Identifying Patterns in Learner's Behavior Using Markov Chains and N-gram Models

PAVEL ČECH
Faculty of Informatics and Management
University of Hradec Kralove
Rokitanského 62, 500 03 Hradec Králové
CZECH REPUBLIC
pavel.cech@uhk.cz

Abstract: - The paper aims at the utilization of Markov chains and n-gram models in order to determine common patterns in the learner's behavior. The patterns represent sequences of units of learning that are visited by various users. The patterns would be used to recommend next units of learning with regard to previously visited units of learning. In this way a novice student might easily navigate through a complex structure of units of learning in a specific domain of interest without prior knowledge of the structure and links between topics.

Key-Words: - Learners behavior, Markov chain, n-gram model, Content adaption, Pattern identification

1 Introduction

Recent advances in computer technologies directed the attention to personalization of products and services in many areas of everyday life. The education is not an exception. Currently the trend is to strive for more efficient learning by tailoring the learning experience based on the learner's needs. At the heart of these efforts there is a trend to content adaption or content sequencing in virtual learning environments.

Our approach is based on the identification of common patterns in the sequences in which the units of learning are being navigated by experts or other users. The purpose is to help novice learners to orient in a particular domain of interest without any prior knowledge of the structure and links among various topics and thereby avoid so called cold start problem [2].

The identification of patterns will be first formulated using Markov chain model [9] in which units of learning would represent states and navigation to another unit of learning is the transitions between states. Formulating the problem as Markov chain would enable to answer simple questions like: What is the most probable unit of learning visited in the next step? or What is the probability of visiting a certain unit of learning in n steps?

Using the Markov chain only successive states can be modeled so the patterns would be consists of only two states. In order to identify to find longer sequences the patterns will be modeled as n-grams [8] and the process of their finding elaborated.

2 Related work

There exist various approaches and technologies that are concerned with adaptation of virtual learning environments in general and with content adaptation in particular. Hence, approaches such as data mining [10, 19] multi-agent cooperation [6, 15, 20] knowledge engineering, ontology modeling and reasoning [5, 11, 16] or collaborative filtering [2, 3, 17] are widely used.

There also particular approaches to content sequencing. For example the decision tree techniques were used by Wang [19] to extract automatically the optimal learning sequences of teaching content, as indicated by students' diverse characteristics, and performances. Hsieh, T.-C., & Wang [10] applies Apriori algorithm and Formal Concept Analysis to build a concept lattice, using keywords extracted from some selected documents. The learning path is then created based on the mutual relationships among documents and both the preference-based and the correlation-based algorithms for recommending the most suitable learning objects.

The problem of identifying frequent patterns or sequences of a certain phenomena in order to predict a future event or determine a certain behavior is fairly common in many areas.

ISBN: 978-1-61804-056-5

In the field of data mining the approach to finding sequential patterns in huge datasets is called sequential pattern mining [1, 7]. The sequential pattern corresponds to the n-gram model. The sequential pattern mining aims at discovering frequent itemsets and determine association rules.

Predicting the future page to be visited is explored also in the clickstream data analysis. In the clickstream analysis the sequence of pages being visited is extended with other characteristics such as the amount of time spent on a given page or the page exposure [13].

Further to the above approaches two broad categories called content filtering and collaborative filtering can be discerned in order to adapt the content. The former is based on observing the user behavior and thus determining the user preferences in order to find similar content. The later attempts to find similarities among users, so that to provide them with similar content. In our paper the focus is on the later approach so that the cold start problem is avoided or mitigated.

In our previous study we employed knowledge engineering and ontological modeling to recommend particular topics and resources that might be of interest to the learner while completing a given task. The approach was based on the specification of explicit and inferring the implicit relations among topics (or units of learning) by the teacher or by a domain expert. The major problem in that approach was the necessity to specify the relations in advance while avoiding contradictory assertion by different experts. The automatic identification of patterns in navigation in a given domain would solve the problem.

The automatic identification of patterns using the n-gram model and Markov chain formalism is frequently utilized in the field of natural language processing. The n-gram can be composed of words or letters and can model the statistical distribution in a given language. The nature of the navigation and the sequences of states are in a sense similar, apart from the fact that in the navigation among units of learning there would not be repetitions as common as repetitions of words and letters in a language.

