A Fairness-aware Hybrid Recommender System

Golnoosh Farnadi
University of California, Santa Cruz
gfarnadi@ucsc.edu

Pigi Kouki
University of California, Santa Cruz
pkouki@soe.ucsc.edu

Spencer K. Thompson
University of California, Santa Cruz
spkthomp@ucsc.edu

Sriram Srinivasan
University of California, Santa Cruz
ssriniv9@ucsc.edu

Lise Getoor
University of California, Santa Cruz
getoor@soe.ucsc.edu

ABSTRACT

The increasing use of recommender systems in domains affecting people's lives has raised concerns about possible biases and discrimination that such systems might exacerbate. There are two primary kinds of biases inherent in recommender systems: observation bias and bias stemming from imbalanced data. Observation bias exists due to a feedback loop which causes the model to learn to only predict recommendations similar to previous ones. Imbalance in data occurs when systematic societal, historical, or other ambient bias is present in the data. In this paper, we address both biases by proposing a hybrid fairness-aware recommender system. Our model provides efficient and accurate recommendations by incorporating multiple user-user and item-item similarity measures, content, and demographic information, while addressing recommendation biases. We implement our model using a powerful and expressive probabilistic programming language called *probabilistic soft logic*. We experimentally evaluate our approach on a popular movie recommendation dataset, showing that our proposed model can provide more accurate and fairer recommendations, compared to a state-of-the art fair recommender system.

KEYWORDS

Fairness, Recommender Systems, Movie Recommender Systems, Probabilistic Soft Logic

1 INTRODUCTION

Targeted recommendations have become increasingly important to business owners in order to reach their potential customers in a variety of domains such as commerce, employment, dating, health, education, and governance. However, when targeting users, biases can have a negative feedback on subgroups of users. For instance, one study [1] shows that female users of Google have a lower chance of accessing hiring ads with high-paying executive jobs. Studying biases and fairness in machine learning is an emerging research area that is receiving increasing attention [2, 3, 4]. Methods mitigating unfairness in machine learning systems [5, 6, 7] can be extended to the case of fairness-aware recommender systems.

Defining fairness, especially for recommender systems, is challenging. In this paper, we do not address this question, we assume that someone has given us the definitions of which attributes/subpopulation we want to maintain fairness toward, and present a scalable, declarative formulation for achieving fairness relative to the given subgroups. In the fairness domain, a population of vulnerable individuals known as the *protected group*, can be defined by an attribute value upon which discrimination is based (such as gender, ethnicity, or religion). A fair recommender system should provide rankings to the protected group that are the same as the *unprotected group*. The majority of popular recommender system algorithms (e.g., collaborative-filtering) make use of user behavior to generate recommendations. Powerful as they are, these methods usually inherit the biases that exist in the data which may cause the system to present unfair recommendations.

There are two primary kinds of biases that can be inherited from data: observation bias and bias that comes from imbalance in data [8]. Observation bias is due to the existence of a feedback loop in the system. An item displayed by the recommender system will get an action, which is then used to retrain the model and thus reinforces the recommender system's algorithm to show more items similar to previous recommendations. If a user is never exposed to an item, that user cannot provide an opinion on it. For example, if a user on a movie streaming website was never shown a movie from the action genre, it is difficult for the system to know the user's interest level in action movies. Imbalance in data is caused when a systematic bias is present in the data, due to societal, historical, or other ambient biases. Since the model is unaware of such biases, addressing them is not straightforward. For example, in job recommendation, due to social bias, the data might have large evidence to show that nursing is a successful profession for females. However, this does not mean that female users must be recommended to pursue professions in nursing in order to be successful. Since the data is heavily biased towards successful female nurses, the model ends up using this single source of knowledge to decide that an optimal recommendation is to suggest nursing to female users.

These biases have been explored and addressed in different contexts with use of multi-arm bandits and diversity-based recommendations [9, 10, 11]. Even though these approaches tend to handle biases by increasing the diversity of a recommender system, they do not directly address the issue of fairness. More recently, fairness in recommender systems has been explored through the use of fairness metrics. For instance, Yao and Huang [8], show that fairness can be both measured and imposed on a matrix factorization (MF) method, using five different fairness metrics. Burke et al. [12] recently introduced an approach that aspires for both personalization and fairness via neighborhood balancing with the sparse linear method. Other work studies the issue of modeling in the presence of gender-imbalanced data. As an example, Sapiezynski et al. [13] found that gender representation-imbalance in academic data on students led to a higher accuracy in detecting struggling male students, as opposed to their female classmates.

