

Two Algorithms for Web Applications Assessment

<http://dx.doi.org/ijet.v6i3.1660>

Stavros Valsamidis, Sotirios Kontogiannis and Alexandros Karakos
Democritus University of Thrace, Xanthi, Greece

Abstract—The usage of web applications can be measured with the use of metrics. In a LMS, a typical web application, there are no appropriate metrics which would facilitate their qualitative and quantitative measurement. The purpose of this paper is to propose the use of existing techniques with a different way, in order to analyze the log file of a typical LMS and deduce useful conclusions. Three metrics for course usage measurement are used. It also describes two algorithms for course classification and suggestion actions. The metrics and the algorithms were in Open eClass LMS tracking data of an academic institution. The results from 39 courses presented interest insights. Although the case study concerns a LMS it can also be applied to other web applications such as e-government, e-commerce, e-banking, blogs e.t.c.

Index Terms—eLearning, metrics, algorithms, assessment.

I. INTRODUCTION

Learning Management Systems (LMSs) are extensively used nowadays and they provide a variety of information and communication channels for the users [21]. Among the features they provide are the development, management, distribution, diffusion and presentation of the educational material as well as tools for the management of users and courses. One of the main problems of the LMSs is the lack of exploitation of the acquired information due to its volume. Most of the times, these systems produce certain reports with statistical data, which however, do not help instructors to draw useful conclusions either regarding the course or the students and are useful only for the administrative purposes of each LMS

Some of the most well known commercial LMS platforms used for educational purposes worldwide are Blackboard, WebCT and TopClass, while Claroline, Moodle, Ilias and aTutor are freely distributed under appropriate licenses [16]. In Greece, the Greek University Network (GUNet) uses the platform Open eClass [7], which is an evolution of [4]. This system is an asynchronous distant education platform which uses Apache as a web server, MySQL as database server and has been implemented in PHP.

Server log files store information containing the page requests of each individual user [22]. Data mining techniques have been used to discover the sequence patterns of students' web usage after the analysis of log files data [18]. The extraction of sequential patterns has been proven to be particularly useful and has been applied to many different educational tasks [17].

The objectives of this paper are the analysis of the log file of a typical LMS and deduce useful conclusions. Metrics which were firstly introduced by the authors, are also used [23, 24]. Finally, two algorithms for classification of the courses and suggestions to the users are presented.

The paper is organized as follows. Section II describes the background theory. Section III describes the logging data and pre-processing procedure. Section IV describes the data processing procedure with the introduced metrics. Section V describes two algorithms. Section VI presents the conclusions along with future directions.

II. BACKGROUND

There are several studies that show the impact of data mining on eLearning. While data mining methods have been systematically used in a lot of e-commercial applications, their utilization is still lower in the LMSs [26]. It is important to notice that traditional educational data sets are normally small [8], if we compare them to files used in other data mining fields such as e-commerce applications that involve thousands of clients [19]. This is due to the typical, relatively small size of the classroom, although it varies depending on the type of the course (elementary, primary, adult, higher, tertiary, academic or/and special education); corresponding transactions are therefore also fewer. The user model is also different in both systems [18].

Very interesting is the iterative methodology to develop and carry out maintenance of web-based courses, to/in which a specific data mining step was added [6]. The proposed system finds, shares and suggests the most appropriate modifications to improve the effectiveness of the course. The obtained useful information is used directly by the educator of the course in order to improve instructional/learning performance. This system recommends the necessary improvements to increase the interest and the motivation of the students. It is well known that motivation is essential for learning: lack of motivation is correlated to learning rate decrease [3]. There are several specialized web usage mining tools that are used in the e-learning platforms. CourseVis [12] is a visualization tool that tracks web log data from an LMS. By transforming this data, it generates graphical representations that keep instructors well-informed about what precisely is happening in distance learning classes. GISMO [13] is a tool similar to CourseVis, but provides different information to instructors, such as a student's details in using the course material. Sinergo/ColAT [2] is a tool that acts as an interpreter of the students' activity in a LMS. Mostow et al. [15] describes a tool which uses log files in order to represent the instructor-student interaction in hierarchical struc-

ture. MATEP [27] is another tool acting at two levels. Firstly, it provides a mixture of data from different sources suitably processed and integrated. These data originate from e-learning platform log files, virtual courses, academic and demographic data. Secondly, it feeds them to a data webhouse which provides static and dynamic reports. Analog is another system [25] which consists of two main components. The first performs online and the second offline data processing according to web server activity. Past users activity is recorded in server log files which are processed to form clusters of user sessions.

In addition, Khribi et al. [10] proposes an automatic personalization approach. This approach provides online automatic recommendations for active learners without requiring their explicit feedback. Similar to Analog system, it consists of two main components: an off-line and an on-line. The off-line component preprocesses the appropriate data in order to model both learner and content. The online component uses the produced models on-the-fly to recognize the students' needs and goals, and provides learners with recommendation lists.

