ProtoMF: Prototype-based Matrix Factorization for Effective and Explainable Recommendations

Alessandro B. Melchiorre
Navid Rekabsaz
alessandro.melchiorre@jku.at
navid.rekabsaz@jku.at
Institute of Computational Perception,
Johannes Kepler University Linz and
Human-centered AI group, Linz
Institute of Technology
Linz, Austria

Christian Ganhör
christian.ganhoer@jku.at
Institute of Computational Perception,
Johannes Kepler University Linz
Linz, Austria

Markus Schedl
markus.schedl@jku.at
Institute of Computational Perception,
Johannes Kepler University Linz and
Human-centered AI group, Linz
Institute of Technology
Linz, Austria

ABSTRACT

Recent studies show the benefits of reformulating common machine learning models through the concept of *prototypes* – representatives of the underlying data, used to calculate the prediction score as a linear combination of similarities of a data point to prototypes. Such prototype-based formulation of a model, in addition to preserving (sometimes enhancing) the performance, enables explainability of the model's decisions, as the prediction can be linearly broken down into the contributions of distinct definable prototypes. Following this direction, we extend the idea of prototypes to the recommender system domain by introducing ProtoMF, a novel collaborative filtering algorithm. ProtoMF learns sets of user/item prototypes that represent the general consumption characteristics of users/items in the underlying dataset. Using these prototypes, ProtoMF then represents users and items as vectors of similarities to the corresponding prototypes. These user/item representations are ultimately leveraged to make recommendations that are both *effective* in terms of accuracy metrics, and *explainable* through the interpretation of prototypes' contributions to the affinity scores. We conduct experiments on three datasets to assess both the effectiveness and the explainability of ProtoMF. Addressing the former, we show that ProtoMF exhibits higher Hit Ratio and NDCG compared to other relevant collaborative filtering approaches. As for the latter, we qualitatively show how ProtoMF can provide explainable recommendations and how its explanation capabilities can expose the existence of statistical biases in the learned representations, which we exemplify for the case of gender bias.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Collaborative filtering*.

KEYWORDS

recommender systems, explainability, prototypes, bias

ACM Reference Format:
Alessandro B. Melchiorre, Navid Rekabsaz, Christian Ganhör, and Markus Schedl. 2022. ProtoMF: Prototype-based Matrix Factorization for Effective and Explainable Recommendations. In *Sixteenth ACM Conference on Recommender Systems (RecSys '22), September 18–23, 2022, Seattle, WA, USA*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3523227.3546756

1 INTRODUCTION

Prototype-based models have introduced a novel paradigm for learning and characterizing latent factors, providing new possibilities, particularly for effective and explainable machine learning [3, 14, 28, 30, 35, 71]. In this context, a prototype is defined as an entity (e. g., in the form of an embedding) that is representative of a set of similar instances and is part of the observed data points, or an artifact that summarizes a subset of instances with similar characteristics [28]. In principle, prototype-based models first identify a set of prototypes from the underlying data and then utilize them to make the prediction for a given data point by linearly combining the relatedness of the data point to the prototypes. This linear combination provides a clear separation of the contribution of each prototype to the final prediction and hence enables understanding of the models' decisions through analyzing these contributions and interpreting the prototypes.

Few recent studies have leveraged the concept of prototypes for recommender systems (RSs) in the context of cold/few-start scenarios [38, 55], or as more effective recommendation algorithms [7]. Within this context, we present ProtoMF, a novel collaborative filtering algorithm based on prototypes. The ProtoMF model builds upon latent factor models [21] and particularly the seminal Matrix Factorization (MF) [31, 53]. The proposed ProtoMF model, besides enhancing the performance of the model's recommendations, unlike previous work, enables explainable recommendations by leveraging *prototypical users and items* that capture the item-consumption characteristics of the system's users and items. For example, a user prototype might personify the overall user preference for Drama and Romance movies, while an item prototype might represent specific musical genres or product categories. The ProtoMF approach utilizes these prototypes to define new users' and items' representations in terms of their similarities to the corresponding prototypes. By leveraging these new representations, ProtoMF finally computes the user-item affinity scores as a linear combination

RecSys '22, September 18–23, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9278-5/22/09.
https://doi.org/10.1145/3523227.3546756

of user/item prototype similarities and the corresponding item/user embeddings.

Considering this design, our prototype-based approach enhances the model's transparency, as the predictions can be deconstructed into a (linear) composition of the contributions stemming from the prototypes, in a similar spirit to previous studies on classification tasks [3, 28, 35]. To further explain the model's decisions, one also requires an interpretation of prototypes, whether the ones of users or items. Related literature in the classification domain approaches this in two ways. The first approach interprets a prototype by observing the model's output given some crafted synthetic inputs that maximally *activate* the prototype [35, 71]. The second method interprets a prototype through a set of maximally close entities to it [3]. In ProtoMF, we adopt the first and second approach to interpret user and item prototypes, respectively. Using these interpretations, we explain the recommendation of ProtoMF through the contributions of the prototypes to the affinity score.

ProtoMF's approach to explainability is aligned with the algorithmic transparency [4, 28, 37] aspect of interpretability discussed by Arrieta et al. [4], namely "*understanding the process followed by the model to produce any given output from its input data*". This is achieved by combining the interpretable capacity of a linear model with the natural *explanation-by-example* provided by the prototypes. Our ProtoMF approach is also aligned with several works that leverage explainability to unveil the existence of societal biases and stereotypes [13, 15, 32, 50, 51], and to help mitigate unfair treatment of individuals and groups [26, 42, 43, 49, 57, 68]. These aspects are particularly critical when abiding by regulations such as the EU Regulatory Framework for AI [19] or the EU Digital Service Act [18], encouraging the development of effective *and* transparent RS models, which are able to explain their predictions and can offer a way to correct possible misconducts [54].

We carry out extensive experiments to assess ProtoMF's effectiveness against relevant baselines. In particular, we evaluate ProtoMF on three real-world datasets (MovieLens, Amazon Video Games, and the LFM2b music dataset), showing that ProtoMF significantly outperforms Matrix Factorization [31], as well as two prototype-based approaches [7, 38] in terms of Hit Ratio and NDCG. In the context of transparency, we showcase ProtoMF's explanation capabilities in two steps. First, by qualitatively demonstrating that the learned prototypical users and items capture general item-consumption behaviors of real users (e. g., preference for movie genres or a specific movie storyline); and second, by exhibiting how the system leverages these learned prototypes to provide an explainable recommendation. Furthermore, utilizing the datasets containing male/female gender information of their users (MovieLens and LFM2b), we expose the existence of gender biases in the learned user prototypes. To this end, we identify prototypes with significant inclinations to either of the genders by analyzing the gender representations of their closest users.

Our contribution is three-fold:

- We propose ProtoMF, a novel collaborative filtering model which leverages user/item prototypes to provide effective and explainable recommendations.
- We perform extensive quantitative and qualitative experiments to assess, respectively, the accuracy and explainability of our model.
- We investigate and expose latent statistical (gender) biases in the learned user prototypes.

Our paper is structured as follows: in Section 2 we review the relevant literature. We introduce our method in Section 3, and the experiment setup in Section 4. We show the evaluation results and the explanation capabilities of ProtoMF, followed by showcasing the existence of gender bias in the model in Section 5.

2 RELATED WORK

We review the relevant literature on prototypes in RSs, prototype-based explanations in machine learning as well as explainability in RSs. Finally, we discuss some studies regarding bias and fairness in RSs.

2.1 Prototypes in Recommender Systems

A common application of prototypes in RSs is approaching cold/few-start problem [2, 38, 55, 58]. In this context, a few *representative* users/items are selected, whose consumption patterns are used to provide recommendations to new users or on new items. As a representative example of this line of work, Liu et al. [38] introduce Representative-based Matrix Factorization (RBMF) which proposes to align the latent factors resulting from matrix factorization with some specific users as the representatives of the system. With this alignment, RBMF also enables some degree of interpretability as recommendations can be explained by user-to-representatives similarity scores and the representatives' ratings. Our ProtoMF generalizes the concept of representatives offered by RBMF by learning prototypical users that incorporate general item consumption patterns.

In the context of prototype-based approaches to RS explainability, and closely related to the study at hand, Barkan et al. [7] recently introduce Anchor-based Collaborative Filtering (ACF). In this work, the authors define a set of *anchor* vectors – generic representatives of tastes and preferences – and use the same set to represent both users and items, based on which recommendations are made. In contrast to ACF, not only the prototypes in our proposed models are separately defined for users and items, but we also allow to trace back the contributions to the prototype vectors, facilitating a direct explanation for recommendations.

