ISSN 2076-2429 (print)
ISSN 2223-3814 (on line)

UDC 004.9

Sebastian Schwarzrock, B.Sc., Beuth University for Applied Sciences, Berlin, Germany

ANALYSIS OF LEARNER NAVIGATION ON WEB-BASED PLATFORMS USING ALGORITHMS FOR SEQUENTIAL PATTERN MINING

С. Шварцрок. **Аналіз навігації учня по веб-платформам з використанням алгоритмів отримання моделей послідовності.** Комплексні інтерактивні системи, такі як системи управління навчанням, збирають значну кількість даних, що описують поведінку користувачів. Ці дані можуть бути використані в якості основи для величезної кількості досліджень. Крім того, існує гостра необхідність у таких дослідженнях для поліпшення структур цих систем і виявлення їх недоліків. Особливо цікавою областю для досліджень є пошук частотних навігаційних моделей в призначених для користувача даних, записаних в лог-файлах або базах даних. У статті розглядається веб-застосунок для аналітики процесу навчання, в якому автори використовують різні алгоритми отримання моделей послідовності.

Ключові слова: аналітика процесу навчання, отримання моделі послідовності.

С. Шварцрок. **Анализ навигации учащегося по веб-платформам с использованием алгоритмов получения моделей последовательности.** Комплексные интерактивные системы, такие как системы управления обучением, собирают огромное количество данных, описывающих поведение пользователей. Эти данные могут быть использованы в качестве основы для огромного количества исследований. Кроме того, существует острая необходимость в таких исследованиях для улучшения структур этих систем и выявления их недостатков. Особенно интересной областью для исследований является поиск частотных навигационных моделей в пользовательских данных, записанных в лог-файлах или базах данных. В статье рассматривается веб-приложение для аналитики процесса обучения, в котором авторы используют различные алгоритмы получения моделей последовательности.

Ключевые слова: аналитика процесса обучения, получение модели последовательности.

S. Schwarzrock. **Analysis of learner navigation on web-based platforms using algorithms for sequential pattern mining.** Complex interactive systems like learning management systems gather huge amounts of data describing the users' behavior. This data can be used as a basis for a vast number of analyses. Furthermore there is a strong need for these analyses to enhance the structure of the systems and to identify their shortcomings. A particularly interesting field is the search for frequent navigational patterns within the user data, recorded in log files or databases. This paper deals with a web-based application for learning analytics, in which we use different sequential pattern mining algorithms.

Keywords: learning analytics, sequential pattern mining.

Introduction. With the increasing usage of learning management systems and the data stored within the according systems, the need for analysis is growing. As the Horizon Report 2012 [4] indicates, learning analytics is one of the most important emerging fields of research in the educational context. Within the LeMo project we are developing an application that offers the users different analyses to get an overview of the usage of content within the Learning Management Systems. The software shall enable administrators and teachers to monitor the usage of the learning materials provided in the system. One of the main aspects of the development is to offer the same analyses for heterogeneous platforms. A large part of the functionality demanded by our projects partners dealt with the analysis of the user navigation within the learning management systems. A lecturer might want to know if the learning objects provided in a course are accessed in the intended

order, or if some objects are left out by many users. A detailed knowledge of the users' navigational behavior can give information of the way users interact with the system's content and help to identify shortcomings within the content's structure. Caused by the large amount of data that has to be processed to determine reoccurring navigational patterns, efficient algorithms had to be found to provide the information.

1. Extracting user paths from interactive systems. In the LeMo project, we differentiate between two types of platforms: personalizing and non-personalizing. When using a personalizing platform, the user has to be registered within the system in order to access the learning materials. All interactions with the system are stored in a database and every one of them can be assigned to a single user. An example for this kind of system is the learning management system Moodle [5].

Non-personalizing platforms do not restrict the access to their content. Common examples for non-personalizing platforms are online encyclopedias like Chemgapedia [6]. The main source for usage data within these systems are the server log files.

Therefore different approaches to data extraction are needed for both types of systems and different problems need to be addressed. We have already implemented data extractors for three different platforms: The learning management systems Moodle and Clix [7] and the online-encyclopedia Chemgapedia. While importing data from personalizing systems basically means to copy values from one database to another, gathering data from server log files is more complex. While there are more detailed information on the content and the users on personalized systems, server log files just offer data about accesses.

