Applying Web Usage Mining to an Educational Digital Library Service

Bart C. Palmer Mimi Recker
bart.palmer@gmail.com mimi.recker@usu.edu
Department of Instructional Technology & Learning Sciences
Utah State University, Logan, UT, USA

This case study in educational data mining (EDM) describes the motivation, methods, results, and best practices used while studying the usage data from users of a web-based tool, the **Instructional Architect** (IA.usu.edu). Our focus is twofold: (a) to highlight lessons learned for those new to data mining and (b) to introduce the analysis techniques of latent class modeling as an analysis tool. The context for this work is the field of educational digital libraries (EDL) and teacher use of Internet technologies (especially online learning resources) in the classroom.

Motivation

The IA is a simple EDL, end-user authoring service, developed with funding from the U.S. National Science Foundations' National Science Digital Library program (NSDL.org). In the IA, teachers can *collect*, *sequence*, *annotate*, and *share* online learning resource (from the NSDL or any web resource) with their students and other teachers (Recker, 2006). The dual purposes of the IA are: (a) to provide an authoring tool for teachers, and (b) to serve as a focal point for research concerning teacher adoption and use of technology.

To date, approaches to IA user studies have mainly consisted of traditional 'snapshot' methods of educational inquiry (e.g., interviews, surveys, observations, reflection papers) with limited statistical analyses of user behaviors as captured by the web server log and

the backend relational database (Khoo, Recker, Pagano, Palmer, Washington, & Donahue, 2008; Recker, Dorward, Dawson, Mao, Li, Palmer, et al., 2005). As may be typical, our questions have developed iteratively as we have explored teacher use of technology; the same iterative process has also shaped our methodology and use of data as our understanding and need for empirical evidence has increased.

This chapter reports on recent efforts to analyze the vast amount of longitudinal web usage data to better characterize IA users. In addition, as the IA collects both teacher and student web usage data, our analyses focus on understanding differences between and within these two groups. Finally, our analyses move beyond simply evaluating search engine effectiveness, typical of EDL research, to investigating how online resources are used once found.

The next section describes our methodology. We then describe the context for our case study, its data sources, and an initial application of the framework. We conclude with a discussion of the application of EDM process in order to inform the greater EDL community of the merits, caveats, and tools that can assist them as they seek to better understand their users.

Methodology

The methodological framework underlying our work is the process of Knowledge Discovery from Data/Databases (KDD, see Han & Kamber, 2006). This process entails data cleaning, selection, mining, and evaluation, and will be described next. Following that, we will introduce latent variable modeling, a psychometric analyses which enables the discernment and characterization of user behavior classes.

KDD

With the rise of the information age, algorithms, and methods for discovering patterns in large and complex data stores have been developed that make the extraction of *knowledge from data/databases* (KDD or knowledge discovery, see Vickery, 1997; Benoit, 2002; Trybula, 1997) more explicit and procedural. Since the 1980s, the KDD process, and its well-known step of *data mining* (DM; or automated and convenient analysis), have become popular and are responsible for enormous advances that have been made in many fields such as biology, chemistry, math, computer science, and business (Han & Kamber, 2006; Vickery, 1997; Witten & Frank, 2005; Ye, 2003).

While knowledge discovery from data is not a new concept, the last 20 years has seen a general set of procedures emerge and become relatively well defined (for a review, see Palmer, 2008; *[other chapters in this book]*). Many variations exist, according to the questions asked and the kinds of data available. The process of KDD entails the following four steps (Han & Kamber, 2006; Liao, 2003; Vickery, 1997).

1. Cleaning and integrating

Often KDD is applied to multiple and distributed data sources which were independently developed and maintained. This can lead to missing, incomplete, incompatible, or otherwise *noisy* data. Noise in data can pose a significant problem for conducting any meaningful analysis and therefore data must be cleaned and distributed sources integrated before further work may proceed. Reducing noise is one of the most important and time-consuming problems in KDD. The cleaning process, therefore, may include activities such as: filtering noise, filling gaps, correcting erroneous data, and matching date fields.

Web log data has particular characteristics that further complicate this step. For example, the common metric of *duration of visit* is necessarily estimated when there is only one page visited. Depending on the purpose for data mining, results may likely be more reliable and meaningful if activity from *bots* and other *bounce* (single request visit) data were removed from the dataset. Additional information on web data mining is covered in Zhou, Xu, Nesbit, and Winne (this book), some of which will be highlighted below.

