DeepFair: Deep Learning for Improving Fairness in Recommender Systems

Jesús Bobadilla*, Raúl Lara-Cabrera*, Ángel González-Prieto, Fernando Ortega

ETSI Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid (Spain)

Abstract

The lack of bias management in Recommender Systems leads to minority groups receiving unfair recommendations. Moreover, the trade-off between equity and precision makes it difficult to obtain recommendations that meet both criteria. Here we propose a Deep Learning based Collaborative Filtering algorithm that provides recommendations with an optimum balance between fairness and accuracy. Furthermore, in the recommendation stage, this balance does not require an initial knowledge of the users' demographic information. The proposed architecture incorporates four abstraction levels: raw ratings and demographic information, minority indexes, accurate predictions, and fair recommendations. Last two levels use the classical Probabilistic Matrix Factorization (PMF) model to obtain users and items hidden factors, and a Multi-Layer Network (MLN) to combine those factors with a 'fairness' (ß) parameter. Several experiments have been conducted using two types of minority sets: gender and age. Experimental results show that it is possible to make fair recommendations without losing a significant proportion of accuracy.

Keywords

I. Introduction

FAIRNESS in Recommender Systems (RS) is a very important issue, since it is part of the path to get a fair society. Nowadays, recommendations come to us from a variety of online services such as Netflix, Spotify, TripAdvisor, Facebook, Amazon, etc. All these services rely on hybrid RS [1] whose kernel is the Collaborative Filtering (CF). CF data is the set of the users' preferences on the items: tens or hundreds of millions of ratings, likes, clicks, etc. It seems great, since in theory, the more the data the better the recommendations; unfortunately, this data is usually biased [2]–[3] and minority groups are the most damaged ones. Common minority groups are female (vs. male) and senior (vs. young); both groups tend to receive unfair recommendations from online services. This situation has a perverse effect: a cycle that feeds back, where unfair recommendations make minority users to lose confidence in the system, to decrease their interaction and, thus, to receive even more unfair recommendations. The time has come to increase research in fair RS to reduce the digital gap [4]–[5] between minority and non-minority groups.

CF RS research has been traditionally focused in accuracy improvement [6], although some other objectives have increased the research attention in the last years: novelty [7], reliability [8], diversity [9] and serendipity [10]–[11] among them. Surprisingly, fairness has not been a main objective in the RS priorities. One of the reasons is the idea that improving fairness does not lead us to more valued recommendations, such as accuracy, novelty or diversity clearly do.

Nevertheless, society needs to point in the opposite direction [12], and a set of new quality goals are growing [13]: relevance, fairness, and satisfaction among them. The historical development of CF has not helped to the fairness research, either: when the k-Nearest Neighbors (kNN) algorithm [14] dominated the field, it was less likely that a reduced set of neighbors produced biased recommendations. However, in a very short time the Matrix Factorization (MF) method prevailed as standard, and the fairness goal relevance grew up [15]. MF makes a compressed version of the ratings that belong to the dataset, catching the essence of them. The compressed models are sensible to the data biases such as the demographic ones: gender, age, etc. [16] making fairness a particularly relevant goal.

As a consequence of the CF research evolution, existing publications to improve fairness using the kNN algorithm are scarce; as an example, in [17] authors look for balanced neighborhoods as a mechanism to preserve personalization (accuracy) while enhancing the recommendations fairness. It is also remarkable the differentiation that takes place, in this context, between consumer-centered and provider-centered fairness. Fairness has been studied in the CF context in two main directions: a) finding that data biases really generates unfair recommendations, and b) providing quality measures or methods to quantify recommendations fairness. From the first block, in [18] authors argue that improving recommendations diversity leads to discrimination among the users and unfair results. The response of CF algorithms to the demographic distribution of ratings is studied in [19]; they find that common CF algorithms differ in the gender distribution of their recommendation lists. A preliminary experimental study on synthetic data was conducted in [20], where conditions under which a recommender exhibits bias disparity and the long-term effect of recommendations on data bias are investigated. From the second block (quality measures) in [21] they claim that biased data can lead CF

* Corresponding author.

E-mail addresses: jesus.bobadilla@upm.es (J. Bobadilla), raul.lara@upm.es (R. Lara-Cabrera).

methods to make unfair predictions for users from minority, and they propose new metrics that help reducing fairness. Disparity scores has also been proposed [18] to obtain fairness measures. Bias disparity can be defined as "how much an individual's recommendation list deviates from his or her original preferences in the training set" [20], whereas average disparity measures how much preference disparity between training data and recommendation list for the minority group of users is different from that for the non-minority group [22]. Fairness quality results in our paper implement these concepts.

Fairness in information retrieval has been focused on study data bias more than acting on the machine learning models: "teams typically look to their training datasets, not their machine learning models, as the most important place to intervene to improve fairness in their products" [12]. The machine learning achievements in the fairness issue have been reviewed in [23], where they find some "frontiers" that machine learning has not crossed yet. The MF disadvantages in CF have been studied in [21], where authors state that the MF model cannot manage the two main types of imbalanced data: population imbalance and observation bias. RS fairness has been even less covered in Deep Learning (DL) than in machine learning; as an example, in this current survey of RS based on DL [24] the fairness goal is not mentioned, not even in its "possible research directions" section. The same happens with the current review paper [25] where fairness is not mentioned despite the complete set of DL-based RS included in the publication. In fact, state of the art research in this area is focused on accuracy improvements [26]–[27] and it has not covered this subject. To afford a DL-based and fair RS is difficult due to the neural black box model [28], that is not easy to explain or vary. Nevertheless, to tackle CF fairness using DL has the advantage of providing a starting base where accuracy is high [29]; it is particularly convenient since the increase in fairness usually leads to the decrease in accuracy.

For the stated reasons, the hypothesis of this paper claims that it is possible to design a DL architecture that provides fair CF recommendations at the cost of reasonable decreases of accuracy. A DL approach to obtain fair recommendation provides a novel scenario in the RS field. This scenario opens the door to reach accurate and fair predictions, but it is not a straightforward how to make the architectural design: we have to deal not just with raw ratings data, but also with the necessary demographic information to determine the target minority groups: female vs. male, senior vs. young, etc. Moreover, the neural network learning model cannot be changed as easily as the kNN approach or even some machine learning algorithms. For all this, the proposed DL approach relies on an enriched set of input data and a tailored loss function that minimizes not only the accuracy errors but also the fairness ones. Fairness errors can be measured using the disparity scores concept [18], but how these scores are fed is a research open issue.

