ASIAN JOURNAL OF PHARMACEUTICAL AND CLINICAL RESEARCH

INNOVARE ACADEMIC SCIENCES
Knowledge to Innovation

Online - 2455-3891
Print - 0974-2441

Special Issue (April)

Full Proceeding Paper

BIG DATA MINING FOR INTERESTING PATTERNS WITH MAP REDUCE TECHNIQUE

NIKHIL JAMDAR, VIJAYALAKSHMI A*

School of Computer Science and Engineering, VIT University, Chennai, Tamil Nadu, India. Email: vijayalakshmi.av@vit.ac.in

Received: 23 January 2017, Revised and Accepted: 03 March 2017

ABSTRACT

There are many algorithms available in data mining to search interesting patterns from transactional databases of precise data. Frequent pattern mining is a technique to find the frequently occurred items in data mining. Most of the techniques used to find all the interesting patterns from a collection of precise data, where items occurred in each transaction are certainly known to the system. As well as in many real-time applications, users are interested in a tiny portion of large frequent patterns. Hence, the proposed user constrained mining approach and will help to find the frequent patterns in which user is interested. This approach will efficiently find user interested frequent patterns by applying user constraints on the collections of uncertain data. The user can specify their own interest in the form of constraints and uses the MapReduce model to find uncertain frequent pattern that satisfies the user-specified constraints.

Keywords: MapReduce model, Frequent pattern, Constraints, Uncertain data, Data search, Mining.

 DOI: http://dx.doi.org/10.22159/ajpcr.2017.v10s1.19634

INTRODUCTION

Many algorithms exist for mining huge amount of data using clustering, classification, anomaly detection, and association rule mining. Frequent pattern mining gives the result on the basis of frequently occurrence of data [1]. It finds the pattern and gives the result for the data which occurs frequently [2]. However, due to the high uncertainty of the data, frequent pattern data mining is difficult to analysis about the presence of the user required data [3].

For handling these types of the uncertainty data, the new data mining technique helps us to use constraint-based pattern mining. The constraint-based mining technique works on the user specified constraints [4]. The user can specifics about data which he/she need and with the help of that constraint pattern it gives the appropriate result. However, there is another problem with this technique, as the data are big, it is difficult to execute and it is a time-consuming process. Improving the efficiency and performance of the existing algorithms the MapReduce concept used in our approach [2]. The MapReduce technique will split the data into the small chunks and it will store into the separate files. These files are processed and executed separately in a parallel manner. Finally, the results are mapped and the result will be generated. So using the MapReduce technique the frequent items are generated with less execution time.

BACKGROUND

In these recent years, data usage has increased rapidly worldwide. So storing the large amount of data are a big problem [5]. The data generated from different data generating factors like machine logs, human generated data are being stored by companies. They use this data for their business improvement and for processing huge amount of data, so new tools and approaches have come into the role [3]. Most data analyst uses Hadoop for analysis of big data mining frequent patterns of itemsets present in transactional datasets, which helps for further data mining tasks such as association mining, correlations, and clustering [6].

LITERATURE SURVEY

Kharat and Gupta proposed an approach for finding frequent itemsets using reverse apriori algorithm [7]. In this paper, the reverse apriori algorithm works exactly, the reverse process of the apriori algorithm. Reverse apriori algorithm forming large frequent itemsets which contain all the combinations of items which occur more frequently in the transactions. For finding frequent patterns user have to provide their interest in terms of constraint and minimum support count. From large dataset, we reduce the combinations of itemset pairs by considering minimum support count and user specified constraints. With less number of comparisons, the user interested items were generated.

Online association rule mining described by Ari and Erdi [6]. In general, for finding the relationship between the items, we can use association rule mining. In this paper, the author proposed receptor system to find the relationship between the items for online data. The online data were created by many users in large amount it uses Hadoop system. Hadoop system was used to process the event processing language queries to find results based on user queries. Using the Hadoop system, the large amount of online data handled easily and the query was processed very fast.

Leung *et al.* proposed the approach for efficiently mine the frequent patterns from the huge uncertain data [8]. In this paper, the author elaborated the strategy of UF-growth algorithm to find the frequent pattern. The UF-growth algorithm was actually motivated from two basic techniques that are frequent-pattern growth algorithm and U-apriori algorithm [9]. This algorithm first scan database and calculate the expected support count for particular items. Arrange in descending order of expected support count. While doing this, it also checks for the minimum support count and user specified constraints. Once it was done with this algorithm again, it has to scan the database and construct the UF tree for finding the frequent pattern. The tree will grow according to the scanning of transactions.

Naik and Mankar proposed the approach for mining of frequent pattern itemsets present in uncertain database with the help of probabilistic support values [3]. Here, aprori algorithm used with the probabilistic support values for finding frequent patterns. Normally, apriori algorithm repeatedly produces the probabilistic frequent itemsets using a bottom-up strategy. Each iteration was performed by the following two steps. First join step, used for generating new candidates and next pruning step, used for calculating the probabilities frequentness of items and calculating the probabilistic frequent itemsets from the generated candidates.

