Collaborative-Frequent Itemset Mining of Big Data Using Mapreduce Framework

A.PADMAPRIYA, M.Phil.,
Research Scholar
Government Arts College
Salem – 636 007.
Tamilnadu.

R. VENKATACHALAM, M.Phil.,
Assistant professor
Government Arts College
Salem – 636 007
Tamilnadu.

Data uncertainty is inherent in many real-world applications such as environmental surveillance and mobile tracking. Mining sequential patterns from inaccurate data, such as those data arising from sensor readings and GPS trajectories, is important for discovering hidden knowledge in such applications. Frequent Item set Mining algorithms are aimed to disclose frequent item sets from transactional database but as the dataset size increases, it cannot be handled by traditional frequent item set mining. Map Reduce programming model solves the problem of large datasets but it has large communication cost which reduces execution efficiency. Implemented FIM algorithm based on MapReduce programming model. Kmeans clustering algorithm focuses on pre-processing, frequent itemsets of size k are mined using Apriori algorithm and discovered frequent itemsets are mined using Eclat algorithm. ClustBigFIM works on large datasets with increased execution efficiency using pre-processing. Experiments are done on transactional datasets, results shown that ClustBigFIM works on Big Data very efficiently and with higher speed. The existing system pre-processed k-means technique applied on Big FIM algorithm. Clust BigFIM uses hybrid approach, clustering using k means algorithm to generate Clusters from huge datasets and Apriori and Éclat to mine frequent item sets from generated clusters using Map Reduce programming model. Results shown that execution efficiency of Clust Big FIM algorithm is increased by applying k-means clustering algorithm before BigFIM algorithm as one of the pre-processing technique. In our proposed system Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). A user expresses his or her preferences by rating items (e.g. Bp, Sugar,Heart Attack) of the system.

KEY WORDS:

Scales linearly in the database size (SPADE) , Probabilistic Frequent Itemset Mining (PFIM), attribute-based encryption (ABE), Cipher text-Policy ABE (CP-ABE), Usage Control Colored Petri Nets (UCPN).

1. INTRODUCTION:

Data mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data. Data mining tools predict behaviors and future trends, allowing businesses to make proactive, knowledge-driven decisions. Data mining tools can answer business questions that traditionally were too time consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations. Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. Although data mining is a relatively new term, the technology is not. Companies have used powerful computers to sift through volumes of supermarket scanner data and analyze market research reports for years. However, continuous innovations in computer processing power, disk storage, and statistical software are dramatically increasing the accuracy of analysis while driving down the cost.In this existing system FIM algorithm based on Map Reduce programming model. K-means clustering algorithm focuses on pre-processing, frequent itemsets of size k are mined using Apriori algorithm and discovered frequent item sets are mined using Eclat algorithm. Clust BigFIM works on large datasets with increased execution efficiency using pre-processing.Here we proposed Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue *x* than to have the opinion on x of a person chosen randomly. For example, a collaborative filtering recommendation system for television tastes could make predictions about which television show a user should like given a partial list of that user's tastes (likes or dislikes).[3] Note that these predictions are specific to the user, but use information gleaned from many users.

2. RELATED WORK

PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth this paper author develop a novel sequential pattern mining method, called PrefixSpan (i.e., Prefix-projected Sequential pattern mining). Its general idea is to examine only the prefix subsequences and project only their corresponding postfix subsequences into projected databases. In each projected database, sequential patterns are grown by exploring only local frequent patterns. To further improve mining efficiency, two kinds of database projections are explored: level-by-level projection and bi-level projection. Moreover, a main-memory-based pseudo-projection technique is developed for saving the cost of projection and speeding up processing when the projected (sub)-database and its associated psuedo-projection processing structure can fit in main memory [1]. Their performance study shows that bi-level projection has better performance when the database is large, and pseudo-projection speeds up the processing substantially when the projected databases can fit in memory. Prefix Span mines the complete set of patterns and is efficient and runs considerably faster than both Apriori based GSP algorithm and Free Span. Among different variations of Prefix Span, bi-level projection has better performance at disk-based processing and psuedo-projection has the best performance when the projected sequence database can fit in main memory.