Strazds and Kapenieks [20] propose a new approach for automatic user satisfaction measurement by identifying and indexing the target groups of various e-learning material such as e-courses, educational games etc through the EDUSA test. Some other researchers such as [1, 14] propose metrics for e-learning evaluation. In e-commerce web usage analysis some metrics are proposed by Lee et al. [11].

A methodology for the maintenance of web-based courses was also proposed by [9] which incorporates a specific data mining step. Publications of the authors relevant to this paper are the automated suggestions and course ranking through a web mining system [23] and the proposal of two metrics, homogeneity and enrichment, for web applications assessment, which are also used in this paper [24].

III. LOGGING THE DATA AND PRE-PROCESSING

A. Logging the data

This stage involves the logging of specific data from LMS. Apache web server uses the following configurations for the production of its log files: Common Log Format (CLF), Extended Log Format (ELF), Cookie Log Format (CKLF) and Forensic Log Format (FLF).

A module that uses FLF format and records attributes before and after web server request processing was implemented. FLF format is used, instead of CLF, because of the advantage that it stores server requests, before and

after request processing at the web server. Such recording contains accurate information regarding user thinking time that indicates content difficulty and complexity and requests processing time that indicates content delivery effort.

In more detail, the data recording module, is embedded in the web server of the e-learning platform and records specific e-learning platform fields. Specifically, thirteen (13) fields (request_time_event, remote_host, request_uri, remote_logname, remote_user, request_method, request_time, request_protocol, status, bytes_sent, referer, agent and user requests) from different courses and user requests are recorded with the use of an Apache module, developed in Perl programming language, as a first step.

The development of such a module has the following two advantages: rapid storage of user information, since it is executed straight from the server API and not by the LMS application, and the produced data are independent of specific formulation used by the LMS platform.

B. Data pre-processing

The data of the log file contain noise such as missing values, outliers etc. These values have to be pre-processed in order to prepare them for data mining analysis. Specifically, this logging data step filters the recorded data. It uses outlier detection and removes extremes. This step is not performed by the LMS platform and thus can be embedded into a variety of LMSs. Also, it facilitates data mining analysis methods construction of robust results.

The produced log file, is filtered, so it includes only the following three fields: (i) courseID, which is the identification string of each course; (ii) sessionID, which is the identification string of each session; (iii) page Uniform Resource Locator (URL), which contains the requests of each page of the platform that the user visited.

C. Study population and context

In detail the dataset was collected from a real LMS environment used in the Technological Education Institute (TEI) of Kavala that uses the Open eClass e-learning platform [7]. The data are from the spring semester of 2009 from the Department of Information Management and involve 1199 students and 39 different courses. The data are in ASCII form and are obtained from the Apache server log file. A view of the collected data in forensic log format is shown in table I.

The log file which is produced, from the previous step, is filtered and pre-processed in order to include the following fields: courseID, sessionID and page Uniform Resource Locator (URL).

TABLE I.
ECLASS DATA IN FLF

remote_host	request_uri	re- mote _log- name	re- mote _user	re- quest _method	request_time	re- quest_pro tocol	status	bytes_sent	refer er	agent
66.249.72.212	/component/search/smb.conf.html	-	-	GET	[03/Mar/2009:18:57:00+0200]	HTTP/1.1	200	9805	-	Mozilla/5.0(compatible; Googlebot/2.1; +http://ww...
66.249.72.212	/mailing-list/56.html	-	-	GET	[03/Mar/2009:18:57:31+0200]	HTTP/1.1	200	12730	-	Mozilla/5.0(compatible; Googlebot/2.1; +http://ww...
66.249.72.212	/newsfeeds.html	-	-	GET	[03/Mar/2009:18:58:01+0200]	HTTP/1.1	200	7967	-	Mozilla/5.0(compatible; Googlebot/2.1; +http://ww...

IV. PROCESSING THE DATA

The aforementioned fields in the previous section are not adequate in order to evaluate the course usage. So, some metrics are used for the facilitation of the course usage evaluation (Table II). First, the indexes Sessions, Pages, Unique pages, Unique Pages per CourseID per Session are computed with the use of a Perl program. Then, the metrics Enrichment, Disappointment, Interest and Homogeneity are calculated.

