한국정보전자통신기술학회논문지(jkiiect)'19-12 Vol.12 No.6
논문 19-12-06-681 http://dx.doi.org/10.17661/jkiiect.2019.12.6.681

User Access Patterns Discovery based on Apriori Algorithm under Web Logs

Cong-Lin Ran*, Suck-Tae Joung**

웹 로그에서의 Apriori 알고리즘 기반 사용자 액세스 패턴 발견

염종림*, 정석태**

Abstract Web usage pattern discovery is an advanced means by using web log data, and it's also a specific application of data mining technology in Web log data mining. In education Data Mining (DM) is the application of Data Mining techniques to educational data (such as Web logs of University, e-learning, adaptive hypermedia and intelligent tutoring systems, etc.), and so, its objective is to analyze these types of data in order to resolve educational research issues. In this paper, the Web log data of a university are used as the research object of data mining. With using the database OLAP technology the Web log data are preprocessed into the data format that can be used for data mining, and the processing results are stored into the MSSQL. At the same time the basic data statistics and analysis are completed based on the processed Web log records. In addition, we introduced the Apriori Algorithm of Web usage pattern mining and its implementation process, developed the Apriori Algorithm program in Python development environment, then gave the performance of the Apriori Algorithm and realized the mining of Web user access pattern. The results have important theoretical significance for the application of the patterns in the development of teaching systems. The next research is to explore the improvement of the Apriori Algorithm in the distributed computing environment.

요 약 웹 사용 패턴 발견은 웹 로그 데이터를 사용하는 고급 수단이며 웹 로그 데이터 마이닝에 데이터 마이닝 기술을 적용한 특정 응용이다. 교육 분야에서 데이터 마이닝 (DM)은 데이터 마이닝 기술을 교육 데이터 (대학의 웹 로그, e-러닝, 적응형 하이퍼미디어 및 지능형 튜터링시스템 등)에 적용한다. 따라서 교육 연구 문제를 해결하기 위해 이러한 유형의 데이터를 분석하는 것이 목표이다. 본 논문에서는 대학의 웹 로그 데이터가 데이터 마이닝의 연구 대상으로 사용되어 진다. 데이터베이스 OLAP 기술을 사용하여 웹 로그 데이터가 데이터 마이닝에 사용될 수 있는 데이터 형식으로 사전 처리되고 그 처리 결과가 MSSQL에 저장된다. 동시에 처리 된 웹 로그 레코드를 기반으로 기본 데이터 통계 및 분석이 완료된다. 또한 웹 사용 패턴 마이닝의 Apriori Algorithm 및 구현 프로세스를 소개하고 Python 개발 환경에서 Apriori Algorithm 프로그램을 개발했다. 그런 다음 Apriori Algorithm의 성능을 보이고 웹 사용자 액세스 패턴의 마이닝을 실현했다. 이 연구 결과는 교육 시스템 개발에 패턴을 적용하는데 중요한 이론적 의미를 갖는다. 다음 연구로는 분산 컴퓨팅 환경에서 Apriori Algorithm의 성능 향상을 연구하는 것이다.

Key Words : Web Logs, Educational Data Mining, Apriori Algorithm, Access Pattern, Frequent Itemsets

This research was supported by the Teaching Reform Project of Jiangxi Provincial Department of Education(No. JXJG-17-17-15).
*Department of Information Technology Center, Jiujiang University, China
**Corresponding Author : Department of Computer and Software Engineering, Wonkwang University, Korea(stjoung@wku.ac.kr)
Received November 06, 2019 Revised December 27, 2019 Accepted December 27, 2019

1. Introduction

In the era where the information explodes, the Internet and information technology are experiencing unprecedented rapid development, based on the technology cloud computing and big data technology are rapidly applied in various industries. In particular, with the successful development and popularization of 5G technology, on the Internet every user is not only the "consumer" of information, but also the "producer" of information. As a result, every day the users visit various Web sites and information systems to generate massive user log data, The persistent growth of data in education continues. More institutes now store terabytes and even petabytes of educational data. A lot of valuable information is hidden in these data. How to effectively use these massive education data to serve education is an urgent problem that educators and government departments need to solve. Therefore Educational Data Mining (EDM) has emerged as a research area in recent years for researchers all over the world from different and related research areas such as: Offline education, E-learning and Learning Management System (LMS), Intelligent Tutoring (ITS) and Adaptive Educational, Hypermedia System (AEHS)[1].