FATREC 2018, October 2018, Vancouver, Canada
2018. ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

In this work, we first start by using a hybrid recommender system, called HyPER [14] to produce recommenations. HyPER incorporates a variety of signals including user-user similarities, item-item similarities, and content and demographic information. We then extend HyPER to a fairness-aware recommender system by addressing observation biases and biases coming from imbalanced data. We implement our fairness-aware recommender system as a single unified model, by using a probabilistic programming language called probabilistic soft logic (PSL) [15]. PSL has been used for providing hybrid recommendations [14], for providing a fairness-aware framework in relational settings [16], and a variety of other tasks. In this work, we unify these two lines of work and propose a fairness-aware hybrid recommendation system. We make use of a constraint-based approach to fairness which extends PSL with a new maximum a posteriori (MAP) inference algorithm that maximizes the a posteriori values of unknown variables subject to fairness guarantees using a set of hard fairness constraints. Like this previous work, we model various protected and unprotected groups with relational dependencies in the model; however, in our proposed work, we address biases in recommender systems with a set of soft fairness constraints that are directly expressed in the model. We design two sets of fairness constraints with latent variables that are able to: 1) detect and address biases in item ratings coming from imbalanced data and 2) integrate rules that address biases coming from item group ratings to prevent observation bias. These two sets of constraints are able to capture relational dependencies among users and items to collectively predict accurate ratings for both protected and unprotected groups.

In this paper, we make the following contributions: 1) we present a probabilistic programming approach for building fair hybrid recommender systems; 2) we experimentally study fairness on the popular Movielens dataset; 3) we show that a fair recommender system can outperform a recommender system not trained for fairness in both accuracy and fairness evaluation metrics; and 4) we experimentally show that our fair recommender system surpasses the current state-of-the-art fair recommender system in both accuracy and a variety of fairness metrics.

The remainder of the paper is structured as follows: In Section 2, we present our model in detail. We start with an overview of the modeling language that we use to build a fair movie recommender system, i.e., PSL (Section 2.1). We briefly describe the movie recommender system that we use, i.e., HYPER (Section 2.2), and then we explain how we can make it fair (Section 2.3). In Section 3 we present our evaluation results. Finally, we conclude with a discussion and our plans for future work in Section 4.

2 APPROACH

In this section, we describe how we extend an existing hybrid recommender system to provide fair recommendations. We first introduce the modeling framework that we use to define our model, called probabilistic soft logic (PSL). PSL is a declarative language that uses first-order logic to define a model. We choose PSL to propose a fairness-aware recommender system because its expressiveness allows us to model recommender systems as well as fairness constraints in a unified model. Next, we describe how we define a hybrid movie recommender system using the basic principles of an existing hybrid recommender system (HyPER). Finally, we discuss how we extend our recommender system to account for fairness by using a set of PSL rules capturing fairness with relational dependencies between users and items.

2.1 PSL

Probabilistic soft logic (PSL) [15] is a probabilistic programming language that uses a first-order logical rules to define a graphical model. PSL uses continuous random variables in the $[0, 1]$ unit interval and specifies factors using convex functions, allowing tractable and efficient inference. PSL defines a Markov random field associated with a conditional probability density function over random variables Y conditioned on evidence $\mathbf{X}$,

$$P(\mathbf{Y}|\mathbf{X}) \propto \exp\Big(-\sum_{j=1}^{m} w_j \phi_j(\mathbf{Y}, \mathbf{X})\Big)\,, \tag{1}$$

where ϕ_j is a convex potential function and w_j is an associated weight which determines the importance of ϕ_j in the model. The potential ϕ_j takes the form of a *hinge-loss*:

$$\phi_j(\mathbf{Y}, \mathbf{X}) = (max\{0, \ell_j(\mathbf{X}, \mathbf{Y})\})^{p_j}\,. \tag{2}$$

Here, ℓ_j is a linear function of $\mathbf{X}$ and $\mathbf{Y}$, and $p_j \in \{1, 2\}$ optionally squares the potential, resulting in a *squared-loss*. The resulting probability distribution is log-concave in Y, so we can solve MAP inference exactly via convex optimization to find the optimal Y. The convex formulation of PSL is the key to efficient, scalable inference in models with many complex inter-dependencies.

PSL derives the objective function by translating logical rules that specify dependencies between variables and evidence into hinge-loss functions. PSL achieves this translation by using the *Lukasiewicz* norm and co-norm to provide a relaxation of Boolean logical connectives [15]. For example, $a \Rightarrow b$ corresponds to the hinge function $\max(a - b, 0)$, and $a \wedge b$ corresponds to $\max(a + b - 1, 0)$. We refer the reader to [15] for a detailed description of PSL.

