E-Learning Platform Usage Analysis

Stavros Valsamidis and Sotirios Kontogiannis
Democritus University of Thrace, Greece

svalsam@ee.duth.gr; skontog@ee.duth.gr

Ioannis Kazanidis
TEI of Kavala,
Greece

kazanidis@teikav.edu.gr

Alexandros Karakos
Democritus University of Thrace,
Greece

karakos@ee.duth.gr

Abstract

E-learning is technology-based learning, such as computer-based learning, web-based learning, virtual classroom, and digital collaboration. The usage of web applications can be measured with the use of indexes and metrics. However, in e-Learning platforms there are no appropriate indexes and metrics that would facilitate their qualitative and quantitative measurement. The purpose of this paper is to describe the use of data mining techniques, such as clustering, classification, and association, in order to analyze the log file of an eLearning platform and deduce useful conclusions. Two metrics for course usage measurement and one algorithm for course classification are used. A case study based on a previous approach was applied to e-Learning data from a Greek University. The results confirmed the validity of the approach and showed a strong relationship between the course usage and the corresponding students' grades in the exams.

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

It should be mentioned that even though the scope of the method is on e-Learning platforms and educational content, it can be easily adopted to other web applications such as e-government, e-commerce, e-banking, blogs, etc.

Keywords: E-learning, indexes, metrics, data mining, algorithm.

Introduction

The advances in Information and Communication Technologies (ICT) have introduced e-learning, an alternative mode of learning, which positively affects the way teaching and learning take place by enabling both educators and students to use their limited time effectively (Delacey & Leonard, 2002; Radcliffe, 2002; Starr, 1977). E-learning is technology-based learning, such as computer-based learning, web-based

Editor: Rowan Wagner

learning, virtual classroom, and digital collaboration. E-learning describes the ability to electronically transfer, manage, support, and supervise learning and educational materials (Cone & Robinson, 2001; Normark & Cetindamar, 2005). Many authors have discussed the way in which e-learning can be used for the delivery of training, assessment, and support (Fichter, 2002). E-learning has made considerable progress since the 1980s, attributable in large measure to technological developments.

Nowadays, the variety of different kinds of E-learning systems is very large. There are systems which support individual learning, collaborative learning, learning content management, learning activity management, formal learning, informal learning, and workplace learning.

One weakness of the E-learning systems is the lack of exploitation of the acquired information due to its volume. Most of the time, these systems produce reports with statistical data, which do not help instructors to draw useful conclusions either about the course or about the students. Moreover, the existing e-learning platforms do not offer concrete tools for the assessment of user actions and course educational content. To enable educators to improve their course and learners to acquire more knowledge, in our experiment two means of course evaluation are used: metrics and questionnaires.

Server log files store information containing the page requests of each individual user (Ueno, 2002). Data mining techniques have been used to discover the sequential patterns of students' web usage after the analysis of log files data (Romero & Ventura, 2007). The extraction of sequential patterns has been proven to be particularly useful and has been applied to many different educational tasks (Romero, Gutierez, Freire, & Ventura, 2008).

The objectives of this paper are the analysis of the log file of an eLearning system and the deduction of useful conclusions. Indexes, metrics, and one algorithm for classification, which were firstly introduced by the authors, are also used (Valsamidis, Kazanidis, Kontogiannis, & Karakos, 2010; Valsamidis, Kontogiannis, Kazanidis, & Karakos, 2010). Finally, data mining techniques were applied disclosing interesting insights.

The paper initially makes a literature review and follows with the background theory, the proposed methodology, the application of the methodology with the use of a case study relating to the eLearning, the practical implications of the results, and the conclusions along with future directions.

Literature Review

There are several studies that show the impact of data mining on eLearning. Whilst data mining methods have been systematically used in a lot of e-commercial applications, their utilization is still lower in the E-learning systems (Zaiane, 2001). It is important to notice that traditional educational data sets are normally small (Hamalainen & Vinni, 2006) if we compare them to files used in other data mining fields such as e-commerce applications that involve thousands of clients (Srinivasa, 2005). This is due to the typical, relatively small class size although it varies depending on the type of the course (elementary, primary, adult, higher, tertiary, academic, or/and special education); the corresponding transactions are therefore also fewer. The user model is also different in both systems (Romero & Ventura, 2007).

Very interesting is the iterative methodology to develop and carry out maintenance of web-based courses, in which a specific data mining step was added (García, Romero, Ventura, & de Castro, 2008). The proposed system finds shares and suggests the most appropriate modifications to improve the effectiveness of a course. The obtained information is used directly by the educator of a 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 (Baker, Corbett, & Koedinger, 2004). There are several specialized web usage mining tools that are used in the e-learning platforms. CourseVis (Mazza & Dimitrova, 2007) is a visualization tool that tracks web log data from an E-learning system. By transforming this data, it generates graphical representations that keep instructors well-informed about what precisely is happening in distance-learning classes. GISMO (Mazza & Milani, 2004) is a tool similar to CourseVis, but provides different information to instructors, such as a student's details regarding the use of the course material. Sinergo/ColAT (Avouris, Komis, Fiotakis, Margaritis, & Voyiatzaki, 2005) is a tool that acts as an interpreter of the students' activity in an E-learning system. Mostow et al. (2005) describe a tool which uses log files in order to represent the instructor-student interaction in hierarchical structure. MATEP (Zorrilla & Álvarez, 2008) 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, MATEP feeds them to a data web house which provides static and dynamic reports. Analog is another system (Yan, Jacobsen, Garcia-Molina, & Dayal, 1996) which consists of two main components. The first performs online and the second offline data processing according to web server activity. Past user activity is recorded in server log files which are processed to form clusters of user sessions. In addition, Khribi, Jemni, and Nasraoui (2009) propose 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.

A methodology for the maintenance of web-based courses was also proposed by (Kazanidis, Valsamidis, & Theodosiou, 2009) 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 (Valsamidis, Kazanidis, et al., 2010) and the proposal of two new metrics, homogeneity and enrichment, for web applications assessment, which are also used in this paper (Valsamidis, Kontogiannis, et al., 2010).

Background Theory

Since the methodology is based on the analysis of the log files of eLearning systems, in the background theory section the foundations of log files are described, as well as the used indexes and metrics.

