Dept. of Computer Science and Engineering  
Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj-8100

**Online judges programming problem recommendation system using collaborative filtering**

**Abstract:** The use of online judges for programming is widespread these days, as they provide a vast collection of programming exercises that students can solve to enhance their programming skills. However, students often face the issue of information overload as they get confused while selecting the appropriate problem to solve due to the large number of problems available. The integration of existing programming online judges into e-learning systems, such as Intelligent Tutoring Systems (ITSs), is challenging because of the lack of necessary information in these systems. Therefore, this research aims to help students overcome the issue of information overload by using a collaborative filtering recommendation method. This method filters out programming problems that are suitable for a student's programming skills. To achieve this, an enriched user-problem matrix is used, which facilitates a better student role representation, resulting in more accurate recommendations by computing closer neighborhoods. A case study is conducted on a real dataset of a ***Codeforces*** to demonstrate that the proposed approach outperforms previous methods.

1. **Introduction**
2. **Introduction**

Programming online judges are software tools designed to offer a vast collection of programming exercises to be solved by users, mainly students. These tools aim to automate the evaluation and compilation processes of users' solutions to the proposed programming problems. Programming online judges have gained significant popularity in recent years and have been successfully used in two different scenarios - as a training tool for students participating in ACM-ICPC-like programming contests, and for systematic practice of programming skills in computer science colleges Traditional programming online judges present a sequential list of programming problems that users can choose and solve at their personalized pace. The widespread acceptance of these tools worldwide has resulted in an increase in the amount of information associated with them.

1. **Motivation**

Collaborative filtering research was pioneered by memory-based techniques, and they are still prevalent due to their simplicity and effectiveness. This paper introduces a memory-based collaborative filtering recommendation method that helps students combat information overload in POJs by personalizing the problems they should attempt based on their current skill level, avoiding failures and frustrating experiences

1. **Aims and Objectives**

Our approach aims to strengthen research and development of programming online judges by offering a customized recommendation technique that is supported by collaborative filtering methods. The approach is based on a user-problem matrix that incorporates data about the Codeforces problems difficulty and category. The use of this enriched matrix leads to better student profiling, making it easier to compute closer neighborhoods and provide more accurate recommendations.

1. **Significance of this research**

For the first time, collaborative filtering methods were used in the Codeforces online judge problems set. Codeforces is a very famous online judge for problem solvers.

1. **Related Works**

Programming online judges are systems that offer a list of programming problems for users to solve. The typical interaction of a user in a online judges involves the following steps:

1. The user selects a problem and attempts to solve it;
2. The user uploads their solution to the POJ;
3. The POJ evaluates the solution using predefined input-output data.

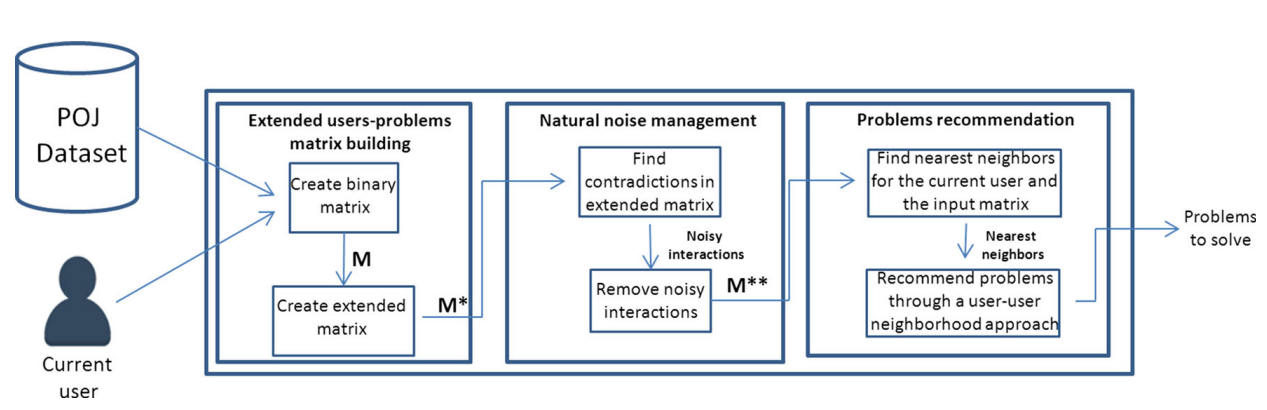
If the output matches the expected result, the solution is deemed correct and the problem is considered solved. If not, the user can either propose an alternative solution for the same problem or choose a different problem. The term programming online judges was first used **by Kurnia et al**. as a system for supporting the automatic evaluation of source codes developed by students for problem resolution. **Mooshak** is a widely used system that supports programming contest management and can also be used as an online judge. **EduJudge,** presented by **Verdu et al.,** is an effective platform that integrates Moodle and QUESTOURnament, a competitive e-learning environment. **OJPOT**, presented by Wang et al., is a teaching approach that combines online judges and practice-oriented programming teaching to develop students' practical abilities through programming practices. Other works, although not employing the term online judge, focus on the automatic evaluation of programming problem solutions.