To illustrate PSL in the movie recommendation context, the following rule encodes that users tend to rate movies of their preferred genres highly:

$$\textsc{LikesGenre}(\mathtt{u}, \mathtt{g}) \wedge \textsc{IsGenre}(\mathtt{m}, \mathtt{g}) \Rightarrow \textsc{Rating}(\mathtt{u}, \mathtt{m})\,,$$

where $\textsc{LikesGenre}(\mathtt{u}, \mathtt{g})$ is a binary observed predicate, $\textsc{IsGenre}(\mathtt{m}, \mathtt{g})$ is a continuous observed predicate in the interval $[0, 1]$ capturing the affinity of the movie to the genre, and $\textsc{Rating}(\mathtt{u}, \mathtt{m})$ is a continuous variable to be inferred, which encodes the star rating as a number between 0 and 1, with higher values corresponding to higher star ratings. For example, we could instantiate `u = Jim`, `g = classics` and `m = Casablanca`. This instantiation results in a hinge-loss potential function in the HL-MRF,

$$\begin{aligned}\max(&\textsc{LikesGenre}(\mathtt{Jim}, \mathtt{classics})\\ &+ \textsc{IsGenre}(\mathtt{Casablanca}, \mathtt{classics})\\ &- \textsc{Rating}(\mathtt{Jim}, \mathtt{Casablanca}) - 1, 0)\,.\end{aligned}$$

PSL has been successfully applied in various domains, such as explanations in recommender systems [17], user modeling in social media [18], stance prediction in online forums [19], energy disaggregation [20], and knowledge graph identification [21].

2.2 PSL Recommendations Model

In recent work, Kouki et al. [14] introduced HyPER, a hybrid recommender system that uses PSL. The model consists of rules that can incorporate a wide range of information sources, such as user-user

and item-item similarity measures, content information, and user and item average predictors. HyPER uses the rules together with the input data to perform inference and define a probability distribution over the recommended items, capturing the extent to which a given user will like a given item. HyPER provides a generic and extensible recommendation framework with the ability to incorporate any other sources of information that are available in any custom dataset. In this work, we focus on movie recommendations. We use a subset of all the rules proposed in HyPER and, given its extensibility, we add rules to leverage dataset-specific information available in our movie dataset. We propose a hybrid movie-recommender system which consists of the following rules:

2.2.1 Mean-Centering Priors. We first encode rules in our model that encourage the ratings to be close to the average for each user and each item. Each individual user of a recommender system has his/her own biases in rating items. Likewise, each item's rating is influenced by its overall popularity. To address such biases, we introduce the following rules:

$$\text{AVERAGEUSERRATING}(\mathtt{u}) \Rightarrow \text{RATING}(\mathtt{u}, \mathtt{i})$$
$$\neg\text{AVERAGEUSERRATING}(\mathtt{u}) \Rightarrow \neg\text{RATING}(\mathtt{u}, \mathtt{i})$$
$$\text{AVERAGEITEMRATING}(\mathtt{i}) \Rightarrow \text{RATING}(\mathtt{u}, \mathtt{i})$$
$$\neg\text{AVERAGEITEMRATING}(\mathtt{i}) \Rightarrow \neg\text{RATING}(\mathtt{u}, \mathtt{i}) .$$

The predicate $\text{RATING}(\mathtt{u}, \mathtt{i})$ takes a value in the interval $[0, 1]$ and represents the normalized value of the rating that a user $\mathtt{u}$ gave to an item $\mathtt{i}$. The predicate $\text{AVERAGEUSERRATING}(\mathtt{u})$ represents the average of the ratings over the set of items that user $\mathtt{u}$ provided in the training set. Similarly, $\text{AVERAGEUSERRATING}(\mathtt{i})$ represents the average of the user ratings an item $\mathtt{i}$ has received. The pair of PSL rules per-user and per-item penalizes the predicted rating for being different from this average.

2.2.2 Neighborhood-based Collaborative Filtering. We define PSL rules that capture the basic principle of the neighborhood-based approach. We introduce the following user-based collaborative filtering rule to capture the intuition that similar users give similar ratings to the same items:

$$\text{SIMILARUSERS}_{\text{SIM}}(\mathtt{u}_1, \mathtt{u}_2) \wedge \text{RATING}(\mathtt{u}_1, \mathtt{i}) \Rightarrow \text{RATING}(\mathtt{u}_2, \mathtt{i}) .$$

The predicate $\text{SIMILARUSERS}_{\text{SIM}}(\mathtt{u}_1, \mathtt{u}_2)$ is binary, with value 1 iff $\mathtt{u}_1$ is one of the k-nearest neighbors of $\mathtt{u}_2$. The above rule represents a template for hinge functions which reduce the probability of predicted ratings as the difference between $\text{RATING}(\mathtt{u}_2, \mathtt{i})$ and $\text{RATING}(\mathtt{u}_1, \mathtt{i})$ increases, for users that are neighbors. Similarly, we can define PSL rules to capture the intuition of item-based collaborative filtering methods, namely that similar items should have similar ratings from the same users:

$$\text{SIMILARITEMS}_{\text{SIM}}(\mathtt{i}_1, \mathtt{i}_2) \wedge \text{RATING}(\mathtt{u}, \mathtt{i}_1) \Rightarrow \text{RATING}(\mathtt{u}, \mathtt{i}_2) .$$