Mining of frequent pattern from uncertain data was described by Tanbeer *et al.* [10] Some cases the decision tree will be useful for finding the frequent items. In this paper, author applied compact tree structure on uncertain database to extract frequent patterns. They proposed the technique of prefix capped uncertain frequent pattern (PUF) tree which will efficiently find the frequent itemset from uncertain database. This technique first used to find the prefix of the particular item set, with the help of these prefixes the compact tree was constructed. Applying the constraint of minimum support counts on the tree it converted into compact tree which satisfy minimum support count constraint. Finally by applying PUF compact tree based algorithm on it finds the frequent itemsets for given uncertain database.

PROPOSED SYSTEM

The proposed approach uses user-specified constraints and MapReduce technique for mining frequent patterns of data. This system used uncertain big data which finds the frequent patterns of user given constraints from huge amount of data [5]. As shown in Fig. 1, first consider the uncertain database. Then, apply user constraint on it such as minimum support count and constraint on attribute. Using MapReduce it finds valid single tone and valid non-single tone patterns from uncertain database. By considering both valid single tone and non-single tone items it finds the valid frequent pattern from uncertain data which satisfy user constraints.

It uses MapReduce function two times for mining process.

1. First, it is used for finding the singleton itemset.
2. Second, it is used for finding non-singleton itemset.

A finding valid singleton items from big data

After considering user constraint, system apply MapReduce function on database for finding the valid single tone pattern. For finding the valid single tone items, it only considers items which satisfy user constraints. It simply adds all the expected values of item from all transaction to find its support counts. Thus it finds the valid single tone item with its support count.

Finding valid non-singleton patterns

After finding the valid singletons from uncertain data which satisfies user constraint next step is to find non-singletons. For that, it finds projected database for each singleton items.

Algorithm for Projected database

For each transaction in sample data set do

    For each item from singletons in transaction do

Return (x, prefix of T_j ends with x).

In this algorithm x refers item, T_j represents transaction number. The output of the algorithm gives projected database of T_j. First calculate projected database and with the help of that we can calculate non single tone item with their user expected support count. For every item in projected database, we can build a tree for item (x) as (x - projected database). If support count is greater than minimum support count then return that item set else do not consider that itemsets. Thus with the help of these projected database of each item, it finds valid non single tone items with its support count.

Constraint check in first map function

If we do constraint check in the first reduce function then it requires less number of constraints. Because map function shuffle and sort all pairs. In opposite if we push constraint check in first map function, it checks each and every item of transactions. Hence, pushing constraint check in reduce function will be more time efficient.

In this approach, we push constraint check in the first map function. It will returns only items which satisfy user defined constraints. Hence, less number of bookkeeping is required and less number of pairs to be shuffled. Hence, it will be time efficient (less no of pairs need to shuffled) and space efficient too (less no of bookkeeping to be stored).

Succinct constraint

If collection of all patterns satisfying the constraints is succinct power set then it is succinct constraints.

Anti-monotone constraint

If all the subset of pattern which satisfies anti-monotone constraint also satisfies same constraint then it is called as anti-monotone constraint.

Mining frequent patterns by applying AM constraints

There are two types of patterns:

1. First one is satisfying both constraints succinct and anti-monotone.
2. And second one is which satisfy only anti-monotone constraint.

Here considering the rules of anti-monotonicity. As per the definition, if any of the frequent patterns is not satisfying anti-monotone constraint then its superset also does not satisfy anti-monotone. So with this, we can remove unwanted pattern. We use the property of anti-monotone and MapReduce approach to find the valid singletons with the first set of MapReduce function.

DISCUSSION AND RESULTS

The process discussed with an example of small uncertain data Table 1. In the example, user constraints are used as follows.

1. Minimum support count should be >0.5 and
2. Minimum wind speed should be >5 kmph from Table 2.

It returns itemset which has support count >0.5 and wind speed >5 kmph from Table 1.

From the information given in Table 1, to find the valid singleton patterns, it can understand that items a, c, d and f from item set satisfy constraint. It does not consider item e because it not satisfies constraint of minimum wind speed. First, map reduce function reads all transactions. For each item (a, [0.2, 0.5, 0.3, 0.9]), (c, [0.8, 0.9]), (d, [0.6, 0.6]), (e, [0.6, 0.5, 0.5]) and (f, [0.9, 0.6]) and returns the singleton items as result (a:1.9), (c:1.7), (d:1.2) and (f:1.5). These are valid singleton items with their corresponding support count. Here, the reduce function does not consider (e, 4 kmph), as well as (b:0.4) because they not satisfy with the users constraint.