Educational Data Mining is a promising discipline, concerned with developing methods for exploring the unique types of data gathered from educational settings, and applying those methods to better understand learner's behavior, and the environment which they learn in [2]. In brief, it uses computational approaches to analyze educational data in order to study educational questions. According to the different objects of data mining, in education the web data mining is still divided into three categories in general [3]: Web Content Mining, Web Structure Mining and Web Usage Mining. In education Web Usage Mining is to find user access patterns from Web logs as a basis to provide better Web services to users. In this paper, we will use data mining technology to explore and discover the potential user access patterns taking a university's Web logs as the analysis and research object.

This paper is organized as follows. Section 2 discusses the concepts related to Apriori Algorithm, then gives important functions and pseudo codes to generate frequent patterns. In section 3 an experimental study is carried out, the data is processed and analyzed. At the same time according to the execution process of Apriori algorithm the programming is developed and the user access patterns are generated, At last we analyze the experimental results and give the experimental performance of the Apriori Algorithm. Conclusion is given in section 4.

2. Apriori algorithm

The Apriori algorithm[4] is proposed by Rakesh Agrawal and Ramakrishnan Skrikant and is the most influential algorithm for mining association frequent item sets in Boolean value. Its core is a recursive algorithm based on the idea of two-stage frequent item sets. The association rule belongs to single-dimensional, single-level and Boolean association rules in

classification [5].

2.1 Basic terminology

For explanation, *A* and *B* represent events, *X* represents a transaction data item of a dat a set, and *D* represents a set of all transactio ns.

2.1.1 Support

$$Support(A=\rangle B) = P(A \cup B) \quad (1)$$

The expression (1) represents the probability that *A* and *B* occur at the same time. Support indicates the frequency of occurrences in the pattern. The support degree *S(X)* of the data item *X* is the ratio of the number of transactions including *X* in the transaction set *D* to the total number of transactions of *D*. The support degree *S* of the mode *X=⟩Y* is defined as the ratio *S%* of the transactions including *X∪Y* in *D*, indicating the ratio of the number of transactions including *X* and *Y* to the total transaction amount of D, that is, the probability that *X* and *Y* occur simultaneously.

2.1.2 Confidence

$$Confidence(A \Rightarrow B) = \frac{Support(AUB)}{Support(A)} \quad (2)$$

The expression (2) indicates the strength of the pattern, it represents the ratio of the probability of simultaneous occurrence of *A* and *B* to the probability of occurrence of *A*. Confidence *C* of mode *X=⟩Y* is defined as that in transaction set *D*, *C%* of transactions contain *X* and *Y*, indicating how likely the transactions containing *X* in *D* contain *Y*.

2.1.3 K-ItemSet

If event *A* contains k elements, then event *A* is called *K* Item Set, and event *A* which me ets the minimum support threshold is called frequent *K-ItemSet*.

2.2 Generating frequent patterns

The steps for generating frequent patterns from frequent itemsets are presented in Table 1.

Table 1. The steps of generating frequent patterns

The steps of Generating Frequent Patterns
(1) The necessary condition for *K-dimensional* data item set L_k to be a frequent item set is that all *K-1* dimensional sub-item sets are also frequent item sets, marked as L_{k-1}. (2) If any *K-1* dimensional subset of the *K-dimensional* data item set L_k is not a frequent item set, then the *K-dimensional* data item set L_k itself is not the largest data item set. (3) L_k is a *K-dimensional* frequent item set. If the number of *K-1-dimensional* subsets containing L_k in all *K-1-dimensional* frequent item sets L_{k-1} is less than *K*, then L_k can not be the *K-dimensional* maximum frequent data item set. (4) Rules that satisfy both the minimum support threshold and the minimum confidence threshold are called strong rules.

All non-empty subsets of any frequent term must also be frequent. That is, when generating a k-item candidate set, if the elements in the candidate set are not in the k-1 frequent set, the element must not be a frequent set. At this time, the support degree does not need to be calculated and can be directly removed. The detailed pseudo-code of the Apriori Algorithm is presented in Table 2.

