Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 50 (2015) 401 – 407

Procedia Computer Science

2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

An Efficient Association Rule Based Clustering of XML Documents

A Muralidhar[a], V Pattabiraman[b]

[a]VIT Unversity, Chennai Campus,Chennai-600127, India
[b]VIT Unversity, Chennai Campus,Chennai-600127, India

Abstract

Mining the web data is one of the emerging researches in data mining. The HTML can be used for maintaining the web data but it is hard to achieve the accurate web mining results from HTML documents. The XML documents make more convenient for finding the properties in web mining. Association rule based mining discovers the temporal associations among XML documents. But this kind of data mining is not sufficient to retrieve the properties of every XML document. Finding the properties for set of similar documents is better idea rather than to find the property of a single document. Hence, the key contribution of the work is to find the meaningful clustered based associations by association rule based clustering. Therefore, this paper proposes a hybrid approach which discovers the frequent XML documents by association rule mining and then find the clustering of XML documents by classical k-means algorithm. The proposed approach was tested with real data of Wikipedia. The comparative study and result analysis are discussed in the paper for knowing the importance of the proposed work.

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)

Keywords : - K-means clustering, Association Rule Mining ,XML documents, Web mining

1. Introduction

The web is in the form of rich information. Suppose, the information in the web is not well structured, it is very difficult to obtain the useful knowledge from the web. The successful extensible Markup Language (XML) represents structured data. The XML data is more readable and mining the data from various web is also feasible. Some tools for mining information from web data is still in growing state. As described in [15], the volume of XML data is growing rapidly, and it causes a need for the languages and specific tools to manage the XML documents, as well as to mine the data patterns from them. There are specific developments available like Xylem [15] which is a big storage space, that integrates the XML data from various sources of web.

The XML is widely used for data representation, storage, and exchange the data across sites. It has surveyed the techniques and approaches for mining the patterns from various XML documents. Learning the mined data, the hierarchical representation of the information, and the relationships between elements are very important steps in web mining. The mining process extracts both the structure and the contents from XML documents. *Mining of structure*, which is essential for mining the XML data, defines the intra-structure mining (mining the structure

Peer-review under responsibility of scientific committee of 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15)
doi:10.1016/j.procs.2015.04.024

inside an XML document) and inter-structure mining (mining the structures between XML documents). *Mining the content* involves on content analysis and also structure clarification .The Content analysis is concerned with analyzing texts within the XML document. The Structural clarification is concerned with extracting frequent XML documents based on their content.

The *fixed* XML documents do not change their content and its structure over the time. For understanding, an XML document containing the details of papers presented at a conference is always a static document. The *Strong* XML documents can change both the structure and content over the time. For example, the content for bookshop has always represented in XML structure, but it would change the information based on daily e-customer behavior.

The classification of the research areas of web mining are content mining, structure mining, and usage mining. The respective web mining classifications are shown in Fig. 1. The Proposed work is concentrating on content mining. This method focused on categorization or clustering of the XML documents based on content. In Big Data, such as Wikipedia, need to filter out the XML data based on the information of whether the XML file is frequent or not. In this connection, use Apriori algorithm to filter out the frequent XML documents.

The proposed algorithm uses the techniques as follows: association rule mining algorithm and classical k-means. Hence, it follows the hybrid approach called as association rule based k-means algorithm, which performs the clustering of frequent XML documents. This method would produce the faster execution of XML clustering results than present method.

The strong association rules are derived from the frequent XML documents. In Wikipedia datasets [16], the size of data is huge; it may be 1GB or more. Therefore, content and structures of these huge data is also increased exponentially. The computation time for finding the similarity features is unexpected in Big Data. Due to increase in execution time, initially remove the infrequent XML documents by Apriori [9] and then process the frequent XML documents by the classical k-means algorithm. The validity of results are tested by the Dunn's Index in the experimental study.

