

Dynamic Time Warping in Analysis of Student Behavioral Patterns

Kateřina Slaninová¹, Tomáš Kocyan², Jan Martinovič²,
Pavla Dráždilová¹, and Václav Snášel²

¹ VŠB - Technical University of Ostrava,
Faculty of Electrical Engineering and Computer Science,
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic
(katerina.slaninova,pavla.drazdilova)@vsb.cz

² VŠB - Technical University of Ostrava,
IT4Innovations,
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic
(tomas.kocyan,jan.martinovic,vaclav.snasel)@vsb.cz

Abstract. E-learning systems store large amount of data based on the history of users' interactions with the system. These pieces of information are usually used for further course optimization, finding e-tutors in collaboration learning, analysis of students' behavior, or for other purposes.

The paper deals with an analysis of students' behavior in learning management system. The main goal of the paper is to find, how selected methods can influence finding of behavioral patterns in learning management system and how we can reduce the amount of extracted sequences. The methods of process mining and sequential pattern mining were used for extraction of behavioral patterns. The authors present the comparison of selected methods for the definition of students' behavior with the focus to influence of dynamic time warping. Obtained patterns and relations between them are presented using complex networks; the visualization and pattern clusters extraction is optimized by spectral graph partitioning.

1 Introduction

E-learning is a method of education which utilizes a wide spectrum of technologies, mainly internet or computer-based, in the learning process. It is naturally related to distance learning, but nowadays is commonly used to support face-to-face learning as well. *Learning management systems* (LMS) provide effective maintenance of particular courses and facilitate communication within the student community and between educators and students [4]. Such systems usually support the distribution of study materials to students, content building of courses, preparation of quizzes and assignments, discussions, or distance management of classes. In addition, these systems provide a number of collaborative learning tools such as forums, chats, news, file storage etc.

LMS based on computer and web-based education environments provide storage of large amount of accessible information. These systems record information about students' actions and interactions onto log files or databases. Within these records, data about students learning habits can be found including favored reading materials, note

taking styles, tests and quizzes, ways of carrying out various tasks, communication with other students in virtual classes using chat, forum, and etc. Other common data, such as personal information about students and educators (user profiles), student results and user interaction data, is also available in the system databases [1].

Such data collections are essential for analyzing students' behavior and can be very useful in providing feedback both to students and educators. For students, this can be achieved through various recommended systems and through course adaptation based on student learning behavior. For teachers, some benefits would include the ability to evaluate the courses and the learning materials, to detect the typical learning behavior or to find possible students suitable for collaborative learning [18].

Regardless of LMS benefits, huge amount of recorded data in large collections makes often too difficult to manage them and to extract useful information from them. To overcome this problem, some LMS offer basic reporting tools. However, in such large amount of information the outputs become quite obscure and unclear. In addition, they do not provide specific information of student activities while evaluating the structure and content of the courses and its effectiveness for the learning process [23]. The most effective solution to this problem is to use data mining techniques [1].

The main goal of the paper is to compare selected data mining methods suitable for the extraction of students' behavioral patterns performed in LMS Moodle. The behavioral patterns are obtained using methods of process mining and sequential mining, the patterns are presented using methods from graph theory. The organization of the paper is as follows: Section 2 consists of the background related to the methods used for the analysis of students' behavior. Process mining issues and selected methods for comparison of sequences are presented here. In Section 3 is presented an extraction of sequences used for students' behavior description from log file of e-learning system. Then, we are presented results of experiments provided on e-learning system Moodle. The experiments are focused to the extraction of students' behavioral patterns and to the comparison of selected methods. For easier analysis of the students' behavior is important the reduction of amount of sequences. We have used spectral clustering algorithm, which determine number of important clusters with behavioral patterns. The last Section 4 contains the conclusion.

