IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 4 Issue 10, October 2017
ISSN (Online) 2348 – 7968 | Impact Factor (2016) – 5.264
www.ijiset.com

For Text Data Mining an Improved FP-Tree Algorithm

Amit Kumar Mishra , Deependra Kumar Jha

Asst. Professor , Department of Computer Science & Engineering, W.I.T.,Darbhanga, Bihar.

Abstract

The goal of data mining is to extract or mine" knowledge from large amount of data. One of the newest areas of data mining is text mining. Text Mining can be defined as the nontrivial extraction of implicit, previously unknown, and potentially useful information from textual data. However, in Text Mining, patterns are extracted from natural language text rather than databases. Text mining refers to a collection of methods used to find patterns and create text data intelligence from the data. The patterns that can be discovered may be descriptive that describe the general properties of the existing data, and predictive that attempt to do predictions based on inference on available data. There are several technique are used for finding frequent pattern.

Among the present existing techniques, the frequent pattern growth (FP-growth) technique is the most effective and scalable approach. We tend to propose an improved technique that extracting association rules from text data knowledge without any preprocessing or post processing. We propose improved algorithm, for mining the entire set of frequent patterns by pattern fragment growth. First Frequent Pattern-tree based mining adopts a pattern fragment growth technique to avoid the costly generation of a large number of candidate sets and a partition-based, divide-and-conquer technique is used.

Keywords: *Data mining, Text Mining, Association mining, FP-Growth*

I. INTRODUCTION

Data mining[1], that is also referred to as knowledge discovery in databases, means that a method of nontrivial extraction of implicit, which was previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in databases. Other terms for data mining are knowledge mining from databases, knowledge extraction, data archaeology, data dredging, data analysis, etc. By data discovery in databases, regularities, interesting knowledge, or high-level information can be extracted from the relevant sets of data in databases and be investigated from different angles, and large databases thereby serve as rich and reliable sources for knowledge generation and verification. Mining information and knowledge from huge databases has been recognized by many researchers as a key research topic in database systems and machine learning [1] and by many industrial companies as an important area with an opportunity of major revenues. The discovered knowledge [2] can be applied to information management, query processing, decision making, process control, and many other applications. Researchers in many different fields, including database systems, knowledge-base systems, artificial intelligence, machine learning, knowledge acquisition, statistics, and spatial databases have shown great interest in data mining.

One of the latest areas of data mining is text mining. Text Mining may be defined as the nontrivial extraction of implicit, previously unknown, and potentially useful information from textual data. However, in Text Mining, patterns are extracted from natural language text rather than databases. Text mining refers to a collection of techniques used to find patterns and generate text data intelligence from the data. The most common use of text mining procedure is in search engine technology. A user types in word or phrase, which can include misspellings, and the search engine searches through a vast repository of documents to find the most relevant documents**.** The patterns that can be discovered may be descriptive that describe the general properties of the existing data, and predictive that attempt to do predictions based on inference on available data. Text mining allows the discovery of knowledge potentially useful and unknown. Whether the knowledge discovered is interesting or not, is very subjective and depends upon the user and the application. The user can put a measurement or constraints on the patterns so that the patterns satisfying the constraints will be considered.

1.1 Text Processing.

Before mining the text data, it undergoes certain preprocessing stages:

Any kind of data on which mining is to be done is converted into text format.

Term extraction: in this process the entire text is split into a set of tokens. A blank space may be used as a delimiter.

Stop words removal: Certain words occur very frequently in text data. Examples are "the", "a", ”of”, etc. These words are referred to as "stop words" [3]. The stop words are words removed from the text document because they have no meaningful information.

Stemming: it means identifying a word by its root, as for example words like technology and technologies have the same root technology.

2. STAGES OF TEXT MINING PROCESS

Text mining involves the techniques from areas like an information retrieval, natural language processing, information extraction and data mining. These different stages of a text-mining process can be combined together in a single workflow.

2.1. Information Retrieval (IR)

Systems discover the documents in a collection which matches the user’s query. The most well known IR systems are search engines such as Google, which identify those documents on the World Wide Web that are relevant to a set of given words. IR systems are also used in libraries, where the documents are not only the books themselves but digital records containing information about the books. IR systems allow us to narrow down the set of documents that are relevant to a particular problem.