Limitations are GSP always searches in the original database. Many irrelevant sequences have to be scanned and checked, which adds to the unnecessarily heavy cost. FreeSpan cannot gain much from projections, where as Prefix Span can cut both the length and the number of sequences in projected databases dramatically.

SPADE: An Efficient Algorithm for Mining Frequent Sequences this paper author present SPADE, a new algorithm for fast discovery of Sequential Patterns. The existing solutions to this problem make repeated database scans, and use complex hash structures which have poor locality [2]. SPADE utilizes combinatorial properties to decompose the original problem into smaller sub-problems, that can be independently solved in main-memory using efficient lattice search techniques, and using simple join operations. All sequences are discovered in only three database scans. SPADE **(Sequential Pattern Discovery using Equivalence classes) algorithm**. Advantages are SPADE outperforms the best previous algorithm by a factor of two, and by an order of magnitude with some pre-processed data. It also has linear scalability with respect to the number of input-sequences, and a number of other database parameters [3]. SPADE not only minimizes I/O costs by reducing database scans, but also minimizes computational costs by using efficient search schemes. The vertical id-list based approach is also insensitive to data-skew.

An extensive set of experiments shows that SPADE outperforms previous approaches by a factor of two, and by an order of magnitude if they have some additional off-line information. Furthermore, SPADE scales linearly in the database size, and a number of other database parameters. Limitations are they observed that simple mining of frequent sequence produces an overwhelming number of patterns, many of them trivial or useless

Model-Driven Data Acquisition in Sensor Networks Author enriches interactive sensor querying with statistical modeling techniques. He demonstrates that such models can help provide answers that are both more meaningful and by introducing approximations with probabilistic confidences, significantly more efficient to compute in both time and energy. An exponential time algorithm is used.

Limitations are it describes an exponential time algorithm for finding the optimal solution to this optimization problem, and a polynomial-time heuristic for identifying solutions that perform well in practice Advantages are author evaluate an approach on several real-world sensor-network data sets, taking into account the real measured data and communication quality, demonstrating that our model-based approach provides a high-fidelity representation of the real phenomena and leads to significant performance gains versus traditional data acquisition techniques.

Finding Frequent Items in Probabilistic Data paper proposed a new definition based on the possible world semantics that has been widely adopted for many query types in uncertain data management, trying to find all the items that are likely to be frequent in a randomly generated possible world. This approach naturally leads to the study of ranking frequent items based on confidence as well. Exact and sampling algorithms is used. Limitations are the exact algorithms are costly and do not scale when data sets increase, although they are able to return exact results the exact algorithms are costly and do not scale when data sets increase, although they are able to return exact results Advantages are Efficient algorithms with theoretical guarantees have been presented for both offline and streaming data, under the widely adopted x-relation model.

Frequent Pattern Mining with Uncertain Data shows the hyper-structure and the candidate generate-and-test algorithms perform much better than tree-based algorithms [4]. This counter-intuitive behavior is an important observation from the perspective of algorithm design of the uncertain variation of the problem. Author test the approach on a number of real and synthetic data sets, and show the effectiveness of two of our approaches over competitive techniques. Limitations are UH-mine algorithm does so without using the FP-tree structure which does not extend well to the uncertain case. Advantages are the UH-mine algorithm proposed in this paper provides the best trade-offs both in terms of running time and memory usage.

Probabilistic Frequent Itemset Mining in Uncertain Databases this paper author introduce new probabilistic formulations of frequent itemsets based on possible world semantics. In this probabilistic context, an itemset X is called frequent if the probability that X occurs in at least minSup transactions is above a given threshold value . To the best of our knowledge, this is the first approach addressing this problem under possible worlds semantics. In consideration of the probabilistic formulations, author present a framework which is able to solve the ***Probabilistic Frequent Itemset Mining (PFIM)*** problem efficiently [5]. An extensive experimental evaluation investigates the impact of our proposed techniques and shows that our approach is orders of magnitude faster than straight-forward approaches. probabilistic frequent itemset mining (PFIM). Advantages are The Probabilistic Frequent Itemset Mining (PFIM) problem is to find itemset in an uncertain transaction database that are (highly) likely to be frequent [6]. To the best of our knowledge, this is the first paper addressing this problem under possible worlds semantics. Our proposed dynamic computation technique is able to compute the exact support probability distribution of an itemset in linear time w.r.t. the number of transactions instead of the exponential runtime of a non-dynamic computation.

