ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your uilt-MATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 7, Issue 5, May 2021

TOWARD PRACTICAL PRIVACY PRESERVING IN FREQUENT ITEM SET IN CLOUD USING APRIORI

S.Kavitha

Asst Professor, Computer Science and Engineering

B.Sangeerthana

PG Scholar, Computer Science and Engineering

P.S.R Engineering College,Sivakasi.

Email: s.geerthi1998@gmail.com

Ph no: 9360872554

Abstract- **Cloud computing has become a big name in present era. It has proved to be a great solution for storing and processing huge amount of data. It provides us demand, scalable, pay-as-you go compute and storage capacity. Data mining techniques implemented with cloud computing paradigm are very useful to analyze big data on clouds. In our dissertation we have used association rule mining as a data mining technique. In particular we have used Apriori algorithm for association rule mining. It has been observed that the original Apriori algorithm was designed for sequential computation so directly using it for parallel computation doesn't seems a good idea. So we have improved the Apriori algorithm (FP Growth) so as to suit it for parallel computation platform. We have used CloudSim Simulator for cloud computing.**

I. INTRODUCTION

With the increase in Information Technology, the size of the databases created by the organizations due to the availability of low-cost storage.

evolution in the data capturing technologies is also increasing. These organization sectors include retail, petroleum, telecommunications, utilities, manufacturing, transportation, credit cards, insurance, banking and many others, extracting the valuable data, it necessary to explore the databases completely and efficiently.

Knowledge discovery in databases (KDD) helps to identifying precious information in such huge databases. This valuable information can help the decision maker to make accurate future decisions. KDD applications deliver measurable benefits, including reduced cost of doing business, enhanced profitability, and improved quality of service. Therefore Knowledge Discovery in Databases has become one of the most active and exciting research areas in thedatabase community.

Cloud computing can be defined as the use of computing resources that are delivered as a service over a network. With traditional computing paradigms we run the software and store data on our computer system. These files could be shared in a network. The importance of cloud computing lies in the fact that the software are not run from our computer but rather stored on the server and accessed through internet. Even if a computer crashes, the software is stillavailable for others to use. The concept of cloud computing has developed from clouds. A cloud can be considered as a large group of interconnected computers which can be personal computers or network servers; they can be public or private.

The concept of cloud computing has spread rapidly through the information technology industry. The ability of organizations to tap into computer applications and other software via the cloud and thus free themselves from building and managing their own technology infrastructure seems potentially irresistible. In fact some companies providing cloud services have been growing at double digit rates despite the recent economic downturn.

Cloud Mining can be considered as a new approach to apply Data Mining. There is a lot of data and unfortunately this huge amount of data is difficult to mine and analyze in terms of computational resources. With the cloud computing paradigm the data mining and analysis can be more accessible and easy due to cost effective computational resources. Here we have discussed the usage of cloud computing platforms as a possible solution for mining and analyzing large amounts of data.

behavior analysis is of great importance in understanding the effectiveness of marketing and merchandising campaigns Deep shopping behavior data can help retailers capture customers' preferences, test new arrivals, and adjust marketing strategies. Mining customer shopping behavior in online stores is achievable by analyzing click streams and shopping carts . However, physical store retailers lack effective methods to identify customer behaviors. The only available information is the sales history, which fails to reflect customer behaviors before they check out, e.g. how customers browse the store, which products they show an interest in, and what products they match up.

Therefore, it is essential to explore new ways of capturing customer behaviors in physical stores.Previous efforts have exploited cameras to monitor customer shopping behaviors in clothing stores . However, suchmethods involve sophisticated computer vision techniques to recognize and analyze arm motions. Alternative methods track customer routes in stores to mine hot zones and popular products ,For example, the more customers traverse a route, the higher attention the items along this route gain. However, these approaches still fail to provide high-fidelity shopping behaviour information such as product browsing, pick-up actions and trying on clothes. RFIDs are emerging as an essential component of Cyber Physical Systems. Many well-known garment manufacturers (e.g., Abercrombie &amp; Fitch, Calvin Klein, Decathlon) adopt passive RFIDs for sales tracking and anti-counterfeiting

II EXISTING AND PROPOSED SYSTEM

2.1 EXISTING CONCEPT

1. In existing, RFID the user had to check the branded details of the particular product from the cluster of information and from those cluster of information the user had to pluck the needed.

