Comprehensive Evaluation of Matrix Factorization Models for Collaborative Filtering Recommender Systems

Jesús Bobadilla, Jorge Dueñas-Lerín, Fernando Ortega, Abraham Gutierrez *

Dpt. Sistemas Informáticos and KNODIS Research Group, Universidad Politécnica de Madrid (Spain)

* Corresponding author: jesus.bobadilla@upm.es (J. Bobadilla), jorgedl@alumnos.upm.es (J. Dueñas-Lerìn), fernando.ortega@upm.es (F. Ortega), abraham.gutierrez@upm.es (A. Gutierrez)..

Received 2 July 2022 | Accepted 4 March 2023 | Early Access 28 April 2023

Abstract

Matrix factorization models are the core of current commercial collaborative filtering Recommender Systems. This paper tested six representative matrix factorization models, using four collaborative filtering datasets. Experiments have tested a variety of accuracy and beyond accuracy quality measures, including prediction, recommendation of ordered and unordered lists, novelty, and diversity. Results show each convenient matrix factorization model attending to their simplicity, the required prediction quality, the necessary recommendation quality, the desired recommendation novelty and diversity, the need to explain recommendations, the adequacy of assigning semantic interpretations to hidden factors, the advisability of recommending to groups of users, and the need to obtain reliability values. To ensure the reproducibility of the experiments, an open framework has been used, and the implementation code is provided.

Keywords

Collaborative Filtering, Matrix Factorization, Recommender Systems.

DOI: 10.9781/ijimai.2023.04.008

I. Introduction

Recommender System (RS) [1] is the field of artificial intelligence specialized in user personalization. Mainly, RSs provide accurate item recommendations to users: movies, trips, books, music, etc. Recommendations are made following some filtering approach. The most accurate filtering approach is the Collaborative Filtering (CF) [2], where recommending to an active user involves a first stage to make predictions about all his or her not consumed or voted items. Then, the top predicted items are recommended to the active user. The CF approach assumes the existence of a dataset that contains explicitly voted items or implicitly consumed items from a large number of users. Remarkable commercial RSs are Amazon, Spotify, Netflix, or TripAdvisor.

Regardless of the machine learning model used to implement CF, the key concept is to extract user and item patterns and then to recommend to the active user those items that he or she has not voted or consumed, and that similar users have highly valued. It fits with the K Nearest Neighbors (KNN) memory-based algorithm [3], and it is the reason why the initial RS research was based on KNN. There are also some other filtering approaches such as demographic, social, content-based, context-aware, and their ensembles. Demographic filtering [4] makes use of user information such as gender, age, or zip code, and item information such as movie genre, country to travel, etc. Social filtering [5], [6] has a growing importance in current RS, due to the social networks boom. The existence of trust relations and graphs [7] can improve the quality of the CF recommendations. In this decentralized and dynamic environment, trust between users provides additional information to the centralized set of ratings. Trust relationships can be local, collective, or global [8]; local information is based on shared users' opinions, collective information uses friends' opinions, whereas global information relates to users' reputation [9]. Content-based filtering [10] recommends items with the same type (content) to consumed items (e.g. to recommend Java books to a programmer that bought some other Java book). Context-aware filtering [11] uses GPS information, biometric sensor data, etc. Finally, ensemble architectures [12] get high accuracy by merging several types of filtering.

Memory-based algorithms have two main drawbacks: their accuracy is not high, and each recommendation process requires to recompute the whole dataset. Model-based approaches solve both problems: their accuracy is higher than that of memory-based methods, and they first create a model from the dataset. From the created model we can make many different recommendations, and it can be efficiently updated when the dataset changes. Matrix Factorization (MF) [13] is the most popular approach to implement current RSs: it provides accurate predictions, it is conceptually simple, it has a straightforward

Please cite this article as: J. Bobadilla, J. Dueñas-Lerín, F. Ortega, A. Gutierrez, "Comprehensive Evaluation of Matrix Factorization Models for Collaborative Filtering Recommender Systems", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 8, no. 6, pp. 15-23, 2024, http://dx.doi.org/10.9781/ijimai.2023.04.008

implementation, the model learns fast, and also updates efficiently. The MF model makes a compression of information, coding very sparse and large vectors of discrete values (ratings) to low dimensional embeddings of real numbers, called hidden factors. The hidden factors, both from the user vector and from the item vector, are combined by means of a dot product to return predictions. This is an iterative process in which the distance between training predictions and their target ratings is minimized.

The Probabilistic Matrix Factorization (PMF) model based on MF [13] scales linearly with the size of the data set. It also returns accurate results when applied to sparse, large, and imbalanced CF datasets. PMF has also been extended to include an adaptive prior on the model parameters, and it can generalize adequately, providing accurate recommendation to cold-start users. CF RSs are usually biased. A typical CF bias source comes from the fact that some users tend to highly rate items (mainly 4 and 5 stars), whereas some other users tend to be more restrictive in their ratings (mainly 3 and 4 stars). This fact leads to the extension of the MF model to handle biased data. An user-based rating centrality and an item-based rating centrality [14] have been used to improve the accuracy of the regular PMF. These centrality measures are obtained by processing the degree of deviation of each rating in the overall rating distribution of the user and the item. non-Negative Matrix Factorization (NMF) [15] can extract significant features from sparse and non-negative CF datasets (please note that CF ratings are usually a non-negative number of stars, listened songs, watched movies, etc.). When nonnegativity is imposed, prediction errors are reduced and the semantic interpretability of hidden factors is easier. The Bernoulli Matrix Factorization (BeMF) [16] has been designed to provide both prediction and reliability values; this model uses the Bernoulli distribution to implement a set of binary classification approaches. The results of the binary classification are combined by means of an aggregation process. The Bayesian non-Negative Matrix Factorization (BNMF) [17] was designed to provide useful information about user groups, in addition to the PMF prediction results. The authors factorize the rating matrix into two nonnegative matrices whose components lie within the range [0, 1]. The resulting hidden factors provide an understandable probabilistic meaning. Finally, The User Ratings Profile Model (URP) is a generative latent variable model [18]; it produces complete rating user profiles. In the URP model, first attitudes for each item are generated, then a user attitude for the item is selected from the set of existing attitudes. URP borrows several concepts from LDA [19] and the multinomial aspect model [20].

The set of MF models mentioned above: PMF, Biased Matrix Factorization (BiasedMF), NMF, BeMF, BNMF, and URP, can be considered representative in the CF area. These models will be used in this paper to compare their behavior when applied to representative datasets. Specifically, the following quality measures will be tested: Mean Absolute Error (MAE), novelty, diversity, precision, recall, and Normalized Discounted Cumulative Gain (NDCG). Prediction accuracy will be tested using MAE [21], whereas NDCG, Precision and Recall [22] will be used to test recommendation accuracy. Modern CF models should be tested not only regarding accuracy, but also beyond accuracy properties [23]: novelty [24], [25] and diversity [26]. Novelty can be defined as the quality of a system to avoid redundancy; diversity is a quality that helps to cope with ambiguity or under-specification. The models have been tested using four CF datasets: MovieLens (100K and 1M versions) [27], Filmtrust [28] and MyAnimeList [29]. These are representative open datasets and are popular in RS research.