As text mining involves in applying very computationally-intensive algorithms to large document collections, IR can speed up the analysis considerably by reducing the number of documents for analysis.

2.2. Natural Language Processing (NLP)

This is the oldest and most difficult problems in the field of artificial intelligence. It is the analysis of human language so that the computers can understand natural languages as humans. As this goal is still some way off, NLP can do some types of analysis with a high degree of success. Shallow parsers recognize only the major grammatical elements in a sentence, such as noun phrases and verb phrases, whereas deep parsers generate a whole representation of the grammatical structure of a sentence. The role of NLP in text mining is to present the systems in the information extraction phase with linguistic data which they need to perform their task. As this is done by annotating documents with information like sentence boundaries, part-of-speech tags, parsing results, which can then be read by the information extraction tools.

2.3. Data Mining (DM)

This is the process of identifying patterns in large sets of data.

Our goal is to expose previously unknown, useful knowledge. When used in text mining, DM is applied to the facts generated by the information extraction phase. We place the results of our DM process in another database which can be queried by the end-user via a appropriate graphical interface. The data generated by such queries can also be represented visually.

2.4. Information Extraction (IE)

This is the process of automatically getting structured data from an unstructured natural language document. Often this involves defining the general form of the information that we are interested in as one or more templates, which are then used to guide the extraction process. IE systems rely mainly on the data generated by NLP systems.

3. Techniques for finding frequent patterns

Repeated patterns are patterns that occur repeatedly in data. repeated patterns help in mining associations, correlations, and many other interesting relationships among data. Moreover, it helps in data classification, clustering, and other data mining tasks as well.

Data mining has been used in a broad range of applications [2]. Most leading-edge organizations realize that data mining provide them the ability to reach their goals into customer relationship management, risk management, fraud and abuse detection, and e-business[2] etc.

Various data mining techniques [4] such as, decision trees, association rules [5], and neural networks are already proposed and become the point of attention for several years. Association rule mining technique is the effective data mining technique which discover hidden or desired pattern among the large amount of data [6]. It is responsible to find correlation relationships among different data attributes in a large set of items in a database.

In the year 2000, Han et al[7] projected the FP-growth algorithm—which was the first pattern-growth concept algorithm. FP-growth algorithm constructs an FP-tree structure and mines repeated patterns by traversing the constructed FP tree. The FP-tree structure is an extended prefix-tree structure involving crucial condensed information of repeated patterns. FP-tree structure: The FP-tree structure has plenty of information to mine complete repeated patterns. It has a prefix tree of frequent 1-itemset and a frequent-item header table. Each node in the prefix-tree has three fields: item-name, count, and node-link.

In the existing techniques, the repeated pattern growth (FP-growth) method is the most proficient and scalable approach. We propose a better technique that extracting association rules from text documents without any preprocessing or post processing. We proposed an algorithm for mining the complete set of repeated patterns with pattern fragment growth. The first repeated Pattern-tree based mining adopts a pattern fragment growth method to avoid the costly generation of a large number of candidate sets and a partition-based, divide-and-conquer method is used. We proposed an association data mining tool for text data mining. This will increase the mining efficiency and also takes less memory.

RELATED WORK

Algorithms which mine association rules [8] from relational data have well developed. Numerous query languages have been proposed, to assist association rule mining. The topic of mining text data has received little attention, as the data mining community has focused on the development of techniques for extracting common structure from text data. For example, [3] has projected an algorithm to construct a frequent tree by finding common sub trees which has been embedded in the text data.

[9] Proposed a new and improved FP tree with a table and a new algorithm for mining association rules. The author projected an efficient association rule mining technique with the help of improved frequent pattern tree (FP-tree) and a mining frequent item set (MFI) algorithm. This algorithm mines all possible frequent item set without generating the conditional FP tree. It also gives the frequency of the frequent items, which is being used to estimate the desired association rules.