For finding the non-singleton items map reduce function reads Transaction T1. Map function returns item and its prefix in T1 as ([c], [a:0.2, c:0.8]) and ([f], [a:0.2, c:0.8, f:0.9]). This function does not returns "a" because there is only one item for its prefix. After reading T2, it will returns ([c], [a:0.5, c:0.9]), for T3 ([d], [a:0.3, d:0.6]) and ([f], [a:0.3, d:0.6, f:0.6]). And at last for T4 it will give ([d], [a:0.9, d:0.6]). Now these returned pairs are shuffled and tree is built on it with item's support count. According to the tree second MapReduce function

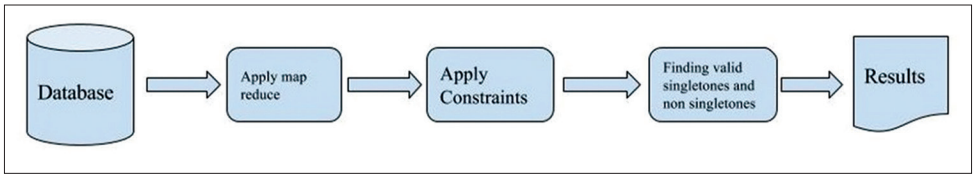

Fig. 1: Proposed system architecture

Table 1: Small uncertain data set

Tid	Content
T1	(a:0.2, b:0.2, c:0.8, f:0.9)
T2	(a:0.5, c:0.9, e:0.6)
T3	(a:0.3, d:0.6, e:0.5, f:0.6)
T4	(a:0.9, b:0.2, d:0.6, e:0.5)

Table 2: Description of data

Item	Wind speed (kmph)
a	8
b	9
c	15
d	12
e	4
f	10

returns non singletons and their support counts. Non singletons itemsets will be ([a, c]: 0.61), ([a, d]: 0.72), ([a, c, f]: 0.72), ([a, d, f]: 0.36).

Example: Suppose we need to calculate expected support count for X=(a, c) in Transaction T1 then Support count (X)=P(c,T1)*max{P(a.T1)}=0.2*0.8=0.16. Like this calculate for all transactions where (a, b) occurs, in our case in T2 we get 0.45 so if we add both we get expected support count for X = (a, b) = 0.61.

We mine the same a tiny sample set of an uncertain Big database as shown in Table 1 with minsup = 0.5 and wind speed = min (x - wind speed) ≥5 kmph. Again, based on the auxiliary information, we learn that domain items a, c, and f (but not b) satisfy user constraint. Then, for the first Transaction T1, the map function outputs (a:1.9), (c:1.7), (d:1.2) and (f:1.5).

(a, [0.2, 0.5, 0.3, 0.9]), (c, [0.8, 0.9]), (d, [0.6, 0.6]), (e, [0.6, 0.5, 0.5]) and (f, [0.9, 0.6]) as valid singletons and their corresponding expected support values.

Pushing constraint checks into the map function is both space-efficient (due to the reduction in the number of pairs returned by the map function) and time-efficient (due to the reduction in the number of pairs to be shuffled and sorted).

CONCLUSION

Mining frequent data from large amount of data are a big challenge in recent trends. This paper deals with the big data analytics which returns a user interesting patterns from huge amount of uncertain data. This approach helps to mine uncertain data using user constraints. Usually, mining the large amount of data is a time-consuming process. Hence, the map reduce helps us for finding user interested patterns more efficiently with respect to time and space.

REFERENCES

1. Cuzzocreaa A, Leungb CK, MacKinnonb RK. Approximation to expected support of frequent itemsets in mining probabilistic sets of uncertain data. Procedia Comput Sci 2015;60:613-22.
2. Kumari D, Patil LH, Thakur UK. Big data mining for interesting patterns from uncertain data using map reduce technique. Int J Adv Res Comput Sci Softw Eng 2016;6(6):241-4.
3. Naik RR, Mankar JR. Mining frequent Item sets from uncertain databases using probabilistic support. Int J Emerg Trends Technol Comput Sci 2013;2(2):432-6.
4. Kharat S, Gupta N. Result analysis of mining fast frequent itemset using compacted data. Int J Inf Sci Tech IJIST 2014;4(1/2):13-21.
5. Leung CK, MacKinnon RK, Jiang F. Reducing the search space for big data mining for interesting patterns from uncertain data. IEEE International Congress on Big Data; 2014.
6. Ari I, Erdi O. Online association rule mining over fast data. IEEE International Congress on Big Data; 2013.
7. Tanbeer SK, Leung CK. PUF-tree: A compact tree structure for frequent pattern mining of uncertain data. Pac Asia Conf Knowl Discov Data Min LNCS 2013;7818:13-25.
8. Leung CK, Carmichael CL, Hao B. Efficient mining of frequent patterns from uncertain data. In: Seventh IEEE International Conference on Data Mining; 2007.
9. Leung CK, Jiang F. Frequent pattern mining from time-fading streams of uncertain data. Data Warehousing Knowl Discov LNCS 2011;6862:252-64.
10. Tanbeer SK, Leung CK, MacKinnon RK. Tightening upper bounds to the expected support for uncertain frequent pattern mining. Procedia Comput Sci 2014;35:328-37.