The integration process brings all the data sources into one large, searchable, filterable, and consistent location. Data cubes, relational databases, and tables are popular ways to organize data with a multidimensional lookup and convenient summarization abilities.

2. Selecting and transforming

With so much data, it is important to remember that complex patterns are very hard to interpret in a meaningful way. Therefore it is desirable to have a parsimonious solution created from just a few select variables. Transformation can be used to aggregate and simplify, extract additional variables, or reshape the data in order to fit the assumptions of analysis.

3. Data mining

The actual DM analyses focus on identifying patterns, outliers, classes/clusters, or other models. Web log analyses can include text, count, click-streams, and many other kinds of data. Such data have typically involved traditional statistical methods (e.g., Recker & Palmer, 2006). However, because of inherent assumptions, these methods are less robust when used with large datasets, varied distributions, or for exploratory purposes.

Machine learning algorithms offer analyses that are capable of handling such irregular data. Within machine learning research, two main classes of methods are used for data mining (Chen & Chau, 2004; Hastie, Tibshirani, & Friedman, 2001). The first are called *supervised algorithms,* in which relationships in the data are mapped into known or named classes. *Unsupervised methods*, on the other hand, discover relationship patterns from the data without the help of class memberships and is often the method used for exploration of data.

4. Evaluating and presenting

The number of results or models from analyses will vary depending on the methods used. For example, with association rule mining (Webb, G. I., 2003), hundreds and thousands of rules may be found in the data. Therefore, evaluation is essential in cull rules that are not helpful, interesting, or have enough support or confidence. In addition, visualizations techniques are commonly used to help present findings.

Latent Variable Modeling

A common approach to identification and characterization of unknown groups in data is to aggregate to the user level and apply traditional cluster analysis to the cross-sectional data (e.g., Luan, 2004). Because traditional cluster techniques are not transparent (i.e., "black-box" analyses), they must be followed by a subsequent analysis (e.g., logistical regression) to characterize the groups (Magidson & Vermunt, 2002). Other drawbacks of traditional cluster analysis (as well as factor and principal component analyses) are that they are not as stable with missing data and their results are unable to be statistically compared (for additional description of these techniques and their limitations, see Apley,

2003; Magidson & Vermunt, 2002, 2004; Stevens, 2002). Similar difficulties exist with longitudinal analyses of detailed data.

Finite mixture modeling (also known as latent class modeling or analysis, LCA), and its longitudinal counterpart, growth mixture modeling (GMM, or latent class growth analysis) have begun to see some widespread use and extensions developed that can be used to classify and characterize profiles with latent variables in one step (Ip et al., 2003; Jung & Wickrama, 2007; Vermunt & Magidson, 2003). This modeling family of analyses uses an estimation algorithm (e.g., maximum likelihood) in determining the unconditional probability of class membership based upon the conditional probabilities of manifest variables. While assumptions remain on the analyses themselves and should be understood and considered when planning analyses and interpreting results (e.g., local solutions, see Bauer, 2007), newer extensions can handle various assumptions on the scale and shape of the data distribution. The output can include probabilistic group memberships, allowing for partial groupings and the ability to examine similarities and differences in group characteristics. Latent approaches (like traditional clustering) still require the researcher to select *k* classes for the analyses, however, empirical methods exist for comparison between two solutions (unlike traditional clustering).

Growth mixture models (GMM, and are the longitudinal version of finite mixture models) allow different latent classes to have different trajectories over time (Jung & Wickrama, 2007). These can quickly become very complex, but are helpful in understanding how user patterns change (Ip et al., 2003).

The IA Case Study

The Instructional Architect (http://IA.usu.edu) is a simple, Internet-based tool designed to help teachers find and use learning resources available on the Internet. It is especially designed to support teachers in finding high quality resources in the U.S. National Science Digital Library (NSDL.org), and elsewhere in the Web (often referred to as *learning objects*, see Prieto, Zapata, & Menéndez, this book). With the IA, teachers can discover, select, sequence, annotate, and reuse online learning resources in order to create Web pages containing instructional resources for their students, for example, lesson plans, study aids, homework. These are collectively called *IA projects* (Recker, 2006).