2. It took lot of time to purchase the single product, which it cause some external physical impacts to the user.

The majority of existing approaches to recommender systems focus on recommending the most relevant items to individual users without taking consideration of any contextual information, such as time, place and the company of other people (e.g., for watching movies or dining out). In other words, traditionally recommender systems deal with applications having only two types of entities, users and items, and do not put them into a context when providing recommendation. It also providesrecommendations that are based on the user's area of interests, customer searches and also suggests products based on it.

For e.g. Amazon uses user view data.If any customer is searching a product from a particular category the system suggests a product form the same category. It is also based on the current search by the user, the site recommends products. commerce recommendation algorithms often operate in a challenging environment.

For example:

• A large retailer might have huge amounts of data, tens of millions of customers and millions of distinct catalog items.

• Many applications require the results set to be returned in real-time, in no more than half a second, while still producing high-quality recommendations.

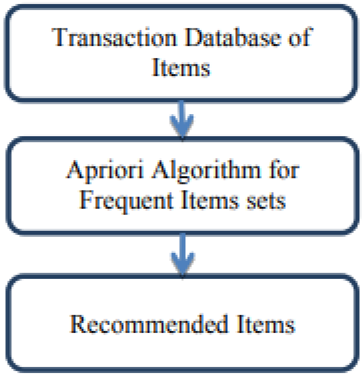
• Older customers can have a glut of information, based on thousands of purchases and ratings.

• Customer data is volatile: Each interaction provides valuable customer data, and the algorithm must respond immediately to new information. However, in many applications,such as recommending a vacation package, personalized content on a Web site, ora movie, it may not be sufficient to consider only users and items – it is also important to incorporate the contextual information into the recommendation process in order to recommend items to users in certain circumstances. Forexample, using the temporal context, a travel recommender system would provide a vacation recommendation in the winter that can be very different from the one in the summer. Similarly, in the case of personalized content delivery on a Web site, it is important to determine what content needs to be delivered (recommended) to a customer and when. Every user who visits the site may not buy a product. They can just go through it and based on those search results the site recommends a product

2.2 PROPOSED SYSTEM

In our proposed system, we are planning to use a methodology dealing of frequent item set based recommendation using Apriori Algorithm. Here we are using a &quot;bottom up&quot; approach, where frequent subsets are extended one item at a time and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Especially important are pairs or larger sets of items that occur much more frequently than would be expected were the items bought independently. The whole point of the algorithm (and data mining, in general) is to extract useful information from large amounts of data. The algorithm

aims to find the rules which satisfy both a minimum support threshold and a minimum confidence threshold.


Transaction Database of Items
Apriori Algorithm for Frequent Items sets
Recommended Items

Here we are describing the Apriori Algorithm for finding frequent item sets. The key idea behind this algorithm is that any item set that occurs frequently together must have each item (or we can say any subset) occur at least as frequently.

First Pass. In this algorithm, firstly we make one pass on all the tuples and retain a count for all the n items. Here we canuse a Hash Table. We set a threshold t and then only keep items that occur at least tn time (that is in at lead t percent of the tuples). For any frequent item set that occurs in at least 100t% of the tuples, must have each item also occur in at least 100t% of the tuples. Second Pass. After the first pass, we make a second pass over all tuples. On this pass, we examine for frequent pairs of items, specifically for those items which occur in at least at- fraction of all baskets. Both items must have been found in the first pass. So we need to deliberate only n1 2 ≈n1 2 /2 pairs of counters for these pairs of elements. After this pass, we can again reject all pairs which occur less than at-fraction of all tuples. After this remaining set is expected far less than n1 2 /2. And these remaining pairs are already relatively interesting. They record all pairs that co- occur in more than at-fraction of purchases. And obviously include those pairs also which are occurring together even more frequently. Further Passes. On the ith pass, we can find sets of i items that occur together frequently (above at- threshold). For example, on the third pass we only need to consider triples were all sub-pairs occur at least at-fraction of times themselves. These triples can be found as follows:

Firstsort all pairs (p, q) by their smaller indexed item (let smaller indexed be p). Then for each smaller indexed item p, consider all completions of this pair q. Now look at the pairs (q, r) with smaller item as q. For each of these pairs, check if the pair (p, r) also remains. Only triples (p, q, r) which pass all of these tests are given counters in the third pass

The Apriori Algorithm

The Apriori-based algorithms find frequent itemsets based upon an iterative bottom-up approach to generate candidate itemsets. Since the first proposal of association rules mining by R. Agrawal, many researchers have been done to make frequent itemsets mining scalable and efficient. But there are still some deficiencies that Apriori based algorithms suffered from, which include: too many scans of the Three algorithms are proposed to investigate the Apriori-like algorithms in the MapReduce paradigm. transaction database when seeking frequent itemsets, large amount of candidate itemsets generated unnecessarily and so on. The proposed of our method is the classical A Priori algorithm. Our contributions are in providing novel scalable approaches for each building block. We start by counting the support of every item in the dataset and sort them in decreasing order of their frequencies. Next, we sort each transaction with respect to the frequency order of their items. We call this a horizontal sort.

We also keep the generated candidate itemsets in horizontal sort. Furthermore, we are careful to generate the candidate itemsets in sorted order with respect to each other. We call this a vertical sort. When itemsets are both horizontally and vertically sorted, we call them fully sorted. As we show, generating sorted candidate itemsets (for any size k), both

horizontally and vertically, is computationally free and maintaining that sort order for all subsequent candidate and frequent itemsets requires careful implementation, but no cost in execution time. This conceptually simple sorting idea has implications for every subsequent part of the algorithm. In particular, as we show, having transactions, candidates, and frequent itemsets all adhering to the same sort order has the following advantages:

Generating candidates can be done very efficiently. Indices on lists of candidates can be efficiently generated at the same time as are the candidates. Groups of similar candidates can be compressed together and counted simultaneously. Candidates can be compared to transactions in linear time. Better locality of data and cache-consciousness is achieved. Our particular choice of sort order (that is, sorting the items least frequent first) allows us to with minimal cost entirely skip the candidate pruning phase

The Apriori algorithm mines all the frequent itemsets in a transactional database, where each transaction ti contains a set of items called itemset. An itemset having k items is called a k-itemset and its length is k. An itemset X is frequent if its support, which is the fraction of transactions containing X in the database, is at least certain userspecified minimum support min_sup. Let Lk denote the frequent itemsets of length k and Ck denote the candidate itemsets of length k. The Apriori algorithm joins Lk-1 to generate Ck, counts the supports of Ck, and determine the Lk in k-thdatabase scanning. The algorithm terminates when no Ck or Lk is generated. Note that usually the discovery of frequent 1-itemsets is accomplished by a simple counting of items in the first pass of database scanning (pass-1).

Starting from pass-2, the hash-trees are used for arranging Ck to facilitate fast support counting.The pruning of candidates using the downward closure property is effective for candidates of length larger than two, starting from pass The fundamentals of parallelizing the Apriori algorithm in the MapReduce framework is to design the map and the reduce functions for candidate generations and support counting. The first proposed algorithm, Single Pass Counting (SPC), finds out frequent k-itemsets at k-th pass of database scanning in a mapreduce phase. The second proposed algorithm, Fixed Passes Combined-counting (FPC), finds out frequent k-, (k+1)-, …, and (k+m)-itemsets in a map-reduce phase. In this paper, FPC discovers frequent k-, (k+1)-, and (k+2)-itemsets.

