A Hybrid Multi-Criteria approach using a Genetic Algorithm for Recommending Courses to University Students

Aurora Esteban
University of Cordoba
Dept. of Computer Science
14071, Cordoba, Spain
i32estoa@uco.es

Amelia Zafra
University of Cordoba
Dept. of Computer Science
14071, Cordoba, Spain
azafra@uco.es

Cristóbal Romero
University of Cordoba
Dept. of Computer Science
14071, Cordoba, Spain
cromero@uco.es

ABSTRACT
This paper describes a multiple criteria approach based on a hybrid method of Collaborative Filtering (CF) and Content-Based Filtering (CBF) for discovering the most relevant criteria which could affect the elective course recommendation for university students. In order to determine which factors are the most important, it is proposed a genetic algorithm which automatically discovers the importance of the different criteria assigning weights to each one of them. We have carried out an in-depth study using a real data set with more than 1700 ratings of Computer Science graduates at University of Cordoba. We have used different proposals and different weights for each criterion in order to discover what is the combination of multiple criteria which provides better results.

Keywords
Educational recommender system, Course recommendation; Hybrid Multi-Criteria Approach; Genetic Algorithm

1. INTRODUCTION
Course recommendation is nowadays an interesting and increasing research line. Specifically, course recommendation for university studies can be viewed as an important educational data mining task [13]. This is a important problem because university studies normally provide a number of elective courses which students have to choose to complete their studies. This decision may not be trivial for students, which usually don't have enough information and get overwhelmed by the amount of available options. Recommender Systems (RS) appear as essential tools capable of helping students choosing relevant elective courses in their curriculum according to different criteria such as their individual ratings, preferences, interests, needs, performance, etc [6]. Although there are some studies which work with hybrid RS approaches [2, 9] and multiple criteria approaches [10, 16], these works are fairly and are not focused on studying the influence of the different factors in the recommendation process. This work presents a preliminary study to determine which are the most relevant criteria to provide better course recommendations for university students. These criteria include both information that describes the students (such as their ratings, their grades and their branch) and information that describes the courses (such as their competences, their theoretical and practical contents, the professors that teach it and their subject area). In order to determine which factors are the most important to achieve better course recommendations, a force brute search and a Genetic Algorithm (GA) are proposed. GA automatically discovers the importance of the different criteria assigning weights to each one of them. Then, these weights are incorporated to the recommendation process in order to make a final suggestion to students. In order to study the advantages and limitations of using different criteria, a real dataset which includes information from the Computer Science degree at University of Cordoba is used.

The rest of this paper is organized as follows. An overview of related work is specified in Section 2. The proposed methodology is presented in Section 3. The description of the experimental study is described in Section 4. Finally, conclusions and future work are presented in Section 5.

2. RELATED WORK
In the past few years, RSs have been thoroughly applied to course recommendation using multiple criteria. One of the first applications of multi-criteria matrix factorization for course rating predictions is explored in [15]. Later, Vialardi et al. [16] proposed multi-criteria techniques for predicting students' grades as a classification problem and Parameswaran et al. [12] explored the application of restrictions to recommendations using multiple criteria. Also, other techniques can be found in course recommendation, for instance, ontology-based approaches [5, 18], neural networks [7] or bio-inspired algorithms with proposals such as ant-colony optimization [14] and artificial immune systems [2]. Most of them based only in students' grades. From other perspective, the study of the importance of the specific moment in which the courses are taken has been studied based on students' grades using Markov chains [8] as well as applying multiple criteria [17]. More recently, both the competences provided to students and their relevance in their recommendation [4, 1] and the application of semantic analysis [11] has been adressed.

In conclusion, even though several techniques have been developed for course recommendation, most of them are mainly focused on the students' performance and do not use further criteria. Even when some other criteria are used, a study to determine each criterion influence on the quality of recommendations is not carried out. In this paper, we propose a multi-criteria approach for discovering the most relevant criteria which could affect the course recommendation. Our approach combines student information (known as Collaborative Filtering, CF) with domain-specific information (known as Content-Based Filtering, CBF).

