sustainability

MDPI

Article

Emerging Pattern-Based Clustering of Web Users Utilizing a Simple Page-Linked Graph

Xiuming Yu [1], Meijing Li [1], Kyung Ah Kim [2], Jimoon Chung [3] and Keun Ho Ryu [1,*]

[1] Database/Bioinformatics Laboratory, College of Electrical and Computer Engineering, Chungbuk National University, Cheongju, Chungbuk 28644, Korea; yuxiuming@dblab.chungbuk.ac.kr (X.Y.); mjlee@dblab.chungbuk.ac.kr (M.L.)

[2] Department of Biomedical Engineering, College of Medicine, Chungbuk National University, Cheongju, Chungbuk 28644, Korea; kimka@chungbuk.ac.kr

[3] Namseoul University, Computer Science, Seoul 331-707, Korea; jmchung@nsu.ac.kr

* Correspondence: khryu@dblab.chungbuk.ac.kr; Tel.: +82-43-261-2254; Fax: +82-43-275-2254

Academic Editor: Marc A. Rosen
Received: 23 December 2015; Accepted: 1 March 2016; Published: 3 March 2016

Abstract: Web usage mining is a popular research area in data mining. With the extensive use of the Internet, it is essential to learn about the favorite web pages of its users and to cluster web users in order to understand the structural patterns of their usage behavior. In this paper, we propose an efficient approach to determining favorite web pages by generating large web pages, and emerging patterns of generated simple page-linked graphs. We identify the favorite web pages of each user by eliminating noise due to overall popular pages, and by clustering web users according to the generated emerging patterns. Afterwards, we label the clusters by using Term Frequency-Inverse Document Frequency (TF-IDF). In the experiments, we evaluate the parameters used in our proposed approach, discuss the effect of the parameters on generating emerging patterns, and analyze the results from clustering web users. The results of the experiments prove that the exact patterns generated in the emerging-pattern step eliminate the need to consider noise pages, and consequently, this step can improve the efficiency of subsequent mining tasks. Our proposed approach is capable of clustering web users from web log data.

Keywords: web usage mining; data mining; association rule mining; frequent pattern mining; emerging patterns; TF-IDF

1. Introduction

With the rapid growth of the Internet, most research on the Internet has revealed some very hot topics, such as social networks [1,2], web mining, and so on. In web mining, there are three categories: web content mining, web structure mining and web usage mining. In Web Usage Mining (WUM), also known as web access, web access pattern tracking can be defined as the web page history; the mining task is a process of extracting interesting patterns from web access logs. Web usage mining is still a popular research area in data mining. With the rapid growth of the Internet, more and more useful information is hidden in web log data. It is essential to learn about the favorite web pages of web users and to cluster web users in order to understand the structures that they use.

Many techniques in web usage mining have been proposed [3–8], and this field is still a hot topic for research in data mining. Most existing web mining techniques are performed based on association rule mining or frequent pattern mining, and these methods aim to find relationships among web pages or predict the behavior of web users. It is difficult to find certain groups of web users with similar favorite web pages. Furthermore, some articles about clustering web users have been published recently [6,9,10], although different clustering algorithms are used. However, all of these articles cluster web users based on frequent-pattern mining of topics in common. In generating frequent patterns, based on a user-specified minimum support threshold, the process can obtain frequent web pages for all web users. This means that if some web pages are frequently accessed by one web user, then they are accessed by other web users with a high probability. These kinds of frequently visited web pages are without discrimination in clustering web users; they are like noise pages in clustering.

The discovery of class comparison or discrimination information is an important problem in the field of data mining. Emerging patterns [11,12], defined as multivariate features where supports change significantly from one class to another, are very useful as a means of discovering distinctions between different classes of data. Using the emerging pattern—mining technique, we can find emerging web pages in web log data. This technique has been detailed in many articles [13,14], and is still a hot topic in the field of computer science. Jumping Emerging Patterns (JEPs) [15] is a special concept of EPs, which has been presented to describe some discriminating features that only occur in one class, but do not occur in other classes at all.

Term Frequency–Inverse Document Frequency (TF-IDF) [16] is a kind of weighted technology commonly used for information retrieval and information mining. Therefore, it is considered one of the measures of the importance of a document. It is widely used in research areas for classification of literature, text mining and other related fields.

Clustering algorithms are used for category based on their cluster model, and many clustering algorithms were proposed, such as the K-means algorithm [17,18], Self-Organizing Map (SOM) algorithm [19,20], Adaptive Resonance Theory (ART1) [21,22] and K-means & TF-IDF [23,24]. In this paper, accessed web pages were collected in users, they were in text form that can be defined as the identification of web users. The K-means & TF-IDF approach was used to cluster web users, because of the advantage of TF-IDF used in text mining.

Folksonomies [25] has been proposed as a collaborative way to classify online items. This kind of classification is determined by the defined frequency of user groups. In addition, many researches have been proposed based on Folksonomies [26,27] up to now. In this paper, we label the clusters based on the concept of Foksonomies.

There were many articles proposed for finding the interests of users [1,28,29]. The first paper proposed a linear regression-based method to evaluate user interest in order to calculate a similarity matrix, and to cluster web users based on a threshold according to the generated matrix. The second paper proposed an entropy-based approach to obtain user interests. The third paper proposed a community-based algorithm to retrieve user interests. The above proposed approaches ignore the characteristic of specific web pages that are frequently accessed by one user but barely accessed by others, which should be a pattern mainly considered.

In this paper, we aim to cluster web users based on user interests found in web log data. We propose an efficient approach by considering the techniques of emerging-pattern mining. In the mining task, emerging patterns of each web user are used to define the interests that are frequently accessed by one user and barely accessed by others. Through finding emerging patterns for all web users, we can discard the noise (nonessential) web pages for each web user, and cluster web users according to the generation of typical web pages.

This paper is organized as follows: The next section introduces our proposed approach; then, we implement our proposed approach with a file of web log data for evaluation; finally, we discuss our conclusions and suggest future work.

2. Proposed Approach

In this section, we generate large web pages from processed web log data, then scan and transform the clean data set into simple page-linked graphs (SPLGs), and then, generate emerging patterns in the generated SPLGs. We cluster web users based on generated emerging patterns, and finally, label the clusters with typical web pages. Our work flow is shown in Figure 1.