As before, the predicate $\text{SIMILARITEMS}_{\text{SIM}}(\mathtt{i}_1, \mathtt{i}_2)$ is binary, with value 1 iff $\mathtt{i}_1$ is one of the k-nearest neighbors of $\mathtt{i}_2$. The similarities can be calculated with any similarity measure SIM. In this model, we use the most popular similarity measures in the neighborhood-based recommendations literature [22]. More specifically, we apply cosine similarity measures to calculate similarities between users and items; for the items we additionally apply the adjusted cosine similarity and the Pearson's correlation measures.

2.2.3 Using Additional Sources of Information. The movie dataset that we use offers demographic information about the users and content information about the items. We can use this information to define similar users using demographic information and similar items using content. We introduce the following rules:

$$\text{SIMILARUSERS}_{\text{DEMO}}(\mathtt{u}_1, \mathtt{u}_2) \wedge \text{RATING}(\mathtt{u}_1, \mathtt{i}) \Rightarrow \text{RATING}(\mathtt{u}_2, \mathtt{i})$$
$$\text{SIMILARITEMS}_{\text{CONTENT}}(\mathtt{i}_1, \mathtt{i}_2) \wedge \text{RATING}(\mathtt{u}, \mathtt{i}_1) \Rightarrow \text{RATING}(\mathtt{u}, \mathtt{i}_2) .$$

In the first rule, the predicate $\text{SIMILARUSERS}_{\text{DEMO}}(\mathtt{i}_1, \mathtt{i}_2)$ represents users that have similar demographic features (which in our case is age, gender, and occupation). In the second rule, the predicate $\text{SIMILARITEMS}_{\text{CONTENT}}(\mathtt{i}_1, \mathtt{i}_2)$ represents items that have similar content-based features (which in our case in the genre of a movie).

2.2.4 Negative Prior. In recommender systems, there is usually a huge number of items available (e.g., movies). However, every single user has rated a small number of these items. To model our general belief that it is unlikely for a user to rate a movie, we introduce the following prior rule:

$$\neg\text{RATING}(\mathtt{u}, \mathtt{i}) .$$

2.3 Fair PSL Recommendations Model

The power of our PSL recommender system lies in the fact that fairness can be modeled with a set of logical rules. This is particularly important given the diverse set of applications powered by recommender systems and that fairness in many of those applications is multi-faceted [23]. For example, a recommender system suggesting job applications needs to ensure that two applicants with a similar professional profile receive similar job recommendations. At the same time, the recommender system needs to ensure market diversity and avoid monopoly domination by giving similar chance to new companies to get a reasonable share of recommendations even though they have had fewer job offers compared to the established companies. In this work, we take into account the disparate impact of recommendation on protected classes of recommendation users.

To encode fairness in our model, we first introduce protected and unprotected groups. The predicate $\text{PROTECTED}(\mathtt{u})$ is binary, which indicates whether a user belongs to the protected group with value 1, or unprotected group with value 0. Note though that the protected group could be any attribute and it can be either an observed attribute in the data or a latent attribute. Here, we consider all female users to be our protected group and all male users to be our unprotected group. The goal of a fair recommender system is to provide fair ratings depending on the fairness metric used, for both protected and unprotected groups.

2.3.1 Imbalance in Data Biases. In recommender systems, various types of users may have tendency to only rate particular items. For instance, female users are more likely to shop for clothes, while male users buy tools with higher frequency. If a recommender system has access only to an imbalanced dataset, it may never recommend a particular item to a specific group of users. To avoid such bias in our model, we define the following rules:

$$\text{PROTECTED}(\mathtt{u}) \wedge \text{RATING}(\mathtt{u}, \mathtt{i}) \Rightarrow \text{PROTECTEDITEMRATING}(\mathtt{i})$$
$$\neg\text{PROTECTED}(\mathtt{u}) \wedge \text{RATING}(\mathtt{u}, \mathtt{i}) \Rightarrow \text{UNPROTECTEDITEMRATING}(\mathtt{i}) .$$