Overall, this paper provides a complete evaluation of MF methods, where the PMF, BiasedMF, NMF, BeMF, BNMF, and URP models have been tested using representative CF quality measures, both for prediction and recommendation, and also beyond accuracy ones. As far as we know this is the experimental most complete work evaluating current MF models in the CF area.

The rest of the paper is structured as follows: Section II introduces the tested models, the experiment design, the selected quality measures, and the chosen datasets. Section III shows the obtained results and provides their explanations in Section IV. Section V highlights the main conclusions of the paper and the suggested future works. Finally, a references section lists current research in the area.

II. Methods and Experiments

This section abstracts the fundamentals of each baseline model (PMF, BiasedMF, NMF, BeMF, BNMF, URP), introduces the tested quality measures (MAE, precision, recall, NDCG, novelty, diversity), and shows the main parameters of the tested datasets (`Movielens`, `FilmTrust`, `MyAnimeList`). Experiments are performed by combining the previous entities.

The vanilla MF [13], [30] is used to generate rating predictions from a matrix of ratings R. This matrix contains the set of casted ratings (explicit or implicit) from a set of users U to a set of items I. Since regular users only vote or consume a very limited subset of the available items, matrix R is very sparse. The MF key concept is to compress the very sparse item and user vectors of ratings to small size and dense item and user vectors of real numbers; these small size dense vectors can be considered as embeddings, and they usually are called 'hidden factors', since each embedding factor codes some complex non-lineal ('hidden') relation of user or item features. The parameter K is usually chosen to set the embedding (hidden factors) size. MF makes use of two matrices: $P(|U|*K)$ to contain the K hidden factors of each user, and $Q(|I|*K)$ to contain the K hidden factors of each item. To predict how much a user u likes an item i, we compare each hidden factor of u with each corresponding hidden factor of i. Then, the dot product $u \cdot i$ can be used as suitable CF prediction measure. MF predicts ratings by minimizing errors between the original R matrix and the predicted $\hat{R}$ matrix:

$$R \approx P \times Q^T = \hat{R} \tag{1}$$

$$\hat{r}_{ui} = p_u \cdot q_i^T = \sum_{k=1}^{K} p_{uk} \cdot q_{ki} \tag{2}$$

Using gradient descent, we minimize learning errors (differences between real ratings r and predicted ratings $\hat{r}$).

$$e_{ui}^2 = (r_{ui} - \hat{r}_{ui})^2 = \left(r_{ui} - \sum_{k=1}^{K} p_{uk} \cdot q_{ki} \right)^2 \tag{3}$$

To minimize the error, we differentiate equation (3) with respect to p_{uk} and q_{ki}:

$$\frac{\partial}{\partial p_{uk}} e_{ui}^2 = -2(r_{ui} - \hat{r}_{ui}) q_{ki} = -2 e_{ui} q_{ki} \tag{4}$$

$$\frac{\partial}{\partial q_{ki}} e_{ui}^2 = -2(r_{ui} - \hat{r}_{ui}) p_{uk} = -2 e_{ui} p_{uk} \tag{5}$$

Introducing the learning rate α, we can iteratively update the required hidden factors p_{uk} and q_{ki}:

$$p'_{uk} = p_{uk} + \alpha \frac{\partial}{\partial p_{uk}} e_{ui}^2 = p_{uk} + 2\alpha e_{ui} q_{ki} \tag{6}$$

$$q'_{ki} = q_{ki} + \alpha \frac{\partial}{\partial q_{ki}} e_{ui}^2 = q_{ki} + 2\alpha e_{ui} p_{uk} \tag{7}$$

CF datasets have biases, since different users vote or consume items in different ways. In particular, there are users who are more demanding than others when rating products or services. Analogously, there are items more valued than others on average. Biased MF [14] is designed to consider data biases; The following equations extend the previous ones, introducing the bias concept and making the necessary regularization to maintain hidden factor values in their suitable range:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u \cdot q_i^T \tag{8}$$

where μ, b_u, b_i are the average bias, the user bias and the item bias.

We minimize the regularized squared error:

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda\left(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2\right) \tag{9}$$

where λ is the regularization term.

Obtaining the following updating rules:

$$b'_u = b_u + \alpha(e_{ui} - \lambda b_u) \tag{10}$$

$$b'_i = b_i + \alpha(e_{ui} - \lambda b_i) \tag{11}$$

$$p'_u = p_u + \alpha(e_{ui} \cdot q_i - \lambda p_u) \tag{12}$$

$$q'_i = q_i + \alpha(e_{ui} \cdot p_u - \lambda q_i) \tag{13}$$

NMF [15] can be considered as a regular MF subject to the following constraints:

$$R \geqslant 0, P \geqslant 0, Q \geqslant 0 \tag{14}$$

In the NMF case, predictions are made by linearly combining positive coefficients (hidden factors). NMF hidden factors are easier to semantically interpret than regular MF ones: sometimes it is not straightforward to assign semantic meanings to negative coefficient values. In the CF context, another benefit of using NMF decomposition is the emergence of a natural clustering of users and items. Intuitively, users and items can be clustered according to the dominant factor (i.e. the factor having the highest value). In the same way, the original features (gender, age, item type, item year, etc.) can be grouped according to the factor (from the k hidden factors) on which they have the greatest influence. This is possible due to the condition of positivity of the coefficients.

BeMF [16] is an aggregation-based architecture that combines a set of Bernoulli factorization results to provide pairs <prediction, reliability>. BeMF uses as many Bernoulli factorization processes as possible scores in the dataset. Reliability values can be used to detect shilling attacks, to explain the recommendations, and to improve prediction and recommendation accuracy [31]. BeMF is a classification model based on the Bernoulli distribution. It adequately adapts to the expected binary results of each of the possible scores in the dataset. Using BeMF, the prediction for user u to item i is a vector of probabilities $(p_{ui}^1, \ldots, p_{ui}^D)$, where p_{ui}^s is the probability that i is assigned the s-th score from user u. The BeMF model can be abstracted as follows:

Let $S = \{s_1, \ldots, s_D\}$ be the set of D possible scores in the dataset (e.g. 1 to 5 stars: $D = 5$). From R we generate D distinct matrices $(R^{s1}, \ldots, R^{sD})$; each $R^s = (R_{ui}^s)$ matrix is a sparse matrix such that $R_{ui}^s = 1$. BeMF will attempt to fit the matrices $R^{s1}, \ldots, R^{sD}$ by performing D parallel MFs