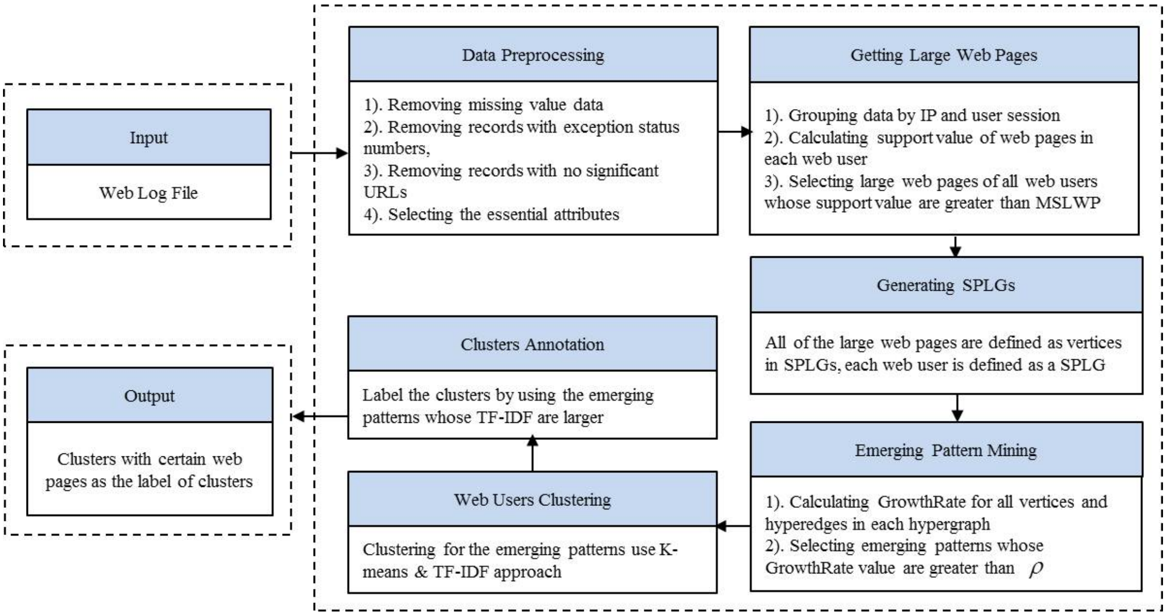

Input
Web Log File
Data Preprocessing
1). Removing missing value data
2). Removing records with exception status numbers,
3). Removing records with no significant URLs
4). Selecting the essential attributes
Getting Large Web Pages
1). Grouping data by IP and user session
2). Calculating support value of web pages in each web user
3). Selecting large web pages of all web users whose support value are greater than MSLWP
Generating SPLGs
All of the large web pages are defined as vertices in SPLGs, each web user is defined as a SPLG
Emerging Pattern Mining
1). Calculating GrowthRate for all vertices and hyperedges in each hypergraph
2). Selecting emerging patterns whose GrowthRate value are greater than ρ
Web Users Clustering
Clustering for the emerging patterns use K-means & TF-IDF approach
Clusters Annotation
Label the clusters by using the emerging patterns whose TF-IDF are larger
Output
Clusters with certain web pages as the label of clusters

Figure 1. Work flow of our proposed approach.

2.1. Preprocessing of Data Set

Web log data is automatically recorded in web log files on web servers when web users access the web server through their browsers. Not all of the records sorted into the web log files have the right format or are necessary for the mining task, so before analyzing the web log data, a data cleaning phase needs to be implemented.

2.1.1. Removing Records with Missing Value Data

Some of the records sorted in the web log file will not be complete, because some of the parameters of the records were lost. For example, if a click-through to a web page was executed while the web server was shut down, then, in the log file, only the IP address, user ID, and access time will be recorded; the method, URL, referrer, and agent are lost. This kind of record cannot be used for our mining task, so these records must be removed.

2.1.2. Removing Records with Exception Status Numbers

Some records are caused by errors in the requests or are caused by the server. Even if those records are intact, the activity did not execute normally. For example, records with status numbers 400 or 404 are caused by HTTP client errors, bad requests, or when a requested page was not found. Records with status numbers 500 or 505 are caused by HTTP server errors, in which the internal server cannot connect, or when the HTTP version is not supported. These kinds of data are not needed for our task, so the records must be removed.

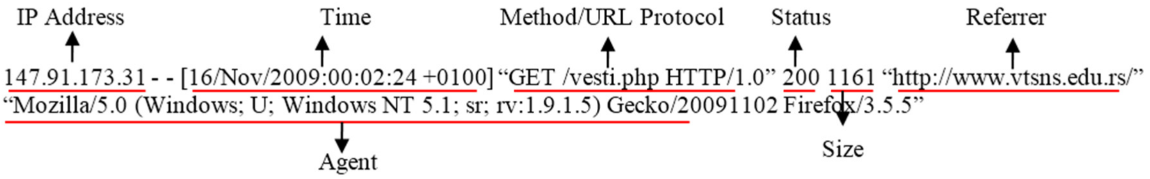
2.1.3. Removing Irrelevant Records with No Significant URLs

Some URLs in the records consist of .txt, .jpg, .gif, or .js extensions, which are automatically generated when a web page is requested. These records are irrelevant to our mining task, so they must be removed.

2.1.4. Selecting the Essential Attributes

As shown in the common log format of the web log data in Figure 2, there are many attributes in one record, but for web-usage mining, not all the attributes are necessary. In this paper, the attributes

IP address, time, URL, and referrer are essential to our task, so they must remain; the rest should be discarded.



```
147.91.173.31 - - [16/Nov/2009:00:02:24 +0100] "GET /vesti.php HTTP/1.0" 200 1161 "http://www.vtsns.edu.rs/"
"Mozilla/5.0 (Windows; U; Windows NT 5.1; sr; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5"
```

Figure 2. Common log format of web log data.

2.2. Generation of Large Web Pages

A large page set is a set of frequent web pages. We define frequent web pages as those where support thresholds are greater than, or equal to, a user-specified minimum support threshold.

In this paper, a web log file denotes a data set; Large Web Pages (LWPs) denote the set of web pages that are accessed by web users with sufficient frequency over a period of time. A special period of time called a user session is an important definition for generating LWPs for web users. Generally, the value of a user session is defined by web designers according to the desired level of security. For some websites with high security, the user session is always set to a short amount of time, such as 15 min or less, for safety. For example, a web user who is not active for a long time may have left the computer to do other things, so if someone else is using his account to do something, but the original user does not know about it, it is not safe. For other general websites, the period of a user session can be longer, such as a half hour or one hour; it can also be indefinite. If, for simplicity, the user session time is defined as one hour, in the process of generating large web pages, we should group the experimental data by periods of one hour for each web user.

After cleaning the data set, the data are sorted by the values of their IP address field, and split by user session. As a result, a session-based data set is obtained, which serves as input for our proposal. An example of a session-based data set is shown in Table 1, and a special period of time for the user session is defined as one hour. According to the session-based data, candidate large web pages of each web user are extracted, and their supports are calculated. To calculate the support count for each candidate, we need to count the visit times of each web page accessed in different user sessions for each web user. The equation of support is Equation (1), where $N_{P_{ij}}$ is the visit times of web page j in all sessions of web user i, and $N\_Session_i$ is the number of sessions for web user i. Finally, a user specified Minimum Support threshold for Large Web Page (MSLWP) must be defined. The MSLWP denotes a kind of abstract level that is a degree of generalization. The support value will be determined by the proportion of web users accessing web pages at certain times. The selection of an MSLWP is very important; if it is low, then we can obtain information for a detailed event. If it is high, then we can obtain information for general events. The pseudocode for obtaining large web pages is shown in Algorithm 1.

$$Sup_{P_{ij}} = \frac{N_{P_{ij}}}{N\_Session_i} \tag{1}$$

Considering the session-based data set of Table 1 as input to Algorithm 1, when setting the parameter value of MSLWP at 0.25, the implementation of steps 5–19 results in a candidate data set, as shown in Table 2. Afterwards, the execution of lines 20–26 results in large web pages for user 1: {p6, p12, p14, p19}.

Algorithm 1. getLWPs (List SD, double MSLWP)

```
Input: A set of session-based web data SD; a user-specified minimum support MSLWP.
Output: A set of large web pages for each web user.
Define tmp_IP = $SD_1$.IP;
Define i = 1;
Define out_LWP[][];
Define $N\_Session_i$ = 0; // initialize the number of sessions for web user i
for each sequence data $SD_n$ in SD
    if ($SD_n$.IP == tmp_IP)
        $N\_Session_i$++;
        for (int j = 1; j $\leqslant$ the number of web pages; j++)
            if ($SD_n$.URLs contain $P_{ij}$) // $P_{ij}$ is the jth web page for web user i
                $N_{P_{ij}}$++; // the visit time of web page j by web user i, add one
                break;
            end if
        end
    else
        $SD_n$.IP == tmp_IP;
        i++;
        $N\_Session_i$ = 1;
    end if
end
for each web user i
    for each web page j
        if ($N_{P_{ij}}/N\_Session_i$>=MSLWP) // check if web page j for web user i is a large web page
            out_LWP[i][j].add($P_{ij}$);
        end if
    end
end
```

Table 1. Example of session-based data set.

