IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 17, 2018, accepted October 31, 2018, date of publication November 13, 2018,
date of current version December 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881074

Hybrid Collaborative Filtering Based on Users Rating Behavior

FERNANDO ORTEGA[1], DIEGO ROJO[1], PRISCILA VALDIVIEZO-DÍAZ[2,3], AND LAURA RAYA[1]
[1]U-tad: Centro Universitario de Tecnología y Arte Digital, 28290 Madrid, Spain
[2]Department of Information Systems, Universidad Politécnica de Madrid, 28031 Madrid, Spain
[3]Computer Science and Electronic Department, Universidad Técnica Particular de Loja, Loja 1101608, Ecuador

Corresponding author: Priscila Valdiviezo-Díaz (pmvaldiviezo@utpl.edu.ec)

This work was supported by the Universidad Técnica Particular de Loja.

ABSTRACT Several collaborative filtering (CF) approaches have been developed in order to improve the quality of the recommendations. However, this improvement has always been measured as the average quality of the performed recommendations across all the users. It has not been analyzed for each individual user. In this paper, the existence of a more precise CF approach for each user is demonstrated. So, a novel hybrid method that merges recommendations provided by different CF approaches based on a multi-class classification algorithm is proposed. This classification is performed based on the user rating behavior. Experiments have been carried out on the MovieLens and Netflix datasets. The experimental results demonstrate an improvement on quality of both predictions and recommendations using the proposed hybrid CF approach. In addition, experiments have compared state-of-the-art baselines with the results obtained by the proposed approach.

INDEX TERMS Recommender systems, collaborative filtering, hybrid CF, KNN, matrix factorization.

I. INTRODUCTION

Recommender Systems (RS) [1] provide a relevant tool to mitigate the information overload problem. RS act as a filter that allows to pass the relevant information to the user and blocks the irrelevant one. RS have been used to recommend a wide variety of items [2]: movies, books, e-commerce, educational resources, etc. Collaborative Filtering (CF) [3] is the most popular implementation of RS. CF recommendations are computed based on the ratings that the community of users has made over a set of items.

CF has been evolving during the last decade. Initial CF implementations used a memory-based approach [4], [5], i.e. recommendations were computed with methods that act directly on the rating matrix. The most popular implementation was based on k Nearest Neighbors (KNN) algorithm: a similarity metric was used to compute the k most similar users with respect to an active one, and the recommendations were performed based on the favorite items of the k neighbors. Several similarity metrics have been developed to improve the overall accuracy of the RS [6]–[9]. JMSD [6] has demonstrated to be one of the most accurate similarity metrics, whereas other similarity metrics like PIP [7] report better performance in cold-start situations (new users or items of the RS with a low number of ratings).

Nowadays, CF implementations are focused on model-based approach. Recommendations are computed using a model built from the rating matrix. Matrix Factorization (MF) [10] models, such as Non-Negative Matrix Factorization (NMF) [11], Probabilistic Matrix Factorization (PMF) [12] or Bayesian Non-Negative Matrix Factorization (BNMF) [13] , are the models that has achieved better results in accuracy and performance. MF CF is more accurate and more precise than KNN CF. Furthermore, MF CF provides a higher scalability than KNN CF. PMF [12], the main reference in MF CF, factorizes the rating matrix into two matrices that represent the users and items in a hidden k dimensional latent space. Other factorization methods, such as BNMF [13], have reported better performance than PMF.

The rest of the paper is structured as follows: Section II contains the definition of the proposed method. Section III includes the experiments design and empirical results to measure the quality of the performed recommendations. Section IV shows the related work and Section V encloses the conclusions of this contribution and future work.

II. PROPOSED METHOD

A. MOTIVATION

Several CF based recommendation approaches have been proposed in order to improve the quality of both predictions

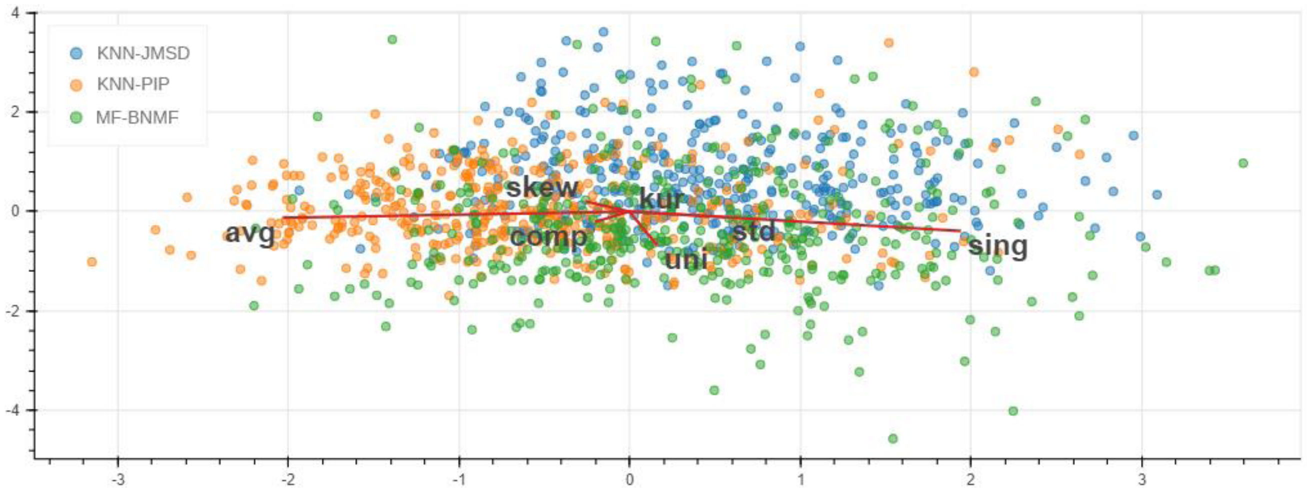


FIGURE 1. Star coordinates visualization of users on MovieLens 10M dataset. Axes vectors represent users' features. Colors denote CF approach that provides less prediction error.

and recommendations. MF CF has demonstrated its superior performance against traditional KNN CF [14]. In general, MF CF reports higher overall accuracy and precision than KNN CF. This improvement is usually measured as the average error of all the users. For example, to compute the accuracy of an RS, the accuracy of each user of that RS is computed and, after that, the accuracy values of the whole users are averaged. This methodology provides the global quality of an RS. However, the real impact of different recommendation approaches on each individual user is not analyzed. If an RS approach reports higher accuracy than another RS, does it mean that the recommendations to all the users are more accurate with the first RS than with the second one? As MF CF has demonstrated its superior performance against KNN CF, does it means that MF CF is the best recommendation approach for each user of the RS?