The BeMF assumes that, given the user P matrix and the item Q matrix containing $k > 0$ hidden factors, the rate R_{ui} is a Bernoulli distribution with the success probability $\psi(P_u \cdot Q_i)$. The mass function of this random variable is:

$$p(R_{ui} \mid P_u, Q_i) = \begin{cases} \psi(P_u Q_i) & \text{if } R_{ui} = 1 \\ 1 - \psi(P_u Q_i) & \text{if } R_{ui} = 0 \end{cases} \tag{15}$$

The associated likelihood is:

$$\ell(R \mid UV) = \sum_{R_{ui}=1} \log\left(\psi(P_u Q_i)\right) + \sum_{R_{ui}=0} \log\left(1 - \psi(P_u Q_i)\right) \tag{16}$$

The BeMF updating equations are:

$$P'_u = P_u + \gamma\left(\sum_{\{i|R_{ui}=1\}} (1 - \text{logit}\,(P_u Q_i))Q_i + \sum_{\{i|R_{ui}=1\}} \text{logit}\,(P_u Q_i)Q_i - \eta P_u\right) \tag{17}$$

$$Q'_i = Q_i + \gamma\left(\sum_{\{i|R_{ui}=1\}} (1 - \text{logit}\,(P_u Q_i))P_u + \sum_{\{i|R_{ui}=1\}} \text{logit}\,(P_u Q_i)P_u - \eta Q_i\right) \tag{18}$$

And the aggregation to obtain the final output Φ:

$$\Phi(u,i) = \frac{1}{\sum_{\alpha=1}^{s} \psi\left(P_u^{S_\alpha} Q_i^{S_\alpha}\right)}\left(\psi\left(P_u^{S_1} Q_i^{S_1}\right), \ldots, \psi\left(P_u^{S_D} Q_i^{S_D}\right)\right) \tag{19}$$

where $\Phi(u,i) = (p_{ui}^1, \ldots, p_{ui}^D)$, $0 \leqslant p_{ui}^\alpha \leqslant 1$, $\sum_\alpha p_{ui}^\alpha = 1$. Let $\alpha_0 = \text{argmax}_\alpha\, p_u^\alpha$; the prediction is: $\hat{R}_{ui} = s_{\alpha_0}$, and the reliability is $p_{ui}^{\alpha_0}$.

BNMF [17] provides a Bayesian-based NMF model that not only allows accurate prediction of user ratings, but also to find groups of users with the same tastes, as well as to explain recommendations. The BNMF model approximates the real posterior distribution $\rho(\emptyset_u, k_{ik}, z_{ui} \mid \rho_{ui})$ by the distribution:

$$q(\emptyset_u, k_{ik}, z_{ui}) = \Pi_{u=1}^{N} q_\emptyset(\emptyset_u) \Pi_{i=1}^{M} \Pi_{k=1}^{K} q_{k_{ik}}(k_{ik}) \Pi_{r_{ui} \neq *} q_{z_{ui}}(z_{ui}) \tag{20}$$

where:

- $z_{ui} \sim \text{Cat}\,(\emptyset_u)$ is a random variable from a categorical distribution.
- $\rho_{ui} \sim \text{Bin}\,\left(R, k_{i,z_{ui}}\right)$ is a random variable from a Binomial distribution (which takes values from 0 to $D-1$)
- $p_{ui} = \sum_{k=1\ldots K} a_{uk} \cdot b_{ik}$ (a and b are hidden matrices).
- $q_{ui} = \begin{cases} 1 & \text{if } 0 \leqslant p_{ui} < 0.2 \\ 2 & \text{if } 0.2 \leqslant p_{ui} < 0.4 \\ \text{etc.} \end{cases}$
- $q_{\emptyset_u}(\emptyset_u) \sim \text{Dir}\,(\gamma_{u1}, \ldots, \gamma_{uk})$ follows a Dirichlet distribution.
- $q_{k_{ik}}(k_{ik}) \sim \text{Beta}\,(\epsilon_{ik}^+, \epsilon_{ik}^-,)$ follows a Beta distribution.
- $q_{z_{ui}}(z_{ui}) \sim \text{Cat}\,(\lambda_{ui1}, \ldots, \lambda_{uik})$ follows a categorical distribution
- λ_{uik} are parameters to be learned: $\lambda_{ui1} + \cdots + \lambda_{uik} = 1$

BNMF iteratively approximates parameters $\{\gamma_{uk}, \epsilon_{ik}^+, \epsilon_{ik}^-, \lambda_{uik}\}$:

$$\gamma_{uk} = \alpha + \sum_{\{i|r_{ui} \neq *\}} \lambda_{uik} \tag{21}$$

$$\epsilon_{ik}^+ = \beta + \sum_{\{i|r_{ui} \neq *\}} \lambda_{uik} \cdot r_{ui}^+ \tag{22}$$

$$\epsilon_{ik}^- = \beta + \sum_{\{i|r_{ui} \neq *\}} \lambda_{uik} \cdot r_{ui}^- \tag{23}$$

$$\lambda'_{uik} = exp(\psi(\gamma_{uk}) + r_{ui}^+ \cdot \psi(\epsilon_{ik}^+) + r_{ui}^- \cdot \psi(\epsilon_{ik}^-) - (D-1) \cdot \psi(\epsilon_{ik}^+ + \epsilon_{ik}^-)) \tag{24}$$

$$r_{ui}^+ = \rho_{ui} = (D-1) \cdot r_{ui}^* \tag{25}$$

$$r_{ui}^- = (D-1) - \rho_{ui} = (D-1) \cdot (1 - r_{ui}^*) \tag{26}$$

$$r_{ui}^* = \frac{\rho_{ui}}{(D-1)} \tag{27}$$

where ψ is the digamma function as the logarithmic derivative of the gamma function.

URP is a generative latent variable model [18]. The model assigns to each user a mixture of user attitudes. Mixing is performed by a

Dirichlet random variable:

$$P(\theta, z \mid \alpha, \beta, r^u) \approx q(\theta, z \mid \gamma^u, \emptyset_u) = q(\theta \mid \gamma^u) \Pi_{y=1}^{M} q(Z_y = z_y \mid \emptyset_y^u) \quad (28)$$

$$\emptyset_{zy}^u \approx \Pi_{v=1}^{V} \beta_{vyz}^{\delta(r_y^u, v)} \exp\left(\psi(\gamma_z^u) - \psi\left(\sum_{j=1}^{k} \gamma_j^u\right)\right) \quad (29)$$

$$\gamma_z^u = \alpha_z + \sum_{y=1}^{M} \emptyset_{zy}^u \quad (30)$$

$$\beta_{vyz} \approx \sum_{u=1}^{N} \emptyset_{zy}^u \delta(r_y^u, v) \quad (31)$$

$$\psi(\alpha_z) = \psi\left(\sum_{z=1}^{K} \alpha_z\right) + \frac{1}{N} \cdot \left(\sum_{u=1}^{N} \psi(\gamma_z^u) - \psi\left(\sum_{u=1}^{N} \gamma_z^u\right)\right) \quad (32)$$

$$\alpha_z = \psi^{-1}(\psi(\alpha_z)) \quad (33)$$

In this paper, baseline models will be tested using a) prediction measure, b) recommendation measures, and c) beyond accuracy measures. The chosen prediction measure is the MAE, where the absolute differences of the errors are averaged. Absolute precision and relative recall measures are tested to compare the quality of an unordered list of N recommendations. The ordered lists of recommendations will be compared using the NDCG quality measure. From the beyond accuracy metrics, we have selected novelty and diversity. Novelty returns the distance from the items the user 'knows' (has voted or consumed) to his recommended set of items. Diversity tells us about the distance between the set of recommended items. Recommendations with high novelty values are valuable, since they show to the user unknown types of items. Diverse recommendations are valuable because they provide different types of items (and each type of item can be novel, or not, to the user).