1. **Methodology/Proposed Research Plan**
   1. **Data Description**

The dataset consists of two primary sources of data, a user-problem matrix and a dataset containing problem categories and difficulty levels. The user-problem matrix is a sparse matrix that represents user interactions with problems, where each row corresponds to a unique user, and each column represents a unique problem in the dataset. The dataset containing problem categories and difficulty levels provides information about the characteristics of each problem, where each row represents a unique problem, and columns represent features such as problem category and difficulty level. The dataset can be used for various analyses and machine learning tasks such as building recommendation systems, identifying problem categories that require additional support, or predicting user performance based on problem characteristics. The dataset is of reasonable size and structured format, making it easy to import into various software tools.

Overall, this dataset provides valuable insights into user performance and problem characteristics, which can be useful for educational and research purposes.

* 1. **Proposed Model**



Categorized problem matrix interactions

* 1. **Timeline**

|  |  |  |
| --- | --- | --- |
| **Serial** | **Topics** | **Time** |
| 01 | Study about recommendation systems | 3 weeks |
| 02 | Generate idea about my research | 2 weeks |
| 03 | Dataset collections | 4 weeks |
| 04 | Data preprocessing | 2 weeks |
| 05 | Noise management | 1 weeks |
| 06 | Implementations | 4 weeks |
| 07 | Report Generations | 2 weeks |

1. **Expected Outcomes/Results**

Table: F1 measurement of k=40



1. **Conclusion**

This study presents a recommendation approach using the Codeforces online judges dataset to suggest problems for users to solve. The proposed approach is based on collaborative filtering techniques and involves the building of an extended user-problem matrix, preprocessing of the matrix to manage natural noise, and problem recommendation. The findings of this research demonstrate the effectiveness of the proposed approach in improving the accuracy of problem recommendations, particularly in comparison to related previous research. Moreover, this study contributes to the field of recommender systems in the e-learning domain, particularly in investigating the use of the Codeforces online judges dataset as a source of data for developing recommendation approaches. Overall, this research provides valuable insights into the development of recommendation approaches for e-learning systems, highlighting the importance of data preprocessing techniques in improving the accuracy of problem recommendations.

1. **References**
2. Fantozzi, P. and Laura, L., 2021. Collaborative recommendations in online judges using autoencoder neural networks. In *Distributed Computing and Artificial Intelligence, 17th International Conference* (pp. 113-123). Springer International Publishing.
3. Lumauag, R., Sison, A. and Medina, R.P., 2019. An enhanced memory-based collaborative filtering algorithm based on user similarity for recommender systems. *Int. Journal of Recent Technology and Engineering*, *7*, pp.778-782.
4. Yera, R., Rodríguez, R.M., Castro, J. and Martínez, L., 2018. A Recommender System for Supporting Students in Programming Online Judges. In *Smart Education and e-Learning 2017 4* (pp. 215-224). Springer International Publishing.
5. Zhao, W.X., Zhang, W., He, Y., Xie, X. and Wen, J.R., 2018. Automatically learning topics and difficulty levels of problems in online judge systems. *ACM Transactions on Information Systems (TOIS)*, *36*(3), pp.1-33.
6. Wang, M., Han, W. and Chen, W., 2021. MetaOJ: A massive distributed online judge system. *Tsinghua Science and Technology*, *26*(4), pp.548-557.
7. Ahmed, D.M., Abdulazeez, A.M., Zeebaree, D.Q. and Ahmed, F.Y., 2021, June. Predicting university's students performance based on machine learning techniques. In *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)* (pp. 276-281). IEEE.
8. Caro-Martinez, M. and Jimenez-Diaz, G., 2017. Similar users or similar items? Comparing similarity-based approaches for recommender systems in online judges. In *Case-Based Reasoning Research and Development: 25th International Conference, ICCBR 2017, Trondheim, Norway, June 26-28, 2017, Proceedings 25* (pp. 92-107). Springer International Publishing.
9. Amatriain X, Pujol JM (2015) Data mining methods for recommender systems. In: Recommender Systems Handbook. Springer, pp 227–262