A Framework for Assessing Usage of Web-Based e-Learning Systems

Chien-Chung Chan
Department of Computer Science
University of Akron
Akron, OH 44325-4003
chan@cs.uakron.edu

Abstract

The knowledge of students enables an e-learning system to provide individualized lessons for each student. This paper presents the use of web usage mining method to discover the knowledge of students from their usage sequences. The structure of lessons offered by an e-learning system can be used as website ontology to facilitate usage mining. We also present a framework for representing assessment results and usage patterns by a hierarchy of flow graphs, which can be used as decision algorithms for evaluating relationships between usage patterns and performance of students. Usage patterns can be translated into decision rules or trees for predicting student performance and for providing timely lessons.

1. Introduction

Fast developments in communication and information technologies have fueled the widespread use of World Wide Web, which has changed all lines of works. Information in all kinds of forms is being made available on the Web daily. The Internet has become an information center that is accessible pervasively. It is indeed an excellent platform for delivering and acquiring information. However, raw information does not necessarily equal to knowledge. It takes extra processings to make information into knowledge. This is what e-learning or online learning systems can provide. They can engage users in a process of learning by providing necessary information in a timely and effective way.

According to the free encyclopedia Wikipedia, the term e-learning in general refers to computer-enhanced learning, and the first general-purpose e-learning system was the Programmed Logic for Automatic Teaching Operations (PLATO) developed by the Computer-based Education Research Laboratory (CERL) of the University of Illinois at Urbana-Champaign in early 1960's [1, 2]. Research in intelligent tutoring systems has been one of the major topics in AI [3]. One important quality of an intelligent tutoring system is the ability to provide an individualized learning environment for each student. Previous work has identified four major components of such a system: the student model, the pedagogical module, the domain knowledge module, and the communication module [4]. These components may be implemented as agent-based systems with varying number of agents. For example, in the Intelligent Multiagent Pedagogical System (IMAPS), which is a knowledge-based web-enabled system, there are seven agents [5].

Independent of the number of agents used in a system, it is certain that an effective e-learning system must tailor its learning lessons based on the performance of students. In order to accomplish this, we believe that the knowledge of usage patterns and performance of students might be useful for lesson planning.

In web-based learning systems, student interactions are recorded in log files of web servers. The basic idea of our work is to use web usage mining techniques to identify student groups based on their usage sequences. Then, a hierarchy of flow graphs [6] is used to relate and store assessment results and students' usage patterns. The flow graphs can be used to evaluate students' performance and to predict the needed guidance for lesson planning.

The paper is organized as follows. In Section 2, we discuss the representation of users' click streams and the structure of assessment trees. The steps and requirements of usage mining algorithms are outlined in Section 3. In Section 4, we present a brief review of the concept of flow graphs and related properties. An example of a hierarchy of flow graphs is given here. In Section 5, we discuss the simplification of usage patterns and flow graphs into decision rules. Conclusions are given in Section 6, followed by references.

2. Assumptions

Interactions between users and a web-based e-learning system are recorded in web server log files. The formats are defined by W3 consortium [7]. We assume that each user's usage of the system is represented by a pair of sequences: (P, T) where P = <p_1, …, p_n> is a sequence denoting the accessed page links and T = <t_1, …, t_n> is a sequence denoting time spent at corresponding pages

0-7695-2882-1/07 $25.00 ©2007 IEEE

capped by a maximum threshold $t_{\mathbf{max}}$. Actual causes of reaching the time limit, which may be actual time spent on learning activities or may be a long idling time, are not distinguished by the system. In other words, we do not consider the content of learning activities invoked at each page. However, the length of usage sequences of different users is taken into consideration by the system and they are not required to be the same, because pages may be visited multiple times.

We assume that each lesson of an e-learning system is a finite non-empty set of linked pages. The entire activity of taking a lesson by a user is represented as one usage sequence described above, and at the end of each activity, there is an assessment test. A course is a structure of lessons represented by an **assessment tree** with structure similar to a textbook's table of contents. It is a bipartite tree where internal nodes are called **assessment nodes** and terminal (leaf) nodes are called **usage nodes**. An assessment node stores information of assessment results, such as quizzes or tests scores. A leaf node stores the relationship of usage patterns and assessment results. We do not consider usage patterns related to assessments acquired in internal nodes. It is possible to allow review activities before taking assessments of internal nodes. Here we assume that internal nodes store only assessment results. Lesson labels of leaf nodes in the bipartite tree are used by usage mining algorithms to group usage sequences.

