This is the accepted manuscript of the article, which has been published in
MEDES '19: Proceedings of the 11th International Conference on Management of Digital EcoSystems, 95-102.
https://doi.org/10.1145/3297662.3365798

Enhancing Long Term Fairness in Recommendations with Variational Autoencoders

Rodrigo Borges
University of São Paulo & Tampere University
São Paulo, Brazil
rcborges@ime.usp.br

Kostas Stefanidis
Tampere University
Tampere, Finland
konstantinos.stefanidis@tuni.fi

ABSTRACT

Recommender systems have become indispensable for several Web sites, helping users deal with big amounts of data. They are capable of analyzing user/item interactions taking place on-line, and provide each user with a list of suggestions sorted by relevance. Items with the same or very close relevance, however, may occupy different positions in the ranking and may be exposed to completely different levels of attention. This promotes unfair treatment and can only be addressed by a long term strategy.

Variational Autoencoders (VAEs) were recently proposed as the state-of-the-art for collaborative filtering recommendations, but as every other approach, they generate homogeneous prediction scores among the highest positions. In this paper, we propose incorporating randomness in the regular operation of VAEs in order to increase the fairness (mitigate the position bias) in multiple rounds of recommendation. We argue that adding a noise component when sampling values from VAE's latent representation provides long term fairness, despite of a tolerable decrease in ranking quality (NDCG). We calculate the trade-off between unfairness and NDCG when introducing 4 different noise distributions.

The solution has proved to be a very practical one and the results point for a clear positive effect of turning recommendation far more fair, despite some small NDCG loss in Movie Lens, Netflix and MSD datasets. In our best scenario, the unfairness was reduced by 76% despite a decrease of 5% in the quality of ranking.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Fairness in Ranking; Variational Autoencoder; Collaborative Filtering; Recommendation Systems; Position Bias

1 INTRODUCTION

Recommender systems are ubiquitous nowadays, and can affect many aspects of life: from listening music to employment. The most popular approach for implementing them is Collaborative Filtering (CF), which assumes similarity between users as a matter of how they interact with a collection of items, i.e. how they share similar patterns in their previous experiences. This is taken as the basis for predicting the probabilities (scores) of a user interacting with each unseen item according to his/her preference.

In most cases, suggestions are presented as an ordered list from which one item is selected. However users usually pay more attention in the first positions, and the level of attention decreases as the position in the ranking gets higher[1]. In a situation where the greatest estimated probabilities are quite close or equal to each other, the system needs to arrange them in a proper order and necessarily present high scores in high positions. This promotes an unfair result that can only be mitigated in the long term, by changing the position of items in sequential rounds of ranking [5].

Autoencoder-based CF solutions usually take the sparse set of ratings each user gave to items as the input data. The information is encoded in a latent space as nonlinear combinations of the input [23], and the predictions of unrated items are then obtained from this new representation space back to the original input dimension, through the decoding phase.

Variational Autoencoders (VAE) were recently presented as the state-of-the-art for the CF task. With a multinomial likelihood generative model and a controlled regularization parameter, Liang et al. [17] demonstrates the possibility of estimating normal distribution parameters in the middle layer of the MLP, that enriches the rating data representation and outperforms previous neural network based approaches. The situation requires drawing samples from the inferred distributions in order to propagate values to the decoder, but it is not a trivial task to take gradients when having a sampling step. Kingma and Welling [13] proposed the *reparametrization trick* in which the sampled values are reparametrized by incorporating a normal distributed noise, so the gradient can back-propagate through the sampled variable during the training.

The fact that VAE's middle layer is learned as normal distributions has some advantages over standard autoencoders. Firstly, it ensures tolerance to noisy inputs and avoids overfitting [14]. Its ability to capture per-data-point variance provides also flexibility for obtaining a generative model. By changing its inner layer values along the range defined by their variances, it is possible to obtain coherent different outputs for the same input.

[1] We're here considering high position as having high index, and consequently lower relevance in the rankings.

Previous solutions based on VAE applied *reparametrization trick* during the training phase and considered only the mean values of the inner distributions in the test phase for obtaining deterministic results [17, 22]. In this paper, we propose incorporating the stochastic component also in the test phase. We argue that the noisy effect will vary the output scores when having the same data as input, and that unfairness is reduced despite of a small decrease in the quality of the ranking in a simulated sequential recommendation session. The higher the variance of the new component, the greater the effect in the predicted scores, and consequently in the ranking order.

In a nutshell, the contributions of this work can be summarized as follows:

- We provide an analysis of incorporating a random component in the regular operation of Variational Autoencoders applied here in the task of CF, but that can be extended to any other of its applications.
- To our knowledge, this is the first attempt to address the position bias in rankings by incorporating the solution inside the model instead of a post processing phase.
- The unfairness measurements decrease as the variance of the normal distributed stochastic component increases despite of a small reduction in the ranking quality, until an optimum variance value.

2 FAIRNESS PROBLEM

When applying machine learning solutions to decision-making situations it is crucial to have a fair treatment among the algorithm's targets (objects), as well as among the possible options available for selection (subjects). In the case of classifiers, as for example of an algorithm selecting students for entering the university, the system must ensure that there is no bias in the output that would promote unbalanced treatment among groups of students. In the case of ranking solutions, for example when ranking drivers in an online taxi platform, it is necessary to ensure that every equally relevant option has equal opportunity of been presented to users.

