Web Log Mining using K-Apriori Algorithm

Ashok Kumar D
Department of Computer Science
Government Arts College
Trichy, India

Loraine Charlet Annie M.C.
Department of Computer Science
Government Arts College
Trichy, India

ABSTRACT

Web log mining is a data mining technique which extracts useful information from the World Wide Web's (WWW) client usage details. Automated data gathering has resulted in extremely large information about web access and it can be represented in binary form. A novel method called K-Apriori algorithm is proposed here, to find the frequently accessed web pages from the very large binary weblog databases. Experimental results show that the proposed method has shows higher performance in terms of objectivity and subjectivity.

General Terms

Web mining

Keywords

Wiener Transformation, K-Apriori, Web mining.

1. INTRODUCTION

Web is a collection of inter-related files on one or more Web servers. Web mining discovers and extracts useful information from the World Wide Web (WWW) documents and services using the data mining techniques. Most users obtain WWW information using a combination of search engines and browsers; however these two types of retrieval mechanism do not address all of a user's information needs. The resulting growth in on-line information combined with the almost unstructured web data necessitates the development of computationally efficient web mining tools. Web Mining can be classified [1] as, web content mining, web structure mining and web usage mining. Web content mining means automatic search of information resources available online [2], in short, mining the data on the Web. Web structure mining means mining the web document's structure and links, in short, mining the Web structure data. Web usage mining includes the data from server access logs, user registration or profiles, user sessions or transactions, in short, mining the Web log data. Web mining subtasks are (a) resource finding and retrieving, (b) information selection and pre-processing, (c) patterns analysis and recognition, (d) validation and interpretation, and (e) visualization [3].

Web mining advantages makes this technology useful to corporations including the government agencies. In personalized marketing, E-commerce is enabled which eventually results in higher trade volumes. To classify threats and fight against terrorism, the government agencies are using this technology. By utilizing the acquired insight of customer requirements the corporations can find, attract and retain customers. In web mining client profiles are created which can be utilized by companies which can increase their profit. Companies can even retain the customer by providing promotional offers, thus reducing the risk of losing customers.

Web mining has ethical concerns when it affects the privacy of individuals. Using this technology information concerning an individual is obtained and can be used without their knowledge which affects the privacy of the individuals. Information collected for a specific purpose by the websites can be used unknowingly for a different purpose will affect the users' interests for the website. The collected data is being made anonymous and much extra information can be inferred by combining two separate unscrupulous data from the user.

Weblog databases are very large binary databases maintained by web servers in which each web page visit have an update as 1 if it accessed otherwise 0, for particular time duration like 1 day or 1 hour depending on the websites' usage. Web Servers maintains these types of weblogs for particular websites' maintenance. The databases are updated every time so it will be very complex for processing. At the end of the week or a day, the pages visited or accessed need to be identified to improve the sales if it is an online store website or to design promotional campaigns. Web servers will search based on pages visited, keywords used for search. These web servers are used for website maintenance and report generation.

Data mining techniques such as Clustering and Association rules are used here for efficient web log mining. Clustering is finding groups of objects such that the objects in a group will be similar or correlated to one another and different from or unrelated to the objects in other groups. Clustering is based on information found in the data that describes the objects and their relationships. The main objective of clustering in web mining is to understand the related documents for browsing or to summarize the large data sets. Web Documents are divided into groups based on a similarity metric. Partitional Clustering is a division of data objects into non-overlapping subsets or clusters such that each data object is in exactly one subset. Association Rules discovers affinities among sets of items across transactions. $X \rightarrow Y$ where X, Y are sets of items with c% confidence and s% support. Each rule is a binary partitioning of a frequent itemset. The support of an itemset is defined as the proportion of transactions in the data set which contain the itemset. Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern. Association Rules obtained from the same itemset have identical support but can have different confidence.

Web mining has lots of pros, and the cons can be limited by law. Hence lots of web log mining algorithms are being developed to improve the performance due to its advantages. Web log mining is an important area of data mining which deals with the extraction of interesting knowledge from the World Wide Web. For frequent itemset generation and for deriving association rules, large number of algorithms available like Apriori [4, 5], FP-Growth [6], Opportune

Project [7], CT-PRO [8], Eclat [9], RAAT [10]. The first and arguably most influential algorithm for efficient association rule discovery is Apriori. Apriori-inspired algorithms show good performance with sparse datasets such as weblog data, market basket data so here Apriori algorithm is considered. The Apriori algorithm extracts a set of frequent itemsets from the data, and then pulls out the rules with the highest information content.

