

# Scalable Personalised Item Ranking through Parametric Density Estimation

Riku Togashi  
CyberAgent, Inc., Waseda University  
Tokyo, Japan  
rtogashi@acm.org

Masahiro Kato  
CyberAgent, Inc.  
Tokyo, Japan  
masahiro\_kato@cyberagent.co.jp

Mayu Otani  
CyberAgent, Inc.  
Tokyo, Japan  
otani\_mayu@cyberagent.co.jp

Tetsuya Sakai  
Waseda University  
Tokyo, Japan  
tetsuyasakai@acm.org

Shin'ichi Satoh  
CyberAgent, Inc.  
Tokyo, Japan  
satoh@nii.ac.jp

## ABSTRACT

Learning from implicit feedback is challenging because of the difficult nature of the one-class problem: we can observe only positive examples. Most conventional methods use a pairwise ranking approach and negative samplers to cope with the one-class problem. However, such methods have two main drawbacks particularly in large-scale applications; (1) the pairwise approach is severely inefficient due to the quadratic computational cost; and (2) even recent model-based samplers (e.g. IRGAN) cannot achieve practical efficiency due to the training of an extra model.

In this paper, we propose a learning-to-rank approach, which achieves convergence speed comparable to the pointwise counterpart while performing similarly to the pairwise counterpart in terms of ranking effectiveness. Our approach estimates the probability densities of positive items for each user within a rich class of distributions, viz. *exponential family*. In our formulation, we derive a loss function and the appropriate negative sampling distribution based on maximum likelihood estimation. We also develop a practical technique for risk approximation and a regularisation scheme. We then discuss that our single-model approach is equivalent to an IRGAN variant under a certain condition. Through experiments on real-world datasets, our approach outperforms the pointwise and pairwise counterparts in terms of effectiveness and efficiency.

## CCS CONCEPTS

• **Information systems** → **Recommender systems; Learning to rank**; • **Computing methodologies** → **Learning from implicit feedback**.

## KEYWORDS

personalised recommendation; collaborative filtering; implicit feedback; learning to rank

## ACM Reference Format:

Riku Togashi, Masahiro Kato, Mayu Otani, Tetsuya Sakai, and Shin'ichi Satoh. 2021. Scalable Personalised Item Ranking through Parametric Density Estimation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, July 11–15, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3404835.3462933>

## 1 INTRODUCTION

Recommender systems play a central role in alleviating information overload problems by helping users to discover content relevant to their interests. To find relevant content from databases that are updated daily, implicit user feedback (e.g. clicks) is an important clue to users' recent interests because it is automatically collected and thus abundant. However, the increase in the number of recommendation candidates leads to the sparsity of user feedback, overwhelming the accumulation of observed feedback. Moreover, the use of implicit feedback incites the *one-class problem* [47] in model training, i.e., we can observe only positive examples.

For learning personalised ranking from positive-only data, a prevalent approach is to solve pairwise ranking problems [50]. The pairwise approach has the advantage in ranking effectiveness, but training is severely inefficient due to the quadratic computational cost of the expected loss (i.e. the risk) for each user. By contrast, the pointwise approach can be trained efficiently in large-scale settings, but it returns less effective rankings. Taking the advantage of the pointwise approach, non-sampling methods [12, 13] based on mini-batch stochastic gradient descent (SGD) have demonstrated excellent convergence time with competitive effectiveness. In addition, the SGD-based approaches enable the use of non-uniform weights for negative instances, which is often infeasible when using alternating least squares (ALS) [46, 47].