Finally, we should also mention clustering-based [45, 56, 61, 63, 65–67] and group discovery [27, 39, 70] approaches in RSs as they share conceptual similarities with prototype-based approaches in terms of benefiting from shared subtleties of users/items. In principle, these approaches exploit clustering of users/items into subgroups and then use the subgroups to provide recommendations using the information of in-group/neighboring entities. Differently from these, prototype-based methods and particularly ProtoMF (1) redefine users and items by employing the similarities of the users/items to prototypes instead of performing clustering-based assignments to subgroups, and (2) linearly aggregate the similarity scores in the final prediction, enabling the decomposition of the recommendation score and hence an easier interpretation.

2.2 Explainability

Outside of the RS literature, various prototype-based explanation methods are proposed in a variety of machine/deep learning tasks.

These methods particularly differ in the way the prototypes are identified in the first place. As examples of such studies, Li et al. [35] explore the utilization of prototypes in the context of image classification by showing that the decision of a network to classify the image of a digit can be explained by the similarity of the image to the prototypes that look like the digit. In their work, a decoder is trained to visualize and interpret the prototypes. Chen et al. [14] further extend this work by learning latent prototypes that match a portion of the latent representation of inputs, allowing for a more fine-grained explanation. Various approaches to learn prototypes are proposed in the literature. Bien and Tibshirani [11] select prototypes from training data by solving a set cover problem over the inputs, and perform classification based on top-1 nearest neighbor search. Wu and Tabak [64] find prototypes as a convex combination of inputs and utilize them in regression tasks. In contrast to the mentioned studies and similar to our work, Li et al. [35] learn the prototypes from scratch, allowing us to flexibly measure the similarity between the prototypes and data instances in the latent space.

Explainability in RSs has been the focus of several works. Zhang et al. [69] and later Cheng et al. [17] exploit external information, such as opinionated reviews, to provide interpretable user/item representations in terms of *aspects*, namely attributes that characterize a user/item. Barkan et al. [6] propose to model a user via an attentive mixture of personas which explain the recommendation of an item based on the affinity between the user's personas and the item itself. Another approach is to use several statistical tools to extract post-hoc explanations of the existing models in order to provide rationales for explaining the recommendations. Some of these post-hoc methods include using association rules [48], influence functions [16], and linear models [44]. More related to the work at hand, Fusco et al. [25] and Pan et al. [46] focus on designing interpretable models, which can inherently provide explanations in terms of contributions of the user/item features. Our ProtoMF model differs from the above-mentioned approaches in the following ways: (1) ProtoMF does not leverage fixed external features to explain the recommendations, allowing any type of external information to be used to interpret the prototypes and (2) ProtoMF models provide a novel explanation approach based on analyzing the contributions of user and item prototypes to the recommendation.

2.3 Fairness and Bias in Recommender Systems

Another topic related to explainability is fairness and bias in RSs. In this direction, recent studies show that RS algorithms deliver different recommendation performances to different groups of users (e. g., in the sense of gender, age, or personality) [33, 42, 43], raising the concern that these algorithms (may unwantedly) encode personal/sensitive information. For example, Ekstrand et al. [22] and more recently Melchiorre et al. [42] show that a variety of common RS algorithms perform worse in terms of accuracy and beyond-accuracy metrics on female users. Motivated by the mentioned studies and further contributing to this line of research, we explore whether some of the learned user prototypes also capture the gender information of users.

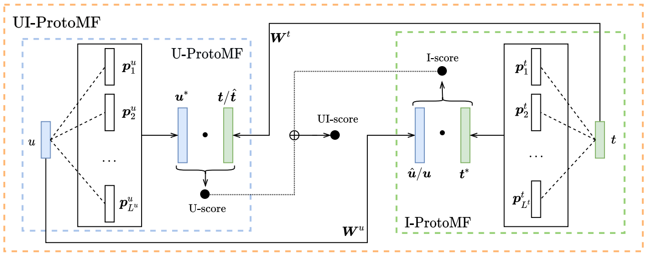


Figure 1: ProtoMF models.

3 METHODOLOGY

In this section, we describe our ProtoMF models in detail. We first introduce the *User Prototype Matrix Factorization* (U-ProtoMF) model, followed by its item-based equivalent *Item Prototype Matrix Factorization* (I-ProtoMF), and finally the *User-Item Prototype Matrix Factorization* (UI-ProtoMF) model achieved by combining the first two models.

Let $\mathcal{U} = \{u_i\}_{i=1}^N$ and $\mathcal{T} = \{t_j\}_{j=1}^M$ be the set of N users and M items, respectively. We assume that we only have access to the implicit interaction data $\mathcal{I} = \{(u_i, t_j)\}$, where (u_i, t_j) indicates that user u_i has interacted with item t_j. For brevity, we omit the user and item indexes when referring to any user or item.

Our ProtoMF models build on top of the widely-used Matrix Factorization (MF) methods [31, 53], which carry out recommendations by assigning an embedding vector $\boldsymbol{u}$ to each user and $\boldsymbol{t}$ to each item, both with embedding size d. This results in the set of user vectors $\{\boldsymbol{u}_i\}_{i=1}^N$ and item vectors $\{\boldsymbol{t}_j\}_{j=1}^M$. Using these embeddings, MF defines the recommendation score as the dot product of the corresponding user and item embeddings. The models explained in what follows also utilize user/item embeddings and expand MF by including user/item prototypes.

3.1 U-ProtoMF

The U-ProtoMF model is founded on the assumption that there exist several *prototypical* users, characterized by the various patterns of item consumption, shared among the users of the system. For example, in the context of music and movie recommendation, a user prototype may embody the preference of users in listening to Folk Metal music tracks or in highly enjoying Drama and Romance movies.

The U-ProtoMF model follows this idea by introducing a set of L^u user prototypes ($L^u \ll N$) denoted with $\mathcal{P}^u$, where each user prototype is defined as an embedding vector $\boldsymbol{p}^u$ with dimension d. The model then provides a new representation of each user u as $\boldsymbol{u}^*$, defined as the vector of the similarities of $\boldsymbol{u}$ to each of the user prototype vectors $\boldsymbol{p}^u$, as formulated below:

$$\boldsymbol{u}^* = \begin{bmatrix} \text{sim}(\boldsymbol{u}, \boldsymbol{p}_1^u) \\ \dots \\ \text{sim}(\boldsymbol{u}, \boldsymbol{p}_{L^u}^u) \end{bmatrix} \in \mathbb{R}^{L^u}, \qquad \text{sim}(\boldsymbol{a}, \boldsymbol{b}) = 1 + \frac{\boldsymbol{a}^\top \boldsymbol{b}}{\|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|} \quad (1)$$

where the similarity function sim is defined as the shifted cosine similarity and $\|\boldsymbol{x}\|$ is the L^2-norm of $\boldsymbol{x}$. This definition of the similarity function guarantees that all the values of $\boldsymbol{u}^*$ are positive in the range of 0 to 2. Lastly, U-ProtoMF measures the user-item affinity score (that u will interact with t) as a linear combination of the new user representation with the corresponding item embedding as shown below:

$$\text{U-score}(u,t) = \sum_{l=1}^{L^u} s_l^{\text{user}}, \qquad \boldsymbol{s}^{\text{user}} = \boldsymbol{u}^* \odot \boldsymbol{t} \tag{2}$$

where $\odot$ indicates the element-wise multiplication, $\boldsymbol{s}^{\text{user}}$ is the resulting user score vector, and $\boldsymbol{t} \in \mathbb{R}^{L^u}$ the item embedding. The above formulation is in fact the dot product of $\boldsymbol{u}^*$ and $\boldsymbol{t}$, and can be written as: $\text{U-score}(u,t) = \boldsymbol{u}^{*\top}\boldsymbol{t}$. We intentionally formulate $\text{U-score}(u,t)$ as in Eq. 2, as in this form the vector $\boldsymbol{s}^{\text{user}}$ breaks down the final $\text{U-score}(u,t)$ into separate user prototype scores. As we will see in Section 5.2, this characteristic is particularly beneficial to explain recommendations. A scheme of U-ProtoMF is shown in the left side of Figure 1. To train our model, we opt for the cross-entropy/softmax loss [52] given the data $\mathcal{I}$ over the model parameters Θ, defined below:

$$\begin{aligned} \mathcal{L}_{rec}(\mathcal{I},\Theta) &= -\sum_{(u,t)\in\mathcal{I}} \ln p(t|u) + \lambda_{L2}\,\|\Theta\| \\ p(t|u) &= \frac{e^{\text{U-score}(u,t)}}{\sum_{j=1}^{M} e^{\text{U-score}(u,t_j)}} \end{aligned} \tag{3}$$

where $\|\Theta\|$ indicates the L^2-norm, added to the loss through the hyperparameter λ_{L2} as regularization term. Inspired by Li et al. [35], we introduce two additional interpretability terms to the recommendation loss. These terms aim to ensure that each user is associated with at least one prototype and vice versa, done by increasing the similarity values of the most similar pairs. These terms in fact impose an *inclusion criteria* [7, 35] by forcing each user (and each prototype) to "get matched" with at least one prototype (one user). The first term $R_{\{\mathcal{P}^u\to\mathcal{U}\}}$ defines this criterion from the side of user prototypes to users, by increasing the similarity of each user prototype to the corresponding user with the largest similarity value, formulated as follows:

$$R_{\{\mathcal{P}^u\to\mathcal{U}\}} = -\frac{1}{L^u}\sum_{l=1}^{L^u} \max_{i\in[1,..,N]} \text{sim}(\boldsymbol{u}_i, \boldsymbol{p}_l^u) \tag{4}$$