The proposed neural network learns from data that accomplish the current disparity concept: "deviation from the list of recommendations and the training data". We have specified it into two related indexes: the items one, that assigns a minority value to each item (e.g. a femininity value to a film, that depends on the female and the male preferences on this movie), and the users one, that assigns a minority value to each user (e.g. a femininity value to a user, that depends on the femininity of the items preferred for this user). Once both indexes have been set, it is possible to design a neural network loss function that rewards equality between each user minority value and his/her recommended items minority values. An additional design decision we have taken is to choose a regression approach [8] instead a classification one [27]: since we need to simultaneously minimize accuracy and fairness errors in the loss function, it is straightforward to pack them into a combined value so that the neural network provides us with balanced fairness/accuracy regression results. Finally, we have chosen a combined

MF and DL approach [8] [30]; this design allows us to decouple the accuracy and the fairness abstraction levels by assigning accuracy to the MF and fairness to the DL stage.

A main advantage of the proposed architecture is that, once the model has learned, recommendations can be made to users that do not have associated demographic information; that is: we can fairly recommend to users without knowing its minority nature. It is possible because the neural network can learn the minority pattern in the same process that it learns to minimize the accuracy/fairness prediction error. It is a commercial advantage since many users avoid filling in their personal data.

In summary, designing recommender systems that are capable of providing fair recommendations without a high loss of accuracy is a significant contribution not only to the field of fairness in the ML-based RS, but also to the DL-based ones. As mentioned above, the former has merely proposed metrics for measuring unfairness in recommendations while the latter does not even consider fairness as a current goal.

As already discussed in Section I, existing recommender systems are primarily focused on providing recommendations as accurately as possible. Recommendations provided to minority groups of users are currently very unbalanced due to the RS datasets bias, and it leads to unfair recommendations made to the groups. State of the art in RS fairness is centered in memory-based methods, that are no longer commercially used due to their lack of accuracy. Research in model-based fair methods is scarce, and it is focused on trust-based systems, that usually require social information not available in most of the commercial RS. Our approach is a model-based one, making use of DL technology and which only needs the ratings information.

Based on the above, this paper's main research objective is to find a balance between accuracy and fairness in the recommendations made to the RS users. To this end, we propose a DL CF approach that can automatically adjust fairness and accuracy in recommendations.

The rest of the paper has been structured as follows: in Section II the proposed method is explained and the experiments design is defined. Section III shows the experiments' results and their discussions. Finally, Section IV contains the main conclusions of the paper and the future work.

II. Materials and Methods

This section is devoted to describing our proposed method as well as the experimental setup we have used to evaluate it.

A. Proposed Method

The proposed architecture incorporates four different abstraction levels, as depicted in Fig. 1, to get the desired fair recommendations: a) raw ratings and demographic information, b) minority indexes for both users and items, c) accurate predictions, and d) fair recommendations. Level 'b' just makes some simple statistical operations by combining ratings and demographic information; level 'c' uses the classical Probabilistic Matrix Factorization (PMF) model in order to obtain users and items hidden factors; finally, level 'd' makes use of a Multi-Layer Network (MLN) to combine hidden factors and a 'fairness' (ß) parameter. This MLN generates the desired fair recommendations.

We will develop each of the three levels that make up our architecture: first, in the lowest level we create two related indexes: 1) *items minority index* (*IM*), and 2) *users minority index* (*UM*). The *IM* index will assign a minority value to each item in the dataset, e.g. when the minority group is 'female' we could call to the index 'femininity'. It will contain values [-1, 1] where negative ones mean feminine preferences and positive ones mean masculine preferences.
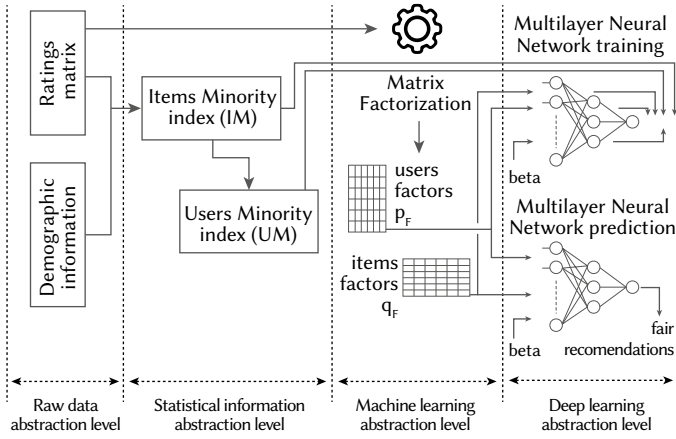
Fig. 1. Architecture overview.

Then, when an item has been assigned a negative value it means that it has been rated better by women than men. Once the *IM* index has been created it contains the minority values of all the items. By using the *IM* index, we will create the *UM* index. The *UM* index will assign a minority value to each user in the dataset. It also will contain values [-1, 1], where negative ones mean minority preferences and positive ones mean not minority preferences (masculine, in our example). A user assigned a negative *UM* value means that this user prefers negative *IM* items, and vice versa. Please note that, on many occasions, female users may have assigned positive *UM* values and male users may have assigned negative *UM* values, since there exist women with masculine preferences and men with feminine ones; same as young and older persons or any other minority versus majority groups. Thus, an important concept is that both the *IM* and *UM* indexes do not contain disjoint minority/majority demographic values; they contain minority/majority preferences. This design accurately fits the existing diversity of preferences contained in the CF based RS.

Now, we will explain the *IM* and *UM* indexes design that we will take as a base to get fair recommendations in the DL stage. First, we will differentiate between relevant and not relevant votes: relevant votes are those that indicate that the user liked the item; conversely not relevant votes (in our context) are those that indicate that the user did not liked the item. There can also exist votes that indicate indifference on the part of the user. In our formulation, relevant and not relevant votes are chosen by means of two thresholds; e.g. in a dataset where votes must be in the set {1, 2, 3, 4, 5} we can establish 4 as the relevant threshold and 2 as the non-relevant threshold. In this way the relevant set is {5, 4}, the non-relevant set is {2, 1} and {3} would be the 'indifference' set.

We define the *IM* index (11) for each item *i* as the majority score of *i* minus the minority score of *i*. The majority score (resp. minority score) of the item *i* is the number of majority (resp. minority) users that voted *i* as relevant minus the number of majority (resp. minority) users that voted *i* as non-relevant, divided by the total amount of majority (resp. minority) users that did not consider *i* as indifferent, see Equations (9) and (10) (resp. (7), (8)). When the proportion of the minority user preferences exceeds the proportion of the non-minority ones, the *IM* index values are negative. In the gender example, equation (11) can be read as: "proportion of males that liked item *i* minus males that did not like it, minus the proportion of females that liked item *i* minus females that did not like it". We have also set a minimum number of 5 votes to consider both the minority and non-minority sides of equation (11).

Once the *IM* index has been created, we can use it to establish the *UM* index values. Each *UM* value corresponds to a user of the RS dataset, and it provides the minority value of the user. Each user minority value will be defined by the minority of his/her preferences:

to obtain each user *UM* value we just make the average of the *IM* minority values of the items that the user has voted, weighting each *IM* minority value with its corresponding user rating. Equation (13) models the explained behavior.