2 Analysis of Students' Behavior

Several authors published contributions with relation to mining data from e-learning systems to extract knowledge that describe students' behavior. Among others we can mention for example [11], where authors investigated learning process of students by the analysis of web log files. A 'learnograms' were used to visualize students' behavior in this publication. Chen et al. [3] used fuzzy clustering to analyze e-learning behavior of students. El-Hales [6] used association rule mining, classification using decision trees, E-M clustering and outlier detection to describe students' behavior. Yang et al. [22] presented a framework for visualization of learning historical data, learning patterns and learning status of students using association rules mining. The agent technology and statistical analysis methods were applied on student e-learning behavior to evaluate findings within the context of behavior theory and behavioral science in [2].

However, contributions oriented to analysis of students' behavior in e-learning systems describe the behavior using statistical information, for visualization and representation of obtained information are mostly used only common statistical tools like figures or graphs. They usually do not provide information about behavioral patterns with effective visualization, nor information about relations between students based on their behavior.

2.1 Process Mining

Our subject of interest in this paper is student behavior in LMS, which is recorded in form of events and stored in the logs. Thus, we can define the student behavior with the terms of process mining which are used commonly in business sphere. Aalst et al. [20, 19] defines event log as follows:

Definition 1. Let A be a set of activities (also referred as tasks) and U as set of performers (resources, persons). $E = A \times U$ is the set of (possible) events (combinations of an activity and performer). For a given set A , A^* is the set of all finite sequences over A . A finite sequence over A of length n is mapping $\sigma = \langle a_1, a_2, \dots, a_n \rangle$, where $a_i = \sigma(i)$ for $1 \leq i \leq n$. $C = E^*$ is the set of possible event sequences. A simple event log is a multiset of traces over A .

Then, student behavior in LMS can be described by set of event sequences. More detailed description is presented in Section 3.

The paper is oriented to finding behavioral patterns. Behavioral patterns are discovered using similarity of extracted sequences. A sequence is an ordered list of elements, denoted $\langle e_1, e_2, \dots, e_l \rangle$. Given two sequences $\alpha = \langle a_1, a_2, \dots, a_n \rangle$ and $\beta = \langle b_1, b_2, \dots, b_m \rangle$. α is called a subsequence of β , denoted as $\alpha \subseteq \beta$, if there exist integers $1 \leq j_1 < j_2 < \dots < j_n \leq m$ such that $a_1 = b_{j_1}, a_2 = b_{j_2}, \dots, a_n = b_{j_n}$. β is then a super sequence of α .

In the problem of finding similar behavior, we do not use traditional methods of sequential pattern mining where usually frequently repeated patterns are extracted. For finding the behavioral patterns, we need to use the methods for the sequence comparison, described in Section 2.2.

2.2 Comparison of Sequences

There are generally known two basic groups of algorithms for the comparison of two or more categorical sequences. The first group divides the algorithms by the fact, whether the sequences consist of ordered or unordered elements. The second group of algorithms focuses on the comparison of the sequences with the different lengths and with the possible error or distortion.

The basic approach to the comparison of two sequences, where the order of elements is important, is *The longest common substring* (LCS) method [10] (see example in Table 1). As obvious from the name of the method, the main principle of the method is to find the length of the common longest substring. Given the two sequences x and y , we can find such subsequence $z = \langle z_1, z_2, \dots, z_p \rangle$, where $z_k = x_{i+k-1} = y_{j+k-1} \forall k = 1, \dots, p$ and $p \leq m, n$.

The LCS method respects the order of elements in the sequence. However, the main disadvantage is, that it can find only identical subsequences, where no extra element is presented in the sequence. For some domains, typically where is large amount of different sequences, gives this fact too strict limitation.

As a solution of this problem we can consider *The longest common subsequence* (LCSS) described for example in [12] (see example in Table 1). Contrary to The longest common substrng, this method allows (or ignores) the inserted extra elements in the sequence, and therefore, it is immune to slight distortions.

	a	b	c
Sequence X	E <u>ABC</u> F	E <u>AEB</u> CE	<u>ABB</u> CC
Sequence Y	<u>Z</u> ABC <u>T</u>	F <u>ABC</u> F	E <u>AB</u> CE
Longest Common substrng	ABC	BC	AB
Common subsequence (LCSS)	ABC	ABC	ABC
Common subsequence (TWLCS)	ABC	ABC	ABBCC

Table 1. Results of algorithms for subsequence detection

Whether we define the similarity of compared sequences as a function using a length of common subsequence, we can find one characteristic of this method. The length of the common subsequence is not immune to recurrence of identical elements, which can occur only in one of the compared sequences. We can find such situations, for example due to inappropriate sampling or due to any kind of distortion.

