Unbiasing Collaborative Filtering for Popularity-Aware Recommendation

(Discussion Paper)

Luciano Caroprese[1], Giuseppe Manco[1], Marco Minici[1], Francesco Sergio Pisani[1] and Ettore Ritacco[1]

[1]ICAR-CNR, Via Bucci, 8-9c, Rende (Italy)

Abstract

We analyze the behavior of recommender systems relative to the popularity of the items to recommend. Our findings show that most popular ranking-based recommenders are biased towards popular items, thus affecting the quality of recommendation. Based on these observations, we propose a new deep learning architecture with an improved learning strategy that significantly improves the performance of such recommenders on low-popular items. The proposed technique is based on two main aspects: resampling of negatives and ensembling of multiple instances of the algorithm. Experimental results on traditional benchmark datasets show that the proposed approach substantially improves the recommendation ability by balancing accurate contributions almost independently from the popularity of the items to recommend.

Keywords

Recommender Systems, Collaborative Filtering, Deep Learning, Big Data

1. Introduction

The massive amount of information available to users in the form of digital catalogs and online services allows users to access and consume online content, but at the same time it poses a choice paradox. Purchasing items on e-commerce sites, selecting movies on streaming platforms or connecting with other peers on social networks implies making choices among a huge amount of elements. Recommender Systems play a crucial role within this context since they provide users with a better experience through the recommendation of new content and data that users are likely to appreciate.

Recommendations can have a disruptive impact: if on one side, they assist users in their choices, on the other side they can influence the choices themselves. Indeed, Recommender systems can favor particular categories of items or particular brands over others, thus introducing a *bias* within the catalog [1]. A typical bias is intrinsic to the nature of recommendation. In systems involving interaction between significant amounts of users and items, we observe the presence of very few very popular items and many less popular others. This distribution follows the so-called *80-20 rule* that refers to the fact that 80% of users express preferences for only 20%

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy

luciano.caroprese@icar.cnr.it (L. Caroprese); giuseppe.manco@icar.cnr.it (G. Manco); marco.minici@icar.cnr.it (M. Minici); francescosergio.pisani@icar.cnr.it (F. S. Pisani); ettore.ritacco@icar.cnr.it (E. Ritacco)

of the available items. In practice, user preferences follow a *long tail* distribution. Recommender Systems based on collaborative filtering [2], who tend to characterize user preferences from interactions, are affected by a relevant problem: they tend to suggest very popular items and neglect less popular ones. This way, there is a reinforced effect because the most popular items will become more and more popular than the less known ones. This phenomenon is called *popularity bias*.

The quality of the recommendations is inevitably affected by the popularity bias, since most recommendations are prone to contain objects that the user will consider trivial and well known. By contrast, the capability of recommending items belonging to the long tail (i.e., less popular) can disclose new perspectives: niche items, little-known objects, hidden gems that satisfy the user's preferences can greatly improve user engagement, provided that the recommended items are consistent with user's taste.

The current literature has extensively studied the problems of fairness and bias [3, 4, 5] within recommender systems, with particular emphasis to popularity bias [6, 7, 8]. Notably, in [9], it is shown that unfair recommendations are concentrated on groups of users interested in long-tail and less popular items. In this paper we focus our attention on recommender sytems based on ranking [10, 11], which are extremely flexible and as a consequence particularly interesting to unbias. Typical approaches [12, 13] consider techniques to increase the representation of less popular items, by either post-processing techniques or constraining the recommendation score. Despite such recent efforts, unbiasing the recommendation from popular items is still an open problem.

In this paper we devise a ranking-based recommender system for implicit feedback (RVAE), based on a variational autoencoder architecture. The proposed model is a substantial extension of the MVAE model proposed in [14] and in fact it inherits its accuracy and computational efficiency. We analyze the behavior of RVAE and show that the model is characterized by the popularity bias problem. We then propose an experimental study of specific techniques to overcome it. The proposed technique is based on two main concepts: resampling/reweighting items and ensembling of multiple instances of the algorithm. Our experiments show that these simple strategies allow to unbias the algorithm and hence provide more effective recommendations.