To answer this question, the following experiment has been designed. The quality of the predictions on MovieLens 10M dataset using different CF approaches have been measured. We have selected three distinctive CF approaches: (a) MF-BNMF [13], a novel method to factorize rating matrix with an understanding probabilistic meaning that reports better accuracy and precision than traditional MF CF; (b) KNN-JMSD [6], a similarity metric that performs high accurate recommendations using KNN CF; and (c) KNN-PIP [7], a similarity metric designed to improve recommendation accuracy of cold start users. Then, the users have been classified according to the recommendation method that provides less prediction error. 36.5% of the tested users obtain more accurate predictions using MF-BNMF, 40.5% using KNN-PIP and 23.0% using KNN-JMSD. Finally, several features of each user based on his/her rating behavior have been extracted (these features are discussed in Subsection II.C.) and an exploratory data analysis has been done using [15] to try to understand the relevance of each studied feature to the classification.

Fig. 1 shows a star coordinates visualization from the exploratory data analysis in which the axes are placed so that the scatter plot matches the Linear Discriminant Analysis 2D projection, which maximizes the separation between the classes. The red axes vectors associated with each feature show their direction and magnitude of influence to the separation of the classes. It is very clear that there is one cluster for each CF approach: MF-BNMF users are positioned at the bottom of the chart, KNN-JMSD users are located at the top-right corner, and KNN-PIP users are placed on the left side of the figure. These positions provide information on the main features of the users of each cluster, through the red axes that represent the users' features.

KNN-PIP users have a high average in their ratings. This happens because PIP is a KNN similarity metric designed for cold-star users, that is, new users in the RS that only rate their favorite items. KNN-JMSD and MF-BNMF users, located on the right side of the chart, have a high singularity in their ratings and/or a low rating average. MF-BNMF users have higher completeness (a greater number of ratings) and uniformity (ratings distributed uniformly into all plausible rating values) than KNN-JMSD. This happens because MF CF has better performance when the amount of information about the users is higher.

B. HYPOTHESIS

Based on the experiment results shown in the Fig. 1, the following hypothesis is formulated: The quality of the recommendations provided for a CF RS to a user depends on both the goodness of the recommender and rating behavior of the user. If this hypothesis is true, we can design a hybrid CF that merges the recommendations of different CF approaches based on the features extracted from the rating behavior of the user who will receive the recommendations.

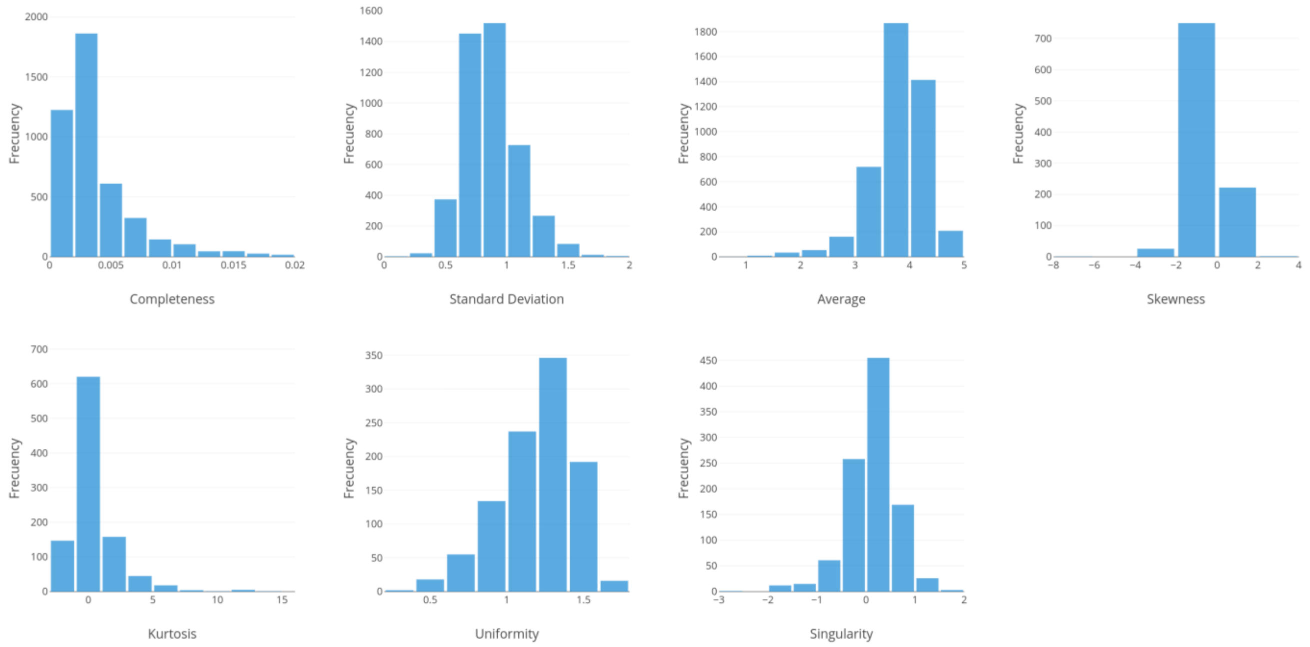


FIGURE 2. User features values for MovieLens 10M dataset.