Specifically, teachers can use the IA in several ways. In the 'My Resources' area of the IA, teachers can directly search for and save STEM resources from the NSDL Data Repository (NDR). Teachers can also select any Web resource including interactive and Web 2.0 content (such as RSS feeds and podcasts), and add it to their list of saved resources. In the 'My Projects' area, teachers can design web pages in which they select a look and feel for their project, input selected online resources and provide accompanying text. Finally, teachers can 'Publish' their IA projects for only their students, or the wider web world.

Figure 1 shows an example of a simple, teacher-created IA project: the background shows teacher content and instructions, while the foreground shows an online learning resource (in this case, a simulation of weather).

The IA has been collecting and analyzing increasingly complex webmetrics data since 2002 (Khoo et al., 2008; Recker and Palmer, 2006). Since early 2006, it has been

engineered to collect data with Google Analytics. Current webmetrics analyses show that the IA's usage continues to grow: from 2002 to March 2009, over 3,700 users registered, using over 32,000 online resources. From August 2006 to March 2009, the now 7,277 IA projects created by teachers have been viewed nearly 500,000 times.

Questions

With so much data available it is tempting to begin exploring every possible variable for significance and meaning. However, in order to not let the "data-tail" wag the "research-dog" it is critical to have focused, simple questions that can be answered and built upon in an iterative process. As mention above, we have some a priori knowledge of how some users are behaving on the IA site which helps us to know what data will be useful in discerning (at least) those perceived differences.

Our research objective was to take a longitudinal look at what we term *meaningful user activity*, meaning those actions by users that support the purposes and intentions of the site design. Meaningful IA activity includes things such as persistence and frequency of return visits, source of online resources (NSDL, IA, or Web), quantity of content created and frequency of project changes, and use of projects and resources. These thus bound the study for our overarching question: With respect to meaningful IA activity, do subgroups of users exist? If they do, what are their characteristics?

Instructional Architect Data Sources

In the case of the IA, the large and complex data being collected comes in three forms: 1) the IA relational database (IARD, implemented as a PostgreSQL database), 2) the IA web server log (WSL), and 3) Google Analytics (GA).

The IARD is the most granular, storing user profile data (captured during registration), as well as page-tracking variables on a per-user basis. Early in 2008 we implemented a database table for tracking page hits with associated user session IDs, in order to simplify the reconstruction of user sessions. Each entry consists of essentially the same information as Apache's *combined* log format with some exclusions and additions. We did not capture the HTTP username or password, server status, or size of the resulting page. The additions were the PHP session id, user id (both as registered user and students logging in using a teachers passphrase), and a comment on any user actions.

The WSL stores standard Apache combined web server log data, including IP address, timestamp, referrer, and requested URL. Third, GA stores similar data to the IARD and WSL using *page tagging*, but is not identifiable by IP address or user. GA collects a variety of longitudinal data including location, visit (total, unique, returning), bounce-rate, exit page, goal conversion, visit referrer, session length (time), time to next session, and number of page views. Each GA metric is available at various levels of granularity. Table 1 shows a summary of web usage data, its source, and type when aggregated at the user and session levels.

It is reasonable to expect that other Web sites contemplating EDM can use this kind of information, as they can either: (a) implement the same kind of database insert, or (b) configure their server to log session IDs (this can be done natively or as an add-on like the Apache module *mod-session*; http://httpd.apache.org/docs/trunk/mod/mod_session.html). They can also adopt third-party trackers like GA to provide good but less flexible data and analyses.

Application

This section describes the application of the 4-step KDD process to the IARD. An important part of this study was to document the process, the difficult lessons learned, decisions made along the way, and the study's limitations in order to inform the educational digital library community of the effort involved in EDM.

As the KDD process unfolded, it became clear that it is not linear; instead, like other design activities, it is iterative. We have identified points that were pivotal to in furthering our understanding and will be useful to other researchers contemplating data mining.

1. Cleaning and Integrating

As described by Zhou, Xu, Nesbit, and Winne (this book), there are important caveats to analyzing web usage data. In our case, two particular thorny problems were encountered: (a) spam noise, and (b) session ID reuse.

Noise. Noise is data that has little or no meaning for the current task. Because of its open Web 2.0 authoring capabilities, spammers abused the IA. While the initial problem was dealt with, incoming requests still mimic a legitimate IA GET request URL, but are meant to exploit or break the web application, and therefore may also break the cleaning code if processed. These noisy spam requests can be filtered through regular expression matching on specific known URL patters, thus avoiding them altogether.