At a high level, we introduce two latent variables for each item, i.e., $\text{PROTECTEDITEMRATING}(\mathtt{i})$ and $\text{UNPROTECTEDITEMRATING}(\mathtt{i})$. These two predicates capture ratings from protected and unprotected users for each item in the data. To encode fair ratings for both groups, we add the following constraints in the model which force the value

of these two latent variables for each item to be equivalent, for both protected and unprotected groups:

$$\text{PROTECTEDITEMRATING}(\texttt{i}) \Rightarrow \text{UNPROTECTEDITEMRATING}(\texttt{i})$$

$$\text{UNPROTECTEDITEMRATING}(\texttt{i}) \Rightarrow \text{PROTECTEDITEMRATING}(\texttt{i}) .$$

Using the above rules, we are able to balance the ratings for both types of users by un-biasing the ratings for each item. Extending the recommender model that we described in Section 2.2 with these rules enables us to address imbalanced data biases. For the case of movie recommendation, our protected and unprotected groups are female users and male users respectively. Therefore, we can replace predicate PROTECTED(u) with ISFEMALE(u) that indicates whether a user is female or male.

2.3.2 Observation Biases. In addition to biases coming from imbalance in data, users may likely prefer items belong to a certain item group. For example, for the item group "genre" in the context of movie recommendations, female users are more likely to rate romance movies, while male users rate action movies with higher frequency. If users are never recommended a particular item, they will likely never provide rating data for that item. To avoid such observation biases we introduce fairness rules towards item groups in the model. Similar to the fairness rules of the previous section, we introduce latent variables for each item group with the following rules:

$$\text{PROTECTED}(\texttt{u}) \wedge \text{RATING}(\texttt{u}, \texttt{i}) \wedge \text{ITEMGROUP}(\texttt{i}, \texttt{g}) \Rightarrow \text{PROTECTEDITEMGROUPRATING}(\texttt{g})$$

$$\neg\text{PROTECTED}(\texttt{u}) \wedge \text{RATING}(\texttt{u}, \texttt{i}) \wedge \text{ITEMGROUP}(\texttt{i}, \texttt{g}) \Rightarrow \text{UNPROTECTEDITEMGROUPRATING}(\texttt{g}) .$$

Here, we introduce two latent variables for each item group by introducing two predicates, i.e., PROTECTEDITEMGROUP RATING(g) and UNPROTECTEDITEMGROUPRATING(g), that capture ratings from protected and unprotected users for items within each item group. Similarly, we force the value of these latent variables for each item group in the model to be equal using the following rules:

$$\text{PROTECTEDITEMGROUPRATING}(\texttt{g}) \Rightarrow \text{UNPROTECTEDITEMGROUPRATING}(\texttt{g})$$

$$\text{UNPROTECTEDITEMGROUPRATING}(\texttt{g}) \Rightarrow \text{PROTECTEDITEMGROUPRATING}(\texttt{g}) .$$

Our fair recommender model includes both types of fairness rules to address observational biases, and avoid biases coming from imbalanced ratings for both protected and unprotected groups. The definition of what consists an item group depends on the specific context of the recommender system. In the movie recommendation setting, we can use a movie's genre to define item groups. Therefore, in the above rules, we can replace the predicate ITEMGROUP(i, g) with the predicate ISGENRE(i, g) to capture the affinity of a movie to a genre. Note that one movie could have more than one genre, for instance the movie *Casablanca* has three genres: *classics*, *drama*, and *romance*.

We use the rules presented in this section with the rules presented in Section 2.2 to collectively infer ratings for all users. Next, we present our experimental setup for evaluating our proposed fair movie recommender system.

3 EXPERIMENTAL VALIDATION

3.1 Dataset Description

For our experiments, we use the Movielens 1M dataset [24], which has 1 000 209 ratings (ranging from 1 to 5) for 3 952 movies, from 6 040 users. Demographic data for the users (e.g., gender, age, occupation) and meta data on the movies (e.g., genre) are also provided. For movies, we follow the preprocessing steps proposed by Yao and Huang [8] and consider only movies that are tagged with at least one of the following 5 genres: *action, romance, crime, musical, sci-fi*. These genres have a relatively large difference between the number of ratings by males and females, as well as a noticeable difference in average rating by each gender. This yields a subset of the dataset that expresses a strong population imbalance and gives the potential for an unfair recommender system. This is evident from the gender-based statistics of movie genres reported in Table 2 of [8]. For example, the number of ratings per female user for romantic movies is 54.67, while for men it is 36.97. In another example, the number of ratings per female user for sci-fi movies is 31.19, while for male users it is 50.46. Again, following the filtering process proposed in Yao and Huang, we further filter the dataset by only considering users that rated more than 50 movies. These preprocessing steps produce a subset of the original Movielens 1M dataset, consisting of 443 079 ratings for 1 305 movies from 2 965 users.