TABLE II.
METRICS NAME AND DESCRIPTION

Index/Metric name	Description of the index/metric
Sessions	The total number of sessions per course viewed by users
Pages	The total number of pages per course viewed by users
Unique pages	The total number of unique pages per course viewed by users
Unique Pages per CourseID per Session (UPCS)	The total number of unique pages per course per session viewed by users
Enrichment	The enrichment of courses
Disappointment	The disappointment of users
Interest	It is the one's complement to the disappointment
Homogeneity	Homogeneity of courses

The number of sessions and the number of pages viewed by all users are counted for the calculation of course activity. The metric unique pages measures the total number of unique pages per course viewed by all users. The Unique Pages per Course per Session (UPCS) metric expresses the unique user visits per course and per session; it is used for the calculation of the course activity in an objective manner. Because some novice users may navigate in a course and visit some pages of the course more than once, UPCS eliminates duplicate page visits, since it considers the visits of the same user in a session only once.

Enrichment is a metric which is proposed in order to express the "enrichment" of each course in terms of educational material. Enrichment is defined as the complement of the ratio of the unique pages over total number of course web pages as proposed in [23, 24].

$$\text{Enrichment} = 1 - (\text{Unique Pages} / \text{Total Pages}) \quad (1)$$

where $\text{Unique Pages} \leq \text{Total Pages}$.

Enrichment values are in the range [0, 1). When users follow unique paths in a course this is 0 while in a course with minimal unique pages this is close to 1. Since it offers a measure of how many unique pages were viewed by the users, it shows how much information included in each course is handed over to the end user inferring that the course contains rich educational material.

Disappointment is a metric which combines sessions and pages viewed by users and it measures the disappointment of the users in the course, in the sense that when a user views few pages of the course, s/he logs out of the course. Disappointment metric is defined as the rate of sessions per LMS course to total number of course web pages.

$$\text{Disappointment} = \text{Sessions} / \text{Total Pages} \quad (2)$$

In other words, the disappointment metric reflects how quickly the users discontinue viewing pages of the courses. Disappointment values are in the range (0, 1]. Due to the negative nature of the Disappointment metric, it was replaced by another metric which has positive sounding manner, Interest. Interest metric is defined as the one's complement to the disappointment.

$$\text{Interest} = 1 - \text{Disappointment} \quad (3)$$

Both disappointment and interest metrics were proposed in [23]. A low interest in a course means that there were not many unique pages viewed per session; therefore the course is not so popular among the students. This may be so either because students were not pleased with the educational material or there are not many pages to visit. High interest indicates that users are interested in course content and continue further with their study. When the quality of the educational material does not fulfil user requirements, the user is led to log out of the course.

Homogeneity metric is another metric which is defined as the ratio of unique visited course pages to the number of sessions that visited the course.

$$\text{Homogeneity} = \text{Unique pages} / \text{Total Sessions} \quad (4)$$

where $\text{Total Sessions per course} \gg \text{Unique course pages}$.

Homogeneity metric value ranges from [0,1), where 0 means that no user followed a unique path and 1 that every user follows unique paths. It is a course quality index and characterizes the percentage of course information discovered by each user participating in a course. The aforementioned metrics contribute to the evaluation of courses usage. The results for the 39 courses are presented in Table III.

TABLE III.
LMS DATA AND GRADE FOR 39 COURSES

Cour se ID	Ses- sions	Pa ge s	Uniq ue page s	U P C S	Inter est	Enrich ment	Ho- mo- gene- ity
IMD	18	73	9	42	0,753	0,877	0,500
IMD	23	94	11	52	0,755	0,883	0,478
IMD	26	10	12	42	0,752	0,886	0,462
IMD	17	61	8	28	0,721	0,869	0,471
IMD	33	14	9	79	0,768	0,937	0,273
IMD	11	25	8	24	0,560	0,680	0,727
IMD	14	38	7	34	0,632	0,816	0,500
IMD	32	11	9	61	0,717	0,920	0,281
IMD	53	20	11	89	0,743	0,947	0,208
IMD	11	24	7	20	0,542	0,708	0,636
IMD	25	80	7	54	0,688	0,913	0,280
IMD	12	30	5	22	0,600	0,833	0,417
IMD	87	33	8	17	0,743	0,976	0,092
IMD	45	13	8	82	0,667	0,941	0,178
IMD	91	29	11	21	0,694	0,963	0,121
IMD	38	11	6	56	0,664	0,947	0,158
IMD	72	21	7	13	0,668	0,968	0,097
IMD	45	12	7	59	0,631	0,943	0,156
IMD	32	74	9	64	0,568	0,878	0,281
IMD	56	14	9	10	0,611	0,938	0,161
IMD	22	47	7	39	0,532	0,851	0,318
IMD	75	20	6	13	0,641	0,971	0,080
IMD	22	46	7	38	0,522	0,848	0,318
IMD	33	71	7	45	0,535	0,901	0,212
IMD	30	62	6	46	0,516	0,903	0,200
IMD	22	43	5	40	0,488	0,884	0,227
IMD	51	10	6	80	0,528	0,944	0,118
IMD	38	80	3	46	0,525	0,963	0,079
IMD	14	23	5	21	0,391	0,783	0,357

