

A Recommendation System for Online Courses

David Estrela¹, Sérgio Batista¹, Diogo Martinho², and Goreti Marreiros^{2(✉)}

¹ CIICESI, Center for Research and Innovation in Business Sciences and Information Systems, ESTG, School of Management and Technology, Polytechnic of Porto, Porto, Portugal
{8100550,8110167}@estg.ipp.pt

² GECAD, Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Institute of Engineering, Polytechnic of Porto, Porto, Portugal
{diepm,mgt}@isep.ipp.pt

Abstract. Nowadays there are many courses available for students, and sometimes it is hard for a student to perceive information related to those courses and decide which course to take. This work aims to build a system to suggest online courses to users based on their profile and the similarity with other users. For this work three techniques were used to extract the information and suggest online courses: Content Based, Collaborative filtering and Hybrid. By combining these three techniques the system can offer more accurate recommendations and only considers the interests of each user. Thus, users will not feel tired while perceiving information of their interest and will keep engaged and interested to use the system.

Keywords: Recommendation systems · Content based · Collaborative filtering and hybrid

1 Introduction

Recommendation systems are becoming very popular and sometimes people do not even realize they exist. Facebook for example, recommends friends based on our mutual friends, work, location, and other criteria. Another example, are the E-commerce websites that always offer many recommendations just to lure the client into buying more products. For instance, Amazon, provides recommendation based on the user similarity. If a substantial number of clients purchases a product “A” and “B”, and another client is interested in buying product “A”, the product “B” will likely be recommended to him. This type of recommendation has increased substantially the amazon sales [1]. Amazon also offers recommendations based on the users search history. When a user searches for a certain product, amazon recommends similar products based on the product’s description.

People seek for recommendation on daily bases for many purposes. One could be to ask a friend for some recommendation on a book or to ask the teacher about which framework would be most adequate to develop a recommendation system. [1]. A recommendation system does not exactly need to be an informatics program. For example, it could be a travel agency which provides suggestions based on the feedback of other

tourists [2]. Recommendations are almost everywhere: in bookstores, libraries or even video stores [3]. Through the understanding of a customer's preference it is possible to create a suggestion list with products that fulfil the user's preferences totally or partially.

Nevertheless, this just shows us that we are unsure about what we want and nowadays through the internet, we have at our disposal a huge amount of information, but unfortunately most of it is useless to us. Therefore, this information should be filtered. But what is the best way to filter information? Is it enough to add the traditional filters such as "filter by categories", "filter by distance", "filter by grades"? Maybe not. These filters could be helpful, but we could improve the user's experience much more by getting to know the real interests of the user. Through the user's profile, it is already possible to apply the traditional filters automatically, and provide the user more directed information (Content based filtering). And through the implementation of likes and dislikes it is possible to get to know the user rapidly and compare users. Through this approach, it is possible to calculate user similarity, which allows us to filter similar users.

For this work, we focused our study in the user online course recommendation. E-learning is sometimes considered to be more effective than traditional educational environments and offers many advantages such as flexibility, accessibility, or adjustability [4]. However, this type of learning also offers some disadvantages such as the lack of social interaction, which may affect its effectiveness. It is important to keep the student engaged in the online learning process otherwise it is very likely that he may lose focus and consequentially decrease his performance. Therefore, it is important to motivate the student to pursue online learning activities and for that systems that can recommend courses that meet the student preferences must be developed. For instance, in our country (Portugal) we have a centralized platform (DGES¹) where all students must apply for collage, and we realized that the students have some difficulties choosing/filtering the large number of available courses. This is one of the main reasons behind the idea of implementing an engine/prototype on how to use recommendations on e-learning platforms. We strongly believe that this kind of implementation would have a large impact on the students in the moment of making a decision.