In classification algorithms the quality of fair is usually tied to avoiding discrimination against individual or group of users. *Fairness through awareness* [8] ensures indifference to sensitive attributes (e.g. age, gender, race) as a strategy for balancing the output result. Hardt et al. [11] suggests that the rate of true positives must be the same across all groups, providing equal oportunity. For a wide list of definitions of fair classification the reader may refer to [9].

When ranking items in a certain order, the ratio of protected individuals that appear within a prefix of the ranking must be above a given proportion, in order to satisfy statistical tests of representativeness as described in [34]. The attention received by the items in different positions in the ranking is also not the same: items ranked in first positions are exposed to much more attention than the lower ones. One possible solution to this is to re-rank items after the scores were calculated in order balance opportunity [25].

The problem we tackle here is specifically of having a list to be presented as a recommendation where the items in the first positions have the same or very similar relevance. When it happens, there is a decision to be made of which items are being top-ranked and which are not. One possible solution to this situation was proposed by Biega et al. [5], a mechanism called *amortized fairness*, in which the position index is a proxy for the level of attention an item is exposed, and the output of the prediction algorithm corresponds to the relevance. Accumulated attention across a series of rankings should be proportional to accumulated relevance as indicating long term ranking fairness.

2.1 Amortized fairness

We follow [5] and formalize the problem defining: $u_1, \ldots, u_n$ as a set of subjects to be ranked; $\rho^1, \ldots, \rho^m$ as a sequence of rankings; the position in the ranking as a proxy for for the level of attention (a), and the score given by the model as a proxy for the relevance (r); $r_i^j \in [0 \ldots 1]$ as the normalized relevance score of subject u_i in ranking ρ^j; $a_i^j \in [0 \ldots 1]$ as the normalized attention received by subject u_i in ranking ρ^j. The accumulated attention across subjects is defined as A, and the accumulated relevance across subjects as R.

In the core, we require that ranked subjects receive attention that is proportional to their relevance in a series of rankings, as defined in:

$$\frac{\sum_{l=1}^{m} a_{i1}^l}{\sum_{l=1}^{m} r_{i1}^l} = \frac{\sum_{l=1}^{m} a_{i2}^l}{\sum_{l=1}^{m} r_{i2}^l}, \forall u_{i1}, u_{i2} \tag{1}$$

From this perspective, the unfair position one item appears in a single ranking can be compensated in the next ones when it changes position, and the whole session contemplates long term fairness.

The unfairness can be measured as the distance between attention (A) and relevance (R) distributions. We propose normalizing the calculation by the number of items and the number of rounds in order to have its value independent of how many items and rounds are applied in the experiment:

$$unfairness(\rho^1, ..., \rho^m) = \frac{1}{N}\frac{1}{M}\sum_{i=1}^{n}\left\|\sum_{j=1}^{m} a_i^j - \sum_{j=1}^{m} r_i^j\right\| \tag{2}$$

The act of reducing unfairness imply reduction in the ranking quality. Discounted Cumulative Gain (DCG) is a standard for measuring ranking quality and it gives emphasis on having higher relevance scores at first positions:

$$\text{DCG@k}(r) = \sum_{i=1}^{k} \frac{2^{r(i)} - 1}{log_2(i+1)}. \tag{3}$$

Where k stands for the size of the ranking analyzed. The DCG value is then normalized by the DCG of the perfect ranking order, for obtaining NDCG. The original ranking is ρ, and the $NDCG{-}quality$ is calculated for a new item positioning (ρ^*) as:

$$\text{NDCG-quality@k}(\rho, \rho^*) = \frac{DCG@k(\rho^*)}{DCG@k(\rho)}. \tag{4}$$

When maintaining the original scores in the ranking and changing their positions, $NDCG - quality$ can only be greater then 0 and lower than 1. In the current experiment, however, the score values change each round, and then the metric has a slightly different interpretation compared to the original. A further discussion on this is presented in section 5.

3 FAIRNESS-AWARE VARIATIONAL AUTOENCODER

Autoencoders (AE) are designed to learn its parameters in order to have the output as close as possible to the input, and to force the middle layer as a dense representation of the data in a dimensionality reduction fashion [12].

In the specific case of CF tasks, the set of ratings users gave to each item are the input data propagated through the Multi Layer Perceptron (MLP), so the middle layer will represent a dense version of their preferences [23]. The estimation occurs when decoding the hidden representation back to the original dimension, this time containing probabilities values also for unseen items. The main idea is estimating preference for unrated items based on rated ones, the same way as in matrix factorization approaches [15] but in this case applying *nonlinear* activation functions.

Variational Autoencoders (VAE) were proposed as being capable to capture per-data-point variance when inferring normal distributions in its hidden layer, as illustrated in Figure 1. This provides better overall results in terms of generalization, and a richer representation of the training data.

