MapReduce Based Frequent Itemset Mining for Improved Electronic Evidence Analysis Using Palm-Print

Mayuresh Sonawane, Navnath Borade, Manoj Jadhav, Vikas Mandal, Prof. A. S. Chandgude

Department of Computer Engineering,
S.N.D COE & RC, Yeola.

***Abstract*- In Data mining association rule, mining becomes one of the important tasks of expressive technique which can be defined as discovering meaningful patterns from large collection of data mining frequent itemset is very fundamental part of association rule mining. In this, frequent itemset mining plays an important role in mining relations, associations and many everyday data mining fields such as electronic evidence analysis area. The developed associative rule miner gets the MapReduce scalability to huge datasets and to thousands of processing nodes. The earlier procedures are not suit for this problem especially in criminological region So, Along with these algorithms i.e. a novel parallelized algorithm called PISPO based on the cloud-computing framework MapReduce, which is widely used to cope with large scale data and captures both the content and state to be distributed to the changed and original of the transactions dataset to SPO-tree, we introduce palm print matching algorithm to increase the efficiency of finding the criminal records or criminal profile.**

***Index Terms*- computer crime, PISPO, ISPO-tree, MapReduce, frequent itemset, data mining, association rules.**

I. INTRODUCTION

Nowadays databases are contained of terabytes or more data in it. As they are able to provide accommodations to huge mass of diverse data, different variety of strategic evidence lies hidden inside it. So, through effective data mining only we can able to draw significant conclusions which are the basic purpose of data mining. As data are being accumulated Uninterruptedly as well as rapidly whether it is a research field, education, and market products, medical science, electronic information, media, entertainment etc. it is difficult to get faster and suitable information by traditional manual analysis which is dull as well as very Difficult. So, data mining is used basically to decrease costs through proper detection and prevention of waste and scam, Gaining precise and up-to-date information increase incomes through improved marketing plans. We can take an example of study of finding presence of water in the planet Mars. Scientists receive dissimilar data from Mars through satellites. Those data are varying time to time as the satellite provides new groups of data in different time period. So it is mandatory to have a widespread research and investigation about the data to draw any deduction about the presence of water in Mars. The job is very challenging as well as essential effective research analysis. So through an effective data mining analysis scientists can able to find the result more systematically which is not that possible or easier in old-style analysis. So data mining finds patterns and relations in data which are stored in databases, data warehouses, or other info repositories providing more advanced and effective information which is increased by utilizing models equipped with classy techniques. [1] In other way, Data Mining allow to explore massive data in such a way that its end result is obtaining of fruitful knowledge information. It is a multidisciplinary field, drawing work from various areas like:-

- Artificial intelligence
- Neural networks
- Machine learning
- Database technology
- Statistics
- Pattern recognition
- Signal processing
- Spatial data analysis
- Business
- Economics
- Bio-informatics.

Data mining can be done in several types of data but one of the exciting evidence is that it can process three-dimensional data which is used for geographical, chip design, medical and satellite image Databases. Data mining is also known as KDD. KDD is the automatic extraction of novel, logical and possibly useful patterns indirectly stored in large databases, data Warehouses and other massive information repositories encompassing textual, numerical, graphical, spatial data. Pre mining and post mining tasks required for knowledge discovery. It is required the following essential pre mining and post mining tasks for knowledge discovery to be performed.

II. LITERATURE SURVEY

Y. S. Tan et al [7] proposed two algorithms to handle electronic evidence: one algorithm improves FP-Growth algorithm on the generation method of the frequency 1-set and the sensitivity of the new crime. The other improves Growth algorithm on regarding different nature of crime record as different weight, so that the different nature of the criminal record has a different importance, greatly improves these records' possibility of generating association rules. The first algorithm can find the latest emerging frequent itemsets and also cannot produce the latest association rules. But those update crimes have value, we cannot ignore them. Then we think all the item have values and cannot distinct to tree with them. We also discuss five existing FP-tree based algorithms namely CanTree, CP-Tree, SPO-tree, and PFP algorithm [6].

