Data Mining Technology for the Evaluation of Web-based Teaching and Learning Systems

Claus Pahl, Dave Donnellan
Dublin City University
School of Computer Applications
Dublin 9, Ireland
[cpahl|ddonnell]@computing.dcu.ie

Abstract: Instructional design for Web-based teaching and learning environments causes problems for two reasons. Firstly, virtual forms of teaching and learning result in little or no direct contact between instructor and learner, making the evaluation of course effectiveness difficult. Secondly, the Web as a relatively new teaching and learning medium still requires more research into learning processes with this technology. We propose data mining – techniques to discover and extract knowledge from a database – as a tool to support the analysis of student learning processes and the evaluation of the effectiveness and usability of Web-based courses. We present and illustrate different data mining techniques for the evaluation of Web-based teaching and learning systems.

Motivation

Web technologies allow us to create Web-based teaching and learning systems that support a variety of different educational features in an integrated form, including audio-based lectures, video-conferencing, interactive tutorials, and various forms of animations and simulations. This opportunity to create novel and innovative course material is currently the focus of an area of active research. Instructional design for Web-based courses is a relatively young discipline. Important quality criteria are the effectiveness of the approach and the usability of the system. Two central difficulties can be identified in this context. Firstly, there is a lack of direct contact between instructor and learner, which makes it more difficult for the instructor to assess and evaluate the quality of teaching and learning. Secondly, new forms of teaching and learning are implemented for which methods of best practice do not exist. Therefore, an evaluation needs to create a broader picture of the process of student learning. Furthermore, an evaluation needs to be incorporated into an incremental, evolutionary instructional design approach.

Data mining technologies can provide essential support for this evaluation of learning processes based on Web technologies (Agrawal & Srikant 1995), (Zaiane & Luo 2001). Data mining is the discovery and extraction of knowledge from a database. Web mining is the application of data mining to the Web. Web logs, created by Web servers that record document requests on the Web, form the database. The classical application domain of data and Web mining are business systems. Some of the classical data and Web mining technologies can be used in the context of Web-based teaching and learning systems. However, teaching and learning systems differ from commerce and business systems substantially. The goal is a different, more long-termed one, which obviously effects the usage of a Web-based system. Frequency, regularity, number and purpose of visits and behaviour are different. As a consequence, some new technologies are needed to address in particular the process of learning. Techniques are required that for example identify and extract the students' learning processes from Web logs.

The objective of this paper is to introduce Web mining as an evaluation technique for educational systems. We will present data mining techniques, including some novel ones, that are suitable for the evaluation of Web-based teaching and learning systems. We will illustrate their use using a virtual course system taught by the first author since several years as a case study, (Pahl 2001a).

Data and Web Mining for Educational Systems

Data mining is concerned with the discovery and extraction of knowledge from a database. Typically, this knowledge can be classified into rules and patterns that help users in analysis and decision making processes. Web mining is the analysis of data in Web-based systems. Usually, the database is the access log created by a Web server. Each request of any text document or other type of resource – a URL – is recorded in the access log. Each entry in this log consists of the requester, the requested URL and time/date of the request. The typical format of an entry is described in Tab. 1.

Field	Description
Client	IP address
Ident	requestor ID (rarely used)
User	(authenticated) user name
Date	date of request
Method	HTTP GET or POST
Request	URL of requested document
Protocol	HTTP version
Status	success indicator (200 is success)
Bytes	bytes requested/transferred

Table 1: Web Log Format

An example of a few requests is the following:

```
136.206.18.130 - rkyne   [08/Nov/2001:11:38:15 +0000] "GET /CA309/ch5ov.html HTTP/1.0"  200 43
136.206.18.14  - lgavin  [08/Nov/2001:11:38:18 +0000] "GET /CA309/ch32c.html HTTP/1.1"  200 2048
136.206.18.16  - bahern [08/Nov/2001:11:38:25 +0000] "GET /CA309/Asgn.html HTTP/1.1"   200 2018
```

Besides HTML-documents, as in this example, any other file type including executables, images and other media can be requested.

Technically, Web mining based on access logs is a server-side evaluation technique that is in contrast to client-side evaluation of Web-based teaching and learning systems. Client-side evaluation can observe the students more directly and evaluates the student behaviour based on data that is acquired at the student's side, e.g. the Web browser. Usually, client-side techniques allow more precise observations. Server-side techniques can not capture all student activities. However, based on suitable implementations, the essential ones can be captured. Most dynamic Web pages result in log entries at the server side, or can generate these, e.g. in technologies like Java applets (Pahl 2001b). An advantage of the server-side technology is the possibility to monitor all students constantly without the need for any additional monitoring equipment to be installed. Data available from the Web logs can provide sufficient and adequate knowledge for the evaluation process.