2. APRIORI ALGORITHM

Apriori is an influential algorithm for mining frequent itemsets which are used for Boolean association rules generation. Apriori algorithm uses prior knowledge of frequent itemset properties [6]. Apriori [4], is a level-wise, breadth-first algorithm which counts transactions, which is explained in Algorithm 1. Apriori uses an iterative approach known as a level-wise search, in which n-itemsets are used to explore (n+1)-itemsets. First, the set of frequent 1-itemsets is found. This set is denoted L_1. L_1 is used to find L_2, the frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent n-itemsets can be found. Finding of each L_n requires one full scan of the database. To improve the efficiency of the level-wise generation of frequent itemsets Apriori property is used here. Apriori property insists that all non-empty subsets of a frequent itemset must also be frequent. This is made possible because of the anti-monotone property of support measure - the support for an itemset never exceeds the support for its subsets. A two step process is followed here, which consists of join and prune actions.

The join step: n-itemsets candidate set L_n is generated by joining L_{n-1} with itself and this set of candidates is represented here as Cd_n. Consider l_1 and l_2 are the itemsets in L_{n-1}. The notation $l_i[j]$ refers to the j^{th} item in l_i. Apriori assumes that itemsets are sorted in increasing lexicographic order. The join, is performed, where members of L_{n-1} are joinable if their first (n - 2) items are in common. The condition $l_1[n - 1] < l_2[n - 1]$ simply ensures that no duplicates are generated.

The prune step: Cd_n is a superset of L_n , it means that its members may or may not be frequent, but all of the frequent n-itemsets are included in Cd_n. To determine the count of each candidate in Cd_n a database scan is done which results in the determination of L_n i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to L_n. Cd_n will be huge, and so this involves heavy computations. To reduce the size of Cd_n, the Apriori property is used here as that any (n-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any (n-1)-subset of a candidate n-itemset is not in L_{n-1}, then the candidate cannot be frequent either and so can be removed from Cd_n. This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets.

2.1 Generation of Association Rules

Strong association rules can be generated directly when the frequent itemsets from transactions in a database X have been found [6]. Strong association rules satisfy both minimum support and minimum confidence. The rules are generated from frequent itemsets; hence each rule automatically satisfies minimum support. Based on confidence, association rules can be generated as follows.

a. Generate all non-empty subsets of each frequent itemset, F.

b. For every non-empty subset E, of F, output the rule E → F-E; if the minimum confidence threshold (min_conf) is satisfied.

support_count(F) / support_count(E) >= min_conf

Algorithm 1: Apriori algorithm for Frequent Itemset Mining

```
Cd_n : Candidate itemset of size n
L_n: frequent itemset of size n
L_1 = {frequent items};
For (n=1; L_n != $\phi$; n++)
        Do begin
        Cd_{n+1} = candidates generated from L_n;
        For each transaction T in database do
            Increment the count of all candidates in Cd_{n+1}
                that are contained in T
            L_{n+1}= candidates in Cd_{n+1} with min_support
        End
Return $\cup_n L_n$
```

The main issues of Apriori algorithm is the Database scanning of the whole dataset for every iteration, full database scan is needed every time so the computational efficiency is very less. In situations with many frequent itemsets, long itemsets, or very low minimum support, it still suffers from the cost of generating a huge number of candidate sets and scanning the database repeatedly to check a large set of candidate itemsets. Discovering pattern of length 100 requires at least 2^{100} candidates it means n_r of subsets, repeated Database scanning is very costly. To overcome these issues, a novel frequent itemset mining method K-Apriori algorithm is proposed here which is explained in section 4.

3. WIENER TRANSFORMATION

The binary data is pre-processed by transforming into real data using the Wiener Transformation, which is a statistical transformation. The approach is based on a stochastic framework. Wiener Transform is efficient on large linear spaces.

The input for wiener transformation is stationary with known autocorrelation. It is a causal transformation. It is based upon linear estimation of statistics [11]. The Wiener transformation is optimal in terms of the mean square error. The Wiener filter is a filter proposed by Norbert Wiener. The syntax for Wiener filter is Y = wiener2 (X, [p, q], noise) for two-dimensional images, which is normally used for image restoration. The same equation is used for data mining task of weblog databases.