1.1 Preprocessing of platform data. Reviewing the results of the first brief usage statistics of the non-personalizing platform Chemgapedia, it became apparent that the server log data has to be preprocessed in order to exclude traffic data produced by web crawlers and web robots. An analysis of the log data suggested that approximately half of the accesses that have been documented in the server logs were not performed by human users. To solve this problem we added an optional functionality that excludes user sessions showing suspicious behavior. We implemented filters for the data extraction process that identifies and ignores user sessions with one of the following characteristics [8]: multiple accesses per second, reoccurring time intervals between accesses and accessing the same page more than 10 times in one session. All sessions starting with an access to the web-site's robot.txt were excluded beforehand. Using the filters we were able to reduce the number of log entries by 47 % while excluding only 5 % of the users. As indicated, the use of the filters is optional and we plan to improve the functionality for data that has been derived from non-personalizing platforms.

2. Identification of frequent user paths. User paths are already used by our application to create an overview of the navigational behavior within a course and for the search for frequent paths. We will describe the latter in detail. The basic idea was to determine whether reoccurring sequences can be found in the user paths of different users. Reviewing the most frequent sub-paths gives the lecturer the opportunity to find out, whether the learning objects have been accessed in the intended order by the users or not. Since the access of the lecturers is limited to data of their own courses the analyzed user paths only consist of the learning objects the courses contain. Furthermore there is the possibility to limit the length of the paths to accesses made within a specified time interval.

First, we have to define, how a frequent path is defined. We are using the words path and sequence synonymously. As describe in [2] a path $S_a=a_1, a_2, a_3,\ldots,a_n$ is contained in $S_b=b_1, b_2, b_3,\ldots, b_m$, if there exist integers $1 \leq i_1 < i_2 < \ldots < i_n \leq m$ such that $a_1=b_{i1}$, $a_2=b_{i2}$, …, $a_n = b_{in}$. We call S_a a sub-path of S_b. The number of different user paths containing the sequence is called absolute support. The relative support is the percentage of user paths containing the sequence. A sub-path or sequence is called frequent, when its relative support exceeds a given minimum support threshold.

The algorithm we first used to identify frequent sequences in the usage data is called BIDE [2]. It returns all frequent closed sequences (or sub-paths) for a given set of sequences.

ISSN 2076-2429 (print)
ISSN 2223-3814 (on line)

Table 1

Example (BIDE-algorithm): The sequence A,B,C is contained in three of four user paths and has therefore an absolute support of 3 and a relative support of 0,75

User	User path
1	A, B, C
2	B, A, B, D, D, C, E
3	A, B, D, D, E, C, C
4	D, E, E, B

One of our projects partners pointed out, that while the results of the BIDE-algorithm offer some information on the users' behavior, they were more interested in frequent paths were all items have been accessed consecutively by the users. Since the BIDE-algorithm doesn't allow other constraints than the minimum support we were forced to find an algorithm allowing the definition of a maximum gap between the items of a frequent sequence. So we added an analysis that uses the Fournier-Viger-algorithm [3], which is a version of the algorithm proposed by Hirate and Yamana [9]. Using the maximum interval constraint, we were able to match our partner's needs. The algorithm also provides a constraint for a minimum and a maximum time interval between the first and the last item of a sequence.

Table 2

Example (Fournier-Viger-algorithm): Using the Fournier-Viger-algorithm with a maximum item gap of 1, the only sequence containing more than one item, exceeding a minimum support of 0,6, is: A, B (minimum support: 0,75).