3.1 Encoder

VAEs are based on the idea of inferring a latent variable space Z by approximating the intractable posterior $p_\theta(z_u|x_u)$. The true posterior is approximated by a tractable fully factorized variational distribution

$$q(z_u) = \mathcal{N}(\boldsymbol{\mu}_u, \text{diag}\{\boldsymbol{\sigma}_u^2\}). \tag{5}$$

The idea is to minimize the Kullback-Leiber divergence between both distributions $KL(q(z_u)||p(z_u|x_u))$ by optimizing parameters $\{\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u^2\}$.

The standard approach introduces a data dependent function parametrized by ϕ with $\mu_\phi(x_u)$ and $\sigma_\phi(x_u)$ as K dimension vectors defined as

$$q_\phi(z_u|x_u) = \mathcal{N}(\mu_\phi(x_u), \text{diag}\{\sigma_\phi^2(x_u)\}). \tag{6}$$

The inference model then outputs the variational parameters of $q_\phi(z_u|x_u)$ having x_u as an input, which approximates the original $p(z_u|x_u)$.

Minimize the divergence $KL(q(z_u)||p(z_u|x_u))$ is equivalent to maximize the lower bound of the log marginal likelihood of the data, also known as Evidence Lower Bound (ELBO), defined for a single user u as:

$$\log p(x_u;\theta) \geq \mathbb{E}_{q_\phi(z_u|x_u)}[\log p_\theta(x_u|z_u)] - KL(q_\phi(z_u|x_u)||p(z_u))) \tag{7}$$

The first term can be interpreted as a reconstruction error while the second as a regularization term forcing the variational posterior $q_\phi(z_u|x_u)$ to be near $p(z_u)$[2].

3.1.1 Reparametrization Trick. It is possible to sample $z_u \sim q_\phi$ and perform gradient ascent to optimize ELBO, however taking gradients through the sample process can be challenging. The *reparametrization trick* [13] proposes sampling from a noise distribution $\epsilon \sim \mathcal{N}(0, \mathbf{I}_K)$ and reparametrize $z_u = \mu_\phi(x_u) + \epsilon \odot \sigma_\phi(x_u)$.

[2]Great part of the mathematics in this section is borrowed from [17]. The original work mentions the parameter β for controlling the strength of regularization which we do not discuss here but incorporate in our implementation.

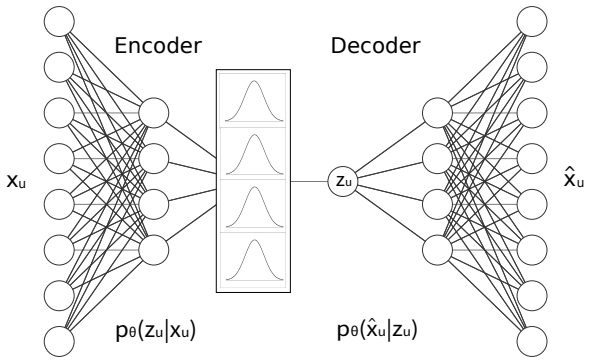

Figure 1: Variational Autoencoder

This way the gradient with respect to ϕ can propagate through the sampled z_u.

3.2 Decoder

We assume $u \in 1, \ldots, U$ as index for users, and $i \in 1, \ldots, I$ as index for items. In the case of implicit feedback the rating matrix is $\mathbf{X} \in \{0,1\}^{U \times I}$. Each row represents the binary collection x_u of items user u interacted with. We define $I_u = \{i \in I | x_{u,i} = 1\}$.

The model considered as a reference is the Multinomial Variarional Autoencoder (MVAE) from [17]. The generative process assumes a Gaussian prior from where latent representation z_u for user u was sampled from. This representation has dimension K, zero mean and identity covariance matrix:

$$z_u \sim \mathcal{N}(0, \mathbf{I}_K). \tag{8}$$

z_u is transformed via a non-linear f_θ and a softmax function to a probability distribution over I items:

$$\pi(z_u) \propto \exp\{f_\theta(z_u)\}. \tag{9}$$

Finally, x_u is assumed to have been drawn from a multinomial distribution with probability $\pi(z_u)$:

$$x_u \sim \text{Mult}(N_u, \pi(z_u)), \tag{10}$$

with $N_u = \sum_i x_{ui}$. The likelihood for x_u is:

$$\log p_\theta(x_u|z_u) = \sum_i x_{ui} \log \pi_i(z_u). \tag{11}$$

Typically, for the case of a user collection x, $z = \mu_\phi(x)$ is calculated and through $\pi(z)$ the probabilities for the whole set are obtained. Unseen items are ranked according to the associated probabilities.

3.3 Enhancing Fairness

We here propose incorporating the noise variable ϵ in the test phase of VAE to enhance fairness in sequential rounds of recommendations. Different noise distributions are expected as generating direct effect in the final ranking, depending on how frequently the latent values vary around the mean inside the interval defined by the variance. To measure this effect, we apply 4 different noise distributions:

Table 1: Data description after filtering: Number of total ratings, users, items, the density of the rating matrix, the quantile value corresponding to the maximum number of items 1/3 and 2/3 of users interacted with, and the amount of users taken in the test phase.