C. USER RATING BEHAVIOR

In order to classify the users of the RS to each recommendation method, some features of them need to be extracted first. These features cannot be their ratings to the items due to the high sparsity of the rating matrix (users who have not rated any item in common cannot be compared). So, user classification based on their rating behavior is proposed.

User rating behavior will be defined using seven different features extracted for the ratings of each user. These features are:

- ***Completeness*****:** It measures the number of items rated by the user with respect to the total number of items. It can be computed as:

$$comp_u = \frac{\#\{i \in I \left| r_{u,i} \neq \bullet \right.\}}{\#I} \tag{1}$$

- ***Ratings standard deviation:*** It measures the deviation of the ratings from the mean that the user has made. It can be computed as:

$$std_u = \frac{\sum_{i \in I_u} (r_{u,i} - \bar{r}_u)^2}{\#I_u} \tag{2}$$

- ***Rating average:*** It measures the average rating of a user. It can be computed as:

$$avg_u = \frac{\sum_{i \in I_u} r_{u,i}}{\#I_u} \tag{3}$$

- ***Skewness:*** It measures the asymmetry of the rating distribution about its mean. It can be computed as:

$$skew_u = \frac{\sum_{i \in I_u} (r_{u,i} - \bar{r}_u)^3 \#I_u}{\sigma_u^3} \tag{4}$$

- ***Kurtosis:*** It measures the tailedness of the rating distribution of the user. It can be computed as:

$$kur_u = \frac{\sum_{i \in I_u} (r_{u,i} - \bar{r}_u)^4 \#I_u}{\sigma_u^4} \tag{5}$$

- ***Uniformity:*** It measures the inequality in the proportion of votes of each plausible rating. It can be computed as:

$$uni_u = \sum_{k \in R} \left| \frac{\#\{i \in I_u \left| r_{u,i} = k \right.\}}{\#I_u} - \frac{1}{\#R} \right| \tag{6}$$

- ***Singularity:*** It measures the singularity of the ratings of the user with respect to the average rating of each item. It can be computed as:

$$\sin g_u = \frac{\sum_{i \in I_u} (r_{u,i} - \bar{r}_i)}{\#I_u} \tag{7}$$

Where I is the set of items, I_u is the set of items rated by the user u, $r_{u,i}$ is the rating of the user u to the item i, $\bar{r}_u$ represents the average rating of the user u, σ_u denotes the standard deviation of ratings of user u, $\bar{r}_i$ represents the average rating of the item i, R is the set of plausible ratings, $\bullet$ denotes the absence of vote and # denotes the cardinality of a set.

Fig. 2 contains the bars diagram of the values of each feature for MovieLens 10M dataset users. It is observed that the selected features take a wide range of values for different users of the dataset.

D. CF APPROACH CLASSIFIER

The hypothesis proposed in this work claims to find the most suitable CF approach for each user based on his/her rating signature. This is a multi-class classification problem in which the classifier receives as input the users' rating behavior and predicts the most suitable CF approach (each CF approach can be interpreted as a class) for each user.

To perform this classification, we propose to use a multi-class logistic regression [16]. This classifier has been trained as explained in the sub-section III.A. The input of the classifier is a set of seven features extracted from the user rating behavior described in section II.C. The output of the classifier is the probability to belong to each class, i.e. the probability that each CF approach is the most suitable for a user based on his/her ratings.

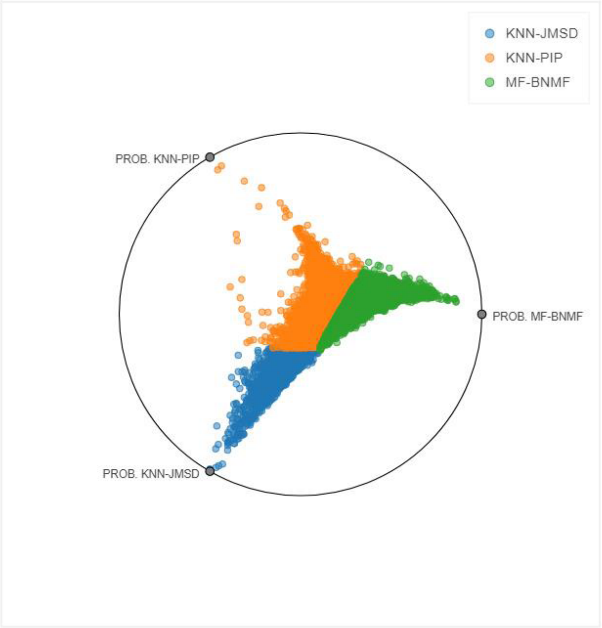


FIGURE 3. **RadViz visualization of the probability distribution returned by the classifier for each MovieLens 10M dataset user.**

Fig. 3 contains a RadViz visualization of the probability distribution returned by the classifier for each MovieLens 10M dataset user. MF-BNMF, KNN- JMSD and KNN-PIP have been used as CF approaches. Each user has been colored according to the CF approach with the highest probability. It is observed that there are more users with uncertainty about his/her best CF approach, in the center of the chart, than with a reliable CF approach, near the grey circumference. It is also shown that KNN-PIP users are identified with more difficulty than MF-BNMF or KNN-JMSD users. This is because on MovieLens 10M dataset there are less cold-start users than not cold-start ones.