Let Θ↑ be the like threshold (1)

Let Θ↓ be the dislike threshold (2)

Let I be the set of items in the dataset (3)

Let U be the set of users in the dataset (4)

We will assign the following meanings to super index numbers: *m* for minority and *M* for non-minority:

Let U^m be the set of minority users (5)

Let U^M be the set of non-minority users (6)

Let $U_\uparrow(i) = \{u \in U \,|\, r_{u,i} \geq \Theta_\uparrow\}$ be the set of users who liked item i (7)

Let $U_\downarrow(i) = \{u \in U \,|\, r_{u,i} \leq \Theta_\downarrow\}$ be the set of users who did not like item *i* (8)

The majority score is

$$\mathfrak{I}^{\mathfrak{M}}(i) = \frac{|U_\uparrow(i) \cap U^M| - |U_\downarrow(i) \cap U^M|}{|U_\uparrow(i) \cap U^M| + |U_\downarrow(i) \cap U^M|}$$ (9)

The minority score is

$$\mathfrak{I}^{m}(i) = \frac{|U_\uparrow(i) \cap U^m| - |U_\downarrow(i) \cap U^m|}{|U_\uparrow(i) \cap U^m| + |U_\downarrow(i) \cap U^m|}$$ (10)

The *IM* and *UM* indexes are

$$IM(i) = \mathfrak{I}^{\mathfrak{M}} - \mathfrak{I}^{m}$$ (11)

$$IM = \{(i, IM(i)) \,|\, i \in I\}$$ (12)

$$UM(u) = \frac{\sum_{\{i \in I | r_{u,i} \neq \circ\}} \left(r_{u,i} - \frac{\Theta_1 + \Theta_2}{2} \cdot IM(i)\right)}{\left(N - \frac{\Theta_1 + \Theta_2}{2}\right) \cdot |\{i \in I | r_{u,i} \neq \circ\}|}$$ (13)

$$UM = \{(u, UM(u)) \,|\, u \in U\}$$ (14)

where ∘ means "not voted item" and *N* is the maximum possible vote.

Fig. 2. Data-toy example to get *IM* and *UM* minority values.

Fig. 2 shows a data-toy example containing five users and four items. We will suppose that women are a minority group in this RS, compared to the men. We can observe that '*item a*' is clearly 'masculine', since it has been voted as 'relevant' for all the male users and it has been voted as 'non-relevant' for all the female users. The opposite situation is stated in '*item b*': it is a 'feminine' item according to the female relevant votes and the male non-relevant ones. '*Item c*' is quite masculine, although a female user liked it. Finally, '*item d*' shows the opposite situation to '*item c*'. According to it, the proposed *IM* equations return the following item minority values:

{⟨item a, 1⟩,⟨item b, −1⟩,⟨item c, 0.5⟩,⟨item d, −0.6⟩}

that fits with the explained behavior (Table I). Once the items' minority values *IM* are obtained, we can get the users minority ones (*UM*). First, we can observe how '*male 2*' and '*male 3*' users in the data-toy example have casted very 'masculine' ratings, since they have voted 'relevant' to the more 'masculine' items, and 'non-relevant' to the more 'feminine' items. This is not the case for the '*male 1*' user, that has a 'relevant' vote casted on the 'feminine' '*item d*'. The female users comparative is more complicated: '*female 1*' has casted all her votes in a 'feminine' way, whereas the '*female 2*' vote to the 'masculine' '*item c*' was 'relevant'; nevertheless, the '*female 2*' feminine votes are higher than the '*femenine 1*' ones. In this way, we expect the following results: a) positive *UM* values to male users and negative ones to female users, and b) a more 'minority' (feminine) value be assigned to '*male 1*' than to '*male 2*' and '*male 3*' . Table I shows the Fig. 2 data-toy *IM* results and Table II shows the *UM* ones.

TABLE I. Data-toy IM Results

Item	Value
a	[(3−0)−(0−2)]/5 = 1
b	[(0 − 3) − (2 − 0)]/5 = −1
c	[(2−0)−(1−1)]/4 = 0.5
d	[(1 − 2) − (2 − 0)]/5 = −0.6

TABLE II. Data-toy UM Results

Item	Value
male 1	(5−3)·1+(2−3)·(−1)+(4−3)·(−0.6)=2.4/5=0.48
male 2	(5−3)·1+(2−3)·(−1)+(4−3)·0.5+(2−3)· (−0.6)=4.1/5=0.82
female 1	(2−3)·1+(4−3)·(−1)+(1−3)·0.5+(4−3)·(−0.6)= −3.6/5=−0.72
female 2	(1−3)·1+(5−3)·(−1)+(4−3)·0.5+(5−3)·(−0.6)= −4.7/5= −0.94
male 3	(4−3)·1+(1−3)·(−1)+(4−3)·0.5+(2−3)·(−0.6)= 4.1/5=0.82

Our architecture uses the PMF method to reduce the ratings matrix dimension and to get a condensed knowledge representation. From the condensed results we will be able to make accurate predictions. Equations (15)-(24) show the model formalization: the original ratings matrix is condensed in the two lower dimension matrices P and Q (equation (15)). P is the users' matrix and Q is the items' matrix. Both P and Q have a common dimension of F hidden factors, where $F \ll M$ and $F \ll N$ (note that M is numbers of users, and N the number of items). Once the model has learnt, each user will be represented by a vector $\overrightarrow{p_u}$ of F factors, and each item will be also represented by a vector $\overrightarrow{q_i}$ of F factors. Each prediction of an item u to a user i is obtained by processing the dot product of these vectors (equation (16)). Since the users and the items hidden factors share the same semantic, predictions will be relevant when high values (positive or negative) of the factors line up in each user and item.

$$R \approx \hat{R} = P \cdot Q^t \tag{15}$$

$$\widehat{r_{u,i}} = \overrightarrow{p_u} \cdot \overrightarrow{q_i} = \sum_{f=1}^{F} p_{u,f} \cdot q_{i,f} \tag{16}$$

The P and Q factors will be used in our architecture to feed the DL process input as well as to set the output target labels. Factors are obtained by means of the gradient descent algorithm. The loss function just minimizes the prediction error: the difference between the predicted value and the existing rating (equation (17)).

$$\text{loss}(u, i) = \left(r_{u,i} - \widehat{r_{u,i}}\right)^2 \tag{17}$$

In order to achieve the gradient descent minimization process we obtain the partial loss derivatives: $\delta\text{loss}/\delta\overrightarrow{p_u}$ and $\delta\text{loss}/\delta\overrightarrow{q_i}$ (equations (18) and (19)).