In this work, we developed an engine, which creates a list of recommendations based on the user's profile, user similarity and the mix of this two (hybrid). To demonstrate this, we created an online platform, and a data bank with some fake courses. We took our inspiration on the Facebook like/dislike system and the amazon recommendation system component-based. Although we did not use a large data collection of courses and users, we could affirm that the developed recommendation system is very useful and can be adjusted to each user profile. From the moment, the user provides some knowledge about him the engine can start filtering courses based their preferences (content based). It also allows the users with just a few likes and dislikes immediately filter many non-interesting courses.

¹ <http://www.dges.gov.pt/pt>.

2 Recommendation Techniques

There are many recommendation techniques like: Knowledge based, Fuzzy set-based, social network-based, trust-based, context awareness-based and group recommendation [1]. For this work in particular, we will be focusing on the traditional techniques: Content Based, Collaborative Filtering, and Hybrid.

2.1 Content Based

Content based (CB) recommends products based on similar products that the user likes. This can be achieved by comparing the products attributes with the user’s preferences [1, 5]. For this work, we compare the courses categories with the user’s profile, and check what are the categories in common. Through this, it is possible to generate recommendation using the traditional methods of data extraction such as *cosine similarity measure* [1].

2.2 Collaborative Filtering-Based

The Collaborative filtering recommendation technique helps the user to take a decision based on other users’ opinions, which share similar tastes [6]. Collaborative filtering can be divided in two parts, user-based and item-based [1].

In the user-based approach, the user receives products, which similar users like. On the other hand, in the item-based approach the user receives recommendation based on the similarity between the items he likes. For this project, we chose the user-based approach, we will be giving recommendation based on the other users likes.

To calculate the similarity this can be achieved using: *Pearson correlation-based similarity*, *constrained Pearson correlation based-similarity*, *cosine-based similarity* or even *adjusted cosine-based measure* [1]. Through this calculation only users that have at least one product in common are considered.

In Fig. 1, it is possible to see a collaborative filtering recommendation, in which user A and B have similar preferences. Unfortunately, the collaborative filtering may not be very accurate, since it is being used on a small data sample. This would be almost equivalent to asking a friend with similar testes for a recommendation but the friend’s recommendation will be limited to his knowledge and opinion. The more variety of different people are compared the better the results.

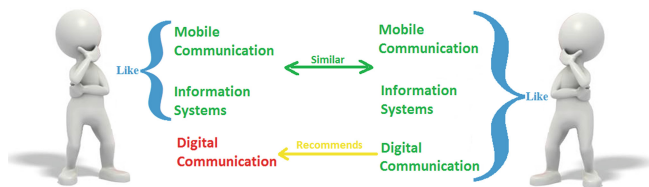


Fig. 1. Collaborative filtering recommendation

2.3 Hybrid

The hybrid technique was created to achieve a better performance and overcome the disadvantages of the traditional recommendation techniques. This technique combines the better part of two or more techniques. According to [1, 7] there are seven basic mechanisms of hybridization in the construction of a recommendation system, which are: weighted, mixed, switching, feature combination, feature augmentation, cascade and meta-level. The practice most used in existing hybrid recommendation system, is by combining Collaborative filtering techniques with others, to avoid cold-start, sparseness and problems of scalability [1].

In this case, we will be combining Content based and Collaborative filtering to achieve the most accurate possible results.

3 E-Learning

As shown in the last section, recommendation systems can be used in many distinct areas, and in this case, we will be focusing on the e-learning recommendation system.

E-learning recommendation systems, have become quite popular in the past 15 years. This kind of recommendation system are usually focused on helping students choosing online courses bases on the curricular unite or lecture content which are in the student's interests. In the past years, there have been created some E-learning recommendation systems. In 2002, Osmar R. Zaíane [8] proposed a new approach to develop a software agent base system using data mining techniques, such as, association rule mining for the construction of a model representation of the user's online behaviour. Using this model representation, it is possible to suggest what is most relevant for the user. The generated suggestions can help the user in his online search, since the information is presented to the user more rapidly through the shortcut that the recommendation systems provide [9]. There is also an article that proposed a personalized e-learning material recommender system (PLRS) [10]. In this system, every time there is something new added to an already existing product, the user's information must be collected so it can be analysed through computational analysis modal. Afterwards, through matching rules, the user's requirements are matched with the contents and the new recommendations are generated providing the user a new list of relevant content.