3.2 Evaluation Metrics

To measure the accuracy of the movie recommender system, we report the root mean squared error (RMSE) and the mean absolute error (MAE). To measure the fairness (or unfairness) of the movie recommender system, we use the popular demographic parity measure [25] and a set of new metrics, recently introduced by Yao and Huang [8]. Here is the list of the fairness metrics we report in our experimental evaluation, along with a short description:

- *Non-parity unfairness*: measures the absolute unfairness in making predictions for two groups (a protected and an unprotected one). This metric is computed as the absolute difference between the overall average ratings of users belonging to the unprotected group and those of users belonging to the protected group.
- *Value unfairness*: measures the inconsistency in signed estimation error across the protected and unprotected user groups. This metric becomes large when predictions for one group are consistently overestimated while predictions for the other group are consistently underestimated.
- *Absolute unfairness*: measures the inconsistency in absolute estimation error across user groups. This metric is sign-agnostic and its value becomes large if one group of users consistently receives more accurate recommendations than the other.
- *Underestimation unfairness*: measures the inconsistency in how much the predictions underestimate the true ratings. This metric becomes large when the recommender system constantly predicts lower rating values than the true ratings.
- *Overestimation unfairness*: measures the inconsistency in how much the predictions overestimate the true ratings. This metric is the opposite of underestimation unfairness, i.e., when overestimation unfairness increases in a system then underestimation unfairness decreases (and vice versa). This metric becomes large

Model	RMSE (SD)	MAE (SD)	Overestimation (SD)	Absolute (SD)	Non-Parity (SD)	Underestimation (SD)	Value (SD)	Balance (SD)
(MC) Baseline	0.997 (0.003)	0.794 (0.002)	0.280 (0.001)	0.302 (0.002)	0.144 (0.001)	**0.104** (0.001)	0.385 (0.002)	0.192 (0.002)
MF [8]	0.944 (0.002)	0.760 (0.002)	0.256 (0.001)	0.282 (0.001)	0.084 (0.000)	0.139 (0.001)	0.395 (0.002)	0.198 (0.002)
Fair MF (non-parity) [8]	0.945 (0.002)	0.760 (0.002)	0.252 (0.001)	0.281 (0.001)	**0.083** (0.000)	0.145 (0.001)	0.396 (0.002)	0.199 (0.001)
Fair MF (value) [8]	0.948 (0.002)	0.762 (0.002)	0.250 (0.002)	0.279 (0.002)	0.131 (0.000)	0.140 (0.000)	0.390 (0.002)	0.197 (0.001)
(MC+CF) PSL	0.922 (0.002)	0.734 (0.001)	**0.144** (0.001)	0.261 (0.001)	0.224 (0.000)	0.210 (0.001)	**0.354** (0.002)	**0.177** (0.001)
(MC+CF+DC) PSL [14]	0.916 (0.002)	0.732 (0.002)	0.180 (0.001)	0.260 (0.002)	0.206 (0.001)	0.177 (0.001)	0.357 (0.002)	0.179 (0.001)
Fair (MC+CF) PSL	**0.908** (0.002)	**0.727** (0.001)	0.158 (0.001)	0.251 (0.001)	0.159 (0.001)	0.201 (0.001)	0.358 (0.002)	0.180 (0.001)
Fair (MC+CF+DC) PSL	0.911 (0.002)	0.730 (0.002)	0.177 (0.001)	**0.250** (0.001)	0.160 (0.001)	0.180 (0.001)	0.357 (0.002)	0.179 (0.001)

Table 1: Overall Performance of different PSL and fair MF models. Numbers in parenthesis indicate standard deviations. For each metric, we report the best value in bold. For all metrics, the smaller the value, the more accurate/fair the model is.

when the recommender system constantly predicts higher rating values than the true ratings.

- *Balance unfairness*: measures the inconsistency in how much the predictions overestimate and underestimate the true ratings. This metric is the average of underestimation and overestimation unfairness.

3.3 Experiments

We evaluate the following different versions of the PSL model:

- *Mean-centering (MC) model*: This model uses the mean-centering priors for users and items described in Section 2.2.1 and the negative prior (Section 2.2.4). Note that this model has no rules to model relational dependencies, and therefore is not using PSL capabilities. We call this model **MC Baseline**.
- *Mean-centering (MC) and collaborative filtering (CF) PSL model*: We enrich the above model by adding user and item similarity rules explained in Section 2.2.2. We note again that these similarities are computed using only collaborative filtering information. The number of similar users and items is typically set to between 20 and 50 in the literature [22], and so for each user we use the 20 most similar neighbors. This limit applies to all similarities that we use. We call this model **MC+CF PSL**.
- *Mean-centering (MC), collaborative filtering (CF), and demographic and content (DC) PSL model*: We extend the above model by adding additional information about users and items (Section 2.2.3). Specifically, we use demographic information on individual users, i.e. gender, age, and occupation, to compute the cosine similarity between user pairs. Item similarities are computed for each pair of movies, using cosine similarity among vector representations of the movies genres. We call this model **MC+CF+DC PSL**.
- *Fair Mean-centering and collaborative filtering PSL model*: We enrich the model **MC+CF PSL** by adding all the fair rules described in Section 2.3. We call this model **Fair MC+CF PSL**.
- *Fair Mean-centering, collaborative filtering, and demographic and content PSL model*: We enrich the model **MC+CF+DC PSL** by adding all the fair rules described in Section 2.3. We call this model **Fair MC+CF+DC PSL**.

