On combining collaborative filtering and using an optimal path in a Concept Graph*

A. Fuster-López[1] and I. Nowak[2]

[1]Computer Science, Univ. Málaga, alejandrofuster@uma.es
[2]Mechanical Engineering and Production, HAW Hamburg, ivo.nowak@haw-hamburg.de

April 11, 2024

Abstract

The aim of the iMath project is to develop a recommender system where a student is recommended a next question or literature in a learning platform MathE. This paper describes the systems developed based on collaborative filtering Python application Surprise by the partners of the project based in Universidad de Málaga and its combination with using concept maps by the Professional University in Hamburg. It describes further development towards hybridization of recommenders and the inclusion of concept maps.

1 Introduction

The aim of the iMath project is to personalize an e-learning path for an individual student in the environment of an e-learning system on mathematics called the MathE portal, cf. [1, 5]. The idea is that a teacher already has a learning path in mind based on concept to be taught. However, the best path does not necessarily coincide with the one a teacher would choose. After studying various recommendation methods with our partners at UMA, our team decided to develop an automated recommender system (RS) based on collaborative filtering which was implemented by the UMA team. In such an environment, starting for example from a list of 40 exercises on a given mathematical topic (e.g., matrices and determinants), the objective is to automatically detect to which learners a given student is sufficiently similar to be able to recommend which are the next 5 exercises from a list of most favorably answered exercises among those learners similar to the student. Our question is to combine such an approach

*Funded by iMath project 2021-1-PT01-KA220-HED-000023288.

with a concept graph, such that in the end, the student captures all identified concepts.

Since it is quite impractical to have learners rate exam questions as they would rate films or songs, explicit techniques are ill-suited to provide the rating needed for a recommender system to perform collaborative filtering effectively. Instead, relying on implicit feedback, such as user interactions or behaviours, could be a more viable and practical approach in this context. In the first iteration of our recommender system development and testing done by the UMA, we employed the seamless 5-rate question marking scheme seen in Table 1 which gave us a primer framework to work with. However, in this second iteration, we advanced to an improved scheme with additional difficulty levels provided and curated by the IPB (*Instituto Politécnico de Bragança*), making a crucial evolution in our approach, since the additional data should provide us with a better collaborative filtering. Thus, we would like to extend our appreciation and gratitude to the IPB for their contribution in enhancing this recommender system.

Table 1: Seamless 5-rate question marking scheme

$5 \equiv$ Right Answer for a Difficult Question
$4 \equiv$ Right Answer for a Basic Question
$3 \equiv$ Wrong Answer for a Difficult Question
$2 \equiv$ Wrong Answer for a Basic Question
$1 \equiv$ "I Don't Know" Answer for a Question
$0 \equiv$ Question has not been answered

The provided levels by IPB differentiate between 5 levels of difficulty. Therefore, we experienced a substantial transformation in our approach. One of the critical implications of this evolution lies in how it impacts the fitting of data for the collaborative filtering algorithm. In the initial phase with a 5-rate question marking scheme, our data was relatively straightforward, allowing for a basic alignment of user feedback with question correctness. However, as we progressed to the scheme with ten distinct difficulty levels, our rating became richer and more granular. This granularity has a profound impact on data fitting and enhances our understanding of it.

The marking scheme, as calculated for the difficulty levels provided by IPB presented in Table 2, offers valuable insights. It is essential to recognise that the highest ratings in this scheme correspond to correctly answered, highly challenging questions, while the lower ratings correspond to incorrectly answered, relatively easier questions. Importantly, the rating values range from 0 to 5, mirroring the original Table 1. This consistency allows for straightforward comparisons between the two rating systems, facilitating a one-to-one analysis

Furthermore, this multi-level marking scheme extends beyond the current level differentiation. It can be effortlessly extended to accommodate various clustering or categorizations of the questions. This flexibility makes the system versatile and adaptable to the specific needs and goals of the educational platform. This can be easily achieved by modifying the lambda function for the rating with the needed parameters for the different categorizations.