IP Address	Session_ID	URLs
Web user 1	1	p1, p2, p6
Web user 1	2	p3, p5, p6
Web user 1	3	p6, p4
Web user 1	4	p6, p12
Web user 1	5	p11, p12
Web user 1	6	p6, p9, p18, p19
Web user 1	7	p12, p13
Web user 1	9	p12, p14
Web user 1	12	p14, p13, p15
Web user 1	13	p10, p14
Web user 1	14	p14, p17
Web user 1	17	p7, p19
Web user 1	18	p19, p16, p20
Web user 1	21	P8, p19

Table 2. Candidate large web pages in example data set.

URL	Support Count
p1	1
p2	1
p3	1
p4	1
p5	1
p6	5
p7	1
p8	1
p9	1
p10	1
p11	1
p12	4
p13	2
p14	4
p15	1
p16	1
p17	1
p18	1
p19	4
p20	1

2.3. Generation of Simple Page Linked-Graph (SPLG)

After generating large web pages for each web user, all of the large web pages are defined as vertices in the SPLG.

In regular page-linked graphs, each edge consists of every two web pages that are contained in one session. An example of a page link graph for web user 1 is shown in Figure 3 (left). However, in a SPLG, each edge consists of every two large web page of the web user. Applying the concept of the SPLG to the structure of web page links can reduce large and complex regular page-linked graphs to simple ones in order to reduce noise web pages. In the SPLG, links between each of the two large web pages should be checked. To check the link between every two vertices, the direction of link does not need to be considered, if the two vertices are visited by one user in one session, then they are connected. The pseudocode for checking the links is shown in Algorithm 2.

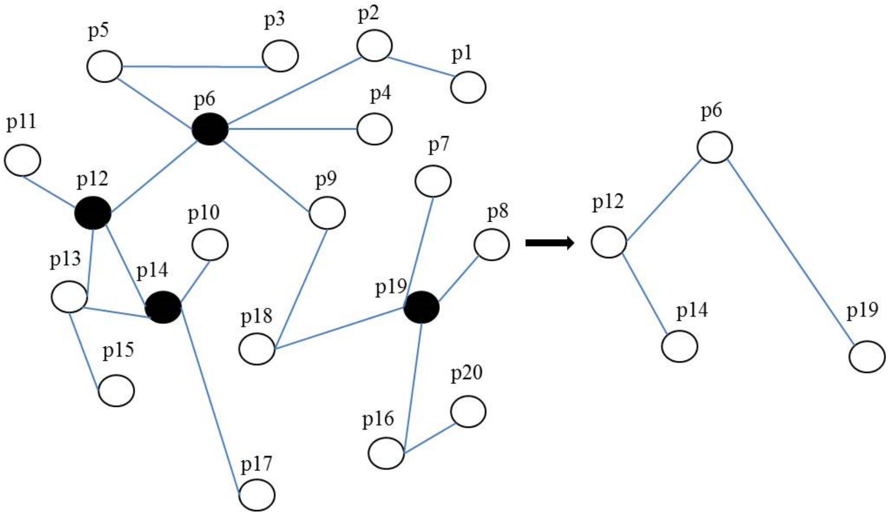


Figure 3. The simple page-linked graphs (SPLG) of web user 1.

Algorithm 2. checkLinks (List SD, String [][] LWP)

```
Input: A set of session-based web data SD; a set of large web pages LWP[i][j], where i is the index
       of web users, and j is the index of large web pages for user i
Output: A set of links with IP address.
Define flag_Link = 1;
Define HashMap out_List;
for each web user i
    for each large web page j
        if each two large web pages LWP[i][m] and LWP[i][k] are both included in SD$_i$
            flag_Link = 1; // a link between LWP[i][m] and LWP[i][k] is found
        Else
            flag_Link = 0; // no link between LWP[i][m] and LWP[i][k]
        end if
        if (flag_Link == 1)
            out_List.put(SDi.IP, LWP[i][m]:LWP[i][k]);
        end if
    end
end
```

After generating all of the links, the generated links with the same IP address are grouped and linked with the same vertices. Then, the SPLGs of all web users can be generated. For example, for the experimental data set in Table 1 for web user 1, who visited 20 web pages {p1, p2... p20} in 14 user sessions, we define the MSLWP as 0.25, and the large web pages of user 1 are {p6, p12, p14, p19}, which was generated in the previous section. After implementing Algorithm 2, links {(web user 1, [p6, p12]), (web user 1, [p6, p19]), (web user 1, [p12, p14])} are obtained, and then the SPLG of web user 1 can be described as shown in Figure 3 (right).

2.4. Generation of Emerging Patterns

After generating SPLGs for all web users, we try to find emerging patterns in these SPLGs. Examples of SPLGs for some web users are shown in Figure 4.

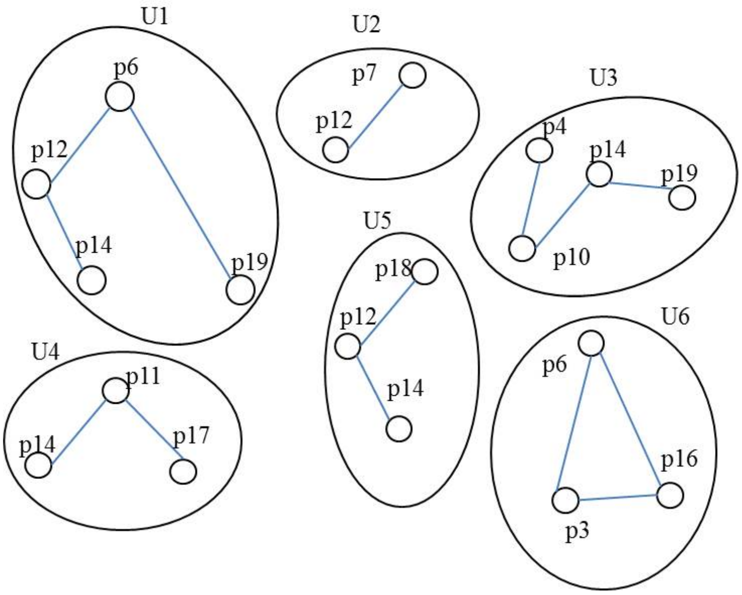


Figure 4. Example SPLGs of some web users.

In the process of emerging pattern mining, we will use the ideas of ρ-EP [30] and JEP [31,32]. For example, we set the SPLG for web user U1 as class 1, and set other SPLGs for other web users as class 2. Table 3 shows the web pages in these two classes. Table 4 lists all the possible EPs, their support, and the growth rates of all EPs. The equation of support is Equation (2), and the equation for the growth rate is Equation (3), where N_{p_i} is the number of patterns (p_i) in the class, N is the number of all patterns in the class, $Support(Class1)$ is the support value of class 1, and $Support(Class2)$ is the support value of class 2. If we set the minimum growth rate threshold, ρ, to 1.5, there are nine EPs in class 1: four normal EPs ({p6}, {p12}, {p19} and {p12, p14}), where the value of the growth rate is greater than the specific value of ρ, and five JEPs ({p6, p12}, {p6, p19}, {p6, p12, p14}, {p6, p12, p19} and {p6, p12, p14, p19}) where the value of the growth rate is infinite.

$$Support(p_i) = \frac{N_{p_i}}{N} \tag{2}$$

$$GrowthRate(Class1) = \frac{Support(Class1)}{Support(Class2)} \tag{3}$$

Table 3. Sample dataset split into two classes.

