A social network algorithm for detecting communities from weighted graph in Web Usage Mining system

Yacine SLIMANI and Abdelouahab MOUSSAOUI

Laboratoire de Recherche en Informatique Appliquée LRIA – UFAS, Algeria
slimany09@gmail.com , moussaoui.abdel@gmail.com

***Abstract*— Web Usage Mining is the process of discovering user's navigation pattern and predicting user's behavior. The quantity of the Web usage data to be analyzed and its low quality are the principal problems in WUM. Several algorithms of data mining have been applied in order to extract the behaviors of the Web sites' users. In this present work, we have implemented a community detection technique in WUM process that is based on the modularity function and we have organized the preprocessed data as a weighted graph. The obtained results illustrate the aptitude of the proposed algorithm to determine a pertinent design of the web site from the discovered communities.**

***Keywords*— Data Mining, Web Usage Mining, log files, community discovery, weighted graph, social network, modularity.**

1. Introduction

One of the most significant axes of the Web Mining is the Web Usage Mining (WUM) which is interested in the extraction of the access pattern to the Web from the used data. The principal interest of the Web Usage Mining is that it provides information on the way in which the users browse the Web site [1].

In this work, we are interested in the analysis of the user browsing behavior. The objective is to understand the navigational practices of users (teachers, students and administrative staff).

Cooley [2] divides the WUM in three main steps: preprocessing, pattern discovery and pattern analysis. The preprocessing task within the WUM process involves cleaning and structuring data to prepare it for the pattern discovery task. In the phases of discovered and analyzes knowledge, the Web Usage Mining represents a field of research to discover the behavioural models of the users [3].

In our work, we have first cleaned the data by removing no relevant information and the noise. The remaining data are arranged in a coherent way in order to identify, in a precise way, the users sessions.

We then defined a new approach of extraction which treats the data resulting from the preprocessing phase as being a set of communities. Our aim is to extend the application of the recent community detection methods in the Web Mining context in order to profit from their classifying capacity in the communities discovery.

The rest of the paper is organized as follows. Section 2 describes the Web usage data preprocessing which we intend to increase the quality of the data obtained at the end of the preprocessing step. In section 3, we present an approach that extract interesting correlations from the data based on discovery community method. Section 4 contains our experimental results. General remarks and conclusions are presented in section 5.

2. Preprocessing method

The generic process WUM is adapted to each axis of the Web mining according to the nature of the used data (text, logs, edges…). The functional structure of the process of the web usage mining is structured in six modules principal like representing in figure 1.

2.1 Data transformation module

The entry of the data transformation module is a log file which is a textual file that records the requests made to the Web server in chronological order. The most used formats for log files are CLF (Common Log Format) and the ECLF (Extended CLF). We use the standard ECLF. An example of this format is as follow:

```
41.200.89.109 - - [12/Oct/2008:20:18:23 +0100]
"GET/citic2008/soumission.html HTTP/1.1" 200 23247
"http://www.univ-setif.dz/citic2008/index.html"
"Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.9.0.3)
Gecko/2008092417 Firefox/3.0.3
```

1) the name or IP address of the appealing machine.
2) the name and the login HTTP of the user.
3) the date and the hour of the request.
4) method used by the request (Get, Post, etc.)
5) the URL of the request.
6) the used Protocol.
7) the request statute .
8) size of the sent file.
9) the URL which referred the request.
10) the Agent (navigator and the operating system)

The analysis of Web log files permits to identify useful patterns of the browsing behavior of users which can be exploited in the process of Web personalization.