Dataset	#Ratings	#Users	#Items	Density (%)	1/3 Quantile	2/3 Quantile	Heldout Users
Movie Lens	9,990,682	136,677	20,720	0.353	24	63	10,000
Netflix	56,785,778	461,285	17,767	0.693	33	114	40,000
MSD	33,633,450	571,355	41,140	0.143	31	57	50,000

- **Gaussian Noise (σ^2 = 0.5)**: The unfairness measurements are supposed to decrease but not too much, due to the small variance effect introduced by the noise. The same for NDCG. The items should change position in the final ranking closely to the original case when taking only the mean (μ_ϕ) into account.
- **Gaussian Noise (σ^2 = 1.0)**: The unfairness measurements are supposed to decrease significantly, with the noise component having the same variance as the one applied in the training phase. As well as NDCG.
- **Gaussian Noise (σ^2 = 2.0)**: The unfairness measurements are supposed to decrease depending on the dataset, once the variance is greater than the one used for training. The items should change position each round more than in previous cases. NDCG is supposed to decrease as well.
- **Uniform Noise**: Every value in the range defined by the latent variance (σ^2_ϕ) has the same probability in the sampling process. This is proposed as a reference of exploring the effect of variance independent of the mean.

4 EXPERIMENTS

The effect of incorporating randomness in the decoding phase of VAEs is measured in 3 datasets:

MovieLens-20M[3]: Movie ratings collected from 1995 to 2015. The data was converted to binary as in the case of implicit feedback. Users who interacted with less than 5 movies were removed.

Netflix: Movie rating data collected from 1998 to 2005 [2]. The data was also converted to binary and users who interacted with less than 5 movies were also removed.

Million Song Dataset (MSD)[4]: Song listening data [4]. We remove users who listened to less than 200 songs and songs that were listened by less than 20 users.

The filtered data description is summarized in Table 1. The train was conducted with a batch size of 500, the validation batch size of 2000, and the rest of parameters as proposed by [17]. We separate a group of heldout users to ensure the power of generalization, as indicated in Table 1. The f_θ is selected as an one-hidden Multi Layer Perceptron $[I \rightarrow 600 \rightarrow 200 \rightarrow 600 \rightarrow I]$.

After training, $DCG@100$ is calculated once for each test set and stored for posterior normalization. We select 1000 random users among the heldout ones and evaluate the final ranking according to $DCG@100$ and $Unfairness@100$ values for 100 rounds of recommendation[5].

We define 3 classes of users according to the quantity of items they interacted with: (i) *Sparse*: Users who interacted with fewer items. (ii) *Regular*: Users who interacted with between 1/3 and 2/3 of items in the distribution of number of items interactions. (iii) *Dedicated*: Users who interacted with the highest portion of items.

5 RESULTS

The normal distribution with higher variance ($\mathcal{N}(0, 2.0)$) presents the higher compensation of unfairness for all dataset, as shown in Table 2. The sparse users results are also unanimous among all classes of users. In Table 4 the same results are presented but normalized by the original unfairness value, when no noise is incorporated in the test. The greatest compensation was calculated for the regular users from the MSD dataset, when the unfairness was reduced to 23.7% of the original, a reduction of 76.3%.

A customized version of NDCG was proposed by Biega et al. [5] for representing the relation between two DCGs calculated for the same set of scores arranged in different and independent orders. It is called $NDCG - quality$, and should be interpreted differently from the original metric which normalizes DCG with the ideal order of scores. In this work the quality of ranking is also expressed as a comparison between two independent rankings but this time they can have even different score values. We maintain it as NDCG from this point on.

$NDCG@100$ indicates the relation between the DCG for the ranking obtained by applying noise distributions, normalized by the original DCG obtained from regular operation of VAE. Both calculated for the 100 top ranked items. The loss in the ranking quality is small in the case of distribution $\mathcal{N}(0, 0.5)$ applied for dedicated users in all datasets (Table 3). The complementary values are indicate as 1-NDCG in Table 4.

In order to have a general idea about the results, we sum up both relative values (Unfairness and 1-NDCG) for the top 100 items as shown in Table 5. The lowest value indicate the greater combined effect of increasing fairness and maintaining the ranking quality, and it happens in the case of regular users of the MSD dataset.

In Figure 2, it is possible to notice that the highest original unfairness value was calculated for the MSD dataset. These are the only ratings referred to music listening (MovieLens and Netflix were extracted from movie watching activities), and comprehend approximately twice the number of items than the other two. Its density is also the lowest, a third part of the second highest.

A decreasing trend of both unfairness and NDCG is observed in all datasets, and in all groups of users, as the variance of the normal noise distributions increases, as one can see in Figures 3, 4 and 5. When uniform noise is applied it increases again, and the results become similar to $\mathcal{N}(0, 0.5)$.

[3]https://grouplens.org/datasets/movielens/
[4]http://millionsongdataset.com/tasteprofile/
[5]The code for reproducing the experiment is available at https://github.com/rcaborges/variational-fairness.