$$\frac{\delta\text{loss}}{\delta\overrightarrow{p_u}} = \frac{\delta}{\delta\overrightarrow{p_u}}\left(r_{u,i} - \overrightarrow{p_u} \cdot \overrightarrow{q_i}\right)^2 = -2\overrightarrow{q_i} \cdot \left(r_{u,i} - \overrightarrow{p_u} \cdot \overrightarrow{q_i}\right) = -2\overrightarrow{q_i} \cdot e_{u,i} \tag{18}$$

$$\frac{\delta\text{loss}}{\delta\overrightarrow{q_i}} = \frac{\delta}{\delta\overrightarrow{q_i}}\left(r_{u,i} - \overrightarrow{p_u} \cdot \overrightarrow{q_i}\right)^2 = -2\overrightarrow{p_u} \cdot \left(r_{u,i} - \overrightarrow{p_u} \cdot \overrightarrow{q_i}\right) = -2\overrightarrow{p_u} \cdot e_{u,i} \tag{19}$$

This gives rise to the corresponding gradient descent factors update Equations (20) and (21).

$$p'_{u,f} = p_{u,f} + 2\gamma \cdot q_{f,i} \cdot e_{u,i} \tag{20}$$

$$q'_{f,i} = q_{f,i} + 2\gamma \cdot p_{q,f} \cdot e_{u,i} \tag{21}$$

Finally, we can add a regularization term for controlling the growing of the factors during the learning process, which gives rise to the loss function and the update rules shown in Equations (22) to (24).

$$\text{loss}(u, i) = \left(r_{u,i} - \widehat{r_{u,i}}\right)^2 + \frac{\lambda}{2}\sum_{f=1}^{F}(|P^2| + |Q^2|) \tag{22}$$

$$p'_{u,f} = p_{u,f} + \gamma\left(2q_{f,i} \cdot e_{u,i} - \lambda \cdot p_{uf}\right) \tag{23}$$

$$q'_{f,i} = q_{f,i} + \gamma\left(2p_{uf} \cdot e_{u,i} - \lambda \cdot q_{f,i}\right) \tag{24}$$

The highest semantic level of the proposed architecture is based on an MLN. Our MLN (see Fig. 3) model will take input vectors containing the following information: a) user hidden factors p_u, b) item hidden factors q_i, and c) $\beta \in [0, 1]$ value. The β parameter is used to balance fairness and accuracy in predictions and recommendations: high β values will enhance accuracy, whereas low β values will enhance fairness. This balance is a key objective of our method: "To obtain fair recommendations just losing an acceptable degree of accuracy". Please note that we do not include demographic information to feed the MLN input, so once the MLN has learnt it will be able to make fair recommendations to users that have not filled demographic forms asking for gender, age, etc. This is an important commercial advantage, since it allows to make better marketing processes, to improve fairness, to focus prediction tasks, etc. It is also a challenge to the proposed machine learning framework because it is more difficult to increase recommendation fairness when demographic data is missing. The learning process has been based on input vectors containing the specified three information sources: : ⟨$p_{u,f}$, $q_{f,i}$, β⟩. We have set 11 input vectors to the MLN for each (user u, item i) rating of the dataset:

⟨$p_{u,f}$, $q_{f,i}$, 0.0⟩, ⟨$p_{u,f}$, $q_{f,i}$, 0.2⟩, ..., ⟨$p_{u,f}$, $q_{f,i}$, 1.0⟩

The objective is to teach to the neural network on eleven fairness levels for each rating, as it can be seen in the left side of Fig. 3.

Once the MLN input vectors have been established, it is necessary to define their corresponding output labels to let the back-propagation algorithm learn the pattern. In our case we will design a loss function that minimizes both the prediction error and the fairness error. Equation (25) shows the typical prediction loss function, as we did in

equation (17). We define the fairness error as the distance between the user's minority and the item's minority; e.g. films recommended to a user (male or female) with an assigned 0.8 *UM* femininity value should be as similar as possible to a 0.8 IM in order to fit in the fairness issue. Since *UM* and *IM* vector values do not have the same distribution, we will apply a [0, 1] normalization in both of them and we will use the *UM'* and *IM'* names for the normalized versions. Then, to obtain the fairness error we establish equation (26). Finally, to combine equation (25) (accuracy) and equation (26) (fairness) the β parameter is added (equation (27)).

$$e_{u,i}^{accuracy} = \left(r_{u,i} - \sum_{f=0}^{F} p_{u,f} \cdot q_{i,f}\right)^2 \tag{25}$$

$$e_{u,i}^{\text{fairness}} = (IM_i' - UM_u')^2 \tag{26}$$

$$\text{loss}_{u,i} = \beta \cdot e_{u,i}^{\text{accuracy}} + (1 - \beta) \cdot e_{u,i}^{\text{fairness}} \tag{27}$$
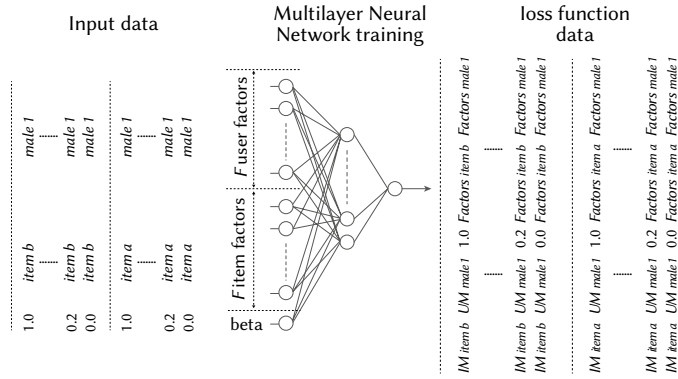

Fig. 3. Training information for the proposed MLN.

In the feed forward prediction stage, for each testing input data $\langle p_{u,f}, q_{f,i}, \beta \rangle$, the proposed neural network returns a real number whose meaning is the predicted loss error for the item *i* to the user *u* recommendation. The lower the predicted loss error, the better the combined ⟨accuracy, fairness⟩ values given the chosen ß accuracy vs. fairness balance. Once the network has learnt and the RS is in production phase, to make recommendations to an active user *u*, first we fix the ß value and then we feed the MLN with all the inputs $\langle \vec{p}_u, \vec{q}_i, \beta \rangle$ where *i* runs over the set of items that the user *u* has not voted (equation (28)).

$$X = \left\{\langle p_{u,f}, q_{f,i}, \beta \rangle \mid u \in U, i \in I, r_{u,i} \neq \circ \right\} \tag{28}$$

The set of N recommendations for the user $u, Z_{u,N}$ is the collection of N items with minimum loss function $h(\vec{p}_u, \vec{q}_i, \beta)$, where the *h* function represents N feed forward operations.