The GroupLens research group [27] made available several CF datasets, collected over different intervals of time. MovieLens 100K and MovieLens 1M describe 5-star rating and free-text tagging activity. These data were created from 1996 to 2018. In the Movielens 100K dataset, users were selected at random from those who had rated at least 20 movies, whereas the MovieLens 1M dataset has not this constraint. Only movies with at least one rating or tag are included in the dataset. No demographic information is included. Each user is represented by an 'id', and no other information is provided. The dataset files are written as comma-separated values files with a single header row. Columns that contain commas (,) *areescapedusing* double-quotes ("). These files are encoded as UTF-8. All ratings are contained in the file named 'ratings.csv'. Each line of this file after the header row represents one rating of one movie by one user, and has the following format: 'userId, movieId, rating, timestamp'. The lines within this file are ordered first by 'userId', then, within user, by 'movieId'. Timestamps represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970. FilmTrust is a small dataset crawled from the entire FilmTrust website in June, 2011. As the Movielens datasets, it contains ratings voted from users to items; additionally, it provides social information structured as a graph network. Finally, MyAnimeList contains information about anime and 'otaku' consumers (anime, manga, video games and computers). Each user is able to add 'animes' to their completed list and give them a rating; this data set is a compilation of those ratings. The MyAnimeList CF information is contained in the file 'Anime.csv', where their main columns are 'anime_id': myanimelist.net's unique 'id' identifying an anime; 'name': full name of anime; 'genre': comma separated list of genres for this anime; 'type': movie, TV, OVA, etc; 'episodes': how many episodes in this show; 'rating': average rating out of 10 for this anime. These datasets are available in the Kaggle and GitHub repositories, as well as in the KNODIS research group CF4J [32] repository https://github.com/ferortega/cf4j.

Table I contains the values of the main parameters of the selected CF data sets: Movielens 100K, Movielens 1M, FilmTrust and MyAnimeList. We have run the explained MF models on each of the four Table I datasets, testing the chosen quality measures. Please note that the MyAnimeList dataset ratings range from 1 to 10 , whereas MovieLens datasets range from 1 to 5 and FilmTrust ranges from 0 to 5 with 0.5 increments. It is also remarkable the sparsity difference between FilmTrust and the rest of the tested datasets.

TABLE I. Main Parameter Values of the Tested Datasets

Dataset	#users	#items	#ratings	Scores	Sparsity
MovieLens100k	943	1682	99,831	1 to 5	93.71
MovieLens1M	6,040	3,706	911,031	1 to 5	95.94
MyAnimeList	19,179	2,692	548,967	1 to 10	98.94
FilmTrust	1,508	2,071	35,497	0 to 5	87.98

Experiments have been performed using random search and applying four-fold cross-validation. To ensure reproducibility, we used a seed in the random process. Results shown in the paper are the average of the partial results obtained by setting the number k of latent factors to {4, 8, 12}, and the number of MF iterations to {20, 50, 75, 100}. Additionally, to run the PMF, BiasedMF, and BeMF models, both the learning rate and the regularization parameters have been set to {0.001, 0.01, 0.1, 1.0}. The BNMF model requires two specific parameters: α and β; the chosen values por these parameters are: α = {0.2, 0.4, 0.6, 0.8}, and β = {5, 15, 25}. The tested number of recommendations N ranges from 1 to 10. We have used 4 stars as recommendation threshold θ for datasets whose ratings range from 1 to 5 , while the testing threshold has been 8 when MyAnimeList was chosen. The experiments have been implemented using the open framework [33] and the code has been made available at https://github.com/KNODIS-Research-Group/choice-of-mf-models.

III. Results

The prediction quality obtained by testing each baseline model is shown in table II. The bold numbers correspond to the best results, and, of them, those highlighted gray are the top ones. As can be seen, BiasedMF and BNMF models provide the best CF prediction results. PMF, NMF, BeMF and URP seem to be more sensitive to the type of CF input data.

TABLE II. Prediction Quality Results Using the Mean Absolute Error (MAE). The Lower the Error Value, the Better the Result

	PMF	BiasedMF	NMF	BeMF	BNMF	URP
MovieLens 100K	0.770	**0.754**	0.804	0.805	**0.748**	0.837
MovieLens 1M	0.729	**0.712**	0.744	0.748	**0.693**	0.795
FilmTrust	0.863	**0.652**	0.876	0.712	**0.666**	0.831
MyAnimeList	1.110	**0.926**	1.147	1.034	**0.943**	1.159

Fig. 1 shows the quality of recommendation obtained using the Precision measure. The most remarkable in Fig. 1 is the superiority of the models PMF and BiasedMF. For the remaining models, URP and BeMF provide the worst results, whereas the nonnegative NMF and BNMF return an intermediate quality. It is important to highlight the good performance of the BiasedMF model for both the prediction and the recommendation tasks.