Table 2. The pseudo-code of the Apriori Algorithm

The Pseudo-code of the Apriori Algorithm
C_1={candidate *1-itemsets*};

```
L1={c∈|c.count≥minsupport};
For(k=2,Lk-1≠Φ,k++)
    Ck=sc_candidate(Lk-1);
    for all t ransactions t∈ D  //scan D for counts
      Ck=count_support(Ck,t);   //get the subsets of t
 that are candidates
       for all candidates c∈Ck
         c.count = c.count+1;
    next
  Lk={c∈Ck|c.count≥minsupport};
 next
  result set = resultset ∪Lk
```

Among them, *D represents* the database; *minsupport* represents the given minimum support; *resultset* represents all the largest item sets.

2.3 Sc_ candidate function

The parameter of the function is L_{K-1}, that is, all the maximum *k-1* dimension item sets, and the result of the candidate item set C_K is returned with *K* items. In fact, C_K is a superset of the largest *k-dimensional* item set. The function *Count_support* is used to calculate the support of the item set, and then L_K is generated. The detailed descriptions how to complete these functions are given. First of all, C_K is generated by L_{K-1} self-connect operation, which is called the join. The pseudo-code of the Join operation can be expressed in Table 3.

If $C_{k'}$= { X∪ X'|X, X'∈ L_{k-1} , |X∩ X'|=k-2},then for any *c*, $c \in C_k$, the item sets which are not included in item sets of L_{k-1} will be delete from *k-1-dimensional* subsets of C_k, the candidate item set C_k will be generated. The step is called as the prune. The pseudo-code of the prune operation is presented in Table 4.

Table 3. The Pseudo-code of the Join Operation

The Pseudo-code of the Join Operation
inser *t* into C_k select $P.item_1$, $P.item_2$, . . $P.item_{k-1}$, $Q.item_{k-1}$ from L_{k-1} P, $_{Lk-1}Q$ where $P.item_1 = Q.item_1$, . . $P.item_{k-2} = Q.item_{k-2}$, $P.item_{k-1} < Q.item_{k-1}$

Table 4. The Pseudo-code of the Prune Operation

The Pseudo-code of the Prune Operation
for all itemset c∈ C_k for all subset *S* of c in *k-1-dimensional* if (*s* not belong to L_{k-1}) then delete *c* from C_k; C_k = = { X∈ C_k| All *k-1-dimensional* subsets of *X* are in L_{K-1} }

2.4 Sc_candidate function

In Apriori algorithm, the parameters of the *count_support* function are the candidate item set C_k and a certain transaction record *T*, in results the candidate item sets of the transaction *T* are returned. Its function is to find all the candidate item sets contained in the transaction *T*. For any item set *C* in C_k, starting from the root node, by hashing each item in *C*, if the item is in a transaction, the same operation will be continued to be applied with subsequent levels in the hash tree. If the item is not in transaction *T*, the hash operation is not continued. If it is currently in the *I* level, only the items after *I* need to be considered. Because the items in each item set are arranged in alphabetical order.

3. Empirical research

Obtaining users' access patterns on the Internet is one of the main tasks of Web log mining. User's access pattern is the way that users browse Web sites. User's access pattern can be obtained by mining user's traversal path. Frequent traversal path is a

sequence of continuous pages satisfying a certain degree of support in the maximum forward path (MFP) [6-8]. The number of MFPs containing frequent traversal paths is called the degree of support of frequent traversal paths. The length of the frequent traversal path is defined as the number of pages which it contains.

Web log mining is divided into four stages: log preprocessing, session identification, pattern discovery and pattern analysis [9]. Cong-Lin Ran, Suck-Tae Joung et al. [10] conducted detailed research on log collection and pre-processing. On this basis, the experimental research was carried out on the following three stages of work.