The second term $R_{\{\mathcal{U}\to\mathcal{P}^u\}}$ states the criterion from the side of users to user prototypes:

$$R_{\{\mathcal{U}\to\mathcal{P}^u\}} = -\frac{1}{N}\sum_{i=1}^{N} \max_{l\in[1,..,L^u]} \text{sim}(\boldsymbol{u}_i, \boldsymbol{p}_l^u) \tag{5}$$

The final loss is therefore defined as follows:

$$\mathcal{L}_{\text{U-Proto}} = \mathcal{L}_{rec} + \lambda_1 R_{\{\mathcal{P}^u\to\mathcal{U}\}} + \lambda_2 R_{\{\mathcal{U}\to\mathcal{P}^u\}} \tag{6}$$

where λ_1 and λ_2 are hyperparameters, tuning the degrees of the effects of the inclusion criteria. In practice, since the number of users (N) is commonly very high, the full computation of $R_{\{\mathcal{P}^u\to\mathcal{U}\}}$ and $R_{\{\mathcal{U}\to\mathcal{P}^u\}}$ over all users in every training batch is very costly. To mitigate this problem, we compute these terms over a sampled subset of users, namely the ones appearing in each given training batch. Since the data is expected to be randomly shuffled, our in-batch sampling approach can be considered as an unbiased approximation of Eq. 4 and Eq. 5.

The U-ProtoMF model enables an easier interpretation of the system and its recommendations. First, the representation of every user is now (re)defined as a vector $\boldsymbol{u}^*$ of positive values. Each value of $\boldsymbol{u}^*$ corresponds to a specific consumption characteristic, where the characteristics are defined by user prototypes. Second, since the recommendation score is the dot product of $\boldsymbol{u}^*$ and $\boldsymbol{t}$ (Eq. 2), the item embeddings dimensions can be seen as weights of the corresponding characteristic (defined by user prototypes). For example, in music recommendation, a Heavy Metal song will likely have a higher value (weight) for the feature corresponding to the user prototype representing Metal fans. Lastly, the definition of the recommendation score as a linear function – the summation of the weighted prototype similarities in $\boldsymbol{s}^{\text{user}}$ – provides a favorable characteristic for interpretability by allowing to discern the different contributions of user prototypes.

3.2 I-ProtoMF

The I-ProtoMF model follows the same structure as U-ProtoMF while introducing the concept of prototypes only from the item side. In particular, I-ProtoMF assumes the existence of several prototypical items intended to capture the different co-consumption patterns arising within the dataset. For example, an item prototype might be a representative of the items that fall within a specific musical genre or product category.

Following U-ProtoMF, I-ProtoMF first defines a set of L^t item prototypes $\mathcal{P}^t$, each defined with an embedding $\boldsymbol{p}^t$ with dimension d (expectedly, $L^t \ll M$). I-ProtoMF then provides a new representation for each item t as the similarity of its vector to the item prototype vectors, formulated below:

$$\boldsymbol{t}^* = \begin{bmatrix} \text{sim}(\boldsymbol{t}, \boldsymbol{p}_1^t) \\ \dots \\ \text{sim}(\boldsymbol{t}, \boldsymbol{p}_{L^t}^t) \end{bmatrix} \in \mathbb{R}^{L^t} \tag{7}$$

Using $\boldsymbol{t}^*$, the final score is computed as:

$$\text{I-score}(u,t) = \sum_{l=1}^{L^t} s_l^{\text{item}}, \qquad \boldsymbol{s}^{\text{item}} = \boldsymbol{t}^* \odot \boldsymbol{u} \tag{8}$$

where user embeddings are in $\mathbb{R}^{L^t}$. Similarly to U-ProtoMF, I-ProtoMF's score is also in fact $\text{I-score}(u,t) = \boldsymbol{u}^\top\boldsymbol{t}^*$, while defining the intermediate vector s_l^{item} supports the recommendation explainability, as discussed in Section 5.2. We show I-ProtoMF's architecture on the right of Figure 1. Similar to U-ProtoMF, I-ProtoMF is enriched with two inclusion criteria defined below.

$$\begin{aligned} R_{\{\mathcal{P}^t\to\mathcal{T}\}} &= -\frac{1}{L^t}\sum_{l=1}^{L^t} \max_{j\in[1,..,M]} \text{sim}(\boldsymbol{t}_j, \boldsymbol{p}_l^t) \\ R_{\{\mathcal{T}\to\mathcal{P}^t\}} &= -\frac{1}{M}\sum_{i=1}^{M} \max_{l\in[1,..,L^t]} \text{sim}(\boldsymbol{t}_i, \boldsymbol{p}_l^t) \end{aligned} \tag{9}$$

Putting all together, the loss function is defined as:

$$\mathcal{L}_{\text{I-Proto}} = \mathcal{L}_{rec} + \lambda_3 R_{\{\mathcal{P}^t\to\mathcal{T}\}} + \lambda_4 R_{\{\mathcal{T}\to\mathcal{P}^t\}} \tag{10}$$

where λ_3 and λ_4 are hyperparameters and $\mathcal{L}_{rec}$ is equivalent to Eq. 3 replacing U-score with I-score. Similar to U-ProtoMF, this formulation enables the interpretation of recommendation scores (in this case from the perspective of items), via the different contributions of the item prototypes.

	ml-1m	lfm2b-1mon	AmazonVid
# Users	6,034	3,555	6,950
# Males/Females	4,326/1,708	2,965/590	-
	(72%/28%)	(83%/17%)	-
# Items	3,125	77,985	14,494
# Interactions	574,376	877,365	132,209

Table 1: Statistics of the datasets after filtering.

3.3 UI-ProtoMF

The U-ProtoMF and I-ProtoMF models enable the explanation of recommendations in terms of prototypes from the user and item side, respectively. A natural extension is to simply combine these two models to exploit the benefits of both under the hood of one single model. We provide this by introducing the UI-ProtoMF model, which computes the recommendation score as the sum of the scores of both models.

While UI-ProtoMF contains both U-ProtoMF and I-ProtoMF as two separate units, the embeddings of users and items can be shared across these two units. To this end, UI-ProtoMF defines two linear transformations, one from the user embeddings to the space of item prototypes, and the other from the item representations to the user prototypes space, defined below:

$$\hat{\boldsymbol{u}} = \boldsymbol{W}^u \boldsymbol{u} \in \mathbb{R}^{L^t}, \quad \hat{\boldsymbol{t}} = \boldsymbol{W}^t \boldsymbol{t} \in \mathbb{R}^{L^u} \tag{11}$$

Using these embeddings, the final score of UI-ProtoMF is computed as the sum of the dot products, formulated below:

$$\text{UI-score}(u,t) = \text{U-score}(u,t) + \text{I-score}(u,t) = \boldsymbol{u}^{*\top} \hat{\boldsymbol{t}} + \hat{\boldsymbol{u}}^\top \boldsymbol{t}^* \tag{12}$$

Figure 1 depicts a diagram of UI-ProtoMF. Accordingly, the loss is the sum of the loss functions:[1]

$$\mathcal{L}_{\text{UI-Proto}} = \mathcal{L}_{\text{U-Proto}} + \mathcal{L}_{\text{I-Proto}}$$

In the definition of UI-score, we particularly opt for the sum of the scores and avoid any non-linear combinatorial function. This design choice enables us to easily separate the contribution of each unit (U-ProtoMF or I-ProtoMF) to the final recommendation score. Each score can then be traced back to its corresponding unit for providing interpretations.