4 Algorithms

In this section the Content based and Collective filtering techniques will be explained using mathematical formulas, these formulas can later be easily implemented in a wide variety of computer languages. In this case, we choose PHP programming language, a common computer language used on the server-side.

4.1 Content Based

This algorithm is straightforward, since we only need to compare two lists of attributes and check with the ones that are similar. In this case, we have on one hand the user’s preferences, which are a list of categories that the user is interested, and on the other hand we have the course, which has its categories. For example, in Fig. 2, the course mobile communication, has the following categories; informatics, electronics, and engineering. If the user has preference for these three categories, this will be most likely an interesting course to recommend.

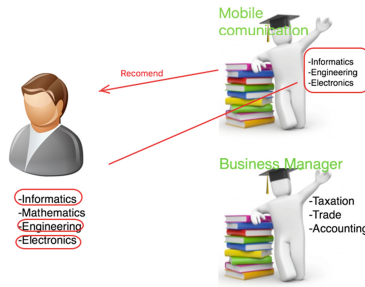


Fig. 2. Content based recommendation

The probability of a user liking a course is the number of courses they have in common divided by the total number of categories the course has:

$$P(U, C) = \frac{(UserCategories \cap CourseCategories)}{\sum CourseCategories}$$

Where:

- P(U,C) = Probability of a user (U) liking a course(C)
- UserCategories = Set of all categories preferred by the user
- CoursesCategories = Set of all available courses categories

This will return a value between 0 and 1, being 0 a no match and 1 a full match.

4.2 Collaborative Filtering-Based

Collaborative filtering is more complex than content based technique. First it is necessary to calculate the similarity of the user before calculating the probability of liking the course. To do so, two sections are considered. In the first section, it is measured the similarity value between two users and in the second section it is measured the probability of a user liking a certain course.

a- Calculate the similarity value To identify the similarity level two different situations should be considered as users are not only similar if they like the same course, but also if they dislike the same course. Therefore, the system must take in consideration opposite

tastes which represents if a user likes a course and another user dislikes it. This can be measured using the following formula:

$$S(U_1, U_2) = \frac{(|L_1 \cap L_2| + |D_1 \cap D_2| - |L_1 \cap D_2| - |L_2 \cap D_1|)}{|L_1 \cup L_2 \cup D_1 \cup D_2|}$$

Where:

- S(U1, U2) = Similarity between two users
- L1 = Likes of the user
- L2 = Likes of another user
- D1 = Dislikes of the user
- D2 = Dislikes of another user

In the first step, it is calculated the likes in common between both users plus the number of dislikes in common. In the second step, it is subtracted the opposites tastes. For that, it is measured the intersection between the courses which the user likes but the other user does not like with the intersection of courses the other user likes with the courses the user dislikes, and that is subtracted to the result of the first step. Finally, the denominator corresponds to the total number off all courses that either user liked and disliked.

b- Probability of a user liking a certain course After measuring the similarity between users, it is possible to calculate the probability of a user liking a certain course based on the similarity with other users. For that the following formula is considered:

$$P(U, C) = \frac{(Z_L - Z_D)}{(M_L + M_D)}$$

$$P(U, C) = \frac{(Z_L - Z_D)}{(M_L + M_D)}$$

Where:

- P(U, C) = probability of a user (U) liking a certain course (C)
- ZL = sum of the similarity with all the user that like this course
- ZD = sum of the similarity with all the users that dislike this course
- ML = number of user which like this course
- MD = number of user which dislike this course

In this step, we use the formula identified in the first section to calculate the similarity of the user with all the users who liked a certain course, subtract from the similarity of users who disliked the same course and divide it by the number of users who liked and disliked it. Figure 3 exemplifies collaborative filtering recommendation algorithm where Users A and B share the same interest for mobile communication course but User B also has preference on Cyber-security course. This course will then be recommended to User A.