Log Files

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) (Pabarskaite & Raudys, 2007). In detail:

1. Common Log Format (CLF): This is the typical format used by several web server applications. It outputs a format string per HTTP request that describes recorded attributes based on format symbols that express notations presented in Table 1:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
```

Table 1: CLF format symbols and notation

Symbol	Notation
%h	Client hostname or IP address
%l, %u	Client user log name (identified by auth service). If no value is present, a "-" is substituted
%t	Date Time of the client system following the format: [dd/MMM/yyyy:hh:mm:ss +-hhmm]
%r	Client HTTP request. The HTTP request. The request field contains three pieces of information. The main piece is the requested resource. The request field also contains the HTTP method (GET/PUT/POST) and the HTTP protocol version (1.0/1.1).
%s	Request status returned by the web server: (200: OK, 3xx : Request redirection error, 4xx: Client request error, 5xx: Request not found or internal server error that occurred from request
%b	The bytes field is a numeric field containing the number of bytes of data transferred as part of the HTTP request, not including the HTTP header.

In addition Combined log format (CombLF) is the same as the Common Log Format (CLF), with the addition of two more fields. The additional fields are:

> `%{Referrer}i` field : The "Referrer" HTTP request header. This gives the site that the client reports having been referred from. (This should be the page that links to or includes the current requested object
>
> `%{User-agent}i field:` The User-Agent HTTP request header. This is the identifying information that the client browser reports about itself.

Combined and Common log formats are considered by the authors as one and shall be referred to with the CLF notation.

2. Extended Log Format (ELF): A log file in the extended format contains a sequence of lines and each line may correspond to either a directive or an entry. Entries consist of a sequence of fields relating to a single HTTP request, similar to CLF fields. Directives record information about the logging process itself. Lines beginning with the # character contain directives. A typical output of ELF is the following:

```
#Software: Microsoft Internet Information Server 4.0
#Version: 1.0
#Date: 1998-11-19 22:48:39
#Fields: date time c-ip cs-username s-ip cs-method cs-uri-stem cs-uri-query sc-status
sc-bytes cs-bytes time-taken cs-version cs(User-Agent) cs(Cookie) cs(Referrer)

1998-11-19 22:48:39 206.175.82.5 - 208.201.133.173 GET /global/images/test.gif - 200
540 324 157 HTTP/1.0 Mozilla/4.0+(compatible;+MSIE+4.01;+Windows+95)
USERID=CustomerA;+IMPID=01234 http://webserver/index.html
```

The difference between CLF and ELF is that ELF is the CLF derivative of IIS (Internet Information Server) logging for Microsoft platforms.

3. CooKie Log Format (CKLF) is part of the Apache web server mod-log-config module and offers additionally to the CLF format the capability of logging cookies content per HTTP request. The storage of cookies values follows the CLF format under the symbol:

> %ic for incoming cookies and %oc for outgoing cookies.

4 .Forensic Log Format (FLF) also follows CLF directives and attributes, logging symbols, and notation. FLF maintains all CLF symbols and, additionally, keeps two records for each client request: (a) At the arrival of the request at the web server and (b) after the request has been processed and a reply has been sent to the client. For the recognition of requests a unique request ID is assigned per request and a pair of +/- symbols is placed before the ID that signifies the processing status of the request (logging information has been recorded prior to or after processing of the request by the web server). FLF offers the capability of request data setting and identification. It also offers an estimation of the processing effort of each request and an estimation of the user thinking time since requests coming from the same client session keep the same ID until session expiration or client IP address or request port change or session (HTTP v1.1 maintains the same client port for client connections in contrast to HTTP v1.0. This means that user requests cannot be assigned to a specific user based on client request port value but from session ID value.). FLF is part of Apache mod_log_forensic module and its forensic capability can also be used independently by CLF, by using the notation %{forensic-id}n at the apache configuration file. A typical ELF logfile output is the following:

```
+yQtJf8CoAB4AAFNXBIEAAAAA|GET/manual/de/images/down.gif_HTTP/1.1|Hos
t:localhost|User-Agent:Mozilla/5.0 (X11; U; Linux i686; en-US;
rv%3a1.6) Gecko/20040216 Firefox/0.8
-yQtJf8CoAB4AAFNXBIEAAAAA
```

The first line logs the forensic ID, the request line and all received headers, separated by pipe characters (|). The plus character at the beginning indicates that this is the first log line of this request. The second line just contains a minus character and the same ID. If a request is still pending or not completed the – sign with the request ID is not recorded at the log file.

Indexes and Metrics

The aforementioned fields in the previous sub-section are not adequate in order to evaluate the course usage. So, some indexes and metrics are used for the facilitation of the course usage evaluation (Table 2). First, the indexes Sessions, Pages, Unique pages, Unique Pages per Coursed per Session are computed with the use of a Perl program (Kazanidis et al., 2009; Valsamidis, Kontogiannis, et al., 2010). Then, the metrics Enrichment, Disappointment, Interest and Homogeneity are calculated.

Table 2: 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 Course ID per Session (UPCS)	The total number of unique pages per course per session viewed by users
Enrichment	The enrichment of courses
Homogeneity	The homogeneity of courses
Quality	The average of enrichment and homogeneity values

The number of *sessions* and the number of *pages* viewed by all users are counted for the calculation of course activity. Each session reflects when a user logs in to the platform and, after some activity, logs out from the platform. If there is no activity, there is a timeout of 30 seconds. The number of pages reflects how many pages were viewed by all users. There are some pages of the course which were viewed by many users but there were also some other pages not so popular. In order to refine the situation, we define another index which is called *unique pages* and measures the total number of unique pages per course viewed by all users. It counts each page of the course only once, independently on how many times they were viewed by the users. The *Unique Pages* *per Course per Session (UPCS)* index 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 Valsamidis, Kontogiannis, et al. (2010):

Enrichment = 1- (Unique Pages/Total Pages) (1)

where Unique Pages<=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.

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

Homogeneity =Unique pages/Total Sessions (2)

where Total Sessions per course >> 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 followed 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 course usage. Quality is the average of the metrics Enrichment and Homogeneity. It is a course quality index and reflects the users behaviour related to the variance of the educational material.

Methodology

The usage analysis methodology consists of 6 (six) steps: Logging the data, Data pre-processing, Indexes computation, Metrics calculation, Classifier algorithm, and Data mining. For better comprehension of the proposed methodology, this methodology is depicted in Figure 1 and described below.

Logging the Data

A module that uses FLF format and records attributes before and after web server request processing was implemented.

This step involves the logging of specific platforms derived from E-learning. In detail, a request and data recording module, installed at the web server or embedded in the E-learning platform, records specific e-learning platform fields. More specifically, our installed module at the web server of the E-learning platform monitors, from different courses, 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*, which are recorded with the use of an Apache module, developed in Perl programming language.

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 E-learning application, and the produced data are independent of specific formulation used by the E-learning platform.

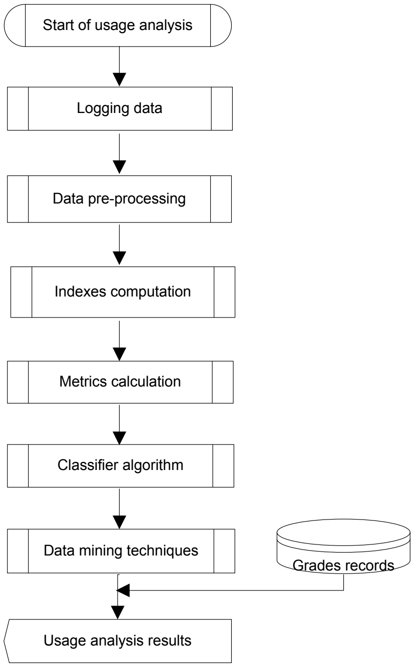


Figure 1: Diagram of the steps for usage analysis.

Data Pre-processing

The data of the log file contain noise such as missing values and outliers. These values have to be pre-processed in order to prepare them for data analysis. Specifically, in this step the recorded data are filtered. Outlier detection is performed and extremes values are removed. This step is not performed by the E-learning platform and thus can be embedded into a variety of E-learning systems. Also, data analysis methods are facilitated for the 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.

Indexes Computation

The aforementioned fields in the previous sub-section are not adequate in order to evaluate the course usage. So, the static accessory indexes Sessions, Pages, Unique pages, UPCS are used for the facilitation of the course usage evaluation (Table 2).

Metrics Calculation

In this step, the metrics Enrichment and Homogeneity are calculated.

The evaluation of courses usage is based on two metrics: the metric Quality and the metric Final. The metric Quality is the mean of Enrichment and Homogeneity, while the metric Final is the product of Quality with UPCS. These two metrics allow us to classify and group the courses depending on their usage in the electronic platform.

The evaluation of the courses is examined initially with UPCS, which is a quantitative index. It is quantitative metric because it measures precisely the number of instances. The courses with a high value of UPCS are popular enough among the students.

Since some courses have the same value of UPCS or values very close to each other, we wanted to clarify the situation and add one absolutely qualitative metric, that is named Quality, which combines appropriately the metrics Enrichment and Homogeneity with equal weights. The total result derives from the product of Quality with UPCS.

Classifier Algorithm

Online courses incorporate specific characteristics according to their educational material. Authors should be able to assess and improve their courses so as to provide a better learning experience for learners. In this section an algorithm which classifies online courses according to specific metrics is presented, The Classifier algorithm may be used with online tools in order to automatically provide feedback to authors about their courses along with useful suggestions for improvement. Some crucial factors of an online course are the quantity of each course educational material, content update, and, of course, acceptance and usefulness. These are the factors used with Clacifier Algorithm in order to classify online courses into specific groups with identical characteristics. In order to succeed in this it makes use of specific metrics which are proposed earlier in this paper.

More specifically, Classifier algorithm classifies courses through three distinct stages. The first stage of the algorithm aims to identify how rich or poor the educational content is through the use of the Enrichment metric. A high Enrichment value indicates a course with rich educational content while correspondingly a low Enrichment value points to a course with poor content that needs additional material to be added. In order to classify courses at this stage they are sorted in descending order according to their Enrichment value and are characterized either as courses with high or low Enrichment. Those courses whose Enrichment is higher than the average Enrichment of all courses are characterized as high Enrichment courses while the others as low Enrichment courses. This stage leads to two clusters of courses: those with rich and those with poor educational material respectively.