3. PROPOSED METHODOLOGY

This section describes the proposed methodology (Figure 1). First, a description and analysis of data set is presented. Then, the recommendation approaches and the criteria used in each one of them are detailed. Finally, the evaluation methodology is addressed.

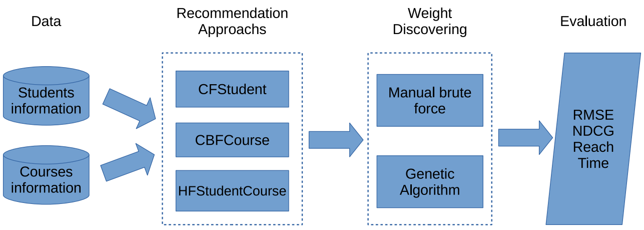

Figure 1: Methodology overview.

3.1 Data description and preparation

This work has been developed using real information gathered from the degree of Computer Science at University of Cordoba, Spain. This includes information about students and courses.

3.1.1 Student information

Student information was obtained by means of surveys which students filled in their last academic year. The factors obtained for each student are represented in the following way (see Figure 2):

- A rating of the overall students' satisfaction for each course. It is a integer value from 0 to 5 if the course is taken or it is empty otherwise.
- The grade obtained by students on each course. It is a decimal value in the range $[0, 10]$ if the course is taken or an empty value otherwise.
- The branch selected by students for specializing in a particular computer science area. Concretely, Computer Science degree offers three branches: Computation, Computer Engineering or Software Engineering. The chosen of the student will be represented as a numeric identifier (from 1 to 3).

In total, more than 1700 ratings along with their corresponding grades were obtained for the 63 courses included in Computer Science degree in University of Cordoba, Spain. The data was gathered over a period of two years (2016-2017).

To avoid global effects in the grades and ratings *subtractive normalization* [15] is applied. This normalization subtracts a combination of the student and course mean to the original value.

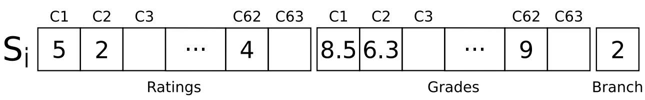

Figure 2: Student information.

3.1.2 Course information

Course information was obtained from the University official degree web page[1]. The factors selected for each course are represented in the following way (see Figure 3):

- The professors involved in the course, represented as a vector with an index for each professor in the degree. Its value is 1 if the professor is involved in this course or 0 otherwise.
- The competences or skills that the course provides, represented as a vector with an index for each competence in the degree. Its value is 1 if it is provided by the course or 0 otherwise.
- The subject area to which the course belongs, represented as a numeric identifier. Eight subject areas are considered in the degree (integer value from 1 to 8).
- The contents of the course, represented as a frequency vector of keywords obtained by text mining/preprocessing the theoretical and practical contenst of the course.

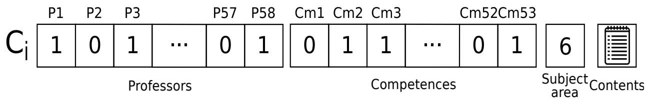

Figure 3: Course information.

3.2 Recommendation Approaches

Three different recommendation approaches are approposed to evaluate the influence of students and courses criteria.

3.2.1 Collaborative Filtering using student information - CFStudent

This proposal follows a CF approach where each student is represented using different factors, such as, the ratings vector, the grades vector and the branch. For the courses not taken by a student, the estimated preferences are obtained based on the neighborhood built using a similarity function.

