A Framework to Provide Personalization in Learning Management Systems through a Recommender System Approach

Hazra Imran, Quang Hoang, Ting-Wen Chang, Kinshuk, and Sabine Graf

Athabasca University, Edmonton, Canada
{hazraimran,tingwenchang,kinshuk,sabineg}@athabascau.ca,
hoangdangquang@yahoo.com

Abstract. Personalization in learning management systems (LMS) occurs when such systems tailor the learning experience of learners such that it fits to their profiles, which helps in increasing their performance within the course and the quality of learning. A learner's profile can, for example, consist of his/her learning styles, goals, existing knowledge, ability and interests. Generally, traditional LMSs do not take into account the learners' profile and present the course content in a static way to every learner. To support personalization in LMS, recommender systems can be used to recommend appropriate learning objects to learners, not only based on their individual profile but also based on what worked well for learners with a similar profile. In this paper, we propose a framework to integrate a recommender system approach into LMS. The proposed framework is designed with the goal of presenting a flexible integration model which can provide personalization by automatically suggesting learning objects to learners based on their current situation as well as successful learning experiences of learners with similar profiles in a similar situation. Such advanced personalization can help learners in many ways such as reducing the learning time without negative impact on their marks, improving learning performance as well as increasing the level of satisfaction.

Keywords: Personalization, E-Learning, Learning Management Systems, Recommender System.

1 Introduction

With the advancement in technology, e-learning is becoming more and more popular. E-learning can comprise either fully online or blended courses. While in fully online courses, everything is delivered in an online mode, blended courses have an online and a face-to-face component. To facilitate the delivery and organization of e-learning, especially in large-scale educational institutions, learning management systems (LMSs) are typically used. According to Szabo [1], a "Learning Management System is the infrastructure that delivers and manages instructional content, identifies and assesses individual and organizational learning or training goals, tracks the progress towards meeting those goals, and collects and presents data for supervising

N.T. Nguyen et al. (Eds.): ACIIDS 2014, Part I, LNAI 8397, pp. 271–280, 2014.

the learning process of an organization as a whole". Courses in LMSs typically consist of learning objects (LOs). LOs can be defined as "any entity, digital or non-digital, that may be used for learning, education or training" [2]. Generally, LMSs deliver the same kind of course structure and LOs to each learner [3, 4]. This is coined as "one size fits all" approach. But, each learner has different characteristics, and therefore, a "one size fits all" approach does not support most learners particularly well. One of the possible ways to support each learner individually based on his/her characteristics is the use of personalization. Personalization in LMS refers to the functionality which enables the system to uniquely address a learner's needs and characteristics such as levels of expertise, prior knowledge, cognitive abilities, skills, interests, preferences and learning styles [5] so as to improve a learner's satisfaction and performance within the course. Personalization in the form of recommendations for resources and learning materials is an area that has gained significant interest from researchers recently. Recommendations exhibit prominent social behavior in day-to-day life [6]. In real life, people seek and trust the recommendations of others in making decisions. Reflecting this societal behavior, recommender systems are increasingly being adopted in different fields in order to support users in their decision making processes and help them in making wise choices with less effort. Many online companies, such as Amazon [7] and Netflix [8] are using recommender systems to offer users personalized information to help them in their decisions [9]. Such successful integration of recommender systems in e-commerce has prompted researchers to explore similar benefits in the e-learning domain [10, 11] since the integration of recommender systems in e-learning has high potential for achieving advanced personalization.

This paper presents a novel framework that integrates a recommender system into a LMS in order to provide personalization to learners based on their situations and successful learning experiences of other learners with similar characteristics in similar situations. The proposed framework is designed to:

- Integrate a recommender system into LMSs
- Consider a learner's profile consisting of characteristics like learning style, expertise level, prior knowledge and performance to provide advanced personalization.
- Form a neighborhood of learners based on their profile and discover associations among learning objects (through association rule mining) that led to successful learning experiences of other similar learners in similar situations.
- Create a personalized list of recommendations of learning objects to be presented to an individual learner in situations where members of his/her neighborhood benefitted from the suggested learning objects.

An important feature of the proposed framework is the approach to find similar learners for building a neighborhood of learners. The approach is advanced as it is considering different characteristics of the learners such as their learning styles, prior knowledge, expertise level and performance within the course. Accordingly, we get more similar learners in a neighborhood, which enables our approach to generate more suitable recommendations that fit to the learners' situations more accurately.