An example of assessment tree is shown in Figure 1. There are three chapters C_1, C_2, and C_3 in a given course C. Chapter C_1 has two sections C_{11} and C_{12}, Chapter C_2 has three sections C_{21}, C_{22}, and C_{23}, and Chapter C_3 has two sections C_{31} and C_{32}. Circle nodes are assessment nodes and rectangles are usage nodes, where group IDs and their assessments are stored. An example of the information stored in internal node C_3 is shown in Figure 2, where we assume that the assessment results are either Successful or Failed. The other three columns are Bayesian factors: strength (σ), certainty (cer), and coverage (cov), which will be defined in Section 4. An example of usage node C_{31} is shown in Figure 3, it is assumed that there are three groups of user usage sequences. An example of usage node C_{32} is shown in Figure 4, where the groups are independent of the groups in node C_{31}.The number of groups is data dependent and is generated by a usage mining component described in the next section.

3. Usage Mining

Typical web usage mining includes the following six steps: data cleaning, user identification, session identification, data filtering, clustering and classification [8, 9, 10, 11, 12]. In data cleaning phase, pages related to the navigation structure are kept. User identification is to determine which pages are accessed by which users, and the end result is that one user is associated with one sequence. Session identification is to break a user's usage sequence into subsequences called sessions, where each session corresponds to one website-defined activity. Data filtering is to remove house keeping pages which are not called directly by the user rather they are called internally by requested pages. These four steps consist of data preprocessing for clustering algorithms, which will then divide users with similar sequences into groups. Finally, classification algorithms can be applied to generate usage patterns for each group of sequences.

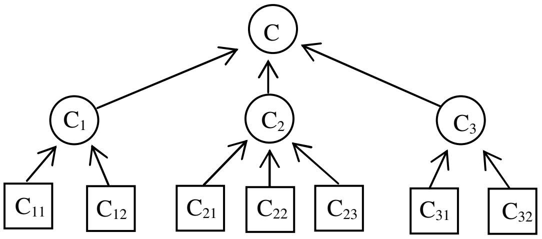

Figure 1. An assessment tree of lessons.

Table 1. Flow distribution stored in an assessment node:

C_{31}	C_{32}	C_3	σ	cer	cov
S	S	S	0.21	0.78	0.38
S	S	F	0.06	0.22	0.14
S	F	S	0.15	0.43	0.27
S	F	F	0.20	0.57	0.45
F	S	S	0.13	0.65	0.23
F	S	F	0.07	0.35	0.16
F	F	S	0.07	0.39	0.12
F	F	F	0.11	0.61	0.25

Table 2. Distribution of performance of lesson C_{31} based on three user usage groups:

Group_ID	C_{31}	σ	cer	cov
311	S	0.30	0.75	0.46
311	F	0.10	0.25	0.29
312	S	0.15	0.50	0.23
312	F	0.15	0.50	0.42
313	S	0.20	0.67	0.31
313	F	0.10	0.33	0.29

Table 3. Distribution of performance of lesson C_{32} based on two user usage groups:

Group_ID	C_{32}	σ	cer	cov
321	S	0.2	0.67	0.40
321	F	0.1	0.33	0.20
322	S	0.3	0.43	0.60
322	F	0.4	0.57	0.80

In usage mining for web-based e-learning systems, user and session identification are straightforward, because it is reasonable to assume that each user has a login ID and each session corresponds to each lesson ID. The data cleaning and data filtering are similar to web usage mining. For clustering algorithms, they must be able to deal with different sequence sizes, and similarity measures must take into consideration of times spent at pages.

4. Hierarchical Flow Graphs

Given a collection of users' usage sequences of a web-based e-learning system, the result of usage mining through a clustering algorithm is a set of clusters of usage sequences. Because each sequence denotes an activity of learning a lesson, so it is associated with an assessment result. It is clear that sequences in the same group may not have the same assessment result. To summarize the learning performance of usage groups, we use the approach of flow graph for intelligent data analysis introduced by Pawlak [6].

4.1. Flow Graphs

The concept of flow graphs has been used by many researchers for data analysis [13, 14, 15]. In the following, we will review the approach introduced in [6]. The basic idea is that each branch of a flow graph is interpreted as a decision rule and the entire flow graph describes a decision algorithm. Each decision rule is associated with three Bayesian coefficients, namely, strength, certainty, and coverage factors. More precisely, a flow graph is a directed acyclic finite graph G = (V, E, w), where V is a set of nodes, $E \subseteq V \times V$, is a set of directed branches, and w: $E \rightarrow R^+$ is a flow function and R^+ is the set of non-negative real numbers. The **throughflow** of a branch (x, y) in E is denoted by **w(x, y)**. For each branch (x, y) in E, x is an input of y and y is an output of x. For x in V, let I(x) denote the set of all inputs of x and O(x) be the set of all outputs of x. The inputs and outputs of a graph G are defined by I(G) = {x in V | I(x) is empty} and O(G) = {x in V | O(x) is empty}.