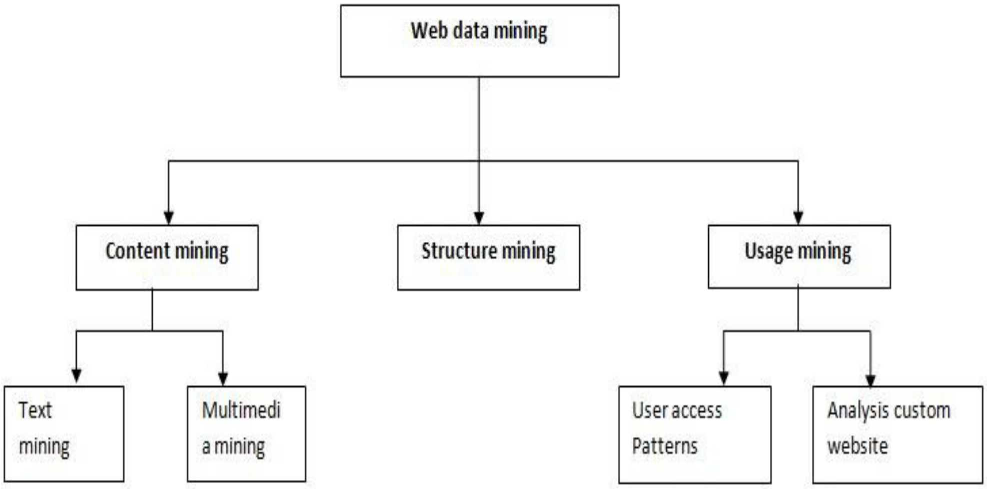

Figure 1: Web Mining Classifications

Frequent pattern mining is commonly used for finding the frequent patterns among the XML patterns in a given XML data set. Apriori runs too slow because each transaction consists of density of each pattern. After analyzing, it is concluded that Apriori algorithm takes more time in mining frequent patterns from XML hierarchical datasets. When it moves from top to bottom in a hierarchy, more detailed, simplified, specific, and less rules are always be generated.

Outline of the paper is described as follows: Section 2 presents the related work, Section 3 describes the proposed methodology, Section 4 discussed the datasets and experimental results and Section 5 presents the conclusion.

2. Related Work

Algorithms for mining association rules from relational data have been considerably trained. Several database query languages have been offered, to assist association rule mining. The matter of mining XML data has got slight attention, as the data mining has centered on the development of techniques for extracting common structure of various XML data.

For example, [4] has suggested an algorithm to make a frequent tree by finding common sub trees embedded in various XML data. On the other hand, some researchers focus on producing a standard model to interpret the patterns derived from the data using XML. For instance, the Predictive Model Markup Language (PMML) is an XML-based nomenclature, which furnishes a means for applications to define statistical and data mining models and to share models between PMML compliant applications.

2.1 Clustering

2.2 Association Rules

This section describes the basic concepts of association rule mining. Association rule mining was first introduced by Agarwal, and was used for market basket analysis. The problem of mining association rules can be described as follows: There is an atmost I = i1, i2…. in, where I is a set of n distinct items, and a set of transactions D, where each transaction T is a set of items such that, Table 1 yields a lesson where a database D contains a set of transactions T.

Table 1: An Example Database

Tid	Items
1	{bread,butter,milk}
2	{bread, butter, milk, ice_cream}
3	{ice_cream, coke}
4	{battery, bread, butter, milk}
5	{bread,butter, milk}
6	{battery,ice_cream, bread, butter}

An association rule is an implication of the form X → Y, where X, Y are in I and the rule X→ Y has supported in the transaction set D if s%of transactions in D contain X & Y. The support for a rule is defined as support(X/Y). The rule X→ Y holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y. The confidence of a rule is defined as support (X /Y) /support (X). For instance, look at the database in Table 1. When people buy bread and butter, they also buy milk at 66% of the cases and 80% of the transactions with bread and butter also contains milk. All the rules generated by the algorithm may not useful and the number of rules generated may be tremendous. Therefore, the task of mining association rules is to generate all association rules that have support and confidence greater than the user-defined minimum support (minsup) and minimum confidence (minconf) respectively.

An itemset with minimum documentation is predicted the large (or frequent) itemset. The rule X→Y is a solid pattern of X /Y is in the large itemset and its trust is greater than or equal to minimum confidence. The Apriori algorithm generates the frequent XML patterns. Apriori algorithm doesn't classify the XML documents. For more analysis of properties of XML documents, need to cluster the XML documents. We discuss the details in the following section.

3. Proposed Method

In exsiting k-means ,the collected XML documents are placed into a respective clusters for identifying of similar XML documents. Extracting and clustering of useful or frequently used XML documents takes more time. Therefore, we use apriori to acquire the frequent (or useful) documents before using k-means. All the infrequent documents are ignored in our methodology since apriori has produces only the frequent doucments. Hence, the proposed method is a hybrid technique, because it utitilizes the techniques of Parallel Apriori and K-means clustering (PAK) approach. Parallel Apriori is a distributed computing technique. It filterouts the frequent XML documents from huge amount of XML datasets. The present K-means simply performs the clustering of various XML documents by comparing the content and structures and it is difficult to execute for dataset like Wiki . Huge number of XML documents are availbale in wiki datasets. Hence it always takes the frequent XML documents for further clustering of the present PAK algorithm. Therefore , implement Parallel Apriori before clustering of XML documents.