The paper is organized as follows. The next section begins with an analysis of the state of the art in the field of providing personalization through recommender systems using data mining technique. Section 3 describes the proposed framework and its main components. Section 4 concludes the paper and discusses future research directions.

2 Related Work

Recommender systems use behavior or opinions of a group with similar characteristics/behavior to help individual users in making decision from vast available choices. Recently, some recommender systems have been applied in the e-learning domain. In this section, such works are described based on two directions: First, we discuss research works that focus on providing recommendations based on learners' activities in a course. These works use association rule mining to find associations among the activities done by learners and then recommendations are provided accordingly to the individual learner. In these works, recommendations are based on learners' activities in a course rather than learner characteristics, needs and/or profiles. Second, we describe research works that provide recommendations based on similar learners who have similar characteristics. These works either used clustering techniques based on learners' characteristics to create groups or compute similarity between the learners based on the ratings they provided. Subsequently, recommendations were provided based on what worked well for similar learners.

Research work falling under the first group used association rule mining to find rules based on which recommendations were provided to learners. For example, Zaiane [12] built a recommender agent that provides recommendations of learning activities within a course based on learner access histories. Khribi, Jemni and Nasraoui [13] developed a recommender system based on learners' recent navigation histories, and similarities and dissimilarities among the contents of the learning materials. The first group of research works considers the web usage data of the learners in a course as well as associations between the activities of learners in a course. These works focus on grouping learners based on their activities. Our work is different from these works as we are finding similar learners based on their characteristics (e.g., learning styles, skills, prior knowledge and performance) rather than activities, which has potential to allow for a more accurate grouping since we are considering the underlying reason for learners' behavior (e.g., not much background knowledge, a certain learning style) rather than just the actions themselves.

The second group of research works finds similar learners and then recommendations are provided based on the information from these similar learners. For example, Tang and McCalla [14] proposed an evolving web-based learning system that finds the relevant content from the web. They use a clustering technique to cluster the learners (based on their learning interests) to calculate learners' similarities for content recommendation. Tai, Wu and Li [15] proposed a course recommender system by using self-organizing maps and data mining techniques. Self-organizing maps were used to classify learners based on similar interests into groups. Then a data mining technique was used to elicit the rules of the best learning path for each group of learners. Kerkiri, Manitsaris and Mavridou [16] proposed a framework that uses reputation metadata in a recommender system. Reputation is the cumulative scale of user opinions regarding persons, products, and ideas. The system describes the learning resources based on learning object metadata and the leaners profile based on PAPI [17]. The registered learners were asked to provide information for their profile including qualifications, skills, licenses etc. The similarity between the learners is calculated by using the Pearson's r correlation coefficient. The learners were asked to provide ratings to learning resources, which is termed as reputation metadata. Having all the information about learners and learning resources (metadata and reputation metadata), collaborative filtering was applied to recommend personalized learning resources. An experiment showed that the use of reputation metadata augmented learners' satisfaction by retrieving learning materials which were evaluated positively by learners. Yang, Sun, Wang, and Jin [18] proposed a personalized recommendation algorithm for curriculum resources based on semantic web technology using a domain ontology. The algorithm first collects curriculum resources of interest in terms of user evaluation and user browsing behavior. Yang et al. [18] assume that "different users evaluate different core concepts, according to domain knowledge, as there is a certain similarity between core concepts, so there are similarities between the user's interests". Therefore, similarity among users can be computed from similarity between core concepts. The users were asked to provide ratings to the learning resources. The similarity among learners was computed based on their ratings. Then the interest degree of users is calculated for each interest category of the nearest neighbors and finally recommendations were provided based on interest of the nearest neighborhood.

The research works in the second group provide recommendations based on similar learners. However, these works mainly used the learner interest and ratings from the learner as the parameter for creating groups. In the e-learning domain, we generally do not have ratings for the content. If a learner is asked to provide ratings for each learning object in a course, it puts a lot of effort on the learner. In our work, we aim at providing automatic recommendations without requiring any additional effort from learners. Instead of using ratings, information about whether or not a certain learning object was helpful for a particular learner is retrieved from his/her navigation and behavior in the course as well as his/her performance. Furthermore, our work is different in that it considers students' characteristics, including their learning styles, expertise level and prior knowledge, together with their performance in the course. By identifying similar learners based on multiple characteristics, we expect to place a learner together with learners who learn in a very similar fashion, leading to more accurate recommendations.