In some applications, it is suitable (or sometimes even required) to eliminate such type of distortions and to work with them like with equivalent elements. The solution is in another method, *The time-warped longest common subsequence* (T-WLCS) [9] (see example in Table 1). The method combines the advantages of LCSS method with dynamic time warping[13]. Dynamic time warping is used for finding the optimal visualization of elements in two sequences to match them as much as possible. This method is immune to minor distortions and to time non-linearity. It is able to compare sequences, which are for standard metrics evidently not comparable.

The method emphasizes recurrence of elements in one of the compared sequences. Due to this fact the length of the common subsequence can be longer than the shorter length of the compared sequences.

In the experiments described in the paper, the authors compare the impact of LCSS and T-WLCS methods to the construction of derived network based on similar behavior of students in e-learning system.

3 Sequence Extraction in LMS Moodle

In this section is presented the extraction of students' behavioral patterns performed in the e-learning educational process. The analyzed data collections were stored in the Learning Management System (LMS) Moodle logs used to support e-learning education at Silesian University, Czech Republic.

The logs consist of records of all events performed by Moodle users, such as communication in forums and chats, reading study materials or blogs, taking tests or quizzes etc. The users of this system are students, tutors, and administrators; the experiment was limited to the events performed only by students.

Let us define a set of students (users) U , set of courses C and term *Activity* $a_k \in A$, where $A = P \times B$ is a combination of activity prefix $p_m \in P$ (e.g. course view, resource view, blog view, quiz attempt) and an action $b_n \in B$, which describes detailed information of an activity prefix (concrete downloaded or viewed material, concrete test etc.). *Event* $e_j \in E$ then represents the activity performed by certain student $u_i \in U$ in LMS. On the basis of this definition, we have created a set S_i of sequences s_{ij} for the user u_i , which represents the students' (users') paths (sessions) on the LMS website. *Sequence* s_{ij} is defined as a sequence of activities, for example $s_{ij} = \langle a_{1j}, a_{2j}, \dots, a_{qj} \rangle$, which is j -th sequence of the user u_i .

The sequences were extracted likewise the user sessions on the web; the end of the sequences was identified by at least 30 minutes of inactivity, which is based on our previous experiments [5]. Similar conclusion was presented by Zorrilla et al. in [23].

Using this method, we have obtained a set of all sequences $S = \cup_{\forall i} S_i$, which consisted of large amount of different sequences s_i performed in LMS Moodle. We have selected the course Microeconomy A as an example for the demonstration of proposed method. In Table 2 is presented detailed information about the selected course.

Records	Students	Prefixes	Actions	Sequences
65 012	807	67	951	8 854

Table 2. Description of Log File for Course Microeconomy A

Sequence appearance in the selected course follows the power law distribution.

As mentioned in Section 3, the obtained set S of sequences consisted of large amount of different sequences, often very similar. Such large amount of information is hard to clearly visualize and to present in well arranged way. Moreover, the comparison of users based on their behavior is computationally expensive with such dimension. Therefore, we present the identification of significant behavioral patterns based on the sequence similarity, which allows us to reduce amount of extracted sequences.