Algorithms for mining association rules from relational data have been implemented since long before. Association rule mining was first introduced at 1993 by R. Agrawal, T. Imielinski, and A. Swami [10]. The Apriori algorithm [11] uses the bottom-up breadth-first approach to discover the large item set. As it was proposed to grip the relational data this algorithm cannot be applied directly to mine complex data. Another well-known algorithm is FP growth algorithm. It adopts divide-and-conquer approach. Primarily it computes the repeated items and characterizes the repeated items in a tree called frequent-pattern tree. This tree can also utilize as a compressed database. The association rule mining is performed on the compressed database with the help of this FP tree. This indicates that the dataset needs to be inspecting once. These algorithms do not need the candidate item set generation. So, in comparison with Apriori algorithm, it is much better in terms of efficiency [10] .But like other algorithms it also have certain disadvantages, it generates a large number of conditional FP trees. This generates the FP trees recursively as a procedure of mining. That's why the efficiency of the FP growth algorithm is not reasonable. But in proposed improved FP tree and MFI algorithm no need to generate conditional FP tree[11] because the recursive element is separately stored in a different table. This reduces the existing bottleneck of the FP growth algorithm.

Plenty of modified algorithm and technique has been proposed by different authors. Such as FP- tree and COFI based approach is proposed for multilevel association rules. Here apart from the FP tree, a new type of tree called COFI- tree is proposed [7] .An Apriori based data mining method is described at [3] .We use that example as the input of our proposed MFI algorithm and it is easily understandable that the new approach collect the association rule more efficiently.

In the year 2000, Han et al[7] projected the FP-growth algorithm—which was the first pattern-growth concept algorithm. FP-growth algorithm constructs an FP-tree structure and mines repeated patterns by traversing the constructed FP tree. The FP-tree structure is an extended prefix-tree structure involving crucial condensed information of repeated patterns. FP-tree structure: The FP-tree structure has plenty of information to mine complete repeated patterns. It has a prefix tree of frequent 1-itemset and a frequent-item header table. Each node in the prefix-tree has three fields: item-name, count, and node-link.

Construction of FP-tree: The FP-growth has to scan the TDB *(*Transactional Database*)* two times to construct an FP-tree. The first scan of TDB retrieves a set of repeated items from the TDB. Then, the retrieved repeated items are ordered by descending order of their supports. The ordered list is called an F-list. In the second scan, a tree *T* whose root node *R* labeled with "null" is created. Then, these steps are applied to every transaction in the TDB. Here, let a transaction represent [p\P] where *p* is the first item of the transaction and *P* is the remaining items.

In each transaction, infrequent items are discarded.

Then, only the repeated items are sorted by the same order of F-list.

Advantages and Disadvantages

This method is advantageous because, it doesn't generate any candidate items. It is disadvantageous because, it suffers from the issues of special and temporal locality issues.

(A) **FP-Tree (Frequent Pattern Tree)**

A tree structure [12] in which all items are arranged in descending order of their frequency or support count. After constructing the tree, the frequent items can be mined using FP-growth.

(a) Creation of FP-Tree

First Iteration: Take a transactional database which contains a set of transactions along with their transaction id and list of items in the transaction. Then scan the entire database. Collect the count of the items present in the database. Then sort the items in decreasing order based on their frequencies (no. of occurrences).

(b)Second Iteration

Now, once again scan the transactional database. The FP-tree is constructed as follows. Start with an empty root node. Add the transactions one after another as prefix sub trees of the root node. Repeat this process until all the transactions have been included in the FP-tree. Then construct a header table which consists of the items, counts and their head-of-node links.

(c)Finding Frequent Patterns from FP-Tree

Once the FP-tree is constructed, the repeated patterns can be mined using an iterative approach FP-growth. This approach looks up the header table and selects the items that support the minimum support. It removes the infrequent items from the prefix-path of an existing node and the remaining items are considered as the frequent item sets of the specified item.

Advantages and Disadvantages

This method is advantageous because, it doesn't generate any candidate items. It is disadvantageous because, it suffers from the issues of special and temporal locality issues.

III OUR APPROACH

Proposed Improved Algorithm

(a) FP-tree structure

The FP-tree structure has sufficient info to mine complete frequent patterns. It consists of a prefix tree of frequent 1-itemset and a frequent-item header table. Every node within the prefix-tree has 3 fields: *item-name*, *count*, and *node-link*.

- *item-name* is that the name of the item.
- *count* is that the number of transactions that incorporates the frequent 1-items on the path from root to the present node.
- *node-link* is that the link to the subsequently same item name node within the FP-tree. Every entry within the frequent-item header table has 2 fields: *item-name* and *head of node-link*.
- *head of node-link* is that the link to the primary same item-name node within the prefix-tree.