Class 1	Class 2
p6	p3
p12	p4
p14	p6
p19	p7
p6, p12	p10
p6, p19	p11
p12, p14	p12
p6, p12, p14	p14
p6, p12, p19	p16
p6, p12, p14, p19	p17
	p18
	p19
	p7, p12
	...

Table 4. Discovering emerging patterns (EPs) with ρ = 1.5.

Web Pages of U1	Support		Growth Rate of EPs	
	Class 1	Class 2	Class 1	Class 2
p6	0.6	0.09	6.67	-
p12	0.6	0.15	4	-
p14	0.4	0.35	1.14	-
p19	0.4	0.12	3.33	-
p6, p12	0.4	0	∞	-
p6, p19	0.3	0	∞	-
p12, p14	0.3	0.06	5	-
p6, p12, p14	0.2	0	∞	-
p6, p12, p19	0.2	0	∞	-
p6, p12, p14, p19	0.1	0	∞	-

JEPs {p6, p12}, {p6, p19}, {p6, p12, p14}, {p6, p12, p19} and {p6, p12, p14, p19} are the JEPs for class 1, with support of non-zero values in class 1, and zero in class 2. It can be seen that these JEPs appear many times in class 1, but never in class 2; so, these different values can be usefully implemented to distinguish different favorite web pages from different web users.

2.5. Clustering of Web Users

In this paper, we execute a K-means clustering algorithm [24,25] on emerging patterns to cluster the web users. First, we generate a TF-IDF based weighted matrix which can reflect how important a web page is to a web user. In the process of matriculation, a TM matrix is defined as U by P, where U is the number of web users, P is the number of web pages that are emerging patterns of all web users, and $\mathrm{TM_{ij}}$ represents a measure of the TF-IDF weighted value for web page j visited by web user i, $i \in [1, U]$ and $j \in [1, \mathrm{P}]$. According to Equation (4), we can get the value of the TF-IDF for web page j of web user i, and then the TF-IDF based TM matrix can be obtained. Then, we execute the K-means algorithm on the generated TM with a specified K value to get the clusters of web users:

$$TM_{ij} = \frac{n_{ij}}{\sum\limits_{k=1}^{U} n_{kj}} \times \log\frac{|U|}{\left|\left\{\mathrm{u}_i \in \mathrm{U} : \mathrm{p}_j \in u_i\right\}\right|} \tag{4}$$

where n_{ij} is the number of occurrences of web page j for user i, $\sum\limits_{k=1}^{U} n_{kj}$ is the number of occurrences of web page j for all web users, $|U|$ is the number of web users, and $|u_i \in U : p_j \in u_i|$ is the number of web users who accessed web page j.

2.6. Annotation of Clusters

After clustering, we label the clusters based on the concept of Folksonomies. Each cluster is defined as one user group, and the web pages in each cluster are defined as online items, we use TF-IDF to calculate the frequency of each web page in each cluster. According to Equation (5), we can calculate the TF-IDF value of each web page in each cluster, and then we can select some web pages where TF-IDF values are among the Top N (N can be the number chosen by a user with freedom, where N is smaller than the number of web pages in each cluster) and the largest in each cluster is the label of this cluster:

$$T_{ij} = \frac{n_{ij}}{\sum\limits_{k=1}^{K} n_{kj}} \times \log\frac{|K|}{\left|\left\{c_i \in \mathrm{C} : \mathrm{p}_j \in c_i\right\}\right|} \tag{5}$$

where n_{ij} is the number of occurrences of web page j in cluster i, $\sum\limits_{k=1}^{K} n_{kj}$ is the number of occurrences of web page j in all the clusters, $|K|$ is the number of clusters, and $|c_i \in C : p_j \in c_i|$ is the number of clusters that contain web page j.

3. Experiments and Analysis

Based on the proposed approach presented in this paper, we performed experiments on a set of web log data to evaluate its efficiency.

3.1. Experimental Data Set

In the experiments, we used a web log file from the web site www.vtsns.edu.rs as the experimental data. There were 5999 records in the raw file. After data cleaning, there were 1222 records left from 243 user sessions. There were 31 different kinds of web pages accessed during the user sessions in these data.

3.2. Experimental Results and Discussions

This section shows several experiments based on our proposed approach. In the first subsection, we try to analyze the effects of the parameters (MSLWP and ρ) used in our proposed approach. We implemented our approach to show the results from generating emerging patterns, clustering, and the annotation of clusters.

3.2.1. Analysis of the Parameters

In the process of getting large web pages, we extracted events that satisfy a user-defined Minimum Support of Large Web Pages (MSLWP). We can discard infrequent events to reduce the size of the experimental database, reduce the search space and time, and maintain the accuracy of the whole mining task. To evaluate the effect of the MSLWP parameter, we compared the number of web users who have large web pages, and generated large web pages by changing the values of MSLWP. The experimental results are shown in Figures 5 and 6. We can see that the bigger the MSLWP, the fewer generated web users and large web pages. There always exists a value for MSLWP, and from this value, the number of web users and large web pages will either not change at all, or change by a negligible amount. This value is always selected as an empirically suitable value for the MSLWP parameter in the whole approach. In this experiment, we can see that when the value of MSLWP is 0.6, the number of web users and large web pages sees a small decline. Consequently, we always choose this value as the value for MSLWP in the performance tests. In addition, from Figure 5, we can also see that comparing the number of web users after executing the proposed approach with the number of records after data cleaning, it becomes clear that our proposed approach can greatly eliminate the noise pages of web users from the data set to improve efficiency.

In the process of generating emerging patterns, we tried to find patterns where growth rates satisfy a user-defined ρ value. This can be a criterion for the selection of emerging patterns. To evaluate the effect of the ρ parameter, we first defined the value of MSLWP as 0.6 (or 60%), then we compared the number of emerging patterns by changing the values of in different web users. The experimental results are shown in Figure 7. We can see that the bigger the value of ρ, the fewer emerging patterns are generated for each web user. There always exists a value of ρ, and from this value, the number of emerging patterns will not change, or will change very little. This value is always selected for use as the value of the minimum growth rate in the experiment. From the result, we can also see that the number of emerging patterns saw a small decline, compared to increases in the value of ρ.

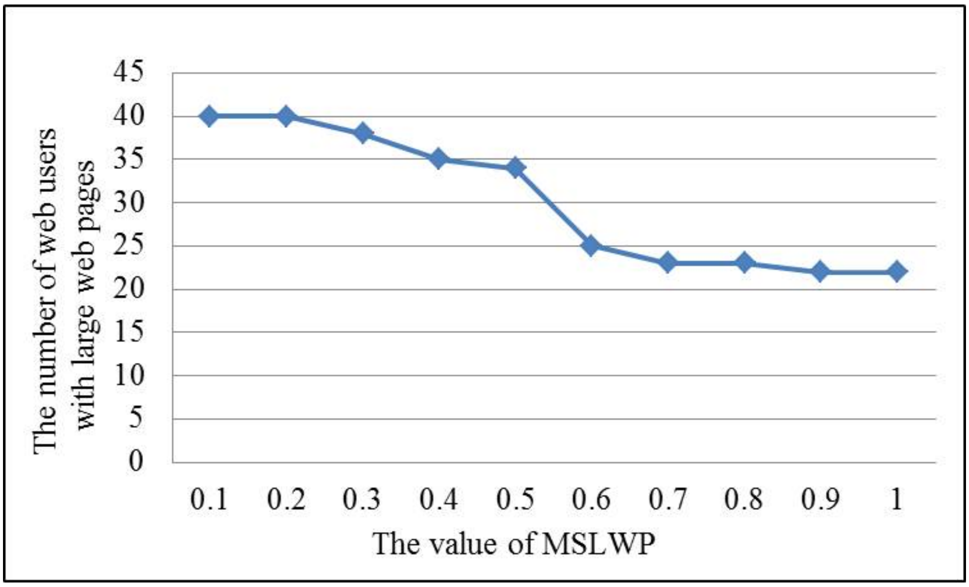


Figure 5. Effect of parameter MSLWP related to the number of web users.