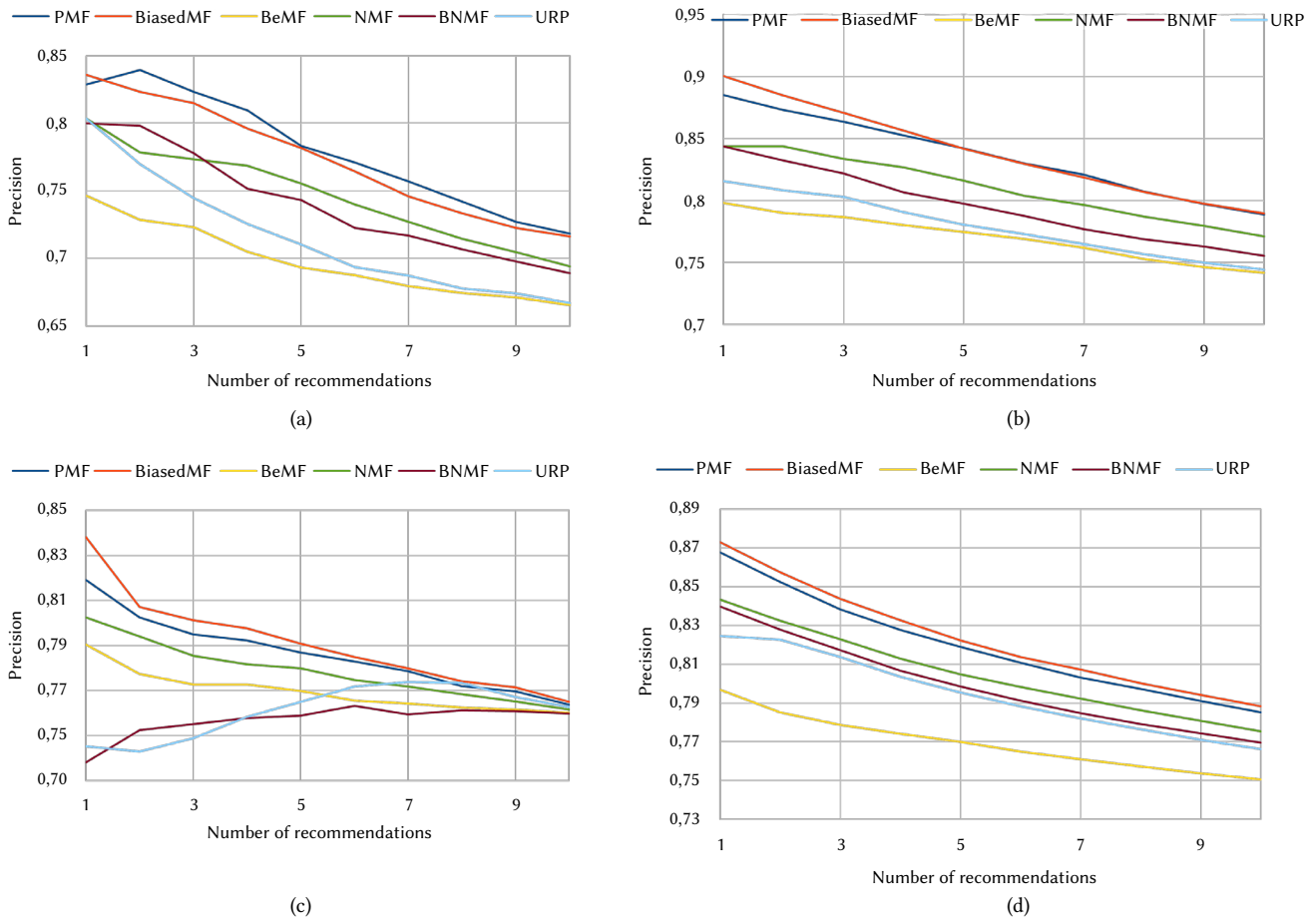

Fig. 1. Precision recommendation quality results; a) `MovieLens100K`, b) `MovieLens 1M`, c) `FilmTrust`, d) `MyAnimeList`. The higher the values, the better the results.

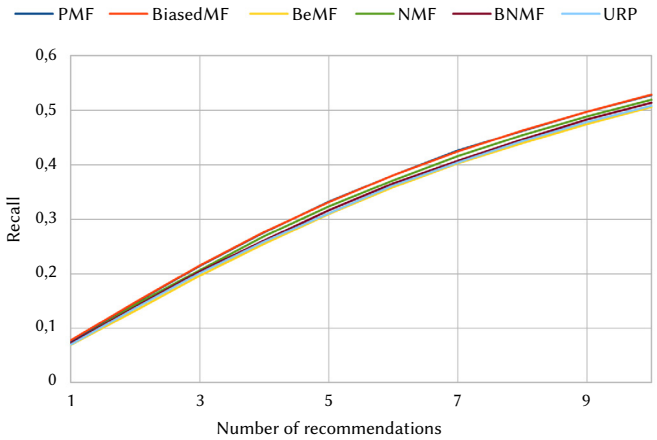

Fig. 2. Recall Recommendation quality results obtained in the `MovieLens` 1M dataset. The results of the other three considered datasets are very similar to this one; to maintain the paper as short as possible, the results of other datasets are not shown.

To test the quality of CF recommendations of unordered recommendations, precision and recall measures are usually processed, and they are provided separately, or joined in the F1 score. We have done these experiments and we have not found appreciable differences in Recall values for the tested models in the selected datasets. In order to maintain the paper as short as possible, Fig. 2 only shows the Recall results obtained by processing the `Movielens 1M` dataset. Results from the rest of datasets are very similar; consequently, the Recall quality measure does not help, in this context, to find out the best MF models in the CF area.

Fig. 2. Recall Recommendation quality results obtained in the MovieLens 1M dataset. The results of the other three considered datasets are very similar to this one; to maintain the paper as short as possible, the results of other datasets are not shown.

In the RSs field, recommendations are usually provided in an ordered list. Users' trust in RSs quickly decays when the first recommendations in the list do not meet their expectations; for that reason, the NDCG quality measure particularly penalizes errors in the first recommendations of the list. Fig. 3 (NDCG results) shows a similar behavior to Fig. 1, where the BiasedMF and PMF models provide the best recommendation quality. So, these two models perform fine both in recommending ordered and unordered lists.

Traditionally, RSs have been evaluated attending to their prediction and recommendation accuracy; nevertheless, there are some other valuable beyond accuracy aims and their corresponding quality measures. The Diversity measure tests the variety of recommendations, penalizing recommendations focused on the same 'area' (Star Wars III, Star Wars I, Star Wars V, Han Solo). Fig. 4 shows the Diversity results obtained by testing the selected models; the most diverse recommendations are usually returned when the BiasedMF model is used, followed by both PMF and NMF. This fact is particularly interesting, since it is not intuitive that the same model (BiasedMF) can, simultaneously, provide accurate and diverse recommendations.

Novelty is an important beyond accuracy objective in RSs. Users appreciate accurate recommendations, but they also want to discover