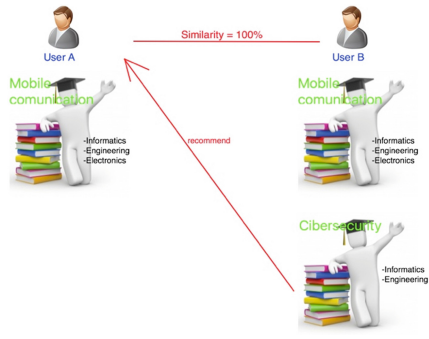


Fig. 3. Collaborative filtering recommendation

4.3 The Hybrid Technique

The hybrid technique will be the most accurate since it will be the intersection of the Content based and Collaborative filtering technique result.

$$P(U, C) = \text{Content based} \cap \text{Collaborative filtering}$$

5 Courses Recommendation System

As prove of concept an online platform was developed using the Laravel Framework (at the time Laravel 5.2) and a MySQL database. This platform follows the typical MVC architecture as can be seen in Fig. 4.

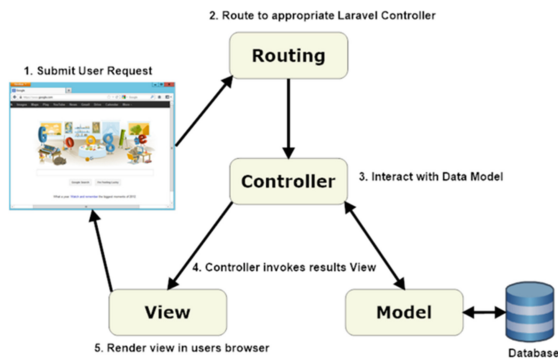


Fig. 4. Platform architecture

The database has a small sample of courses to allow users to rate the courses. The database will also be used to save user information, preferences, and the courses liked and disliked.

The developed platform allows the user to perform different actions. First the user must register himself, and fill up his user profile, which is basically providing us the categories he may have be interested in. After the registration, the system can provide some recommendations to the user based on the categories that where signaled in the registration phase, using the Content based techniques. Although the recommended courses may not be the most interesting to the user, they will most likely be familiar to the user, already offering him a good experience while using the website.

After the user likes and dislikes available courses (Fig. 5) it will be possible to use the collaborative filtering technique, and it will be possible to give the user recommendations based on others user likes and dislikes, and use the hybrid technique. Using the hybrid technique, the recommendation will be more targeted to the user, since two main factors are being compared: the user’s preferences and the other similar user’s feedback.

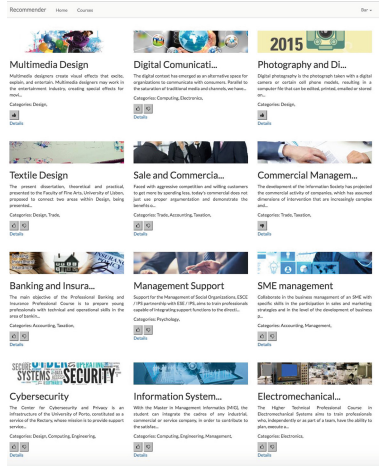


Fig. 5. Available courses

Giving the user this kind of recommendation (Fig. 6) is very important, not only to help him find new courses, but also to keep a fast, clean, short, and highly probable target list of courses that the user may be interested in.

Recommendations may also have constraints, for instance in this case, there is the constraint of age. Each course has a minimal and max age and there are a couple of possibilities to solve this issue; removing the course before generating the recommendation, or removing it after the recommendation is generated. To solve this issue, the course is removed after the recommendation is generated, since it is important to calculate similarities containing as much data as possible.