(b) Construction of FP-tree

FP-growth [14] has to scan the *TDB* twice to construct an FP-tree. The first scan of *TDB* retrieves a set of frequent items from the *TDB*. Then, the retrieved frequent items are ordered by descending order of their supports. The ordered list is called an F-list. In the second scan, a tree *T* whose root node *R* labeled with "null" is created. Then, these steps are applied to each and every transaction in the *TDB*. Here, let a transaction represent [p\P] where *p* is the first item of the transaction and *P* is the remaining items.

The function insert tree (p\P, R) appends a transaction [p\P] to the root node *R* of the tree *T*. Pseudo code of the function insert tree (p\P, R) is shown.

```
 Input: Transactional database
Output: Efficient FP Tree
Create the root of tree R
Initially R=NULL
-In each transaction, infrequent items are discarded.
-Then, only the frequent items are sorted by the same order of F-list. Let N be a direct child node of R, such that N's item-name = p's item-name.
if ( R has a direct child node N ) { increment N 's count by 1.
}
else{
create a new node M linked under the R .
set M 's item-name equal to p .
set M 's count equal to 1.
}
Recursively call insert_tree ( P ,N).
}

for each item i in FP-tree where (i! = R) do
        if supprt is equal to frequency of item then
                frequency of item set = S
        generate item set P with the frequency of item set
else if support is greater than frequency then
        frequent item=frequency + count
else
        Generate item set all possible combination of item and node in FP Tree.
End for
End
```

CONCLUSION AND FUTURE WORK

Data mining involves the employment of refined data analysis tools to find previously unknown, valid patterns and relationships in huge data sets. In recent years, text data mining is a used widely for mining text documents. In future we can do a lot of improvement in our projected algorithm so it can forecast pattern easily.

References

[1] Arun K Pujai "Data Mining techniques". University Press (India) Pvt. Ltd. 2001

[2] J. Han and M. Kamber.Data Mining: Concepts and Techniques. Morgan Kaufman, San Francisco, CA,2001.

[3] Qin Ding and gnanasekaran Sundaraj, " Association rule mining from XML data", Proceedings of the conference on data mining.DMIN'06

[4]Jayalakshmi.S, Dr k. Nageswara Rao, "Mining Association rules for Large Transactions using New Support and Confidence Measures", Journal of Theoretical and applied Information Technology, 2005.

[5] R Srikant, Qouc Vu and R Agrrawal. "Mining Association Rules with Item Constrains". IBM Research Centre, San Jose, CA 95120, USA.

[6] Ashok Savasere, E. Omiecins ki and Shamkant Navathe"An Efficient Algorithm for Mining Association Rules in a Large Databases". Proceedingsof the 21st VLDB conference Zurich, Swizerland,1995.

[7] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation", Proceedings of the ACM SIGMOD, Dallas, TX, May 2000, pp. 1-12.

[8] C. Silverstein, S. Brin, and R. Generalizing Association Rules to Dependence Rules," Data Mining and Knowledge Discovery, 2(1), 1998, pp 39–68

[9] A.B.M.Rezbaul Islam, Tae-Sun Chung, An Improved Frequent Pattern Tree Based Association Rule Mining Technique, 2011 IEEE, pp- 978-985

[10] R. Agrawal, T. Imielinski, and A. Swami.. "Mining association rules between sets of items in large databases". In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, May 26-281993.

[11] Qihua Lan,Defu Zhang, Bo Wo, , "A new algorithm for frequent itemset mining based on apriori and FP-tree", Global Congres on Intelligent System,2009.

[12] Muthaimenul Adnan and Reda Alhajj, "A Bounded and Adaptive Memory-Based Approach to Mine Frequent Patterns From Very Large Databases" IEEE Transactions on Systems Management and Cybernetics- Vol.41,No. 1,February 2011

[13] W. Cheung and O. R. Zaiane, "Incremental mining of frequent patterns without candidate generation or support constraint," in Proc. IEEE Int.Conf. Database Eng. Appl., Los Alamitos, CA, 2003, pp. 111–116.

[14] J. S. Park, M.-S. Chen, and P. S. Yu, "An effective Hash-Based Algorithm for Mining Association Rules",Proceedings of the ACM SIGMOD, San Jose, CA, May1995, pp. 175-186.

[15] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data", Proceedings of the ACM SIGMOD, Tucson, AZ, May 1997, pp. 255-264.