With regards to the approaches used in the data mining it can be noted that although the logic and also objectives are similar, however, in data mining the perspectives is from the point of view of finding rules that can be used to predict another item in a sequence. In our approach the stress is not on finding rules but the particular structures (patterns) and the probability would be used to predict a given occurrence of such a pattern.

3 Pattern identification

The patterns in navigation represent units of learning that are, by a group of learners visited, in sequence. The simplest pattern consists of two units of learning and specifies that there is a high probability that one unit of learning would be visited after the other. Hence, the sequence could be modeled as Markov chain and also as n-gram of units of learning.

The formal representation of the problem is as follows. Visiting a unit of learning in a time t is considered as being in a state s_t. Being in a s_t is a random variable. S represents the sequence of states in an order given by t. Let us assume that the observations of states are independently identically distributed. Thus, the sequence of states $S = (s_1, s_2, \ldots s_x)$ can be characterized as the joint probability distribution between all its states (random variables).

The probability p_{ij} is the probability of transition between states s_i and s_j (i.e. that the learner moves from state s_i to the state s_j). If the state s_j should be read from state s_i in time t the then the probability is computed as follows:

$$p_{ij}(t) = P(S_{t+1}=j|S_t=i) \quad (1)$$

The pattern consisting of n units of learning will be represented as n-gram model that corresponds to a Markov chain of order n-1 [8]. The probability of a certain pattern can be decomposed according to Bayes' rule into a product of conditional probabilities. Hence, the n-gram model predicts the state s_i based on $s_{i-(n-1)}, \ldots, s_{i-1}$. Expressed in probability terms $P(s_i| s_{i-(n-1)}, \ldots, s_{i-1})$.

There are several algorithms that can be used to mine for the frequent sequences (patterns, n-grams). A classical one is the Apriori algorithm [4, 10, 12, 18]. The Apriori algorithm is based on an anti-monotone heuristic: *if any length k pattern is not frequent in the database, its length (k + 1) super-pattern can never be frequent* [14]. Hence, the essential idea of the algorithm is to get k-length pattern that are frequent and then generate a set of candidate patterns of length $k + 1$ and test their frequency of occurrence. The pattern is extended always with frequent patterns tested in the previous step. Since at the initial step the $k = 1$ only items that are frequent on its own are considered in further phases. Similarly, the pattern $P'_{(i)} = \{1,5,7\}$ can be frequent only if patterns $\{1,5\},\{5,7\},\{1,7\}$ were frequent.

ISBN: 978-1-61804-056-5

The approach taken in this paper is a adapted and simplified version of an algorithm presented in [1]. Due to Markov property in which the probabilistic behavior of a Markov chain is determined just by the dependencies between successive random variables, the iterative and incremental pattern finding approach was selected. As opposed to Apriori algorithm the approach considers all items (states – to be accurate all seen states) when patterns are extended. The idea is that even thought the state or the pattern of states might not be frequent one, it still can be a good predictor for a next state given the relative frequency of transition is high enough.

The core of the approach consists of initial bi-grams (or pattern with $n = 2$) determination followed by iterative search for patterns of size $n+1$.

Assuming the data from navigation sessions by experts and advanced users are observed and available. During each session the visits of units of learning of a particular user would be recorded as one sequence of states.

3.1 The algorithm description

The problem of finding patterns is then split into the following steps.

Step 1: In the initial step the transitions between all successive states will be computed. Assuming there will be a relatively low number of states – approximately around 50 – the transitions can be stored in the probability transition table. In the initial phase the row and columns consists of all possible states so the matrix is $S' \times S'$. Note that in case of higher or unknown number of states or low number of data from navigation by expert users the matrix might be too sparse and thus other data structures would have be more efficient.

Once the initial step is completed the probability transition table can be used to answer question regarding the probability to reach a state in a specified number of steps given the learner is at a current state.