Web mining for the educational context is based on a central concept, a basic unit of learning: a session. A session reflects a period of active learning of a particular student. The first step in actually applying educational Web mining to Web logs is to clean the logs and to partition the log into individual sessions. We define a session as a sequence $P = <P_1, ... , P_n>$ of requests P_i from one student for a period of time in which the student is active. A request P_i is in our case a page request, i.e. a URL. Inactivity for a period of about 20 minutes – a heuristically determined value – indicates the end of a session.

We propose the following techniques as central in a Web mining-based evaluation of teaching and learning systems:

- Session statistics: basic statistics about sessions such as average session length in time or in number of requests.
- Session patterns: the determination of student learning processes extracted from navigation and request behaviour.
- Time series of session data: the analysis of the development of session statistics and session patterns over a period of time.

Besides these three classes, other Web and data mining techniques exist that can provide useful information. Standard tools for Web mining, e.g. (Analog 2002), produce hit lists, i.e. rankings of URLs based on the number of requests for some time interval. This data might be used to analyse the visits of individual pages.

Session Statistics

A session is a sequence of Web log entries that reflects the navigation and request behaviour of a student in a period of active study. Some basic measures can help to answer for example questions about the investment of time per student for a given learning activity. The measurements that can be calculated from the Web log are for example:

- the session length in time,
- the session length in number of requests.

These measurements are simple numerical values derived from the session logs. Simple statistical analyses based on these values that provide some useful information including mean and standard deviations of length measures, or correlations between lengths in time and lengths in number of requests. The purpose of the statistical evaluation is for example to measure the investment of time of a student for the course or for a particular part of it. Any of the results can be compared against the expectations of the educator or instructional designer. Explicitly formulated expectations can form a master model.

There are other statistics that might result in useful insights – as we have briefly discussed in the previous section. The total number of requests by interval or total numbers ranked by resource provide relevant information for an evaluation. These measures give an idea about 'what' resources are used, but not 'how' they are used. The latter shall be addressed in the next section.

Session Patterns

The central knowledge for successful instructional design of Web-based courses is how students actually learn the best in a Web-based environment. Due to the lack of accepted standards of best practice, an evaluation of student learning can lead to useful insights. At the core of this form of evaluation is the extraction of the students' learning processes from the Web logs. Two forms of knowledge discovery are important:

- The extraction of purpose patterns from session data. The purpose pattern analysis is based on classification. The site resources are divided into different parts, e.g. lectures, tutorials, administrative/organisational material, etc. One or several main session purposes can be determined from the resource classification. Then, the overall association of sessions and purposes can be analysed. A somewhat surprising result of a purpose analysis for our virtual database course was that students very often log on to the system to look up organisational resources, such as timetables, exam related material, etc. More than 30% of all session included administrative/organisational elements.
- The extraction of behavioural patterns from session data. The behavioural pattern analysis has the determination of the student learning processes as the objective. The starting point is the Web log. The Web log provides only sequences of resource requests. However, behavioural concepts such as repetition need to be identified. The instructional design might work with concepts such as options and choice. As a consequence, the actual behaviour (as a sequence) has to be measured against an expert model that reflects the instructor's intended or expected learning process and learning path.

The purpose of both analyses is to determine how students learn. This is reflected in the Web logs through the way students navigate and which resources they request, and in which order. As we have explained earlier on, this approach can capture the essential learning activities.

The behavioural pattern analysis is an advanced and novel technology that we will illustrate here. The starting point is the extraction of sequential patterns based on sequences of log entries in sessions. Let $Q = <Q_1, ... , Q_m>$ be a sequence of log entries. A sequence $P = <P_1, ... , P_n>$ is contained in sequence Q if $P_1 = Q_{i1} , ... , P_n = Q_{in}$ such that $i1 < ... < in$. This means that each element of the P-sequence can be found in the Q-sequence, and additionally that the P-elements appear in the Q-sequence in the same order in which they appear in the P-sequence. The idea of containment is necessary to filter out activities not relevant for the learning process under investigation – for instance students might go back to previous pages, lookup other pages, even leave the system temporarily. The P-sequence is a candidate sequential pattern. Elements of Q that are not in P are the irrelevant

requests. A sequence is called maximal if it is not contained in any other sequence. Maximality allows us to ignore shorter sequences that are contained in others. These would not provide any additional information.