B. Experimental Setup

Experiments have been conducted using a well-known dataset called MovieLens 1M [31]. It contains 1,209,000 votes, 6040 users and 3952 items. We have used eleven different values of the β parameter (from 0.0 to 1.0, step 0.2); consequently, the MLN has been trained using 13,299,000 input vectors and output target values. Training, validation, and test sets have been established: 70%, 10% and 20%, respectively. The PMF process has been run using 30 hidden factors (*F*), 80% training ratings, 20% testing ratings. Please note that these are the MLN parameters of the proposed method, different to the previously ones specified for the DL stage. The designed MLN contains an input layer of 30+30+1 = 61 values (Fig. 3). The first MLN internal layer has been set to 80 neurons (*relu* activation), followed by a 0.2 dropout layer to avoid overfitting. The second internal layer has been set to 10 neurons (*relu* activation) and, finally, the output layer contains just

one neuron with no activation function. The chosen loss function has been *mae* and the optimizer *rmsprop*.

III. Results

The experiments we have conducted are:
- Item Minority Index (IM) and User Minority Index (UM) distributions.
- User Minority Index (*UM*) comparative between each minority and non-minority group.
- Fairness prediction improvement using the heuristic algorithm.
- Fairness recommendation improvement using the heuristic algorithm.
- Fairness error and accuracy error for recommendations using the proposed DL architecture.

This section contains a subsection for each of the above set of performed experiments. We have selected two types of minority sets: a) gender: female vs. male, and b) youth: young vs. senior. Results are provided showing both minority types in two separated graphs of each figure. The MovieLens dataset, like in many other CF RS happens, is biased towards male and young people.
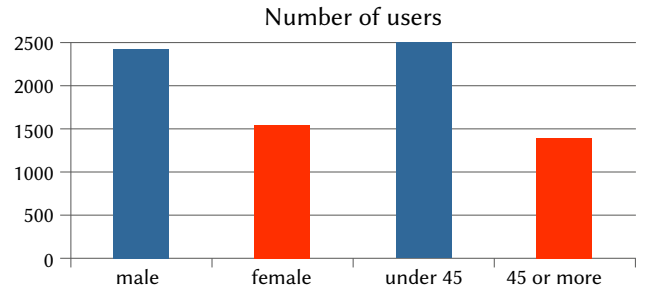

Fig. 4. Proportion of users in the MovieLens gender and age minority and non-minority groups.
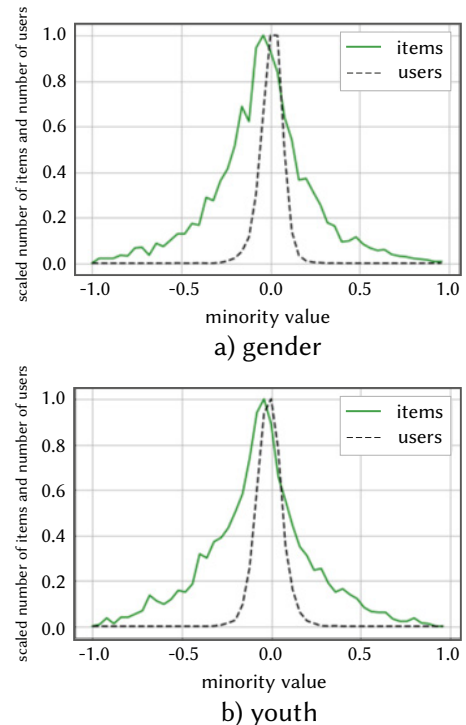

Fig. 5. Item Minority Index (*IM*) and User Minority Index (*UM*) distributions.

Thus, the chosen minority types are relevant and representative for this experimental study. Specifically, the MovieLens dataset contains more males than females; most of them are under 45 years old. Fig. 4 shows the proportions. Equations (11) and (13) describe both indexes behavior. The IM index semantic is simple and convincing, but it is necessary to be aware that we are not working with absolute values: in order to prevent data biases and to maintain the index values in a bounded range, we are working with preferences proportions; e.g. "proportion of male users that liked the items minus proportion of female users that liked the item". Since we expect a significant number of items that both minority and non-minority groups simultaneously like or dislike, *IM* proportions will be similar for both groups and consequently a significant number of *IM* values will concentrate around the 0.0 value. Fig. 5 shows the items and users minority indexes distributions, both for the gender and the youth minority groups.

The *UM* index values are obtained from the ratings that each user has casted to the items and from the *IM* value of each of those items. We can see in Fig. 5 that the users *UM* indexes (both for gender and youth) have a large concentration of values around 0. It provides us an important conclusion:

"In the reference dataset, most users have similar preferences regarding to the chosen minority groups". Looking at the *UM* distributions we can also yield another main conclusion: "Although users have similar preferences, there is a clear separation between minority groups" (left and right side of the graphs). Since the *UM* index is only used to feed internal DL processes the relevant information here is the proportion of the differences between values, and not their absolute values.

A. User Minority Index (UM) Comparative Between Each Minority and Non-minority Group

In the above section we have confirmed two facts: 1) Users preferences are similar, even if they belong to different minority groups, and 2) Despite the previous conclusion, there is room to find minority behaviors of users. In this section we deepen in the minority *UM* values of users, to clear out our specific groups: male vs. female and senior vs. young. Fig. 6 shows the results: we can observe, in both cases, that groups have different behaviors and that they share a relevant number of preferences. Groups present different behaviors because they do not completely intersect their user minority values; as expected, minority groups return a mean less than zero whereas non-minority groups return it greater than zero. Groups share a relevant number of preferences because there exist a proportion of minority and non-minority users that share *UM* values (areas around 0.0 under both curves).

number of users that have been classified as belonging to the minority or to the non-minority groups. Minority users (female, young) have an expected *UM* index less than zero. Non-minority users (male, senior) have an expected *UM* index greater than zero.
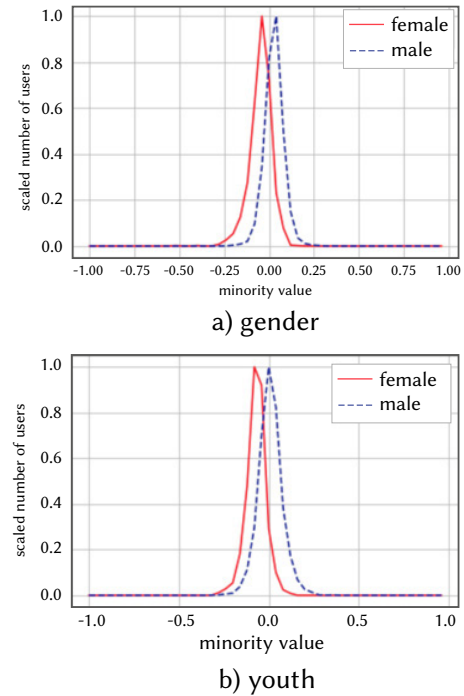

a) gender

b) youth

Fig. 6. User Minority Index (*UM*) comparative.