3 Framework for Integrating a Recommender System Approach into LMSs

This section describes the proposed framework for integrating a recommender system approach into LMSs. The aim of the framework is to enable LMSs to provide recommendations to learners based on the successful learning experience of other similar learners. The framework is illustrated in Figure 1. The modules in this framework are designed in such a way that they are not dependent on the LMS and hence, can be integrated easily into different LMSs with minimum required changes. In the following subsections, the modules are described in more detail.

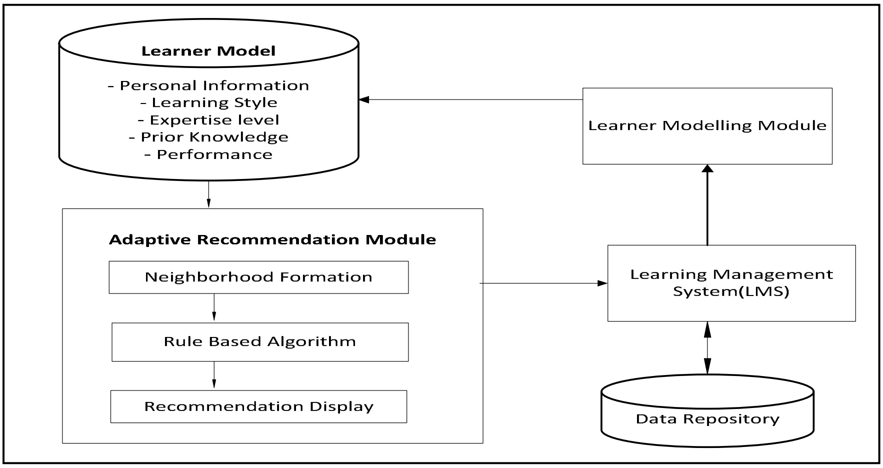

Fig. 1. Architecture of proposed framework

3.1 Learner Modelling Module

The Learner Modelling Module aims to generate the Learner Model. The Learner Model contains information gathered from the learner, i.e. personal information (first name and last name), previous knowledge (related to the course), expertise level (i.e., Beginner, Intermediate or Expert), learning styles and performance. When learners register in the LMS through a registration form, the Learner Model is initialized. During the registration, learners provide personal information such as first name and last name, which is stored in the Learner Model. Furthermore, they are asked about their prior knowledge and expertise level. In addition, the Learner Model aims at gathering information about the learning styles of learners. Every learner learns in a different and unique way, as each one has their own preferences, need and approaches toward learning. These individual differences are coined as learning styles. According to Dunn, Dunn, and Freeley [19], learning styles can be defined as “unique manners in which learners begin to concentrate on, process, absorb, and retain new and difficult information”. To identify the learning styles, the Learner Modelling Module uses a well-investigated and commonly used questionnaire, called Index of Learning Styles (ILS)[20] developed by Felder and Solomon, which identifies the preferences of learning in four dimensions based on the Felder-Silverman Learning Style Model [21]. These four dimensions are: active/reflective, sensing/intuition, visual/verbal and sequential/global. At the time of registration, a learner is asked to fill out the ILS questionnaire, consisting of 44 questions. Based on a learner’s responses, the result is calculated as four values between +11 to -11 indicating the preference on each of the four learning style dimensions. These four values are stored in the learner model and are used as the identified learning styles of learners. Performance data describe a learner's performance in the course units. The performance data are gathered from the learner's performance on assignments and quizzes within each unit.

3.2 Learner Model

The Learner Model aims to store the information about the learner for personalization purpose, including the four values of the learner's learning style, their prior knowledge, expertise level and the performance of the learner within the course. The learner model information is used by the Adaptivity Recommendation Module to generate recommendations.