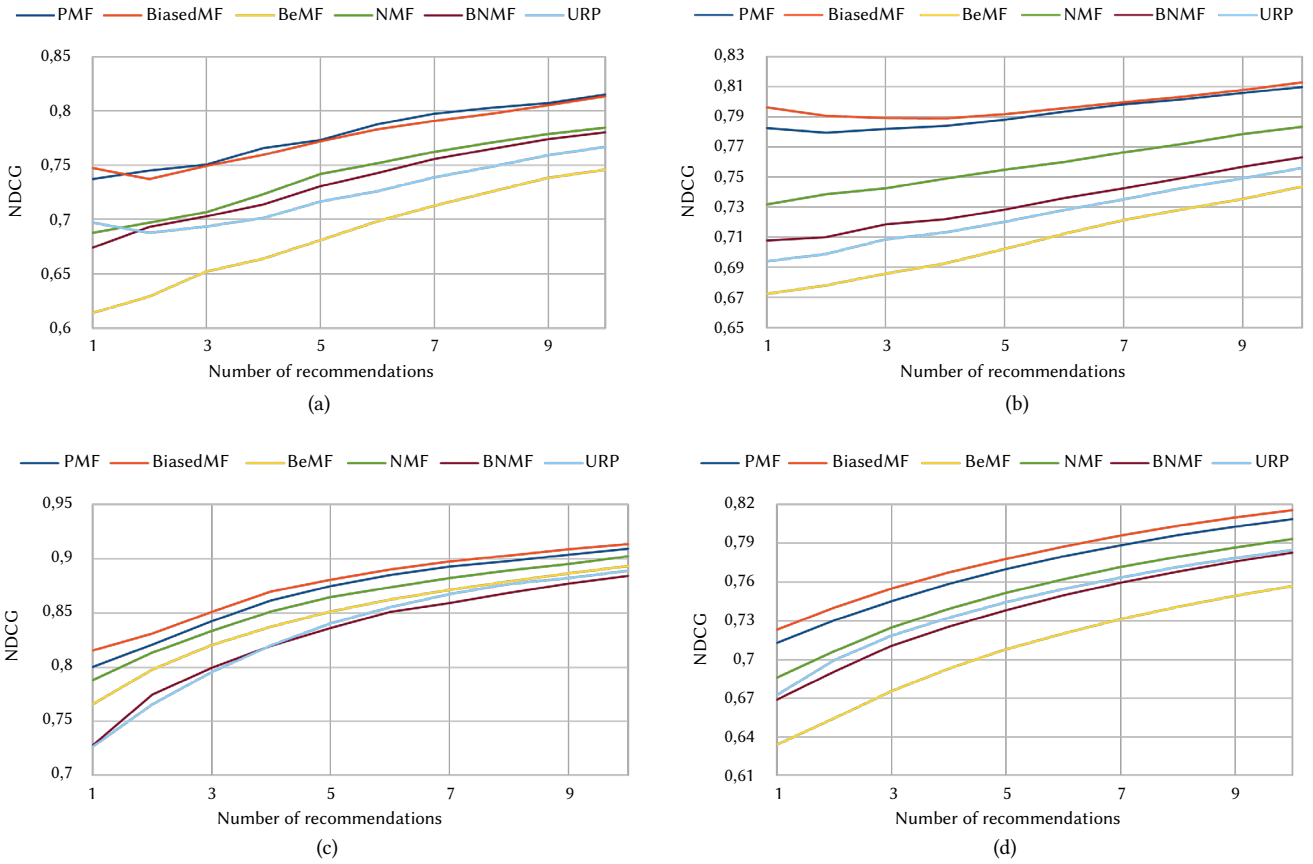

Fig. 3. Normalized Discounted Cumulative Gain recommendation quality results; a) `MovieLens100K`, b) `MovieLens 1M`, c) `FilmTrust`, d) `MyAnimeList`. The higher the values, the better the results.

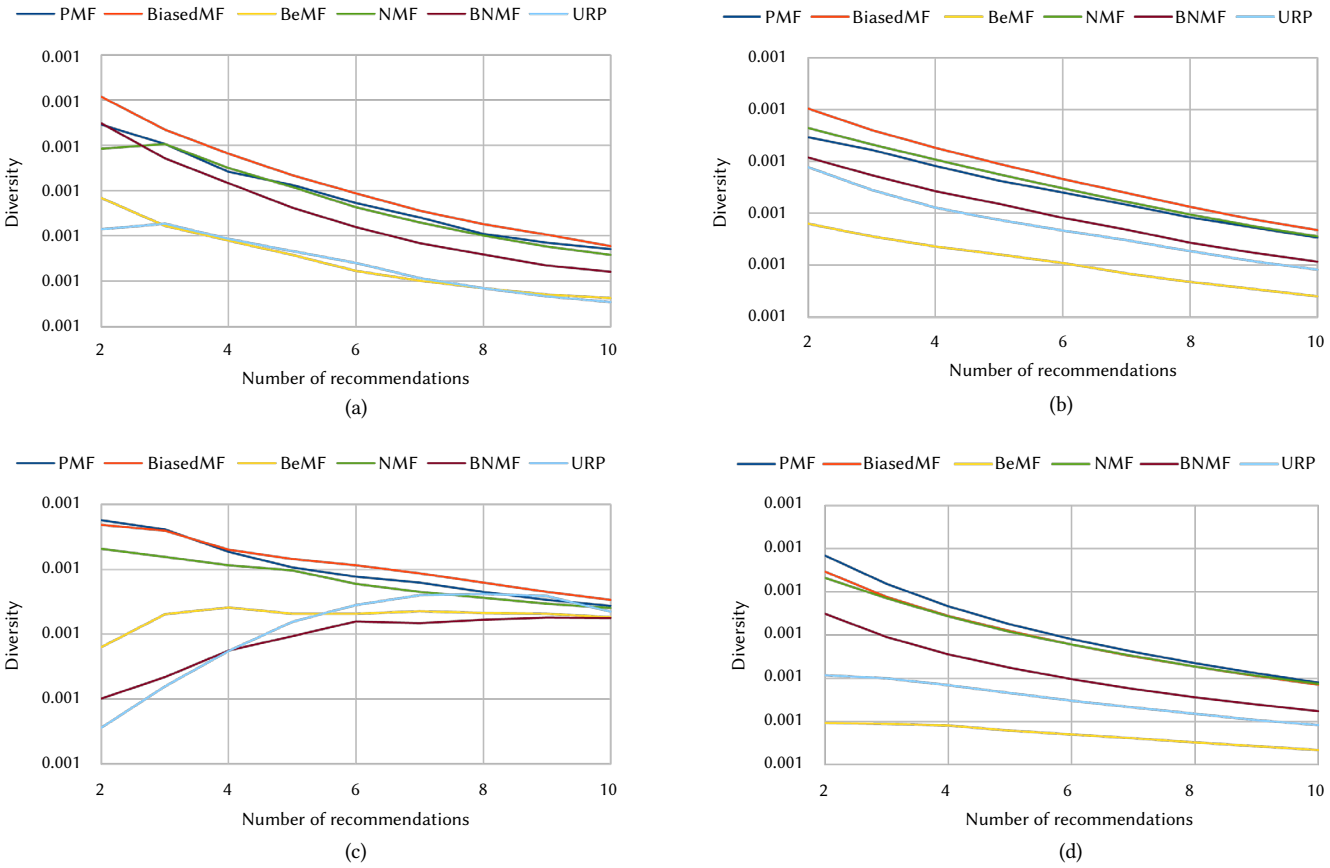

Fig. 4. Diversity beyond accuracy results; a) `MovieLens100K`, b) `MovieLens 1M`, c) `FilmTrust`, d) `MyAnimeList`. The higher the values, the better the results.