The second stage of the algorithm tries to spot how often course information is added or updated by educators. At this stage the algorithm further classifies the previous set of courses using the Homogeneity value. The higher the Homogeneity value the more frequently 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 at this stage depends on both the average Enrichment value and the average Homogeneity value of the high and low Enrichment clusters accordingly. Therefore, this stage of the algorithm results in four clusters.

The third stage takes into consideration UPCS value in order to check whether users find each course useful and as a consequence visit its pages. Therefore, the previous clusters were further split into high and low UPCS courses according to UPCS average value.

The outcome of the algorithm execution is eight clusters of courses. Each cluster is constituted by courses with similar characteristics. The Classifier algorithm provides a description of each cluster courses (Table 3) so that educators may be informed about the characteristics of their courses and make the appropriate improvements.

Table 3: Course classification according classifier algorithm

Cluster ID	Enrichment	Homogeneity	UPCS	Course Classification
I	High	Frequently Updated	High	Frequent course content updates visited by users
II	High	Frequently Updated	Low	Frequent course content updates not visited by users
III	High	Static Content	High	Static content visited frequently by users
IV	High	Static Content	Low	Static content visited occasionally by users
V	Low	Frequently Updated	High	Abandoned course open for view
VI	Low	Frequently Updated	Low	Garbage course that needs further justification
VII	Low	Static Content	High	Poor content that still contains useful information for students
VIII	Low	Static Content	Low	Abandoned course

Data Mining Techniques

Data mining techniques have been applied to E-learning systems data by many researchers. Apart from the analytical review by Romero and Ventura (2007), there is some more domain specific. Castro, Vellido, Nebot, and Mugica (2007), among others, deal with the assessment of the students' learning performance, provide course adaptation and learning recommendations based on the students' learning behaviour, deal with the evaluation of learning material and educational web-based courses, provide feedback to both teachers and students of e-learning courses, and detect a typical student's learning behaviour. Another similar study described the detection of typical behaviour in the grouping structure of the users of a virtual campus (Castro, Vellido, Nebot, & Minguillon, 2005). A survey by Koutri, Avouris, and Daskalaki (2005) provides an overview of the state of the art in research of web usage mining, while discussing the most relevant criteria for deciding on the suitability of these techniques for building an adaptive web site. Two relevant studies, one (Kotsiantis, Pierrakeas, & Pintelas, 2004) predicts the students' performance as well as to assess the relevance of the attributes involved and the other (Minaei-Bidgoli, Kashy, Kortemeyer, & Punch, 2003) predicts student performance by applying data mining methods in an educational web-based system.

Students are assessed in the final exams of the courses, and they are assigned a grade according to their performance on the course. Having measured the students' activity of the E-learning system according to the indexes and metrics, it is possible to investigate whether there is a relationship between student activity in the platform of the E-learning system and the grades of the students in the final exams.

In this sub-section we try to discover the relationship between the attribute Grade and the other attributes shown in Table 2.

There are various classification methods. Kotsiantis, Pierrakeas, and Pintelas (2003) compared six classification methods (Naive Bayes, decision tree, feed-forward neural network, support vector machine, 3-nearest neighbour, and logistic regression) to predict drop-outs in the middle of a course. In the *classification* step, the algorithm *1R* (Witten & Frank, 2005) may be applied. It uses the minimum-error attribute for prediction, discretizing numeric attributes (Holte, 1993). The attribute Grade has to be used as class since it describes the education outcome. In this step the attribute/s which best describe the classification will be discovered.

The *clustering* step contains course clustering, based on the Grade attribute. Clustering of user visits is performed with the use of k-means algorithm (MacQueen, 1967), an efficient partitioning algorithm that decomposes the data set into a set of k disjoint clusters. It is a repetitive algorithm in which the items are moved among the various clusters until they reach the desired set of clusters. With this algorithm both a large degree of similarity for the items in the same cluster and a noticeable difference in items which belong to different clusters are achieved. The Manhattan distance will be used instead of the default Euclidean distance, so that the centroids will be computed as the component-wise median rather than mean. The number of clusters is proposed to be 2, since our goal is to separate the courses into high activity and low activity ones.

Association rule mining is one of the most well studied data mining tasks. It discovers relationships among attributes in databases, producing if-then statements concerning attribute-values (Agarwal, Imielinski, & Swami, 1993). An association rule X ⇒ Y expresses a close correlation among items in a database, in which transactions in the database where X occurs, there is a high probability of having Y as well. In an association rule X and Y are called respectively the antecedent and consequent of the rule. The strength of such a rule is measured by values of its support and confidence. The *confidence* of the rule is the percentage of transactions with antecedent X in the database that also contain the consequent Y. The *support* of the rule is the percentage of transactions in the database that contain both the antecedent X and the consequent Y in all transactions in the database.

The Weka system has several association rule-discovering algorithms available. The Apriori algorithm (Agarwal, Mannila, Srikant, Toivonen, & Verkamo, 1996) will be used for finding association rules over discretized LMS data. Apriori (Agrawal & Srikant, 1994) is the best-known algorithm to mine association rules. It uses a breadth-first search strategy to count the support of item sets and uses a candidate generation function that exploits the downward closure property of support. It iteratively reduces the minimum support until it finds the required number of rules with the given minimum confidence.