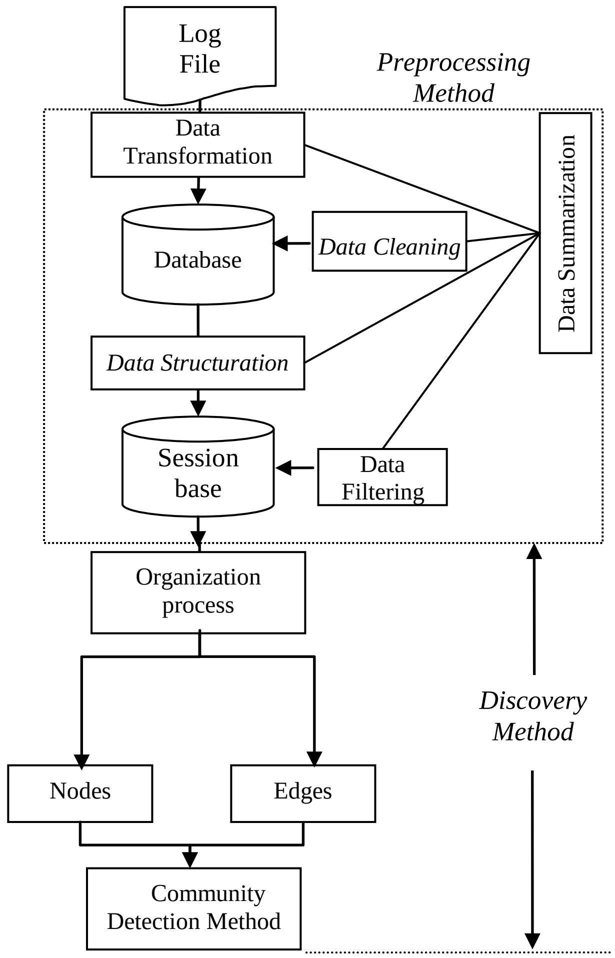

Fig.1 Architecture of the web Usage Mining Process.

2.2 Data cleaning module

The data cleaning module is used to remove the useless records in order to maintain only users' data which can be accurately exploited to identify browsing behavior of users. The choice of the data to be removed depends on the ultimate objective of the personalization system of the Site. In our work, the objective is to develop a WUM system to offer personalized dynamic links to the site's visitors. Ttherefore the system has to keep only records relating to explicit requests that represent users' actions. Consequently, the data cleaning module was developed to eliminate the following requests:

2.2.1 Method different from "GET"

In general, the requests containing a value different from "GET" are not explicit requests of the users, but they often relate to accesses with CGI, of the visits of robots, etc. Consequently, these requests are regarded as non significant and are withdrawn from the access log files.

2.2.2 Failed and corrupted requests

These requests are represented by records containing a HTTP error code. A status with value different from 200 represents a failed request (e.g. a status of 404 indicates that the requested file was not found at the expected location).

2.2.3 Requests for multimedia objects

In the HTTP protocol, an access request is carried out for every file, image, multimedia object embedded in a requested Web page. As a consequence, a single request for a Web page may often produces several entries in the log file that corresponds to files automatically downloaded without an explicit request of the same user. The requests of this type of files can be easily identified since they contain a particular URL name suffix, such as gif, jpeg, jpg, and so on. The conservation or removal of these multimedia objects depends on the kind of the Web site to personalize and theirs natures. In general, these requests do not represent the effective browser activity of the user visiting the site, hence they are removed. In other cases, eliminating requests for multimedia objects may cause a loss of useful information.

2.2.4 Requests originated by Web robots

Log files contain some number of records corresponding to requests originated by Web robots. Web robots are programs that automatically download complete Web sites by following every hyperlink on every page within the site in order to update the index of search engine. These requests are not regarded as usage data and, consequently, have to be removed. To identify web robots' requests, the data cleaning module implements two different heuristic [4].

Firstly, all records containing the name "robots.txt" in the requested resource name (URL) are identified and removed. The second heuristic is based on the fact that web robots retrieve pages in an automatic and exhaustive manner, so they are characterized by a very high browsing speed that is equal to total number of visited pages / total time spent to visit those pages. Therefore, for each different IP address we calculate the browsing speed and all requests having this value exceeding a threshold (pages/second) are regarded as made by robots and are consequently removed. The threshold value is determined after reviewing the log files. After data cleaning, only requests for relevant resources are saved in the database. At the end of this step, we formally define $R = \{r_1, r_2, ..., r_{n_r}\}$ as the set of all distinct resources requested from the Web site under analysis.