The input X is a two-dimensional matrix and the output matrix Y is of the same size. Wiener2 uses a element-wise adaptive Wiener method based on statistics, estimated from a local neighborhood of each element. Wiener estimates the local mean μ and variance σ^2 around each element of the matrix using the equations (1) and (2) given below.

$$\mu = \frac{1}{pq}\sum_{n_1,n_2 \epsilon \eta} X(n_1, n_2) \qquad (1)$$

$$\sigma^2 = \frac{1}{pq}\sum_{n_1,n_2 \epsilon \eta} X^2\big((n_1, n_2) - \mu\big) \qquad (2)$$

where η is the local neighborhood of each element in the input matrix W. Wiener2 then creates a element-wise wiener transformation for each vector based on the neighborhood of the objects using equation (2) and (3) estimates in equation(3),

$$Y(n_1, n_2) = \mu + \frac{\sigma^2 - \lambda^2}{\sigma^2}(X(n_1, n_2) - \mu) \qquad (3)$$

where λ^2 is the average of all the local estimated variances. Since clustering finds similarity between objects, neighborhood property of the wiener transformation helps to find good clusters and makes it computationally efficient.

3.1 K-means algorithm

The wiener transformed data is clustered using the Standard K-means algorithm [12, 13] which is a multi-pass technique. Its main advantage is the short computational time it takes to find a solution so that the clustering is very efficient. The algorithm iterates between the following steps till convergence:

a) Initialize K centroids at random for K clusters and assign each vector to the closest cluster centroid.
b) Compute the centroids of all current clusters.
c) Generate a new partition by assigning each item to the closest cluster centroid.
d) If cluster memberships change compared to the last iteration, go to step 2, else stop.

Since clustering finds similarity between objects, the wiener transformation which is based on neighborhood of objects helps to find good clusters.

4. K-APRIORI ALGORITHM

In K-Apriori algorithm, the binary data is transformed into real domain using linear wiener transformation, based on its neighborhood property. The Wiener transformed data is partitioned into K clusters using the multi-pass K-means algorithm. Apriori procedure is used for the K similar groups of data from which, frequent itemsets can be generated and association rules are derived. Large datasets are partitioned so that the candidate itemsets generated will be very less and database scanning will also be done for adequate data which increases the efficiency. The K-Apriori algorithm is described in Algorithm2.

Algorithm2:K-Apriori Algorithm for Frequent Itemset Mining

```
Input: Binary data matrix X of size p x q, K
Output:  Frequent Itemsets and Association rules
//Binary data is transformed to real data using Wiener transformation on a vector basis.
V = Call function wiener2 ($X_i$)
// $X_i$ is a vector i of X
//Calculate K clusters ($C_1, C_2, \ldots C_K$) for V using
  K-means algorithm
$C_1, C_2, \ldots C_K$ = Call function kmeans (V, K)
For each cluster $C_i$
    $Cd_n$ : Candidate itemset of size n
    $L_n$: frequent itemset of size n
    $L_1$ = {frequent items};
    For (n=1; $L_n$ != ϕ ; n++)
      Do begin
        $Cd_{n+1}$ = candidates generated from $L_n$;
        For each transaction T in database do
          Increment the count of all candidates in $Cd_{n+1}$
                which are contained in T
        $L_{n+1}$= candidates in $Cd_{n+1}$ with min_support
      End
    $\bigcup_n L_n$ are the frequent itemsets generated
    End
End
```

Function wiener2 (X_i)

```
Input    : Binary data vector $X_i$ of size 1 X q
Output   : Transformed data vector $Y_i$ of size 1 X q
Step 1: Calculate the mean μ for the input vector $X_i$ around
        each element
            $\mu = \frac{1}{pq}\sum_{n_1,n_2 \epsilon \eta} X(n_1, n_2)$
        where η is the local neighborhood of each element
Step 2: Calculate the variance $\sigma^2$ around each element for the
        vector
            $\sigma^2 = \frac{1}{pq}\sum_{n_1,n_2 \epsilon \eta} X^2((n_1, n_2) - \mu)$
        where η is the local neighborhood of each element
Step 3: Perform wiener transformation for each element in
        the vector using equation Y based on its
        neighborhood
            $Y(n_{1,}, n_{2)} = \mu + \frac{\sigma^2 - \lambda^2}{\sigma^2}(X(n_{1,}, n_2) - \mu)$
        where $\lambda^2$ is the average of all the local estimated
        variances.
```