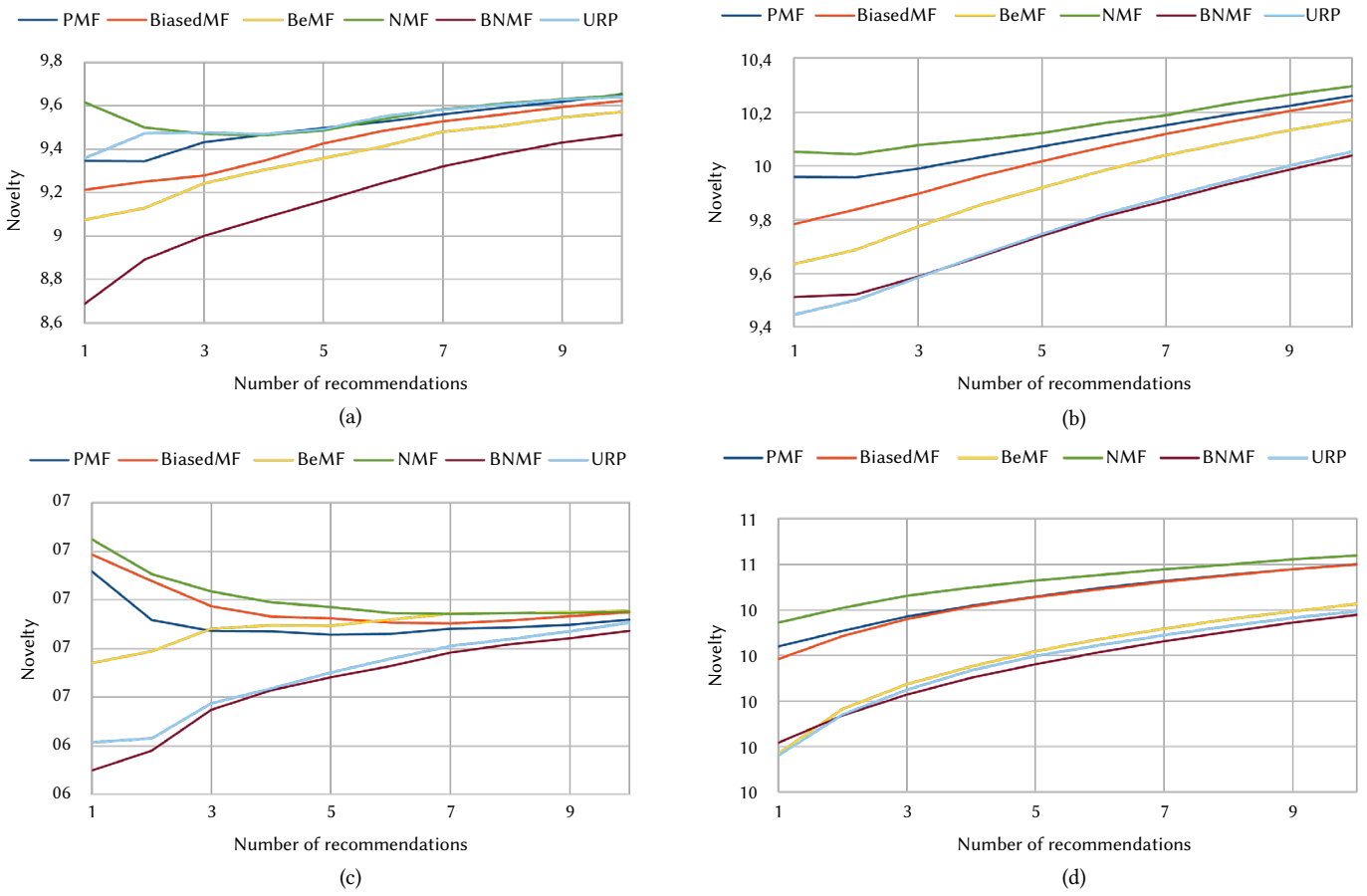

Fig. 5. Novelty beyond accuracy quality results; a) `MovieLens100K`, b) `MovieLens 1M`, c) `FilmTrust`, d) `MyAnimeList`. The higher the values, the better the results.

unexpected (and accurate enough) recommendations. Please note that a set of recommendations can be diverse and not novel, as they can be novel and not diverse. It would be great to receive, simultaneously, accurate, novel, and diverse recommendations, but usually improving some of the objectives leads to worsening others. Fig. 5 shows the results of the novelty quality measure: NMF returns novel recommendations, compared to other models; NMF provides a balance between accuracy and novelty. BiasedMF and PMF also provide novel recommendations compared to BeMF and URP.

IV. Discussion

In this section, we provide a comparative discussion of the most adequate MF models when applied to a set of different CF databases. To judge each MF model, we simultaneously measure a set of conflicting goals: prediction accuracy, recommendation accuracy (unordered and ordered lists) and beyond accuracy aims. We will promote some MF models as 'winners', attending to their high performance (overall quality results) when applied to the tested datasets. We also provide a summary table to better identify those MF models that perform particularly fine on any individual quality objective: novelty, diversity, precision, etc., as well as any combination of those quality measures.

Table III summarizes the results of this section. BiasedMF is the most appropriate model when novelty of recommendations is not a particularly relevant issue. PMF can be used instead BiasedMF when simplicity is required (e.g. educational environments). BeMF should only be used when reliability information is required or when reliability values are used to improve accuracy [31]. NMF and BNMF are adequate when semantic interpretation of hidden factors is needed. NMF is the best choice when we want to be recommended with novel items. BNMF provides good accuracy and it is designed to recommend to group of users.

TABLE III. MF Models Comparative

	PMF	BiasedMF	NMF	BeMF	BNMF	URP
MAE	++	+++	+	+	+++	+
Precision	+++	+++	++	+	++	+
NDCG	+++	+++	+	+	+	+
Diversity	++	+++	++	+	+	+
Novelty	++	++	+++	+	+	+
Total	12	14	9	5	8	5

V. Conclusions

This paper makes a comparative of relevant MF models applied to collaborative filtering recommender systems. Prediction, recommendation, and beyond accuracy quality measures have been tested on four representative datasets. The results show the superiority of the BiasedMF model, followed by the PMF one. BiasedMF arises as the most convenient model when novelty is not a particularly important feature. PMF combines simplicity with accuracy; it can be the best choice for educational or not commercial implementations. NMF and BNMF are adequate when we want to do a semantic interpretation of their non-negative hidden factors. NMF is preferable to BNMF when beyond accuracy (novelty and diversity) results are required, whereas it is better to make use of BNMF when prediction

accuracy is required or when recommending to group of users, or when explaining recommendations is needed. NMF and BiasedMF are the best choices when beyond accuracy aims are selected, whereas PMF or BiasedMF performs particularly well in recommendation task, both for unordered and ordered options. BeMF can only be selected when reliability values are required or when they are used to improve accuracy. Finally, URP does not seem to be an adequate choice in any of the combinations tested. As future work, it is proposed to add new MF models, quality measures, and datasets to the experiments, as well as the possibility of including neural network models such as DeepMF or Neural Collaborative Filtering (NCF).

Acknowledgments

This work has been co-funded by the *Ministerio de Ciencia e Innovación* of Spain and European Regional Development Fund (FEDER) under grants PID2019-106493RB-I00 (DL-CEMG) and the *Comunidad de Madrid* under *Convenio Plurianual* with the *Universidad Politécnica de Madrid* in the actuation line of *Programa de Excelencia para el Profesorado Universitario.*

References

[1] 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, pp. 1–37, 2019.