We compare our fair PSL models with the fair state-of-the-art models proposed by Yao and Huang [8]. Yao and Huang proposed six different models, where five models optimized for different fairness metric. We ran all six models and we report the results of the following three: 1) **MF**, a matrix factorization algorithm that does not optimize for fairness (we include this model as the simplest MF without fairness), 2) **Fair MF (non-parity)** which optimizes for the non-parity unfairness metric (this model performed the best in terms of RMSE and MAE in the 5 splits of the dataset that we used for our experiments), and 3) **Fair MF (Value)** which optimizes for the value unfairness metric (this model performed the best in the 5 splits of the dataset that Yao and Huang [8] operated on). To run these models, we used the default values for regularization and epochs, i.e., reg=0.001 and epochs=100.

We compute the metrics described in Section 3.2 by performing 5-fold cross-validation in the filtered MovieLens dataset described in Section 3.1. We report the average cross-validated errors and unfairness values along with the standard deviation for all the models described above using the same split. We report our results in Table 1. For each metric, we report the best value in bold. For all metrics, the smaller the value, the more accurate/fairer the model is.

3.4 Results

We observed the following from Table 1:

PSL shows improved accuracy compared to MF methods: A first clear conclusion from the results is that the all PSL models outperform all three MF (fair and non-fair) models on accuracy metrics. With one exception (the simplest PSL model, MC Baseline that only uses average ratings), PSL produces a statistically significant improvement in both RMSE and MAE as measured by a paired t-test with $\alpha = 0.05$.

Adding fairness rules improves the performance of the PSL models: The addition of fairness rules to the model MC+CF PSL (which results in the model Fair MC+CF PSL) results in a relatively small decrease of 0.014 (absolute value) for the RMSE and 0.007 (absolute value) for the MAE. Similarly, the addition of fairness rules to the model MC+CF+DC PSL (which results in the model Fair MC+CF+DC PSL) results in a decrease of 0.005 (absolute value) for the RMSE and 0.002 (absolute value) for the MAE. We do not observe the same behavior for the different fair and non-fair matrix factorization models. For these cases, when trying to optimize for different fairness metrics, we observe a very small increase for the RMSE and MAE. In particular, when trying to optimize the simple MF model for non-parity unfairness(which results in the model Fair MF (non-parity)) we observe a small increase of 0.001 for the RMSE, while the MAE stays the same. Similarly, when trying to optimize the simple MF model for value unfairness metric (which results in the model Fair MF (value)) we observe a small increase of 0.004 for the RMSE and a small increase of 0.002 for the MAE.

Fair PSL models outperform fair MF models w.r.t. balance unfairness: PSL models (except for the (MC) Baseline) are better in avoiding underestimating the true ratings of the female users,

while MF models are better in avoiding overestimating the female ratings. However, by looking at the balance unfairness metric, PSL models produce more balanced ratings for female and male users when compared to MF methods.

Fair MF models outperform fair PSL models w.r.t. non-parity unfairness: All MF methods perform significantly better for the non-parity unfairness metric when compared to the PSL models. Also, optimizing for non-parity unfairness in MF causes an increase or no change in almost all the other unfairness metrics, which is consistent with the results presented in [8]. For PSL, we note that fair PSL models perform significantly better than non-fair PSL models with respect to non-parity unfairness metric.

There is no model that can be fair in all metrics: There is always a trade-off between various fairness measures. Each recommender system, according to its goal, can choose a setting which satisfies its needs. According to the results presented in Table 1, there is no method that outperforms all fairness metrics. However, all PSL models (except for the (MC) Baseline model) outperform MF models in both performance and fair predication for the following metrics: RMSE, MAE, absolute unfairness, value unfairness, and balance unfairness. Also, our proposed fair PSL models outperform all models in decreasing RMSE and MAE when predicting the true ratings of all users, and, at the same time, they consistently predict accurate ratings for female and male users.

