CodERS: A Hybrid Recommender System for an E-learning System

MohammadHossein Ansari[1], Mohammad Moradi[1], Omid NikRah[2], Keyvan M. Kambakhsh[3]
[1]Young Researchers and Elite Club, Qazvin Branch, Islamic Azad University ,Qazvin, Iran
[2]Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University ,Qazvin, Iran
[3]Faculty of Computer Science and Engineering , Shahid Beheshti University, Tehran, Iran
Mha@codelearnr.com; {Moradi.c85|Omidnikrah} @gmail.com; Keyvan@codelearnr.com

***Abstract*— For years, E-learning systems are around to provide students and learners with virtual educational environments in which they have no need to others' assistance in the process of learning. From technical point of view, the major goal of such systems is to improve education and learning levels of users through leveraging the system's facilities. In this regard, it is essential for the system to be able to analyze users' functions and behaviors to prepare them with most appropriate learning materials. In other words, the core component of a working and efficient e-learning system is its recommender system. Since requirements and processes of any recommender system strongly depends on the context, users' behaviors and case-specific goals, in most of situations they should be designed exclusively. Due to this fact, in this paper we present a hybrid and context-specific recommender system (CodERS) for our interactive programming e-learning system, CodeLearnr, and provide an overview on its features including conceptual architecture, workflow and sample outputs.**

Keywords—component; Recommender system; e-learning; behavior analysis; Programming education; Data fusion

I. INTRODUCTION

To improve and develop educational approaches and with respect to the technological advancements, e-learning systems, specifically in the new millennium, have been gained a notable attention. In fact, development of such systems is a natural answer to the reasonable demand of a large number of people who want to be educated beyond the border of traditional systems in the form of self-learning, time and location independent and technology-driven platforms. To address this issue, over the recent years many academic and practical[1] works have been done, such as [1- 3].

Due to intrinsic nature of such systems and to preserve quality of learning, lack of teachers/instructors should be compensated in some way. In other words, an efficient and working e-learning platform should be designed in such a way to investigate individual needs, functions and performance of learners to provide them with learning materials. Therefore, the core component of a working and effective e-learning system is its recommendation component. Of course, since any e-learning system has its own features and requirements, managing and improving the system is a case-specific task. This led us to design a special recommender system for our system, CodeLearnr[2] [4], the first interactive programming e-learning platform for Persian-speaking users. Since the learning materials in CodeLearnr are programming courses, our proposed recommender system, CodERS, has been equipped with some platform-specific features and processes. In this paper, the architecture and details of CodERS are described and its front-end results for users as well as initial experiments are presented. To our knowledge, CodERS is the first recommender system for online programming learning platforms and the authors believe it could inspire researchers to develop next generation of such systems. The rest of the paper is organized as follows:

Related works and background of the present study are discussed in section 2 and 3 respectively. CodERS architecture and workflow are explained in next sections, 4 and 5. Finally, the preliminary experiments and the current features and benefits of the system and possible future works are presented in section 6 and 7, respectively.

II. RELATED WORKS

Due to importance of personalization of e-learning platforms, developing recommender systems for such systems has been become one of the most important research areas in the field. In this regard, some of notable works are as follows: in [5] a personalized e-learning recommender system, PLRS, has been proposed with the goal of helping students finding learning materials they need to study. In another work, authors in [6] introduced an ontology-based hybrid recommender system. The main benefit of this system is its ability to semantic search of learning materials. Another similar research conducted in [7]. The underlying idea of the work reported in [8] is to recommend learning materials to students based on the content similarity and learners' feedbacks in the form of rating. As a step further, the recommender system proposed in [9] had no need to explicit feedback and provides automatic personalization based on recent navigation history of users. Moreover, similarity between learning materials and users' preferences is another factor to make proper recommendations. Notion of leveraging web mining to build recommender agent for e-learning has been introduced in [10].

III. BACKGROUND

Regarding the importance of programming education, specifically for juveniles and young students, we have designed and implemented the first interactive programming learning platform for Persian-speaking users, called CodeLearnr[4]. Within its user-friendly environment and without any additional requirements but a web browser, users can easily study the programming courses and take their learning into practice through its live programming facilities. Moreover, the system (specifies and) notifies users of their errors and mistakes. In fact, it manages and guides the learning process as a virtual, dedicated teacher/instructor.