For each pair of students, i and j, the similarity measure designed considers on one hand the ratings ($R_{i,j}$) and the grades ($G_{i,j}$). These similarities are calculated using metrics like Pearson or Spearman correlation coefficients and euclidean or taxicab distances. On the other hand, it is considered the branch similarity ($B_{i,j}$). This similarity is computed considering whether it is equal or not. All these measures are mapped into the $[0, 1]$ interval and the final similarity measure is computed as a parametric linear combination of the three factors:

$$D_{U_{i,j}} = \alpha \cdot R_{i,j} + \beta \cdot G_{i,j} + \gamma \cdot B_{i,j} \qquad (1)$$
$$where\ \alpha + \beta + \gamma = 1$$

The significance of each criterion can be studied according to the weight (α, β or γ) assigned to each criterion. Finally, the final preference for student i and course j, $U_{i,j}$, is calculated using the parametrized similarity measure (equation 1).

[1] `http://www.uco.es/eps/node/619`

3.2.2 Content-Based Filtering using course information - CBFCourse

This proposal follows a CBF approach where each course is represented as a series of features, such as, the subject area, the contents, the professors and the competences. In this approach, the course recommendations for a student are based on the estimated ratings of the most similar courses to those that they have already taken.

For each pair of courses, i and j, the similarity measure is designed attending to the following criteria: their professors ($P_{i,j}$), their competences ($Cm_{i,j}$) and their respective subject area ($S_{i,j}$). These similarities are computed considering whether they are shared or not. Also, it is considered a semantic analysis based on their contents ($Cn_{i,j}$). All measures are mapped into the $[0, 1]$ interval. The final similarity measure is computed as a parametric linear combination of these four factors:

$$\begin{aligned} D_{C_{i,j}} =& \alpha \cdot P_{i,j} + \beta \cdot Cm_{i,j} + \gamma \cdot S_{i,j} + \delta \cdot Cn_{i,j} \\ & where\ \alpha + \beta + \gamma + \delta = 1 \end{aligned} \tag{2}$$

The significance of each criterion can be studied according to the weight (α, β, γ or δ) assigned to each factor (equation 2). To compute similarities based on professors and competences, a boolean data based approach is followed. Thus, similarity metrics like Jaccard index or the log-likelihood function can be used.

Similarity based on course contents is stored as keywords obtained by preprocessing the theoretical and practical contents described in the course official guide. Therefore, semantic similarity is applied to each pair of courses in the following manner:

1. First, the documents are indexed: a custom text parser has been implemented based on the language (in our case, Spanish) and it is used a set of stop words adapted to the domain. As a result, for each document, a list of tokens is obtained along with their frequency as well as the number of times that each one appear in the document.

2. For each pair of courses, i, j, a set B is created as the union of the tokens of both courses. For each course, a vector $\vec{i}$ or $\vec{j}$ is built with as many elements as there are in B, represented as n. This vector contains the frequency of each token. Finally, each vector is normalized using the l_1 norm, thus it is obtained the relative frequencies to each pair of courses.

3. Cosine similarity is applied to both frequency vectors in order to integrate the course content criterion into the similarity measure between courses.

$$cos(\theta) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\| \cdot \|\vec{j}\|} = \frac{\sum_{k=1}^{n} i_k j_k}{\sqrt{\sum_{k=1}^{n} i_k^2} \sqrt{\sum_{k=1}^{n} j_k^2}} \tag{3}$$

Finally, the final preference for student i and course j, $C_{i,j}$, is calculated using the parametrized similarity measure (equation 3).

3.2.3 Hybrid Filtering using student and course information - HFStudentCourse

To avoid some of the problems of CF and CBF systems, a hybrid approach is proposed. The course preference estimation for each student and course is obtained using a linear aggregation of the estimated preference based on student information described in section 3.2.1 and the estimated preference based on course information described in section 3.2.2. Both estimations are decimal numbers in range from 1 to 5, so they are combined with certain weights α and β to provide a final preference estimation also in this range. Hence, for the student i and the course j, the preference estimations according to CFStudent ($U_{i,j}$) and to CBFCourse ($C_{i,j}$) are combined into a final estimation ($p_{i,j}$):

$$\begin{aligned} p_{i,j} = \alpha \cdot U_{i,j} + \beta \cdot C_{i,j} \\ where\ \alpha + \beta = 1 \end{aligned} \tag{4}$$

This hybrid approach implies two different configuration levels. A first level where student and course information are used separately to obtain two preference estimations. Then, a second one where it is configured the relevance of each criterion in the final recommendation.