4 EXPERIMENT SETUP

In this section, we describe our experiment setup, namely the datasets, baselines, training and evaluation methods, and hyperparameter tuning. To ensure reproducibility, we publicly share our code and settings on **https://github.com/hcai-mms/ProtoMF**.

Datasets. We conduct our experiments on three datasets, covering movies, video games, and music domains. We consider an implicit feedback setting where user-item interactions are provided as binary values: 1 if the user interacted with the item and 0 otherwise. The statistics of the datasets are summarized in Table 1.

(1) MovieLens-1M[2] **(ml-1m)** [29] contains 1 million movie ratings on a scale from 1 to 5. As commonly done, ratings above 3.5 are treated as positive interactions [7, 36]. The dataset also contains demographic information of the users, including gender (see Table 1). Additionally, we perform 5-core filtering, namely we only keep the users that interact with at least 5 distinct items and only the items that were consumed by at least 5 distinct users. **(2) LFM2b-1Month (lfm2b-1mon)** [42] is a one-month extract of the large LFM-2b dataset.[3] The dataset contains music listening histories of Last.fm users.[4] The considered subset corresponds to the last month of the dataset (20/02/2020 - 19/03/2020) and considers only users whose gender information are provided. We further filter the dataset by removing the outlier users that listened to more than the 99th percentile of all the users, keeping only users with age between 10 to 95, and performing 10-core filtering. **(3) Amazon Video Games**[5] **(AmazonVid)** [40] consists of the ratings on the Amazon's Video Games category on a 1 to 5 scale. We consider ratings above 3.5 as positives and perform 5-core filtering.

Performance Comparison and Evaluation. We evaluate the recommendation performance of our introduced models to assess their effectiveness in practice. We evaluate U-ProtoMF, I-ProtoMF, and UI-ProtoMF, and compare them with three baseline algorithms, namely *Matrix Factorization* (MF) [31, 53], *Representative-based Matrix Factorization* (RBMF) [38], and *Anchor-based Collaborative Filtering* (ACF) [7]. MF is the baseline matrix factorization model that computes the affinity score as the dot product of the learned latent user/item representations. RBMF and ACF are representative prototype-based methods, as explained in Section 2. We evaluate the performance of the algorithms with two standard accuracy metrics, namely Hit Ratio (HitRatio) and Normalized Discounted Cumulative Gain (NDCG), and report the results at a cutoff of 10 (the results for other cutoffs are provided in the repository). To obtain a final score, we average the metrics over all the users. We test the significance of improvements using Mann-Whitney U test [41], correct p-values for multiple comparisons using Bonferroni correction [12], and aggregate the p-values over the seeds using Fisher's method [24]. We consider an improvement significant if $p < 0.01$.

Data Splits. We split each dataset according to the leave-one-out strategy [20] for every user. More specifically, for each user we order their item interactions according to the timestamps (we keep only the earliest interaction if multiple ones with the same item exist). The last interaction of the user is used as test, while the penultimate one as validation set. The rest of the interactions constitutes the training set. During training and evaluation, for each positive user-item interaction we sample x negative items not interacted with by the user, and rank the positive item among the sampled ones. We then compute loss and performance metrics on the resulting ranking. We fix the number x of negative samples (sampled uniformly at random) to 99 for evaluation, while we treat x as a hyperparameter for training.

Hyperparameter Tuning. We carry out an extensive hyperparameter optimization to evaluate the effectiveness of our approach. In summary, for all models we tune: optimization and loss-related hyperparameters, negative sampling hyperparameters for training, embedding size, and batch size. For ACF and ProtoMF we further tune the strength of the regularization losses and the number of

[1] $\mathcal{L}_{rec}$ is included only once and is based on the UI-score.

[2] https://grouplens.org/datasets/movielens/1m/

[3] http://www.cp.jku.at/datasets/LFM-2b/

[4] https://www.last.fm/

[5] http://jmcauley.ucsd.edu/data/amazon/index_2014.html

Model	ML-1M		AmazonVid		LFM2B-1MON	
	NDCG	HitRatio	NDCG	HitRatio	NDCG	HitRatio
MF	.326	.571	.140	.255	.118	.215
RBMF	.282	.505	.093	.166	.279†	.384†
ACF	.335	.597†	.202†	.392†	.291†	.517†
U-ProtoMF	.333	.583	.152†	.276†	.179†	.322†
I-ProtoMF	.303	.544	.194†	.371†	.251†	.457†
UI-ProtoMF	**.383**†‡	**.657**†‡	**.220**†‡	**.401**†	**.347**†‡	**.579**†‡

Table 2: Evaluation results w.r.t. accuracy metrics at cutoff 10. The sign † indicates significant improvement over MF while ‡ indicates significant improvement over ACF.

anchors/prototypes. The complete table of hyperparameters and their relative value ranges is reported in the repository.[6] We employ Tree-structured Parzen Estimators [8, 9] and evaluate, for each model, 100 sampled hyperparameter configuration. We fix the number of epochs to 100, however, we prematurely stop training if we observe no improvement of HitRatio @10 over the validation set for 10 consecutive epochs. For the LFM2B-1MON dataset, we further employ the trial-scheduler HyperBand [34] to speed up the experiments. Finally, we pick the model with the highest HitRatio @10. We repeat the whole procedure for three unique seeds and report the mean of the metrics on the test set.

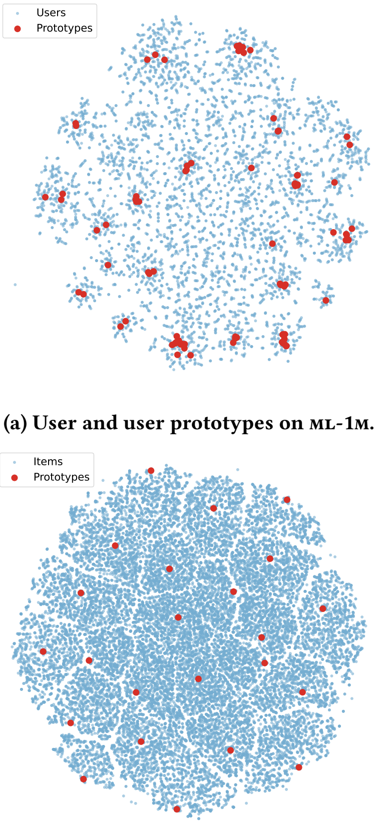
5 RESULTS

In this section, we first report the obtained results in terms of accuracy metrics. We then explain the methods to interpret the learned prototypes and lay out our approach to provide explanations for recommendations using ProtoMF models. Lastly, we showcase the existence of gender bias in the UI-ProtoMF model.

5.1 Evaluation Results

Table 2 shows the evaluation results of the models for the three datasets.[7][8] The sign † shows the significant improvements of the models over MF, and ‡ over ACF. Based on the results, we observe that all three ProtoMF models mostly provide significant improvements to MF, where UI-ProtoMF in particular shows consistent improvements on the three datasets and two metrics. Comparing among the baselines, ACF shows consistently better performance. Our proposed UI-ProtoMF method also significantly outperforms the ACF model on both accuracy metrics over all datasets (with the only exception for HitRatio on AmazonVid). These results indicate the high effectiveness of UI-ProtoMF for recommendations in comparison with the baselines, achieved by combining the benefits of the U-ProtoMF and I-ProtoMF models.

To provide a full picture, we also compare the models in terms of parameter complexity. To this end, let us assume a simplified setting with N users, M items, K (user or item) prototypes, and dimension d for any latent vector. MF contains $(N + M) \times d$ parameters, while ACF and UI-ProtoMF add $K \times d$ and $4K \times d$ parameters to MF, respectively. However, we should consider that in RS scenarios (as in our experiments) N and M are commonly much larger than both



(a) User and user prototypes on ML-1M.

(b) Item and item prototypes on LFM2B-1MON.

Figure 2: t-SNE visualization (perplexity=5, metric=cosine) of users/items and prototypes latent space.

d and K, and therefore these extra parameters (for both ACF and UI-ProtoMF) only add a small portion to the parameters of baseline MF. In the following sections, we use UI-ProtoMF and discuss how this model can provide explanations for its recommendations.