Either way this platform can be easily adapted, and new constraints/filters can be added. Despite the platform being implemented for online courses it can be easily adapted to other purpose. For this, it is only necessary to add data to the database, and reformulate the conceptual model, since the attributes that will be analyzed have already been defined. Once this is set, the engine returns a list of recommendation from the most probable to the less probable object.

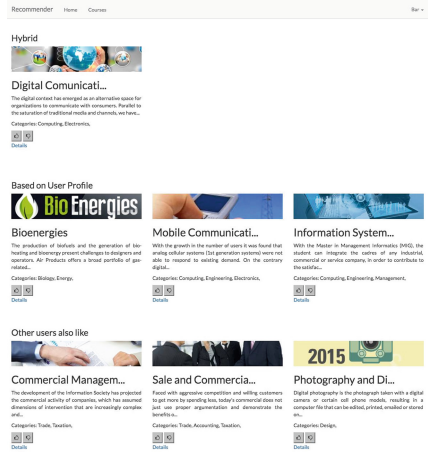


Fig. 6. Recommendation

6 Conclusion

Nowadays, recommendation systems are very useful and on high demand. There is a lot of data on the web, and without filters or recommendation it is hard to find what we are looking for. But recommendation systems are not just helpers for the user to find what he is looking for, but also provide him the best possible user experience so that he keeps using website or even show him new data that he might not know it existed and still be interested.

We want to show in this work that it is possible to help students filter faster and more efficient the wide range of existing courses and only recommend courses according to their preferences. Also, as we can see in this article, the recommendation algorithms are quite simple and can be applied to almost anything in any programming language. One drawback of the developed system is related to the difficulty to provide recommendations right after the user registers since there is still no information about the user. To solve this problem, in the registration phase, we kindly ask the user to provide some useful inputs about him so that the system can start providing recommendations. Another drawback is related to the size of the dataset, since the one used in this work was quite small, so the recommendation precision was not as accurate as expected.

As future work the implemented recommendation system platform will provide recommendations not only based on user configurations but also based on information related to institutions which may be on high demand, offering two ways of recommendation. We also intend to conduct a case study with real participants in order to measure the usability of the system and identify areas of improvement to make the system more appealing to the user (always based on each user profile).

With this work, we hope to see an increase of use-cases where recommendation systems are implemented, since our engine can be customized to a different dataset, making the implementation of a recommendation system very versatile.

References

1. Lu, J., Wu, D., Mao, M., et al.: Recommender system application developments: a survey. *Decis. Support Syst.* **74**, 12–32 (2015)
2. Cao, L., Luo, J., Gallagher, A., et al.: A worldwide tourism recommendation system based on geotagged web photos Kodak Research Laboratories, Eastman Kodak Company Dept. Computer Science, University of Illinois at Urbana-Champaign. In: ICASSP, pp. 2274–2277 (2010)
3. Davidson, J., Liebold, B., Liu, J., Nandy, P.: The YouTube video recommendation system (2010)
4. Richardson, J., Swan, K.: Examining social presence in online courses in relation to students' perceived learning and satisfaction (2003)
5. Pazzani, Michael J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *The Adaptive Web*. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-72079-9_10](https://doi.org/10.1007/978-3-540-72079-9_10)
6. Deshpande, M., Karypis, G.: Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.* **22**, 143–177 (2004)
7. Burke, R.: Hybrid web recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *The Adaptive Web*. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-72079-9_12](https://doi.org/10.1007/978-3-540-72079-9_12)
8. Zaïane, O.: Building a recommender agent for e-learning systems. In: 2002 Proceedings of Computers in Education (2002)
9. Ridwan, M.: Building a Recommendation Engine: An Algorithm Tutorial | Toptal. <https://www.toptal.com/algorithms/predicting-likes-inside-a-simple-recommendation-engine>
10. Lu, J.: Personalized e-learning material recommender system. In: International Conference on Information Technology (2004)