2. Basic Framework

We start by setting the notation that we shall use throughout the paper. In the context of collaborative filtering, $u \in U = \{1, \dots, M\}$ indexes a user and $i, j \in I = \{1, \dots, N\}$ index items for which the user can express a preference. We model implicit feedback, thus assuming a preference matrix $\mathbf{X} \in \{0, 1\}^{M \times N}$, so that $x_{u,i} = 1$ whenever user u expressed a preference for item i, and $x_{u,i} = 0$ otherwise. Also, $\mathbf{x}_u$ is the (binary) row indexed at u, representing all the item preferences for user u. Given $\mathbf{x}_u$, we define $I_u = \{i \in I | x_{u,i} = 1\}$ (with $N_u = |I_u|$). The preference matrix induces a natural ordering between items: $i \prec_u j$ has the meaning that u prefers i to j, i.e. $x_{u,i} > x_{u,j}$ in the rating matrix. Our objective is to devise a model for such an ordering.

Preference Modeling. We consider a general framework where preferences are modeled as the effect of latent factors ultimately characterizing users and/or items. We shall consider two

basic instantiations of this general idea, and will provide a unified framework.

The first situation we consider is the Multinomial Variational Autoencoder (MVAE) framework proposed in [14]. Within this framework, for a given user u the related $\mathbf{x}_u$ is modeled as the effect of a multinomial distribution governed by a prior $\mathbf{z}$, i.e.

$$\begin{aligned} \mathbf{x}_u &\sim \mathit{Discrete}\left(\pi(\mathbf{z})\right) \\ \pi(\mathbf{z}) &\propto \exp\left\{f_\phi(\mathbf{z})\right\} \end{aligned}$$

Here, $f_\phi(\cdot)$ represents a neural network parameterized by ϕ. The latent variable $\mathbf{z}$ is modeled by a prior $P(\mathbf{z})$ (typically a gaussian distribution). Thus, the probability of preferences for a given user can be expressed as

$$P(\mathbf{x}_u) = \int P(\mathbf{x}_u|\mathbf{z})P(\mathbf{z})\ \mathrm{d}\mathbf{z}$$

Due to the intractability of the above integral, [15] devise a variational approach based on a proposal $Q(\mathbf{z}|\mathbf{x}_u)$ that approximates the posterior distribution. Again, Q is modeled as a gaussian distribution

$$Q(\mathbf{z}|\mathbf{x}_u) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_u, \boldsymbol{\sigma}_u),$$

where $\boldsymbol{\mu}_u, \boldsymbol{\sigma}_u = g_\theta(\mathbf{x}_u)$ and g_θ is a neural network parameterized by θ. By exploiting the inequality

$$\log P(\mathbf{x}_u) \geq \mathbb{E}_{\mathbf{z}\sim Q(\cdot|\mathbf{x}_u)}\left[\log P(\mathbf{x}_u|\mathbf{z}) - P(\mathbf{z})\right]$$

we can finally learn the ϕ, θ parameters by optimizing the loss

$$\ell_{MVAE}(\phi, \theta) = \sum_u \left\{\mathbb{E}_{\mathbf{z}\sim Q_\theta(\cdot|\mathbf{x}_u)}\left[\log P_\phi(\mathbf{x}_u|\mathbf{z})\right] - \mathbb{KL}[Q_\theta(\mathbf{z}|\mathbf{x}_u)\|P(\mathbf{z})]\right\}$$

The overall framework is based hence on regularized encoder-decoder scheme, where $Q_\theta(\mathbf{z}|\mathbf{x}_u)$ represents the encoder, $P_\theta(\mathbf{x}_u|\mathbf{z})$ represents the decoder and the term $\mathbb{E}_{\mathbf{z}\sim Q_\theta(\cdot|\mathbf{x}_u)}\left[P(\mathbf{z})\right]$ acts as a regularizer. In the training phase, for each u a latent variable $\mathbf{z} \sim Q_\theta(\cdot|\mathbf{x}_u)$ is devised. Next, $\mathbf{z}$ is exploited to devise the probability $P_\phi(\mathbf{x}_u|\mathbf{z})$. Users with low probability are penalized within the loss and the network parameters can be updated accordingly.

Prediction for new items is accomplished by resorting to the learned functions P_ϕ and Q_θ: given a (partial) user history $\mathbf{x}_u$, we compute $\mathbf{z} = \boldsymbol{\mu}_u$ and then devise the probabilities for the whole item set through $\pi(\mathbf{z})$. Unseen items can then be ranked according to their associated probabilities.