E. COMPUTING RECOMMENDATIONS

The output of the classifier described in section II.D is a dense vector that contains the probability that a user belongs to each CF approach knowing the user rating behavior. Based on these probabilities, the predictions computed can be merged for each CF approach using weighted average aggregation. The prediction $\hat{r}_{u,i}$ of the item i for the user u can be computed as follows:

$$\hat{r}_{u,i} = \frac{\sum_{k \in CF} \theta_u^k \hat{r}_{u,i}^k}{\sum_{k \in CF} \theta_u^k} \tag{8}$$

Where CF is the set of CF approaches used on the hybrid model, and $\theta_{k,u}$ is the probability that the user u belongs to the CF approach k provided by the CF classifier.

Predictions computed on equation 8 are the predictions provided by the hybrid CF proposed in the paper. The hybrid CF can be classified as a weighted hybridization technique according to Burke's taxonomy [17]. These predictions can be used to compute the recommendations for a user by selecting the items with a higher prediction value.

III. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

Experiments will be performed using MovieLens 10M [18] and Netflix [19] datasets. MovieLens contains 10,000,054 ratings from 69,878 to 10,677 items in a 5-star scale, with half-star increments. Netflix contains 100,480,507 ratings from 480,189 to 17,770 items in a 5-star scale.

The Hybrid CF based on Users Rating Behavior (HCFURB) proposed in this work will be compared with several CF approaches in order to validate the hypothesis of subsection II.B. On the one hand, HCFURB will be compared with the CF approaches used to build the hybrid method, i.e. MF-BNMF [13], KNN-JMSD [6], and KNN-PIP [7]. On the other hand, proposed method will be compared with CombSum [20], a novel hybrid method that combines the prediction of multiple CF methods by aggregating them using the summation operation. CombSum has been tested using the same CF approaches than HCFURB.

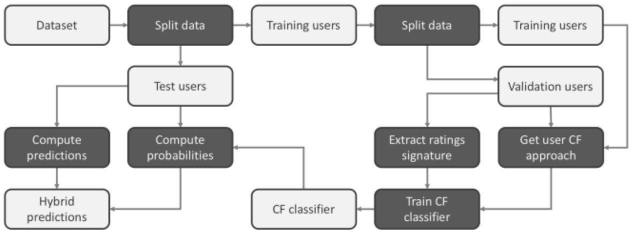


FIGURE 4. **Flow chart of experiments.**

Perform recommendations using HCFURB requires the training of both CF classifier and CF approaches. Fig. 4 summarizes experimentation process. Initial dataset is split into training and test users: training users will be used to train both CF and classification methods and test users will be used to measure the performance of hybrid model. Training users set are again split into training and validation users: training users will be used to obtain the most accurate CF approach for each validation user and validation users will be used to train the CF classifier. Once the CF classifier is trained, the probability that a user belongs to a CF approach based on his/her rating signature can be obtained for each test user. Predictions performed for each CF approach will be aggregated based on this probability distribution to obtain hybrid predictions. Hybrid predictions will be used to compute quality measures of recommender systems. These experiments have been carried out using CF4J [21], a Java's CF library designed to carry out experiments in research of CF RS. Table 1 summarizes the main parameters used to perform these experiments.

TABLE 1. Main parameters used to perform experiments.

Parameter	MovieLens	Netflix
Test user's percent	0.2	0.04
Training user's percent	0.5	0.9
Validation user's percent	0.3	0.06
Test items percent	0.2	0.15
Training items percent	0.8	0.85

CF classifier is built using a multi-class logistic regression as explained in section II.D. This classification method requires some parameters to work. These parameters have been tuned to obtain the best classification accuracy. Table 2 contains the parameters obtained for each dataset. Classification step has been performed using Azure Machine Learning Studio [22].

TABLE 2. Main parameters of logistic regression classifier used in the experiments.

Parameter	MovieLens	Netflix
Optimization tolerance	1E-07	1E-07
L1 weight	1	0.01
L2 weight	0.1	0.01
Memory size	20	5

CF approaches selected for the hybrid model requires several parameters to work. Table 3 contains the parameters involved in recommendation process for each dataset. These parameters have been tuned to maximize the quality of both predictions and recommendations provided for each CF approach.

B. QUALITY MEASURES

In order to analyze the behavior of HCFURB compared to other CF approaches, the following quality measures have been used: Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) to measure the quality of the predictions; precision, recall and normalized Discounted Cumulative Gain (nDCG) to measure the quality of recommendations.

TABLE 3. Main parameters of recommendation methods.

	Parameter	MovieLens	Netflix
BNMF	Number of topics	7	11
	α	0.8	0.8
	β	8	10
	Number of iterations	300	300
JMSD	Number of neighbors	475	525
PIP	Number of neighbors	475	525

We define MAE_u as the mean absolute difference between the test ratings of the user u and the predicted ones:

$$MAE_u = \frac{\sum_{i \in \hat{I}_u} \left| r_{u,i} - \hat{r}_{u,i} \right|}{\# \hat{I}_u} \tag{9}$$

Where $\hat{I}_u$ is the set of the test items rated by the user u.

And MAE as the averaged MAE_u for all the users of the RS:

$$MAE = \frac{MAE_u}{\# U} \tag{10}$$

We define $RMSE_u$ as the root mean squared difference between the test ratings of the user u and the predicted ones:

$$RMSE_u = \sqrt{\frac{\sum_{i \in \hat{I}_u} \left(r_{u,i} - \hat{r}_{u,i} \right)^2}{\# \hat{I}_u}} \tag{11}$$

And $RMSE$ as the averaged $RMSE_u$ for all the users of the RS:

$$RMSE = \frac{RMSE_u}{\# U} \tag{12}$$

We define $precision_u@N$ as the proportion of the N items recommended to the user u that are relevant to him/her:

$$precision_u@N = \frac{\# \left\{ i \in R_u^N \left| r_{u,i} \geq \theta \right. \right\}}{N} \tag{13}$$

Where R_u^N is the set of N items recommended to the user u and θ is the threshold to discriminate if a recommendation is relevant or not.