Limitations are the consideration of existential uncertainty of item(sets), indicating the probability that an item(set) occurs in a transaction, makes traditional techniques inapplicable.

2.1 SYSTEM MODEL

Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue *x* than to have the opinion on x of a person chosen randomly. For example, a collaborative filtering recommendation system for television tastes could make predictions about which television show a user should like given a partial list of that user's tastes (likes or dislikes).[3] Note that these predictions are specific to the user, but use information gleaned from many users. The underlying assumption of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue *x* than to have the opinion on x of a person chosen randomly.

1. A user expresses his or her preferences by rating items (e.g. Bp, Sugar, Heart Attack) of the system. These ratings can be viewed as an approximate representation of the Patient Details interest in the corresponding domain.
2. The system matches this user's ratings against other users and finds the people with most "similar" Disease.
3. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

3. METHODOLOGY:

- **Patient Registration**
- **Doctor Registration**
- **Doctor Details**
- **Entry Details**
- **Fetching Data from database**
- **Collaborative filtering Method**
- **Map reduce Program**

3.1METHODOLOGY DESCRIPTIONS:

3.1.1 Patient Registration:

The patient registration details are mainly used to admin because the collect all the information about patient, they give individual password to every user and finally we can store the information in cloud database.

3.1.2 Doctor Registration:

The doctor registration details are mainly used to admin because the collect all the information about doctors and finally we can store the information in database.

3.1.3 Doctor Details:

The admin maintain the entire doctor’s information because any emergency of patient side we can easily find out the doctor’s information so doctors can fix appointment to particular patient in case of emergency.

3.1.4 Entry Details:

The admin maintain all the patient entry information because how many patient entry in hospital and they maintain accounts details of the hospital and we can finally store the information in database.

3.1.5 Fetching Data from database

Data Set for Hospital An outpatient (or out-patient) is a patient who is hospitalized for less than 24 hours. Even if the patient will not be formally admitted with a note as an outpatient, they are still registered, and the provider will usually give a note explaining the reason for the service, procedure, scan, or surgery, which should include the names and titles and IDs of the participating personnel, the patient's name and date of birth and ID and signature of informed consent, estimated pre- and post-service time for a history and exam (before and after), any anesthesia or medications needed, and estimated time of discharge absent any (further) complications. Treatment provided in this fashion is called ambulatory care. Sometimes surgery is performed without the need for a formal hospital admission or an overnight stay. This is called outpatient surgery. Outpatient surgery has many benefits, including reducing the amount of medication prescribed and using the physician's or surgeon's time more efficiently. More procedures are now being performed in a surgeon's office, termed office-based surgery, rather than in a hospital-based operating room. Outpatient surgery is suited best for healthy patients undergoing minor or intermediate procedures (limited urologic, ophthalmologic, or ear, nose, and throat procedures and procedures involving the extremities).

3.1.6 Collaborative filtering Method

Collaborative filtering is a method of making automatic predictions (e.g. Bp, Sugar, and Heart Attack) (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue *x* than to have the opinion on x of a person chosen randomly. For example, a collaborative filtering recommendation system for television tastes could make predictions about which television show a user should like given a partial list of that user's tastes (likes or dislikes). Note that these predictions are specific to the user, but use information gleaned from many users.

A user expresses his or her preferences by rating of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.

The system matches this user's ratings against other users' and finds the people with most "similar" tastes.

With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

3.1.7 Map reduce Program

The model is inspired by the map and reduce functions commonly used in functional programming, although their purpose in the Map Reduce framework is not the same as in their original forms, provided that all outputs of the map operation that share the same key are presented to the same reducer at the same time, or that the reduction function is associative. While this process can often appear inefficient compared to algorithms that are more sequential (because multiple rather than one instance of the reduction process must be run), Map Reduce can be applied to significantly larger datasets than "commodity" servers can handle – a large server farm can use Map Reduce to sort.

3.1.8 Patient Report

The Patient report is mainly maintained to admin because every patient having unique id so easily send to the mail so easily view report to patient side and finally we can take report by print out.