Algorthm : PAK(Parallel Apriori and K-means clustering)

```
1. Use the Aprori by Map Reduce
   a. Split the data as data1,data2…..
   b. Find frequent items and frequency by Apriori of data1,data2,….
   c. Find global freqent items and frequency across data1,data 2,… by Map Reduce
2. Detect the freqent XML documents from step 1c
3. Extract the contents and structures detected from Step 2 documents.
4. Find similarity features between the frequent XML documets
   a. Use Euclidean distance metric and to find distance between documents
   b. Construct Euclidean based dissimialrity matrix D
5. Use Dunn's Index and find the k value
   a. Construct Graph G for D
   b. Construct MST for Graph by Prim's Algorithm
   c. Remove largest weight edge from MST and create two subtrees (i.e two clusters). Find Dunn's Index for the resulting clusters. Repeat the same step until predicted iterations
   d. Comapre the Dunn's Index, choose the maximum value and also dectect the k value.
6. Use k-means and detect the XML documents clustering results
```

In Step 1, it uses the Hadoop technology, and run Apriori algorithm in a way of parallel approach by the technique of map reduce. Step 2 retrieves the frequent XML documents. The contents and structure for every frequent XML documents is extracted by the mechanism of xpath . The similarity features between frequent XML documets are extarcted using a distnace metric , this is described in Step 4. Wrong k value may attemtps wrong clustering results. So, the right k-value is derived from Dunn's Index in Step 5. Finally, implements the classical k-means algorithm for deriving of XML documents clustering results.

4. Datasets and Experimental Results

In this experiment, use the Wikipedia dataset [16] to find the experimental results of PAK. The description of wiki datasets are presented in the following Table 2. The frequent XML documents are generated by Parallel Apriori.

Table 2: Description of XML Datasets

No. of XML Documents(frequent)	Generated Content files	Generated Structured files
884(wiki)	884	884
600(ebay)	600	600
442(ubid)	442	442

The simulated results on the Wiki datasets for 10 XML documents and the dissimialrity matrix of these 10 documents are decribed as follows:

Table 3 : Dissimilarity Matrix for sample wiki (10 XML Documents) .

	A	B	C	D	E	F	G	H	I
A	0	0.2799	0.353	0.3336	0.8223	0.3696	0.2747	0.6523	0.2472
B	0.2799	0	0.3584	0.3421	0.827	0.3823	0.2799	0.6618	0.2622
C	0.353	0.3584	0	0.4053	0.8455	0.4376	0.3517	0.689	0.3421
D	0.3336	0.3421	0.4053	0	0.8336	0.4117	0.3321	0.6693	0.3234
E	0.8223	0.827	0.8455	0.8336	0	0.835	0.8194	0.9648	0.8177
F	0.3696	0.3823	0.4376	0.4117	0.835	0	0.3683	0.6893	0.3604
G	0.2747	0.2799	0.3517	0.3321	0.8194	0.3683	0	0.6552	0.253
H	0.6523	0.6618	0.689	0.6693	0.9648	0.6893	0.6552	0	0.6516
I	0.2472	0.2622	0.3421	0.3234	0.8177	0.3604	0.253	0.6516	0
J	0.7424	0.7456	0.7723	0.7523	1	0.7639	0.7372	0.9097	0.7411

In the above table, the column heading A to I represents cluster distance between the corresponding objects.Dunn's Index values are derived for the 10 XML documents and also the value of k (number of clusters)is found . Adopt this value in k-means for extracting the clustering results. It is discovered the strong associations between the frequent XML documets of each cluster through proposed method.

Table 4 : Dunn's Indexes

Dataset	Dunn's Index at Cluster1	Dunn's Index at Cluster2	Dunn's Index at Cluster3
Wiki	0.8988	1.1862	1.4889
ebay	0.9122	0.9901	1.276
ubid	0.6122	1.112	1.012

The following graph depicts the Dunn's Index values for wiki, ebay, and ubid XML datasets. This Dunn's Index value is maximized at c=3 for wiki datasets, at c=3 for ebay dataset, and at c=2 ubid datasets. These Dunn's Index value is calculated based on the inter and intra cluster similarity of XML datasets. From this we have received the clustering tendency (i.e number of clusters) for XML documents.