3.3 Weights selection

Two different ways to select the weights have been used in order to configure each recommendation approach.

3.3.1 Exhaustive search

A brute-force search or exhaustive search has been used to find the best weights. This method consists on systematically enumerating all possible weight configurations and checking which configuration obtains the best results. In our case the different weights studied have been considered as decimal numbers between 0 and 1 with increases of 0.1. This type of search has been used for the CFStudent and CBFCourse approaches due to the fact that they do not have a very high number of weight combinations.

3.3.2 Genetic Algorithm

A GA has been also used to automatically discover the best weights. This has only been used for the HFStudentCourse approach due to the larger number of parameters and, therefore, more potential configurations. Its purpose is to find the optimal weights of the different criteria concerning student and course information, as well as the weights of the final linear aggregation to obtain the final preference estimation. The more relevant factors achieve higher weights and the less relevant ones, the lowest values. The main components of the used GA algorithm are:

- The chromosome is defined with integer values to represent the weight of each factor. The integer value of each gene is ranged from 0 to 10 and it would represent to the percentage in the range of $[0, 1]$. A total of 9 weights have to be assigned in this approach, three weights assigned to student information, four weights assigned to course information, and finally, two weights to determine the relevance in the final estimation considering CFStudent and CBFCourse approaches.

  The previous study of exhaustive search allows assigning restrictions to assign specific weights to particular criterion to reduce the search space. Thus, three different parameters are optimized deducing the rest of the problem restrictions.

- The individual fitness function is the Root-Mean-Squared Error (RMSE) of the recommendation when using the weight configuration given by the chromosome.
- The genetic operators are single point crossover and a random mutation which changes the value of one gene in a possible value in the fixed range.
- Parent selection is done by binary tournament.

3.4 Evaluation Metrics

There are several standpoints from which a RS performance can be evaluated [3]. In this proposal four metrics have been selected attending to accuracy, relevance or capability of making recommendations.

3.4.1 Root-Mean-Squared Error

The Root-Mean-Squared Error (RMSE) is used to measure the accuracy of the recommendations. This measure is suitable for the prediction of ratings and it tends to penalize larger errors more severely than other metrics. If $p_{i,j}$ is the predicted rating for student i over course j, and $v_{i,j}$ is the true rating and $K = \{(i, j)\}$ is the set of hidden student-course ratings, then the RMSE whose purpose is to minimize is defined as:

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in K}(p_{i,j} - v_{i,j})^2}{\#K}} \tag{5}$$

3.4.2 Normalized Discount Cumulative Gain

Attending to Information Retrieval (IR), normalized Discount Cumulative Gain (nDCG) is used as measure of ranking quality.

$$nDCG = \frac{DCG}{IDCG} \tag{6}$$

DCG at a particular rank position p, if rel_i is the graded relevance of the result at position i, is defined as:

$$DCG = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \tag{7}$$

Normalization is given by the division by the Ideal DCG at position p (IDCG).

3.4.3 Reach

CF is based on similarities between students. Depending on the criteria used, some outlier users exist for which no satisfactory similarities are found, and so no recommendation can be made for these users. This behavior will be measured by the reach of the RS whose purpose is to maximize. If $K = \{(i, j)\}$ is the set of hidden student-course ratings and $p_{i,j}$ is the predicted rating, reach is defined as:

$$Reach = \frac{\#K - \sum_{(i,j)\in K} p_{i,j}}{\#K} \ \forall\ p_{i,j} = \emptyset \tag{8}$$

3.4.4 Time

The execution time of each approach is also important. The mean execution time is analyzed once each model has been learned. It is calculated the time that each approach takes on building the recommendation ranking for a user. It is important to mention that our testing platform is a personal computer with Ubuntu 16.04 64-bit as operative system, a Intel Core i5-3317U processor and 12 GiB RAM memory, and our recommender runs under the Java Virtual Machine.