4. ALGORITHM DESCRIPTIONS

4.1 SYSTEM IMPLEMENTATION

4.1.1 Attribute Based Encryption (ABE)

Functional encryption presents a vision for public key cryptosystems that provide a strong combination of flexibility, efficiency, and security. In a functional encryption scheme, cipher texts are associated with descriptive values *x*, secret keys are associated with descriptive values *y*, and a function *f*(*x*, *y*) determines what a user with a key for value *y* should learn from a cipher text with value *x*. One well-studied example of functional encryption is **attribute-based encryption (ABE)**, first introduced, in which cipher texts and keys are associated with access policies over attributes and subsets of attributes. A key will decrypt a cipher text if and only if the associated set of attributes satisfies the associated access policy. There are two types of ABE systems: **Cipher text-Policy ABE (CP-ABE),** where cipher texts are associated with access policies and keys are associated with sets of attributes, **and Key-Policy ABE (KPABE)**, where keys are associated with access policies and cipher texts are associated with sets of attributes.

To achieve desired flexibility, one strives to construct ABE systems for suitably expressive types of access policies over many attributes. Current constructions allow Boolean formulas or linear secret sharing schemes as access policies. This high level of flexibility means that keys and cipher texts have rich structure, and there is a very large space of possible access policies and attribute sets. This presents a challenge to proving security, since a suitable notion of security in this setting must enforce collusion resistance, meaning that several users should not be able to decrypt a message that none of them are individually authorized to read. Hence a security proof must consider an attacker who can collect many different keys, just not a single one that is authorized to decrypt the cipher text.

Concert assesses the compliance of a workflow by analysing the five established elements required to check for rule adherence in workflows: activities, data, location, resources, and time limitation. A rule describes which activities may, must or must not be performed on what objects by which roles. In addition, a rule can further prescribe the order of activities, i.e. which activities have to happen before or after other activities. The formalization of rules as Petri nets patterns has been proposed by Catt et al. And Huang and Kirchner. In contrast to Catt et al., Huang and Kirchner cannot cope with the expression of usage control policies.
Catt et al. employ **Usage Control Colored Petri Nets (UCPN)** for the formalization and enforcement of diverse types of obligations, i.e. actions to be performed before, during and after an activity. However, their approach assumes that the rules are integrated into the workflow, so that UCPN cannot be singled out for reuse for other workflows. Acting as a security automata, rules in Concert are captured as Petri net patterns and are not integrated into the workflow. Together with the classification of compliance requirements, this makes it possible to organize compliance rules in categories according To their intent and semantics, thereby facilitating their formalization as Re-usable Petri net patterns or templates in other modular policy languages for usage control.

To allow fine-grained and scalable access control for PHRs control attribute based encryption (ABE) techniques to encrypt every patient's PHR data. Different from earlier works in protected data outsourcing center on the multiple data owner scenario and separate the user in the PHR system into multiple security domains that really decreases the key managing complexity for owners and users. In this way a high degree of patient privacy is assured concurrently by developing multi-authority ABE and EC-MAABE.

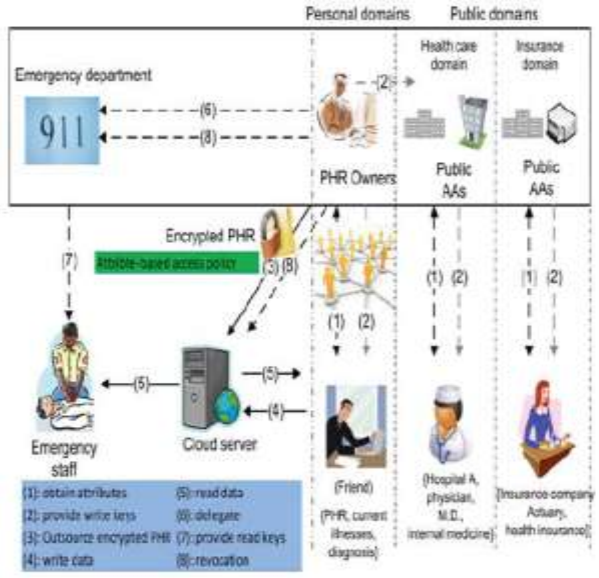

Fig 4.1.1 Architecture Diagram

4.1.2 FLAME ALGORITHM INTRODUCTION:

FLAMES algorithm

```
flames (s,m) {
    l ←| s |
    if l = 1 then
        Print s
        Return
    else
        p←  m mod l
        if p = 0 then
            p ←l
        end if
        Print $s_p$
        x←$pref_p(s)$
        y←$suff_p(s)$
        flames(yx;m)
    end if
}
```