2.3 Data structuration module

The data structuration module regroups requests of the log file in user sessions. A session is defined as a limited set of resources accessible by the same user within a particular visit. The identification of user sessions from the log data is a difficult task because many users can use the same computer and the same user can use different computers. Therefore, one main problem is how to identify the user. For websites that require user registration, the log file contains the user login that can be used for user identification. When the user login is not available, the user is identified from the IP address, i.e. we consider each IP address as a different user (being aware that an IP address might be used by several users) [5].

We define $U = \{u_1, u_2, ..., u_{n_u}\}$ as the set of all the users that have accessed that website. We use a time-based method to identify sessions [2] [8]. A user session represents the set of all access originating from the same user within a predetermined time. This period is determined by considering a maximum elapsed time $\Delta t_{\max}$ between two consecutive accesses. Moreover, to better handle special situations for example, when users access several times to the same page due to the slow connections or intense network traffic, a minimum elapsed time $\Delta t_{\min}$ between consecutive accesses is also fixed [4]. We define a user session as:

$s^{(i)} = (u^{(i)}, t^{(i)}, r^{(i)})$ Where:

$u^{(i)} \in U$: is the user identification.

$t^{(i)}$: is the access time of the whole session.

$r^{(i)}$: is the set of all resources requested during the i-th session (with corresponding access time), namely:

$$r^{(i)} = ((t_1^i, r_1^i), (t_2^i, r_2^i), ..., (t_{n_i}^i, r_{n_i}^i)) \qquad (1)$$

with $r_j^i \in R$

Where access time t_k^i to a single resource satisfies the following:

$t_{k+1}^i \geq t_k^i$ and $\Delta t_{\min} < t_{k+1}^i - t_k^i < \Delta t_{\max}$

Summarizing, after the data structuration phase, a set of n sessions $s^{(i)}$ is identified from the log data. We denote the set of all identified sessions by: $S = (s^{(1)}, s^{(2)}, ..., s^{(n_s)})$.

Once all sessions have been identified, the data structuration module presents a panel that lists the extracted sessions and allows us to view and save the details (IP address, requested resources in the session, date and time of the requests) of each user session.

2.4 Data filtering module

After the identification of user sessions, we perform a data filtering step to remove the less requested resources and retain only the most requested ones. For each resource r_i , we consider the number of sessions NS_i that required the resource r_i , and we compute the quantity $NS = \max_{i..n_R} NS_i$. Then, we define a threshold ε, and we remove all request with $NS_i < \varepsilon$ are removed. In this way, the data filtering module can significantly reduce the number of relevant requested resources, which facilitates treatment of the next phases of the web usage mining.

2.5 Data Summarization Module

The Data Summarization Module generates reports summarizing the information obtained after the application of pre-processing step. This statistical information permit to obtain a schematic and concise description of the usage data mined from the analyzed log file. It provides the necessary information to detect some particular aspects related to the user browsing behavior or to the traffic of the considered site log file.

3. Discovery method

Once the raw logs have been preprocessed, data mining techniques can be applied on the dataset to discover new patterns. Such techniques include, but are not limited to: association rules mining, sequential pattern mining and clustering. In our work, we have suggested the use of the recent method of community detection in order to identify groups of users with similar behavior for which personalized versions of the Web site may be created.

In the second phase of WUM process and in order to find a pattern discovery, we have applied an organization process which consists in analyzing the pretreated data of the session base and to model them via a functional graph, such as the resources will be represented by nodes and the browsing sequences of users during each session will be represented by edges. After obtaining this graph, we proceed to the identification of the users clusters which have similar behaviors in term of visited content, our choice is based on Newman algorithm [6] and the modularity function [9] to identify the community structure and thus to define the suitable pattern discovery.

3.1 Concepts of community structure

In complex networks, the communities are groups of nodes which share probably a common proprieties and/or similar functions. The communities may be correspond , for example, to groups of Web pages accessible over the Internet that have the same subject [10], functional modules as cycles and pathways in metabolic networks[11], a set of people or groups of people with some pattern of contacts or interactions between them [12, 13], and subdivisions in the food webs [14,15]. In this paper, the communities correspond to groups of web pages which show the same browsing behavior of users. Newman and Girvan [16] introduce a measure of quality of a particular partition which they called “modularity” to detected if communities are good or no and to value such partitions. The modularity is based on assortative mixing measure [17].