[1] https://www.codecademy.com/
[2] https://www.CodeLearnr.com

978-1-5090-5820-4/16/$31.00 ©2016 IEEE

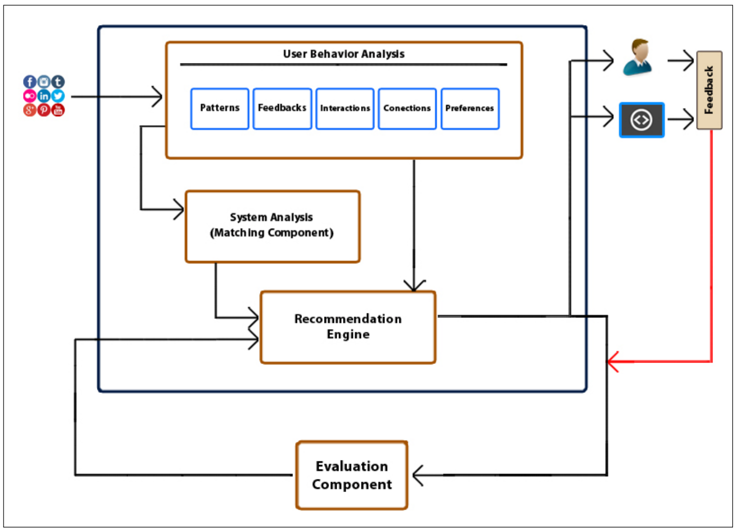

Fig.1. The conceptual architecture of CodERS

Nonetheless, this is the first step in a long journey of an e-learning system since the vision is improving education level and users' learning performance. In this regard, the CodeLearnr's ultimate goal is to simulate presence of an intelligent teacher in a virtual dedicated classroom that provides students with personalized and tailored learning materials, exercises and experiences.

To reach this achievement and equipping the system with capability of analyzing behavior and performance of learners for preparing user-specific educational materials, the core component of this e-learning system is its recommender system. Doing so, we designed Codelearnr's recommender system, CodERS, that its architecture and details will be discussed in the following sections.

IV. System Architecture

According to the intended applications and functionalities, the conceptual architecture of CodERS is illustrated in Fig.1.The system is comprised of four main components as follows:

- **User Analysis Component**: This subsystem is the most important and determining component of the system. Code learners' behaviors and activities, namely their coding patterns, feedbacks, interactions, connections and preferences, would be analyzed in this step (as in (1)) to specify their requirements and needs. In fact, such data are main sources of system for further processes and operations. Moreover, users' social media information (u_iSM) will be investigated for more accurate and personalized decisions and recommendations. To integrate these different types of data, a data fusion process will be leveraged. The anticipated results of this procedure are a list of user-specific (behavioral) patterns for further decision making (Fig. 2).
Specifically, such patterns uncover hidden relationships between different characteristics of users. For example, it may be discovered that there is a relationship between a group of connected users and their coding style and common errors.

$$BA(u_i) = AF_c\,(u_iCP, u_iF, u_iI, u_iC, u_iP, u_iSM) \quad (1)$$

- **System Analysis Component**: This component (2) that could also be named *Matching Component* is in charge of matching users' features and needs with the system's facilities and different capabilities for further decisions. In other words, it considers that for a given user (u_i), what type of recommendations should be generated by the system and how they should be presented to enhance his/her learning and performance. In other words, it specifies how the system could help a given users with its features and possibilities. The underlying algorithm for the matching process is neural network (Fig. 3).

$$SA_i(BA(u_i), S) \Rightarrow (M_1, M_2, \dots, M_n) \quad (2)$$

- **Recommendation engine**: All results of the previous steps would be aggregated in this component to generate most appropriate recommendations for users (3). These recommendations will be presented to the user both explicitly in the form of notifications and implicitly within the learning process (through adapting materials and user interface elements based on users' needs).