There are different techniques of categorization for association rule mining. Most of the subjective approaches involve user participation in order to express, in accordance with his/her previous knowledge, which rules are of interest. One technique proposes the division of the discovered rules into three categories (Minaei-Bidgoli, Tan, & Punch, 2004). (1) *Expected and previously known*: This type of rule confirms user beliefs, and can be used to validate our approach. Though perhaps already known, many of these rules are still useful for the user as a form of empirical verification of expectations. For education, this approach provides opportunity for rigorous justification of many long held beliefs. (2) *Unexpected*: This type of rule contradicts user beliefs. This group of unanticipated correlations can supply interesting rules, yet their interestingness and possible action ability still requires further investigation. (3) *Unknown*: This type of rule does not clearly belong to any category and should be categorized by domain specific experts.

Case Study

In this section, the results after the application of usage analysis methodology will be described. The dataset was collected from a real E-learning environment used at the Kavala Institute of Technology that operates the Open eClass e-learning platform (GUNet, 2011). 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 is shown in Table 4.

Table 4: eClass data from log file

remote_host	request_uri	remote_logname	remote_user	request_method	request_time	request_protocol	status	bytes_sent	referrer	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	/newsfeeds.html	-	-	GET	[03/Mar/2009:18:58:01 +0200]	HTTP/1.1	200	7967	-	Mozilla/5.0(compatible; Googlebot/2.1; +http://ww...

As described in the second step of the methodology, the produced log file is filtered and preprocessed in order to include the following fields: courseID, sessionID and page Uniform Resource Locator (URL).

In the third and the fourth steps, the indexes are computed and the metrics are calculated.

The results for the 39 courses are presented in Table 5. The data are ranked in descending order in the column Final Score.

Table 5: E-Learning data and grade for 39 Courses

Course ID	Sessions	Pages	Unique pages	UPCS	Enrichment	Homogeneity	Quality	Final score	Grade
IMD105	91	297	11	216	0.963	0.121	0.542	117.055	6.78
IMD35	87	338	8	179	0.976	0.092	0.534	95.612	6.14
IMD132	152	230	7	184	0.970	0.046	0.508	93.437	5.75
IMD36	72	217	7	134	0.968	0.097	0.532	71.353	6.16
IMD125	93	164	6	134	0.963	0.065	0.514	68.871	6.13
IMD129	75	209	6	131	0.971	0.080	0.526	68.860	6.21
IMD41	98	185	8	129	0.957	0.082	0.519	66.976	5.87
IMD66	56	144	9	107	0.938	0.161	0.549	58.754	5.96
IMD17	53	206	11	89	0.947	0.208	0.577	51.360	6.83
IMD111	33	142	9	79	0.937	0.273	0.605	47.769	6.96
IMD8	45	135	8	82	0.941	0.178	0.559	45.859	5.34
IMD11	51	108	6	80	0.944	0.118	0.531	42.484	6.11
IMD44	48	82	10	75	0.878	0.208	0.543	40.739	5.93
IMD26	50	90	8	71	0.911	0.160	0.536	38.024	5.99
IMD61	32	74	9	64	0.878	0.281	0.580	37.108	5.78
IMD98	32	113	9	61	0.920	0.281	0.601	36.649	6.96
IMD62	23	94	11	52	0.883	0.478	0.681	35.392	7.12
IMD14	45	122	7	59	0.943	0.156	0.549	32.396	5.82
IMD133	25	80	7	54	0.913	0.280	0.596	32.198	6.75
IMD34	38	113	6	56	0.947	0.158	0.552	30.934	6.21
IMD115	18	73	9	42	0.877	0.500	0.688	28.911	7.15
IMD9	26	105	12	42	0.886	0.462	0.674	28.292	7.10
IMD112	30	62	6	46	0.903	0.200	0.552	25.374	6.18
IMD122	33	71	7	45	0.901	0.212	0.557	25.054	6.23
IMD120	38	80	3	46	0.963	0.079	0.521	23.953	6.07
IMD64	22	47	7	39	0.851	0.318	0.585	22.800	6.23

Course ID	Sessions	Pages	Unique pages	UPCS	Enrichment	Homogeneity	Quality	Final score	Grade
IMD80	14	38	7	34	0.816	0.500	0.658	22.368	6.89
IMD60	22	43	5	40	0.884	0.227	0.555	22.220	6.02
IMD50	22	46	7	38	0.848	0.318	0.583	22.154	6.11
IMD10	17	61	8	28	0.869	0.471	0.670	18.752	7.03
IMD114	28	42	4	34	0.905	0.143	0.524	17.810	5.79
IMD21	11	25	8	24	0.680	0.727	0.704	16.887	6.95
IMD23	30	38	4	32	0.895	0.133	0.514	16.449	6.51
IMD96	20	31	5	30	0.839	0.250	0.544	16.331	5.83
IMD130	12	30	5	22	0.833	0.417	0.625	13.750	6.71
IMD134	25	27	4	27	0.852	0.160	0.506	13.660	6.42
IMD15	11	24	7	20	0.708	0.636	0.672	13.447	6.79
IMD49	14	23	5	21	0.783	0.357	0.570	11.967	6.03
IMD67	18	23	4	22	0.826	0.222	0.524	11.531	5.71

Application of Classification Algorithm

The Classification algorithm classifies the courses. The 39 courses were initially ranked according to the Enrichment metric. The algorithm was tested by picking the best and worst 12 LMS courses from a list of 39 courses, which are shown in Table 6. That is, best and worst cases from students' usage point of view.

Table 6. 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 5 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 Table 7.

Table 7: 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	0.667	0.677
		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 7, for each one of the four classes the LMS courses are ordered based on the UPCS metric value. So courses IMD105 and IMD35 are the representatives of Cluster I which means courses with rich educational content which is frequently updated and high UPCS values. Respectively IMD36 and IMD129 belong to Cluster II and have identical characteristics with first cluster courses but low UPCS value. In Table 8, these courses and the classifier algorithm evaluation feedback for each one of these courses are presented.

Table 8: Clustering of the 12 courses based on the classifier algorithm

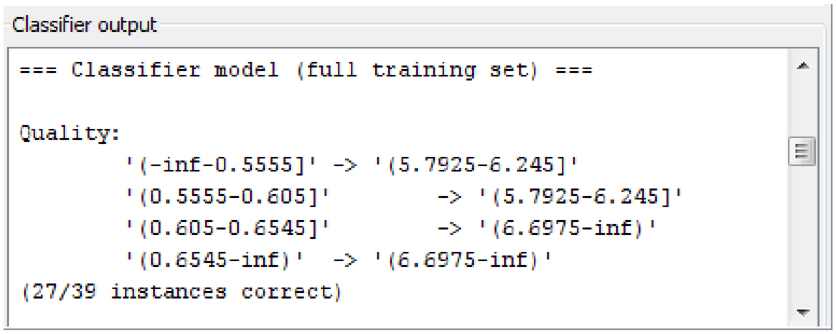
Cluster ID	Course ID	CCA Evaluation
I	IMD105, IMD35	High Activity LMS with updates followed by users
II	IMD36, IMD129	High Activity LMS with frequent educator updates that are not followed by users
III	IMD132	High Activity LMS with Static content, frequently updated and followed by users
IV	IMD125, IMD41	High Activity LMS with Static content, frequently updated but poorly followed by users
V	-	Abandoned course of dynamic content, open for view
VI	IMD112, IMD122	Garbage course or Forum with updates- Need for further evaluation
VII	IMD66	Course of poor static content that still contains information followed by users (or forced to follow)
VIII	IMD17, IMD8	Abandoned course of poor static content occasionally followed by curious users

Data Mining Techniques

The application of the data mining techniques was achieved with the use of the open source data mining tool Weka (Witten & Frank, 2005). Weka is open source software that provides a collection of machine learning and data mining algorithms for data pre-processing, classification, regression, clustering, association rules and visualization. The attributes of Table 5 were inserted in .cvs format into Weka. The attributes Course ID, Homogeneity, and Enrichment were removed before the application of classification, clustering, and association. Course_ID is different for each instance, and Homogeneity and Enrichment are dependent on the other attributes by formulas. All the remaining attributes were discretized.

Classification

In the classification step, the algorithm *1R* is applied. The attribute Grade is used as class. The results (Figure 2) show that the best attribute which describes the classification is Quality. This means that Quality is more closely related to Grade than the other variables and therefore courses with higher usage help students to achieve better grades and improve their educational performance.

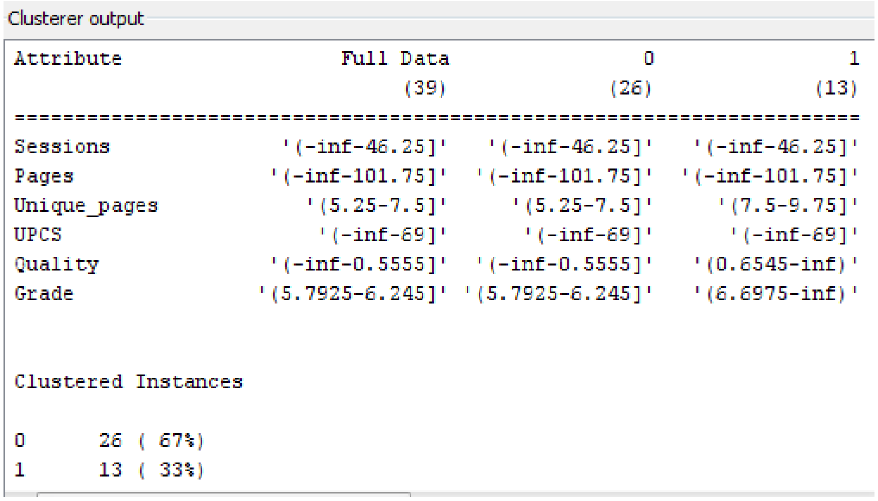