Leung et al [8] proposed the Canonical-Ordered Tree for incremental mining. This algorithm is designed so that it only needs one scan of dataset. In CanTree, items are arranged by some canonical order. The items are arranged according to a prefixed tree structure, so cannot be affected by the item Frequency. CanTree generates compact trees, if and only if the majority of the transactions contain a common pattern-based by canonical order. Otherwise, it may produce skewed trees with too many branches and thus with too many nodes.

Example: CanTree. Consider the dataset in Table 1. Fig. 1 shows the resulting CanTree tree after each transaction is added. This method keeps track of all items. this step, the item e is swapped with its ancestors c and d. As there are no further common items the remaining item in I2, which is item f, is inserted as a new branch to e. When I3 arrives, item b will being swapped with item a and moved up. The rest of the transactions will be inserted in the same manner.

TABLE I. EXAMPLE OF DATASET

TID	Transactions
I1	{ *a*, *b*, *c*, *d*, *e* }
I2	{ *a*, *f*, *b*, *e* }
I3	{ *b* }
I4	{ *d*, *a*, *b* }
I5	{ *a*, *c*, *b* }
I6	{ *c*, *b*, *a*, *e* }
I7	{ *a*, *b*, *d* }
I8	{ *a*, *b*, *d* }

Tanbeer et al [9] proposed a tree structure different from CanTree, called CP-tree that builds a compact prefixed structure. The CP-tree has a frequency descending structure through capturing some by some data from the dataset and restructuring itself dynamically. The construction operation consists of two phases: inserting and restructuring phase. Inserting phase inserts transactions into CP-tree according to current sorted order of the item-list and updates frequency based on it.

Y.S. Koh et al [10] proposed a novel tree structure called SPO-Tree (Single Pass Ordered Tree). The tree contains the content of the dataset and rearranges them into a more compact representation. The SPO-tree sorts the items of the transactions in descending order of frequency considering the current frequency of the items before inserting the transaction *In*, that is to say, frequency of items prior to the insertion of *In*-1 transaction. The tree is reconstructs once the proportion of the edit distance of the items in the sorted order changes above predefined threshold. So it reorganized the tree periodically based on a parameter called Edit Distance.

The FP-growth method is a depth-first algorithm. In the method, Han et al. proposed a data structure called the FP-tree (frequent pattern tree). The FP-tree is a compact representation of all relevant frequency information in a database.. An FP-tree T has a header table, T header, associated with it. Single items and their counts are stored in the header table in decreasing order of their frequency.

- *MapReduce Environment(Existing System)*

Map Reduce is emerging as an important programming model for large-scale data-parallel applications such as web indexing, data mining, and scientific simulation. Hadoop is an open-source implementation of Map Reduce enjoying wide adoption and is often used for short jobs where low response time is critical. tasks that appear to be stragglers. In practice, the homogeneity assumptions do not always hold.

The Map Reduce model popularized by Google is very attractive for ad-hoc parallel processing of arbitrary data. Map Reduce breaks a computation into small tasks that run in parallel on multiple machines, and scales easily to very large clusters of inexpensive commodity computers. Its popular open-source implementation, Hadoop, was developed primarily by Yahoo, where it runs jobs that produce hundreds of terabytes of data on at least 10,000 cores. If a node crashes, Map Reduce reruns its tasks on a different machine. Equally importantly, if a node is available but is performing poorly, a condition that we call a straggler, Map Reduce runs a speculative copy of its task on another machine to finish the computation faster. In this work, we address the problem of how to robustly perform speculative execution to maximize performance. Hadoop's scheduler starts speculative tasks based on a simple heuristic comparing each task's progress to the average progress. Although this heuristic works well in homogeneous environments where stragglers are obvious, we show that it can lead to severe performance degradation when its underlying assumptions are broken [3].