Modularity measures when the division is a good one, in the sense that there are many edges within communities and only a few between them.

3.2 Organization process

Once user sessions have been identified, we have to use them to extract the Web graph that represents the analytic network. We have applied an algorithm that can be used to obtain the degree and the edges for a Web resource. In WUM process, we need to treat the case of weighted graph because resources may be visited several times in the same session and also through the different sessions. In fact, as can be seen in figure 2, we have computed the degree of resource as strictly related to the frequency of accesses to that resource and we have determined the whole weighted social network.

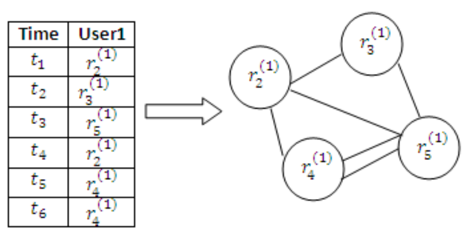

Fig. 2 *Organization process*

In this example the result is a weighted network containing far more information than a simple binary adjacency matrix. Thus the adjacency matrix represents the weight of connection from r_v to r_w .

We could obtain insight into the behavior of weighted graphs very simply by mapping them onto unweighted multigraphs and any techniques that can normally be applied to unweighted graphs can be applied to the multigraph as well [19].

3.3 Pattern discovery task

Naturally, in addition to dividing the graph top down into clusters, one may also work bottom up merging singleton sets of nodes iteratively into clusters. Such methods are called agglomerative clustering algorithms. In our method, we have used the fast algorithm [6] and we have applied the idea presented in [19] in order to focus on weighted social network.

Let $G=(V,E)$ be a weighted graph describing a pretreated database of session with V the set of nodes and E the set of edges. We define the Modularity function as it is define in [18], thus the is

$$Q=\frac{1}{2m}\sum_{r_v r_w}\left[A_{r_v r_w}-\frac{k_{r_v}k_{r_w}}{2m}\right]\delta(c_{r_v},c_{r_w}) \qquad (2)$$

Where m denotes the total number of edges of the graph, so $m=\frac{1}{2}\sum_{r_v r_w} A_{r_v r_w}$

k_{r_v} is the degree of node r_v, we write $k_{r_v}=\sum_{r_v} A_{r_v r_w}$

The element $A_{r_v r_w}$ of the adjacency matrix of the network represents the weight of connection from r_v to r_w. The nodes are dived into communities such that node r_v belongs to community c_{r_v} and the $\delta-$function yields one if nodes r_v and r_w are in the same community, zero otherwise.

The steps of the algorithm are as follows.

1. We construct the weighted adjacency matrix $A_{r_v r_w}$ of the analyzed graph G.
2. Initially, we consider that each community is composed of a single node.
3. Joining the pair of communities whose amalgamation produces the largest increase in $\Delta Q_{r_v r_w}$ but do not join the pair of communities whose there are no edges between them. The $\Delta Q_{r_v r_w}$ value is written as follows

$$\Delta Q_{r_v r_w}=\begin{cases}\frac{1}{2m}-\frac{k_{r_v}k_{r_w}}{(2m)^2} & \text{if } r_v, r_w \text{ are connected},\\ 0 & \text{otherwise.}\end{cases} \qquad (3)$$

4. We update the elements of the matrix $\Delta Q_{r_v r_w}$ and the modularity matrix $Q_{r_v r_w}$ in the manner that they correspond to the joined communities.
5. Repeat step 3 until only one community remains.

For a network of n vertices, after $(n-1)$ such joins we are left with a single community and the algorithm stops. The partition corresponding to the maximum value of modularity on this graph should be the best or at least a very good one.

4. Analysis and experiments

4.1 Preprocessing results

Our preprocessing method has been tested on log files stored by the Web site Server of Ferhat Abbas University of Setif (Algeria) available at the URL www.univ-setif.dz. The treated file covers the site activities during the period from 17 January 2010 to 14 February 2010. Table I presents the results of the preliminary analysis of log files and synthesizes the results provided during the data summarization phase.