Function kmeans (V, K)

```
Input: Wiener Transformed data matrix V and
        number of clusters K.
Output: K clusters
Step 1: Choose initial cluster centroids $Z_1, Z_2, \ldots, Z_K$ randomly
        from the N points;   $X_1, X_2, \ldots X_p$ , $X_i \in R^q$
        where q is the number of features/attributes
Step 2: Assign point $X_i$, i = 1, 2, …, p to cluster $C_j$,
        where j = 1,2,…,K,  if and only if
        $\| X_i - Z_j \| < \| X_i - Z_t \|$, t = 1, 2,…,K. and $j \neq t$.
        Ties are resolved arbitrarily.
Step3: Compute the new cluster centroids $Z_1^*, Z_2^*, \ldots, Z_K^*$ as
        $Z_i^* = \frac{1}{l_j}\sum_{X_j \epsilon C_j} X_i$
        where i = 1, 2,…,K, and $l_j$ =Number of points in $C_j$.
Step 4: If $Z_i^* = Z_i$ , i = 1, 2,…, K then terminate.
        Otherwise $Z_i \leftarrow Z_i^*$ and go to step 2.
```

Clustering groups the similar web access records from the weblogs using the linear wiener transformation. Using the similarity property of the records in the clsuters, K-Apriori algorithm generates the frequently accessed web pages efficiently. From the frequently accessed webpages, Association Rules can be derived which helps to improve the performance of the website.

5. EMPRICAL STUDY AND RESULTS

K-Apriori algorithm is an enhanced version of Apriori algorithm. The qualitative evaluation of quantitative results is done in this section.

Here, weblog database is a binary database maintained by a web server. The sample weblog database considered here has 52 attributes which describes 22 web pages and 30 keywords. If a client accesses a page, that attribute in the record will be updated as 1, otherwise 0. Web access for one minute is considered as a record. A record has entries as value 1 for each page visited and also for search keywords; otherwise the entry will be 0. If more than 1 minute the client accesses the website, new record is created in the binary weblog database for the same client. If more number of frequent accessed pages and corresponding association rules are generated means, then the website usage can be improved by developing more number of links for these frequently accessed web pages and the frequent keywords can be used in search engines for the particular website which increases the usage of the website correspondingly its popularity. K-Apriori algorithm efficiency is evaluated for frequent itemsets and association rules generation with different confidence levels.

Frequent itemsets and ARs generated for the Weblog dataset with two clusters (K=2) of Apriori and K-Apriori algorithm are given as a summary in Table 1.

Table 1. Apriori & K-Apriori Result analysis for Weblog Dataset with Support=15%

Confidence (%)	Maximum Number of Frequent Itemsets		Total Number of Frequent Itemsets		Total Number of Association rules	
	Apriori	K-Apriori	Apriori	K-Apriori	Apriori	K-Apriori
50	3	3	25	43	32	50
60	3	3	25	43	25	35
70	3	3	25	43	17	35
80	3	3	25	43	12	27
90	3	3	25	43	12	21
100	3	3	25	43	6	8

Table 1 compares the results of Apriori and K-Apriori algorithm for 15% support based on the n-itemsets generated, total number of Frequent itemsets generated and total number of Association rules derived for weblog dataset. For K-Apriori, the number of ARs generated increases for higher confidence levels, it implies that the website's usage and robustness can be increased using these ARs.

From Table 1, it is observed that for 15% support, A, B, H, J, M, O, R, T and U are the 1-itemsets, OR, MU, MT, MR, MO, JM, HO, HM, BM, AO, AM and AH are the 2-itemsets and AHM, AMO, HMO and MOR are the 3-itemsets generated by Apriori algorithm. For 100% confidence the Association Rules (AR) generated by Apriori algorithm are given below.

A→M B→M

AH→B HO→A

AO→M T→M

From Table 1, it is observed that for 15% support, AHM, AMO, HMO and MOR are the 3-itemsets generated by K-Apriori algorithm for 1st cluster. A, B, H, J, M, O, R, T and U are the 1-itemsets. OR, MU, MT, MR, MO, JM, HO, HM, BM, AO, AM and AH are the 2-itemsets generated. For 100% confidence the ARs generated are given below.

A→M B→M

AH→B HO→A

AO→M T→M

For 15% support of 2nd cluster, G,H,I, J,L,M and N are the 1-itemsets, MN, LM, IM, HN, HM, HI, GM and GH are the 2-itemsets generated. GHM, HIM and HMN are the 3-itemsets generated.

For 100% confidence of 2nd cluster, ARs generated are

GH→M L→M

From the results of K-Apriori algorithm, it is observed that M is the most heavily used page, for improving the performance of the website, M must be made as the home page and should have more links to access it. A is the other page which is accessed more, hence put some links from various pages will improve the sales of the online store if it is an online store website.