4. EXPERIMENTAL WORK

We have carried out two experimental studies. Firstly, we show the criteria weight optimization and then the comparative study between the different approaches developed. As mentioned in section 3.1, the dataset used comes from real ratings and grades gathered from students of University of Cordoba.

The different RS approaches have been implemented using Apache Mahout[2] and the GA has been developed using the JCLEC library [3].

It is important to notice that in order to guarantee a greater robustness in the results and so they can be generalized to an independent data set, a 10-fold cross validation has been used. We have stratified students' data according to the volume of received ratings on each course [3]. In essence, a portion of ratings from each student will be taken away to train the RSs with the remaining ratings. Then, data are divided into ten partitions, and each partition in turn is used as a test set. In this way, the obtained results in the different evaluation measures represent the average values of the test data set for each fold considered. The advantages of the cross-validation approach are to allow the use of more data in ranking algorithms, and to take into account the effect of training set variation.

4.1 Criteria Weight optimization

The main objective of this first experimental study is to find the optimal weights for each criterion used in the proposed RSs. Thus, it is evaluated the influence of the weights in the course recommendation.

Firstly, an initial experimental study is carried out to configure some common parameters, such as, the similarity metrics, where the Jaccard index and the log-likelihood function have been evaluated for categorical values, and the Pearson correlation and the euclidean and taxicab distances have been evaluated for numerical values. Also, neighborhood size has been evaluated with the values of 5, 10 and 15 in the case of CFStudent and HFStudentCourse. The final selected configuration according to this study is shown in Table 1. This configuration of common parameters will be used by our three RS approaches.

Next, the weight optimization of each criterion used in CFStudent and CBFCourse approaches is carried out by means of exhaustive search. Figure 4 shows the evolution of the average RMSE and its standard deviation for the CFStudent approach, varying the weight assigned to the ratings

[2] `https://mahout.apache.org/`
[3] `http://jclec.sourceforge.net/`

Table 1: Similarity measure and neighborhood size.

Similarity by ratings	
CFStudent	Euclidean distance
HFStudentCourse	Euclidean distance
Similarity by grades	
CFStudent	Taxicab distance
HFStudentCourse	Euclidean distance
Similarity by professors	
CBFCourse	Log-likelihood function
HFStudentCourse	Log-likelihood function
Similarity by competences	
CBFCourse	Jaccard index
HFStudentCourse	Jaccard index
Neighborhood size	
CFStudent	10
HFStudentCourse	15

and grades criteria, maintaining fixed and with 0.1 value the weight for branch factor. According to these values, it can be affirmed that ratings criterion is considered more relevant than grades criterion. Thus, higher weights for the ratings factor provide better recommendations (lower RMSE values). However, if only the ratings criterion is used (assigning a weight of 1.0 and 0.0 for the other criteria), it can be appreciated that the RMSE value is worse than when using the rest of criteria with lower values. Concretely, the best weight configuration is shown in Table 2. In this manner, although with lower relevance, it is also important to consider these criteria (grade and branch) in order to improve the results.

In the case of the CBFCourse approach, Figure 5 shows the RMSE evolution, attending to its average and its standard deviation, varying the weights of content and professor criteria (considered the two factors more representative in this approach) and maintaining fixed and with minimum values (that is, 0.1 value) the weight for competences and subject area factors.

The results demonstrate that the lowest RMSE values are obtained when both factors use averaged weights. Specifically, the best configuration gives a lower weight to the competences and subject area factors. Then, the content factor is also representative but its weight is slightly lower than the weight assigned to the professor criterion. The best configuration is shown in Table 2.