4 CONCLUSIONS

In this paper, we proposed a fairness-aware hybrid recommender system that integrates multiple sources of user and item data to accurately recommend items to users, and addresses observation bias and biases coming from imbalance in the data. We implemented our system in a unified model with an expressive language, called *probabilistic soft logic*. Empirical evaluation on the movie recommendation domain shows that our proposed model is able to offer more accurate and, oftentimes, fairer recommendations compared to a state-of-the-art fair recommender system.

There are many avenues for expanding our work. In addition to the fairness rules that we proposed in our model, we plan to extend our fairness-aware recommender system with other rules to address other types of biases, such as biases of item providers or explicit bias by advertisers. Bias by advertisers has the potential to have a polarizing affect on recommendations. In certain cases, these biases might not stem from imbalanced data but rather from the modeling practices used. Moreover, in a real-world recommender system settings, the user-item matrix is very sparse, contrary to the sample of the MovieLens dataset that we operated on. We plan to explore the robustness of our approach when data sparsity is present. Finally, we are interested in applying our solution to other domains where fairness has legal and policy implications, such as the job recommendation setting.

REFERENCES

[1] A. Datta, Michael C. Tschantz, and A. Datta. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1), 2015.

[2] E. Bozdag. Bias in algorithmic filtering and personalization. *Ethics and Information Technology*, 15, 2013.

[3] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*, (ITCS), 2012.

[4] D. Pedreshi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *Knowledge Discovery and Data Mining Conference*, (KDD), 2008.

[5] F. Kamiran, T. Calders, and M. Pechenizkiy. Discrimination aware decision tree learning. In *International Conference on Data Mining*, (ICDM), 2010.

[6] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases*, (ECML-PKDD), 2012.

[7] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *International Conference on International Conference on Machine Learning*, (ICML), 2013.

[8] S. Yao and B. Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*, (NIPS). 2017.

[9] Q. Wang, Ch. Zeng, W. Zhou, T. Li, L. Shwartz, and G. Grabarnik. Online interactive collaborative filtering using multi-armed bandit with dependent arms. *arXiv preprint arXiv:1708.03058*, 2017.

[10] J. Wasilewski and N. Hurley. Incorporating diversity in a learning to rank recommender system. In *Florida Artificial Intelligence Research Society Conference*, (FLAIRS), 2016.

[11] N. Tintarev. Presenting diversity aware recommendations: Making challenging news acceptable. In *FATREC Workshop on Responsible Recommendation at RecSys*, (RecSys), 2017.

[12] R. Burke, N. Sonboli, and M. Mansoury. Balanced neighborhoods for fairness-aware collaborative recommendation. In *FATREC Workshop on Responsible Recommendation*, (RecSys), 2017.

[13] P. Sapiezynski, V. Kassarnig, and S. Lehmann. Academic performance prediction in a gender-imbalanced environment. In *FATREC Workshop on Responsible Recommendation*, (RecSys), 2017.

[14] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, and L. Getoor. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In *Recommender Systems Conference*, (RecSys), 2015.

[15] S. Bach, M. Broecheler, B. Huang, and L. Getoor. Hinge-loss markov random fields and probabilistic soft logic. *Journal of Machine Learning Research (JMLR)*, 18(109), 2017.

[16] G. Farnadi, B. Babaki, and L. Getoor. Fairness in relational domains. In *Conference on AI, Ethics, and Society*, 2018.

[17] P. Kouki, J. Schaffer, J. Pujara, J. O' Donovan, and L. Getoor. User preferences for hybrid explanations. In *Recommender Systems Conference*, (RecSys), 2017.

[18] G. Farnadi, S. H Bach, M-F. Moens, L. Getoor, and M. De Cock. Soft quantification in statistical relational learning. *Machine Learning*, 106(12):1971–1991, 2017.

[19] D. Sridhar, J. Foulds, M. Walker, B. Huang, and L. Getoor. Joint models of disagreement and stance in online debate. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

[20] S. Tomkins, J. Pujara, and L. Getoor. Disambiguating energy disaggregation: A collective probabilistic approach. In *International Joint Conference on Artificial Intelligence*, (IJCAI), 2017.

[21] J. Pujara, H. Miao, L. Getoor, and W. Cohen. Knowledge graph identification. In *International Semantic Web Conference*, (ISWC), 2013.

[22] X. Ning, C. Desrosiers, and G. Karypis. *A comprehensive survey of neighborhood-based recommendation methods*. Recommender Systems Handbook, Second Edition, Springer US, 1 2015.

[23] R. Burke. Multisided fairness for recommendation. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, (FAT/ML), 2017.

[24] M. Harper and J. Konstan. The movielens datasets: History and context. *Transactions on Interactive Intelligent Systems*, 2015.

[25] T. Calders, F. Kamiran, and M. Pechenizkiy. Building classifiers with independency constraints. In *International Conference on Data Mining Workshops*, (ICDMW), 2009.