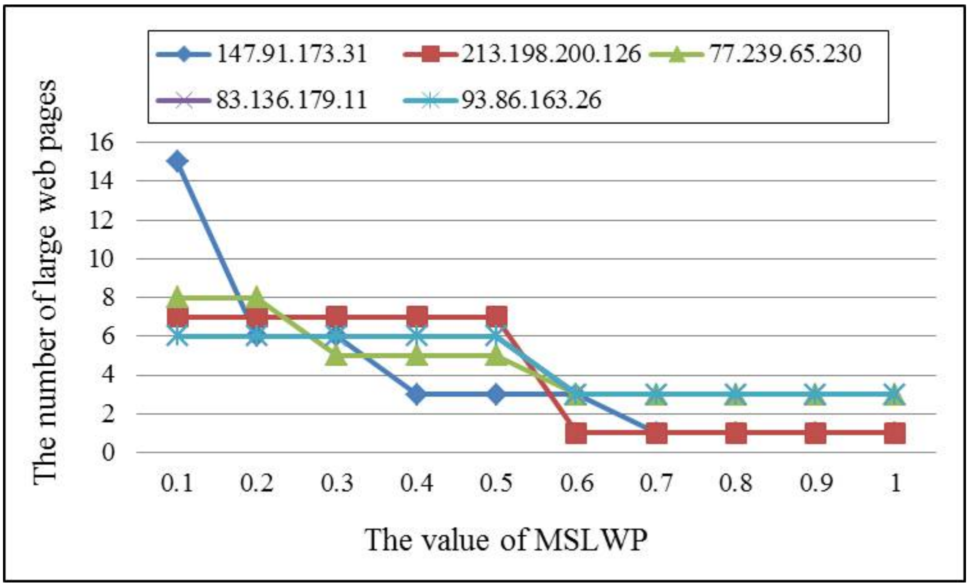


Figure 6. Effect of parameter MSLWP related to the number of large web pages.

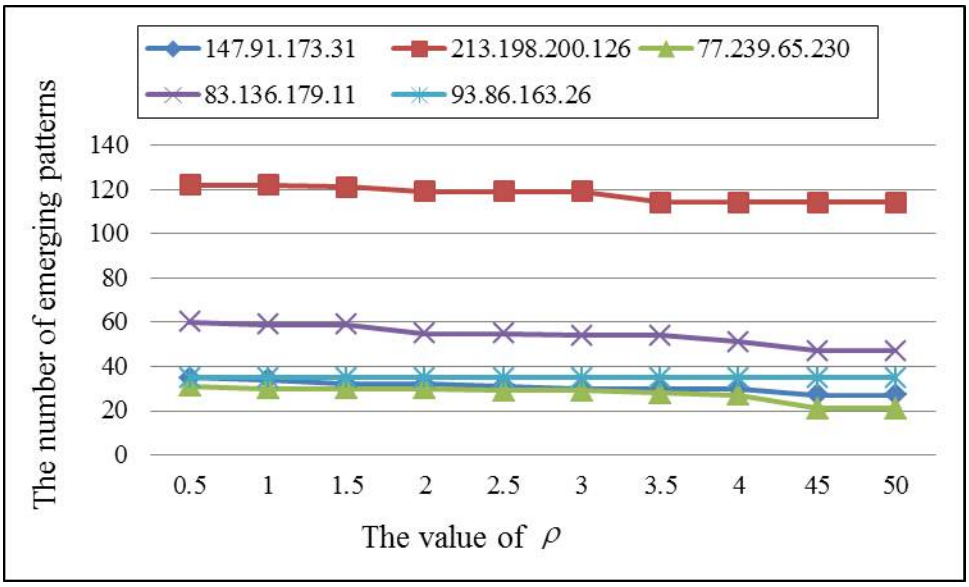


Figure 7. Effect of parameter ρ in sample web users.

Then, we tried to analyze the relationship between MSLWP and the number of emerging patterns of web users. We defined the value of ρ at 1.5. The result is shown in Figure 8. From the result, we can see that the bigger the MSLWP, the fewer generated emerging patterns in web users. There always exists a value of MSLWP, and from this value, the number of web users and large web pages will not change, or will change very little. This value is always selected for use as the value of MSLWP in the whole approach. In addition, the number of emerging patterns saw a big decline compared to increases in the value of MSLWP. From the experiments, we know that MSLWP is the main parameter to control the number of web users with favorite web pages, and ρ is the minor parameter to calibrate.

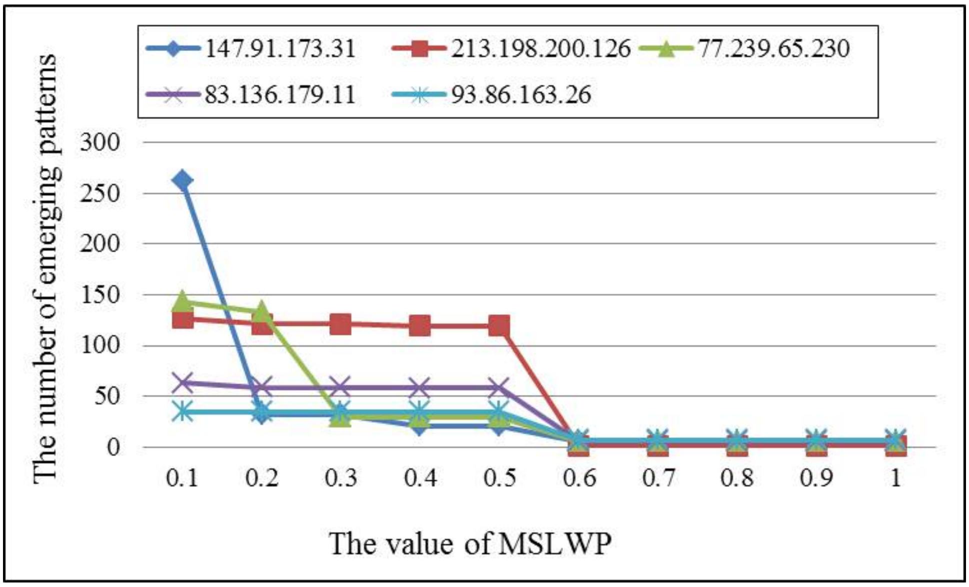


Figure 8. Effect of parameter Minimum Support threshold for Large Webpage (MSLWP) related to the number of emerging patterns.

3.2.2. Results of Emerging Patterns

In this section, we executed our proposed approach and show the results of generated emerging patterns with the parameter MSLWP at 0.6 and ρ at 1.5. The result is shown in Table 5. From the result, we can see that there are 25 web users who exhibit some emerging patterns. From this result, we can see that there are few emerging patterns for each web user, which are represented as the favorite web pages of those web users, which is very efficient for using these typical records to cluster web users.

Table 5. Generation of emerging patterns with MSLWP = 0.6 and ρ = 1.5.