The second formulation we consider is inspired by the Bayesian Personalized Ranking (BPR) model introduced in [10]. The idea underlying this model is that a preference $i \prec_u j$ can be directly explained as closeness in a latent space where both items and users can be mapped. Mathematically this can be devised by computing a factorization rank $\mathbf{p}_u^T\mathbf{q}_i$ for each pair (u, i), and modeling preferences by means of a Bernoulli process:

$$\begin{aligned} i \prec_u j &\sim \mathit{Bernoulli}(p) \\ p &= \sigma\left(\mathbf{p}_u^T(\mathbf{q}_i - \mathbf{q}_j)\right) \end{aligned}$$

where $\sigma(a) = (1 + e^{-a})^{-1}$ is the logistic function. The optimal embeddings $\mathbf{P}$ and $\mathbf{Q}$ can hence be obtained by opimizing the loss

$$\ell_{BPR}(\mathbf{P}, \mathbf{Q}) \approx \sum_{u} \sum_{\substack{i,j \\ i \prec_u j}} \log \sigma\left(\mathbf{p}_u^T(\mathbf{q}_i - \mathbf{q}_j)\right)$$

We combine the two frameworks by adapting the BPR loss to the MVAE model. In particular, instead of modeling $P(\mathbf{x}_u|\mathbf{z})$, we directly model $P(i \prec_u j|\mathbf{z})$ within a similar variational framework. In short, the current preferences are encoded within a latent variable $\mathbf{z}$ that is further exploited to decode all ranks:

$$\begin{aligned} i \prec_u j &\sim Bernoulli(p_{i,j}) \\ p_{i,j} &= \sigma(\zeta_i - \zeta_j) \\ \boldsymbol{\zeta} &= f_\phi(\mathbf{z}) \end{aligned}$$

Here, $\boldsymbol{\zeta}$ represents the output of a neural network parameterized by ϕ. For a given item i, the value ζ_i represents then the associated rank which can be used sort all preferences. The model can be obtained by optimizing the loss:

$$\ell_{RVAE}(\phi, \theta) = \sum_{u} \left\{ \mathbb{E}_{\mathbf{z} \sim Q_\theta(\cdot|\mathbf{x}_u)} \left[\sum_{\substack{i,j \\ i \prec_u j}} \log P_\phi(i \prec_u j|\mathbf{z}) \right] - \mathbb{KL}[Q_\theta(\mathbf{z}|\mathbf{x}_u) \| P(\mathbf{z})] \right\} \tag{1}$$

We call this model *Ranking Variational Autoencoder* (RVAE).

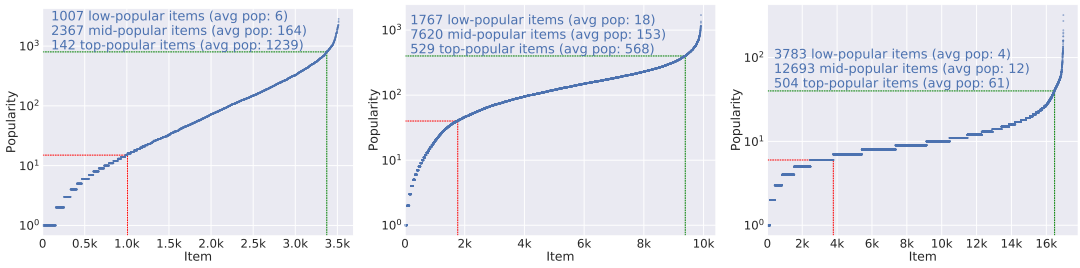
Learning by Negative Sampling. In the above formulation there are some details worth further discussion. When learning the RVAE model, optimizing the likelihood requires that all pairs of items are considered within Eq. (1). This is unrealistic with large item bases, and it is usually customary to only consider a subset $\mathscr{P}_u \subset \{(i,j)|i,j \in I; i \prec_u j\}$. The sampling of $\mathscr{D}_u$ is critical for determining the behavior of any predictive model; the most used approach in literature is to uniformly sample, for each user u and item i (called positive item), a fixed number of items $\{j_1, \dots j_n\} \subset I - I_u$ with the underlying assumption that $\forall k : i \prec_u j_k$. Thus, Eq. (1) can be rewritten as:

$$\ell_{RVAE}(\phi, \theta) = \sum_{u} \left\{ \mathbb{E}_{\mathbf{z} \sim Q_\theta(\cdot|\mathbf{x}_u)} \left[\sum_{(i,j) \in \mathscr{D}_u} \log P_\phi(i \prec_u j|\mathbf{z}) \right] - \mathbb{KL}[Q_\theta(\mathbf{z}|\mathbf{x}_u) \| P(\mathbf{z})] \right\} \tag{2}$$

This will be the basis loss upon which we develop our study next.