IMD	50	90	8	71	0,444	0,911	0,160
IMD	98	18	8	12	0,470	0,957	0,082
IMD	48	82	10	75	0,415	0,878	0,208
IMD	93	16	6	13	0,433	0,963	0,065
IMD	125	4	4	4			
IMD	96	20	31	5	0,355	0,839	0,250
IMD	114	28	42	4	0,333	0,905	0,143
IMD	132	152	23	7	0,339	0,970	0,046
IMD	67	18	23	4	0,217	0,826	0,222
IMD	23	30	38	4	0,211	0,895	0,133
IMD	134	25	27	4	0,074	0,852	0,160

V. ALGORITHMS

In this section, two algorithms which classify the LMS courses and suggest actions to the educators for course improvement are used.

A. Classifier algorithm

The first algorithm classifies LMS courses based on poor or rich quantity of course information material. Afterwards, based on LMS courses with adequate information material, it tries to spot how often course information is added or updated by educators based on homogeneity classification. Finally, using the UPCS metric it identifies whether updates of course information can increase the student's interest in the specific course. Classifier algorithm schema is depicted in Figure 1.

According to the above, the proposed algorithm is based on Enrichment, Homogeneity and UPCS and consists of the corresponding stages.

In the first stage of the algorithm, the Enrichment metric is involved in order to identify courses with poor or rich educational content (poor equals to small enrichment value while rich equals to high enrichment value). A set of N courses are placed in an N-ordered table based on Enrichment, where $N \leq \text{Total LMS platform courses}$, the courses with the highest Enrichment metric values.

In the second stage, the algorithm classifies the previous set of N courses using the values of Enrichment and Homogeneity. The classification of LMS courses is performed using four clusters as shown in Figure 1. The higher the Homogeneity value the more frequent the course updates or the more dynamic the course content, depending on Enrichment value. The lower the Homogeneity value then the LMS is more static in content or of poor content updates. The classification of the courses depends on the average Enrichment value of the N LMS courses and the average Homogeneity value of the high and low Enrichment clusters accordingly.

The aim of the third stage of the algorithm is to identify whether the content can be characterized as rich or poor, and whether it is static, frequent or dynamic. In order to do this, each cluster of courses is ranked based on the value of the UPCS.

B. Application of classification algorithm

The 39 courses were initially ranked according to the Enrichment metric. The algorithm was tested by picking the best and worst LMS courses from a list of 39 courses which are shown in Table IV. That is, best and worst cases from students' usage point of view.

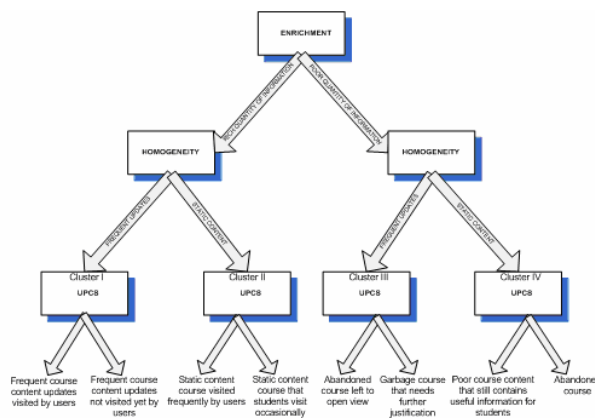


Figure 1. Classifier algorithm schema

TABLE IV. PROCESSED DATA FOR 12 COURSES WITH AVERAGE ENRICHMENT VALUE OF 0.898

Course ID	Sessions	Pages	Unique pages	UPCS	Homogeneity	Enrichment
IMD132	152	230	5	184	0.033	0.978
IMD35	87	338	9	179	0.103	0.973
IMD125	93	164	6	134	0.065	0.963
IMD129	75	209	8	131	0.107	0.962
IMD105	91	297	12	216	0.132	0.960
IMD41	98	185	8	129	0.082	0.957
IMD36	72	217	10	134	0.139	0.954
IMD17	53	206	21	89	0.396	0.898
IMD66	56	144	16	107	0.286	0.889
IMD8	45	135	18	82	0.400	0.867
IMD122	33	71	21	45	0.636	0.704
IMD112	30	62	20	46	0.667	0.677

Based on the previous order by Enrichment Table 1 of 12 LMS courses, the Classifier algorithm was applied by using an average Enrichment value of 0.898 and average homogeneity value for the high enrichment cluster of 0.09 and for the low enrichment cluster of 0.45. The classification of the algorithm produced four clusters, which are shown in the Table V.