Finally, let us have a look at a visualization of the space of prototypes. Figure 2a shows the learned embeddings of users (blue) and user prototypes (red) of the ML-1M dataset, projected onto a two-dimensional space using t-SNE [62]. Evidently, the prototypes appear at the center of formed user clusters, and there are no outliers among prototypes (as a result of the inclusion regularization terms – see Section 3). The visualization also indicates that users might be close to more than one prototype, enabling a higher capacity for the model to (re)define user embeddings, as users are inherently complex in their consumption behavior, whose encoding may therefore require more than one prototype. Figure 2b provides a similar visualization with respect to items and item prototypes on the LFM2B-1MON dataset.

5.2 Explaining UI-ProtoMF Recommendations

Our first step towards explaining recommendations is to interpret what patterns of item consumption the prototypes capture, considering the user and item prototypes separately. In particular, we

[6] https://github.com/hcai-mms/ProtoMF/blob/main/protomf_appendix.pdf
[7] The standard deviations of the results, averaged over the model type, are as follows. HitRatio: .008 for ML-1M, .014 for AmazonVid, .034 for LFM2B-1MON. NDCG: .007 for ML-1M, .006 for AmazonVid, .021 for LFM2B-1MON.
[8] Results for other cutoff values are at https://github.com/hcai-mms/ProtoMF/blob/main/protomf_appendix.pdf.

User Prototype 71	User Prototype 55	User Prototype 37	Item Prototype 3	Item Prototype 6	Item Prototype 24
Fugitive, The *Action|Thriller*	**Star Trek: First Contact** *Action|Adventure|Sci-Fi*	**Cinderella** *Anim.|Children's|Musical*	**City of Angels** *Romance*	**Friday the 13th: The Final Chapter** *Horror*	**Terminal Velocity** *Action*
Seven (Se7en) *Crime|Thriller*	**Star Trek: Generations** *Action|Adventure|Sci-Fi*	**Little Mermaid, The** *Anim.|Child.|Com.|Musical*	**It Could Happen to You** *Drama|Romance*	**Friday the 13th Part 3: 3D** *Horror*	**Drop Zone** *Action*
In the Line of Fire *Action|Thriller*	**Star Trek VI: The Undiscovered Country** *Action|Adventure|Sci-Fi*	**Sleeping Beauty** *Anim.|Children's|Musical*	**Walk in the Clouds, A** *Drama|Romance*	**Friday the 13th Part 2** *Horror*	**Sudden Death** *Action*
Heat *Action|Crime|Thriller*	**Forbidden Planet** *Sci-Fi*	**She's All That** *Comedy|Romance*	**One Fine Day** *Drama|Romance*	**Friday the 13th Part VII: The New Blood** *Horror*	**Marked for Death** *Action|Drama*
Die Hard *Action|Thriller*	**Star Trek IV: The Voyage Home** *Action|Adventure|Sci-Fi*	**101 Dalmatians** *Animation|Children's*	**Sommersby** *Drama|Mystery|Romance*	**Friday the 13th Part VI: Jason Lives** *Horror*	**Glimmer Man, The** *Action|Thriller*

Table 3: Top-5 related items of three representative user prototypes (left) and item prototypes (right) based on the UI-ProtoMF model on the ml-1m dataset.

approach the interpretation of user prototypes by observing ProtoMF's recommendations when fed with synthetic user inputs that maximally activate the prototypes, similar to previous studies [35, 71]. We interpret item prototypes by identifying the items closest to each prototype, similar to Alvarez-Melis and Jaakkola [3]. The following examples are taken from the trained UI-ProtoMF model on the ml-1m dataset. Due to lack of space, more examples for lfm2b-1mon are provided in the repository.[9]

Interpreting User Prototypes. To interpret which item-consumption characteristics a user prototype embodies, we create a synthetic similarity vector $\boldsymbol{u}^*$ of an imaginary user, where a maximum value is given to the corresponding user prototype in the vector, and all other values are set to zero. We then compute recommendations for this imaginary user. Adopting this method, the left part of Table 3 shows the recommendations of three representative user prototypes. Each of these captures a specific movie consumption behavior. For example, prototype 71 represents a prototypical user who enjoys action movies and thrillers, while prototype 55 prefers Sci-Fi movies, mostly of the same series; and the last one's top movie recommendations mostly consist of animated movies.

Interpreting Item Prototypes. Since item and item prototype embeddings lie in the same space, interpretation of an item prototype can be achieved by simply identifying its nearest item neighbors. The right part of Table 3 shows three representative examples. As can be seen, each item prototype tends to match a specific movie genre. Here, item prototype 3 is close to Drama and Romance movies. Item prototype 6's closest neighbors are all part of the same Horror movie series, while item prototype 24 is a representative of Action movies.

Explaining Recommendations. Having elaborated the methods to interpret user and item prototypes, we now focus on the explainability of UI-ProtoMF's recommendations. Our approach to generate explanations utilizes the degree to which each prototype has contributed to the final affinity score. As discussed in Section 3, this score is the sum of the scores stemming from U-ProtoMF and I-ProtoMF (see Eq. 12).

Referring to U-ProtoMF, the final U-score is a sum of the user prototypes contributions s_l^{user} (see Section 3.1). Now, understanding the recommendation score of U-ProtoMF involves first assessing the user prototypes' contributions based on the values of $\boldsymbol{s}^{\text{user}}$ (for example by focusing on the ones with the highest contributions). The recommendation is then explained based on the interpretation of user prototypes, described before. In addition, we can further deepen our explanation by recalling how the score s_l^{user} is computed: as the product of a user- and an item-specific component. In fact, a value s_l^{user} can be high (or low) due to the corresponding values of the underlying user prototype and item embedding, respectively, u_l^* and $\hat{t}_l$. A similar procedure can be applied to the score of I-ProtoMF, by using s_l^{item} to detect the most contributing item prototypes.

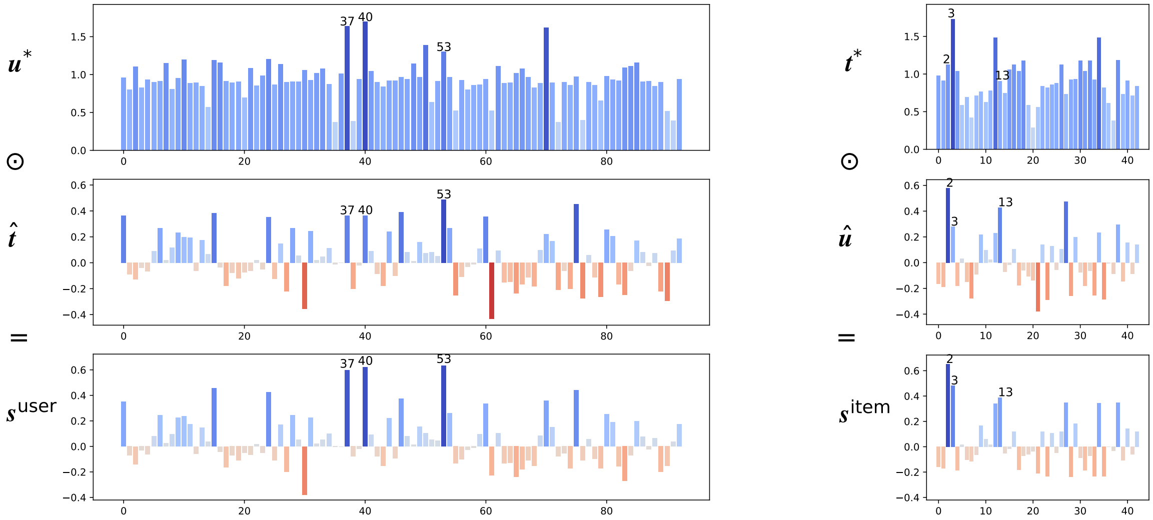
Let us clarify this procedure with an example, focusing on the UI-ProtoMF recommendations of an arbitrary user in the ml-1m dataset. As the first recommendation, the model predicts *Pretty Woman*, a movie in Comedy and Romance genre. Figure 3a and Figure 3b show the values of the vectors involved in this prediction for U-ProtoMF and I-ProtoMF, respectively. In particular, the user-to-prototype/item-to-prototype similarity values, the values of user/item embeddings, and the scores are shown in the top, the middle, and the bottom plots, respectively.