For every node x in G, **inflow(x)** is the sum of throughflows from all its input nodes, and **outflow(x)** is the sum of througflows from x to all its output nodes. The inflow and outflow of the whole flow graph can be defined similarly. It is assumed that for any node x in a flow graph, **inflow(x) = outflow(x) = throughflow(x)**. This is also true for the entire flow graph G.

Every branch (x, y) of a flow graph G is associated with the certainty (cer) and coverage (cov) factors defined as:

$cer(x, y) = \sigma(x, y) / \sigma(x)$ and

$cov(x, y) = \sigma(x, y) / \sigma(y)$

where $\sigma(x, y) = w(x, y) / w(G)$, $\sigma(x) = w(x) / w(G)$, and $\sigma(y) = w(y) / w(G)$ are normalized throughflows, which are also called strength of a branch or a node, and we have $\sigma(x) \neq 0$, $\sigma(y) \neq 0$, and $0 \leq \sigma(x, y) \leq 1$.

Properties of the coefficients were studied in [6, 15].

A directed path from x to y, $x \neq y$ in G is a sequence of nodes $x_1, \ldots, x_n$ such that $x_1 = x$, $x_n = y$ and (x_i, x_{i+1}) in E for every i, $1 \leq i \leq n-1$. A path from x to y is denoted by [x … y]

The certainty, coverage, and the strength of a path $[x_1 \ldots x_n]$ are defined respectively as:

$cer[x_1 \ldots x_n] = \prod cer(x_i, x_{i+1})$, for i=1, …, n-1,

$cov[x_1 \ldots x_n] = \prod cov(x_i, x_{i+1})$, for i=1, …, n-1, and

$\sigma[x \ldots y] = \sigma(x)cer[x \ldots y] = \sigma(y)cov[x \ldots y]$.

A connection from x to y, denoted by <x, y>, is the set of all paths from x to y ($x \neq y$). Definitions of the certainty, coverage, and strength of a connection <x, y> follow from the above definitions.

4.2. Example of Hierarchy of Flow Graphs

In the following, we will show the hierarchy of flow graphs corresponding to the subtree rooted at C_3. Suppose that we have 100 users accessed the lessons C_{31} and C_{32}, and all have taken assessment tests for C_{31}, C_{32} , and C_3 with results as Successful or Failed. In Figure 2, there are three groups 311, 312, and 313 for the less C_{31}. There are 40 sequences in group 311, 30 in group 312, and 30 in group 313. The picture shows the distribution of throughflows of branches in the graph. Similarly, there are 100 users have taken lesson C_{32}, they are divided into two groups 321 and 322. To compute Bayesian factors associated with the branches, we use the total throughflow of the graph, i.e., 200 as a denominator to normalize the throughflows of the branches. The numbers are shown in Table 2 and 3.

The structure of assessment node C_3 consists of the subgraph with Successful and Failed nodes from C_{31}, C_{32} , and C_3 and branches, which includes possible combinations of Successful and Failed nodes from C_{31} and C_{32}. In our example, there are four possible combinations. Together with the Successful and Failed nodes of C_3, we have eight entries in the assessment node C_3 as shown in Table 1. Note that normalization is based on throughflows of this subgraph only.

The above example shows how to combine decision values from children nodes. To determine the decision values of a path or a chain of nodes in the hierarchy of flow graphs. We use the definitions given in Section 4.1. For more useful properties of flow graphs, they have been studied in [6].

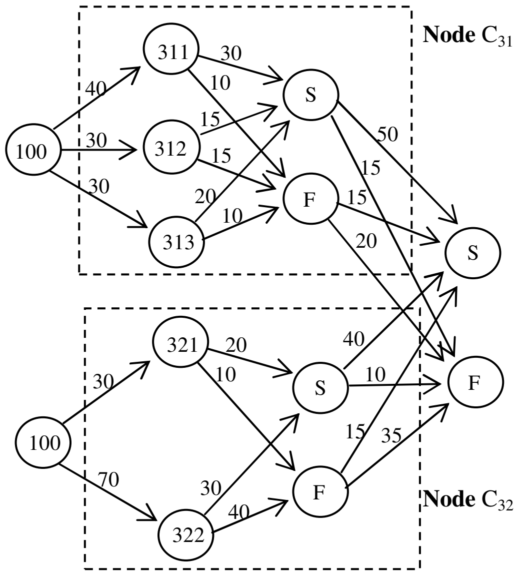

Figure 2. Flow graphs for usage nodes C_{31} and C_{32}.