Table 2. Apriori & K-Apriori Result analysis for Weblog Dataset with Support=25%

Confidence (%)	Maximum Number of Frequent Itemset		Total Number of Frequent Itemsets		Total Number of Association rules	
	Apriori	K-Apriori	Apriori	K-Apriori	Apriori	K-Apriori
50	2	3	5	16	4	13
60	2	3	5	16	3	2
70	2	3	5	16	3	2
80	2	3	5	16	2	2
90	2	3	5	16	2	7
100	2	3	5	16	0	3

Frequent itemsets and ARs generated for the Weblog dataset with two clusters (K=2) of Apriori and K-Apriori algorithm are given as a summary in Table 2 for 25% support.

From Table 2, it is observed that for 25% support, MO and HM are the frequent 2-itemsets are generated by Apriori algorithm. H, M and O are the frequent 1-itemsets generated. Strong rules generated for 90% confidence by Apriori algorithm are H→M and O→M.

From Table 2, it is observed that K =2. For 1st cluster, A, H, M, O and R are the 1-itemsets; OR, MR, MO, HO, HM and AM are the 2-itemsets and HMO & MOR are the 3-itemsets generated by K-Apriori algorithm. For 100% confidence, ARs generated for 1st cluster are

A→M H→A HO→M

For 2nd cluster, H,M are the 1-itemsets and HM is the 2-itemset generated for K-Apriori algorithm. H→M is the AR generated for 90% confidence of 2nd cluster.

From the results obtained, it is observed that M is the most heavily used page, for improving the performance of the website, M should has more links to access it, for instance it can be the Home page of the website. Using these frequent itemsets generated website usage and its performance can be increased.

6. CONCLUSION

Weblog databases are homogenous databases in binary format. In the very large binary databases to find the correlation among the accessed web pages, frequent itemsets concept of Apriori algorithm are used here from which association rules can be derived. A new K-Apriori Algorithm is proposed here to perform frequent itemset mining in an efficient manner; the anti-monocity property makes it simple and perfect for binary databases. Initially the binary data is clustered using the multi-pass K-means algorithm based on the linear wiener transformation. The similar groups of data or clusters are used in Apriori procedure for frequent itemset generation. Apriori algorithm is used for K clusters and the frequent itemsets are generated from which Association rules are derived. Experiments are performed using real and synthetic data, and found K-Apriori algorithm is more efficient compared to Apriori algorithm.

7. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their thorough reviews, and constructive suggestions which significantly enhance the presentation of the paper.

8. REFERENCES

[1] Raymond Kosala and Hendrik Blockeel. 2000, "Web Mining Research: A Survey", ACM SIGKDD.

[2] Sanjay Kumar Madria, Sourav S Bhowmick, Ng W.K. and Lim E.P. 1999, "Research Issues in Web Data Mining", Springer.

[3] Qingyu Zhang and Richard S. Segall. 2008, "Web Mining: A Survey Of Current Research, Techniques, And Software", In International Journal of Information Technology and Decision Making, Volume: 07, Issue: 04, pp. 683-720.

[4] Agrawal R and Srikant R . 1984. " Fast algorithms for mining association rules", In Proceedings of the 20th VLDB conference, pp. 487–499.

[5] Borgelt C. 2003. "Efficient Implementations of Apriori and Eclat", Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI), Melbourne, Florida.

[6] Han J, Pei H, and Yin Y. 2000. "Mining Frequent Patterns without Candidate Generation", In Proc. Conf. on the Management of Data SIGMOD, Dallas, TX. ACM Press, New York, USA.

[7] Liu J,Pan Y,Wang K, andHan J. 2002. "Mining Frequent Item Sets by Opportunistic Projection", Proceedings of ACM SIGKDD, Edmonton, Alberta, Canada.

[8] Gopalan R. P and Sucahyo Y. G. 2004. "High Performance Frequent Pattern Extraction using Compressed FPTrees", Proceedings of SIAM International Workshop on High Performance and Distributed Mining (HPDM),Orlando, USA.

[9] Han J and Kamber M. 2001. "Data Mining: Concepts and Techniuqes", Morgan Kaufmann Publishers, San Francisco, CA.

[10] Wanjun Yu, Xiaochun Wang, Fangyi Wang, Erkang Wang and Bowen Chen, 2008. "The research of improved apriori algorithm for mining association rules", 11th IEEE International Conference on Communication Technology, pp. 513 - 516.

[11] Frederic Garcia Becerro. 2007, "Report on Wiener filtering", Image Analysis, Vibot. [Online] http://eia.udg.edu/~fgarciab/.

[12] McQueen J. 1967. "Some methods for classification and analysis of multivariate observations", In Proc. of 5th Berkeley Symp Mathematics,statistics and probability, pp.281-296.

[13] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu, 2002. "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", IEEE Transactions on Pattern Analysis and Machine Intelligence", pp. 881-892.