Following experiment is oriented to exploration, how the different methods for measurement of sequence similarity can influence finding of behavioral patterns. We have used LCSS a T-WLCS methods for the similarity measurement of sequences, described in Section 2.2, with comparison to the common one, cosine similarity. Cosine similarity [15] is well known method for similarity measurement in informational retrieval while working with vector model. Both methods LCSS and T-WLCS find the longest common subsequence α of compared sequences β_x and β_y , where $\alpha \subseteq \beta_x \wedge \alpha \subseteq \beta_y$, with relation to both methods, see Section 2.2. Similarity was counted by the Equation 1.

$$Sim(\beta_x, \beta_y) = \frac{l(\alpha) * h}{l(\beta_x) * l(\beta_y)}, \quad (1)$$

where $l(\alpha)$ is a length of the longest common subsequence α for sequences β_x and β_y ; $l(\beta_x)$ and $l(\beta_y)$ are analogically lengths of compared sequences β_x and β_y , and

$$h = \frac{\text{Min}(l(\beta_x), l(\beta_y))}{\text{Max}(l(\beta_x), l(\beta_y))} \quad (2)$$

Table 3 shows cosine similarity and similarity counted by the methods LCSS and T-WLCS for selected sequences and sequence '4408,4409,4407,2181,2133'.

Sequence	Similarity		
	Cosine	LCSS	T-WLCS
'4408,4409,4407,2181,2133'	1	1 (5)	1 (5)
'4409,4407,4408,2181,2133'	1	0.64(4)	0.64 (4)
'4407,4409,4408,2181,2133'	1	0.36 (3)	0.36 (3)
'4408,4409,4407,2181'	0.991	0.512 (4)	0.512 (4)
'4409,4407,4408,2181'	0.991	0.288 (3)	0.288 (3)
'4409,4407,4408,2181,2133,225'	0.979	0.370 (4)	0.370 (4)
'4408,4409,4407,2181,225,2133'	0.979	0.579 (5)	0.579 (5)
'4407,4408,4409,2181,2133,225'	0.979	0.370 (4)	0.370 (4)
'4407,4408,4409,2181,2133,225,2133'	0.976	0.233 (4)	0.364 (5)
'4408,4409,4407,4408,2181,2133'	0.972	0.579 (5)	0.579 (5)

Table 3. Example: Similarity of '4408,4409,4407,2181,2133'to Selected Sequences

Numbers in the brackets present the length of the founded longest common sequence for each method. From Table 3 (for example from the second row of the table) is evident the significant disadvantage of cosine similarity: it does not take into consideration the ordering of events in the sequence, while the methods LCSS and T-WLCS do. However, cosine similarity supports weighted vector model, where frequency of attributes is taken into consideration. In our method, tf-idf weighting was used. From the 9th row we can see the difference between the methods LCSS and T-WLCS. T-WLCS method takes into consideration the recurrence of elements in one of the compared sequences.

On the basis of selected method for finding the similarity of sequences, we have constructed the similarity matrix for sequences ($|S| \times |S|$) which can be represented using tools of graph theory. For the visualization of network was constructed weighted graph $G(V, E)$, where weight w is defined as function $w : E(G) \rightarrow R$, when $w(e) > 0$. Set V is represented by set of sequences S , weights w are evaluated by the similarity of sequences, see Equation 1, depending on selected method. In Table 4 is more detailed description of weighted graphs of sequences, where weight is defined by cosine similarity and similarity counted on the basis of LCSS and T-WLCS method for selected threshold θ . The number of nodes for each graph is 5908.

From Table 4 we can see, that each graph consists of large amount of similar sequences. Moreover, they are dense and very large for further processing. Better interpretation of results is possible by finding the components, which can represent the

Cosine Measure				
θ	Isolated Nodes	Edges	Avg. Degree	Avg. Weighted Degree
0.1	0	13292202	2249.865	464.377
0.2	2	4651152	787.263	261.303
0.3	5	2040406	345.363	155.013
0.4	32	1050138	177.748	97.387
0.5	122	554278	93.818	60.066
0.6	395	290632	49.193	35.747
0.7	897	147984	25.048	20.181
0.8	1851	67584	11.439	10.034
0.9	3289	21966	3.718	3.524
LCSS				
θ	Isolated Nodes	Edges	Avg. Degree	Avg. Weighted Degree
0.1	113	1622206	274.578	45.563
0.2	672	352504	59.666	17.112
0.3	1711	85266	14.432	6.037
0.4	2586	39922	6.757	3.309
0.5	3812	13796	2.335	1.351
0.6	4891	3320	0.562	0.371
0.7	5697	390	0.066	0.049
0.8	5900	12	0.002	0.002
0.9	5908	0	0	0
T-WLCS				
θ	Isolated Nodes	Edges	Avg. Degree	Avg. Weighted Degree
0.1	31	5577366	944.036	179.431
0.2	143	1739042	294.354	88.883
0.3	606	534648	90.496	38.735
0.4	1200	271826	46.010	22.998
0.5	2465	103028	17.439	10.430
0.6	3781	29298	4.959	3.596
0.7	5038	8080	1.368	1.269
0.8	5517	5914	1.001	0.997
0.9	5568	5788	0.980	0.980