Session ID reuse. When we initially began recording the PHP session ID, we assumed that all we would have to do is group by that column and we would have valid session information. However, since PHP uses a pseudo-random session ID generation algorithm,

these IDs do get reused on occasion. Fortunately, we did not experience this reuse at the same time with two users (i.e., session collision), but it did add one more step to cleaning the data by finding session breaks by comparing the request timestamp. Any difference of more than 30 minutes was considered a new session, and the *php_session_id* column was updated accordingly.

The integration of our data with user registration information was not difficult since we had already linked the user identification within the data generation and could match on unique IDs. As a note, this kind of database logging can also create a slow loading times for the end user when tracking scripts take too long. For IA users, this was becoming a problem on our older, slower server. However, we have since upgraded our server, and see no further drawback to tracking user activity this way. On the other hand, if we were working with only the web server logs instead of the database, we would have much more trouble determining particular user activity. As such there are trade-offs between data granularity and processing speed of standard web server logs, as opposed to custom logging.

2. Selecting and Transforming

Because of the vast amount of data collected, not all are useful for determining characteristics of interest. Therefore, a selection of relevant proxy or manifest variables were selected or created from the clean raw data. The data described in Table 2 was parsed and interpreted in order to produce the data in Table 1, which are to be used as proxy or manifest variables for the meaningful user activity class analysis. These new variables were then stored in new database tables for further selection and transformation. We have chosen to transform the detailed user actions into aggregates at the session or

visit level for each user, and again at the whole user level for different views and insights. Because of the amount of noise in the data, the process was particularly time consuming.

Extracting information from URLs. In order to extract and identify meaningful user activities, we utilized the referrer, request URL, and comment columns (see Table 2). Initially, we began with a PHP script that would clean while simultaneously parsing and interpreting the URL as an indication of user activity. However, given the size of the dataset (870, 443 entries and over 250, 650 sessions for 2008) and the complexity of the data extraction (string processing), looping over these entries in PHP would take a very long time. Therefore, we learned that becoming expert in the use of the appropriate database query language was essential.

Additionally, when we copied the data to the analysis database, we neglected to notice the lack of proper indexing on the tables. This wasted a lot of time as we assumed the long completion times were a result of the size and complexity of our operations. For example, the creation of just one index changed an estimated 11 hours with the PHP scripts to a mere 141 seconds with proper indexes. Perhaps the PHP scripting option would have worked after this improvement after all. On the other hand, keeping data manipulation in the database server is highly desirable—and can also lend to future automation through triggers and stored procedures, keeping new data ready for analysis.

Once the data was cleaned, extracted, and transformed, selection of the actual data for analyses was simply a database query of the baseline, and two months of session information for our users.

3. Data Mining

In order to apply growth mixture modeling to the data in Table 1, appropriate longitudinal manifest variables must be assigned. We are using the user activity of a sequence of sessions as the unit of analysis as opposed to temporally spaced observations (see the top row of boxes in Figure 2). User registration (i.e., how they became familiar with the IA) and user aggregated data (e.g., total number of projects created, date registered) are included as covariates. The latent classes (C) are then calculated, each with an individual slope (S) and intercept (I). A second approach may also be taken where user baseline data are analyzed into latent classes which can then be fed into a second round of longitudinal analyses as described above.

4. Evaluation and Presentation

Analyses of data are currently ongoing. In the fourth step, it is critical to address the presentation of inferred classes in understandable and interpretable ways. Fruitful approaches include charts of trajectories and scatter plots showing "factor" or variable loadings for each class.

As mentioned above, the researcher still must choose a specific number of classes in which to divide the data and a resulting models must be evaluated for fit. One convenient aspect of GMM is that this unsupervised-like mining algorithm uses modeling which can then be compared (as opposed to other unsupervised approaches). This statistical comparison can be accomplished by using one of a number of ways, depending on the exact model and data (e.g., the loglikelihood statistic, L^2; Bayesian information criterion, BIC; or Akaike information criterion, AIC; for more see IP et al., 2003). In addition,

visual analysis of trajectories and graphs can provide further insight and ability to communicate the results.