And $precision@N$ as the averaged $precision_u@N$ for all the users of the RS:

$$precision@N = \frac{precision_u@N}{\# U} \tag{14}$$

We define $recall_u@N$ as the proportion of relevant items recommended to the user u with respect to the total relevant items rated:

$$recall_u@N = \frac{\# \left\{ i \in R_u^N \left| r_{u,i} \geq \theta \right. \right\}}{\# \left\{ i \in \hat{I}_u \left| r_{u,i} \geq \theta \right. \right\}} \tag{15}$$

And $recall@N$ as the averaged $recall_u@N$ for all the users of the RS:

$$recall@N = \frac{recall_u@N}{\# U} \tag{16}$$

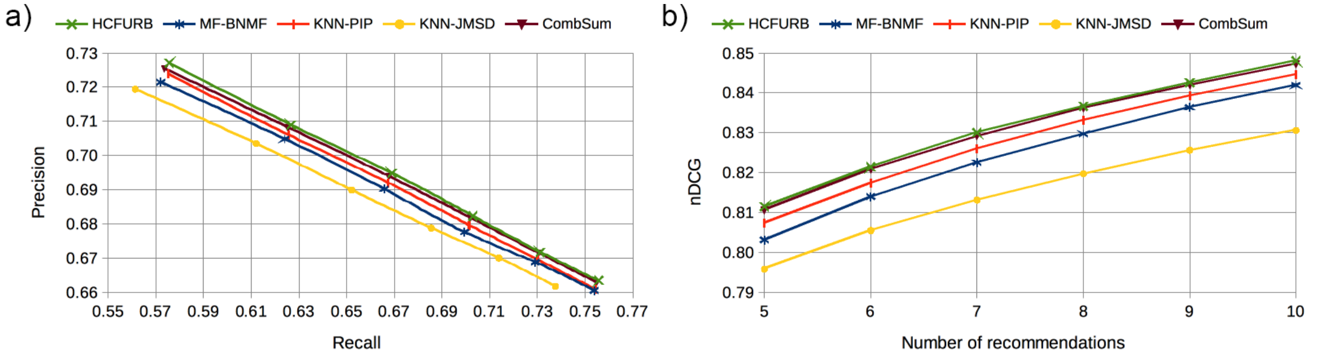


FIGURE 5. (a) Precision & Recall and (b) normalized Discounted Cumulative Gain (nDCG) of each recommendation method for MovieLens 10M dataset.

We define $nDCG_u@N$ as the normalized relevance of N recommendations provided to user u based on its position in the recommendation list:

$$DCG_u@N = \sum_{p=1}^{N} \frac{2^{r_{u,x_{u,p}}} - 1}{\log_2(p+1)} \tag{17}$$

$$IDCG_u = \sum_{p=1}^{\#\hat{I}_u} \frac{2^{r_{u,y_{u,p}}} - 1}{\log_2(p+1)} \tag{18}$$

$$nDCG_u@N = \frac{DCG_u@N}{IDCG_u} \tag{19}$$

Where $x_{u,p}$ is the item recommended at the p-th position if items recommended to user u are sorted from higher to lower prediction ($\hat{r}_{u,i}$) and $y_{u,p}$ is the item at the p-th position if test items rated by user u ($\hat{I}_u$) are sorted by its rating value ($r_{u,i}$).

And $nDCG@N$ as the averaged $nDCG_u@N$ for all the users of the RS:

$$nDCG@N = \frac{nDCG_u@N}{\#U} \tag{20}$$

C. EXPERIMENTAL RESULTS

HCFURB requires training the multi-class logistic regression classifier to compute the probability to perform recommendations with each CF approach. This training process is performed with the training ratings of the dataset as shown in Fig. 4. Training data, which is split into training and validation sets, is used to perform predictions to each validation user with each CF approach. The most accurate CF method for each validation user is set as the user class. Table 4 contains the number of validation users assigned to each CF method during the experimentation phase for both tested datasets. Fig. 1 contains a graphical representation of these users. These experimental results confirm the hypothesis proposed in this work: accuracy of each CF approach is conditioned by the user rating signature. There is no CF approach that improves the prediction accuracy for every user of the dataset.

Table 5 contains the MAE and RMSE values for both MovieLens 10M and Netflix dataset. It is observed that HCFURB provides more accurate predictions than any other studied CF approach. HCFURB is significantly better than CF approaches used during the hybridization process in both MovieLens 10M and Netflix datasets. Furthermore, proposed method provides a slightly improvement in MAE and an important improvement in RMSE with respect to CombSum hybrid CF approach.

TABLE 4. Number of validation users assigned to each CF method.

CF method	MovieLens	Netflix
MF-BNMF	1,531	1,782
KNN-JMSD	1,546	1,250
KNN-PIP	810	775

TABLE 5. Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) of each recommendation method for both MovieLens and Netflix datasets.

	MovieLens 10M		Netflix	
	MAE	RMSE	MAE	RMSE
MF-BMF	0.70331	0.85066	0.75154	0.89842
KNN-PIP	0.70204	0.86891	0.73810	0.90762
KNN-JMSD	0.70052	0.85482	0.73282	0.89207
CombSum	0.69931	0.85521	0.75680	0.91780
HCFURB	**0.68614**	**0.83589**	**0.73202**	**0.88165**

Fig. 5 contains the quality values for MovieLens 10M dataset. Precision and Recall have been tested using the relevance threshold at value 4 to discriminate if the test rating is a positive or negative recommendation ($\theta = 4$) in both MovieLens 10M and Netflix datasets. We can observe that HCFURB provides better precision & recall than other tested CF approaches.

Moreover, HCFURB provides higher nDCG than non-hybrid approaches and the same nDCG than CombSum method. Fig. 6 contains the quality values for Netflix dataset.