Missing pairs of users and items could complement the positive-only data because these may provide users' negative signals. When dealing with large-scale data, it is challenging to find informative negatives from a large number of candidates. Therefore, a negative sampling/weighting strategy is a critical factor for efficient model training [49, 50, 70]. For alleviating the training inefficiency of the pairwise approach, previous studies explored strategies based on item popularity [47], predicted scores of the ranker model [49, 70], an adversarial sampler model [61], and reinforcement learning [20]. Since a sampling strategy aims to find items with high preference

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-8037-9/21/07...\$15.00  
<https://doi.org/10.1145/3404835.3462933>

scores from unobserved samples, the strategy will behave like a recommender model. Hence, advanced strategies often require an extra sampler model which learns to rank negative candidates according to their informativeness. The IRGAN framework [61], which is based on generative adversarial networks (GANs) [25], provides an optimised negative sampler (i.e. the generator) for the ranker (i.e. the discriminator). However, such a model-based sampler faces two challenges that result from the training of the two models. First, the training process introduces hyper-parameters that are difficult to tune, such as the number of iterations for training the discriminator and generator in each epoch. Second, it leads to inefficient training because multiple models need to be trained.

Herein, we propose a learning-to-rank approach for personalised ranking from implicit feedback based on parametric density estimation. We derive a loss function as a likelihood of generative distributions of user feedback. In our formulation, we can obtain the negative sampling distribution for a user as the estimated generative distribution of the user's feedback. To enable the use of the various models such as graph neural networks (GNNs), we also propose risk approximation and regularisation techniques that encourage stable and efficient model training using SGD-based empirical risk minimisation. Furthermore, we show the equivalence between our approach and an IRGAN variant based on Wasserstein GAN (WGAN) [3] under a certain condition. Based on this, we propose a variant of our approach, which estimates the distribution of positive items over *unobserved items* through the minimisation of the Wasserstein distance. We then show that our approach behaves theoretically similarly to the pairwise ranking approach with a model-based negative sampler while enabling drastically faster convergence in empirical experiments.

## 2 BACKGROUND AND RELATED WORK

### 2.1 Problem Formulation

For a set of users  $\mathcal{U}$  and a set of items  $\mathcal{I}$ , we observe  $N$  i.i.d. user feedback logs  $(u_1, i_1, y_1), \dots, (u_N, i_N, y_N)$ , where  $u \in \mathcal{U}$ ,  $i \in \mathcal{I}$ , and  $y \in \{-1, +1\}$  are user, item, and class labels, respectively. However, if only implicit feedback is available, we can observe only positive samples, in other words, those labelled as  $y = +1$ . Given a user, a ranker sorts items in the order in which they are most likely to interact with the user.

### 2.2 Paradigms of Personalised Ranking

In top- $K$  personalised ranking, most pointwise methods solve a binary classification problem to predict if a user prefers an item [31, 34, 47]; we will refer to this as the *class-posterior probability estimation (CPE)* approach. The CPE approach, which estimates class-posterior  $p(y = +1|u, i)$ , often impairs ranking performance, whereas it is useful for the tasks such as CTR prediction. On the other hand, owing to pointwise loss functions, the CPE approach has an advantage in the training efficiency compared with the pairwise counterpart. Methods based on the CPE approach often introduce weighting strategies for selecting important unobserved samples. Early works explored non-uniform weighting schemes such as item popularity [46, 47], but non-uniform weights lead to severe computational costs. For the sake of training efficiency, static and uniform weights

have been adopted widely [31]. Recent studies have proposed efficient methods that use non-uniform weights based on ALS [30] and SGD [11–13]. Several studies have explored weighting strategies for the CPE approach based on item popularity, user exposure models [37], and social information [15]. As the inefficiency issue of adaptive weighting [37] is problematic in a large-scale setting, Chen et al. [14] have proposed an efficient optimisation method with adaptive weighting strategy for matrix factorisation (MF).