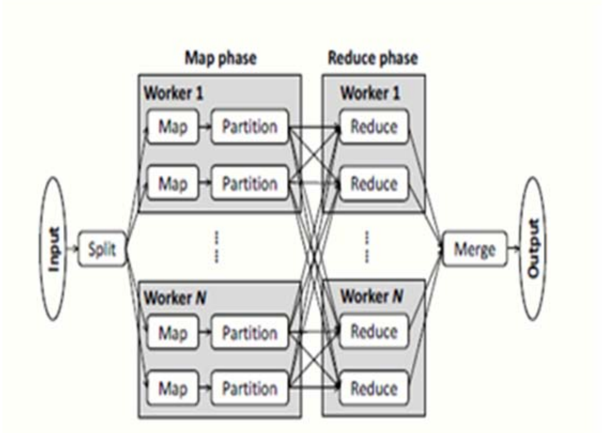

Fig 1.1: Workflow of MapReduce

Hadoop Map Reduce is an implementation of the Map Reduce programming model, freely available through the Apache license. It is not only a framework to implement and run Map Reduce algorithms but also a handy tool for developing alternative and improved systems for Map Reduce The Hadoop architecture is presented in Fig. 1. Map Reduce jobs are submitted to and managed by centralized service called job tracker. This notion of running the task in the node that has its input is called locality. Each node available to run Hadoop tasks runs a software service called task tracker that launches the tasks. Hadoop has mechanisms to tolerate the crash of task trackers and the tasks they execute. A task tracker sends heartbeat messages to the job tracker periodically [4] [5]. Map Reduce is standard software architecture, developed by Google, which aids in the design and execution of large scale data processing tasks. There are two primary components to the architecture: MAP and REDUCE. The MAP step takes in a chunk of input data and emits key value pairs which represent that data. In the word count example below the keys are the text of each word, and the value is a count. The reduce step received key- value pairs in which all identical keys are guaranteed to arrive at the same reduce function. In the word-count example the reduce function simply sums the values for each key and outputs the key with a total count. The output from all reduce steps is appended to an output file.

III. PROPOSED SYSTEM

Studies of Frequent Itemset (or pattern) Mining is acknowledged in the data mining field because of its broad applications in mining association rules, correlations, and graph pattern constraint based on frequent patterns, sequential patterns, and many other data mining tasks. As the minimum threshold becomes lower, an exponentially large number of itemsets are generated. Therefore, pruning unimportant patterns can be done effectively in mining process and that becomes one of the main topics in frequent pattern mining. Consequently, the main aim is to optimize the process of finding patterns which should be efficient, scalable and can detect the important patterns which can be used in various ways The Apriori variations (DHP, DIC, Partition, and Sample) algorithms among them HP tries to reduce candidate itemsets and others try to reduce database scan. DHP works well at early stages and performance deteriorates in later stages and also results in I/O overhead. For DIC, Partition, sample algorithm performs worse where database scan required is less then generating candidates. Vertical Layout based algorithms claims to be faster than Apriori but require larger memory space then horizontal layout based because they needs to load candidate, database and TID list in main memory. For projected layout based algorithms include FP-Tree and H-mine, performs better then all discussed above because of no generation of candidate sets but the pointes needed to store in memory require large memory space. FP-Tree variations include COFI-Tree and CT-PRO performs better than classical FP tree as COFI-tree performs better in dense datasets but with low support its performance degrades for sparse datasets and for CT-PRO algorithm performs better for sparse as well for dense data sets bu41t difficult to manage the compress structure [2] [12]. Therefore these algorithms are not sufficient for mining the frequent itemsets for large transactional database.

- *Palm Print Technique* (*proposed system*)

The use of palm prints for person identification traces back to Chinese deeds of sale in the 16th century [13]. Later in 1684, Grew introduced dermatoglyphics, a study of the epidermal ridges and their arrangement on the hand. The first systematic capture of hand, finger, and palm images for identification purposes was done by Herschel in 1858 [14]. Galton [15] discussed the basis of contemporary fingerprint science, and introduced palmar ridges and creases. He suggested that the ridges on the finger tips, palms, and soles are persistent and unique. Galton defined the peculiarities in the ridges as minutiae and introduced several different minutiae types. He also divided the palm into three regions and analyzed the correlation between the ridge flow and the major creases in each region.