Table 2: Unfairness@100 for 100 rounds of recommendation

Dataset	Original	$\mathcal{N}(0, 0.5)$	$\mathcal{N}(0, 1.0)$	$\mathcal{N}(0, 2.0)$	Uniform
Movie Lens	0.185 (0.054)	0.134 (0.048)	0.094 (0.037)	0.061 (0.019)	0.126 (0.047)
Movie Lens (Sparse)	0.181 (0.060)	0.120 (0.045)	0.078 (0.025)	**0.052 (0.007)**	0.111 (0.041)
Movie Lens (Regular)	0.174 (0.051)	0.123 (0.043)	0.085 (0.028)	0.058 (0.009)	0.116 (0.041)
Movie Lens (Dedicated)	0.194 (0.046)	0.154 (0.046)	0.116 (0.041)	0.074 (0.025)	0.148 (0.046)
Netflix	0.194 (0.045)	0.126 (0.043)	0.082 (0.031)	0.060 (0.014)	0.117 (0.041)
Netflix (Sparse)	0.197 (0.047)	0.111 (0.036)	0.067 (0.018)	**0.054 (0.008)**	0.101 (0.033)
Netflix (Regular)	0.188 (0.043)	0.120 (0.036)	0.077 (0.021)	0.058 (0.007)	0.111 (0.034)
Netflix (Dedicated)	0.200 (0.048)	0.151 (0.048)	0.108 (0.041)	0.069 (0.022)	0.143 (0.048)
MSD	0.242 (0.052)	0.171 (0.052)	0.112 (0.045)	0.061 (0.024)	0.160 (0.051)
MSD (Sparse)	0.248 (0.052)	0.163 (0.052)	0.099 (0.043)	**0.054 (0.020)**	0.151 (0.051)
MSD (Regular)	0.245 (0.050)	0.170 (0.050)	0.109 (0.042)	0.058 (0.020)	0.159 (0.049)
MSD (Dedicated)	0.241 (0.051)	0.185 (0.051)	0.131 (0.046)	0.072 (0.028)	0.175 (0.051)

Table 3: NDCG@100 for 100 rounds of recommendation

Dataset	Original	$\mathcal{N}(0, 0.5)$	$\mathcal{N}(0, 1.0)$	$\mathcal{N}(0, 2.0)$	Uniform
Movie Lens	1.000 (0.000)	0.996 (0.004)	0.984 (0.012)	0.944 (0.040)	0.994 (0.004)
Movie Lens (Sparse)	1.000 (0.000)	0.993 (0.004)	0.972 (0.011)	0.903 (0.033)	0.990 (0.005)
Movie Lens (Regular)	1.000 (0.000)	0.996 (0.002)	0.985 (0.008)	0.945 (0.023)	0.995 (0.023)
Movie Lens (Dedicated)	1.000 (0.000)	**0.999 (0.001)**	0.995 (0.003)	0.982 (0.011)	0.998 (0.001)
Netflix	1.000 (0.000)	0.996 (0.004)	0.987 (0.012)	0.955 (0.038)	0.995 (0.004)
Netflix (Sparse)	1.000 (0.000)	0.993 (0.003)	0.973 (0.010)	0.912 (0.023)	0.990 (0.004)
Netflix (Regular)	1.000 (0.000)	0.997 (0.002)	0.990 (0.006)	0.964 (0.017)	0.996 (0.002)
Netflix (Dedicated)	1.000 (0.000)	**0.999 (0.001)**	0.997 (0.003)	0.990 (0.008)	0.999 (0.001)
MSD	1.000 (0.000)	0.996 (0.002)	0.985 (0.007)	0.948 (0.022)	0.995 (0.003)
MSD (Sparse)	1.000 (0.000)	0.994 (0.002)	0.979 (0.006)	0.927 (0.017)	0.993 (0.002)
MSD (Regular)	1.000 (0.000)	0.996 (0.002)	0.985 (0.005)	0.947 (0.014)	0.995 (0.002)
MSD (Dedicated)	1.000 (0.000)	**0.998 (0.001)**	0.992 (0.004)	0.971 (0.012)	0.997 (0.002)

Table 4: Relative Unfairness@100 and NDCG@100

	Unfairness@100 (%)				1-NDCG@100			
Dataset	$\mathcal{N}(0, 0.5)$	$\mathcal{N}(0, 1.0)$	$\mathcal{N}(0, 2.0)$	Uniform	$\mathcal{N}(0, 0.5)$	$\mathcal{N}(0, 1.0)$	$\mathcal{N}(0, 2.0)$	Uniform
Movie Lens	0.724	0.508	0.330	0.681	0.004	0.016	0.056	0.006
Movie Lens (Sparse)	0.663	0.431	**0.287**	0.613	0.007	0.028	0.097	0.010
Movie Lens (Regular)	0.707	0.489	0.333	0.667	0.004	0.015	0.055	0.005
Movie Lens (Dedicated)	0.794	0.598	0.381	0.763	**0.001**	0.005	0.018	0.002
Netflix	0.649	0.423	0.309	0.603	0.004	0.013	0.045	0.005
Netflix (Sparse)	0.563	0.340	**0.274**	0.513	0.007	0.027	0.088	0.010
Netflix (Regular)	0.638	0.410	0.309	0.590	0.003	0.010	0.036	0.004
Netflix (Dedicated)	0.755	0.540	0.345	0.715	**0.001**	0.003	0.010	**0.001**
MSD	0.707	0.463	0.252	0.661	0.004	0.015	0.052	0.005
MSD (Sparse)	0.657	0.399	**0.218**	0.609	0.006	0.021	0.073	0.007
MSD (Regular)	0.694	0.445	0.237	0.649	0.004	0.015	0.053	0.005
MSD (Dedicated)	0.768	0.544	0.299	0.726	**0.002**	0.008	0.029	0.003