Table 2: Question marking scheme adapted to the IPB levels

5	$\equiv$ Right Answer for a Question of Level 5
4.6	$\equiv$ Right Answer for a Question of Level 4
4.2	$\equiv$ Right Answer for a Question of Level 3
3.8	$\equiv$ Right Answer for a Question of Level 2
3.4	$\equiv$ Right Answer for a Question of Level 1
3	$\equiv$ Wrong Answer for a Question of Level 5
2.6	$\equiv$ Wrong Answer for a Question of Level 4
2.2	$\equiv$ Wrong Answer for a Question of Level 3
1.8	$\equiv$ Wrong Answer for a Question of Level 2
1.4	$\equiv$ Wrong Answer for a Question of Level 1
1	$\equiv$ "I Don't Know" Answer for a Question
0	$\equiv$ Question has not been answered

After studying several types of RSs, we chose to use the so-called collaborative filtering (CFs) methodology for its ability to provide predictions about unknown learner/exercise pairs, due to its speed of prototyping and combinability with other types of RSs which other partners of the project are currently working on. Collaborative filtering is able to include graph-based techniques that take advantage of information based on concept maps (CMs). Moreover, we can use easy to use standard metrics (e.g., RMSE and/or MAE) that enable measuring objectively whether a new algorithmic improvement has brought a benefit. Considering different ways of combining RSs to obtain hybrid RSs, we have opted for a non-invasive technique in which, instead of incorporating characteristics of one RS to another, a combination is made on the basis of results obtained by both. In this way, we can measure the progress obtained from combining two RSs even if the metric used by another partner (e.g., based on machine learning classification) is not RMSE and/or MAE. To test the feasibility of our approach, we implemented an algorithm which alternatively recommends with the two most promising CFs among those based on matrix factorization and on clustering, according to tests with randomly generated data.

Currently we are working with the partners in Málaga (Spain) on a framework for combing CF (or other RS) methods with additional information offered by concept maps. In particular, a novel algorithm for generating learning paths based on both error rates and a concept map was developed and implemented. The concept map may be automatically constructed or provided by a teacher. A massive test with virtual answers in the second semester of 2023 looks promising as a prelude to test the entire system with real students, preferably in the first semester of 2024.

The rest of this manuscript describes the applied methodology and refers to literature on which the algorithms are based.

2 Rating scheme, notation and evaluation metric

learner	Q1	Q2	Q3	Q4
Wilma		4		1
Juan	3		0	3
Pablo	0	2	4	
John	1		2	3
Filipe		4	0	
Ivo		0	1	0
Ana	5	4	4	5
…	…	…		

Figure 1: Example of a rating matrix

To translate the recommender system to the usual terminology used in collaborative filtering, see [6], Users are Students, Items are Questions either about a specific subtopic as matrices and determinants, or about any of the subtopics available in the MathE portal. A historical database of ratings is assumed to be available, either that from the very beginning or one constructed during the first months of 2023 by using initially completely random choice among the questions in a given subtopic as recommendation device, i.e. Algorithm 0. The row structure with the fields expected from that historical database is

u_{15}	i_{23}	`ans`	`alg`	`StartTime`	`EndTime`

.

Its interpretation is that student number 15 was recommended question number 23 at `StartTime` by algorithm number `alg` and answered `ans` at `EndTime`. Based on the collaborative filtering recommenders, our RS provides the next five questions as a recommendation offered one after another. The RS quality is measured by standard metrics, i.e. RMSE, MAE, etc.

As students give an answer to the database question, we have a transparent way to rate the question for the student which assumes that the student likes most her recent *dynamic* correct answers. Notice that the characteristic difficult or easy question may change in time. Distinguishing more difficulty levels, will also extend the range of the rating.

We started to study the behavior of the recommender using artificial answer generation generated assuming that teachers should have assigned a specific rating to each of the 5 possible answers. However, we left this idea, because teachers who add questions to the MathE database do not assign a *static* valuation. Thus, as existing historical data does not include the previous static valuation of each choice of all the questions, we have used the coding of Table 1 which is sketched with an example in Figure 1.

One of the challenges we run into is that the historical database does not consider ratings 0 and 1 to be different. Nor its information has been saved. In order to obtain more information in terms of collaborative filtering, we suggested a 0-1 rating, where 0 represents "I Don't Know" and be more explicit to rate a wrong answer as 2 or 3 depending on the difficulty of the question.