3.3 Adaptivity Recommendation Module (ARM)

This module is responsible for creating and displaying recommendations based on similar learner profiles. Currently, the proposed framework can provide the recommendations for 11 types of learning objects (LOs) namely Commentaries (give a brief overview on what the unit/section is about), Content Objects (are the learning material of the course and are rich in content), Reflection Quizzes (contain open-ended questions about the topics in the section), Self-Assessment Tests (include closed-ended questions about the topics in a section), Discussion Forums (allow learners to ask question and join/initiate a discussions with their peers and instructor), Additional Reading Materials (provide additional sources of reading materials about the topics in a section), Animations (explain the concepts of a section in an animated multimedia format), Exercises (allow learners to practice their knowledge and skills), Examples (illustrate the theoretical concepts in a more concrete way), Real-Life Applications (demonstrate how the learned material can be applied in a real-life situations) and Conclusions (summarize the topics learned in a section).

ARM has information about learners' behavior through accessing log data tracked by the LMS, which include what learning objects have been visited by each learner and how much time he/she spent on each learning object. This is information that every LMS typically tracks. In order to provide recommendations, ARM finds the neighbors of a learner who have similar characteristics. We are making the assumption that since learners within a neighborhood are similar to each other, successful learning experiences of one learner can be beneficial to other similar learners. The overall aim of ARM is to provide recommendations of learning objects to the learner in a situation where the learner is visiting different learning objects than other similar learners. For example, a learner may be advised to consult some unread material that other similar learners have read before attempting a particular reflection quiz. ARM has three main steps: neighborhood formation, rule generation and recommendation display. Each step is discussed in the next subsections in more detail.

Neighborhood formation. In ARM, we assume that if a learner visits particular LOs and performed well in the course, the learner had a successful learning experience. Accordingly, those LOs might be helpful to other similar learners who have not yet visited those LOs. These other similar learners build the neighborhood of a learner and are learners with similar characteristics (i.e., learning styles, prior knowledge, expertise level and performance). The purpose of the neighborhood formation step is to find such other similar learners. There were two main requirements for our algorithm to build a neighborhood: (1) the number of learners in the neighborhood of a particular learner should not be predefined but flexible and (2) the neighborhood should include the data points (learners) that are close to another. Based on the above stated requirements, we choose a neighborhood approach for finding similar learners. Such neighborhood approach does not demand the number of neighborhoods or neighbors as input a priori and can use a distance measure to place a learner only together with learners who have very similar characteristics.

To find the neighborhood, we use an algorithm that describes each learner, L_i *(I =1,…,m)* as a vector and compute similarities between learners based on the commonly used distance measure, Euclidean distance. As mentioned before, we are using different characteristics of learners including learning styles, expertise level, prior knowledge and performance. Each characteristic has a different scale of values. To ensure the equal impact of each characteristic, we normalize the data between 0 to1. Once the characteristics values are normalized, Euclidean distance is used to compute the similarity between learners based on their characteristics. Euclidean distance (L_i, L_j) is the distance between the vectors representing two learners. The formula to calculate the Euclidean distance between two learners is shown in Formula (1).

$$Euclidean\_distance\ (L_i, L_j) = \sqrt{\sum_{k=1}^{n} (L_{ik} - L_{jk})^2} \quad , \qquad (1)$$

where L_{ik} denotes the characteristic k of learner i.

In order to calculate the neighborhood of a learner, a threshold t is used as radius for the neighborhood. Accordingly, for a learner L_i, we consider every other learner L_j (j=1 … m and j != i) as a member of the neighborhood if $Euclidean\_distance\ (L_i, L_j)$ <= t. To determine a suitable value for a threshold t, we assume that two learners can be considered as similar if the difference between each characteristic is on average equal or lower than 0.25 (on a scale from 0 to 1). Accordingly, the Euclidean distance between two such learners would be 0.66. Therefore, we consider 0.66 as threshold to calculate the neighborhood.

Rule Generation. In order to generate recommendations, some data processing is needed. The learning objects visited by learners are recorded and are converted into transactions consisting of learner ID and all learning objects visited by the learner within the course. Table 1 shows an example of such transactions.