3.1 Data processing

The log data used in this study comes from a comprehensive University with nearly 50,000 students and teachers. The University has been recruiting international students. It has the characteristics of local university development and advanced information technology application. The school has built nearly 100 different teaching websites. One part of the experimental data come from the IIS WWW service logs of the selected University. These collected unstructured log files are imported into the MSSQL database using Log Parser Lizard. The other part is structured data that is directly exported from WAF (Web Application Firewall). Direct access to these exported large-capacity CSV data requires a large amount of computer resource overhead. In order to uniformly archive data into the SQL server database, it is difficult to import quickly using conventional methods. We adjust the experimental parameters for different encodings of the database using the SQL statement "bulk insert TableName from 'D:\xx\WebAccessLog.csv' With (Fieldterminator = ',', Rowterminator = '0x0a')" (where the parameters are based on different CSV files). The format needs to be properly adjusted.)Several large-capacity CSV files are successfully imported into the database, which facilitates multi-dimensional analysis operations such as Drill-up and Drill-down, Slice and Dice, and Pivot using OLAP technology of the database.

After eliminating hackers' attacks on illegal access dates from raw 20G log data which include the logs accessed by search engines such as Baidu and Google, nearly 20000 user access records were obtained. Finally, the noise data can be removed by programming, then the user access data of 12 selected columns are focused for data analysis.

3.2 Data analysis

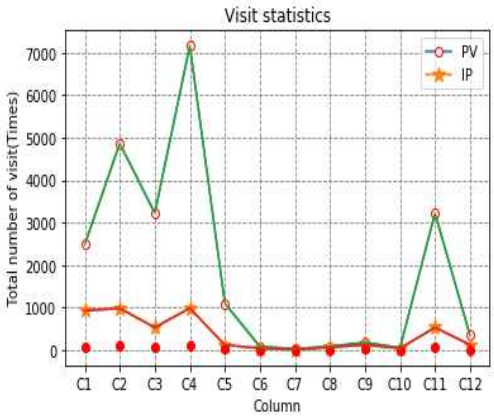

Fig. 1. The statistics of the data

The statistics of the data obtained at this time are shown in Fig. 1. The green polyline represents the number of valid users accessing PV obtained during the observation period of 12 columns, and the

yellow line indicates the IP number of the test period of the 12 columns. The red origin map shows the average access of the column. From the data chart, the top five columns in the access ranking are $C_1C_2C_3C_4C_5$. It can be seen from the statistics that the visits of these columns are random. The page view increased from column C_1 to column C_4, and the page view of column C_4 increased dramatically. The page view was almost 0 from column C_5 to column C_{10}, at the last the page view increased and then decreased again from column C_{10} to column C_{12}.

3.3 Experiment

The problem of mining access pattern rules can be described as follows:

Each user U_{id} visiting the Web site is defined as a column access sequence C_i in a 24-hour visit cycle, and all page column sequences C_i visited by the same independent IP user in the visit cycle constitute an independent transaction $Item_i$ (i = 1, 2, 3...), Transaction D consists of a set of independent transactions $Item_i$. User U_{id} has visited the column marked as *1*, instead marked as *0*, such as user access log transaction in Fig. 2. In order to facilitate the implementation of functions in the program, the sequence of a marked as *1* in user access transaction log of the Fig. 2 is used as *A*, *B*, *C*, *D*, *G*, *E* respectively. Represents the visited columns, which constitute the transaction item set T_{id}. Assuming the minimum support *minsupport = 3,* we can get the Frequent-item projection about the user access transaction log, it's shown in Fig. 3.

The problem to be solved is to find out the relationship between the sequences after a given confidence level through transaction data analysis, and to reflect the rules and importance of user access column association between Web site columns.

Uid	*C1*	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*	*C8*	*C9*	*C10*	*C11*	*C12*
001	0	1	1	1	0	0	1	0	0	0	0	1
002	1	1	1	0	0	1	1	0	0	0	0	0
003	1	0	1	1	1	0	0	0	0	1	0	0
004	1	0	1	0	0	0	0	0	0	0	0	1
005	1	1	0	1	0	0	0	1	1	0	0	0

Fig. 2. User access transaction log

T_{id}	Items	Frequent-item projection
001	B,C,D,G,L	B,C,D
002	A,B,C,F,G	A,B,C
003	A,C,D,E,J	A,C,D
004	A,C,L	A,C
005	A,B,D,H,I	A,B,D

Fig. 3. The frequent-item projection

The Apriori Algorithm finds that the frequent item set execution process is shown in Fig. 4. Firstly, the transaction data is filtered to create C_1 according to the preset degree of support, and the data set L_1 is obtained after filtering. Then all the elements in L_1 are connected to each other to generate a new data set C_2,and then the data set L_2 is filtered according to the conditions, and the data sets L_2 are sequentially iterated. The operation cannot be continued until the end, and the loop ends. According to the preset confidence level, an association is generated, that is, the frequent access Patterns of the website users.