Finally, in the case of HFStudentCourse, because of the increase in complexity, nine different factors have to be optimized, the weights have been estimated using the GA proposed whose main parameters are population size: 100, number of generations: 500, mutation probability: 0.2 and crossover probability: 0.9. For this approach, the Figure 6 shows the evolution of the best weight configuration obtained by the GA in different generations showing the RMSE mean values and the obtained weights of the most relevant factors in the two hybridized proposals. Note that there are some secondary criteria whose weights aren't reflected in the graph since they were pre-fixed. Concretely, the branch criterion in CFStudent approach with a specific weight of 0.1, and subject area and competences with a weight of 0.1 for each one of them in CBFCourse approach. For the best configuration obtained in the last generation, the weights are not exactly the same values than the other approaches separately, but the tendency is similar: the ratings criterion obtains higher weight values than other criteria of student information and the professor obtains slightly higher weight values with respect to content criterion. Moreover, the weights to determine the importance that should be given to the results of CFStudent approach and CBFCourse approach for combining them and obtaining a final recommendation show that the best combination is obtained by maintaining a balance between both criteria. In our case, the best configuration has a weight of 0.6 for CFStudent approach, 0.4 for CBFCourse approach and the rest of weights shown in Table 2.

.

Table 2: The best weight configurations.

Criterion	CFStudent	CBFCourse	HF[1]
Ratings	0.8	–	0.6
Grades	0.1	–	0.3
Branch	0.1	–	0.1
Professors	–	0.4	0.5
Subject area	–	0.1	0.1
Competences	–	0.1	0.1
Content	–	0.4	0.3
CFStudent	–	–	0.6
CBFCourse	–	–	0.4

[1]HFStudentCourse

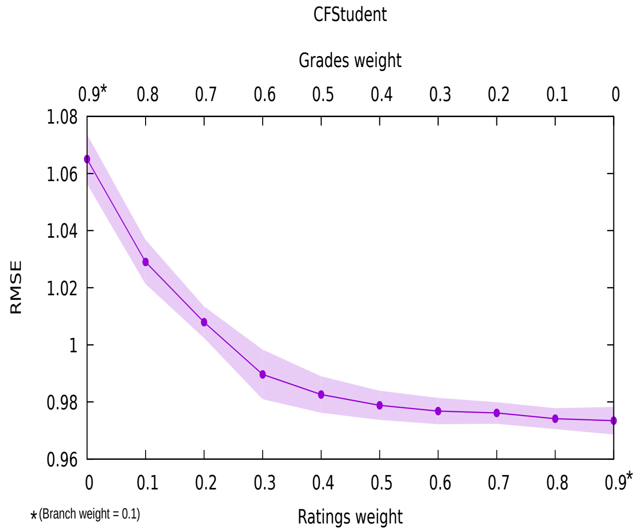

Figure 4: Weighted criteria of CFStudent approach.

Starting obtaining the best configuration for each approach, the following conclusions can be obtained:

- The weight assigned to each criterion indicates that the most important criterion for student information is the ratings. In the case of course information, course contents and professors' criteria take the lead.
- The similarity measures for ratings and grades based on distance predominate over the ones based on lin-

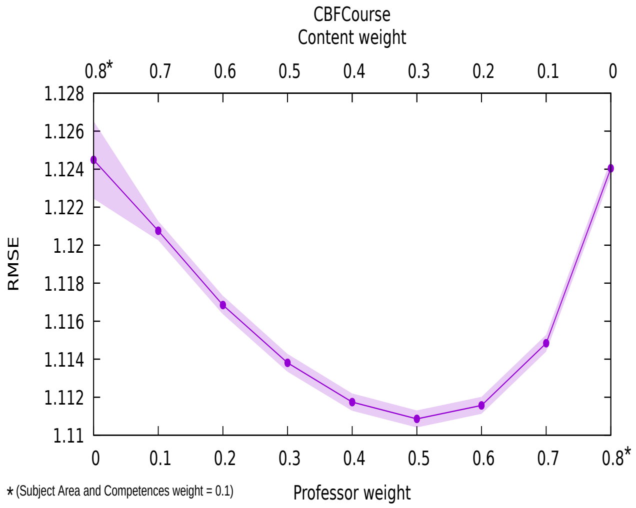

Figure 5: Weighted criteria of CBFCourse approach.