Using the terminology introduced by Hug in [3], we define

- R: the set of all ratings.
- R_{train}, R_{test}, and $\hat{R}$ denote the training set, the test set, and the set of predicted ratings, respectively,
- U: the set of users (students) with indices (alias) u and v,
- I : the set of items (questions) with indices i and j,
- U_i : subset of users that have rated item i,
- $U_{ij} \equiv U_i \cap U_j$: subset of users that have rated both items i and j,
- I_u: subset of items rated by user u,
- $I_{uv} \equiv I_u \cap I_v$: subset of items rated by both users u and v,
- r_{ui} : the rating obtained by user u for item i (groundtruth),
- $\hat{r}_{ui}$: the estimated rating of user u for item i,
- b_{ui} : the baseline rating of user u for item i,
- μ : the mean of all ratings,
- μ_u : the mean of ratings obtained by user u,
- μ_i : the mean ratings for item i,
- σ_u : the standard deviation of the rating of user u,
- σ_i : the standard deviation of the rating of item i,
- $N_i^k(u)$: the k nearest neighbour (according to a similarity metric) of user u having rated item i,
- $N_u^k(i)$: the k nearest neighbour (according to a similarity metric) of item i that are rated by user u.

To compare the relative accuracy of the approaches, we adopt frequently used standards. In particular, we performed a cross-validation RMSE-based procedure consisting of averaging RMSE calculated after splitting L times the set $R = R_{\text{train}} + R_{\text{test}}$ into two disjoint sets TR and TS. One then computes prediction $\hat{r}_{ui}$ using only R_{train} for all (u, i) pairs in test set TS and finally compares it to the observed ratings r_{ui} available at R_{test}. So

$$RMSE_\ell = \sqrt{\frac{1}{|TS|} \sum_{(u,i)\in TS} (r_{ui} - \hat{r}_{ui})^2}, \qquad RMSE = \frac{1}{L} \sum_{\ell=1}^{L} RMSE_\ell.$$

This computation is implemented in the cross validation procedure `cross_validate` in Surprise [3] with `rmse` as the `measures` parameter.

3 Algorithms

The implementations of collaborative filtering of the University of Málaga are based on SVD^{++} (matrix-factorization) amd KNN with Means (Clustering) as described in their technical note. One of the main questions was to derive a Non-invasive hybrid CF combining algorithms.

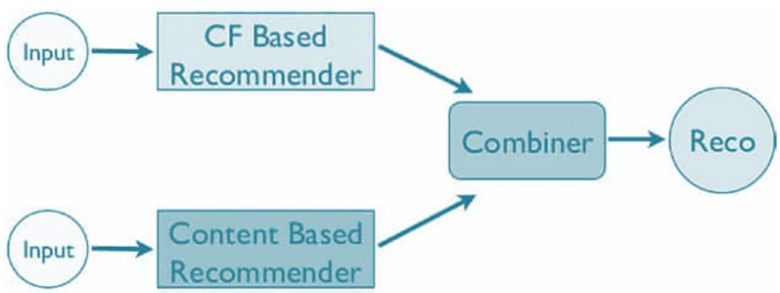


Figure 2: Non-invasive Hybrid RS, [4, p. 9]

As pointed out in [4, pp. 8–9], to overcome the drawbacks of two different types of RSs (e.g., a content-based RS and a CF) or two RSs of the same type (e.g., two CFs), one may design a hybrid RS. There are two ways to construct such an hybrid RS:

- An **invasive hybrid RS** incorporates the algorithmic features of one RS into the other.
- A **non-invasive hybrid RS** uses the *results* of one RS to be combined by the *results* obtained by the other. Hence you can combine the results obtained by different RSs although they are using different metrics.

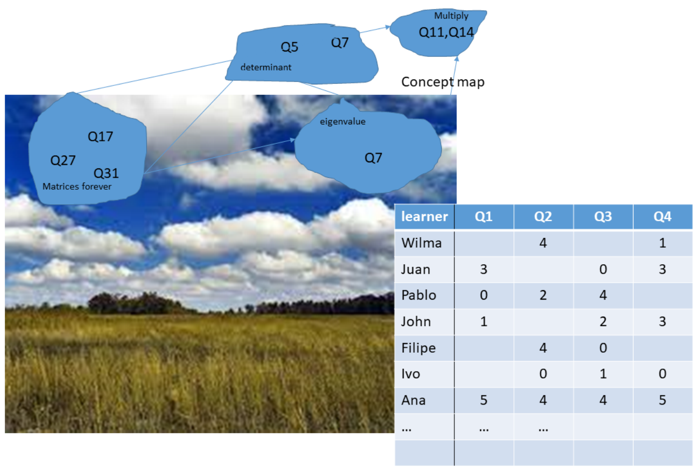
We have chosen a non-invasive technique (Algorithm 3) in which, instead of incorporating characteristics of one RS to another, a combination is made on the basis of results obtained by both after alternatively recommending with Algorithm 1 and Algorithm 2. In this way, we can measure the progress obtained from combining two RSs even if the metric used by another partner (e.g., based on machine learning classification) is not RMSE and/or MAE.