Table 1. Example of Transaction

Learner ID	Learning objects visited by the learner
1	{Content Object1, Example1}

After pre-processing, association rule mining algorithm [22] is applied to the transaction within the neighborhood to discover associations between the learning objects among similar learners. In the following, an example of rules, resulting from the association rule mining algorithm is presented:

R1 : {Content Object1, Forum1} → *{Self-Assessment Test1}*

According to R1, learning objects, Content Object1 and Forum1 are associated with Self-Assessment Test1. That mean, before performing Self-Assessment Test1, learners have visited Content Object1 and Forum1. To provide recommendations to a learner, ARM consults the association rules to check for mismatches between the learning objects visited by the current learner and the learning objects visited by the learners within the neighborhood. For example, suppose the current learner has not visited Forum1 yet and he/she is trying to attempt Self-Assessment Test1, but other similar learners in his/her neighborhood have visited Forum1 before completing Self-Assessment Test1 successfully. In such case, the recommendation to be provided (to the current learner) is to visit Forum1 before Self-Assessment Test1. Such recommendations are then passed to the recommendation display for being presented to the learner.

Recommendation Display. In this step, the personalized recommendations are displayed to the learner in an informative, precise and simple way. Recommendations include links to the recommended learning objects so that the learners can go to these learning objects easily. A learner can either click on the links or choose to close the recommendation. As and when the learner clicks on any recommended learning object, the learning object pops up and other recommendations (if any) are saved so that the learner can visit them later on. Figure 2 shows an example of a recommendation for a learner. In the example, when the learner tries to attempt Reflection Quiz1, the recommender system recommends two learning objects namely, Forum1 and Example1. The learner may choose to visit Forum1 first by clicking on the respective link. In this case, Forum1 pops up and Example1 is saved as further recommendation. When the learner tries to attempt Reflection Quiz1 again, then the other recommendation of Example1 is displayed if Example1 has not been visited by the learner already. If the learner clicks the Continue button, he/she can proceed without the recommendation with Reflection Quiz1.

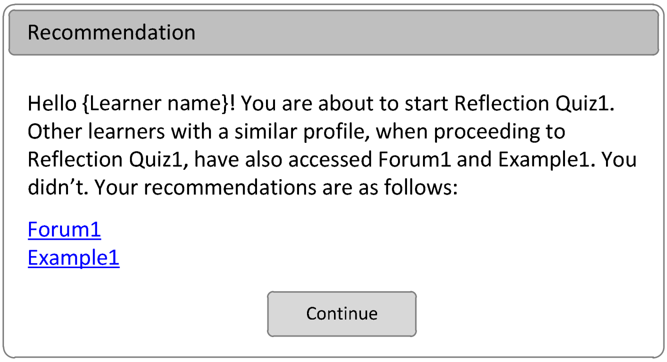

Fig. 2. Example of personalized recommendation

4 Conclusion

This paper introduces a framework to integrate a recommender system approach into learning management systems, enabling these systems to provide recommendations of learning objects to learners based on successful learning experiences of similar learners. The recommendation mechanism uses association rule mining and a neighborhood algorithm. The main contributions of the work are: First, to find similar learners, our framework does not consider ratings given by learners as done in most of the traditional recommender systems. Instead, it uses different characteristics/attributes of learners like learning style, previous knowledge, expertise levels, and performance to identify highly similar learners. Second, recommendations are provided to learners for appropriate learning objects based on what worked well for other learners with similar characteristics in similar situations. Third, in most of the previous works similar learners are found by using a clustering approach. In our work, we consciously decided against a clustering algorithm. Clustering algorithms typically aim at assigning each learner to a group/cluster. This leads to several relevant drawbacks such as the risk of creating clusters that include data points (or learners) that are actually not too close, the risk of getting different clusters when running the same clustering algorithm again, meaning that the clustering algorithm does not always group the nearest data points (or learners), or the need for a predefined number of clusters. Since our aim is to find learners who are close to a particular learner, a neighborhood approach is more accurate and free of the abovementioned drawbacks. By using such neighborhood approach, we expect to place a learner only together with learners who learn in a very similar way, and use the experience of similar learners to provide accurate recommendations. Fourth, while most other works focus on using a recommender system in a particular e-learning systems, the aim of our work is to integrate a recommender system into any LMS. LMSs are commonly used by educational institutions and by enhancing LMSs with personalized functionality to provide individual recommendations, teachers can continue using the systems that they are already using for online learning and learners are receiving additionally some personalized support. The provided recommendations can help learners to better navigate the course (by suggesting learning objects that could improve their performance within the course) as well as improve their learning performance and satisfaction. Currently, we are providing recommendations of learning objects within a course. As a future work, we will extend the framework to additionally provide recommendations from the web.

