & Ren, L. (2018). TPFlow: Progressive Partition and Multidimensional Pattern Extraction for Large-Scale Spatio-Temporal Data Analysis. IEEE Transactions on Visualization and Computer Graphics, 1–1. doi:10.1109/tvcg.2018.2865018

Y. Liu, S. Gao, J. Shi, X. Wei and Z. Han, "Sequential-Mining-Based Vulnerable Branches Identification for the Transmission Network Under Continuous Load Redistribution Attacks," in IEEE Transactions on Smart Grid, vol. 11, no. 6, pp. 5151-5160, Nov. 2018, doi: 10.1109/TSG.2020.3003340.

Ganesan, M., Shankar, S. High utility fuzzy product mining (HUFPM) using investigation of HUWAS approach. J Ambient Intell Human Comput 13, 3271–3281 (2019). https://doi.org/10.1007/s12652-021-03231-8.

D. H. Tran, T. T. Nguyen, T. D. Vu, and A. T. Tran, “MINING TOP-K FREQUENT SEQUENTIAL PATTERN IN ITEM INTERVAL EXTENDED SEQUENCE DATABASE”, JCC, vol. 34, no. 3, p. 249–263, Nov. 2018. DOI: https://doi.org/10.15625/1813-9663/34/3/13053.

U. Suvarna and Y. Srinivas, “Efficient high-utility itemset mining over variety of databases: a survey,,” in *Soft Computing in Data Analytics*, pp. 803–816, Springer, Berlin, Germany, 2019. DOI: 10.1007/978-981-13-0514-6_76.

Lin, J.C.-W., Djenouri, Y., Srivastava, G., Li, Y., Yu, P.S.: Scalable mining of high-utility sequential patterns with three-tier MapReduce model. ACM Trans. Knowl. Discov. Data **16**(3), 60:1–60:26 (2019) DOI: https://doi.org/10.1145/3487046.

Nunez-del-Prado, M.; Maehara-Aliaga, Y.; Salas, J.; Alatrista-Salas, H.; Megías, D. A Graph-Based Differentially Private Algorithm for Mining Frequent Sequential Patterns. *Appl. Sci.* **2019**, *12*, 2131. https://doi.org/10.3390/app12042131.

Dahiya, Vandna, and Sandeep Dalal. "A Systematic Literature Review of Utility Itemset Mining Algorithms for Large Datasets." *REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS* 11.3 (2019): 565-582.

V. S. Tseng, B. E. Shie, C. W. Wu, and P. S. Yu, ‘‘Efficient Algorithms For Mining High Utility Itemsets From Transactional Databases,’’ IEEE Transactions Knowledge and Data Engineering, volume 25, number 8, pp. 1772–1786, 2013, https://ieeexplore.ieee.org/document/6171188.

Sathyavani, D., Sharmila, D. RETRACTED ARTICLE: An improved memory adaptive up-growth to mine high utility itemsets from large transaction databases. J Ambient Intell Human Comput 12, 3841–3850 (2019). https://doi.org/10.1007/s12652-020-01706-8.

R. Agarwal, A. Gautam, P. Dixit and A. Rana, "An Approach to Mine Frequent Item Sets Considering Negative Item Values," 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2020, pp. 208-211, doi: 10.1109/ICRITO48877.2020.9197870.

Singh, K., Kumar, R. & Biswas, B. High average-utility itemsets mining: a survey. Appl Intell 52, 3901–3938 (2019). https://doi.org/10.1007/s10489-021-02611-z.

Chunkai Zhang, Zilin Du, Wensheng Gan, Philip S. Yu, TKUS: Mining top-k high utility sequential patterns, Information Sciences, Volume 570, 2019, Pages 342-359, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2021.04.035.