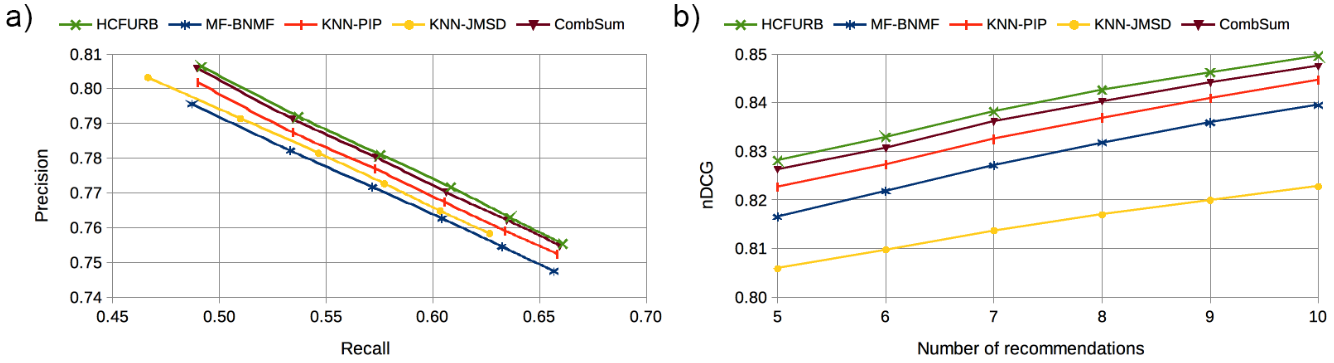


FIGURE 6. (a) Precision & Recall and (b) normalized Discounted Cumulative Gain (nDCG) of each recommendation method for Netflix dataset.

We can observe that HCFURB provides better precision & recall and nDCG than any other tested CF approach. In brief, HCFURB provides better recommendation accuracy than any other tested baseline for both datasets.

Additionally, paired t-tests to confirm that the proposed method improvement on the quality of both predictions and recommendations with respect to CombSum baseline is not due to random have been performed. In our experiments, the null hypothesis considers that the quality of both hybrid methods is the same, while the alternative hypothesis considers that the quality of HCFURB is better than the quality of CombSum. Table 6 contains the p-values of these t-tests. As we see, the p-values obtained are all below the typical significance level 0.05. Consequently, we can conclude that there is statistic evidence for rejecting the null hypothesis and accepting the alternative hypothesis, i.e. the improvement of the quality of both predictions and recommendations made by HCFURB with respect to CombSum hybrid CF approach is statistically significant.

TABLE 6. p-values of paired t-test performed to compare HCFURB with CombSum hybrid approach.

	MovieLens 10M	Netflix
MAE	2.8073E-92	3.9139E-99
RMSE	1.1732E-127	9.1394E-157
Precision@10	0.0269	0.0253
Recall@10	0.0015	3.3461E-07
nDCG@10	0.0497	1.8783E-05

D. COMPLEXITY ANALYSIS

Experimental results show an improvement in both accuracy of predictions and quality of recommendations using HCFURB. However, proposed method has a higher computational cost than the CF approaches that it combines. In this section we study the computational complexity of HCFURB.

Compute predictions using HCFURB requires 3 stages: (a) train the multiclass classifier to obtain the suitability of a user to each CF approach; (b) train the CF approaches used to perform recommendations; and (c) combine the output of CF approaches to obtain the final predictions.

In the stage (a), a multi-class logistic regression is used. The computational complexity of this classification algorithm is $O(U{\cdot}N)$, where U is the number of users and N is the number of features. In our case, the number of features has been fixed to 7 (subsection II.C), so the computational complexity can be simplified as $O(U)$. The classifier can be trained once when the RS is setting up. It can be updated later when new ratings are incorporated into the RS.

In stage (b), 3 different CF approaches has been used: MF-BNMF, KNN-JMSD and KNN-PIP. All these CF approaches require a training phase in order to predict the missing ratings of the rating matrix: MF based approaches must compute the latent factors of the model and KNN based approaches must compute the k nearest neighbors of each user. The computational complexity of the training phase of these CF approaches can be observed in Table 7. Computational complexity of MF-BNMF has been simplified due the low value of F (around 10) and T (around 250) compared with the number of users and items. Despite KNN-JMSD and KNN-PIP have the same complexity, KNN-JMSD is faster than KNN-PIP because its computation requires less mathematical operations. All these CF approaches are independent among them, so its training can be parallelized, and the computational complexity of this stage is equal to the worst computational complexity, i.e. KNN-PIP's computational complexity ($O(U^2 \cdot I)$). CF approaches can be trained

TABLE 7. Computational complexity of CF approaches used in proposed method. ***U*** is the number of users. ***I*** is the number of items. ***F*** is the number of latent factors for MF. ***T*** is the number of iterations. ***K*** is the number of neighbors for KNN.

	Training phase	Prediction phase for one user
MF-BNMF	$O(I{\cdot}U{\cdot}F^2{\cdot}T) \approx O(I{\cdot}U)$	$O(F{\cdot}I) \approx O(I)$
KNN-JMSD	$O(U^2{\cdot}I)$	$O(K{\cdot}I)$
KNN-PIP	$O(U^2{\cdot}I)$	$O(K{\cdot}I)$

once when the RS is setting up. They can be updated later when new ratings are incorporated into the RS.

In stage (c) the predictions of each CF approach are combined according to the output of the classifier trained in stage (a). On one hand, to get the suitability of a user to each CF approach based on his/her rating behavior has a computational complexity of $O(1)$, assuming that the user features have been previously computed. On the other hand, the computational complexity of generating predictions to a user using each CF approach is shown in Table 7. Like in the stage (b), these calculations can be parallelized due to the independence of each CF approach, so the computational complexity of this stage is equal to the highest one, i.e. KNN-PIP or KNN-JMSD's computational complexity ($O(K{\cdot}I)$).