User	User path
1	A, B, C
2	B, A, B, D, D, C, E
3	A, B, D, D, E, C, C
4	D, E, E, B

2.1 Performance concerns. One of our major concerns is the processing time of the analyses provided in the application due to usability considerations. While running the first tests using the data provided by our partners it became apparent that the time needed for the calculation of frequent sub-paths differs strongly depending on the characteristics of the input data and the value for minimum support. Therefore we tried to find a way to determine a "default" value for the minimum support variable, using known parameters of the course. We identified the following parameters that can be determined without running the algorithm: Number of user paths, average length of user paths and the amount of learning objects contained within the course. The tests were run using generated user data and showed that the length of user paths has a big impact on processing time but the most important influence on the algorithm's processing time is the number of frequent sequences contained in the data. Since BIDE uses the Apriori [10] approach to determine reoccurring sub-paths, the processing time increases dramatically when the number of frequent sequences is large. Thus only few assumptions can be made for the processing time of the algorithm and the time needed for mining different courses with identical parameters can differ greatly according to their number of contained frequent sequences. The Fournier-Viger-algorithm was working faster when the minimum and maximum item gap constraints were set, due to the smaller amount of candidate sets generated for each step.

2.2 Interpretation of frequent paths. As described above, frequent paths from both algorithms have to be interpreted differently. While the BIDE-algorithm is more likely to find results, using a high minimum support, the resulting paths should be interpreted cautiously because it allows gaps between adjacent items of a path. A frequent path, identified by the BIDE-algorithm may indicate that

contents of a learning management system have been accessed by the users in the order intended by the lecturer or course designer, but it does not show the users' navigation between each item of the path. Two user paths that are alike to the BIDE-algorithm can differ greatly. As pointed out by Hirate and Yamana [9] it also does not take into account large time intervals between two accesses within a frequent path. Using the Fournier-Viger-algorithm, all paths that feature large time or item intervals between adjacent items of a frequent path can be excluded.

Another aspect is the amount of data that is returned, especially by the BIDE-algorithm. One data set, that was generated for the performance tests mentioned above, consisting of 25 users paths, 15 learning objects and an average user path length of 60 contained 900000 frequent sub paths, each exceeding the minimum support of 0.4. This amount of paths can by no means be presented properly within a web-application and is of little use for the lecturer itself.

Conclusion. The detection of frequent user paths within the contents of a learning management system can offer an interesting view on the users' navigational behavior, when interpreted correctly. The algorithms mentioned are useful tools for the calculation of frequent paths, though there have to be further considerations concerning performance aspects to guarantee a high level of usability.

Acknowledgment. The “European Regional Development Fund Berlin” and the “Institute für Angewandte Forschung IFAF” support this work. I would also like to thank all members of the LeMo project for the given advice and support.

References

1. Beuster, L. LeMo-Lernprozessmonitoring auf personalisierenden und nicht personalisierenden Lernplattformen / [L. Beuster, M. Elkina, A. Fortenbacher and others] // Grundlagen Multimedialen Lehrens und Lernen GML2, 2012, Waxmann Verlag. — pp. 63 — 77.
2. Wang, J. BIDE: Efficient mining of Frequent Closed Sequences / J. Wang, J. Han // 20th International Conference on Data Engineering, 2004, Proceedings. — 2004.
3. Fournier-Viger, P. A Knowledge Discovery Framework for Learning Task Models from User Interactions in Intelligent Tutoring Systems / P. Fournier-Viger // 7th Mexican International Conference on Artificial Intelligence. — Atizapán de Zaragoza, Mexico. — 2008.
4. Johnson, L. The NMC Horizon Report: 2012 Higher Education Edition / L. Johnson, S. Adams, M. Cummins // Austin, Texas: The New Media Consortium. — 2012.
5. Moodle [Electronic resource]. — Access: https://moodle.org/. — 10.12.12.
6. Chemgapedia [Electronic resource]. — Access: http://www.chemgapedia.de. — 17.12.12.
7. Clix [Electronic resource]. — Access: http://www.im-c.de. — 01.11.12.
8. Tan, P. Discovery of Web Robot Sessions Based on their Navigational Patterns / P. Tan, V. Kumar // Data Mining and Knowledge Discovery. — 2002. — Vol. 6, Issue 1.
9. Harate, Y. Generalized Sequential Pattern Mining with Item Intervals / Y. Harate, H. Yamana // Journal of Computers. — Vol. 1, No 3. — 2006. — pp. 51 — 60.
10. Agrawal, R. Fast algorithms for mining association rules / R. Agrawal, R. Srikant // In VLDB’94. — Santiago, Chile. — 1994.

Reviewer Ph. D., Prof. Odessa nat. polytechnic univ. Brovkov V.G.

Received December 18, 2012.