On the U-ProtoMF side, the model detects that user prototypes 53, 40, and 37 have the highest contribution to the final score (highest values in $\boldsymbol{s}^{\text{user}}$). The interpretation results of these three user prototypes are reported in Table 4, indicating similar movies in genres such as Comedy and Romance for prototypes 53 and 40, and mostly animated movies for prototype 37. We further observe that the high values of these three prototypes in $\boldsymbol{s}^{\text{user}}$ are caused by different components. In particular, the user-to-prototype similarities $\boldsymbol{u}^*$ have high values for prototypes 40 and 37, while a relatively lower value for prototype 53 (see the lower plot in Figure 3a). On the other hand, the item embedding $\hat{\boldsymbol{t}}$ (representing the movie) has a high value on prototype 53 and lower values on the other two (middle plot in Figure 3a), resulting in overall high values in the final scores.

Similarly, on the I-ProtoMF side, item prototypes 2, 3, and 13 represent the major contributors. The interpretation of these item prototypes are shown in Table 4, demonstrating the tendency toward Romance and Drama genres, with prototype 2 further including Comedy and Musical genres. Similarly, the scores in $\boldsymbol{s}^{\text{item}}$ can be traced back to the corresponding user embedding values $\hat{\boldsymbol{u}}$ and item-to-prototype similarities $\boldsymbol{t}^*$.

As a last remark, ProtoMF's explanations can be flexibly conveyed in different manners to different target audiences [1, 4]. A global analysis of the prototypes scores and similarities values can

[9] https://github.com/hcai-mms/ProtoMF/blob/main/protomf_appendix.pdf



(a) U-ProtoMF

(b) I-ProtoMF

Figure 3: Visualizing the prototype similarities, weights, and scores of UI-ProtoMF for the recommendation of the movie "Pretty Woman" for an arbitrary user of ML-1M.

User Prototype 53	User Prototype 40	User Prototype 37	Item Prototype 3	Item Prototype 2	Item Prototype 13
Roman Holiday *Comedy|Romance*	**Runaway Bride** *Comedy|Romance*	**Cinderella** *Anim.|Children's|Musical*	**City of Angels** *Romance*	**Broadway Melody, The** *Musical*	**Chambermaid on the Titanic, The** *Romance*
To Catch a Thief *Com.|Romance|Thriller*	**She's All That** *Comedy|Romance*	**Little Mermaid, The** *Anim.|Child.|Com.|Musical*	**It Could Happen to You** *Drama|Romance*	**Slipper and the Rose, The** *Adventure|Musical|Romance*	**Dreaming of Joseph Lees** *Romance*
Sabrina *Comedy|Romance*	**Affair to Remember, An** *Romance*	**Sleeping Beauty** *Anim.|Children's|Musical*	**Walk in the Clouds, A** *Drama|Romance*	**Penny Serenade** *Drama|Romance*	**Passion of Mind** *Romance|Thriller*
Sleepless in Seattle *Comedy|Romance*	**Double Jeopardy** *Action|Thriller*	**She's All That** *Comedy|Romance*	**One Fine Day** *Drama|Romance*	**Perils of Pauline, The** *Comedy*	**Golden Bowl, The** *Drama*
While You Were Sleeping *Comedy|Romance*	**Ever After: A Cinderella Story** *Drama|Romance*	**101 Dalmatians** *Animation|Children's*	**Sommersby** *Drama|Mystery|Romance*	**Damsel in Distress, A** *Comedy|Musical|Romance*	**Up at the Villa** *Drama*

Table 4: Top-5 related items of three user prototypes (left) and item prototypes (right) mentioned in Figure 3.

interest a more technical audience (e. g., engineers and data analysts) to understand the general behavior of the recommender system and to correct possible misconducts (e. g., biases). We will shortly provide a case for this. At the same time, providing the system's end-users with an interactive visualization of the most contributing prototypes along with their descriptions can largely support the system's transparency for the users.

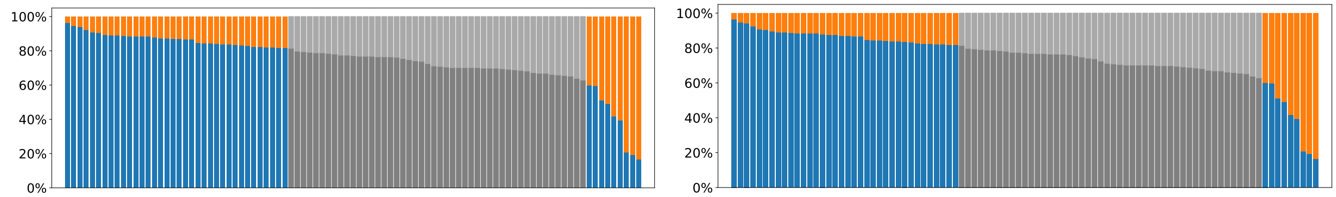
5.3 Showcasing Societal Biases

The trained UI-ProtoMF model can be used to study whether the learned prototypes encode existing gender biases in the datasets. For this reason, we focus on the ML-1M and LFM2B-1MON datasets, since they provide users' gender information. To carry out our study, we select a set of the most similar users to each user prototype and consider them as the representatives of the prototype. This set comprises all users whose similarities to the prototype are above the 95th percentile. We then calculate the gender counts of these representatives and compare them with the corresponding gender distribution in the whole dataset, checking for statistically significant differences. This is done by carrying out several Fisher exact tests [23] with an alpha level of 1%, and further correct for multiple comparisons using the Bonferroni method [12].

The results are depicted in Figure 4a and Figure 4b, where each bar represents the gender distribution, as the proportion of male to female users, of the representative users of a user prototype, sorted from the prototypes with the highest share of males (blue) to the ones with the highest share of females (orange). We also define a neutral area (gray) corresponding to the prototypes with non-significant differences in gender distributions.

We observe 36 male vs. 9 female user prototypes (among a total of 93) in ML-1M, and 7 male vs. 6 female prototypes (among 43) in LFM2B-1MON, evidencing the existence of user prototypes that encode specific gender attributes. Particularly, we notice that ML-1M has a considerably higher number of male-related user prototypes compared to female-related ones. These observations are in accordance with the ones made in previous studies [22, 42], demonstrating the existence of stereotypical biases in recommendations, which we show for specific user prototypes. Table 5a reports the interpretation results of the most female/male-related user prototypes provided in ML-1M. According to these results, the prototypical male users have the tendency to watch Sci-Fi and Thriller movies, while the prototypical female users mostly watch Romance movies.

Steerable Bias Mitigation in Recommendation. In the following, we briefly discuss an interesting capability of ProtoMF, namely



(a) ML-1M

(b) LFM2B-1MON

Figure 4: Gender distribution of the representatives users per prototype. The orange/blue area indicates the proportion of females/males, and the gray area refers to gender neutral prototypes.

User Prototype 14	User prototype 40
2001: A Space Odyssey *Drama|Mystery|Sci-Fi|Thriller*	**Runaway Bride** *Comedy|Romance*
Blade Runner *Film-Noir|Sci-Fi*	**She's All That** *Comedy|Romance*
2010 *Mystery|Sci-Fi*	**Affair to Remember, An** *Romance*
Gattaca *Drama|Sci-Fi|Thriller*	**Double Jeopardy** *Action|Thriller*
Sneakers *Crime|Drama|Sci-Fi*	**Ever After: A Cinderella Story** *Drama|Romance*

(a)

Original ($\lambda = 1.0$)	$\lambda = 0.33$	$\lambda = 0.0$
My Fair Lady *Musical|Romance*	**Sound of Music, The** *Musical*	**Raiders of the Lost Ark** *Action|Adven.*
Sound of Music, The *Musical*	**Braveheart** *Action|Drama|War*	**Braveheart** *Action|Drama|War*
Shakespeare in Love *Comedy|Romance*	**Raiders of the Lost Ark** *Action|Adven.*	**Star Wars: Episode VI** *Act.|Adven.|Romance|Sci-Fi|War*
Gone with the Wind *Drama|Romance|War*	**Shakespeare in Love** *Comedy|Romance*	**Star Wars: Episode IV** *Action|Adven.|Fantasy|Sci-Fi*
Little Mermaid, The *Anim.|Child.|Comedy|Musical*	**My Fair Lady** *Musical|Romance*	**African Queen, The** *Action|Adven.|Romance|War*

(b)

Table 5: (a) Most male-related (left) and female-related (right) user prototypes in ML-1M. (b) Example of applying the controllable bias mitigation method to the recommendations of a sample female user. The effects of some female-related user prototypes are dampened with the factor λ.

providing flexible and controllable recommendations. More specifically, ProtoMF's recommendations can be changed at run-time by manually adjusting the values of user/item-to-prototype similarity vectors ($\boldsymbol{u}^*$ or $\boldsymbol{t}^*$). In fact, since the affinity score is computed as an independent sum of prototypes' contributions, we are able to increase/decrease these values, and therefore change the recommendation. This capability can potentially be exploited in various scenarios, such as user-centric bias mitigation [5, 10, 49], or diversifying item recommendations to counteract filter bubbles [47, 60].