In contrast to the CPE approach that estimates  $p(y = +1|u, i) = p(i|u, y = +1)p(y = +1|u)/p(i|u)$ , the recently-proposed *density ratio estimation (DRE)* approach [57] directly estimates the density ratio  $p(i|u, y = +1)/p(i|u)$  through the minimisation of the Bregman divergences [55], which can be empirically approximated with positive-unlabelled data. As the main distinction of the DRE approach from its CPE counterpart, the ranker model in the DRE approach does not need to estimate class prior  $p(y = +1|u)$  (i.e. user's activity), which is a constant value and does not affect the predicted ranking for  $u$ ; thus, it avoids the estimation error of  $p(y = +1|u)$  in the ranker model. Several studies [45, 57] have shown that the advantage of the DRE-based loss functions for top- $K$  ranking problems by weighing the items in the top of ranked lists in contrast to the CPE approach. Togashi et al. [57] empirically demonstrated the advantage of the DRE-based approach in terms of efficiency and effectiveness for top- $K$  personalised recommendation. Although their framework can be utilised with an arbitrary weighting strategy, their proposed adaptive and ranking-aware weighting strategies are heuristically designed.

By directly modelling the generative process of implicit feedback from each user, estimating  $p(i|u, y = +1)$  is a natural direction; we call this approach the *density estimation (DE)* approach. The intuition behind estimating  $p(i|u, y = +1)$  is that, as  $\int p(i|u, y = +1)di = 1$  holds for each user, the items must compete for the limited budget of the total probability mass. This one-sum property is suitable for top- $K$  ranking tasks because it induces relativity in the model predictions [38, 68]. The multinomial logit model (a.k.a. softmax model) [5, 42, 43] is widely adopted as the model of  $p(i|u, y = +1)$ :

$$q(i|u) = \frac{\exp(f_u(i))}{\sum_{i \in \mathcal{I}} \exp(f_u(i))} \quad (1)$$

where  $f_u(i) \in \mathbb{R}$  is  $u$ 's preference score for  $i$ . Various methods have been proposed based on this parametric density function, such as collaborative competitive filtering (CCF) [68] and VAE-based collaborative filtering [38, 39, 53]. For quantifying the distance between the true and estimated distributions, Kullback-Leibler divergence (KLD) [35] is often utilised. Tanielian et al. [56] recently proposed a DE approach based on the maximum likelihood estimation (MLE) of softmax density functions. However, their MLE formulation leads to intractable log-partition  $\log \sum_{i \in \mathcal{I}} \exp(f_u(i))$  in the log-likelihood function, which makes SGD-based optimisation difficult for large-scale settings. They also discussed that their approach can be regarded as the minimisation of KLD. On the other hand, Tolstikhin et al. [58] have shown that, when using a VAE-based method, minimising the Wasserstein distance (WD) [59] is equivalent to optimising the evidence lower bound (ELBO). This fact allows us to use the WD rather than the KLD, which is inappropriate for the high-dimensional and sparse data that we have when dealing

with user implicit feedback. Based on this fact, recent studies have examined approaches that estimate the probability densities by minimising the WD [39, 44, 69, 71]. However, VAE-based methods restrict the use of flexible models, such as MF [31, 34, 47, 50] and GNNs [28, 63]. Furthermore, the VAE-based models often include a hidden layer that is as large as the number of items, making these models computationally costly for a large-scale setting.

The pairwise approach introduces another binary classification problem of predicting which of any given two items a user will prefer [1, 8, 9, 16, 50]. As the pairwise approach avoids predicting the absolute relevance of a user-item pair, it is often more appropriate than the CPE approach for top- $K$  recommendation tasks. While this approach is advantageous in terms of effectiveness, it has a serious disadvantage in the convergence time. Empirical risk minimisation based on U-statistics (e.g. Kendall's  $\tau$ ) is practically infeasible in large-scale settings due to its quadratic computational cost with respect to  $|\mathcal{I}|$ . Some techniques to alleviate this problem are available [17, 48], but such techniques are not well examined in IR and recommendation literature. Previous studies have proposed the top-emphasised pairwise approaches based on order statistics [65, 66], push-based losses [16] and importance sampling [36].