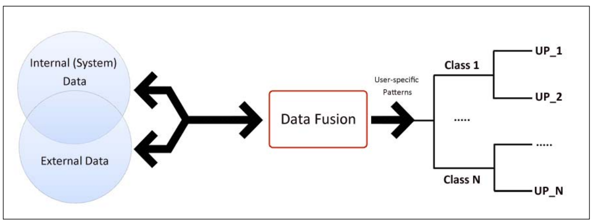

Fig.2. Schematic representation of user analysis component

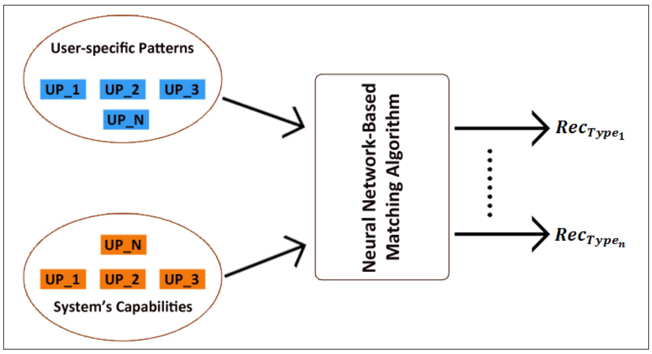

Fig.3. Workflow of system analysis (matching) component

Moreover, some managerial advices will be presented to the system for the sake of improving the current processes.

$$Rec_i\left(SA_i, BA(u_i), \sum M_i\right) \Rightarrow (R_1, R_2, \dots, R_n) \quad (3)$$

The detailed underlying process within the recommendation engine is as follows (Fig.4.):

1. First, all data that come from different sources will be integrated into a specific format to be used for decision making by the component.
2. Based on characteristics of the data, users' needs and system's capabilities, quality of recommendation(s) will be specified in this sub-step. It includes course-specific recommendations, educational resources and so on.
3. Based on the previous steps, the most appropriate algorithm (collaborative Filtering, content-based, etc) or a specific combination of several algorithms (hybrid approach) will be dynamically determined.
4. The outcome of applied algorithm(s) in step 3 will be resulted in generating user-specific/system-specific recommendation(s). In addition to the common advices to users, user interface elements will be rearranged and repositioned in an adaptive manner to improve user experience within the learning process.
5. Finally, the recommendations will be presented in the form of notifications in the user's profile. This process relies on capability of the system to provide adaptive user interface.

- **Evaluation Component**: All the recommendations and their effects on users' performance (Pe) and system maintenance (S_m) as well as acceptance rate (AR) by users will be analyzed and evaluated in this step to build up a knowledge base for the decision making processes and improving future suggestions (4).

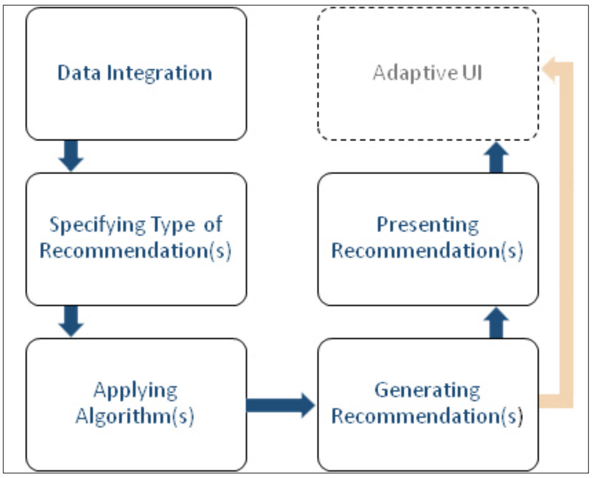

Fig.4. Internal process of recommendation engine

$$Eval\left(\sum R_i\right) = E_c\left(Pe(u_i) + S_m + AR(u_i)\right) \quad (4)$$

V. System Workflow

Principally, CodERS is a hybrid recommendation system that takes benefits of both content-based and collaborative filtering recommendation strategies. Clearly, when it comes to suggesting further learning resources, content-based approach comes in handy and when new courses or friends should be recommended, collaborative filtering method would be leveraged. Nonetheless, the central concept that CodERS shaped around is recommendation based on the user behavior/activity analysis, similar to concepts presented in [14, 15].

Input data that are subject to analysis and will be leveraged to specify users' characteristics and needs to generate recommendations are (but not limited to) followings:

- User-generated codes (patterns, errors, etc.)
- Course/step/exercise completion duration
- Users' questions and (implicit)feedbacks
- Internal and external social media connections and interaction data

The results of the process depicted in Fig.1. are of two types: recommendations for the system and users (Fig.3). The former refers to the system self-recommendations - based on activities and performance of learners- to identify drawbacks and deficiencies of educational materials, system architecture, user-experience design and so on. Moreover, students' progress records may be leveraged to redesign courses and learning materials. In addition to the explicit feedback of users in the form of questions, ratings, etc., the system implicitly infers similar information based on their usage patterns.