```
=== Classifier model (full training set) ===

Quality:
        '(-inf-0.5555]' -> '(5.7925-6.245]'
        '(0.5555-0.605]'        -> '(5.7925-6.245]'
        '(0.605-0.6545]'        -> '(6.6975-inf)'
        '(0.6545-inf)'  -> '(6.6975-inf)'
(27/39 instances correct)
```

Figure 2: Classification using Grade as class.

Clustering

The clustering step contains course clustering, based on the Grade attribute with the use of the *SimpleKmeans* algorithm (Kaufmann & Rousseeuw, 1990; MacQueen, 1967). The number of clusters is defined as 2 and the used distance is *Manhattan* instead of the default Euclidean distance. The produced results (Figure 3) show that 13 (33%) of the courses had high activity and 26 (67%) of the courses had low activity. Since previously presented results show that quality is closely related to the students' grades, authors should improve their courses in order to increase their quality and as a consequence the course usage by the students.



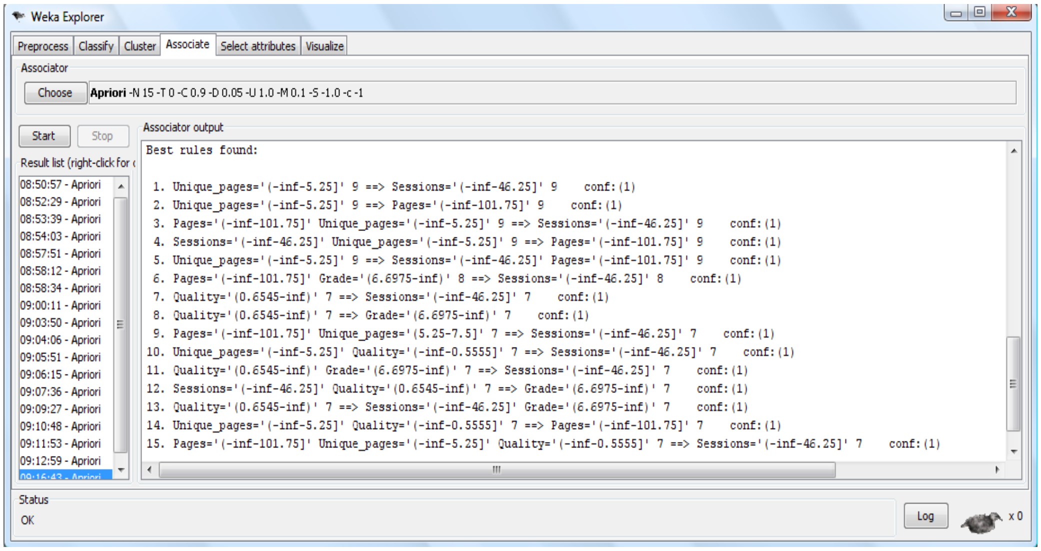
```
Attribute                    Full Data             0             1
                                  (39)          (26)          (13)
====================================================================
Sessions               '(-inf-46.25]'  '(-inf-46.25]'  '(-inf-46.25]'
Pages                 '(-inf-101.75]' '(-inf-101.75]' '(-inf-101.75]'
Unique_pages             '(5.25-7.5]'    '(5.25-7.5]'    '(7.5-9.75]'
UPCS                      '(-inf-69]'     '(-inf-69]'     '(-inf-69]'
Quality              '(-inf-0.5555]' '(-inf-0.5555]'  '(0.6545-inf)'
Grade               '(5.7925-6.245]' '(5.7925-6.245]' '(6.6975-inf)'


Clustered Instances

0      26 ( 67%)
1      13 ( 33%)
```

Figure 3: Clustering using Grade as class.

Association rule mining

The Apriori algorithm (Agarwal et al., 1996) was used to find association rules over the discretized E-Learning data Table 3 of the 39 courses, executing this algorithm with a minimum support of 0.1 and a minimum confidence of 0.9 as parameters.

Because of the obvious dependencies of the attributes Sessions, Pages, and Unique Pages with the attributes Enrichment and Homogeneity, the latter group of attributes was removed from the data table. Weka shows a list of 15 rules (Figure 4) with the support of the antecedent and the consequent (total number of items) at 0.1 minimum, and the confidence of the rule at 0.9 minimum (percentage of items in a 0 to 1 scale).



Apriori -N 15 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

```
08:50:57 - Apriori
08:52:29 - Apriori
08:53:39 - Apriori
08:54:03 - Apriori
08:57:51 - Apriori
08:58:12 - Apriori
08:58:34 - Apriori
09:00:11 - Apriori
09:03:50 - Apriori
09:04:06 - Apriori
09:05:51 - Apriori
09:06:15 - Apriori
09:07:36 - Apriori
09:09:27 - Apriori
09:10:48 - Apriori
09:11:53 - Apriori
09:12:59 - Apriori
09:16:43 - Apriori
```

```
Best rules found:

 1. Unique_pages='(-inf-5.25]' 9 ==> Sessions='(-inf-46.25]' 9    conf:(1)
 2. Unique_pages='(-inf-5.25]' 9 ==> Pages='(-inf-101.75]' 9    conf:(1)
 3. Pages='(-inf-101.75]' Unique_pages='(-inf-5.25]' 9 ==> Sessions='(-inf-46.25]' 9    conf:(1)
 4. Sessions='(-inf-46.25]' Unique_pages='(-inf-5.25]' 9 ==> Pages='(-inf-101.75]' 9    conf:(1)
 5. Unique_pages='(-inf-5.25]' 9 ==> Sessions='(-inf-46.25]' Pages='(-inf-101.75]' 9    conf:(1)
 6. Pages='(-inf-101.75]' Grade='(6.6975-inf)' 8 ==> Sessions='(-inf-46.25]' 8    conf:(1)
 7. Quality='(0.6545-inf)' 7 ==> Sessions='(-inf-46.25]' 7    conf:(1)
 8. Quality='(0.6545-inf)' 7 ==> Grade='(6.6975-inf)' 7    conf:(1)
 9. Pages='(-inf-101.75]' Unique_pages='(5.25-7.5]' 7 ==> Sessions='(-inf-46.25]' 7    conf:(1)
10. Unique_pages='(-inf-5.25]' Quality='(-inf-0.5555]' 7 ==> Sessions='(-inf-46.25]' 7    conf:(1)
11. Quality='(0.6545-inf)' Grade='(6.6975-inf)' 7 ==> Sessions='(-inf-46.25]' 7    conf:(1)
12. Sessions='(-inf-46.25]' Quality='(0.6545-inf)' 7 ==> Grade='(6.6975-inf)' 7    conf:(1)
13. Quality='(0.6545-inf)' 7 ==> Sessions='(-inf-46.25]' Grade='(6.6975-inf)' 7    conf:(1)
14. Unique_pages='(-inf-5.25]' Quality='(-inf-0.5555]' 7 ==> Pages='(-inf-101.75]' 7    conf:(1)
15. Pages='(-inf-101.75]' Unique_pages='(-inf-5.25]' Quality='(-inf-0.5555]' 7 ==> Sessions='(-inf-46.25]' 7    conf:(1)
```

OK

Figure 4: Weka executing Apriori algorithm based on confidence metric.

The application of the Apriori algorithm for association provided useful insights into the LMS usage. Figure 4 shows how a large number of association rules can be discovered. There are also some uninteresting and random rules, like rule 9, or redundant rules (rules with a generalization of relationships of other rules, like rule 5 with rules 1 and 2, and rule 13 with rules 7 and 8). There are some similar rules, rules with the same element in antecedent and consequent but interchanged, such as rules 12 and 13, and 3, 4, and 5. There is also a similar couple of rules, such as rules 12 and 13. But there are also rules that show relevant information, like those that show expected or conforming relationships (such as rules 2 and 4), since there is an indirect dependency of Unique Pages with Pages). And there are also rules that show interesting relationships for educational purposes (such as rules 8, 12, and 13), which can be very useful for the instructor in decision making about the activities of their courses. It is proved that high grades, values greater than 6.9675, correspond to courses with high Quality (rule 6). Also, in rules 11, 12, and 13 there is a composite dependency between the attributes Sessions and Grade. The last two remarks are useful for an instructor, since he/she can pay more attention to the courses with low values of Quality.

Practical Implications

The application of data mining techniques proved that there is a relationship between course usage and the corresponding student Grade. Additionally, rules 8, 12, and 13 offer to the instructors a lot of action ability, since they can pay more attention to the courses with low values of Quality and Sessions. The instructor can be motivated to increase the quality and the quantity of the educational material. More sessions means that more users (students) use the E-Learning platform.

However there is a number of students who try to read the materials only just before the exams. None deny that a good site with good material and which is updated frequently exists but rarely visited. On the other hand, a bad website may have frequent visits because students’ visits are related to their expected grade. So, the frequency and pattern of individual student accessing instructor’s materials will be an indicator to show such student’s “Laziness” or “Diligence”.

Feedback about the approach 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 organized in a few pages have a low interest score. They were asked again to re-organize the material for each course in the E-Learning according to the order they are taught, in order to facilitate use by the students.

The fact that there were investigated only 39 courses in one platform only is a limitation for the study. Especially, the data mining techniques demand larger datasets. However this was ineluctable since case study department had this number of online courses.

In the future we intend to apply the same approach in other universities in other countries with different cultures. In some different cultures the face-to face learning is preferable. It is remarkable that e-Learning usually stands together with conventional learning and supplements it rather than fully substituting it.

In some countries with strict hierarchical structure, people learn not to disagree with those who are older than themselves. So, with E-Learning, a barrier may exist which modifies the interaction between the teacher and the student.

From a pedagogical point of view this approach 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 with increased usage of the course material. According to our experiment, results usage is closely related to students' grades. An increased usage leads to better students' grades and therefore to an improved educational outcome.

Conclusions

The proposed approach uses existing techniques in a different way to perform E-Learning usage analysis. The metrics enrichment, homogeneity, and interest are used. Clustering, classification, and association rule mining of the courses usage is presented. Two algorithms for course classification and suggested actions are used.

It has the following advantages. (i) It is independent of a specific platform, since it is based on the Apache log files and not the platform itself. Thus, it can be easily implemented for every E-Learning platform. (ii) It uses indexes and metrics in order to facilitate the evaluation of each course in the E-Learning and allows the instructors to make any necessary adjustments to their course educational material. (iii) It applies data mining techniques to the E-Learning data. (iv) One algorithm for the E-Learning data classification of the courses is used.

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 and Heffernan (2005), where after long term observation the instructors were informed automatically by email about the quality of the content of their E-Learning courses.

It should be mentioned that even if the scope of the method is on E-Learning platforms and educational content, it can be easily adopted by other web applications such as e-government, e-commerce, e-banking, or blogs. Furthermore, Enrichment and Homogeneity metrics may also be used, for example, by e-government applications, since enrichment shows how much information is handed over to the end user and homogeneity characterizes the percentage of information independently discovered by each user.

References

Agrawal R., Imielinski, T., & Swami, A. N. (1993). Mining association rules between sets of items in large databases. *Proceedings of SIGMOD (pp. 207-216).*

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo A. (1996). Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining* (pp. 307- 328). Menlo Park, CA: AAAI/MIT Press.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of 20th International Conference on Very Large Data Bases* (pp. 487-499).