Conclusion

We have introduced the 4-step process known as knowledge discovery with data/databases (KDD) applied to EDM, and briefly described the possibility of latent variable modeling as a useful data mining analysis. The Instructional Architect, a web-based digital library web service and resource for teachers and students, was presented as a case study of data mining. In this context, EDM was viewed as the search for hidden groupings of user behavior that can be characterized and interpreted.

The KDD framework was used to process raw web log data with user information to produce a set of variables that could be analyzed for latent classes. While proceeding thought the KDD process, several lessons were learned and surprises encountered. In particular, we found that preprocessing web data is a complex, difficult, iterative, and time-consuming task. We also found that storing processed data in a secondary database is a useful step for later analyses.

Perhaps the most important lesson learned about data mining is that selecting appropriately focused research questions will help the data miner answer salient questions in an organized manner. In the case of the IA, while there are a multitude of questions that could be asked, we chose to focus on user activity investigation as the search for hiding classes. Future research can include textual mining of the project content to determine types of projects and their relationships with certain types of resources.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grants No. 840745 & 0434892, and Utah State University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We thank the members of the IA research group and the Center for Methodological & Data Sciences (OMDS) @ USU for their advice.

References

Benoit, G. (2002). Data mining. Annual Review of Information Science and Technology (ARIST), 36, 265-310.

Chen, H., & Chau, M. (2004, 01 January 2004). Web mining: Machine learning for web applications. *Annual Review of Information Science and Technology (ARIST), 38*, 289-329.

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufmann.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

Ip, E., Cadez, I., & Smyth, P. (2003). Psychometric methods of latent variable modeling. In N. Ye (Ed.), *The handbook of data mining* (pp. 215–246). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Khoo, M., Recker, M., Pagano, J., Palmer, B., Washington, A., & Donahue, R. A. (2008, June 16-20, 2008). Using web metrics to analyze digital libraries. Pittsburg, PA.

Liao, S. hsien. (2003). Knowledge management technologies and applications–literature review from 1995 to 2002. *Expert Systems with Applications, 25*, 155-164.

Magidson, J., & Vermunt, J. K. (2002). Latent class models for clustering: a comparison with k-means. Canadian *Journal of Marketing Research, 20*, 36-43.

Muthén, B. (2004). Latent variable analysis: Growth mixture modeling and related techniques for longitudinal data. In D. Kaplan (Ed.), (pp. 345–368). Thousand Oaks, CA: Sage Publications, Inc.

Recker, M. (2006, September 2006). Perspectives on teachers as digital library users: Consumers, contributors, and designers. *D-Lib Magazine, 12* (9).

Recker, M., Dorward, J., Dawson, D., Mao, X., Liu, Y., Palmer, B., et al. (2005). Teaching, designing, and sharing: A context for learning objects. Interdisciplinary Journal of Knowledge and Learning Objects, 1, 197-216.

Recker, M., & Palmer, B. (2006). Using resources across educational digital libraries. In 6th acm/ieee-cs joint conference on digital libraries (p. 240-241). Chapel Hill, NC: ACM.

Palmer, Bart C. (2008). *Web Usage Mining: Application to an Online Educational Digital Library Service*. Unpublished doctoral dissertation proposal. Utah State University.

Trybula, W. J. (1997, 01 January 1997). Data mining and knowledge discovery. *Annual Review of Information Science and Technology (ARIST), 32*, 197-229.

Vickery, B. (1997). Knowledge discovery from databases: an introductory review. *Journal of Documentation, 53* (2), 107-122.

Webb, G. I. (2003). Association rules. In N. Ye (Ed.), *The handbook of data mining* (pp. 25–39). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Witten, I., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Ye, N. (2003). Preface. In N. Ye (Ed.), *The handbook of data mining* (p. xix-xxii). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Table 1. A sample of IA metrics, their type, and data source (IA relational database, the IA web server log, and Google Analytics).

Metric	Type	Source
User registration data	Cross-sectional	IARD
Login activity	Cross-sectional	IARD
# DL resources collected	Cross-sectional	IARD
# IA project created	Longitudinal	IARD
# IA projects	Cross-sectional	IARD
Resource Source (NSDL or Web)	Cross-sectional	IARD
Resource Metadata	Cross-sectional	IARD
Referrer	Longitudinal	IAWSL, IARD
Time since last session	Longitudinal	IAWSL, IARD
Session referrer	Longitudinal	ALL
Session info (length, # projects created)	Longitudinal	ALL
View (time, IP, Platform)	Longitudinal	ALL
Location	Longitudinal	GA
Visits (total, unique, returning)	Longitudinal	GA, IARD
Bounce-rate	Longitudinal	ALL
# page views	Longitudinal	ALL
Session length	Longitudinal	GA, IARD

Cross-sectional is aggregated at the user level
Longitudinal data are available for each session

Table 2. The database columns used to track raw user activity.