In addition to the special feature of self-recommendation, CodERS, based on several factors including learners' progress, efficiency and their detailed records, provides students with some recommendations such as (Fig.5):

- Further educational resources such as related books
- Specific courses and steps
- Redoing exercises
- User-specific exercises and content
- Adaptive courses in which based on the user's records, some specific steps and exercises would be presented again
- Friends (code learners with similar interests and performance within the system-wide social network)

In addition to the system-generated recommendations, it is possible for users to ask for customized recommendations through the specified form (Fig.6).

VI. Preliminary Experiment

We are currently finalizing the implementation of CodERS, however, to evaluate its efficiency we have conducted a preliminary experiment. In this limited test, 12 randomly selected users provided with four recommendations based on their courses and performances, namely recommendations on further learning materials and complementary exercises.

Users' acceptance rates of the generated recommendations are depicted in the chart presented in Fig.7. This measure could be regarded as an appropriate precision indicator of recommended items.

Moreover, to specify total satisfaction of users with the recommended items, they asked to express their views by choosing between 1 (imperfect) to 5 (perfect). The results are as in the Fig. 8.

Fig.5. Recommendations based on the user's courses and performance

Fig.6. facilities to ask for customized recommemdations

All in all, since this test was mainly based on content-based recommendations and took benefit of a limited set of users' data, the results were satisfactory and in accordance with expectations. In this regard, it is anticipated that incorporating users' social data and taking benefits of collaborating filtering, CodERS could provide users with more accurate and useful suggestions.

VII. Current Features and Future Extensions

However CodERS is yet another recommender system for e-learning platforms, its special context (namely programming education) and unique features make it an interesting contribution that could be inspirational for the future works in the field.

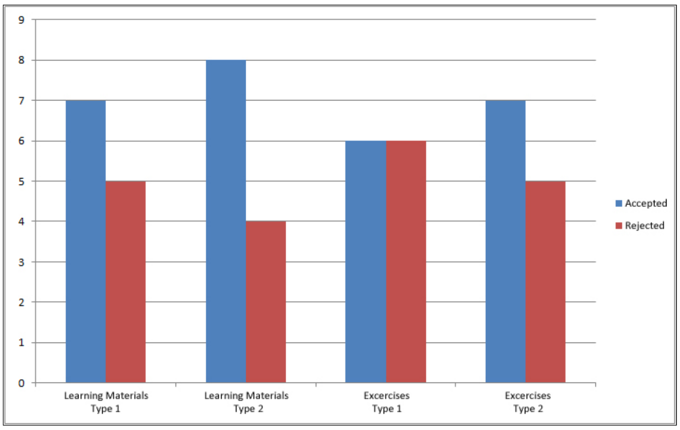

Fig.7. Acceptance rates (or Precisions) of recommended items

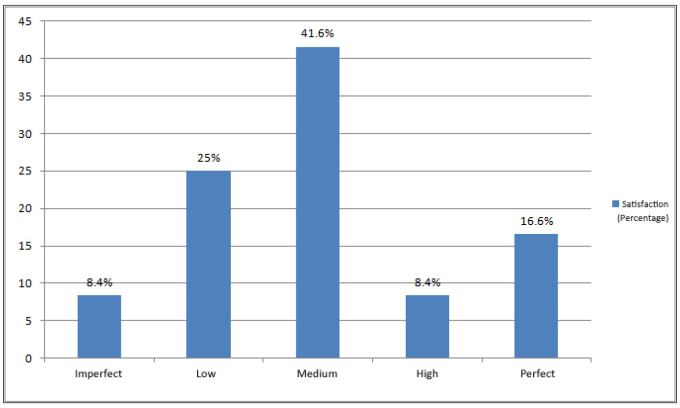

Fig.8. Users' total satisfaction with the recommended Items

To summarize its characteristics, they could be mentioned as follows:

- To our knowledge CodERS is the first recommender system for programming learning platforms
- Beside the common approaches, it works based on users behavior analysis
- User-generated codes and programs are subject to analysis for generating recommendations
- Providing self-recommendations for the system to improve itself
- Providing learners with adaptive courses based on their usage patterns
- Providing explicit and implicit recommendations
- Facilities for asking customized recommendations by the users

To make CodERS even more efficient and intelligent, some of the future plans to enhance the system are as follows:

- Leveraging process mining and machine learning techniques to extract hidden patterns from the system's log and users' behaviors
- Analyzing social media for more personalization of the system based on users' interests and connections
- Providing facilities for allowing users to build their customized educational materials and on-demand courses.