TABLE V. CLUSTERING OF THE 12 COURSES BASED ON THE CLASSIFIER ALGORITHM

Enrichment Class	Homogeneity Clusters	Course ID	Sessions	Pages	Unique pages	UPCS	Homogeneity	Enrichment
High	Dynamic Content or Frequently Updated, Cluster I	IMD105	91	297	12	216	0.132	0.960
		IMD35	87	338	9	179	0.103	0.973
		IMD36	72	217	10	134	0.139	0.954
		IMD129	75	209	8	131	0.107	0.962
	Static Content with frequent updates, Cluster II	IMD132	152	230	5	184	0.033	0.978
		IMD125	93	164	6	134	0.065	0.963
		IMD41	98	185	8	129	0.082	0.957
		Low	Dynamic Content with less updates, Cluster III	IMD112	30	62	20	46
IMD122	33			71	21	45	0.636	0.704
Static Content, Cluster IV	IMD66		56	144	16	107	0.286	0.889
	IMD17		53	206	21	89	0.396	0.898
		IMD8	45	135	18	82	0.400	0.867

As shown in Table V, for each one of the four classes the LMS courses are ordered based on the UPCS metric value. So courses IMD105 and IMD36 are the representatives of high and low UPCS values for cluster I, IMD132 and IMD41 for cluster II, IMD112 and IMD122 for cluster III and IMD66 and IMD8 for cluster IV accordingly.

In Table VI, these courses and the classifier algorithm evaluation feedback for each one of these courses are presented.

TABLE VI.
CLUSTERING OF THE 12 COURSES BASED ON THE CLASSIFIER ALGORITHM

Cluster ID	Course ID	CCA Evaluation
I	IMD105	High Activity LMS with updates followed by users
I	IMD36	High Activity LMS with frequent educator updates that are not followed by users
II	IMD132	High Activity LMS with Static content, frequently updated and followed by users
II	IMD41	High Activity LMS with static content, frequently updated but poorly followed by users
III	IMD112	Garbage course or Forum with updates- Need for further evaluation
III	IMD122	Abandoned course of dynamic content left to open view
IV	IMD66	Course of poor static content that still contains information followed by users (or forced to follow)
IV	IMD8	Abandoned course of poor static content occasionally followed by curious users

C. Suggestion algorithm

The goal of the second algorithm is to allow an automated suggestions system for course improvement. The first step of the proposed algorithm is course ranking in descending order by UPCS. A course placed in the first ranking positions is a popular one, either because of exclusive quality of its educational content or quantity of course material.

The first suggestion rule (Figure 2) of the algorithm compares the Interest metric of each course with the $a * \text{Average}(\text{Interest})$ of all LMS courses, where a is a coefficient parameter. If Interest value is lower than $a * \text{Average}(\text{Interest})$, it means that this course either does not have adequate educational content or its content quality does not meet (satisfy) user requirements. In order to distinguish between these two cases a new condition is applied that checks whether course Unique Pages value is less than the $\text{Average}(\text{Unique Pages})$ value for all LMS courses. If this condition is fulfilled, then course content quality is in need of amelioration, while if not, course content is of fine quality and new content additions do not need to be made due to course interest expressed by users.

```

IF Interest < a * Average(Interest) THEN
IF UniquePages > Average(UniquePages) THEN
  "Improve Quality"
ELSE IF Interest > b THEN
  "Add Content"
  ELSE
  "Add Content and Improve Quality"
  ELSE
IF UniquePages < AVG(UniquePages) THEN
  "Light need for new content"
  ELSE
  "No further Improvements required"
```

Figure 2. First Suggestion rule

Additional course quality improvements are suggested if Interest is less than $b * \text{Average}(\text{Interest})$, where b is a coefficient parameter. Finally, if Interest is more than the $a * \text{Average}(\text{Interest})$, and the number of Unique Pages is less than the $\text{Average}(\text{Unique Pages})$, then there is a slight need for new content addition.

The next suggestion rule (Figure 3), applies the Enrichment metric. A low Enrichment value means that users do not visit course pages due to the lack of course content updates. If Enrichment value of a course is less than $c * \text{Average}(\text{Enrichment})$, where c is a coefficient parameter, then the algorithm suggests that it would be good practice for the author to update course content, so as to motivate users to re-visit his/her course.

```

IF Enrichment < c * Average(Enrichment) THEN
  "Update the course content"
```

Figure 3. Second Suggestion rule

Algorithm's a , b and c coefficient parameters range between 0 and 1. In order to accurately calibrate the coefficient parameters, the algorithm was first applied to a reduced set of LMS courses. Course selection was performed based on best and worst case LMS courses, using UPCS ranking. Then a value was calculated by using best to worst course Interest deviation value. b coefficient value is the average best to worst course Interest value and c was calculated as the median value of the first k LMS courses based on UPCS ranging, where $k=5$:

$$c = \text{median}(\text{Interest}) - k * N * 0.0001 \quad (5)$$

where N is the total number of LMS courses and k the number of selected courses with maximum UPCS ranking values.