Web Users	Emerging Patterns	Growth Rate	Web Users	Emerging Patterns	Growth Rate
109.93.18.182	/oglasna.php	2.43	78.30.166.209	/oglasna.php	3.46
147.91.173.31	/profesor.php,	∞		/ispit_rezultati.php	8.51
	/ispit_rezultati.php	2.41		[/oglasna.php, /ispit_rezultati.php]	55.33
	[/profesor.php, /ispit_rezultati.php]	∞	79.101.207.149	/index.php	∞
	[/profesor.php, /oglasna.php]	∞	79.101.254.115	/raspored_predavanja.php	∞
	[/ispit_rezultati.php, /oglasna.php]	∞	79.101.85.80	/oglasna.php	5.00
	[/profesor.php, /ispit_rezultati.php, /oglasna.php]	∞	80.74.170.57	/oglasna.php	5.00
160.99.81.166	/galerija.php	425.00	81.18.55.98	/oglasna.php	5.00
188.246.68.98	/ispit_rezultati.php	35.42	82.117.202.158	/ispiti.php	6.38
194.106.160.113	/ispiti.php	1.61		/ispit_rezultati.php	3.69
	/ispit_raspored_akt.php	1.74		[/ispiti.php, /ispit_rezultati.php]	∞
	[/ispiti.php, /ispit_raspored_akt.php]	2.51	82.208.207.41	/ispit_odbijeni_spisak.php	∞
	[/ispiti.php, /oglasna.php]	2.51		/galerija.php	5.53
	[/ispit_raspored_akt.php, /oglasna.php]	2.82		[/ispit_odbijeni_spisak.php, /galerija.php]	∞

Table 5. *Cont.*

Web Users	Emerging Patterns	Growth Rate	Web Users	Emerging Patterns	Growth Rate
213.198.193.159	[/ispiti.php, /ispit_raspored_akt.php, /oglasna.php]	2.82	83.136.179.11	/rezultati_ispita.php	∞
	/ispit_raspored_god.php	∞		/ispit_rezultati.php	1.50
	[/ispiti.php, /ispit_raspored_god.php]	∞		[/oglasna.php, /rezultati_ispita.php]	∞
	[/oglasna.php, /ispit_raspored_god.php]	∞		[/oglasna.php, /ispit_rezultati.php]	5.64
	[/ispiti.php, /oglasna.php, /ispit_raspored_god.php]	∞		[/rezultati_ispita.php, /ispit_rezultati.php]	∞
213.198.200.126	/oglasna.php	5.00		[/oglasna.php, /rezultati_ispita.php, /ispit_rezultati.php]	∞
217.24.28.100	/ispit_rezultati.php	2.63	89.216.48.5	/ispiti.php	1.61
77.105.30.146	/oglasna.php	5.00	89.216.77.252	/ispit_raspored_akt.php	30.91
77.239.65.230	/ispiti.php	3.76	93.86.163.26	/admin/user.php	∞
	/ispit_raspored_akt.php	4.10		/noviSajt/vtsns_admin/amfphp/gateway.php	∞
	[/ispiti.php, /ispit_raspored_akt.php]	6.45		/noviSajt/administrator/history/historyFrame.html	∞
	[/ispiti.php, /oglasna.php]	4.23		[/admin/user.php, /noviSajt/vtsns_admin/amfphp/gateway.php]	∞
	[/ispit_raspored_akt.php, /oglasna.php]	4.84		[/admin/user.php, /noviSajt/administrator/history/historyFrame.html]	∞
	[/ispiti.php, /ispit_raspored_akt.php, /oglasna.php]	4.84		[/noviSajt/vtsns_admin/amfphp/gateway.php, /noviSajt/administrator/history/historyFrame.html]	∞
77.239.68.36	/ispiti.php	1.61		[/admin/user.php, /noviSajt/vtsns_admin/amfphp/gateway.php, /noviSajt/administrator/history/historyFrame.html]	∞
	/ispit_raspored_akt.php	1.74	94.189.221.5	/ispiti.php	1.61
	[/ispiti.php, /ispit_raspored_akt.php]	2.51		/ispit_raspored_akt.php	1.74
	[/ispiti.php, /oglasna.php]	2.51		[/ispiti.php, /ispit_raspored_akt.php]	2.51
	[/ispit_raspored_akt.php, /oglasna.php]	2.82		[/ispiti.php, /oglasna.php]	2.51
	[/ispiti.php, /ispit_raspored_akt.php, /oglasna.php]	2.82		[/ispit_raspored_akt.php, /oglasna.php]	2.82
78.30.134.166	/oglasna.php	5.00		[/ispiti.php, /ispit_raspored_akt.php, /oglasna.php]	2.82

3.2.3. Results of Clustering and Annotation

In this section, we executed our proposed approach with parameter MSLWP at 0.6 and ρ at 1.5 to see the result of clustering. When we set the number of clusters at 5, the clustering result is shown in Table 6. Then, we label the clusters by calculating the TF-IDF value of emerging patterns in each cluster according to Equation (5), and the result of TF-IDF of emerging patterns is shown in Table 7. We choose the Top-2 TF-IDF emerging patterns as the labels, and the labeled results are shown in Table 8. From the results of clusters, we can clearly understand the structure of web users who visited web site www.vtsns.edu.rs. For example, the users in cluster 3 frequently visit web pages /ispiti.php and /ispit_raspored_akt.php as their favorite web pages. In contrast, the users in cluster 5 frequently visit web pages /oglasna.php and /raspored_predavanja.php as their favorite web pages. We can design the web page to recommend some favorite web pages to those web users who are in the same clusters.

Table 6. Result of clustering of web users.

Clusters	Web Users
Cluster 1	79.101.207.149
Cluster 2	213.198.193.159
Cluster 3	147.91.173.31 194.106.160.113 77.239.65.230 77.239.68.36 82.117.202.158 82.208.207.41 83.136.179.11 89.216.48.5 89.216.77.252 93.86.163.26 94.189.221.5
Cluster 4	188.246.68.98 217.24.28.100
Cluster 5	109.93.18.182 160.99.81.166 213.198.200.126 77.105.30.146 78.30.134.166 78.30.166.209 79.101.254.115 79.101.85.80 80.74.170.57 81.18.55.98

Table 7. Generation of five clusters with their annotation.

TF-IDF for Each Cluster	
Cluster 1	**/index.php = 1.609438**
Cluster 2	**[/ispiti.php,/oglasna.php,/ispit_raspored_god.php] = 0.4023595**, **[/oglasna.php,/ispit_raspored_god.php] = 0.4023595**, /ispit_raspored_god.php = 0.4023595 [/ispiti.php,/ispit_raspored_god.php] = 0.4023595
Cluster 3	**/ispiti.php = 0.18934564, /ispit_raspored_akt.php = 0.15778804**, [/ispit_raspored_akt.php,/oglasna.php] = 0.12623043, [/ispiti.php,/oglasna.php] = 0.12623043, [/ispiti.php,/ispit_raspored_akt.php] = 0.12623043, [/ispiti.php,/ispit_raspored_akt.php,/oglasna.php] = 0.12623043, /admin/user.php = 0.03155761, [/admin/user.php,/noviSajt/vtsns_admin/amfphp/gateway.php,/noviSajt/administrator/history/historyFrame.html = 0.03155761, [/profesor.php,/ispit_rezultati.php] = 0.03155761, [/noviSajt/administrator/history/historyFrame.html] = 0.03155761, [/ispit_odbijeni_spisak.php,/galerija.php] = 0.03155761, [/ispit_rezultati.php,/oglasna.php] = 0.03155761, [/admin/user.php,/noviSajt/administrator/history/historyFrame.html] = 0.03155761, [/noviSajt/vtsns_admin/amfphp/gateway.php,/noviSajt/administrator/history/historyFrame.html] = 0.03155761, [/rezultati_ispita.php,/ispit_rezultati.php] = 0.03155761, [/noviSajt/vtsns_admin/amfphp/gateway.php] = 0.03155761, [/ispit_odbijeni_spisak.php] = 0.03155761, [/profesor.php,/ispit_rezultati.php,/oglasna.php] = 0.03155761, /rezultati_ispita.php = 0.03155761, [/admin/user.php,/noviSajt/vtsns_admin/amfphp/gateway.php] = 0.03155761, [/oglasna.php,/rezultati_ispita.php,/ispit_rezultati.php] = 0.03155761, [/ispiti.php,/ispit_rezultati.php] = 0.03155761, /profesor.php = 0.03155761, [/profesor.php,/oglasna.php] = 0.03155761, /ispit_rezultati.php = 0.030048564, [/oglasna.php,/ispit_rezultati.php] = 0.017966487, /galerija.php = 0.017966487
Cluster 4	**/ispit_rezultati.php = 0.5108256**
Cluster 5	**/oglasna.php = 1.0729587, /raspored_predavanja.php = 0.13411984**, /galerija.php = 0.076357566, [/oglasna.php,/ispit_rezultati.php] = 0.076357566, /ispit_rezultati.php = 0.0425688

Table 8. Result of clusters with labels.