4.1.2.1Algorithm explanation

The FLAME algorithm is mainly divided into three steps:

1. Extraction of the structure information from the dataset:
   - Construct a neighborhood graph to connect each object to its **K-Nearest Neighbors** (KNN);
   - Estimate a density for each object based on its proximities to its KNN;
   - Objects are classified into 3 types:
     - **Cluster Supporting Object (CSO)**: object with density higher than all its neighbors;
     - **Cluster Outliers:** object with density lower than all its neighbors, and lower than a predefined threshold;
     - the rest.
2. Local/Neighborhood approximation of fuzzy memberships:
   1. Initialization of fuzzy membership:
      - Each CSO is assigned with fixed and full membership to itself to represent one cluster;
      - All outliers are assigned with fixed and full membership to the outlier group;
      - The rest are assigned with equal memberships to all clusters and the outlier group;
   2. Then the fuzzy memberships of all type 3 objects are updated by a converging iterative procedure called Local/Neighborhood Approximation of Fuzzy Memberships, in which the fuzzy membership of each object is updated by a linear combination of the fuzzy memberships of its nearest neighbors.
3. Cluster construction from fuzzy memberships in two possible ways:
   1. One-to-one object-cluster assignment, to assign each object to the cluster in which it has the highest membership;
   2. One-to-multiple object-clusters assignment, to assign each object to the cluster in which it has a membership higher than a threshold.

5. CONCLUSION:

Here presented the difficult of mining frequent sequential patterns for the large uncertain database. In this paper problem of mining **probabilistically frequent sequential patterns (p-FSPs)** in uncertain databases. It is founded on two uncertain sequence data models that are fundamental to many real-life applications. A novel motif mining algorithm called **FLAME** that uses a concurrent traversal of two suffix trees to efficiently explore the space of all motifs. It is also accurate, as it always finds the pattern if it exists. Accordingly it gives a well-defined objective function which can be clearly solved in an iterative technique. Investigational results expression the effectiveness of the suggested process.

As DNA samples are taking as datasets to analyze data effectively with a novel motif mining algorithm called Flexible and Accurate **Motif detector (FLAME) technique** that uses a concurrent traversal of two suffix trees to efficiently explore the space of all motifs. It presents an algorithm that uses FLAME as a building block and can mine combinations of simple approximate motifs under relaxed constraints.

6. FUTURE WORK

The approach that takes in FLAME explores the space of all possible models. In order to carry out this exploration in an efficient way, this paper first construct two suffix trees: a suffix tree on the actual data set that contains counts in each node (called the data suffix tree), and a suffix tree on the set of all possible model strings (called the model suffix tree). To get effective and accurate motif detection in different way using another technique in future.

REFERENCES:

1. J. Pei *et al*., "PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc. 17th ICDE*, Berlin, Germany, 2001.
2. M. J. Zaki, "SPADE: An efficient algorithm for mining frequent sequenes," *Mach. Learn.*, vol. 42, no. 1–2, pp. 31–60, 2001.
3. A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. 13th Int. Conf. VLDB*, Toronto, ON, Canada, 2004.
4. Q. Zhang, F. Li, and K. Yi, "Finding frequent items in probabilistic data," in *Proc. ACM SIGMOD*, Vancouver, BC, Canada, 2008.
5. C. C. Aggarwal, Y. Li, J. Wang, and J. Wang, "Frequent pattern mining with uncertain data," in *Proc. 15th ACM SIGKDD*, Paris, France, 2009.
6. T. Bernecker, H. P. Kriegel, M. Renz, F. Verhein, and A. Zuefle, "Probabilistic
7. frequent itemset mining in uncertain databases," in *Proc. 15th ACM SIGKDD*, Paris, France, 2009.