VIII. Conclusion

In this paper we have presented a context-specific recommender system, CodERS, for the first interactive programming learning platform for Persian-speaking users (CodeLearnr). The main goal of CodERS is to improve learning level and performance of students through providing them with personalized and user-specific recommendations and educational materials. The core concept behind the system is analyzing users' behavior and activities within the system to provide them with most appropriate suggestions. The unique contribution of this study, which made it a pioneer in the field, is its focus on analyzing users' generated programming codes to infer their coding (and mistake) patterns for better recommendations.

References

[1] M. J. Huang, H. S. Huang, and M. Y. Chen, "Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach," Expert Systems with Applications, vol. 33, no. 3, pp. 551-564, October 2007.

[2] N. Soonthornphisaj, E. Rojsattarat, and S. Yim-Ngam, "Smart e-learning using recommender system," In Computational intelligence, Springer Berlin Heidelberg, pp. 518-523, August 2006.

[3] C. M. Chen,"Personalized E-learning system with self-regulated learning assisted mechanisms for promoting learning performance," Expert Systems with Applications, vol. 36, no. 5, pp. 8816-8829, July 2009.

[4] M. H. Ansari, et al., " CodeLearnr: The First Interactive Programming Learning Platform for Persian-speaking Users," In Proceedings of 1st International Conference on New Research Achievements in Electrical and Computer Engineering, 2016.

[5] J. Lu, "Personalized e-learning material recommender system," In Proceedings of International conference on information technology for application (ICITA), pp. 374-379, 2004.

[6] L. Zhuhadar, O. Nasraoui, R. Wyatt, and E. Romero, "Multi-model ontology-based hybrid recommender system in e-learning domain," In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, pp. 91-95, 2009.

[7] O. C. Santos, and J. G. Boticario,"Requirements for semantic educational recommender systems in formal e-learning scenarios," Algorithms, vol.4, no. 2, pp. 131-154, July 2011.

[8] K. I. B. Ghauth, and N. A. Abdullah, "Building an e-learning recommender system using vector space model and good learners average rating," In Proceedings of Ninth IEEE International Conference on Advanced Learning Technologies, ICALT'09, pp. 194-196, 2009.

[9] M. K. Khrib, M. Jemn,and O. Nasraoui,"Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval," In Proceedings of Eighth IEEE International Conference on Advanced Learning Technologies, ICALT'08, pp. 241-245, 2008.

[10] O. R. Zaíane, "Building a recommender agent for e-learning systems," In Proceedings of International Conference on Computers in Education, pp. 55-59, 2002.

[11] T. Y. Tang, G. McCalla, "Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment," International Journal on E-Learning, vol 4, no. 1, pp. 105–129, 2005.

[12] K. I. Ghauth, and N. A. Abdullah,"Measuring learner's performance in e-learning recommender systems," Australasian Journal of Educational Technology, vol.26, no. 6, pp. 764-774, January 2010.

[13] J. E. S. U. S. Bobadilla, F. Serradilla, and A. Hernando, "Collaborative filtering adapted to recommender systems of e-learning," Knowledge-Based Systems, vol. 22, no. 4, pp. 261-265, May 2009.

[14] Z. Yan, P. Zhang, and R. H. Deng, "TruBeRepec: a trust-behavior-based reputation and recommender system for mobile applications," Personal and Ubiquitous Computing, vol. 16, no. 5, pp. 485-506, Jun 2012.

[15] A. Geyer-Schulz, M. Hahsler, A. Neumann, and A. Thede, "Behaviorbased recommender systems as value-added services for scienti.c libraries," in Statistical Data Mining & Knowledge Discovery, H. Bozdogan, Ed. Chapman & Hall/CRC, 2003, pp. 433–454.