Clusters	Labels
Cluster 1	/index.php
Cluster 2	[/ispiti.php,/oglasna.php,/ispit_raspored_god.php], [/oglasna.php,/ispit_raspored_god.php]
Cluster 3	/ispiti.php, /ispit_raspored_akt.php
Cluster 4	/ispit_rezultati.php
Cluster 5	/oglasna.php, /raspored_predavanja.php

3.2.4. Comparison with Existing Approaches

In this section, we executed our proposed approach on the experimental data set with the parameter MSLWP at 0.6 and ρ at 1.5 and compared it with existing approaches of generation of user interests by Zheng and Zhang [28], Xu *et al.* [29] and Tchuente *et al.* [1]. After executing the approaches, we used Purity to evaluate clustering. According to Equation (6), we can calculate the Purity value with different numbers of clusters, where $\Omega = \{w_1, w_2, ..., w_k\}$ is the set of clusters and $C = \{c_1, c_2, ..., c_j\}$ is the set of classes. In this experiment, the results of clustering are defined as the set of clusters, and the results of annotation can be structured as the set of classes, C, with different kinds of web pages.

$$purity(\Omega, C) = \frac{1}{N}\sum_{k} \max_j |w_k \cap c_j| \tag{6}$$

The comparison is shown in Figure 9. From the result, we can see that our proposed approach performed better than the other three existing approaches with greater purity in the clusters. In particular, our proposed approach is outstanding when the number of clusters is five; therefore, we can also say that it is most correct to group web users in this web log data into five clusters.

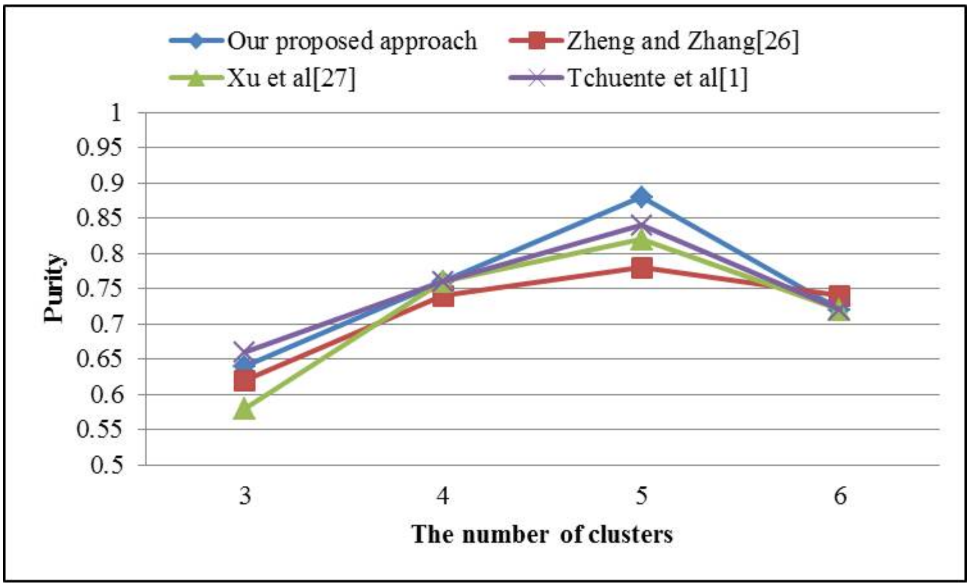


Figure 9. Comparison of generation of user interests with existing approaches.

Then we executed different clustering algorithms on the data set of user interests which is generated by emerging pattern mining technique with the parameter MSLWP at 0.6 and ρ at 1.5. The result is shown in Figure 10, from the result we can see that, K-means & TF-IDF clustering approach performed better than K-means, SOM and ART1 algorithms with greater purity in the clusters. We can also see that TF-IDF is good for information mining on the data set in text form.

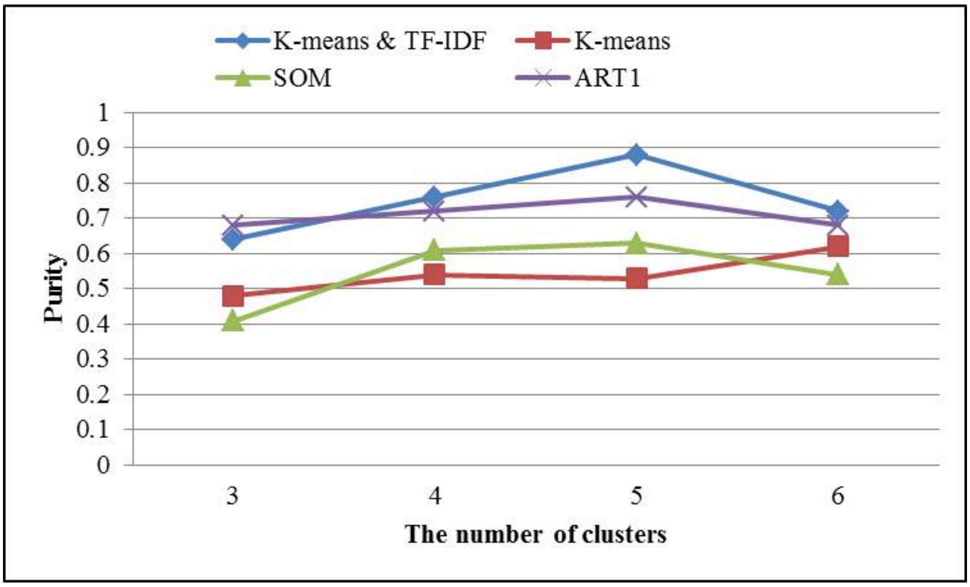


Figure 10. Comparison of clustering with existing approaches.

4. Conclusions and Future Work

In this study, we not only proposed a process for getting large web pages from processed web log data, but also defined these large web pages as vertices, and transformed the session-based data set into SPLGs; we then found emerging patterns in these SPLGs. Afterwards, we clustered the web users and labeled the clusters by considering TF-IDF. The main result of this study is to generate large web pages and emerging patterns to identify the personal favorite web pages of each user by eliminating noise due to overall popular pages. In the experiments, we evaluated the parameters used in our proposed approach and discussed the effect of the parameters on generating emerging patterns. The results of the experiments have proven that the exact patterns generated in the emerging-pattern step eliminated the need to consider noise pages. Consequently, we found that the efficiency of subsequent mining tasks can be improved.

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grant funded by Basic Science Research Program through the National Research Foundation of Korea (NRF) (No. 2013R1A2A2A01068923) and the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1013) supervised by the IITP (Institute for Information & communication Technology Promotion).