D. Application of suggestion algorithm

This experiment had two goals: To determine suitable values for a , b , c parameters and to test the two suggestion rules of the algorithm with respect to their impact on improving the course quality.

The first stage of the algorithm was the ranking of the courses. Table VII, due to space limitation, displays the results for the courses in ranking positions 1-5, 21-39, using the UPCS metric. The ranking of the courses is based on: first UPCS, then Enrichment and then Interest values.

Based on the aforementioned metrics, the courses were classified using the following three steps:

- 1) Course ranking step: primarily, course evaluation was considered using the UPCS value and LMS courses were ranked in descending order.
- 2) First suggestion rule step: The first suggestion rule is used in order to evaluate course content in terms of interest as expressed by course users and provides the appropriate suggestions to the instructors, related to the quantity and the quality of their course educational content.
- 3) Second suggestion rule step: Course content is examined in depth in order to express whether users are satisfied from what they see or course content seems confusing or complex to the end user. Enrichment metric was used to identify courses with poor or rich

educational material confirmed by users and provides suggestions for possible course updates.

In order to perform the final two stages of the algorithm, the coefficient parameter values were firstly calculated. According to the experiment outcome, the values for a, b, c parameters were 0.9, 0.6 and 0.95 respectively.

The second goal of the experiment was to test the suggestion rules by showing them to the course instructors and receive verification feedback on the suggestion accuracy. When instructors applied the proposed suggestions, their courses improved in UPCS ranking position.

TABLE VII.
PROCESSED E-LEARNING DATA FOR 39 COURSES

Position	Course ID	Sessions	Pages	Unique pages	UPCS	Interest	Enrichment	Automated Suggestions		
								Add new content	Improve content quality	Update
1	IMD105	91	297	11	216	0,694	0,963			
2	IMD132	152	230	7	184	0,339	0,970	*	*	
3	IMD35	87	338	8	179	0,743	0,976			
4	IMD36	72	217	7	134	0,668	0,968	-		
5	IMD125	93	164	6	134	0,433	0,963	*		
21	IMD120	38	80	3	46	0,525	0,963	-		
22	IMD112	30	62	6	46	0,516	0,903	-		
23	IMD122	33	71	7	45	0,535	0,901	-		
24	IMD9	26	105	12	42	0,752	0,886			
25	IMD115	10	73	12	42	0,863	0,836			*
26	IMD60	22	43	5	40	0,116	0,884	*		
27	IMD64	22	47	7	39	0,149	0,851	-		
28	IMD50	22	46	7	38	0,152	0,848	-		*
29	IMD114	28	42	4	34	0,095	0,905	*	*	
30	IMD80	14	38	7	34	0,184	0,816	-		*
31	IMD23	30	38	4	32	0,105	0,895	*	*	
32	IMD96	20	31	5	30	0,161	0,839	*	*	*
33	IMD10	17	61	8	28	0,131	0,869			
34	IMD134	25	27	4	27	0,148	0,852	*	*	
35	IMD21	11	25	8	24	0,560	0,680			*
36	IMD130	12	30	5	22	0,600	0,833	-		*
37	IMD67	18	23	4	22	0,217	0,826	*	*	*
38	IMD49	14	23	5	21	0,391	0,783	*		*
39	IMD15	11	24	7	20	0,542	0,708	-		*
Average of all courses				7,25	68,41	0,55	0,892			

Appendix: * Strong suggestion, - Light suggestion

VI. DISCUSSION AND CONCLUSION

The proposed method uses existing techniques in a different way to perform LMS usage analysis. It uses the enrichment, homogeneity and interest metrics. It uses two algorithms for course classification and suggestion actions.

It has the following advantages: (1) It is independent of a specific LMS, since it is based on the Apache log files and not the LMS platform itself. Thus, it can be easily implemented for every LMS. (2) It uses new metrics in order to facilitate the evaluation of each course in the LMS and allows the instructors to make proper adjustments to their course educational material. (3) It uses two algorithms for analyzing LMS data, classifies the courses and suggests the proper actions to the educators.

Feedback about the method was received by the educators. The educators were informed about the indexing results along with abstract directions on how to improve their courses. Most of them increased the quality and the quantity of their educational material. They increased the quality by reorganizing the educational material in a uniform, hierarchical and structured way. They also improved

the quantity by embedding additional educational material. By updating educational material, both quality and quantity were increased. A major outcome through the process of informing the educators about the results is that the ranking of the courses constitutes an important motivation for the educators to try to improve their educational material. Because of their mutual competition, they want their courses to be highly ranked. A few educators complained that their courses organization does not assist them to have high final scores in the ranking list. They argued that for example the metric interest is heavily influenced by the number of web pages used to organize the educational material. Thus, courses that have all their educational material organised in few pages have a low interest score. They were asked again to re-organize the material for each course in the LMS according to the order they are taught, in order to facilitate use by the students.