The third proposed algorithm, Dynamic Passes Combined-counting (DPC), considers the workloads of nodes and finds out as many frequent itemsets of various lengths as possible in a map-reduce phase. For convenience, a map task is called a mapper, and a reduce task is called a reducer in the following context.

In general, the number of mappers is larger than the number of reducers in MapReduce. With the size of cluster increasing, the more mappers can be used for processing data, and the problem can be divided into smaller granularity. In all of our algorithms, each mapper calculates counts of each candidate from its own partition, and then each candidate and corresponding count are output. After map phase, candidates and its counts are collected and summed in reduce phase to obtain partial frequent itemsets. By using count distribution between map phase and reduce phase, the communication cost can be decreased as much as possible. Since frequent 1-itemsets are found in pass- 1 by simple counting of items. Phase-1 of all the three algorithms is the same, as The mapper outputs pairs for each item contained in the transaction. The reducer collects all the support counts of an item and outputs the pairs as a frequent 1-itemset to the file L1 when the count is no less than the minimum support count.

III SYSTEM ARCHITECTURE

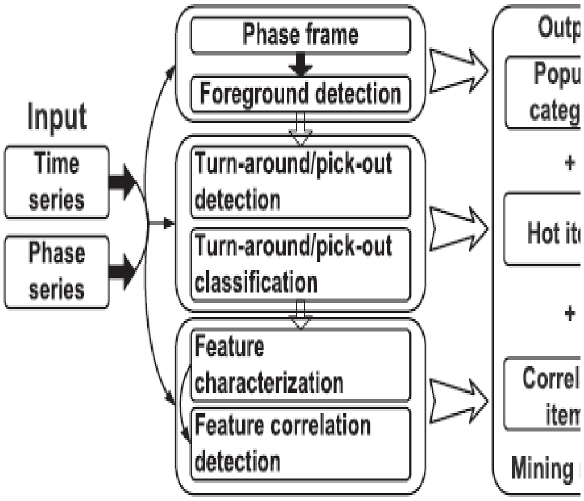
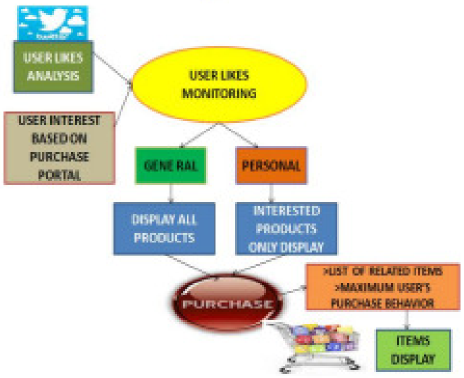


Fig 3.1 SYSTEM ARCHITECTURE



IV MODULES

4.1 SYSTEM IMPLEMENTATION

Introducing the modules used in the implementation are: User Registration In client side user can enter all details. Then user can login using particular username and password. All the inserted also updated items are added into the product list. Then select user wanted items then add all items into cart products with count of the each item. A warning message will display in dialogue box when the customer type the quantity above the constraint value mentioned in the database. All selected items are displayed in the cart product list and purchase the required items

like Application: User can register in application and go for login by giving valid user name and password. If the user name and password is valid the user can login into home page. Once we login in home page the display of several products is to be done. Based on user interest he go for likes to the products. So this likes is going to monitor by server and stored in data base

Purchase Portal Consumer buying behaviour is the sum total of a consumer attitudes, preferences, intentions and decisions regarding the consumer behavior in the marketplace when purchasing a product or service. The study of consumer behaviour draws upon social science disciplines of sociology, and economics. At this stage, the consumer will make a purchasing decision. [6] discussed about a Secure system to Anonymous Blacklisting. The secure system adds a layer of accountability to any publicly known anonymizing network is proposed. Servers can blacklist misbehaving users while maintaining their privacy and this system shows that how these properties can be attained in a way that is practical, efficient, and sensitive to the needs of both users and services. This work will increase the mainstream acceptance of anonymizing networks such as Tor, which has, thus far, been completely blocked by several services because of users who abuse their anonymity. In future the Nymble system can be extended to support Subnet-based blocking. If a user can obtain multiple addresses, then nymble-based and regular IP-address blocking not supported. In such a situation subnet-based blocking is used. Other resources include email addresses, client puzzles and e-cash, can be used, which could provide more privacy. The system can also enhanced by supporting for varying time periods.