In order to find out what patterns students follow, we need to look at the number of students that follow a particular sequence in a session. A student supports a sequence if the sequence can be found in any of that student's sessions. The support of a sequence is defined as the fraction of all students that support this sequence. A sequential pattern is a maximal sequence that has a certain minimum support. The choice of the minimum depends on the system and the objectives of the analysis. It needs to be determined heuristically. A high minimal support will only reveal patterns that are supported by a vast majority of students. A low minimum will show more patterns, which in extreme cases reflect more the behaviour of individual students than that of the whole group. The site structure, and in particular the degree of choice has an influence on the best choice of the threshold. For systems with a high degree of choice, the threshold should be low in order to detect common behaviour.

The sequential patterns describe the actual behaviour of students. The higher the support for a sequence is, the more often this path is followed by students. Typical sequences with very high support are sequences leading from the home page via a table of contents into some resource page:

HomePage ; TableOfContents ; Chapter1Overview ; …

The main purpose of the sequential pattern extraction here is that they can be compared to expected behavioural patterns specified by the instructional designer or educator. Sequential pattern determination is a standard data mining technique, see e.g. (Agrawal & Srikant 1995), that we have adapted here to a Web environment. The behavioural pattern analysis, which we are going to present now in more detail, is a novel technique.

Behavioural pattern analysis is a design tool for instructional designers for the development of Web-based virtual courses. A model of the course topology – the navigation infrastructure and the interactive elements integrated into dynamic pages – underlies the specification of behavioural patterns. A behavioural pattern is a path expression on the course topology that describes an ideal learning path. The following is an example from a tutorial part of our virtual database course:

Tut1;[ExecQuery|Scaffold]+;Tut2; [ExecQuery|Scaffold]+; . . . ;Tut12;[ExecQuery|Scaffold]+

This is a behavioural pattern describing the intended learning path in our interactive tutorial service that teaches programming skills in the database language SQL. This service consists of twelve tutorial units *Tut1, ... , Tut12*. Within each unit students can iteratively (denoted by +) either execute an SQL query (*ExecQuery*) or use any of the scaffolding features (*Scaffold*) that are provided to support the learner. The semicolon denotes sequential composition, the brackets *[..]* enclose optional elements, and choices are separated by a bar |. The control flow combinators for this path expression language are summarised in Tab. 2.

Name	Notation	Description
iteration	P^+	the page P can be access any number of times, but at least once
option	[P]	the page P might or might not be accessed
choice	P | Q	either P or Q can be accessed
sequence	P ; Q	the page Q will be accessed after page P

Table 2: Path Expression Notation for Behavioural Patterns – Combinators

It is important to note that we can see sequential patterns as path expressions of behavioural patterns. We can compare a specification of expected behaviour in terms of path expressions and actual sequential patterns. An ordering relation indicates whether an actual sequential pattern satisfies a behavioural pattern. An ordering $S \leq T$ on path expressions compares actual and intended use and determines whether the actual use conforms with the intended use: $S \leq T$ means that pattern expression S satisfies T. Typically, S will be a sequential pattern and T will be a behavioural pattern. A set of rules – summarised in Tab. 3 – allows us to determine whether a sequential pattern (or a behavioural one) satisfies a given behavioural pattern. In the table, the letters S, T, U, X, and Y stand for path expressions. The expression ST means that S and T are concatenated, i.e. sequentially composed.

Rule	Description
$T^+ \leq T$	means that actual repetitions of T are allowed
$S \leq [S]$	means that the user can choose to access S or not
$SU \leq S[T]U$	means that optional pages T can be left out

Table 3: Pattern Satisfaction Rules

A mathematical property shall be noted: the relation ≤ is reflexive, antisymmetric, and transitive, i.e. forms a partial ordering. A weaker variant of ≤, denoted with ⊆, can also be introduced: $STU \subseteq XY$ if $S \leq X$ and $U \leq Y$ which allows students to deviate for a while from the pattern path. Deviation, choice and repetition in actual navigation sequences are important concepts to express learning behaviour in a Web-based system.

The final step in the analysis is the determination of the support for a behavioural pattern. The support shall be defined as the fraction of sequential patterns that support the behavioural pattern. A high support for a behavioural pattern that is described by the instructor shows the effectiveness of the instructional design. For the aforementioned tutorial pattern, we have achieved 84% support, which shows that the instructional design has been accepted by the students.

The expressive power of the behavioural pattern notation can even be improved. The notation might be extended to include a parallel composition P || Q which says that pages P and Q can be accessed concurrently – e.g. using two Web browser windows. We shall ignore this possibility here. However, we would like to point out that logging and evaluating multi-window activity is important, and will help to obtain a more accurate analysis of student behaviour.