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.*

Baker, R., Corbett, A., & Koedinger, K. (2004). Detecting student misuse of intelligent tutoring systems. *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems, (pp.531–540).*

Castro, F., Vellido, A., Nebot, A., & Minguillon, J. (2005). Detecting a typical student behaviour on an e-learning system. *In Simposio Nacional de Tecnologı´as de la Informacin y las Comunicaciones en la Educacion* (pp. 153–160). Spain: Granada.

Castro, F., Vellido, A., Nebot, A., & Mugica, F. (2007). Applying data mining techniques to e-learning problems: A survey and state of the art. In L. C. Jain, R. Tedman, & D. Tedman (Eds.), *Evolution of teaching and learning paradigms in intelligent environment. Studies in Computational Intelligence 62*, Springer-Verlag.

Cone, J. W., & Robinson, D. G. (2001). The power of e-performance. *Training & Development, 55*(6), 32–41.

Delacey, B., & Leonard, D. (2002). Case study on technology and distance in education at the Harvard Business School. *Educational Technology and Society, 5*(2), 13-28.

Feng, M., & Heffernan, N.T. (2005). Informing teachers live about student learning: Reporting in the assistment system. *Proceedings of the 12th Annual Conference on Artificial Intelligence in Education 2005, Amsterdam.*

Fichter, D. (2002). Intranets and e-learning: a perfect partnership. *Online, 26*(1), 68-71.

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.

GUNet. (2011). Retrieved April 10, 2011 from http://eclass.gunet.gr/

Hamalainen, W., & Vinni, M. (2006). Comparison of machine learning methods for intelligent tutoring systems. *Procceedings of Int. Conf. in Intelligent Tutoring Systems,* (pp. 525–534).

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning, 11*, 63-91.

Kaufmann, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. New York: John Wiley & Sons.

Kazanidis, I., Valsamidis, S., & Theodosiou, T. (2009). Proposed framework for data mining in e-learning: The case of Open e-Class. *Proceedings of Applied Computing 09, Rome, Italy.*

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.

Kotsiantis, S. B., Pierrakeas, C. J. & Pintelas, P. E. (2003). Preventing student dropout in distance learning using machine learning techniques. *Proceedings of 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES)*, (pp 267–274).

Kotsiantis, S. B., Pierrakeas, C. J., & Pintelas, P. E. (2004). Predicting students' performance in distance learning using machine learning techniques, *Applied Artificial Intelligence (AAI), 18*(5), 411 – 426.

Koutri, M., Avouris, N., & Daskalaki, S. (2005). A survey on web usage mining techniques for web-based adaptive hypermedia systems. In S. Y. Chen & G. D. Magoulas (Eds.), *Adaptable and adaptive hypermedia systems* (pp. 125–149). IRM Press.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, (pp. 281–297). California, USA.*

Mazza, R., & Dimitrova, V. (2007). CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. *International Journal of Human-Computer Studies, 65*(2), 125–139.

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, Italy.

Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., & Punch, W. (2003). Predicting student performance: An application of data mining methods with an educational web-based system. *Proceedings of 33rd Frontiers in Education Conference*, (pp. T2A13–T2A18).

Minaei-Bidgoli, B., Tan, P-N, & Punch, W.F. (2004). Mining interesting contrast rules for a web-based educational system. *Proceedings of Int. Conf. on Machine Learning Applications, Louisville, USA 2004* (pp. 320- 327).

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).

Normark, O. R., Cetindamar, D. (2005) E-learning in a competitive firm setting. *Innovations in Education & Teaching International, 42(4)*, 325-335.

Pabarskaite, Z., & Raudys, A. (2007). A process of knowledge discovery from web log data: systematization and critical review. *Journal of Intelligent Information Systems, 28*, 79-114.

Romero, C., Gutierez, S., Freire, M., & Ventura, S. (2008). 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 (pp. 182-186). Montreal, Quebec, Canada.*

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.

Radcliffe, D. (2002). Technological and pedagogical convergence between work-based and campus-based learning. *Educational Technology and Society, 5*(2), 54-59.

Srinivasa, R. (2005). Data mining in e-commerce: A survey. *Sadhana, 30*(2 & 3), 275–289.

Starr, R. M. (1977) Delivering instruction on the World Wide Web: Overview and basic design principles. *Educational Technology, 37*(3), 7-15.

Ueno, M. (2002). Learning-log database and data mining system for e-learning, *Proceedings. of International Conference on Advanced Learning Technologies, ICALT 2002,* (pp. 436-438).

Valsamidis, S., Kazanidis, I., Kontogiannis, S., & Karakos, A. (2010). Automated suggestions and course ranking through web mining. *Proceedings of 10th IEEE International Conference on Advanced Learning Technologies ICALT 2010,* Sousse, Tunisia.

Valsamidis, S., Kontogiannis, S., Kazanidis, I., & Karakos, A. (2010). Homogeneity and enrichment: Two metrics for web applications assessment. *Proceedings of 14th Panhellenic Conference on Informatics*.

Witten, I., & Frank, E., (2005). *Data mining practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann.

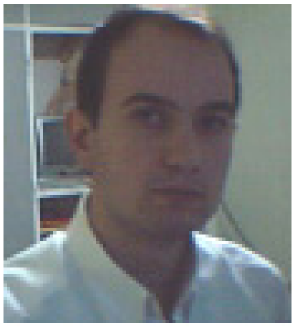
Yan, T. W., Jacobsen, M., Garcia-Molina, H., & Dayal, U. (1996). From user access patterns to dynamic hypertext linking. *Proceedings of the Fifth International World Wide Web Conference on Computer networks and ISDN systems, (pp. 1007-1014), Paris, France.*

Zaiane, O. R. (2001). Web usage mining for a better web-based learning environment. *Proceedings of Conference on Advanced Technology for Education.*

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, (pp. 611-613).*

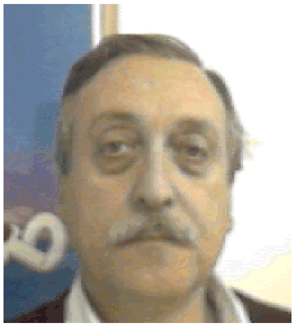
Biographies

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 Professor in the Dept. of Accountancy, Kavala Institute of Technology, Greece. His research interests are in the areas of database and data mining, data analysis and web applications assessment. His e-mail is: svalsam at ee.duth.gr.



S. Kontogianns 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, middleware protocol design, network modelling and computer networks performance evaluation. His e-mail is: skontog at ee.duth.gr.

Ioannis Kazanidis received the MSc degree in Computing from the Coventry University and the PhD degree in Educational Technology and Adaptive Educational Systems at University of Macedonia. Currently he is an adjoined assistant professor at Kavala Institute of Technology Greece. His main research interests lie in the area of adaptive systems, data mining methods and algorithms, and computer supported collaborative learning. He has published 6 papers in International Journals and Book chapters, and more than 12 papers in proceedings of conferences.



A. Karakos received the Degree of Mathematician from the Department of Mathematics from Aristotle 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 at ee.duth.gr.