3. The Impact of Popularity

We start our analysis on the following popular benchmark datasets: i) **Movielens**, a time series dataset containing user-item ratings pairs along with the corresponding timestamp; ii) **Pinterest**, based on the social media that allows users to save or pin an image (item) to their board. The dataset denotes as 1 the pinned images for each user; iii) **CiteUlike**, a dataset obtained from the homonymous service which provides a digital catalog to save and share



(a) ML1M (b) Pinterest (c) CiteUlike

Figure 1: Item popularity distributions.

academic papers. Within fig. 1 we plot all the items within the datasets, by increasing popularity. For each dataset we identify three classes: low, mid and high popular items.

We then study the behavior of the RVAE model with respect to the popularity classes defined. To do so, we adopt the following protocol. For each dataset, 70% of users are randomly sampled with all user's items. Each such user is associated with $\mathbf{x}_u$ and a set $\mathscr{D}_u$ of positive/negative item pairs. In particular, we consider all positive items within $\mathbf{x}_u$, and for each positive item i we sample $n = 4$ negative items. The remaining 30% users are uniformly split into validation and test. In particular, for each user u we consider a random subset $P_u \subset I_u$ representing the 30% of the positive items, and N_u represents a subset of 100 negative items sampled from $I - I_u$. The vector $\mathbf{x}_u$ is masked to remove all elements in P_u. We then feed the masked $\mathbf{x}_u$ to obtain the score vector $\boldsymbol{\zeta}_u$. Now, for a given cutoff value c, let us consider the $c-1$ negative items for u for which RVAE gives the highest score, and, among them the item j having the minimum score. There is an hit with cutoff c, for the user u and the item $i \in P_u$, if $\zeta_{u,i} \geq \zeta_{u,j}$. Let H_u^c the number of hits for the user u with cutoff c. We define the *Hit-Rate at c* on T as $HR@c = \frac{\sum_{u \in T} H_u^c}{\sum_{u \in U} |P_u|}$. We can trivially specialize this definition for items within the low, medium and high popularity, by considering only items in P_u that belong to the specific class. The results of the evaluation are summarized in table 1a. We can see that the model suffers mainly on low-popular items. As a matter of fact, the overexposure of popular items is predominant and the model learn to predict essentially those items. The fact is that popular items are easy to predict. However, it is in the mid and especially on low popular items that the most interesting predictions can take place: niche items are difficult to discover by an end user and hence their accurate suggestions can greatly improve user engagement. The research question is hence: how can we boost the model to improve the performance on low-popular items?

4. Unbiased Recommendation

In a simple experiment, we retrain the model by only considering pairs $(i, j) \in \mathscr{D}_u$ such that i is in the low popular class. We call this model $RVAE^L$. Compared to the results in table 1a, the results for this restricted model (shown in table 1b) show that if attention is placed on low popular items, their response on prediction accuracy can be improved. Similar results can be observed for the mid popular items. Thus, in order to unbias the model we need to rebalance

Dataset	*HR*@1				*HR*@5				*HR*@10			
	Global	Low	Med	High	Global	Low	Med	High	Global	Low	Med	High
Movielens	0.2510	**0.00**	0.16	0.43	0.5853	**0.01**	0.47	0.83	0.7443	**0.05**	0.65	0.94
Pinterest	0.2731	**0.13**	0.23	0.46	0.6992	**0.46**	0.66	0.86	0.8764	**0.65**	0.86	0.95
CiteULike	0.2875	**0.07**	0.22	0.67	0.6021	**0.29**	0.57	0.90	0.7638	**0.48**	0.75	0.95

(a) RVAE

Dataset	*HR*@1		*HR*@5		*HR*@10	
	Global	Low	Global	Low	Global	Low
Movielens	0.0012	**0.15**	0.0042	**0.50**	0.0063	**0.74**
Pinterest	0.0089	**0.43**	0.0166	**0.81**	0.0192	**0.93**
CiteULike	0.0254	**0.32**	0.0561	**0.71**	0.0703	**0.89**

(b) $RVAE^L$

Table 1
Predictive accuracy.

the contribution on low (and mid) popular items with regards to the high popular ones. To achieve this goal, we study three different strategies.