Acknowledgment. The authors are grateful to MITACS for their partial financial support through the ELEVATE program. The authors acknowledge the support of Athabasca University, NSERC, iCORE, Xerox, and the research related gift funding by Mr. A. Markin.

References

1. Szabo, M.: CMI Theory and Practice: Historical Roots of Learning Management Systems. In: Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, pp. 929–936 (2002)
2. IEEE Learning Technology Standardization Committee. Draft standard for learning object metadata (IEEE 1484.12.1-2002). New York, NY (2002)
3. Brusilovsky, P., Miller, P.: Course Delivery Systems for the Virtual University. In: Tschang, F.T., Della Senta, T. (eds.) Access to Knowledge: New Information Technologies and the Emergence of the Virtual University, pp. 167–206. Elsevier (2001)

4. Shishehci, S., Banihashem, S.Y., Zin, N.A.M., Noah, S.A.M.: Review of Personalized Recommendation Techniques for Learners in learning management system Systems. In: Proc. of the Int. Conf. on Semantic Technology and Information Retrieval, pp. 277–281. IEEE Press (2011)
5. Huang, M.J., Huang, H.S., Chen, M.Y.: Constructing a Personalized Learning Management System based on Genetic Algorithm and Case-Based Reasoning Approach. Expert Systems with Applications 33(3), 551–564 (2007)
6. Tseng, C.: Cluster-based Collaborative Filtering Recommendation Approach. Master's Thesis, National Sun Yatsen University (2003)
7. Amazon, `http://www.amazon.com/`
8. Netflix, `http://www.netflix.com/`
9. Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. Internet Computing 7(1), 76–80 (2003)
10. Capuano, N., Iannone, R., Gaeta, M., Miranda, S., Ritrovato, P., Salerno, S.: A Recommender System for Learning Goals. In: Lytras, M.D., Ruan, D., Tennyson, R.D., Ordonez De Pablos, P., García Peñalvo, F.J., Rusu, L. (eds.) WSKS 2011. CCIS, vol. 278, pp. 515–521. Springer, Heidelberg (2013)
11. Manouselis, N., Drachsler, H., Verbert, K., Duval, E.: Recommender Systems for Learning. Springer Briefs in Electrical and Computer Engineering. Springer (2012)
12. Zaïane, O.: Building a Recommender Agent for e-Learning Systems. In: Proceedings of the International Conference in Education, Auckland, New Zealand, pp. 55–59 (2002)
13. Khribi, M.K., Jemni, M., Nasraoui, O.: Automatic Recommendations for E-Learning Personalization based on Web Usage Mining Techniques and Information Retrieval. In: Proc. of the Int. Conf. on Advanced Learning Technologies, pp. 241–245. IEEE Press (2008)
14. Tang, T., McCalla, G.: Smart Recommendation for an Evolving Learning Management System: Architecture and Experiment. International Journal on Learning Management System 4(1), 105–129 (2005)
15. Tai, D.W., Wu, H., Li, P.: Effective Learning Management System Recommendation System based on Self-Organizing Maps and Association Mining. The Electronic Library 26, 329–344 (2008)
16. Kerkiri, T., Manitsaris, A., Mavridou, A.: Reputation Metadata for Recommending Personalized E-Learning Resources. In: Proceedings of the Second International Workshop on Semantic Media Adaptation and Personalization, pp. 110–115. IEEE Press (2007)
17. IEEE Learning Technology Standards Committee, `http://www.ieeeltsc.org/`
18. Yang, Q., Sun, J., Wang, J., Jin, Z.: Semantic Web-Based Personalized Recommendation System of Courses Knowledge Research. In: Proceedings of the International Conference on Intelligent Computing and Cognitive Informatics, pp. 214–217. IEEE Press (2009)
19. Dunn, R., Dunn, K., Freeley, M.E.: Practical Applications of the Research: Responding to Students' Learning Styles – Step One. Illinois State Research and Development Journal 21(1), 1–21 (1984)
20. Felder, R.M., Soloman, B.A.: Index of Learning Styles Questionnaire. NorthCarolina State University (1996), `http://www.engr.ncsu.edu/learningstyles/ilsweb.html`
21. Felder, R.M., Silverman, L.K.: Learning and Teaching Styles in Engineering Education. Engineering Education 78(7), 674–681 (1988), Preceded by a preface in 2002, `http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf`
22. Agrawal, R., Imieliński, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 207–216. ACM Press (1993)