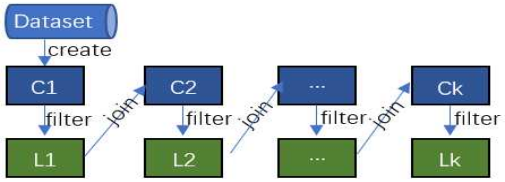

Fig. 4. The Apriori algorithm execution process

The two important steps in the experimental execution process are pruning and join, pruning and join operations to filter and reduce search space, eliminate infrequent item sets, reduce the loops times and improve cycle efficiency. The code of join step is shown in Fig. 5 and the code of pruning step is shown in Fig. 6. During the experiment, we chose the confidence level of 0.5, 0.6, and 0.7 respectively. We found that the C_6 to C_{12} column has very little traffic. If the data is used to interfere with the result very much, it will not generate frequent item sets. Therefore, C_6 to C_{12} is included in the experiment data. The data set from C_1 to C_5 has a strong representative significance. The analysis of the results of program execution shows that the frequent access mode obtained when the confidence is 0.6 is relatively stable, and the results are *[['A'], ['A','D']]*.

```
def aproiri_gen(keys1):
    '''Join step'''
    keys2 = []
    for k1 in keys1:
        for k2 in keys1:
            if k1 != k2:
                key = []
                for k in k1:
                    if k not in key:
                        key.append(k)
                for k in k2:
                    if k not in key:
                        key.append(k)
                key.sort()
                if key not in keys2:
                    keys2.append(key)
    return keys2
```

Fig. 5. Join step code

```
def getCutKeys(keys, C, minSup, length):
    ''' Pruning step '''
    for j, key in enumerate(keys):
        if float(C[j]) / length < minSup:
            keys.remove(key)
    return keys
def keyInT(key, T):

    for k in key:
        if k not in T:
            return False
    return True
```

Fig. 6. Pruning step code

3.4 Experimental result

The Web user access pattern generated by the experiment basically reflects the reality. As can be seen from the experimental results, the user's access pattern to the Web site reflects the importance of the corresponding column of the web site and the content of the Web user's concern is consistent with the degree of attention of the Web builder. "A" corresponds to the "Notification Announcement" column inside and outside the University, and "*D*" corresponds to the "Institutional Activities" column. It should be emphasized here that there are 22 secondary colleges in this University. The University always puts “teaching as the center and student-oriented” in the first place. Therefore, this point has also been reflected in the user access pattern. Another aspect of this access model also reflects the need to adjust the overall organizational structure of the Web site. For other columns and content that users are not very concerned about or interested in, and then make overall planning to provide an important basis for the next revision design.

3.5 Apriori algorithm performance

The experimental operating system is Window10, 64-bit, memory 8G, CPU virtual

multi-core Inter i5, 2.71GHz, memory 8G, hard disk SSD 200G. In order to be compatible with data processing software, database is selected as 32-bit SQL server 2005. The program environment chooses Python 3.7.3, Jupyter notebook and Spyder 3.3.5 are used for IDE development, Conda 4.7.5 is used for package management and environment management.

The data set of Web log used in the experiment is sparse after cleaning, which contains 19773 transaction records. The minimum support thresholds of 40%, 50%, 60% and 70% respectively are shown in Fig. 7, and the time consumed by the Apriori algorithm in mining frequent log access mode. It can be seen from the figure 7 that the time required to run the algorithm increases with the minimum support threshold. The time consumption of the Apriori mining algorithm generally tends to be flat. This is because the dimension of the maximum frequent mode will be larger when the minimum support threshold is smaller, that is to say, the minimum support threshold is inversely proportional to the dimension of the maximum frequent mode. The memory space consumption will also decrease with the increase of the minimum support threshold, and the CPU execution time will also decrease. This is mainly because the increase of support results in the decrease of candidate frequent item set, that is, the increase of infrequent item set, and the decrease of search space and dimension of candidate set.