From a pedagogical point of view this method contributes to the improvement of course content and course usability and the adaptation of the courses in accordance to student capabilities. Improvement of course quality gives students the opportunity of asynchronous study of courses with actualized and optimal educational material.

At present, the calculation of the metrics and the experiments of the algorithms are being generated manually. Therefore, some future work is needed to overcome such limitation. Thus, a plug-in tool is being developed to automate the whole procedure. This tool will run in periodically (each month) and will e-mail the instructors the results and the suggestions for their courses. A similar policy was also applied by Feng [5], where after long term observation the instructors were informed automatically by email about the quality of the content of their LMS courses.

The purpose of another future research project would be the measurement of the course quality with the use of the enrichment, interest and homogeneity metrics. Specifically, a Quality index might be defined as the average of enrichment, interest and homogeneity. The use of different weights for enrichment, interest and homogeneity is another topic for research.

It should be mentioned that even if the scope of the method is on LMS platforms and educational content, it can be easily adopted by other web applications such as e-government, e-commerce, e-banking, blogs etc. Furthermore, enrichment, homogeneity and interest metrics may also be used for example by e-government applications, since enrichment shows how much information is handed over to the end user, homogeneity characterizes the percentage of information independently discovered by each user and interest indicates whether users are pleased with the material of the site and do not log out.

ACKNOWLEDGMENT

The authors would like to thank teaching personnel of the Department of Information Management at TEI Kavalas, for their gentle permission in accessing the log data of Open eclass (LMS), especially professor Chatzis V. We would also like to thank application professor Mr. Mardiris V. for his technical assistance and provisions for the installation of appropriate software used by our tool for logging purposes at the web server of Information Management at TEI of Kavala.