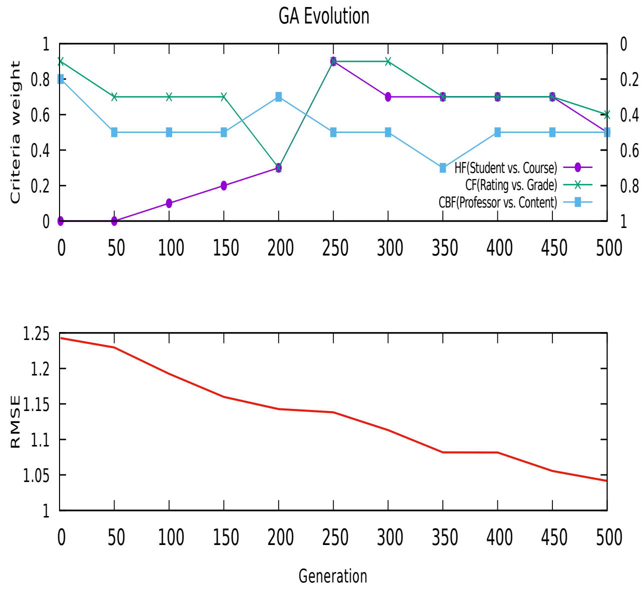

Figure 6: Weighted criteria of HFStudentCourse approach.

ear relationships. Moreover, the optimal neighborhood size grows with the number of criteria used.

- The best weight configurations for CFStudent and CBFCourse are not exactly the same considering the proposals separately or combined in hybrid approach, but the tendency is maintained. Moreover, the hybrid approach assigns a balanced weight to both proposals to obtain the final recommendation. Thus, both approaches are considered necessary to obtain the best recommendations.

4.2 Comparison of the different approaches

This second experimental study compares the results obtained by the best configurations of the previous approaches. We have used an estimation of the ratings (RMSE) as well as the others of the evaluation measures (nDCG, reach and execution time) described in section 3.4.

Table 3: Comparative evaluation between RS.

	RMSE	nDCG	Reach	Time
CFStudent	0.96628	0.7980	96.48%	1.53s
CBFCourse	1.11187	0.2768	99.36%	1.81s
HFStudentCourse	1.04150	0.8955	100%	2.05s

As we can see in the results shown in Table 3 for the RMSE, a better score is obtained when more information about the student and less about the course is used. Nonetheless, course information provides certain advantages, such as increasing the number of ratings capable of estimating (*reach*) or a more diverse set of solutions (*nDCG*), which can translate into a better proficiency in making relevant recommendations. As expected, as the amount of information considered is increased, the time taken in finding the recommendations for a student is also increased. It is then concluded that, regarding RMSE optimization, the best approach consists in using just the student information, improving as multiple criteria based on it are introduced, although explicit ratings still have the most weight. However, this approach has important flaws, as it is the capability of obtaining ratings for all users, because of outlier students for whom it is difficult to find an appropriate enough neighborhood. This shortfall is overcome when information about courses is introduced. It is practically guaranteed that similarities between courses will be found, so the reach score increases significantly.

5. CONCLUSIONS

In this paper several proposals based on CF, CBF and hybrid RS approaches combining multiple criteria have been proposed for the task of elective courses recommendation in university studies. The results confirm that the overall rating that a student gives to a course is the most reliable information source, but when it is complemented with other criteria about the own student or the course then the estimation accuracy can improve it. This work opens a promising line of research geared towards both data enhancement, by applying the RS to a larger volume of students and majors and study transferability, and broadening the used models beyond CF. The application of a GA to search for optimal configurations also has potential, especially on the modeling of chromosomes capable of containing information apart from the weights of the criteria. As future work, we want to evaluate weights to all criteria (including the criteria that we have pre-fixed). Moreover, other parameters such as, size of neighbour and similarity metrics also could be optimized. Finally, it is also important to indicate that our proposed approach could be also applied to other related educational domains such as recommendation of massive open online courses (MOOCs) with only adapting the used factors.

6. ACKNOWLEDGMENTS

Authors gratefully acknowledge the financial subsidy provided by Spanish Ministry of Science and Technology TIN2017-83445-P.