TABLE I
THE DATA TRANSFORMATION SUMMARY.

File size	100 448 034 bytes
Date/ beginning hour	17/01/2010 04:03:30
Date/ending hour	14/02/2010 09:09:00
Number of requests	365 863

In the following, we analyze the log file, during the data cleaning phase, in order to determine the non explicit user requests.

The number of requests corresponding to multimedia objects is very important. They include 72,48 % of the requests. After data cleaning phase, only 27,52% of the requests are maintained in the database.

TABLE III
THE DATA CLEANING SUMMARY.

Multimedia		Method		Status	
Autre	100 691	Get	363 749	200	257 276
.gif	53 177	!get	24	206	20 038
.png	88 413	head	952	301	512
.jpg	69 449	options	290	304	75 776
.ico	12 005	post	705	400	22
.bmp	176	propfind	93	403	226
.css	17 866	put	50	404	11 927
.js	24 077			405	62
				501	24
Total	**265 172**	**Total**	**2 114**	**Total**	**108 587**
72,48%		**0,58%**		**29,68%**	

In case of Get method, we remove the requests that have access methods different from it. Table 2 presents the number of each type of method in the log file. We note that the number of removed requests is very small compared to the total number (0,58%).

The requests which have a status different from 200 are regarded as failed request. We list three major categories of irrelevant requests: 3% of the requests with a status of 404 indicate that the requested file was not found at the expected location, 5% of the requests with a status of 206 and 20% of the requests with a status code of 304 indicate that the requested file have a browser refresh problem. At the end of the data cleaning phase, we retain 70% of the requests.

Table II recapitulates the different removed requests of the database. We observe that overlapping can occur between two removed categories. For example, a request with method “Head” can also be a request for a multimedia object. In this case, the data summarization module counts twice the removal of the request, even though only one record is deleted from the log file. Table III illustrates this overlapping.

TABLE IIIII
OVERLAPPING BETWEEN THE REMOVED REQUESTS CATEGORIES.

	Request category			**Nb**	**%**
	Multi media	**Method ≠ Get**	**Status ≠ 200**		
Cleaned	X			183 583	50,18
		X		1 579	0,43
			X	26 548	7,26
	X	X		25	0,01
	X		X	81 529	22,28
		X	X	475	0,13
	X	X	X	35	0,01
Valid				72 089	19,70
Total	265 172	108 587	2 114	365 863	100 %
%	72,48%	29,68%	0,58%	100 %	

After the data cleaning phase, the log files were processed by the data structuration module in order to define the two sets R and U :

R The whole of the requested resources from the analysed web site.

U The web site users.

Then we apply the structuration algorithm [7] to determine the sessions taking account of the two values: $\Delta t_{\max}$ and $\Delta t_{\min}$, such as the minimal value is used to detect the robots and the web crawler, and the maximal value allow detection of new sessions (Table IV).

TABLE IVV
THE STRUCTURATION DATA SUMMARY

Input data		**Setting**	
number of requests	103975	$\Delta t_{\max}$	30 minutes
Nb IP (U **)**	8 676	$\Delta t_{\min}$	05 seconds
Nb URL (R **)**	7 707	Nb Sessions:	17 379

The last phase of the preprocessing method is resources filtering, we have removed the least requested URLs according to the ε threshold and we have obtained the results illustrated in Table V.

TABLE V
THE FILTERING DATA SUMMARY.

	Before	**ε = 5**	**ε = 50**	**ε = 100**
Nb Req.	103 975	66 792	48 571	29 481
Nb URL	7 707	1 607	193	105
Nb IP	8 676	2 843	2 829	1 480
Nb Sess.	17 379	4 665	4 571	2 199

4.2 Community detection results

In the last phase of our work, we have applied an organization process to extract the functionally graph from the session base. So, we have obtained the network structure that identifies the users' session and all the sessions (the nodes represent resources and edges represent the browsing sequences of users during each session). Any community detecting algorithm requires establishing its analytical network. Figure 3 shows this network structure.