Column	Description	Type
table_id	Unique identifier for the table	integer
php_session_id*	The PHP session identifier (timeout after 30 minutes)	text
user_id	If logged in, this is the unique identifier to the user table	integer
group_id	If set, this is the unique identifier to the groups table	integer
target_id*	The complete requested URL	text
referrer*	If set, this is the referring page – set by Apache	text
time_viewed*	The timestamp of the insertion	timestamp
platform*	The user-agent value – set by Apache	text
Comment	A note about the user activity and the result of requests	text

* available (or analogous to what is available) through the Apache logs with mod-session module

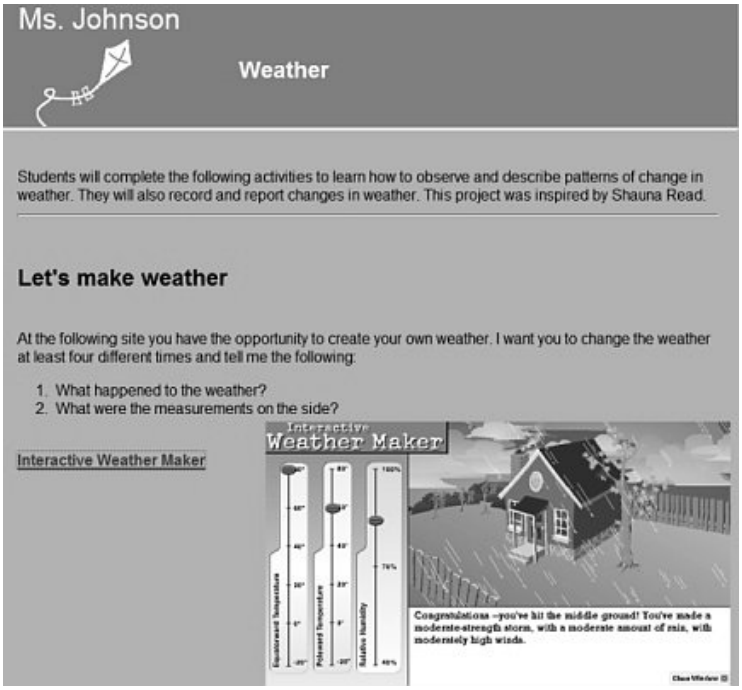

Figure 1. An example of an Instructional Architect project with an overlay of the online resource linked to from the project.

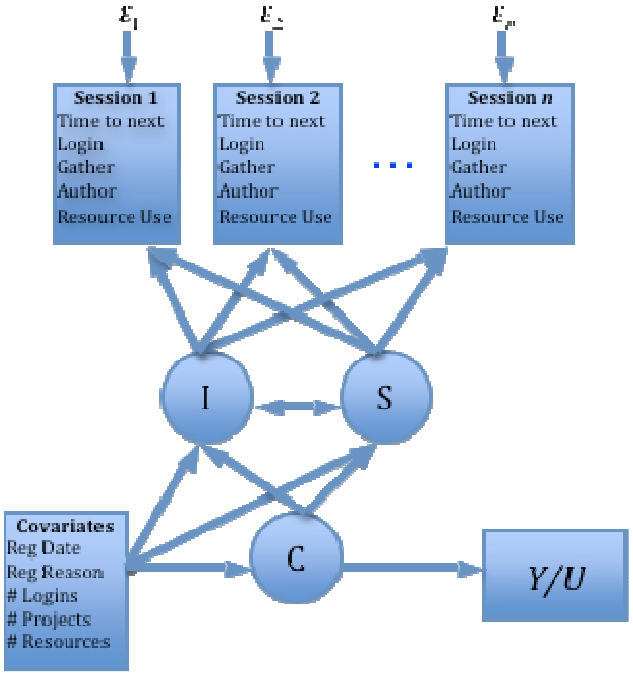

Figure 2. Growth mixture model for clustering IA users from aggregated session data.