Table 5: Comparing noise distributions (Summation of relative Unfairness@100 and 1-NDCG@100)

Dataset	$\mathcal{N}(0, 0.5)$	$\mathcal{N}(0, 1.0)$	$\mathcal{N}(0, 2.0)$	Uniform
Movie Lens	0.728	0.524	0.386	0.687
Movie Lens (Sparse)	0.670	0.459	**0.384**	0.623
Movie Lens (Regular)	0.711	0.504	0.388	0.672
Movie Lens (Dedicated)	0.795	0.603	0.399	0.765
Netflix	0.653	0.436	0.354	0.608
Netflix (Sparse)	0.570	0.367	0.362	0.523
Netflix (Regular)	0.641	0.420	**0.345**	0.594
Netflix (Dedicated)	0.756	0.543	0.355	0.716
MSD	0.711	0.478	0.304	0.666
MSD (Sparse)	0.663	0.420	0.291	0.616
MSD (Regular)	0.698	0.460	**0.290**	0.654
MSD (Dedicated)	0.770	0.552	0.328	0.729

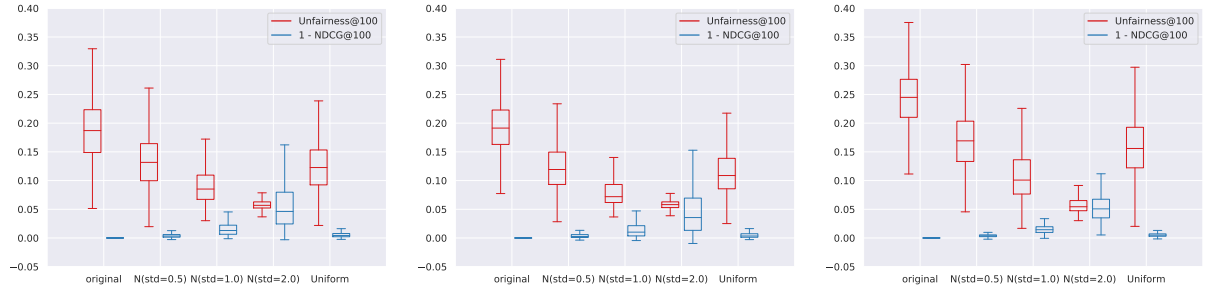

Figure 2: Unfairness and NDCG trade-offs before (original) and after applying four different noise distributions ($\mathcal{N}(0, 0.5)$,$\mathcal{N}(0, 1.0)$,$\mathcal{N}(0, 2.0)$ and uniform) for 1k random users in MovieLens, Netflix and MSD test sets respectively.

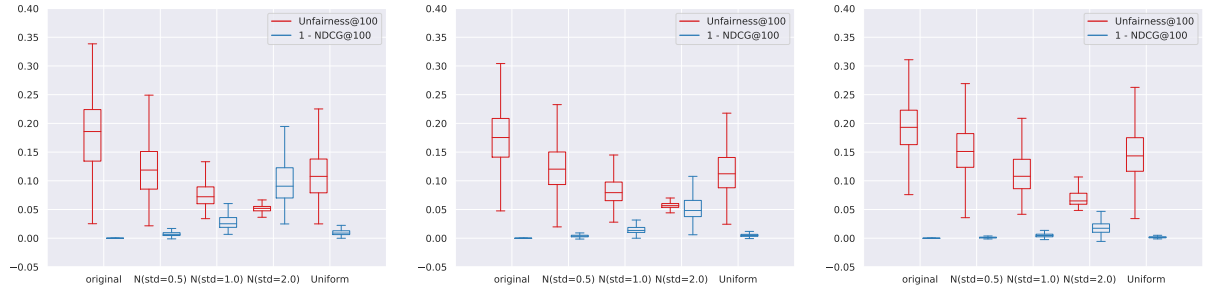

Figure 3: From left to right: sparse, regular and dedicated 1k random sampled users from the MovieLens test set.

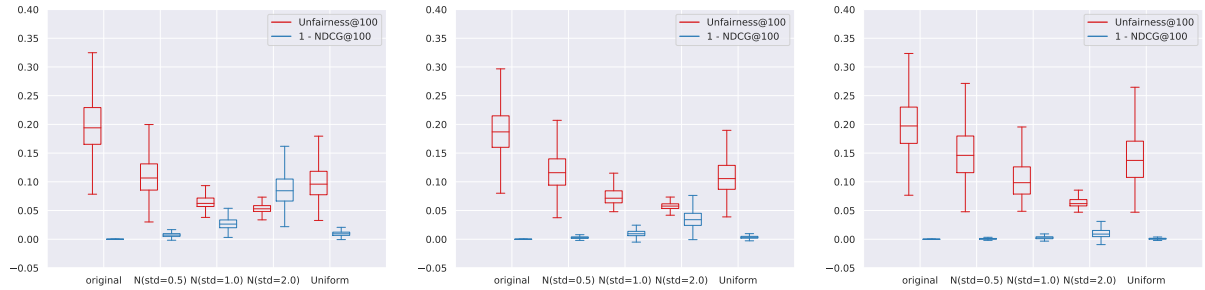

Figure 4: From left to right: sparse, regular and dedicated 1k random sampled users from the Netflix test set.