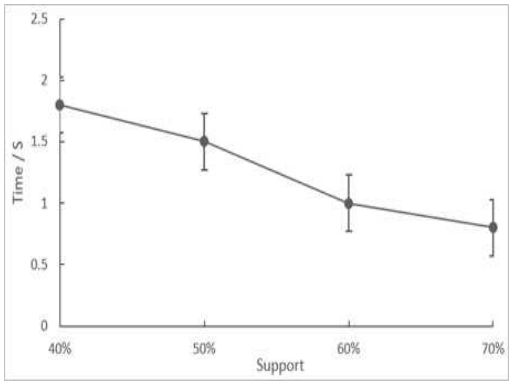

Fig. 7. Apriori algorithm performance

4. Conclusion

In this paper the Apriori algorithm is described for mining user access patterns in large databases, then implemented by programming. The Web log data of a comprehensive university is used as the data set object of the experimental research, and the Apriori algorithm is verified by experiments. The consistency of the Web site user access patterns and the analysis results of the web practice is found. The successful application of these web log mining technologies in education data mining, as well as the discovery of access patterns, will play an extremely important theoretical supporting role in improving the design and service of Web sites, building and optimizing Intelligent Web sites, and improving the reputation and efficiency of Web sites. To develop a more efficient mining algorithm and to quickly mine user access patterns from massive educational data in a distributed computing environment, then to apply the patterns in the development of teaching systems are issues that we need to consider and research further.

REFERENCES

[1] C. Romero and S. Ventura, "Educational Data Mining: A Review of the State-of-the-Art", Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, Vol. 40, pp. 601-618, 2010.

[2] Educational Data Mining. Dec. 2019. Accessed online from http://www.educationaldatamining.org/.

[3] Y. Y. Liao, "The Application of Web Mining in Distance Education Platform", *Proc. of 2nd International Symposium on Computer, Communication, Control and Automationn,* pp. 595-597, 2013.

[4] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", *Proc. of International Conference on Very Large Database,* pp. 487-499, 1994.

[5] The Apriori Algorithm. Aug. 2019. Accessed online from https://en.wikipedia.org/wiki/Apriori_algorithm.

[6] G. Neelima and S. Rodda, "An Overview on Web Usage Mining", Advances in Intelligent Systems and Computing, Vol. 338, pp. 649-652, 2015.

[7] Z. L. Yu, W. T. Zhang and H. Ge, "Hadoop platform based log analysis mode", Computer Engineering and design, Vol. 37, pp. 233-343, 2016.

[8] J. Zhang and Z. H. Tian, "Association-relation mining based on web logs under IIS", Huazhong Univ. of Sci. & Tech.(Nature Science Edition), Vol. 30, pp. 36-39, 2002.

[9] Web mining. Mar. 2019. Accessed online from https://en.wikipedia.org/wiki/Web mining.

[10] C. L. Ran and S. T. Joung, "Research on Data Acquisition Strategy and Its Application in Web Usage Mining", Korea Information Electron Communication Technology, Vol. 12, pp. 232-234, 2019.

Author Biography

Cong-Lin Ran **[Member]**

- July. 2004 ~ Sept. 2018 :Jiujiang University, China, Lecturer
- July. 2010 : Huazhong University of Science and Technology, China, MS
- Sept. 2018 ~ current : Department of Computer and Software Engineering, Wonkwang University, Korea, Ph.D. student

〈Research Interests〉 Big Data Processing & Machine Learning

Suck-Tae Joung **[Member]**

- March. 1996 : Computer Engineering of Tsukuba Univ. Japan, MS
- July. 2000 : Computer Engineering of Tsukuba Univ. Japan, PhD
- Feb. 2001 ~ current : Wonkwang Univ. Dept. of Computer and Software Engineering, Professor

〈Research Interests〉 Big Data Processing&Machine Learning, Visual System