To test the feasibility of our approach, we simply append the results obtained after recommending with Algorithm 2 to the results obtained after recommending with Algorithm 1 to check that the cross-validated RMSE for the whole rating matrix has improved for any (or both) of the combined algorithms.

After analysing the RMSE results obtained from the data collected during the testing phase with real students at the University of Málaga, it became evident that the KNN Basic algorithm is the best choice for the non-invasive hybrid recommendation system we implemented. The cross-validated RMSE scores demonstrated notable improvements when KNN Basic was used in combination with SVD^{++} Consequently, a decision was made to replace the KNN Means algorithm with KNN Basic for the development of the recommendation system.

3.1 Learning-path-oriented question selection methods

Note that choosing the best next question is a greedy user-oriented approach similar to dynamic programming focusing only on the bottom layer, cf. figure 3. However, the recommended best next question does not guarantee following the best learning path.



learner	Q1	Q2	Q3	Q4
Wilma		4		1
Juan	3		0	3
Pablo	0	2	4	
John	1		2	3
Filipe		4	0	
Ivo		0	1	0
Ana	5	4	4	5
...	...	...		

Figure 3: Two layer framework

A long-term item-oriented approach is needed to take the whole learning path into account. Hence we must work in the top layer to deal with relationships among questions (item-item similarities in CF) to be able to take learning indicators into account when doing the recommendation. In this way, we can modify Algorithm 2 (or even Algorithm 1) that does not exploit these explicit relationship between items, by simply restricting the choice among the items to those selected by an algorithm on an higher layer. To achieve it, two ideas have been investigated, both using a directed graph in which *nodes are sets of questions* and arcs are precedence relations between them:

- The sets of questions are not disjoint and concepts are treated as keywords. Here the idea is to restrict the greedy approach to those questions in a current concept, with unweighted arcs defined via a CSV file. Finding the best learning path can be formulated as a constrained shortest path problem. A modified Dijkstra algorithm [2] for solving this problem was implemented and tested using data of the MathE database.

- The sets of questions are disjoint, i.e. a partition/clustering of I is used. Concepts are not equivalent to keywords and weighted arcs are precomputed by combining ℓ concept maps, with an unique weight for an arc after that. Here the idea is a non-standard greedy approach to only recommend within the current concept.

4 Summary

This note described the concepts of collaborative filtering (CF) applied in an e-learning environment where the next question to deal with is recommended by a system. One of the main questions is to combine several CF in a non-invasive way. We argue that it is very feasible to lay a recommended path among concepts over a standard recommender based on collaborative filtering. Basically, we limit the next question to be recommended to follow a best path along a concept graph using the method of Dijkstra.

References

[1] Consortium 2018-1-PT01-KA203-047361. MathE, a toolkit for students for self-evaluation of their knowledge on selected math topics. `https://mathe.pixel-online.org/`.

[2] Edsger Wybe Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269—-271, 1959.

[3] Nicolas Hug. Surprise: A Python library for recommender systems. *Journal of Open Source Software*, 5(52):2174, 2020.

[4] Akshay R. Kulkarni, Adarsha Shivananda, Anoosh Kulkarni, and V. Adithya Krishnan. *Applied Recommender Systems with Python.* APress, 2023.

[5] Maria F. Pacheco, Ana I. Pereira, and Florbela Fernandes. MathE—improve mathematical skills in higher education. In M. Della Ventura, M. Vavalis, and D. Sampson, editors, *Proc. 8th ACM ICEIT Conference*, pages 173–176. ACM Press, March 2019.

[6] F. Ricci, L. Rokach, B. Shapira, and P. Kantor, editors. *Recommender Systems Handbook*. Springer, 1st edition, 2011.