Our approach returns to the simple MLE of  $p(i|u, y = +1)$  within an exponential family [2, 10, 18, 19, 54], which includes the softmax model adopted in the conventional DE approaches. Using the penalised MLE of parametric density functions, we derive a loss function and an appropriate sampling/weighting strategy from a perspective of empirical risk minimisation. Our methodology is applicable to most common model architectures like MF and GNNs. Moreover, it offers two advantages in terms of training efficiency when compared with the pairwise approach; while working in a pointwise fashion, it provides the optimal negative sampler without an extra model.

### 2.3 Negative Sampling and Generative Adversarial Networks

Because the choice of negative sampling strategy has a great impact on both efficiency and effectiveness, various approaches have been explored based on item popularity, ranker's predicted scores [49, 70], additional view data [22], optimised sampler models [61], and reinforcement learning [20].

Early works [49, 70] have developed ranker-based sampling strategies, which select the negatives scored higher by the ranker for training. Ding et al. [21] have recently explored a simple and robust negative sampling strategy, which provides an efficient and effective implementation of negative sampling and simplifies model-based samplers for practical applications. In contrast to a ranker-based sampler, Wang et al. [61] proposed a learning-to-rank framework based on GANs and formulated ranking tasks as a min-max game between two rankers, a discriminator and a generator. For personalised recommendation tasks, the generator model is mainly regarded as the negative sampler for the ranker. Hence, GAN-based methods have often been explored in the context of the negative sampling strategies [23, 26, 40, 60, 62, 70]. On the other hand, since GANs originally aim to estimate the generative distribution of observed data, when using the generator as the ranker model, GAN-based methods are more like the DE approach; the softmax model

is widely adopted for the generator. The disadvantage of the GAN-based approach is the computational cost of the following steps in the model training: (1) Training the two models for the ranker and sampler. (2) Negative sampling from the generator's distribution over all items. These steps also pose a challenge for the practical implementation of large-scale recommender systems. Moreover, several prior studies [20, 64] have reported that GAN-based negative samplers can underperform ranker-based samplers such as dynamic negative sampling (DNS) [70], which adaptively selects the negative item scored highest by the current ranker.

In this study, we aim to develop a risk function incorporating the concept of GAN-based samplers. Remarkably, under a certain assumption on the ranker (discriminator), our MLE-based approach is equivalent to a variant of IRGAN with a regularised generator while omitting the generator model in the formulation of GANs. Whereas our motivation to simplify the recent negative samplers is similar to that of Ding et al. [21], we shall propose a risk function rather than the algorithm for negative sampling.

## 3 PROPOSED METHOD

### 3.1 Parametric Density Functions

We formulate the personalised ranking as the estimation of the densities of items conditioned on a user and the positive class, namely,  $p(\cdot|u, y = +1)$  on  $\mathcal{I}$ . To this end, we first specify the model of  $p(\cdot|u, y = +1)$  with an exponential family parametrised by  $f_u$ .

$$p_f(i|u) := p_0(i) \exp(f_u(i) - A(f_u)), \quad (2)$$

where  $f_u$  denotes the sufficient statistics, and  $p_0$  is a base density function on  $\mathcal{I}$ . Here,  $A(f_u) := \log \int p_0(i) \exp(f_u(i)) di$  is the normalisation term that ensures  $\int p_f(i|u) di = 1$  (a.k.a. log-partition function). In the context of personalised ranking,  $f_u(i)$  is the scoring function to learn, which represents  $u$ 's preference score for  $i$ . In typical models, such as MF and GNNs, each score is computed as the similarity between  $u$  and  $i$  in the latent feature space.

$$f_u(i) := \mathbf{e}_u^\top \mathbf{e}_i, \quad (3)$$

where  $\mathbf{e}_u \in \mathbb{R}^d$  and  $\mathbf{e}_i \in \mathbb{R}^d$  are the  $d$ -dimensional latent embeddings of  $u$  and  $i$ , respectively. Considering practical applications, we assume that  $p_0(i)$  is a uniform probability for all  $i$ ; as a result,  $p_f(i|u) \propto \exp(f_u(i))$  holds, and therefore, the order of  $p_f(i|u)$  is determined only by the order of  $f_u(i)$ . This allows us to utilise approximate nearest neighbour (ANN) techniques [41] to efficiently perform the top- $K$  maximum inner product search (MIPS) for a user. This is essential for real applications in which the number of items is large [51].