[2] J. Bobadilla, S. Alonso, A. Hernando, "Deep learning architecture for collaborative filtering recommender systems," Applied Sciences, vol. 10, no. 7, p. 2441, 2020.

[3] B. Zhu, R. Hurtado, J. Bobadilla, F. Ortega, "An efficient recommender system method based on the numerical relevances and the non-numerical structures of the ratings," *IEEE Access*, vol. 6, pp. 49935–49954, 2018.

[4] J. Bobadilla, R. Lara-Cabrera, Á. González-Prieto, F. Ortega, "Deepfair: Deep learning for improving fairness in recommender systems," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 86–94, 2021, doi: 10.9781/ijimai.2020.11.001.

[5] J. Carbó, J. M. Molina, J. Dávila, "Fuzzy referral based cooperation in social networks of agents," *AI Communications*, vol. 18, pp. 1–13, 2005. 1.

[6] D. Medel, C. González-González, S. V. Aciar, "Social relations and methods in recommender systems: A systematic review," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 4, p. 7, 2022, doi: 10.9781/ijimai.2021.12.004.

[7] M. Caro-Martínez, G. Jiménez-Díaz, J. A. Recio- García, "Local model-agnostic explanations for black-box recommender systems using interaction graphs and link prediction techniques," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. InPress, no. InPress, p. 1, 2021, doi: 10.9781/ijimai.2021.12.001.

[8] S. Afef, Z. Brahmi, M. Gammoudi, "Trust-based recommender systems: An overview," in *27th IBIMA Conference*, 05 2016.

[9] I. Pinyol, J. Sabater-Mir, "Computational trust and reputation models for open multi-agent systems: a review," *Artificial Intelligence Review*, vol. 40, pp. 1–25, Jun 2013, doi: 10.1007/s10462-011-9277-z.

[10] Y. Deldjoo, M. Schedl, P. Cremonesi, G. Pasi, "Recommender systems leveraging multimedia content," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–38, 2020.

[11] S. Kulkarni, S. F. Rodd, "Context aware recommendation systems: A review of the state of the art techniques," *Computer Science Review*, vol. 37, p. 100255, 2020.

[12] S. Forouzandeh, K. Berahmand, M. Rostami, "Presentation of a recommender system with ensemble learning and graph embedding: a case on movielens," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7805–7832, 2021.

[13] R. Salakhutdinov, A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, Red Hook, NY, USA, 2007, p. 1257–1264, Curran Associates Inc.

[14] Z. Wu, H. Tian, X. Zhu, S. Wang, "Optimization matrix factorization recommendation algorithm based on rating centrality," in *International Conference on Data Mining and Big Data*, 2018, pp. 114–125, Springer.

[15] C. Févotte, J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[16] F. Ortega, R. Lara-Cabrera, Á. González-Prieto, J. Bobadilla, "Providing reliability in recommender systems through bernoulli matrix factorization," *Information Sciences*, vol. 553, pp. 110–128, 2021.

[17] 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, pp. 188–202, 2016.

[18] B. M. Marlin, "Modeling user rating profiles for collaborative filtering," *Advances in neural information processing systems*, vol. 16, 2003.

[19] D. M. Blei, A. Y. Ng, M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[20] T. Hofmann, "Learning what people (don't) want," in *European Conference on Machine Learning*, 2001, pp. 214– 225, Springer.

[21] A. Gunawardana, G. Shani, "Evaluating recommender systems," in *Recommender systems handbook*, Springer, 2015, pp. 265–308.

[22] C. C. Aggarwal, "Evaluating recommender systems," in *Recommender systems*, Springer, 2016, pp. 225–254.

[23] J. Bobadilla, A. Gutiérrez, S. Alonso, Á. González- Prieto, "Neural collaborative filtering classification model to obtain prediction reliabilities," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 4, pp. 18–26, 2022, doi: 10.9781/ijimai.2021.08.010.

[24] S. Vargas, P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 109–116.

[25] P. Castells, S. Vargas, J. Wang, "Novelty and diversity metrics for recommender systems: choice, discovery and relevance," in *Proceedings of the 33rd European Conference on Information Retrieval (ECIR'11)*, 2011.

[26] S. Vargas, P. Castells, D. Vallet, "Intent-oriented diversity in recommender systems," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 1211– 1212.

[27] F. M. Harper, J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015, doi: https://doi.org/10.1145/2827872.

[28] J. Golbeck, J. A. Hendler, "Filmtrust: movie recommendations using trust in web-based social networks," *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, vol. 1, pp. 282–286, 2006, doi: 10.1109/CCNC.2006.1593032.

[29] J. Miller, G. Southern, "Recommender system for animated video," *Issues in Information Systems*, vol. 15, no. 2, pp. 321–7, 2014.

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

[31] J. Bobadilla, A. Gutiérrez, S. Alonso, Á. González- Prieto, "Neural collaborative filtering classification model to obtain prediction reliabilities," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 4, pp. 18–26, 2022, doi: 10.9781/ijimai.2021.08.010.

[32] F. Ortega, B. Zhu, J. Bobadilla, A. Hernando, "Cf4j: Collaborative filtering for java," *Knowledge- Based Systems*, vol. 152, pp. 94–99, 2018, doi: https://doi.org/10.1016/j.knosys.2018.04.008.

[33] F. Ortega, J. Mayor, D. López-Fernández, R. Lara- Cabrera, "Cf4j 2.0: Adapting collaborative filtering for java to new challenges of collaborative filtering based recommender systems," *Knowledge-Based Systems*, vol. 215, p. 106629, 2021.

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 full professor 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 teamworking 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.

Jorge Dueñas-Lerín

Jorge Dueñas-Lerín received the B.S. in computer science from the Universidad Politécnica de Madrid. He received the M.S. degree in highschool, vocational training and languages teacher from the Universidad Nacional de Educación a Distancia. He is currently a Ph.D. student as part of the KNOledge Discovery and Information Systems - KNODIS research group.

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 theUniversidad Politécnica de Madrid, in 2010, 2011, and 2015, respectively. He is currently Associate Professor in the Universidad Politécnica de Madrid. He is author of more than 50 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. He is the head researcher of the KNOledge Discovery and Information Systems - KNODIS research group.

Abraham Gutiérrez

Abraham Gutiérrez received the B.S. and the Ph.D. degrees in computer science from the Universidad Politécnica de Madrid. Currently, he is currently an associate professor with the Department of Information Systems, Universidad Politécnica de Madrid. He is the author of search papers in most prestigious international journals. He is a habitual author of programming languages books working with McGraw-Hill, Ra-Ma and Alfa Omega publishers. His research interests include P-Systems, machine learning, data analysis and artificial intelligence. He is in charge of this group innovation issues, including the commercial projects.