B. Fairness Prediction Improvement Using a Heuristic Algorithm

Fig. 6 and table IV show us that most of the users are correctly grouped attending to their *UM* indexes, especially for seniors and males. They also show a considerable number of cases incorrectly classified, particularly for young and female groups. In this situation, we will obtain predictions from the test set and then check their quality in terms of the *IM* index. Table IV contains these experiments results: the IM averages fit the expected ranges (negative *IM* average for minority users, and positive *IM* average for non-minority users). Despite these positive results, ranges can be too narrow to ensure fair predictions. On the other hand, there will be situations in which it is intended to force the recommendations of an RS to move towards minority items, or perhaps towards majority items, depending on the type of users and/or the company policy.

TABLE III. Users Classification Attending to the Minority/non-minority Groups

group	type	correct	incorrect	correct %
gender	female	1147	562	67.11
	male	3648	683	84.22
youth	senior	1231	195	86.32
	young	3144	1470	68.14

Due to the explained results, we can confirm that there is a not negligible proportion of minority users with non-minority preferences and vice versa. In any case, it varies depending on the specific minority group. As an example, we can observe in Fig. 6 how senior users have much less non-minority preferences than female ones, since there are small amounts of senior users whose minority value is greater than zero. Results show the convenience of using modern machine learning approaches to make fair recommendations to those users that share minority and non-minority preferences. Table III shows the specific

TABLE IV. Averaged IM Values for the Predictions Made to Each Users' Group

	female	male	senior	young
IM mean	-0.014	0.041	-0.025	0.028

By filtering on the *IM* index, we can discard those predictions greater than a negative threshold and, in this way, increase the proportion of minority predictions. In the same way we can filter those predictions less than a positive threshold to increase the proportion of majority predictions. We have performed this experiment, calling alpha to the threshold. We can observe the expected behavior in Fig. 7, where growing minority (and majority) *IM* values are obtained in predictions when the alpha parameter increases. It also can be seen that the non-minority users (male, young) always obtain better predictions due to the RS datasets biases. Finally, we can state that, in this case, minority values can reach the starting majority ones by using low values of the alpha parameter (0.025 for gender and 0.05 for age).
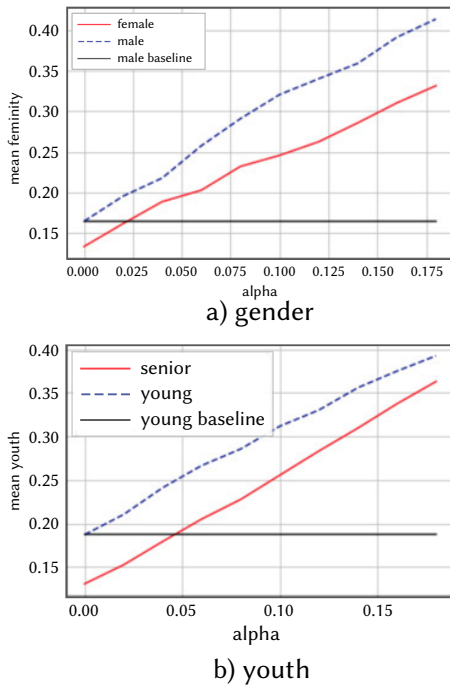
a) gender

b) youth

Fig. 7. Groups quality improvement by filtering predictions. x axis: alpha values used to filter on the IM items index. y axis: averaged minority of the filtered predictions. Minority (female, senior) curves are drawn using their absolute values.
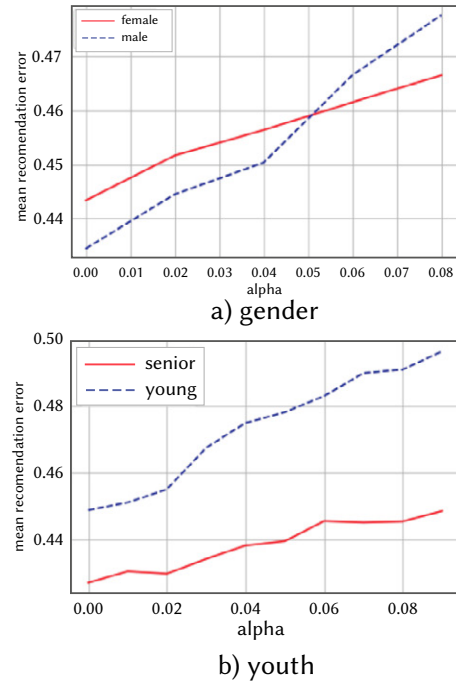

a) gender

b) youth

Fig. 8. Recommendation quality obtained by filtering predictions. x axis: alpha values used to filter on the IM items index. y axis: averaged error of the N recommendations. Lower error values are the better ones.

C. Fairness Recommendation Improvement Using the Heuristic Algorithm

The previous section results show that it is possible to provide a heuristic method to improve recommendations fairness. To conduct the experiment, from the alpha filtered predictions (Fig. 7), we extract the N ones that provide higher prediction values, as usual in the CF operation. Thus, the complete recommendation method involves three sequential phases: 1) to obtain all the prediction value, minority value pairs, 2) to filter the pairs according to the minority threshold alpha parameter and each *minority value*, and 3) to select the N filtered predictions that have the N highest *prediction value* values.

Results in Fig. 8 show the existing correlation between recommendation errors and each chosen alpha value: the highest the alpha value, the better the recommendations fairness (Fig. 7), but as expected, also the worst the recommendation accuracy (higher error values in Fig. 8). Of course, we pay an accuracy price when we force fairer recommendations.

We have chosen a value of N = 10 recommendations to process the set of experiments. From Fig. 7 it can be observed that in the `youth' experiment our method provides better results (lower errors) for the minority `senior' group than for the `young' one. This is a good indication of the proposed heuristic method functioning. The `gender' experiments provide improvement in the minority female group from a specific value threshold (alpha = 0:05). All these results are consistent with Tables II and III values.

D. Fairness Error and Accuracy Error for Recommendations Using the Proposed DL Architecture

Results obtained in the previous subsection tell us that we have designed a method that correctly provides fair recommendations. It is a simple, functional, and easy to implement machine learning approach. Nevertheless, it has some drawbacks:

- Choosing the adequate parameter alpha requires a fine-tuning process.
- Since the parameter alpha sign (less than or greater than

zero) depends on the minority or non-minority nature of the recommended user, this recommendation method can only be applied to users with associated demographic information.

This subsection provides a DL approach that works without the above drawbacks. This method only needs the parameter β: it is used to select the accuracy vs. fairness balance. The β range is $[0, 1]$, whether 0 means 100% fairness and 0% accuracy, and 1 means 100% accuracy and 0% fairness. As it can be seen, to choose a β value is straightforward and intuitive. Moreover: the chosen β value does not change when the user is a minority one or he is not.