### 3.2 Penalised Maximum Likelihood Estimation

To learn  $f_u(i)$  for each  $u$  from the observed implicit feedback, we formulate the MLE of density functions of positive items for users. Because the naïve MLE is ill-posed and can impair generalisation performance, we introduce a penalisation term based on the KLD between the estimated distribution and a uniform distribution. We can express the empirical risk based on the negative log-likelihood

function with KLD penalisation as follows:

$$\begin{aligned} R(f) &:= \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-\log p_f(i|u)] + KLD(p_f(\cdot|u)||p_0(\cdot)) \right] \quad (4) \\ &= \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-\log p_0(i) - f_u(i) + A(f_u)] + KLD(p_f(\cdot|u)||p_0(\cdot)) \right] \quad (5) \\ &= \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-f_u(i)] + A(f_u) + KLD(p_f(\cdot|u)||p_0(\cdot)) \right] + C, \quad (6) \end{aligned}$$

where  $\widehat{\mathbb{E}}_+ = \widehat{\mathbb{E}}_{p(i|u, y=+1)}$  is the empirical expectation over the training positive items for a given user  $u$ , and  $C$  is a constant value with respect to the training of  $f_u$ . In Eq. (5), we used the definition of  $p_f(i|u)$  in Eq. (2). The problem here is that  $A(f_u) = \log \int p_0(i) \exp(f_u(i)) di$  is intractable, and thus,  $R(f)$  is difficult to be approximated through mini-batch SGD. To solve this, we transform the KLD between  $p_f(\cdot|u)$  and  $p_0(\cdot)$  as follows:

$$KLD(p_f(\cdot|u)||p_0(\cdot)) := \int p_f(i|u) \log(p_f(i|u)/p_0(i)) di \quad (7)$$

$$= \int p_f(i|u) \log \exp(f_u(i) - A(f_u)) di \quad (8)$$

$$= \int p_f(i|u) f_u(i) di - A(f_u) \quad (9)$$

$$= \mathbb{E}_{p_f(i|u)} [f_u(i)] - A(f_u). \quad (10)$$

Plugging this into Eq. (6), we can reformulate the risk as follows:

$$R(f) = \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-f_u(i)] + A(f_u) + \mathbb{E}_{p_f(i|u)} [f_u(i)] - A(f_u) \right] \quad (11)$$

$$= \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-f_u(i)] + \mathbb{E}_{p_f(i|u)} [f_u(i)] \right]. \quad (12)$$

Remarkably, the appropriate distribution for negative sampling under ranker  $f_u(\cdot)$  is exactly  $p_f(i|u)$  in our formulation. Note that we use all samples including (even observed) positive ones to compute the second term of the RHS in Eq. (12), although it corresponds to the risk for negative instances in the conventional pointwise or pairwise approaches; we shall discuss this point in Section 4.2.

### 3.3 Risk Approximation

For sampling negatives from ranker-dependent  $p_f(i|u)$ , we need to compute each user's preference scores for all items; this is infeasible in a large-scale setting. Considering this, our approach follows the two-phased sampling strategy [49, 70]. We first sample a set of item candidates from a static sampling distribution and then evaluate the preference scores for them on the fly. In addition, we shall tackle another challenge in negative sampling with ranker-dependent distributions. That is, because  $p_f(i|u)$  changes with the update to  $f$  in each training step, costly Monte-Carlo sampling is often required at every training step to accurately approximate the gradient of  $\mathbb{E}_{p_f(i|u)} [f_u(i)]$ . To achieve better risk approximation without sampling from  $p_f(i|u)$ , we transform  $R(f)$  by leveraging explicit parametric model  $p_f(i|u)$  defined in Eq. (2). By introducing

importance sampling, we can obtain the following risk:

$$R(f) = \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-f_u(i)] + \mathbb{E}_{p_0} \left[ \frac{p_f(i|u)}{p_0(i)} f_u(i) \right] \right] \quad (13)$$

$$= \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-f_u(i)] + \frac{\mathbb{E}_{p_0} [\exp(f_u(i)) f_u(i)]}{\exp A(f_u)} \right] \quad (14)$$

$$= \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-f_u(i)] + \frac{\mathbb{E}_{p_0} [\exp(f_u(i)) f_u(i)]}{\mathbb{E}_{p_0} [\exp(f_u(i))]} \right]. \quad (15)$$