1. Minute Extraction: - The performance of a minutiae extraction algorithm relies heavily on the quality of the input palmprint images. In order to ensure that the minutiae extraction algorithm is robust with respect to the quality of the input palmprint images, an enhancement algorithm that improves the clarity of the ridge structures is necessary. However, when the parameters are incorrect, true ridges may be missed and spurious ridges may be produced after filtering. Hence, reliable ridge direction and frequency estimation is very important for minutiae extraction.

1.1 Ridge Direction and Frequency Estimation

Funada et al. [16] proposed a palmprint enhancement approach, which performs image enhancement and local ridge direction and frequency estimation simultaneously. Local image blocks (8 _ 8 pixels) are modeled by sine waves and the six strongest waves (according to amplitude) are found in each block. In the image formed by the first strongest wave in each block, continuous blocks are clustered into regions. Generally, a region contains only ridges (such region is called ridge region) or only creases (such region is called crease region). Based on certain properties, these regions are classified as ridge or crease regions, and ridge regions are used as a single seed. Region growing algorithm is then used to grow the seed and obtain the enhanced image.

2. Minutiae Matching: - Given the minutiae features of two palmprints, the matching algorithm consists of 1) local minutiae matching— the similarity between each minutia of a partial print and each minutia of a full print is computed, 2) global minutiae matching—using each of the five most similar minutia pairs in step 1) as an initial set, a greedy matching algorithm is used to find additional matching minutia pairs, and 3) matching score computation—a matching score is computed for each set of matching minutia pairs and the maximum score is used as the matching score between two palmprints. A minutia is generally tagged with the following features: location, direction, type (ending or bifurcation), and quality (reliable or unreliable) [17]. Since the relative transformation between the two palmprints to be matched is not known a priori and considering the large size of palmprint images, the minutiae correspondence problem is very challenging.

To reduce the ambiguity in matching, we attach additional distinguishing information to a minutia in the form of a minutia descriptor. In the fingerprint recognition literature, four types of information have been widely used as minutia descriptors, namely image intensity , texture , ridge information , and neighboring minutiae.

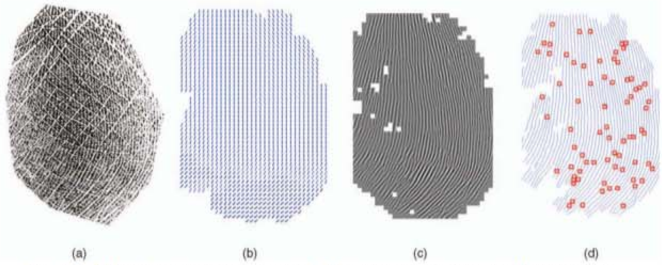

Fig. 2.1. Minutiae extraction. (a) A live-scan partial print (height: 636 pixels, width: 578 pixels) from the tenor region, (b) direction field, (c) enhanced image, and (d) extracted ridge and minutiae.

Among these four types of descriptors, texture and minutiae-based descriptors are known to provide good performance and a combination of texture and neighboring minutiae information can achieve higher accuracy . However, the length of the neighboring minutiae-based descriptor in is variable, depending on the number of neighboring minutiae. Computing the similarity between two variable-length minutiae descriptors is not very efficient.

CONCLUSION

Mining frequent itemset for association rule mining from large dataset is very difficult task. There are many approaches have been discussed; Apriori approach and FP tree approach for extracting frequent itemset, which have scope of improvement. The Improved algorithm name as PISPO based on character of evidence dataset, which need to update. This algorithm reduce number of tree branches but also update tree in real time. When new criminal record has been added to the dataset, it will be inserted according to descending order frequency. This algorithm used novel data and computation distribution scheme, which eliminates communication among computers virtually and makes it possible for us with Map Reduce model. Algorithm is effective when massive data scene should be mining. This algorithm is conceptually simple and resulting rules are clear and understandable.