Step 2: Next, based on the computed transitions the most frequent patterns (bi-grams) would be determined. After the initial phase the probability transition matrix would contain all possible transition even those with zero occurrence i.e. those "unseen" ones. However, only patterns (bi-grams) that have the relative frequency higher than a specified threshold or in other terms with minimal support would be selected. The threshold or the minimal support is computed as the relative frequency of the occurrence of a given pattern. Thus the set of patterns P' with patterns consisting of two states would be created.

Step 3: In the next step the set of patterns P' would be used to create the matrix consisting of patterns and state i.e. $P' \times S'$. Next the navigation data would be iterated and the transitions from identified patterns to next state will be computed.

In order to consider also states that precede a certain pattern computing the transitions for matrix $S' \times P'$ would be performed. This computation would be necessary because of the change to a minimal support as the pattern extends (i.e. as the n gets higher). For instance in a sequence of states s_1, s_2, s_3 the s_2 and s_3 might get the minimal support and create a pattern but s_1 and s_2 might not. In the next pass however the minimal support would be lowered and the pattern s_2 and s_3 might be extended by the preceding state s_1 forming the pattern of s_1, s_2, s_3.

The last two steps will be repeated increasing the length of the patterns (n-grams) until at least one pattern with minimal support is found. In order to improve the efficiency, in each iteration the navigations data that does not contain any pattern of a current size is dropped. However, the count of sessions is kept.

3.2 Computing the support

Support is set as the relative frequency (or probability) of a transition between two states or between a pattern and a state. Basically there are two measures that can be used. The obvious way of computing relative frequencies of transitions would be to get the fraction of actual occurrences for a given transition as compared to the total number of transitions from a given state or pattern. Formally, the computation of the frequency f based on the count of occurrence c of a given pattern of size n can be stated as follows:

$$f(s_n|s_1, s_2, \ldots s_{n-1}) = c(s_1,..,s_n)/c(s_1, \ldots,s_{n-1}) \qquad (2)$$

In this way a measure computing the intra-pattern support can be used. The measure computing the inter-pattern support is based on the total number of all transitions. Theoretically, the total number of transitions would be the number of states raised to the power of two. This would however include also so called unseen transitions. Therefore the total number of seen transitions would be computed based on the observed states in session data. The following formula might be used:

ISBN: 978-1-61804-056-5

$$TT = SC - (n - 1)NS \quad (3)$$

where TT is the total number of transitions; SC is the count of all the states in all the sessions; n – is the size of the pattern that is being searched in a particular iteration; NS is the number of sessions considered;

Both measures can be used to specify a threshold which would determine whether a given pattern would be considered in the next iterations.

3.3 Identification of subsequences

The steps described above would determine the patterns based on the specified support. However, it might happen that one sequence would contain two or more patterns that would be interrupted by any number of "noise" states. Due to the noise states the patterns would not be united into one pattern using the previously described approach. In order to determine such subsequences the matrix consisting of sessions and patterns $Q \times P'$ would be constructed (Q represents the set of sessions). Since it is highly unlikely that a certain unit of learning would be visited many times in one session it could be assumed that each pattern would appear in the sequence only once. Thus the matrix would contain for a given sequence the index of beginning state for a given pattern. The index of a state is zero based. Hence, if for instance the first state of a given pattern is the third state in the sequence 2 is set. If a given pattern is not contained in a particular subsequence -1 is set in that case. Creating matrix like this would enable for sorting the patterns in the sequence according to their appearance in that sequence. For sequences in which more than one pattern where identified the above described algorithm would be used (see section 3.1). However, in this case the patterns of patterns would be determined.

4 Conclusion

The approach is going to be experimentally validated on a course consisting of around 35 units of learning and the results will be compared with other approaches, especially with knowledge based approach employing ontological modeling and reasoning.

Regardless, the validation, the efficiency of the approach is burden by many passes through the data. That might represent a problem especially for a large number of states and for longer sequences. The alternative solution to reduce the number of passes would be based on computation of minimal support without knowing the total number of occurrences. Given the minimal support could be set heuristically or experimentally, the patterns of frequent sequences could be determined in one pass.

Acknowledgement

The research has been supported by the Czech Grant Foundation, Grant No. 406/09/0346.