The first strategy consists in weighting, for each pair $(i, j) \in \mathscr{D}_u$, the contribution to the loss with a factor inversely proportional to the popularity of the item i:

$$w_i = \frac{\gamma\left(1+e^{\alpha(f_i-\beta-1)}\right)^{-1}+1}{\gamma+1}$$

Here, f_i represents the number of occurrences of i, and α, β, γ the parameters representing the steep, center and scale of the decay of high popular items. We experiment with $\alpha = 0.01$, β representing the average frequency of mid-popular items and $\gamma = 100$ and call this variant $RVAE^W$. The above strategy has the advantage of reweighing the contributions of low and mid popular items with respect to high popular ones: the ratio between the most popular and the lowest popular is approximately $1/\gamma$. However, this weighting scheme has a main disadvantage. The weight is relative to a positive item i, but it is associated with pairs $(i, j) \in P_u$. That is, besides weighting the contribution of i, this scheme also overexposes (or dually underexposes) the contribution of the negative item j. To avoid this, an alternative strategy consists in changing the sampling scheme that produces $\mathscr{D}_u$. In practice, rather than uniformly sampling, for each positive item, a fixed number n of negative items, we can apply an inversely stratified sampling where n_i negatives are sampled, with n_i being inversely proportional to the popularity of the item: $n_i = n \cdot \left\lceil \frac{\max(\mathbf{f})}{f_i} \right\rceil$.

The ratio with the above formula is to provide the same visibility to each positive item in the loss function. Thus, the most popular item will be associated with exactly n pairs. By contrast, low and mid popular items will be overexposed in the comparison. We call this variant $RVAE^S$.

The third strategy consists in combining the baseline *RVAE* model with $RVAE^L$. For a user u, let $\boldsymbol{\zeta}^B$ the the score vector produced by *RVAE*, and $\boldsymbol{\zeta}^L$ the score produced by $RVAE^L$. We define $RVAE^E$ and the model that produces the score $\boldsymbol{\zeta}^E$ defined as $\boldsymbol{\zeta}^E = Softmax(\boldsymbol{\zeta}^B) + \delta\mathbf{m} \cdot Softmax(\boldsymbol{\zeta}^L)$.

The vector $\mathbf{m}$ masks all items but the low popular. The scores are normalized (via the *Softmax* function) to make the two models comparable. Finally, δ is a weight aimed at tuning the boost for the low popular items, as devised by $RVAE^L$. We experimentally found an optimal tuning with $\delta = 0.4$.

		HR@1				*HR*@5				*HR*@10			
		Global	Low	Med	High	Global	Low	Med	High	Global	Low	Med	High
Movielens	*RVAE*	**0.2510**	0.00	**0.16**	**0.43**	**0.5853**	0.01	**0.47**	**0.83**	**0.7443**	0.05	**0.65**	**0.94**
	$RVAE^W$	0.0854	0.01	0.13	0.01	0.2958	0.07	0.40	0.10	0.4661	0.11	0.59	0.23
	$RVAE^S$	0.1245	**0.06**	0.12	0.13	0.3875	0.16	0.38	0.40	0.5825	0.28	0.57	0.61
	$RVAE^E$	0.2466	0.05	**0.16**	0.42	0.5506	**0.22**	0.43	0.80	0.6965	**0.37**	0.59	0.91
Pinterest	*RVAE*	**0.2731**	0.13	**0.23**	**0.46**	**0.6992**	0.46	**0.66**	**0.86**	**0.8764**	0.65	**0.86**	**0.95**
	$RVAE^W$	0.2196	**0.29**	**0.23**	0.18	0.6544	**0.68**	**0.66**	0.64	0.8592	**0.80**	**0.86**	0.87
	$RVAE^S$	0.2482	0.11	**0.23**	0.35	0.6770	0.45	0.65	0.80	0.8664	0.64	**0.86**	0.93
	$RVAE^E$	0.2726	0.26	0.22	**0.46**	0.6954	0.63	0.65	**0.86**	0.8620	**0.80**	0.84	0.94
CiteULike	*RVAE*	0.2875	0.07	0.22	**0.67**	0.6021	0.29	0.57	**0.90**	0.7638	0.48	0.75	**0.95**
	$RVAE^W$	0.2711	0.09	0.25	0.46	0.6156	0.37	0.61	0.76	0.7698	0.55	0.77	0.87
	$RVAE^S$	**0.3890**	**0.32**	**0.39**	0.43	**0.7264**	**0.61**	**0.73**	0.76	**0.8399**	**0.73**	**0.84**	0.90
	$RVAE^E$	0.2805	0.16	0.21	0.66	0.5634	0.50	0.50	0.87	0.6941	0.71	0.64	0.92