The proposed DL recommendation method explained in section 2 returns the results shown in Fig. 9. Graphs on the left of the figure contain the main information. Graphs on the right are $[0, 1]$ scaled to find the optimum accuracy vs. fairness balances. The averaged error of the recommendations (equation (25)) is plotted using black lines. Dotted and dashed lines show the minority errors (equation (26)); that is: the distance between the minority value of each recommended user (UM) and the average of the minority values (IM) of their N recommended items. We are looking for recommended items in the minority range of the user; e.g. if a user (male or female) has an $UM = 0.7$ (quite masculine), recommended items near $IM = 0.7$ are the fairest ones, and they generate a low minority ('femininity') error.

'Gender' results are shown in the top-left graph of Fig. 9: as expected, accuracy increases (error decreases) as β increases (more importance to accuracy). The price to pay for this accuracy improvement is the simultaneous increase in the fairness error values. As β decreases (more importance to fairness), the opposite happens: higher prediction errors and lower fairness errors. 'Youth' results are shown in the low-left graph of Fig. 9: curve trends are like the 'gender' results. Graphs on the right of Fig.9 show the same results by using a normalized y axis: in this way we can find the optimum β values to balance accuracy and fairness in the recommendation task. To optimize results in this experiment, it is necessary to choose a β = 0.4 value: a balanced selection, something scored to the fairness objective. This result tells us that the balanced option (β = 0.5) can be the default one.
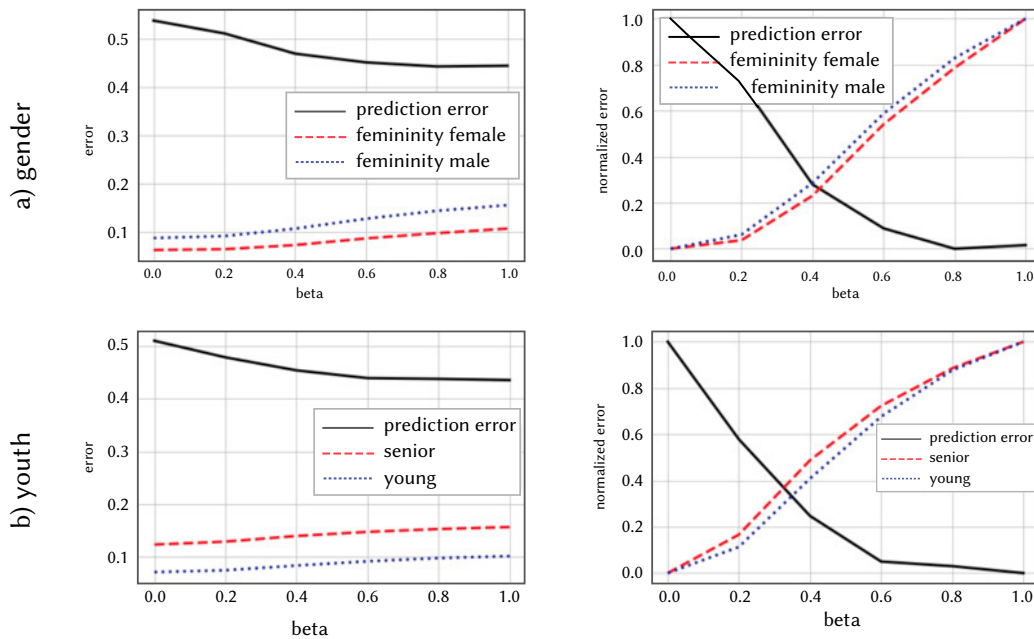
Fig. 9. Recommendation results using the proposed DL approach. y axis: averaged N = 10 recommendations error (normalized in the right graphs); x axis: β balance between fairness and accuracy (0.0 means 100% fairness and 0% accuracy, and 1.0 means 100% accuracy and 0% fairness).

IV. Conclusions

Attending to the obtained results, it is understood that designing methods to improve CF fairness is not a simple task, but it is possible to take it out. Due to the fact that an appreciable proportion of minority and non-minority users share preferences it is necessary to make use of modern machine learning approaches in order to make fair recommendations not only to the 'purest' minority or non-minority users, but also to the users that mix some proportion of minority and non-minority preferences.

State of the art shows a lack of DL approaches to tackle fairness in RS, probably due to the neural networks black box model. The proposed method in this paper relies on an original loss function and input data to balance fairness and accuracy. This method combines several abstraction levels, and it can serve as baseline to DL future works in the field. An original architecture is provided, where machine learning and DL models are combined to obtain balanced accuracy vs. fairness recommendations. The architecture is based on two basement levels: statistical and machine learning, that provide the necessary information to train the DL model which constitutes the third architectural level. The proposed DL method provides a modern approach to tackle fairness in RS. We can easily balance accuracy and fairness, or we can automatically select the optimum tradeoff. That is to say: the proposed method manages the inherent loss of accuracy when fairness is increased. Additionally, once the neural network is trained using demographic information, it can predict and recommend to users whose demographic information is unknown.

Results show adequate trends in the tested quality measures: improvement in fairness at the cost of an expected worsening in accuracy. The proposed machine learning-based heuristic approach and the DL model return similar quality results. Nevertheless, the proposed DL method does not need demographic information in the recommendation feed-forward process. It also can better balance and automatically balance fairness and accuracy.

The main contributions of the paper are:

- A novel Deep Learning based Collaborative Filtering algorithm that provides recommendations with an optimum balance between fairness and accuracy.

- Our proposed method does not require an initial knowledge of the users' demographic information.

- The proposed method relies on an original loss function and input data to balance fairness and accuracy. Also, it can manage the inherent loss of accuracy when fairness is increased, balancing accuracy and fairness of the recommendations.

Proposed future works are: a) architecture simplification, by removing the MF and transferring its functionality to the DL model, b) items and users minority indexes redefinition to better catch the minority versus non-minority differences, c) testing the methods behavior in a variety of CF datasets, d) extending the experiments to different demographic groups (nationality, profession, studies), and e) testing the architecture on not demographic groups (users that share minority preferences).

References

[1] E. C̨ ano, M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent Data Analysis*, vol. 21, no. 6, 2017, pp. 1487–1524.

[2] A. Bellogín, P. Castells, I. Cantador, "Statistical biases in Information Retrieval metrics for recommender systems," *Information Retrieval Journal*, vol. 20, no. 6, 2017, pp. 606–634.

[3] R. Gao, C. Shah, "Toward creating a fairer ranking in search engine results," *Information Processing & Management*, vol. 57, no. 1, 2020, pp. 102138.

[4] M. Fatehkia, R. Kashyap, I. Weber, "Using Facebook ad data to track the global digital gender gap," *World Development*, vol. 107, 2018, pp. 189–209.