In brief, total computational complexity of HCFURB is $O(U) + O(U^2 \cdot I)$ for training and $O(K{\cdot}I)$) for compute the rating prediction to a user.

IV. RELATED WORK

Hybrid CF has been widely studied by researchers to enhance CF properties. Hybrid CF implementations try to merge the advantages of different CF approaches to minimize their drawbacks. Hybrid CF can be classified according to the CF approaches used to perform hybrid recommendations [17].

Memory-based hybrid CF joins several memory-based CF approaches. Reference [23] proposes to combine an item similarity measure based on the KullbackLeibler divergence with a user preference factor and an asymmetric factor to distinguish the rating preference between users. Reference [24] uses an evolutionary multi-objective optimization-based recommendation system to pull up a group of profiles to provide high performances in terms of both accuracy and diversity. Reference [25] uses Pareto dominance to perform a pre-filtering process eliminating the less representative users from the k-neighbors selection process while retaining the most promising ones. Model-based hybrid CF merges multiple CF models in a new one. Neural networks and fuzzy systems are popular models using in this approach. Reference [26] proposes a hybrid method that uses an item-based CF to handle data sparsity and scalability problems. Case Based Reasoning combined with average filling is used to handle the sparsity of data set, while Self-Organizing Map optimized with Genetic Algorithm performs user clustering to reduce the scope for item-based CF. Reference [27] introduces new recommendation methods using Adaptive Neuro-Fuzzy Inference Systems, used for discovering knowledge from users' ratings, and Self-Organizing Map clustering, enabling generation of high quality clusters. Reference [28] presents a systematic mathematical definition of fuzzy RS including theoretical analyses of algebraic operations and properties. They propose a novel user-based hybrid CF method that integrates the fuzzy similarity degrees between users based on the demographic data with the hard user-based degrees calculated from the rating histories.

In the same way [29], designs a fuzzy hybrid multi-agent RS, which uses an interval type-2 fuzzy sets to create user models capable of capturing the inherent ambiguity of human behavior related to diverse users' tastes.

MF based hybrid CF incorporate additional data to the factorization process coming from other CF approaches. Reference [30] combines ontology techniques and dimensionality reduction technique to find the most similar items and users to significantly improve the scalability of the recommendation method.

In [31], authors present a model which not only considers the items' content information, but also the users' demographic and behavior information to capture the users' interests and preferences.

Reference [32] proposes a hybrid model, Collaborative Topic Model for recommending scientific articles to users, based on both items content and users' ratings.

Reference [33] presents a hybrid CF model by incorporating both event-based and user-based neighborhood methods into matrix factorization to solve the problem of predicting users' social influences on upcoming events. Reference [34] describes a hybrid approach which combines content-filtering techniques with a well-known matrix factorization technique in the implementation of the item-based CF algorithm to simplify as much as possible the user task of selecting what program to watch on TV. Reference [35] proposes a unified model for collaborative filtering based on graph regularized weighted non-negative matrix factorization. Two graphs are constructed on users and items. The proposed method not only inherits the advantages of a model-based method, but also owns the merits of a memory-based method, which considers the neighborhood information.

Finally, [20] studies the usage of meta search algorithms to combine top-N recommenders into a hybrid model. They study different rank aggregation methods that no require any training or tuning process. They demonstrate that CombSum aggregation is the best approach in each scenario.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have designed a CF hybrid method that merges recommendations provided by different CF approaches based on a multi-class classification algorithm. This classification is performed based on the user rating behavior.

Experimental results demonstrate an improvement on the quality of predictions and recommendations in all the studied scenarios for both MovieLens and Netflix datasets. These experimental results confirm the hypothesis formulated in Section II.B: The quality of the recommendations provided for a CF based RS to a user depends on both the goodness of the recommender and the features of that user.

Furthermore, the proposed method may reduce the impact of shilling attacks and profile injection attacks [36], [37]. These attacks introduce malicious ratings on an RS to favor or disfavor the recommendation of specific items [38].

How to introduce these malicious items is not trivial: RS implementation details must be known in order to hack it. Proposed method uses several CF approaches to perform recommendations, so if one of them is attacked, the other ones will still provide reliable recommendations. Moreover, by classifying the users according to its more accurate CF approach, attacks will only affect a small group of users.

This work opens a novel research line in hybrid CF recommendations. The proposed model can be extended to incorporate more features to the users rating behavior to improve the accuracy of the classification. Additionally, the proposed model can be enhanced by including more CF approaches that represent users not covered by the CF approaches used in this work.

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[2] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Springer, 2011, pp. 1–35.

[3] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[4] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 4, Aug. 2009, Art. no. 421425.

[5] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*. Springer, 2015, pp. 191–226.

[6] J. Bobadilla, F. Serradilla, and J. Bernal, "A new collaborative filtering metric that improves the behavior of recommender systems," *Knowl.-Based Syst.*, vol. 23, no. 6, pp. 520–528, 2010.

[7] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Inf. Sci.*, vol. 178, no. 1, pp. 37–51, 2008.

[8] J. Bobadilla, F. Ortega, A. Hernando, and A. Arroyo, "A balanced memory-based collaborative filtering similarity measure," *Int. J. Intell. Syst.*, vol. 27, no. 10, pp. 939–946, 2012.

[9] J. Bobadilla, F. Ortega, and A. Hernando, "A collaborative filtering similarity measure based on singularities," *Inf. Process. Manage.*, vol. 48, no. 2, pp. 204–217, 2012.

[10] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[11] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non negative matrix factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2006, pp. 549–553.

[12] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

[13] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowl.-Based Syst.*, vol. 97, pp. 188–202, Apr. 2016.

[14] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 426–434.

[15] D. Rojo, L. Raya, M. Rubio-Sánchez, and A. Sánchez, "A visual interface for feature subset selection using machine learning methods," in *Proc. Spanish Comput. Graph. Conf. (CEIG)*, I. García-Fernández and C. Ureña, Eds. The Eurographics Association, 2018.