7. REFERENCES

[1] B. Bakhshinategh, G. Spanakis, O. Zaiane, and S. ElAtia. A course recommender system based on graduating attributes, January 2017.

[2] P.-C. Chang, C.-H. Lin, and M.-H. Chen. A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems. *Algorithms*, 9(3), 2016.

[3] A. Gunawardana and G. Shani. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10:2935–2962, 2009.

[4] J.-w. Han, J.-c. Jo, H.-s. Ji, and H.-s. Lim. A collaborative recommender system for learning courses considering the relevance of a learner's learning skills. *Cluster Computing-The Journal of Networks Software Tools and Applications*, 19(4):2273–2284, 2016.

[5] C.-Y. Huang, R.-C. Chen, L.-S. Chen, and Ieee. Course-recommender system based on ontology. In *Proceedings of 2013 International Conference on Machine Learning and Cybernetics*, pages 1168–1173, 2013.

[6] T.-N. Huynh-Lv, N. Huu-Hoa, and T.-N. Nguyen. Methods for building course recommendation systems. In M. LeNguyen, L. S. Vinh, L. T. Bui, V. G. Nguyen, Y. S. Ong, and K. Hirata, editors, *Eighth International Conference on Knowledge and Systems Engineering*, pages 163–168, 2016.

[7] A. A. Kardan, H. Sadeghi, S. S. Ghidary, and M. R. F. Sani. Prediction of student course selection in online higher education institutes using neural network. In *Computers & Education*, volume 65, pages 1–11, 2013.

[8] E. S. Khorasani, Z. Zhenge, and J. Champaign. A markov chain collaborative filtering model for course enrollment recommendations. In *IEEE International Conference on Big Data (Big Data)*, pages 3484–3490. IEEE, 2016.

[9] C. Kim, N. Choi, Y. Heo, J. Sin, and . On the development of a course recommender system: A hybrid filtering approach. *Entrue Journal of Information Technology*, 14(2):71–82, 2015.

[10] F. Le Roux, E. Ranjeet, V. Ghai, Y. Gao, J. Lu, and A. T. PRess. A course recommender system using multiple criteria decision making method. In *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering*, Advances in Intelligent Systems Research, 2007.

[11] H. Ma, X. Wang, J. Hou, Y. Lu, and Ieee. Course recommendation based on semantic similarity analysis. In *Conference Proceedings of 2017 3rd Ieee International Conference on Control Science and Systems Engineering (Iccsse)*, pages 638–641, 2017.

[12] A. Parameswaran, P. Venetis, and H. Garcia-Molina. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems*, 29(4):20:1–20:33, 2011.

[13] C. Romero and S. Ventura. Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1):12–27, 2013.

[14] J. Sobecki and J. M. Tomczak. Student courses recommendation using ant colony optimization. In N. T. Nguyen, M. T. Le, and J. Swiatek, editors, *Intelligent Information and Database Systems, Pt Ii, Proceedings*, volume 5991 of *Lecture Notes in Artificial Intelligence*, pages 124–133, 2010.

[15] S. Spiegel. *A Hybrid Approach to Recommender Systems based on Matrix Factorization.* Thesis, Technical University Berlin, 2009.

[16] C. Vialardi, J. Chue, J. Pablo Peche, G. Alvarado, B. Vinatea, J. Estrella, and A. Ortigosa. A data mining approach to guide students through the enrollment process based on academic performance. *User Modeling and User-Adapted Interaction*, 21(1-2):217–248, 2011.

[17] R. Wang. *Sequence-based Approaches to Course Recommender Systems.* Thesis, University of Alberta, 2017.

[18] L. Zhuhadar, O. Nasraoui, R. Wyatt, and E. Romero. Multi-model ontology-based hybrid recommender system in e-learning domain. In R. BaezaYates, B. Berendt, E. Bertino, E. P. Lim, and G. Pasi, editors, *IEEE/WIC/ACM International Joint Conferences on Web Intelligence*, pages 91–95, 2009.