Author Contributions: Xiuming Yu and Meijing Li developed the concept of this proposed approach, implemented the experiment, and drafted the manuscript. Kyung Ah Kim and Jimoon Chung analyzed the results of the experiments and provided valuable comments that helped interpret the results. Keun Ho Ryu revised the manuscript and supervised the overall work. All authors read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tchuente, D.; Canut, M.F.; Jessel, N.; Peninou, A.; Sédes, F. A community-based algorithm for deriving users' profiles from egocentrics networks: experiment on Facebook and DBLP. *Soc. Netw. Anal. Min.* **2013**, *3*, 667–683. [CrossRef]
2. Tchuente, D.; Canut, M.F.; Jessel, N.B.; Péninou, A.; Sédes, F. Visualizing the relevance of social ties in user profile modeling. *Web Intell. Agent Syst. Int. J.* **2012**, *10*, 261–274.

3. Yu, X.; Li, M.; Kim, H.; Lee, D.G.; Park, J.S.; Ryu, K.H. A novel approach to mining access patterns. In Proceedings of the 2011 3rd International Conference on Awareness Science and Technology (iCAST), Dalian, China, 27–30 September 2011; pp. 350–355.
4. Yu, X.; Li, M.; Lee, D.G.; Kim, K.D.; Ryu, K.H. Application of closed gap-constrained sequential pattern mining in web log data. In *Advances in Control and Communication*; Springer Berlin Heidelberg: Berlin, Germany; Heidelberg, Germany, 2012; pp. 649–656.
5. Li, M.; Yu, X.; Ryu, K.H. MapReduce-based web mining for prediction of web-user navigation. *J. Inf. Sci.* **2014**, *40*, 557–567. [CrossRef]
6. Parekh, A.; Patel, A.; Parmar, S.; Patel, V. Web usage Mining: Frequent Pattern Generation using Association Rule Mining and Clustering. *Int. J. Eng. Res. Technol.* **2015**. [CrossRef]
7. Wang, H.; Yang, C.; Zeng, H. Design and Implementation of a Web Usage Mining Model Based On Upgrowth and Preflxspan. *Commun. IIMA* **2015**, *6*. Article 10.
8. Lopes, P.; Roy, B. Recommendation System using Web Usage Mining for users of E-commerce site. *Int. J. Eng. Res. Technol.* **2014**, *3*, 1714–1720.
9. Roul, R.K.; Varshneya, S.; Kalra, A.; Sahay, S.K. A Novel Modified Apriori Approach for Web Document Clustering. *Comput. Intell. Data Min.* **2015**, *3*, 159–171.
10. Wan, M.; Jönsson, A.; Wang, C.; Li, L.; Yang, Y. Web user clustering and Web prefetching using Random Indexing with weight functions. *Knowl. Inf. Syst.* **2012**, *33*, 89–115. [CrossRef]
11. Dong, G.; Zhang, X.; Wong, L.; Li, J. CAEP: Classification by aggregating emerging patterns. In *Discovery Science*; Springer Berlin Heidelberg: Berlin, Germany; Heidelberg, Germany, 1999; pp. 30–42.
12. Li, G.; Law, R.; Vu, H.Q.; Rong, J.; Zhao, X. Identifying emerging hotel preferences using Emerging Pattern Mining technique. *Tour. Manag.* **2015**, *46*, 311–321. [CrossRef]
13. Sherhod, R.; Judson, P.N.; Hanser, T.; Vessey, J.D.; Webb, S.J.; Gillet, V.J. Emerging Pattern Mining to Aid Toxicological Knowledge Discovery. *J. Chem. Inf. Model.* **2014**, *54*, 1864–1879. [CrossRef] [PubMed]
14. Yu, Y.; Yan, K.; Zhu, X.; Wang, G.; Luo, D.; Sood, S. Mining Emerging Patterns of PIU from Computer-Mediated Interaction Events. In *Agents and Data Mining Interaction*; Springer Berlin Heidelberg: Berlin, Germany; Heidelberg, Germany, 2014; pp. 66–78.
15. Dong, G.; Li, J. Efficient mining of emerging patterns: Discovering trends and differences. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999; pp. 43–52.
16. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 21 June–18 July 1965 and 27 December 1965–7 January 1966; University of California Press: Berkeley, CA, USA, 1967; Volume 1, pp. 281–297.
17. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1979**, *28*, 100–108. [CrossRef]
18. Wang, J.; Wang, J.; Ke, Q.; Zeng, G.; Li, S. Fast approximate k-means via cluster closures. In *Multimedia Data Mining and Analytics*; Springer: New York, NY, USA, 2015; pp. 373–395.
19. Cottrell, M.; Fort, J.C.; Pagès, G. Theoretical aspects of the SOM algorithm. *Neurocomputing* **1998**, *21*, 119–138. [CrossRef]
20. Olteanu, M.; Villa-Vialaneix, N. Sparse Online Self-Organizing Maps for Large Relational Data. In *Advances in Self-Organizing Maps and Learning Vector Quantization*; Springer: New York, NY, USA, 2016; pp. 73–82.
21. Carpenter, G.A.; Grossberg, S. Adaptive resonance theory. In *Encyclopedia of Machine Learning*; Springer US: New York, NY, USA, 2010; pp. 22–35.
22. Ramya, C.; Shreedhara, K.S. Clustering of Web Users using ART1 NN based Clustering Approach with a Complete Preprocessing Methodology. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 71–77.
23. Gowtham, S.; Goswami, M.; Balachandran, K.; Purkayastha, B.S. An Approach for Document Pre-processing and K Means Algorithm Implementation. In Proceedings of the 2014 Fourth International Conference on Advances in Computing and Communications (ICACC), Cochin, India, 27–29 August 2014; pp. 162–166.
24. Chen, N.; Xu, Z.S.; Xia, M.M. Hierarchical hesitant fuzzy K-means clustering algorithm. *Appl. Math. A J. Chin. Univ.* **2014**, *29*, 1–17. [CrossRef]
25. Lambiotte, R.; Ausloos, M. Collaborative tagging as a tripartite network. In *Computational Science–ICCS 2006*; Springer Berlin Heidelberg: Berlin, Germany; Heidelberg, Germany, 2006; pp. 1114–1117.

26. Gruber, T. Ontology of folksonomy: A mash-up of apples and oranges. *Int. J. Semant. Web Inf. Syst. (IJSWIS)* **2007**, *3*, 1–11. [CrossRef]
27. Dotsika, F. Uniting formal and informal descriptive power: Reconciling ontologies with folksonomies. *Int. J. Inf. Manag.* **2009**, *29*, 407–415. [CrossRef]
28. Zheng, W.; Zhang, M. The investigation for Web user clustering based on interest. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 September 2011; pp. 553–556.
29. Xu, C.; Chen, S.; Cheng, J. Network User Interest Pattern Mining Based on Entropy Clustering Algorithm. In Proceedings of the 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Xi'an, China, 17–19 September 2015; pp. 200–204.
30. Wu, H.C.; Luk, R.W.P.; Wong, K.F.; Kwok, K.L. Interpreting tf-idf term weights as making relevance decisions. *ACM Trans. Inf. Syst. (TOIS)* **2008**, *26*. Article 13. [CrossRef]
31. Bailey, J.; Manoukian, T.; Ramamohanarao, K. Fast algorithms for mining emerging patterns. In *Principles of Data Mining and Knowledge Discovery*; Springer Berlin Heidelberg: Berlin, Germany; Heidelberg, Germany, 2002; pp. 39–50.
32. Fan, H.; Kotagiri, R. An efficient single-scan algorithm for mining essential jumping emerging patterns for classification. In *Advances in Knowledge Discovery and Data Mining*; Springer Berlin Heidelberg: Berlin, Germany; Heidelberg, Germany, 2002; pp. 456–462.