Table 2
Comparative analysis.

Table 2 summarizes the results of the evaluation. We see that, in general, all the strategies considerably improve the response of the model to the low popular items. However, $RVAE^W$ has a low response on the high popular items. By contrast, $RVAE^S$ succeeds in boosting performance on both the low popular and the mid popular items. Overall, the ensemble $RVAE^E$ provides the best response, by boosting low popular items without substantially degrading over the other classes.

5. Conclusions

The approach proposed in this paper is a preliminary study: We introduce a Ranking collaborative filtering algorithm (RVAE) and study how the algorithm is affected by popularity bias. Next, we show how simple techniques based on reweighting/resampling and/or ensembling can recalibrate the recommendations. There are several aspects that are worth further investigation. First of all, both the weighting and the inverse stratified sampling schemes are based on hyperparameters that need to be carefully tuned. Also, the ensemble strategies are simple and more complex schemes that also take into account other model instantiations can be studied. We reserve the attention to these challenges in a future work.

References

[1] V. Tsintzou, E. Pitoura, P. Tsaparas, Bias disparity in recommendation systems, CoRR abs/1811.01461 (2018). URL: http://arxiv.org/abs/1811.01461.

[2] C. C. Aggarwal, Recommender Systems, Springer, 2016.

[3] B. Friedman, H. Nissenbaum, Bias in computer systems, ACM Trans. Inf. Syst. 14 (1996) 330–347.

[4] Z. Zhu, X. Hu, J. Caverlee, Fairness-aware tensor-based recommendation, in: ACM International Conference on Information and Knowledge Management, CIKM '18, 2018, p. 1153–1162.

[5] Y. Deldjoo, V. W. Anelli, H. Zamani, A. Bellogin, T. Di Noia, A flexible framework for evaluating user and item fairness in recommender systems, User Modeling and User-Adapted Interaction (2021).
[6] O. Celma, P. Cano, From hits to niches?: Or how popular artists can bias music recommendation and discovery, in: 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition, 2008.
[7] H. Steck, Item popularity and recommendation accuracy, in: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, 2011, p. 125–132.
[8] R. Borges, K. Stefanidis, On measuring popularity bias in collaborative filtering data, in: EDBT Workshop on BigVis 2020: Big Data Visual Exploration and Analytics, 2020.
[9] H. Abdollahpouri, M. Mansoury, R. Burke, B. Mobasher, The unfairness of popularity bias in recommendation, CoRR abs/1907.13286 (2019). URL: http://arxiv.org/abs/1907.13286.
[10] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, in: Conf. on Uncertainty in Artificial Intelligence, UAI '09, 2009, pp. 452–461.
[11] T. Ebesu, B. Shen, Y. Fang, Collaborative memory network for recommendation systems, in: ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '18, 2018.
[12] Z. Zhu, J. Wang, J. Caverlee, Measuring and mitigating item under-recommendation bias in personalized ranking systems, in: ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, 2020, p. 449–458.
[13] H. Abdollahpouri, R. Burke, B. Mobasher, Managing popularity bias in recommender systems with personalized re-ranking, CoRR abs/1901.07555 (2019). URL: http://arxiv.org/abs/1901.07555.
[14] D. Liang, R. G. Krishnan, M. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, in: ACM Conf, on World Wide Web, WWW '18, 2018, pp. 689–698.
[15] D. P. Kingma, M. Welling, Auto-encoding variational bayes, in: 2nd International Conference on Learning Representations, ICLR'14, 2014. arXiv:http://arxiv.org/abs/1312.6114v10.