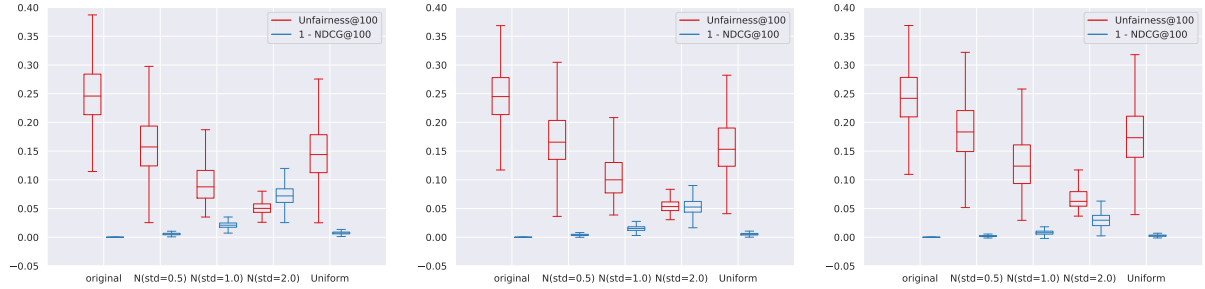

Figure 5: From left to right: sparse, regular and dedicated 1k random sampled users from the MSD test set.

6 RELATED WORK

To facilitate users in their selection process, recommender systems provide suggestions on data items, which might be interesting for the respective users. Nowadays, recommendations have more broad applications, beyond products, like links (friends) recommendations [33], social-based recommendations [28], health-related recommendations [29], open source software recommendations [16], diverse venue recommendations [10], or even recommendations for evolution measures [27]. For achieving efficiency, there are approaches that build user models for computing recommendations. For example, [20] applies subspace clustering to organize users into clusters and employs these clusters, instead of a linear scan of the database, for making predictions.

6.1 Fairness in Recommender Systems

Fairness has emerged as an important category of analysis for machine learning systems in many application areas. By fairness, we typically mean lack of bias. It is not correct to assume that insights achieved via computations on data are unbiased simply because data was collected automatically or processing was performed algorithmically. Bias may come from the algorithm, reflecting, for example, preferences of its designer, or from the actual data, for example, if a survey contains biased questions. In extending the concept of fairness to recommender systems, there is an essential tension between the goals of fairness and those of personalization.

Generally speaking, there are contexts in which equity across recommendation outcomes is a desirable goal. It is also the case that in some applications, fairness may be a multi-sided concept [6, 7], in which the impacts on multiple groups of individuals must be considered.

In recommender systems, fairness can have multiple viewpoints: like fairness for the recommended items [26], for the users [32], for group of users [19, 24] and for providers [18]. Especially, when considering group-based fairness, we can distinguish between demographic parity [30], calibration-based fairness [26] and accuracy-based fairness [3]. To the best of our knowledge, this is the first work focusing on achieving long term fairness, that is, on increasing fairness in multiple rounds of recommendations.

6.2 Multi-rounds Recommendations

Recommender algorithms designed for both single users and groups have been extensively studied. Typically, this category of algorithms focus mainly on one interaction of the user/group with the system. Instead, the case of a multi-round recommendation approach has received significantly less attention. [1] exploits users preferences to suggest sequences of songs. The method is built iteratively. First, it obtains a ranked list of songs, after excluding the songs of recently played artists. Next, the songs with the best preference scores are re-ranked: from a newly produced ranked list, the method removes the songs that at least one user gave rating below a threshold. Finally, users can adjust their ratings through a feedback phase. In a different domain, [21] suggests a sequence of artworks for a group of visitors in a museum. Focusing on maximizing the satisfaction of the proposed recommendations, while taking into consideration both time constraints and the artworks locations in the museum. In our work, we target at algorithms that directly enhance fairness, that has not been considered in previous approaches.

7 CONCLUSIONS

We demonstrate the effect of reducing ranking unfairness in collaborative filtering by introducing a stochastic component to the sampling phase of a trained VAE model. The results point to a positive trade-off between promoting equal treatment among relevance-equivalent items despite a small reduction in the ranking quality in a sequence of recommendations, specially for the case of applying a normal distributed noise with high variance.

By promoting random perturbations in the latent representation of VAEs, we get as its output not exactly approximations of the input, as in the case of classic autoencoders, but values generated according to normal distributions estimated during the training phase. The score given by each item changes at each round of recommendation, and the amount it changes depend on the variance of the noise distribution. That is the reason why the solution proposed here act in the long term as avoiding positional bias in homogeneous scores.