[5] N. S. Santos, A. Garc ́ıa-Holgado, M. C. S ́anchez-Go ́mez, "Gender gap in the digital society: A qualitative analysis of the international conversation in the wyred project", in: *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM'19, New York, NY, USA, 2019, pp. 518–524.

[6] I. Portugal, P. Alencar, D. Cowan, "The use of machine learning algorithms

in recommender systems: A systematic review," *Expert Systems with Applications*, vol. 97, 2018, pp. 205–227.

[7] M. Mendoza, N. Torres, "Evaluating content novelty in recommender systems," *Journal of Intelligent Information Systems*, vol. 54, no. 2, 2020, pp. 297–316.

[8] J. Bobadilla, A. Gutiérrez, F. Ortega, B. Zhu, "Reliability quality measures for recommender systems," *Information Sciences*, vol. 442-443, 2018, pp. 145–157.

[9] M. Kunaver, T. Požrl, "Diversity in recommender systems – A survey," *Knowledge-Based Systems*, vol. 123, 2017, pp. 154–162.

[10] M. de Gemmis, P. Lops, G. Semeraro, C. Musto, "An investigation on the serendipity problem in recommender systems," *Information Processing & Management*, vol. 51, no. 5, 2015, pp. 695–717.

[11] D. Kotkov, S. Wang, J. Veijalainen, "A survey of serendipity in recommender systems," *Knowledge-Based Systems*, vol. 111, 2016, pp. 180–192.

[12] K. Holstein, J. Wortman Vaughan, H. Daumé, M. Dudik, H. Wallach, "Improving fairness in machine learning systems: What do industry practitioners need?," in: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, New York, NY, USA, 2019.

[13] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, F. Diaz, "Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems," in: P*roceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 2243–2251.

[14] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, 2004, pp. 5–53.

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

[16] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, "A survey on bias and fairness in machine learning", 2019. arXiv:1908.09635.

[17] R. Burke, N. Sonboli, A. Ordonez-Gauger, "Balanced neighborhoods for multi-sided fairness in recommendation," in: *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, Vol. 81 of Proceedings of Machine Learning Research, PMLR, New York, NY, USA, 2018, pp. 202–214.

[18] J. Leonhardt, A. Anand, M. Khosla, "User fairness in recommender systems," in: *Companion Proceedings of the Web Conference 2018*, WWW '18, Republic and Canton of Geneva, CHE, 2018, pp. 101–102.

[19] M. D. Ekstrand, M. Tian, M. R. I. Kazi, H. Mehrpouyan, D. Kluver, Exploring author gender in book rating and recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 242–250.

[20] V. Tsintzou, E. Pitoura, P. Tsaparas, "Bias disparity in recommendation systems," arXiv:1811.01461.

[21] S. Yao, B. Huang, Beyond parity: Fairness objectives for collaborative filtering, *CoRR* abs/1705.08804.

[22] M. Mansoury, B. Mobasher, R. Burke, M. Pechenizkiy, "Bias Disparity in Collaborative Recommendation: Algorithmic Evaluation and Comparison," *ArXiv e-prints*.

[23] A. Chouldechova, A. Roth, "The frontiers of fairness in machine learning," *CoRR* abs/1810.08810.

[24] R. Mu, "A Survey of Recommender Systems Based on Deep Learning," *IEEE Access*, vol. 6, 2018, pp. 69009–69022.

[25] Z. Batmaz, A. Yurekli, A. Bilge, C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artificial Intelligence Review*, vol. 52, no. 1, 2019, pp. 1–37.

[26] J. Bobadilla, F. Ortega, A. Gutiérrez, S. Alonso, "Classification-based deep neural network architecture for collaborative filtering recommender systems," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 1, 2020, pp. 68–77.

[27] J. Bobadilla, S. Alonso, A. Hernando, "Deep Learning Architecture for Collaborative Filtering Recommender Systems," *Applied Sciences*, vol. 10, no. 7, 2020.

[28] J. Choo, S. Liu, "Visual Analytics for Explainable Deep Learning," *IEEE Computer Graphics and Applications*, vol. 38, no. 4, 2018, pp. 84–92.

[29] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, L. Sun, "Dual-regularized matrix factorization with deep neural networks for recommender systems," *Knowledge-Based Systems*, vol. 145, 2018, pp. 46–58.

[30] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, "Neural collaborative filtering," in: *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, Republic and Canton of Geneva, CHE, 2017, pp. 173–182.

[31] F. M. Harper, J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive and Intelligent Systems*, vol. 5, no. 4, 2015, pp. 1–19.

Jesús Bobadilla

Jesús Bobadilla received the B.S. and the Ph.D. degrees in computer science from the Universidad Politécnica de Madrid and the Universidad Carlos III. Currently, he is a lecturer with the Department of Applied Intelligent Systems, Universidad Politécnica de Madrid. He is a habitual author of programming languages books working with McGraw-''Hill, Ra-Ma and Alfa Omega publishers. His research interests include information retrieval, recommender systems and speech processing. He oversees the FilmAffinity.com research team working on the collaborative filtering kernel of the web site. He has been a researcher into the International Computer Science Institute at Berkeley University and into the Sheffield University. Head of the research group.

Raúl Lara-Cabrera

Raúl Lara-Cabrera received the M.Sc. and Ph.D. degrees in computer science from the University of Málaga, Spain, in 2013 and 2015, respectively. He is currently an Assistant professor with the Department of Sistemas Informáticos, Universidad Politécnica de Madrid, Spain. His main research interests include computational intelligence, machine learning, video games, and complex systems.

Ángel González-Prieto

Ángel González-Prieto received his Double B.S. in Computer Sciences and Mathematics from Universidad Autónoma de Madrid in 2014, his M.Sc. in Mathematics from the same university in 2015 and his Ph.D. in Mathematics from Universidad Complutense de Madrid in 2018. He has been postdoc at Instituto de Ciencias Matemáticas and, currently, he is Teaching Assistant at Universidad Politécnica de Madrid. His research interests include machine learning, deep learning, and algebraic geometry.

Fernando Ortega

Fernando Ortega was born in Madrid, Spain, in 1988. He received the B.S. degree in software engineering, the M.S. degree in artificial intelligence, and the Ph.D. degree in computer sciences from the Universidad Politécnica de Madrid, in 2010, 2011, and 2015, respectively, From 2008 to 2015, he was a Research Assistant with Intelligent Systems for Social learning and Virtual Environments Research Group. From 2015 to 2017, he was with BigTrueData Leading Machine Learning Projects. From 2017 to 2018, he was an Assistant Professor with the U-tad, Centro Universitario de Tecnología y Arte Digital. Since 2018, he has been an Assistant Professor with the Universidad Politécnica de Madrid. He is author of 30 research papers in most prestigious international journals. He leads several national projects to include machine learning algorithms into the society. His research interests include machine learning, data analysis, and artificial intelligence.