In the third equation, we use  $\exp A(f_u) = \int p_0(i) \exp(f_u(i)) di = \mathbb{E}_{p_0} [\exp(f_u(i))]$ . Finally, we reach the empirically approximated risk with a given mini-batch.

$$\begin{aligned} \widehat{R}(f, \mathcal{U}_B, \mathcal{I}_B) \\ := \frac{1}{|\mathcal{U}_B|} \sum_{u \in \mathcal{U}_B} \left( -\frac{1}{|\mathcal{I}_u^+|} \sum_{i \in \mathcal{I}_u^+} f_u(i) + \frac{\sum_{i \in \mathcal{I}_B} \exp(f_u(i)) f_u(i)}{\sum_{i \in \mathcal{I}_B} \exp(f_u(i))} \right), \end{aligned} \quad (16)$$

where  $\mathcal{U}_B$  and  $\mathcal{I}_B$  are the sets of sampled users and items, respectively. Here,  $\mathcal{I}_u^+$  is the set of observed positive items for  $u$ . Note that  $\widehat{R}(f, \mathcal{U}_B, \mathcal{I}_B)$  is biased for general sampling distributions of  $\mathcal{I}_B$ . However, we found in our preliminary experiments that the corrected risk estimator with inverse sampling probabilities did not perform well due to a large variance of the estimator, and we therefore omit it here for the sake of simplicity.

For mini-batch sampling, we adopted the same user-based sampling strategy as is used in conventional approaches [38, 57]. We uniformly sample  $|\mathcal{U}_B|$  users and then sample all positive items  $\mathcal{I}_u^+$  for each user. For computing  $\widehat{R}(f, \mathcal{U}_B, \mathcal{I}_B)$ , we build a set of items  $\mathcal{I}_B$  in a mini-batch as unique sampled items. We leave the use of advanced samplers for  $\mathcal{I}_B$  as future work.

In practice, we minimise  $\widehat{R}(f, \mathcal{U}_B, \mathcal{I}_B)$  with the  $L_2$  regularisation for the latent vectors as follows:

$$\min_f \widehat{R}(f, \mathcal{U}_B, \mathcal{I}_B) + \lambda \mathcal{R}(f, \mathcal{U}_B, \mathcal{I}_B), \quad (17)$$

where  $\mathcal{R}(f, \mathcal{U}_B, \mathcal{I}_B)$  is the  $L_2$  regularisation term, and  $\lambda$  is the hyper-parameter that controls the intensity of the regularisation.

### 3.4 Norm Clipping Technique

Intuitively, owing to the one-sum property of  $p_f(i|u)$ , the spike of the probability mass for a certain item dominates the total probability mass. This effect leads to overfitting or a large variance in  $\widehat{R}(f, \mathcal{U}_B, \mathcal{I}_B)$ , and thus destabilises the model training. This observation suggests the following smoothness condition for  $p_f(i|u)$ :

$$\max_{i, j \in \mathcal{I}} |p_f(i|u) - p_f(j|u)| \leq L, \quad (18)$$

where  $L \in (0, \infty)$  is the upper bound of the gap between the probability mass assigned to two items. Owing to the monotonicity of the exponential function, we can rewrite this condition as follows:

$$\max_{i, j \in \mathcal{I}} |p_f(i|u) - p_f(j|u)| \leq L \Leftrightarrow \max_{i, j \in \mathcal{I}} |f_u(i) - f_u(j)| \leq \tilde{L} \quad (19)$$