Table 4. Description of Sequence Graphs for Cosine Similarity, LCSS and T-WLCS

behavioral patterns. The graph reduction using only threshold θ leads to undesirable loss of information. Due to this reason, we have used spectral clustering by Fiedler vector and algebraic connectivity [7, 8]. More detailed description of finding components using this method was presented in our previous work [21, 14].

In table 5 are described graphs with different methods for computing similarity between sequences. The threshold was selected $\theta \geq 0.1$ or 0.2 for comparison between the largest components with similar size (bold numbers). We have analysed the largest connected components of each graph and we have obtained significant clusters after spectral clustering.

	Cosine $\theta \geq 0.1$	LCSS $\theta \geq 0.1$	T-WLCS $\theta \geq 0.1$	T-WLCSS $\theta \geq 0.2$
Connected Components	1	122	33	145
Size of the Largest Component	5908	5778	5875	5763
Clusters in the Largest Component	2	19	10	20
Size of Cluster 1	2969	2629	3472	1623
Size of Cluster 2	2939	1573	762	1247
Size of Cluster 3		1138	745	772
Size of Cluster 4		307	557	567
Size of Cluster 5		50	322	561

Table 5. Description of Components After Spectral Clustering

In Figure 1 we can see the weighted graph constructed for better visualization of the components with the similar sequences.

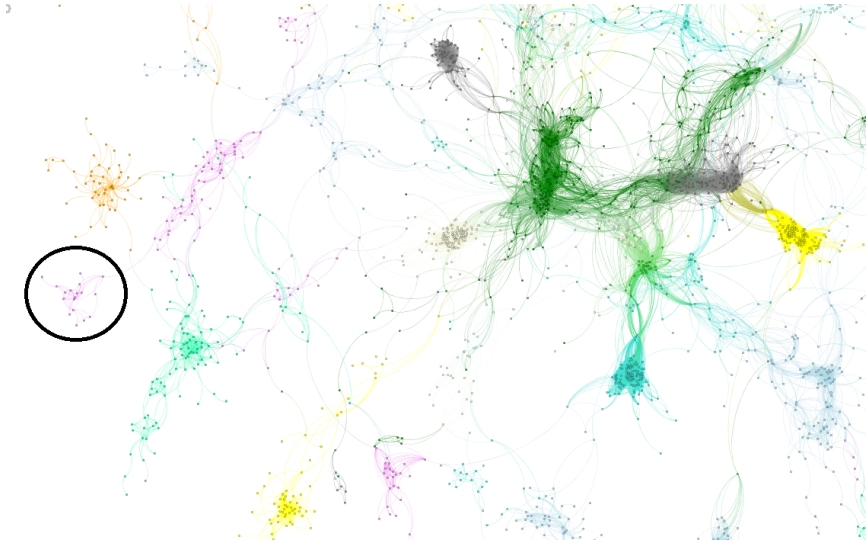


Fig. 1. Weighted Graph With Behavioral Patterns

The graph was constructed using an open source software Gephi³. In Figure 1, the nodes in the graph represent the sequences, while the edges are weighted by their similarity using T-WLCS method. The graph was constructed using threshold $\theta=0.8$. Each component in the graph can represent a behavioral pattern of similar sequences.

³<http://gephi.org>

Figure 2 shows the detailed view to the selected component from the graph on Figure 1. The component consists of 2940 sequences, each node is described by the sequence, while the actions in sequences have its own unique ID.