Time Series of Session Data

Time series are sequences of measurements over a period of time. These measurements can include session statistics and session patterns. The purpose is the detection of change in learning behaviour. This is important for two reasons. Firstly, change might be intended by the instructional designer and the actual occurrence of this change needs to be verified. Secondly, unexpected changes need to be detected.

An example for the first case is an evaluation of scaffolding features that we have carried out for our virtual course (Pahl 2002). Fading use of scaffolds – features that support students in becoming self-reliant and competent in a domain – is an essential characteristic that is expected to happen in an effective scaffolding implementation. An observation that leads to the second case is that, according to our experience, the more students work with computer-supported teaching and learning systems, the more they appreciate this new form of learning. More experience with these systems seems to lead to more effective forms of using the systems, and therefore more effective forms of learning. The evaluation of behavioural patterns can help us to extract the evolution of student learning towards the most effective form of learning from a Web log.

With respect to session statistics, for example the decrease in session lengths for a series of equally difficult tutorial units shows increasing self-reliance and competency of a student. With respect to session patterns we have observed that early patterns often show single purpose use, for example lectures only in a multi-service course system, but later patterns show multi-purpose usage, i.e. integrated usage of different educational services at the same time.

Conclusions

Essential knowledge for an instructional designer confronted with the development of courseware for the Web includes how students learn in a Web environment, i.e. whether students learn effectively and whether the instructional design is effective. In novel environments for teaching and learning, such as the Web, the determination of effective learning behaviour might need to be determined in an incremental and evolutionary way. The integration of instructional design and evaluation into a cyclical development approach is therefore essential. Only an iterative process of design and evaluation allows us determine and evaluate effective learning in a Web environment and to design the most effective courses for the Web. We have presented Web mining technologies for teaching and learning systems that can help us to answer essential questions about student

learning in a Web environment. We have presented a set of mining techniques suitable for the educational context – some of them novel techniques developed for the educational context.

The usage evaluation of Web-based systems can be classified into two dimensions: time and space. Usage in time addresses the frequency and regularity of usage, number of accesses, etc. Usage in space is concerned with usage patterns based on the course topology. Our central evaluation techniques for educational systems are evaluations in space. The combination with an evaluation in time can provide additional valuable information. Various tools that provide statistics on numbers of accesses, frequencies, etc. are available – see e.g. (Analog 2002). Tool support is critical for this Web mining-based type of analysis. In our own virtual database course, we usually have between 250000 and 400000 log entries per term. We have implemented our own tool support in particular for the space dimension.

We have looked at using Web mining for the purpose of behavioural analysis. However, the technology can be used to obtain a wider range of information. This could include monitoring individuals or groups of students, or the identification of weak students. The technique presented here is limited to activities in one browser window. If several windows were used concurrently, then this behaviour would have to be recognised as a concurrent one. A corresponding operator for the path expression notation has been suggested. The extension of the analysis towards concurrent activities is planned for the future. In a first step, the notation for behavioural patterns should be extended to encompass parallel activities and corresponding rules to determine satisfaction. In a second step, the pattern analysis should be extended from sequential to parallel patterns.

References

Agrawal, R., & Srikant, R. (1995). Mining Sequential Patterns. *Proc. 11th International Conference on Data Engineering ICDE*, Taipei, Taiwan.

Analog (2002). *Analog Logfile Analyser*. Web site: http://www.analog.cx

Donnellan, D. (2002). *User Session Classification Tool For The Analysis of Web Server Logs*. M.Sc. Dissertation. Dublin City University. School of Computer Applications.

Pahl, C. (2001a). Interactivity and Integration in Virtual Courses. *Proc. International Conference on Advanced Learning Technologies ICALT 2001*, 395-396. IEEE Computer Society.

Pahl, C. (2001b). XML-technologies for the Support of Active Learning in Web-based Virtual Learning Environments. *Proceedings 6th WebNet Conference*, Orlando, Florida, US. AACE.

Pahl, C. (2002). An Evaluation of Scaffolding for Virtual Interactive Tutorials. *Proc. 7th E-Learn 2002 Conference*, Montreal, Canada. AACE.

Zaiane, O.R., & Luo J (2001). Towards Evaluating Learners' Behaviour in a Web-Based Distance Learning Environment *Proc. International Conference on Advanced Learning Technologies ICALT'01*. 357-360. IEEE Computer Society.

Acknowledgements

This work has been supported by the Dublin City University Teaching and Learning Fund (contract tlf/2000-2001/small-project-00/2).