Let us showcase this capability with an example in the context of gender bias mitigation. Based on the results presented in Figure 4, we first find the top-3 most female-related user prototypes in ML-1M. We then alter the corresponding values of these user prototypes in $\boldsymbol{u}^*$ by multiplying them with a factor λ. We apply this method to the recommendations of a female user, and report the top-5 recommendations in the original case, with $\lambda = 0.33$, and $\lambda = 0$ in Table 5b. As shown, the recommendation of the user moves from movies in Romance and Comedy genres (more strongly associated with the female users in the dataset) to Action and Sci-Fi. This simple method suggests a potentially appealing framework to mitigate or adjust gender bias, as particularly different degrees of interventions can be set at inference time according to the wish of end-users or system designers.

6 CONCLUSION AND OUTLOOK

In this paper, we propose ProtoMF, a novel collaborative filtering approach that leverages user and item prototypes to provide accurate and explainable recommendations. As a result of its design, ProtoMF's recommendations can be explained in terms of contributions of user/item prototypes, the latter representing item-consumption characteristics of real users and items of the system. To this end, we provide an explanation framework that allows us to interpret the user/item prototypes and investigate their contributions to the predicted affinity scores. Furthermore, we show through extensive quantitative experiments that ProtoMF significantly outperforms Matrix Factorization and two prototype-based approaches in terms of Hit Ratio and NDCG. Moreover, we expose the existence of gender biases in the learned user prototypes by identifying prototypes with significant inclinations to the consumption behavior that is stereotypical of male or female users. We conclude with an idea for steering the amount of gender bias in recommendations made by ProtoMF.

As promising future research directions, we envision a thoughtful examination of the effects of gender-related (possibly other demographics as well) user prototypes on the recommendations in terms of accuracy and beyond-accuracy metrics, similarly done in as [42] and mitigate likely biases in the recommendations by exploiting ProtoMF's controllable recommendations. Furthermore, we believe that including external features of users and items, such as contextual information or audio features, into ProtoMF might further benefit the interpretability of the prototypes. Lastly, we would like to assess the usefulness of our explanations in terms of the goals defined by Tintarev and Masthoff [59] by involving a real (technical and non-) audience.

ACKNOWLEDGMENTS

This work received financial support by the Austrian Science Fund (FWF): P33526 and DFH-23; and by the State of Upper Austria and the Federal Ministry of Education, Science, and Research, through grant LIT-2020-9-SEE-113 and LIT-2021-YOU-215. The first author would also like to thank Dunja Životin for her support.

REFERENCES

[1] Darius Afchar, Alessandro B. Melchiorre, Markus Schedl, Romain Hennequin, Elena V Epure, and Manuel Moussallam. 2022. Explainability in Music Recommender Systems. *AI Magazine* 43, 2 (2022), 190–208.

[2] Marharyta Aleksandrova, Armelle Brun, Anne Boyer, and Oleg Chertov. 2017. Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem. *Journal of Intelligent Information Systems* 48, 2 (2017), 365–397.

[3] David Alvarez-Melis and Tommi S. Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 7786–7795.

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115.

[5] Ricardo Baeza-Yates. 2020. *Bias in Search and Recommender Systems.* Association for Computing Machinery, New York, NY, USA, 2. https://doi.org/10.1145/3383313.3418435

[6] Oren Barkan, Yonatan Fuchs, Avi Caciularu, and Noam Koenigstein. 2020. *Explainable Recommendations via Attentive Multi-Persona Collaborative Filtering.* Association for Computing Machinery, New York, NY, USA, 468–473. https://doi.org/10.1145/3383313.3412226

[7] Oren Barkan, Roy Hirsch, Ori Katz, Avi Caciularu, and Noam Koenigstein. 2021. *Anchor-Based Collaborative Filtering.* Association for Computing Machinery, New York, NY, USA, 2877–2881. https://doi.org/10.1145/3459637.3482056

[8] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems* (Granada, Spain) *(NIPS'11)*. Curran Associates Inc., Red Hook, NY, USA, 2546–2554.

[9] J. Bergstra, D. Yamins, and D. D. Cox. 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28* (Atlanta, GA, USA) *(ICML'13)*. JMLR.org, I–115–I–123.

[10] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 405–414. https://doi.org/10.1145/3209978.3210063

[11] Jacob Bien and Robert Tibshirani. 2011. Prototype selection for interpretable classification. *The Annals of Applied Statistics* 5, 4 (2011), 2403–2424.

[12] Carlo E Bonferroni. 1935. Il calcolo delle assicurazioni su gruppi di teste. *Studi in onore del professore salvatore ortu carboni* (1935), 13–60.

[13] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.

[14] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. 2019. This Looks Like That: Deep Learning for Interpretable Image Recognition. *Advances in Neural Information Processing Systems* 32 (2019), 8930–8941.

[15] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions.

[16] Weiyu Cheng, Yanyan Shen, Linpeng Huang, and Yanmin Zhu. 2019. Incorporating Interpretability into Latent Factor Models via Fast Influence Analysis. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 885–893. https://doi.org/10.1145/3292500.3330857

[17] Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C Kanjirathinkal, and Mohan Kankanhalli. 2019. MMALFM: Explainable recommendation by leveraging reviews and images. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–28.

[18] European Commission. 2020. Proposal for a Regulation of the European Parliament and of the Council on a Single Market for Digital Services (Digital Services Act) and amending Directive 2000/31/EC: COM(2020) 825 final.

[19] European Commission. 2021. Proposal for a Regulation of the European Parliament and of the Council. Laying Down Harmonised Rules on Artificial Intelligence (ARTIFICIAL INTELLIGENCE ACT) and Amending Certain Union Legislative Acts). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206

[20] Paolo Cremonesi, Roberto Turrin, Eugenio Lentini, and Matteo Matteucci. 2008. An evaluation methodology for collaborative recommender systems. In *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*. IEEE, 224–231.

[21] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. 2011. Collaborative Filtering Recommender Systems. *Found. Trends Hum.-Comput. Interact.* 4, 2 (Feb. 2011), 81–173.

[22] Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. 2018. All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research, Vol. 81)*, Sorelle A. Friedler and Christo Wilson (Eds.). PMLR, 172–186. https://proceedings.mlr.press/v81/ekstrand18b.html

[23] Ronald A Fisher. 1922. On the interpretation of χ 2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society* 85, 1 (1922), 87–94.

[24] Ronald Aylmer Fisher. 1992. Statistical methods for research workers. In *Breakthroughs in statistics.* Springer, 66–70.

[25] Francesco Fusco, Michalis Vlachos, Vasileios Vasileiadis, Kathrin Wardatzky, and Johannes Schneider. 2019. RecoNet: An Interpretable Neural Architecture for Recommender Systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19.* International Joint Conferences on Artificial Intelligence Organization, 2343–2349. https://doi.org/10.24963/ijcai.2019/325

[26] Christian Ganhör, David Penz, Navid Rekabsaz, Oleg Lesota, and Markus Schedl. 2022. Mitigating Consumer Biases in Recommendations with Adversarial Training. In *Proceedings of the 45th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2022.* ACM.

[27] Fabio Gasparetti, Giuseppe Sansonetti, and Alessandro Micarelli. 2021. Community detection in social recommender systems: a survey. *Applied Intelligence* 51, 6 (2021), 3975–3995.

[28] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51, 5 (2018), 1–42.

[29] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[30] Peter Hase, Chaofan Chen, Oscar Li, and Cynthia Rudin. 2019. Interpretable Image Recognition with Hierarchical Prototypes. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 7, 1 (Oct. 2019), 32–40. https://ojs.aaai.org/index.php/HCOMP/article/view/5265

[31] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[32] Klara Krieg, Emilia Parada-Cabaleiro, Markus Schedl, and Navid Rekabsaz. 2022. Do Perceived Gender Biases in Retrieval Results Affect Relevance Judgements?. In *Proceedings of the Workshop on Algorithmic Bias in Search and Recommendation at the European Conference on Information Retrieval (ECIR-BIAS 2022)*. Springer International Publishing, Cham, 104–116.

[33] Oleg Lesota, Alessandro Melchiorre, Navid Rekabsaz, Stefan Brandl, Dominik Kowald, Elisabeth Lex, and Markus Schedl. 2021. *Analyzing Item Popularity Bias of Music Recommender Systems: Are Different Genders Equally Affected?* Association for Computing Machinery, New York, NY, USA, 601–606. https://doi.org/10.1145/3460231.3478843

[34] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2017. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research* 18, 1 (2017), 6765–6816.