5. Hierarchical Decision Tables

If prediction for performance of future sequences is desirable, then we can apply an inductive learning program such as BLEM2 [16], C4.5 [17] or Weka [18] to generate a classifier for each group of sequences produced by usage mining algorithms. These classifiers can provide more informative descriptions for sequences in a cluster, and some experiments showed that they have higher prediction accuracy than centroid-based methods [11, 12]. In addition, they can be used to provide guided help for users based on their usage patterns.

The number of branches in a flow graph, in general, is the number of possible combinations of nodes. One possible way to simplify the number of branches is to use an inductive learning program such as BLEM2 to generate a set of decision rules from a given flow graph. In this way, we may have a hierarchy of decision rules which provides a possible approximation to a given hierarchy of flow graphs.

6. Conclusions

We have presented a framework for using a hierarchy of flow graphs to represent usage patterns and assessment results of web-based e-learning systems. Known properties of flow graphs can be used to derive decision algorithms from such a hierarchy. Inductive learning programs can be used for further simplification of the hierarchy. Efficacy of the framework will be evaluated in the future using practical data sets.

7. References

[1] Wikipedia: http://en.wikipedia.org/wiki/E-learning .

[2] Woolley, D.R., “PLATO: the Emergence of Online Community,” http://thinkofit.com/plato/dwplato.htm . (1994).

[3] Barr, A. and E.A. Feigenbaum, The Handbook of Artificial Intelligence (Volume 2), William Kaufman, Los Altos, CA, 1981.

[4] Woolf, B., “AI in Education,” in *Encyclopedia of Artificial Intelligence*, Shapiro, S. ed., John Wiley & Sons, Inc., New York, pp. 434 – 444, (1992).

[5] Piramuthu, S., "Knowledge-based web-enabled agents and intelligent tutoring systems," *IEEE Transactions on Education*, Vol. 48, No. 4, 750 – 756, 2005.

[6] Pawlak, Z., “Flow graphs and intelligent data analysis,” *Fundamenta Informaticae* 64 (2005) 369 – 377.

[7] W. W. W. Consortium, "The common log file format. Available at: http://www.w3.org/Daemon/User/Config/Logging.html#common-logfile-format," 1995.

[8] Cooley, R., P.-N. Tan, and J. Srivastava, "Discovery of interesting usage patterns from Web data," presented at WEBKDD, 1999.

[9] Kohavi, R., "Mining e-commerce data: The good, the bad, and the ugly," presented at 7th ACM SIGKDD International Conference on Knowledge Discovery, San Francisco, California, 2001.

[10] Cooley, R., B. Mobasher, and J. Srivastava, "Data Preparation for Mining World Wide Web Browsing Patterns," *Knowledge and Information Systems*, vol. 1, pp. 5-32, 1999.

[11] Khasawneh, N. and C.-C. Chan, “Web Usage Mining using Rough Sets,”*Proc. NAFIPS 2005, Int. Conf. of the North American Fuzzy Information Processing Society,* June 22-25, 2005, Ann Arbor, Michigan, pp. 580-585. *ISBN 0-7803-9188-8 IEEE Catalog No. 05TH8815C*

[12] Khasawneh, N. “Toward Better Website Usage: Leveraging Data Mining Techniques and Rough Set Learning to Construct Better-To-Use Websites,” Ph.D. Thesis, *Department of Electrical and Computer Engineering, the University of Akron*, August, 2005.

[13] Berthold, M. and D. J. Hand, *Intelligent Data Analysis – An Introduction*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[14] Ford, L.R. and D.R. Fulkerson, *Flows in Networks*. Princeton University Press, Princeton, New Jersey, 1962.

[15] Pawlak, Z., “Flow graphs and decision algorithms,” in Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, Proceedings, G. Wang, Y. Yao, and A, Skowron, Eds. *Lecture Notes in Artificial Intelligence, 2639*, Springer, 2003, 1 – 10.

[16] Chan, C.-C. and S. Santhosh, “BLEM2: Learning Bayes’ rules from examples using rough sets,” *Proc. NAFIPS 2003, 22nd Int. Conf. of the North American Fuzzy Information Processing Society*, July 24 – 26, 2003, Chicago, Illinois, pp. 187-190.

[17] Quinlan, J.R., *C4.5: Programs for machine learning*. San Francisco, Morgan Kaufmann, 1993.

[18] Witten I.H. and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, San Francisco, Morgan Kaufmann, 2000.