The experiments were conducted with a simple neural network architecture, which turn the solution reproducible in online platforms. Instead of a post process procedure, we propose a solution that is already incorporated in the normal operation of the model and demands no extra computation.

Varying the latent space for obtaining different rankings should also be useful for promoting diversity in the context of balancing exploration and exploitation in recommendations [31]. This possibility will be addressed in future work.

ACKNOWLEDGMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 88881.189985/2018-01.

REFERENCES

[1] Claudio Baccigalupo and Enric Plaza. 2006. Case-Based Sequential Ordering of Songs for Playlist Recommendation. In *ECCBR*. 286–300.
[2] James Bennett, Stan Lanning, and Netflix Netflix. 2007. The Netflix Prize. In *In KDD Cup and Workshop in conjunction with KDD*.
[3] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael J. Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2017. A Convex Framework for Fair Regression. *CoRR* abs/1706.02409 (2017). http://arxiv.org/abs/1706.02409
[4] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In *ISMIR*.
[5] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *ACM SIGIR*. 405–414.
[6] Robin Burke. 2017. Multisided Fairness for Recommendation. *CoRR* abs/1707.00093 (2017). arXiv:1707.00093 http://arxiv.org/abs/1707.00093
[7] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *FAT*. 202–214.
[8] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness Through Awareness. In *ITCS*. 214–226.
[9] Pratik Gajane. 2017. On formalizing fairness in prediction with machine learning. *CoRR* abs/1710.03184 (2017). arXiv:1710.03184 http://arxiv.org/abs/1710.03184
[10] Xiaoyu Ge, Panos K. Chrysanthis, and Konstantinos Pelechrinis. 2016. MPG: Not So Random Exploration of a City. In *MDM*. 72–81.
[11] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NIPS*. 3323–3331.
[12] G. E. Hinton and R. R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 5786 (2006), 504–507. https://doi.org/10.1126/science.1127647
[13] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.

[14] Diederik P. Kingma and Max Welling. 2019. An Introduction to Variational Autoencoders. *CoRR* abs/1906.02691 (2019). arXiv:1906.02691 http://arxiv.org/abs/1906.02691

[15] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37. https://doi.org/10.1109/MC.2009.263

[16] Miika Koskela, Inka Simola, and Kostas Stefanidis. 2018. Open Source Software Recommendations Using Github. In *TPDL*. 279–285.

[17] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*. 689–698.

[18] L. Machado and K. Stefanidis. 2019. Fair Team Recommendations for Multidisciplinary Projects. In *ACM Web Intelligence*.

[19] Eirini Ntoutsi, Kostas Stefanidis, Kjetil Nørvåg, and Hans-Peter Kriegel. 2012. Fast Group Recommendations by Applying User Clustering. In *ER*. 126–140.

[20] Eirini Ntoutsi, Kostas Stefanidis, Katharina Rausch, and Hans-Peter Kriegel. 2014. "Strength Lies in Differences": Diversifying Friends for Recommendations through Subspace Clustering. In *CIKM*. 729–738.

[21] Silvia Rossi, Francesco Barile, Clemente Galdi, and Luca Russo. 2017. Recommendation in museums: paths, sequences, and group satisfaction maximization. *Multimedia Tools Appl.* 76, 24 (2017), 26031–26055.

[22] Noveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. 2019. Sequential Variational Autoencoders for Collaborative Filtering. In *WSDM*. 600–608.

[23] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *WWW*. 111–112.

[24] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2017. Fairness in Package-to-Group Recommendations. In *WWW*. 371–379.

[25] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *KDD*. 2219–2228.

[26] Harald Steck. 2018. Calibrated recommendations. In *ACM RecSys*. 154–162.

[27] Kostas Stefanidis, Haridimos Kondylakis, and Georgia Troullinou. 2017. On Recommending Evolution Measures: A Human-Aware Approach. In *ICDE*. 1579–1581.

[28] Kostas Stefanidis, Eirini Ntoutsi, Haridimos Kondylakis, and Yannis Velegrakis. 2018. Social-Based Collaborative Filtering. In *Encyclopedia of Social Network Analysis and Mining, 2nd Edition*.

[29] Maria Stratigi, Haridimos Kondylakis, and Kostas Stefanidis. 2017. Fairness in Group Recommendations in the Health Domain. In *ICDE*. 1481–1488.

[30] Virginia Tsintzou, Evaggelia Pitoura, and Panayiotis Tsaparas. 2018. Bias Disparity in Recommendation Systems. *CoRR* abs/1811.01461 (2018). http://arxiv.org/abs/1811.01461

[31] Zhe Xing, Xinxi Wang, and Ye Wang. 2014. Enhancing Collaborative Filtering Music Recommendation by Balancing Exploration and Exploitation. In *ISMIR*. 445–450.

[32] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*. 2921–2930.

[33] Zhijun Yin, Manish Gupta, Tim Weninger, and Jiawei Han. 2010. LINKREC: a unified framework for link recommendation with user attributes and graph structure. In *WWW*. 1211–1212.

[34] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *CIKM*. 1569–1578.