References:

[1] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of International Conference on Data Engineering (ICDE 1995)*, Los Alamitos, 1995, pp. 3-14.

[2] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems,* 2011.

[3] J. Bobadilla, F. Serradilla, and A. Hernando, "Collaborative filtering adapted to recommender systems of e-learning," *Knowledge-Based Systems,* vol. 22, pp. 261-265, 2009.

[4] L. Botturi, M. Derntl, E. Boot, and K. Figl, "A classification framework for educational modeling languages in instructional design," in *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT)*, Kerkrade, 2006.

[5] P. Čech, "Towards an Intelligent Learning Environment Using Ontology Based Reasoning," in *Intelligent Environments 2011, Smart Office and Other Workplaces 2011*, Nottingham, 2011.

[6] L. De-Marcos, J.-J. Marténez, and A. Gutirrez, "Particle Swarms for Competency-Based Curriculum Sequencing," presented at the Proceedings of the 1st world summit on The Knowledge Society: Emerging Technologies and Information Systems for the Knowledge Society, Athens, Greece, 2008.

[7] P. Dráždilová, G. Obadi, K. Slaninová, S. Al-Dubaee, J. Martinovič, and V. Snášel, "Computational Intelligence Methods for Data Analysis and Mining of eLearning Activities," in *Computational Intelligence for Technology Enhanced Learning*. vol. 273, F. Xhafa, S. Caballé, A. Abraham, T. Daradoumis, and A. Juan Perez, Eds., ed:

Springer Berlin / Heidelberg, 2010, pp. 195-224.

[8] G. A. Fink, "n -Gram Models " in *Markov Models for Pattern Recognition*, ed: Springer Berlin Heidelberg, 2008, pp. 95-113.

[9] G. Grimmett, *Probability and random processes*: Oxford University Press, 2001.

[10] T.-C. Hsieh and T.-I. Wang, "A mining-based approach on discovering courses pattern for constructing suitable learning path," *Expert Systems with Applications,* vol. 37, pp. 4156-4167, 2010.

[11] E. Kontopoulos, D. Vrakas, F. Kokkoras, N. Bassiliades, and I. Vlahavas, "An ontology-based planning system for e-course generation," *Expert Systems with Applications,* vol. 35, pp. 398-406.

[12] A. Milani, S. Suriani, and V. Poggioni, "Modeling educational domains in a planning framework," presented at the Proceedings of the 7th international conference on Electronic commerce, Xi'an, China, 2005.

[13] A. L. Montgomery, "Using Clickstream Data to Predict WWW Usage," Carnegie Mellon University, Pittsburgh, 1999.

[14] J. Pei, "Pattern-Growth Methods For Frequent Pattern Mining," Ph.D., Simon Fraser University, 2002.

[15] R. Peredo, A. Canales, A. Menchaca, and I. Peredo, "Intelligent Web-based education system for adaptive learning," *Expert Systems with Applications,* vol. 38, pp. 14690-14702.

[16] P. Phobun and J. Vicheanpanya, "Adaptive intelligent tutoring systems for e-learning systems," *Procedia - Social and Behavioral Sciences,* vol. 2, pp. 4064-4069, 2010.

[17] L. M. Romero-Moreno, F. J. Ortega, and J. A. Troyano, "Obtaining adaptation of virtual courses by using a collaborative tool and learning design," presented at the Proceedings of the 2007 Euro American conference on Telematics and information systems, Faro, Portugal, 2007.

[18] M. A. Sicilia, M. D. Lytras, S. Sanchez-Alonso, E. Garcia-Barriocanal, and M. Zapata-Ros, "Modeling instructional-design theories with ontologies: Using methods to check, generate and search learning designs," *Computers in Human Behavior,* vol. 27, pp. 1389-1398, Jul 2011.

[19] Y.-h. Wang, M.-H. Tseng, and H.-C. Liao, "Data mining for adaptive learning sequence in English language instruction," *Expert Systems with Applications,* vol. 36, pp. 7681-7686, 2009.

[20] M. Yaghmaie and A. Bahreininejad, "A context-aware adaptive learning system using agents," *Expert Syst. Appl.,* vol. 38, pp. 3280-3286, 2011.