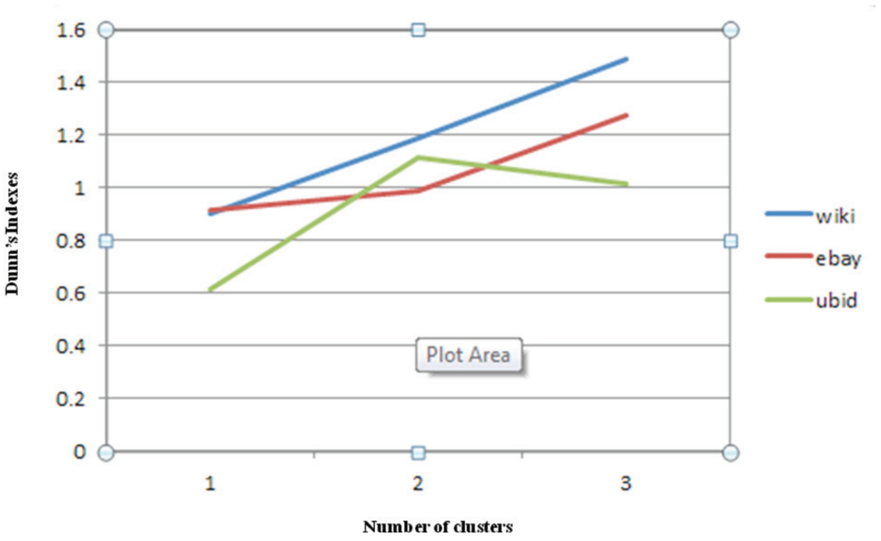

Figure 2 : Number of clusters Vs Dunn's Index Graph

5. Conclusion

The present system performs the clustering by the similarity features of XML documents. But, it needs to perform the clustering for user interested XML documents. So, it is required to determine whether the documents are frequent or not. However, Apriori algorithm is the most established algorithm for finding the frequent XML documents from a used transactional dataset; the proposed method is to discover the meaningful clustered wise associations. The detection of structural similarities among XML documents will help to solve the problem of recognizing different sources that provide the same kind of information or in the structural and content analysis of a Web site. In future work, focus the clustering of same kind of web sources (or their XML files) based on their content and structures information and concentrate on parallel Apriori, because it needs to scan the dataset many times and to generate many candidate XML itemsets. Therefore, frequent pattern mining becomes more problematic when they are accessed Big Data. When the dataset size is huge (big data), both memory use and computational cost can still be very expensive. So, in future, planning to describe the Apriori algorithm based on MapReduce mode, which can handle massive datasets with a large number of nodes on Hadoop platform.

6. References

[1]. Dennis, E. H, et al., "Discovery of temporal associations in Multivariate time series", IEEE Trans. on Knowledge and Data Mining, 2014

[2]. Lucie Xyleme. A dynamic warehouse for XML data of the web. IEEE Data Engineering Bulletin, 2001.

[3]. Termier, M.-C. Rousset, and M. Sebag. Mining XML data with frequent trees. In DBFusion Workshop'02, pages 87–96.

[4] R. Meo, G. Psaila, and S. Ceri. A new SQLlike operator for mining association rules. In The VLDB

Journal, pages 122–133, 1996.

[5]. D. Braga, A. Campi, M. Klemettinen, and P. L. Lanzi. Mining association rules from xml data. In Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2002), September 4-6, Aixen-Provence, France 2002.

[6]. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, SIGMOD93, pages 207–216, Washington, D.C., USA, May 1993.

[7]. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, Proceedings of 20th International Conference on Very Large Data Bases, pages 487–499, Santiago, Chile, September 12-15 1994.

[8]. Guimei, et al., " A flexible approach to finding representative pattern sets" , IEEE Trans. on Knowledge and Data Mining, Vol. 26, No. 7, 2014, pp. 1562-1574

[9]. Victorial Nebot et al., "Finding association rules in semantic web data", Knowledge based systems, Elsevier, 2012, pp. 51-62

[10]. Matthias Steinbrecher, et al.," Visualizing and fuzzy clustering for discovering temporal trajectories of association rules", Journal of computer and system sciences , 2010

[11]. Xindong Wu, "Data Mining with Big Data" , IEEE Trans. On Know. and Data Engg., Vol. 26, No.1, Jan 2014.

[13] D. Braga, A. Campi, M. Klemettinen, and P. L. Lanzi. Mining association rules from xml data. In Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2002), September 4-6, Aixen-Provence, France 2002.

[14] World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Second Edition) W3C Recommendation. http://www.w3.org/XML.

[15] Xyleme. http://www.xyleme.com.

[16] XMLDataRepository,http://www.cs.washington.edu/research/XML datasets/