REFERENCES

- [1] Albeanu, G. (2007). E-Learning metrics. Proceedings of the 2nd International Conference on Virtual Learning (ICVL 2007).
- [2] Avouris, N., Komis, V., Fiotakis, G., Margaritis, M., & Voyiatzaki, G. (2005). Logging of fingertip actions is not enough for analysis of learning activities. Proceedings of Workshop Usage Analysis in learning systems (AIED'05), Amsterdam.
- [3] Baker, R., Corbett, A., & Koedinger, K. (2004). Detecting Student Misuse of Intelligent Tutoring Systems. Proceedings of the Seventh International Conference on Intelligent Tutoring Systems, pages 531–540.
- [4] Claroline, <http://www.claroline.net>.
- [5] Feng, Mingyu, & Heffernan, N.T. (2005). Informing Teachers Live about Student Learning: Reporting in the Assistent System. Proceedings
- [6] García, E., Romero, C., Ventura, S., & de Castro, C. (2008). An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. Journal of User Modeling and User-Adapted Interaction, 19 (1-2), 99-132.
- [7] GUNet, <http://eclass.gunet.gr/>.
- [8] Hamalainen, W., & Vinni, M. (2006). Comparison of machine learning methods for intelligent tutoring systems. Proceedings of Int. Conf. in Intelligent Tutoring Systems, pages 525–534.
- [9] Kazanidis, I., Valsamidis, S., Theodosiou, T. & Kontogiannis, S. (2009). Proposed framework for data mining in e-learning: The case of Open e-Class, Proceedings of Applied Computing 09, Rome, Italy, pages 254-258.
- [10] Khribi, M. K., Jemni, M., & Nasraoui, O. (2009). Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval. Educational Technology & Society, 12 (4), 30-42.
- [11] Lee, J., Hoch, R., Podlaseck, M., Schonberg, E. & Gomory, S. (1999). Analysis and Visualization of Metrics for Online Merchandising. Lecture Notes In Computer Science, Revised Papers from the International Workshop on Web Usage Analysis and User Profiling, 1836, 126-141, ISBN:3-540-67818-2.
- [12] Mazza, R., & Dimitrova, V. (2007). CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. I. Journal of Human-Computer Studies, 65 (2), 125–139.
- [13] Mazza R., & Milani, C. (2004). GISMO: a Graphical Interactive Student Monitoring Tool for Course Management Systems. Proceedings of International Conference on Technology Enhanced Learning '04 (T.E.L.'04). Milan, 18-19 November 2004.
- [14] Manford, C. and McSpornan, M. (2003). e-Learning quality: becoming a level five learning organization, in Mann S. and Williamson A. (eds): Proceedings of the 16th NACCQ, Palmerston North New Zealand (pp. 343-348).
- [15] Mostow, J., Beck, J., Cen, H., Cuneo, A., Gouvea, E., & Heiner, C. (2005). An Educational Data Mining Tool to Browse Tutor-Student Interactions: Time Will Tell!. Proceedings of workshop on educational data mining (pp. 15–22).
- [16] Romero, C., Ventura, S., & García, E. (2008a). Data mining in course management systems: Moodle case study and tutorial. Computers & Education, 51(1) 368-384. <http://dx.doi.org/10.1016/j.compedu.2007.05.016>
- [17] Romero, C., Gutierrez, S., Freire, M., & Ventura, S. (2008b). Mining and Visualizing Visited Trails in Web-Based Educational Systems. In Educational Data Mining 2008, Proceedings of the 1st International Conference on Educational Data Mining. Montreal, Quebec, Canada, 182-186.
- [18] Romero, C. & Ventura, S. (2007). Educational Data Mining: a Survey from 1995 to 2005. Elsevier Journal of Expert Systems with Applications, 33 (1), 135-146. <http://dx.doi.org/10.1016/j.eswa.2006.04.005>
- [19] Srinivasa, R. (2005). Data mining in e-commerce: A survey, In Sadhana, 30 (2 & 3), 275–289.
- [20] Strazds, A. & Kapenieks, A. (2007). Automated satisfaction measurement for e-learning target group identification, ISSN 1558-7908, IEEE Multidisciplinary Engineering Education Magazine, Vol. 2, NO. 3, September 2007.
- [21] Ueno, M. (2004). Data Mining and Text Mining Technologies for Collaborative Learning in an ILMS "Samurai", Proceedings of the IEEE International Conference on Advanced Learning Technologies, August 30-September 01, 2004, ICALT'04, pages 1052-1053. <http://dx.doi.org/10.1109/ICALT.2004.1357749>
- [22] Ueno, M. (2002). Learning-Log Database and Data Mining system for e-Learning, Proc. of International Conference on Advanced Learning Technologies, ICALT 2002, pages 436-438.
- [23] Valsamidis, S., Kazanidis, I., Kontogiannis, S. & Karakos, A. (2010a, July). Automated suggestions and course ranking through web mining. Paper presented at the 10th IEEE International Conference on Advanced Learning Technologies ICALT 2010, Sousse, Tunisia.
- [24] Valsamidis, S., Kontogiannis, S., Kazanidis, I. & Karakos, A. (2010b). Homogeneity and Enrichment, Two Metrics for Web Applications Assessment, 14th Panhellenic Conference on Informatics. <http://dx.doi.org/10.1109/PCI.2010.41>
- [25] Yan T. W., Jacobsen, M., & Garcia-Molina, H., & Dayal, U. (1996). From user access patterns to dynamic hypertext linking. Proceedings of the Fifth International World Wide Web Conference on Computer networks and ISDN systems, 1007-1014, Paris, France.
- [26] Zaiane, O.R. (2001). Web Usage mining for a better web-based learning environment. Proceedings of Conference on Advanced Technology for Education.
- [27] Zorrilla, M. E., & Álvarez, E. (2008). MATEP: Monitoring and Analysis Tool for e-Learning Platforms, Proceedings of the Eighth IEEE International Conference on Advanced Learning Technologies, pages 611-613. <http://dx.doi.org/10.1109/ICALT.2008.33>

AUTHORS

S. Valsamidis is a PhD candidate student at the Dept. of Electrical and Computer Eng., Democritus University of Thrace, Xanthi, Greece. He received a five-year Electrical Eng. diploma from Department of Electrical Eng., University of Thessaloniki, Greece and MSc in Computer Science from University of London, UK. He is an Applications Prof. in the Dept. of Information Technology, TEI of Kavala, Greece. His research interests are in the areas of databases, data analysis and web applications assessment. His e-mail is: svalsam@ee.duth.gr.

S. Kontogiannis is a PhD candidate student at the Dept. of Electrical and Computer Eng., Democritus University of Thrace, Xanthi, Greece. He received a five-year Eng. diploma and MSc in Software Eng. from Department of Electrical and Computer Eng., Democritus University of Thrace. His research interests are in the areas of Distributed systems, computer networks, middleware protocol design, network modelling and computer networks performance evaluation. His e-mail is: skontog@ee.duth.gr.

A. Karakos received the Degree of Mathematician from the Department of Mathematics from Aristoteles University of Thessaloniki, Greece and the Maitrise d'Informatique from the university PIERRE ET MARIE CURIE, Paris. He completed his PhD studies at university PIERRE ET MARIE. He is Assistant Professor at the Dept. of Electrical and Computer Eng., Democritus University of Thrace, Greece. His research interests are in the areas of Distributed systems, data analysis and programming languages. His e-mail is: karakos@ee.duth.gr.

Received May 15th, 2010. Published as resubmitted by the authors August 2nd, 2011.