[35] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network That Explains Its Predictions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) *(AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Palo Alto, California, U.S., Article 432, 8 pages.

[36] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 689–698. https://doi.org/10.1145/3178876.3186150

[37] Zachary C Lipton. 2018. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16, 3 (2018), 31–57.

[38] Nathan N. Liu, Xiangrui Meng, Chao Liu, and Qiang Yang. 2011. Wisdom of the Better Few: Cold Start Recommendation via Representative Based Rating Elicitation. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (Chicago, Illinois, USA) *(RecSys '11)*. Association for Computing Machinery, New York, NY, USA, 37–44. https://doi.org/10.1145/2043932.2043943

[39] Jingwei Ma, Jiahui Wen, Mingyang Zhong, Liangchen Liu, Chaojie Li, Weitong Chen, Yin Yang, Hongkui Tu, and Xue Li. 2019. DBRec: Dual-Bridging Recommendation via Discovering Latent Groups. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1513–1522. https://doi.org/10.1145/3357384.3357892

[40] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. Association for Computing

Machinery, New York, NY, USA, 43–52. https://doi.org/10.1145/2766462.2767755

[41] Patrick E McKnight and Julius Najab. 2010. Mann-Whitney U Test. *The Corsini encyclopedia of psychology* (2010), 1–1.

[42] Alessandro B. Melchiorre, Navid Rekabsaz, Emilia Parada-Cabaleiro, Stefan Brandl, Oleg Lesota, and Markus Schedl. 2021. Investigating gender fairness of recommendation algorithms in the music domain. *Information Processing and Management* 58, 5 (2021), 102666. https://doi.org/10.1016/j.ipm.2021.102666

[43] Alessandro B. Melchiorre, Eva Zangerle, and Markus Schedl. 2020. *Personality Bias of Music Recommendation Algorithms*. Association for Computing Machinery, New York, NY, USA, 533–538. https://doi.org/10.1145/3383313.3412223

[44] Caio Nóbrega and Leandro Marinho. 2019. Towards Explaining Recommendations through Local Surrogate Models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (Limassol, Cyprus) *(SAC '19)*. Association for Computing Machinery, New York, NY, USA, 1671–1678. https://doi.org/10.1145/3297280.3297443

[45] Mark O'Connor and Jon Herlocker. 1999. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR workshop on recommender systems*, Vol. 128. UC Berkeley, Association for Computing Machinery, New York, NY, USA.

[46] Deng Pan, Xiangrui Li, Xin Li, and Dongxiao Zhu. 2020. Explainable Recommendation via Interpretable Feature Mapping and Evaluation of Explainability. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2690–2696. Main track.

[47] Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you.* Penguin UK, 20 Vauxhall Bridge Rd, London.

[48] Georgina Peake and Jun Wang. 2018. Explanation Mining: Post Hoc Interpretability of Latent Factor Models for Recommendation Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 2060–2069. https://doi.org/10.1145/3219819.3220072

[49] Navid Rekabsaz, Simone Kopeinik, and Markus Schedl. 2021. Societal Biases in Retrieved Contents: Measurement Framework and Adversarial Mitigation of BERT Rankers. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 306–316. https://doi.org/10.1145/3404835.3462949

[50] Navid Rekabsaz and Markus Schedl. 2020. *Do Neural Ranking Models Intensify Gender Bias?* Association for Computing Machinery, New York, NY, USA, 2065–2068. https://doi.org/10.1145/3397271.3401280

[51] Navid Rekabsaz, Robert West, James Henderson, and Allan Hanbury. 2021. Measuring Societal Biases from Text Corpora with Smoothed First-Order Cooccurrence. In *Proceedings of the Fifteenth International AAAI Conference on Web and Social Media, ICWSM 2021, held virtually, June 7-10, 2021.* AAAI Press, 549–560.

[52] Steffen Rendle. 2021. Item Recommendation from Implicit Feedback.

[53] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[54] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.

[55] Aravind Sankar, Junting Wang, Adit Krishnan, and Hari Sundaram. 2021. *ProtoCF: Prototypical Collaborative Filtering for Few-Shot Recommendation.* Association for Computing Machinery, New York, NY, USA, 166–175. https://doi.org/10.1145/3460231.3474268

[56] Badrul M Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2002. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, Vol. 1. Citeseer, 291–324.

[57] Markus Schedl, Stefan Brandl, Oleg Lesota, Emilia Parada-Cabaleiro, David Penz, and Navid Rekabsaz. 2022. LFM-2b: A Dataset of Enriched Music Listening Events for Recommender Systems Research and Fairness Analysis. In *ACM SIGIR Conference on Human Information Interaction and Retrieval* (Regensburg, Germany) *(CHIIR '22)*. Association for Computing Machinery, New York, NY, USA, 337–341. https://doi.org/10.1145/3498366.3505791

[58] Lei Shi, Wayne Xin Zhao, and Yi-Dong Shen. 2017. Local representative-based matrix factorization for cold-start recommendation. *ACM Transactions on Information Systems (TOIS)* 36, 2 (2017), 1–28.

[59] Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender systems handbook*. Springer, Berlin/Heidelberg, Germany, 353–382.

[60] Matus Tomlein, Branislav Pecher, Jakub Simko, Ivan Srba, Robert Moro, Elena Stefancova, Michal Kompan, Andrea Hrckova, Juraj Podrouzek, and Maria Bielikova. 2021. *An Audit of Misinformation Filter Bubbles on YouTube: Bubble Bursting and Recent Behavior Changes*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3460231.3474241

[61] Lyle H Ungar and Dean P Foster. 1998. Clustering methods for collaborative filtering. In *AAAI workshop on recommendation systems*, Vol. 1. Menlo Park, CA, 114–129.

[62] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[63] Michail Vlachos, Francesco Fusco, Charalambos Mavroforakis, Anastasios Kyrillidis, and Vassilios G. Vassiliadis. 2014. Improving Co-Cluster Quality with Application to Product Recommendations. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (Shanghai, China) *(CIKM '14)*. Association for Computing Machinery, New York, NY, USA, 679–688. https://doi.org/10.1145/2661829.2661980

[64] Chenyue Wu and Esteban G Tabak. 2017. Prototypal analysis and prototypal regression.

[65] Yao Wu, Xudong Liu, Min Xie, Martin Ester, and Qing Yang. 2016. CCCF: Improving Collaborative Filtering via Scalable User-Item Co-Clustering. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (San Francisco, California, USA) *(WSDM '16)*. Association for Computing Machinery, New York, NY, USA, 73–82. https://doi.org/10.1145/2835776.2835836

[66] Bin Xu, Jiajun Bu, Chun Chen, and Deng Cai. 2012. An Exploration of Improving Collaborative Recommender Systems via User-Item Subgroups. In *Proceedings of the 21st International Conference on World Wide Web* (Lyon, France) *(WWW '12)*. Association for Computing Machinery, New York, NY, USA, 21–30. https://doi.org/10.1145/2187836.2187840

[67] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. 2005. Scalable Collaborative Filtering Using Cluster-Based Smoothing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil) *(SIGIR '05)*. Association for Computing Machinery, New York, NY, USA, 114–121. https://doi.org/10.1145/1076034.1076056

[68] George Zerveas, Navid Rekabsaz, Daniel Cohen, and Carsten Eickhoff. 2022. Mitigating Bias in Search Results Through Contextual Document Reranking and Neutrality Regularization. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2532–2538. https://doi.org/10.1145/3477495.3531891

[69] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit Factor Models for Explainable Recommendation Based on Phrase-Level Sentiment Analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval* (Gold Coast, Queensland, Australia) *(SIGIR '14)*. Association for Computing Machinery, New York, NY, USA, 83–92. https://doi.org/10.1145/2600428.2609579

[70] Gang Zhao, Mong Li Lee, Wynne Hsu, Wei Chen, and Haoji Hu. 2013. Community-Based User Recommendation in Uni-Directional Social Networks. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (San Francisco, California, USA) *(CIKM '13)*. Association for Computing Machinery, New York, NY, USA, 189–198. https://doi.org/10.1145/2505515.2505533

[71] Pablo Zinemanas, Martín Rocamora, Marius Miron, Frederic Font, and Xavier Serra. 2021. An Interpretable Deep Learning Model for Automatic Sound Classification. *Electronics* 10, 7 (2021), 850.