REFERENCES

[1] X. Jiang and G. Sun, "MapReduce-based Frequent Itemset Mining for Analysis of Electronic Evidence," *IEEE Louisville Chapter*, p. 978, 2013.

[2] X. Wu, "Data Mining with Big Data,"*IEEE*, vol. 26, no. 1, pp. 97-107, January 2014.

[3] Haoyuan Li and Yi Wang, "PFP: Parallel FP-Growth for Query Recommendation".

[4] R. V. Abhilash, P. G. Rao and V. H. Bhat, "A data mining approach for data generation and analysis for digital forensic application," *IACSIT International Journal of Engineering and tech*, vol. 2, no. 3, pp. 313-319, 2010.

[5] G.K.Gupta, "Information Privacyh," in *Introduction to data mining with case studies* , Delhi, PHI, 2012, pp. 450-481.

[6] S. K. Tanbeer, C. F. Ahmed and B. S. Jeong, "CP Tree:A tree structure for single pass for single pass frequent pattern mining," in *Advance in Konwledge discovery and data mining*,vol. 5012, Springler, 2008, pp. 1022-1027.

[7] W. Yongwei, Y. Feng, Kang Chen and Weimin Zheng, "Modelling of Distributed File System for Practical Perf ormance Analysis," vol. 25, no. 1, pp. 156-166, Januar y 2014.

[8] J. R. Vacca, Computer Forensics, Boston, Massachusetts: CHARLES RIVER MEDIA, INC..

[9] C. Kai-Sang Leung and Q. I. Khan, "CanTree: a canonical-order tree for incremental frequent-pattern mining," *Springer-Verlag London*, 2006.

[10] Y. S. Koh and G. Dobbie, "SPO Tree:efficient single pass oerdered incremental pattern mining," in *Data warehousing and knowldege discovery and data mining* , Springler, 2011, pp. 265-276.

[11] http://docs.mongodb.org/manual/tutorial/map-reduce examples/ ,(Online)

[12] S. G. Totad, G. RB and P. P. Reddy, "Batch processing for incremental FP tree construction,"*International Journal of Computer Applications*, vol. 5, no. 5, pp. 27-32, 2010.

[13] H. Cummins and M. Midlo, Finger Prints, Palms and Soles: An Introduction to Dermatoglyphics. Dover Publications, 1961.

[14] P. Komarinski, Automated Fingerprint Identification Systems (AFIS). Academic Press, 2004.

[15] F. Galton, Fingerprints (reprint). Da Capo Press, 1965.

[16] J. Funada, N. Ohta, M. Mizoguchi, T. Temma, K. Nakanishi, A. Murai, T. Sugiuchi, T. Wakabayashi, and Y. Yamada, “Feature Extraction Method for Palmprint Considering Elimination of Creases,” Proc. 14th Int’l Conf. Pattern Recognition, pp. 1849-1854, 1998.

[17] A.K. Jain, J. Feng, A. Nagar, and K. Nandakumar, “On Matching Latent Fingerprints,” Proc. CVPR Workshop Biometrics,June 2008.

AUTHORS

First Author – Mayuresh A. Sonawane : Student of BE in Computer Engineering in SND Collage of Engineering and research Center, Babhulgaon, Yeola, Dist. Nashik

Second Author– Navnath S. Borade: Student of BE in Computer Engineering in SND Collage of Engineering and research Center, Babhulgaon, Yeola, Dist. Nashik.

Third Author – Manoj S. Jadhav: Student of BE in Computer Engineering in SND Collage of Engineering and research Center, Babhulgaon, Yeola, Dist. Nashik.

Fourth Author – Vikas A. Mandal: Student of BE in Computer Engineering in SND Collage of Engineering and research Center, Babhulgaon, Yeola, Dist. Nashik.