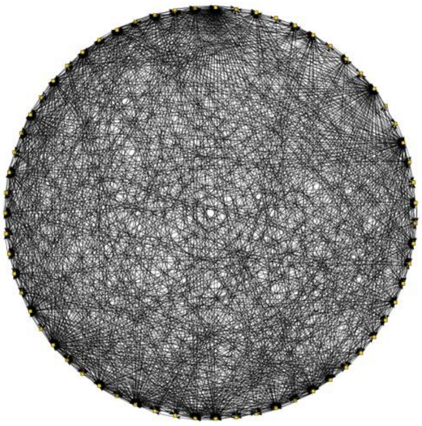

Fig. 3 Network structure.

In pattern discovery step, we intend to identify community structure and detect the browsing behavior of users which can be exploited in the process of Web personalization. A community structure is a set of nodes which have more internal density within the community than with the rest of the network [17]. The proposed discovery method belongs to hierarchical partitioning approach to clustering. It produces a nested sequence of partitions of the set of data points, the used web graph contain 66 nodes and 1180 edges (table 6), which can be displayed as a tree with a single cluster, including all points at the root and singleton clusters (individual points) at the leaves.

TABLE VI
DATA INPUT OF THE USED WEB GRAPH.

Number of nodes	Number of edges
66	1180

The detection community algorithm computes $\Delta Q_{r_v r_w}$ and find the pair of communities r_v, r_w with the largest $\Delta Q_{r_v r_w}$. The output of the algorithm can be represented in the form of a dendrogram (Fig. 4) and the optimal section of the dendrogram found by looking for the optimal value of Q (Fig. 5).

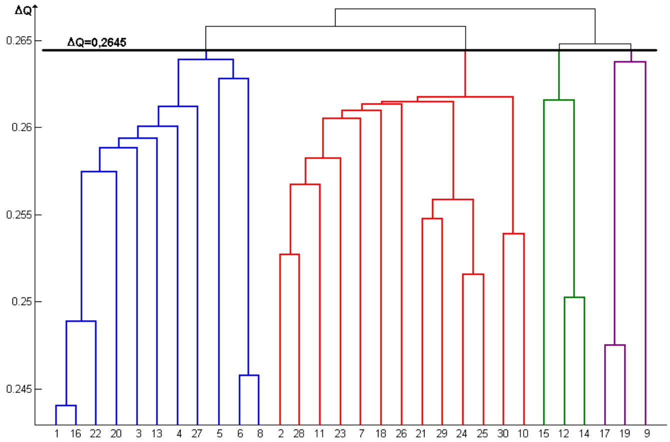

Fig. 4 The dendrogram represent the partitioning of network

As is known to all, modularity Q with the maximum value corresponds to the best partition of community detecting, here the best value that we have obtained is 0.2645.

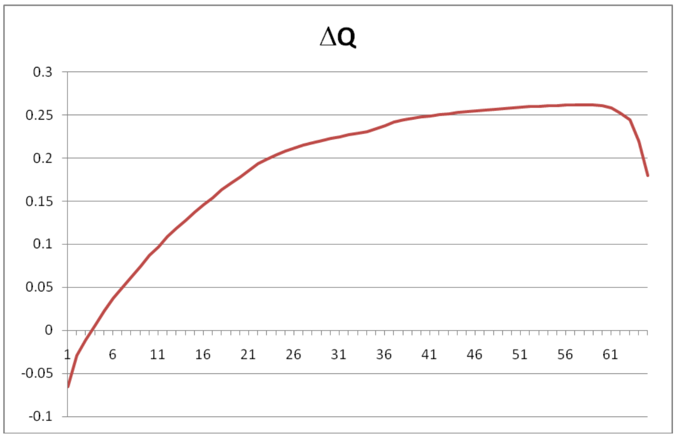

Figure 4. Value of modularity.

By applying the pattern discovery method, we have obtained 4clusters. The cluster number 3 contains the visits to the Web pages of scientific event (e.g. call of paper, program and important dates). In this case, the goal of these users is precise: the visitor interest is to consult the scientific activities of Ferhat Abbas University, the cluster number 4 regroups the visits interested in Web pages of research and valorization, the first cluster detects the visits between the deferent galleries of images and the second cluster shows the visits to deferent faculties. These analyses have permitted to identify homogenous classes of visitors. The graph obtained is presented in Fig. 6.