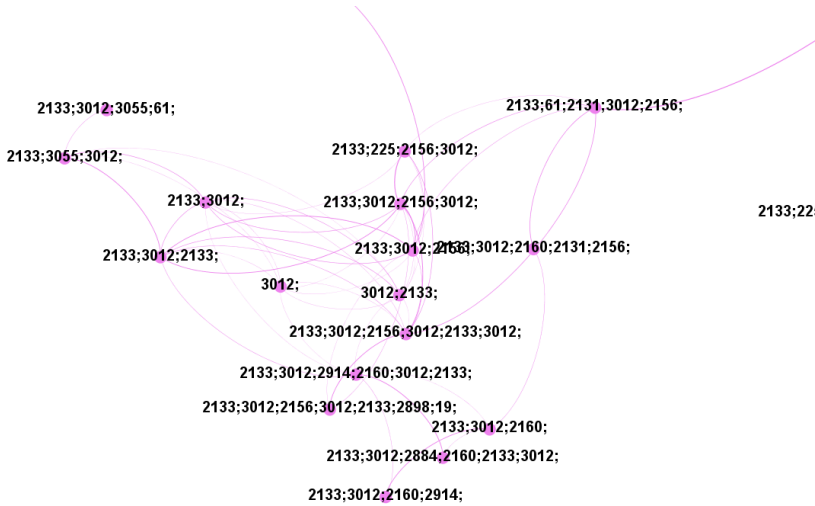


Fig. 2. Selected Component of Similar Sequences

It is possible to generate subgraphs relevant to selected activity, which can be in the area of our interest. The filtering by selected activities is performed by using vector model of sequences \times activities.

4 Conclusion

The paper is oriented to finding the students' behavioral patterns performed in the e-learning system. The behavioral patterns were obtained using the methods of process mining and sequential mining, the patterns were visualized by the methods from graph theory. The authors focused on the comparison of the selected data mining methods suitable for the definition of the sequence similarity.

On the basis of previous experiments with suffix tree method and common vector model [17, 16] we have found, that the sequences are order dependent and it is better to respect this fact while comparing the sequence similarity. Due to this reason, the methods for finding the longest common subsequence were used.

In the experiments, the comparison of methods LCSS and T-WLCS with common vector model was described. Our results showed that each method has its unique characteristics. Vector model does not take into consideration ordering of actions inside the sequences, which is important disadvantage. On the other side, it allows weighting of activities on the basis of their frequency. LCSS and T-WLCS methods work with action

ordering and allow slight distortions in the sequence, while T-WLCS emphasizes the recurrence of elements in one of the compared sequences. These methods allowed to find the similarity between the two sequences more precisely.

On the basis of our experiments we have found that proposed method is usable for sequence extraction. Moreover, it can be effectively used for the reduction of sequence dimension. As we can see from presented results, we need to provide more precise division of extracted components to obtain more accurate behavioral patterns in some cases. The LCSS and T-WLCS methods are more time demanding than common cosine similarity. In our further work we intent to focus on their optimization. Another possible further work can be oriented to the definition of sequence similarity which will exploit the advantages from cosine measure and methods for finding the longest common subsequence.

Proposed method is suitable for finding the students' behavioral patterns in e-learning, which can be useful in providing feedback both to students and educators. Such type of information is valuable neither in e-learning sphere, nor in other areas like business process mining, finding behavior of users on the web, marketing etc.

Acknowledgment

This work was partially supported by SGS, VSB – Technical University of Ostrava, Czech Republic, under the grant No. SP2012/151 Large graph analysis and processing and by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