[16] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, vol. 398. Hoboken, NJ, USA: Wiley, 2013.

[17] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, 2002.

[18] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19, 2016.

[19] J. Bennett *et al.*, "The netix prize," in *Proc. KDD Cup Workshop*, New York, NY, USA, 2007, p. 35.

[20] D. Valcarce, J. Parapar, and A. Barreiro, "Combining top-n recommenders with metasearch algorithms," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2017, pp. 805–808.

[21] F. Ortega, B. Zhu, J. Bobadilla, and A. Hernando, "CF4J: Collaborative filtering for java," *Knowl.-Based Syst.*, vol. 152, pp. 94–99, Jul. 2018.

[22] S. Mund, *Microsoft Azure Machine Learning*. Birmingham, U.K.: Packt, 2015.

[23] Y. Wang, J. Deng, J. Gao, and P. Zhang, "A hybrid user similarity model for collaborative filtering," *Inf. Sci.*, vols. 418–419, pp. 102–118, Dec. 2017.

[24] N. E. I. Karabadji, S. Beldjoudi, H. Seridi, S. Aridhi, and W. Dhii, "Improving memory-based user collaborative filtering with evolutionary multi-objective optimization," *Expert Syst. Appl.*, vol. 98, pp. 153–165, May 2018.

[25] F. Ortega, J. L. Sánchez, J. Bobadilla, and A. Gutiérrez, "Improving collaborative filtering-based recommender systems results using Pareto dominance," *Inf. Sci.*, vol. 239, pp. 50–61, Aug. 2013.

[26] N. P. Kumar and Z. Fan, "Hybrid user-item based collaborative filtering," *Procedia Comput. Sci.*, vol. 60, pp. 1453–1461, 2015.

[27] M. Nilashi, O. B. Ibrahim, and N. Ithnin, "Hybrid recommendation approaches for multi-criteria collaborative filtering," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3879–3900, 2014.

[28] L. H. Son, "HU-FCF: A hybrid user-based fuzzy collaborative filtering method in recommender systems," *Int. J. Expert Syst. Appl.*, vol. 41, no. 15, pp. 6861–6870, 2014.

[29] P. Vashisth, P. Khurana, and P. Bedi, "A fuzzy hybrid recommender system," *J. Intell. Fuzzy Syst.*, vol. 32, no. 6, pp. 3945–3960, 2017.

[30] M. Nilashi, O. Ibrahim, and K. Bagherifard, "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018.

[31] X. Wang, F. Luo, C. Sang, J. Zeng, and S. Hirokawa, "Personalized movie recommendation system based on support vector machine and improved particle swarm optimization," *IEICE Trans. Inf. Syst.*, vol. 100, no. 2, pp. 285–293, 2017.

[32] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, vol. 11, 2011, pp. 448–456.

[33] X. Li, X. Cheng, S. Su, S. Li, and J. Yang, "A hybrid collaborative filtering model for social influence prediction in event-based social networks," *Neurocomputing*, vol. 230, pp. 197–209, Mar. 2017.

[34] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Inf. Sci.*, vol. 180, no. 22, pp. 4290–4311, 2010.

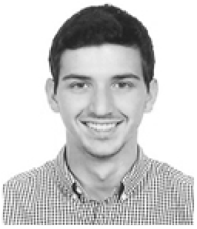
[35] Q. Gu, J. Zhou, and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *Proc. SIAM Int. Conf. Data Mining*, 2010, pp. 199–210.

[36] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: A comprehensive survey," *Artif. Intell. Rev.*, vol. 42, pp. 767–799, Dec. 2014.

[37] I. Gunes and H. Polat, "Detecting shilling attacks in private environments," *Inf. Retr. J.*, vol. 19, pp. 547–572, Dec. 2016.

[38] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Trans. Internet Technol.*, vol. 7, no. 4, p. 23, 2007.

FERNANDO ORTEGA was born in Madrid, Spain, in 1988. He received the B.S. degree in software engineering and the Ph.D. degree in computer sciences from the Universidad Politécnica de Madrid, Madrid, in 2010 and 2015, respectively. He is currently an Assistant Professor with the Department of Engineering, Centro Universitario de Tecnología y Arte Digital. He has published several papers in the most relevant international journals. His main research fields are machine learning, data analysis, and artificial intelligence. He also actively collaborates in various projects with technology companies.



DIEGO ROJO was born in Valladolid, Spain, in 1993. He received the B.S. degree in mathematics from the Universidad de Valladolid in 2015 and the master's degree in computer graphics and simulation from the Centro Universitario de Tecnología y Arte Digital (U-tad), Madrid, in 2016. He is currently an Assistant Professor at the Department of Engineering, U-tad. His main research fields are information visualization, visual analytics, and machine learning.

PRISCILA VALDIVIEZO-DÍAZ received the degree in engineering in computer science and the master's degree in distance education from the Universidad Técnica Particular de Loja, Ecuador, and the Diploma of Advanced Studies in artificial intelligence from the Universidad Nacional de Educación a Distancia, Spain. He is currently a Professor with the Computer Science and Electronics Department, Universidad Técnica Particular de Loja. His research interests include artificial intelligence applied to education, recommender systems, and machine learning.

LAURA RAYA received the M.S. degree in computer science, the M.S. degree in computer graphics, and the Ph.D. degree in computer science from the Universidad Rey Juan Carlos of Madrid in 2008, 2010, and 2014, respectively. Since 2013, she has been the Head of the master's degree and the Manager of the Virtual Reality Projects at the Department of Computer Science, Centro Universitario de Tecnología y Arte Digital (U-tad), Madrid, Spain. She is currently a Professor and a Researcher at U-tad. Her primary research areas are virtual reality, haptics devices, data visualization, gamification, and the benefits of virtual immersion in health.