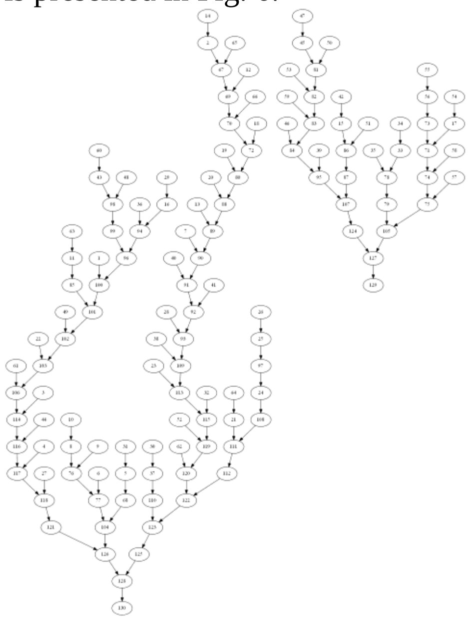

Figure 6. The 4 classes of visitors

5. Conclusions

The Web Usage Mining can be improved by using, in the two steps of the WUM process, enriched information about the structure and content of the Web sites analyzed. The preprocessing method that we have used allows a significant reduction in the number of initial requests and offers a structured session base for the next step of discovery method. The implemented discovery algorithm takes into account more suitable information given by the weighted graph.

References

[1] D. Tanasa. Web usage mining : Contributions to intersites logs preprocessing and sequential pattern extraction with low support. Ph. D. Thesis, University of Nice Sophia Antipolis, 2005.

[2] R. Cooley. Web usage mining : Discovery and application of interesting patterns from web data. Phd thesis, University of Minnesota, 2000.

[3] Pierrakos D. et al. Web usage mining as a tool for personalization: a survey. User Modeling and User-Adapted Interaction, 13(4), p. 311-372 (2003).

[4] Tan,P. N. and Kumar, V.. Discovery of Web Robot Sessions Based on their Navigational Patterns, Data Mining and Knowledge Discovery, 6(1), p. 9 -35 (2002).

[5] Suryavanshi B.S. et al. A Fuzzy Hybrid Collaborative Filtering Technique for Web Personalization. In Proc. of 3rd Workshop on Intelligent Techniques for Web Personalisation (ITWP'05).

[6] Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Physical Review E, Vol 70, 066111

[7] Nasraoui O. World Wide Web Personalization. In J. Wang (ed), Encyclopedia of Data Mining and Data Warehousing, Idea Group (2005).

[8] Paliouras G. et al. Large-scale mining of usage data on Web sites. AAAI Spring Symposium on Adaptive User Interface, Stanford, California, p.92-97 (2000).

[9] Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Physical Review E, Vol 70, 066111

[10] Flake GW, Lawrence S, Lee Giles C, Coetzee FM, Self-Organization and Identification of Web Communities. IEEE Computer, Vol 35, No 3, pp 66–71,2002.

[11] Guimer R, Amaral LAN, Functional cartography of complex metabolic networks. Nature 433, pp 895-900, 2005.

[12] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. PNAS, 99(12) :7821-7826, 2002.

[13] Lusseau D, Newman MEJ (2004), Identifying the role that animals play in their social networks. Proceedings of the Royal Society of London B, Vol 271, pp S477–S481.

[14] Pimm SL (1979) The structure of food webs. Theoretical Population Biolology,Vol 16, pp 144–158.

[15] Krause AE, Frank KA, Mason DM, Ulanowicz RE, Taylor WW (2003) , Compartments exposed in food-web structure. Nature, Vol 426, p 282–285.

[16] M. E. J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69, 026113,2004.

[17] M. E. J. Newman, Mixing patterns in networks, Phys. Rev. E 67, 026126 -2003.

[18] M. E. J. Newman, Fast algorithm for detecting community structure in networks, *Phys. Rev. E* 69, 066133-2004.

[19] M. E. J. Newman, Analysis of weighted networks, *Phys. Rev. E* 70, 056131 (2004).