References

1. F. Castro, A. Vellido, A. Nebot, and F. Mugica. Applying data mining techniques to e-learning problems. *Studies in Computational Intelligence (SCI)*, 62:183–221, 2007.
2. B. Chen, C. Shen, G. Ma, Y. Zhang, and Y. Zhou. The evaluation and analysis of student e-learning behaviour. In *IEEE/ACIS 10th International Conference on Computer and Information Science (ICIS)*, 2011, pages 244 – 248, 2011.
3. J. Chen, K. Huang, F. Wang, and H. Wang. E-learning behavior analysis based on fuzzy clustering. In *Proceedings of International Conference on Genetic and Evolutionary Computing*, 2009.
4. P. Dráždilová, G. Obadi, K. Slaninová, S. Al-Dubae, J. Martinovič, and V. Snášel. Computational intelligence methods for data analysis and mining of elearning activities. In F. Xhafa, S. Caballe, A. Abraham, T. Daradoumis, and J. Perez, editors, *Studies in Computational Intelligence For Technology Enhanced Learning*, volume 273, pages 195–224. Heidelberg, Germany: Springer-Verlag, 2010.
5. P. Dráždilová, K. Slaninová, J. Martinovič, G. Obadi, and V. Snášel. Creation of students' activities from learning management system and their analysis. In A. Abraham, V. Snášel, and K. Wegrzyn-Wolska, editors, *IEEE Proceedings of International Conference on Computational Aspects of Social Networks CASON 2009*, pages 155–160, 2009.
6. A. El-halees. Mining students data to analyze learning behavior: a case study. 2008.
7. M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298–305, 1973.

8. M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25:619–633, 1975.
9. A. Guo and H. Siegelmann. *Time-Warped Longest Common Subsequence Algorithm for Music Retrieval*, pages 258–261. Universitat Pompeu Fabra, 2004.
10. D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
11. A. Hershkovitz and R. Nachmias. Learning about online learning processes and students' motivation through web usage mining. *Interdisciplinary Journal of E-Learning and Learning Objects*, 5:197–214, 2009.
12. D. S. Hirschberg. Algorithms for the longest common subsequence problem. *J. ACM*, 24:664–675, October 1977.
13. M. Müller. *Information Retrieval for Music and Motion*. Springer, 2007.
14. G. Obadi, P. Dráždilová, J. Martinovič, K. Slaninová, and V. Snášel. Using spectral clustering for finding student's patterns of behavior in social networks. In *Proceedings of the DATESO 2010 Annual International Workshop on Databases, TExts, Specifications and Objects*, pages 118–130, 2010.
15. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
16. K. Slaninová, R. Dolák, M. Miškus, J. Martinovič, and V. Snášel. User segmentation based on finding communities with similar behavior on the web site. In *Proceedings - 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT Workshops 2010*, pages 75–78, 2010.
17. K. Slaninová, J. Martinovič, T. Novosád, P. Dráždilová, L. Vojáček, and V. Snášel. Web site community analysis based on suffix tree and clustering algorithm. In *Proceedings - 2011 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2011*, pages 110–113, 2011.
18. V. Snášel, A. Abraham, J. Martinovič, P. Dráždilová, K. Slaninová, T. Daradoumis, F. Xhafa, and A. Martíñez-Monés. E-assessment of individual and group learning processes. *Journal of Computational and Theoretical Nanoscience*, 9(2):286–303, 2012.
19. W. M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Heidelberg, 1st edition, 2011.
20. W. M. P. van der Aalst, H. A. Reijers, and M. Song. Discovering social networks from event logs. *Comput. Supported Coop. Work*, 14(6):549–593, 2005.
21. L. Vojáček, J. Martinovič, K. Slaninová, P. Dráždilová, and J. Dvorský. Combined method for effective clustering based on parallel som and spectral clustering. In V. Snášel, J. Pokorný, and K. Richta, editors, *Proceedings of the 11th Annual Workshop DATESO 2011*, pages 120–131. VŠB - TU Ostrava, 2011.
22. F. Yang, R. Shen, and P. Han. Construction and application of the learning behavior analysis center based on open e-learning platform. 2002.
23. M. Zorrilla, E. Menasalvas, D. Marín, E. Mora, and J. Segovia. Web usage mining project for improving web-based learning sites. In *Computer Aided Systems Theory – EUROCAST 2005*, volume 3643/2005 of *Lecture Notes in Computer Science*, chapter Web Usage Mining Project for Improving Web-Based Learning Sites. Springer Berlin / Heidelberg, 2005.