The ultimate decision may be based on factors such as price or availability. For example, our consumer has decided to purchase a particular model of car because its price was the best shecould negotiate and the car was available immediately

4.2 SERVER

module will monitor the entire User's information in their database and verify them if required. Also the Server will store the entire User's information in their database. Also the Server has to establish the connection to communicate with the Users. The Server will update the each User's activities in its database. The Server will authenticate each user before they access the Application. So that the Serverwill prevent the Unauthorized User from accessing the Application.

The High utility item set feature selection will use the hierarchical manner with fast Apriori-based algorithm to generate the frequent sets of attribute relation rules. With Fast Apriori-based algorithm used to recognize and create features that are associated and change to other features sets in the group, more successful action in hierarchical technique is required. We have to filter the rules that appropriate to research objective. Fast Apriori is a formation to count candidate item sets efficiently. It generates candidate item sets of length k from the k-1 item sets and keeps away from expanding all the item sets. Then it removes the candidates whichhave an infrequent sub pattern. The candidate set contains all frequent k-length item sets. After that, it scans the entire transaction database to determine frequent item sets among the candidates. With fast Apriori technique the algorithm can reduce time processing in generating fewer groups of item sets and avoidinfrequent candidate item sets expansion

V CONCLUSION

Cloud computing is an architecture which is known for its powerful capability of computation and storage and resource sharing. These features make cloud computing favorable to data mining service in network environment. We have discussed association rule mining in cloud environment and various parallel and distributed mining algorithms. Data mining on cloud computing paradigm can benefit us to a great extent. That is why we have implemented data mining technique on cloud platform. Out of many data mining techniques we have studied association rule mining technique in this paper. More specifically we have association rule mining in cloud computing environment.

VI REFERENCES

[1] L. Shangguan et al., “Shopminer: Mining customer shopping behavior in physical clothing stores with cots rfid devices,” in Proc. ACM SenSys, Nov. 2015, pp. 113–125.

[2] D. R. Bell and J. M. Lattin, “Shopping behavior and consumer preference for store price format: Why ‘large basket’ shoppers prefer EDLP,” Marketing Sci., vol. 17, no. 1, pp. 66–88, Feb. 1998.

[3] Y. H. Cho, J. K. Kim, and S. H. Kim, “A personalized recommender system based on Web usage mining and decision tree induction,” Exp. Syst. Appl., vol. 23, no. 3, pp. 329–342, Oct. 2002.

[4] R. Kohavi, N. J. Rothleder, and E. Simoudis, “Emerging trends in business analytics,” Commun. ACM, vol. 45, no. 8, pp. 45–48, Aug. 2002.

[5] M. Popa, A. K. Koc, L. J. M. Rothkrantz, C. Shan, and P. Wiggers, KinectSensing of Shopping Related Actions. Berlin, Germany: Springer, pp. 91–100, 2011.

[6] Christo Ananth, A.Regina Mary, V.Poornima, M.Mariammal, N.Persis Saro Bell, “Secure system to Anonymous Blacklisting”, International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST), Volume 1,Issue 4,July 2015,pp:6-9.

[7] Y. Liu, Y. Zhao, L. Chen, J. Pei, and J. Han, “Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays,”IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 11, pp. 2138–2149, Nov. 2012.

[8] L.-A. Tang et al., “On discovery of traveling companions from streaming trajectories,” in Proc. ICDE, Apr. 2012, pp. 186–197.

[9] T. Staake, F. Thiesse, and E. Fleisch, “Extending the EPC network: The potential of RFID in anti-counterfeiting,” in Proc. ACM SAC, Mar. 2005,pp. 1607–1612.