$$\Leftrightarrow \max_{i, j \in \mathcal{I}} |\mathbf{e}_i^\top (\mathbf{e}_i - \mathbf{e}_j)| \leq \tilde{L} \quad (20)$$

$$\Leftrightarrow \max_{v \in \{u\} \cup \mathcal{I}} \|\mathbf{e}_v\| \leq \tilde{n}, \quad (21)$$

where  $L$  is the upper bound of the gap between two preference scores, and  $\bar{n} \in (0, \infty)$  is the upper bound of the  $L_2$  norms of the latent vectors. To satisfy the smoothness condition, we ensure that the latent vectors for all users and items have upper-bounded  $L_2$  norms. It is worth noting that the  $L_2$  regularisation is not appropriate for ensuring the condition because it penalises the average of the  $L_2$  norms of latent vectors rather than their maxima. Therefore, we propose an  $L_2$  norm clipping technique. For methods based on MF, we define the following clipping operator:

$$\text{clip}(\mathbf{e}, \bar{n}) := \begin{cases} \mathbf{e}, & \text{if } \|\mathbf{e}\| \leq \bar{n} \\ \frac{\bar{n}}{\|\mathbf{e}\|} \mathbf{e}, & \text{if } \|\mathbf{e}\| > \bar{n} \end{cases}, \quad (22)$$

where  $\mathbf{e} \in \mathbb{R}^d$  denotes a latent vector. We apply the clipping operator to each latent vector after updating it in each training step.

This technique is also applicable to the recently-proposed GNN-based method, LightGCN [28], which considers high-order smoothness on the user-item graph. In LightGCN, each final latent vector for prediction is computed as the linear combination of the trainable latent vectors in the first layer:

$$\mathbf{e}_u = \sum_{v \in \mathcal{G}} \beta_{v,u} \mathbf{e}_v^{(1)}, \quad \mathbf{e}_i = \sum_{v \in \mathcal{G}} \beta_{v,i} \mathbf{e}_v^{(1)}, \quad (23)$$

where  $\mathcal{G} = \mathcal{U} \cup \mathcal{I}$  is the set of nodes in the user-item graph, and  $\mathbf{e}_v^{(1)}$  denotes the latent vector for  $v \in \mathcal{G}$  in the first layer. Here,  $\beta_{v,u}$  and  $\beta_{v,i}$  are non-negative coefficients in the linear combination. Therefore, when clipping the latent vectors in the first layer, the norms of the final vectors are also upper-bounded.

## 4 THEORETICAL DISCUSSION

In this section, we first discuss the relationship between our approach and IRGAN; based on the discussion, we propose a variant of our DE approach based on the WD. We then discuss its connection to the pairwise approach.

### 4.1 Relationship to IRGAN

Previous studies pointed out the connection between GANs and density estimation [2, 18, 19]. Following these studies, we discuss the similarity between our single-model approach and an IRGAN variant based on the WGAN [3].

We first assume a method based on WGAN, namely, *IRWGAN*. We can express the dual form of the Wasserstein-1 distance between the true and estimated distributions  $P_u$  and  $Q_u$  for a user as follows:

$$WD(P_u || Q_u) := \sup_{\|f_u\|_{L \leq 1}} \mathbb{E}_{i \sim P_u} [f_u(i)] - \mathbb{E}_{i \sim Q_u} [f_u(i)], \quad (24)$$

where  $\|f_u\|_{L \leq 1}$  requires the 1-Lipschitz continuity of  $f_u$ , namely,  $\max_{i,j \in \mathcal{I}} |f_u(i) - f_u(j)| \leq 1$ ; we here assume that  $\mathcal{I}$  is a discrete metric space.

By formulating density estimation as the WD minimisation under KLD regularisation for the generator, we can obtain the min-max objective of IRWGAN for user  $u$  as follows:

$$\begin{aligned} & \min_{q_u \in \mathcal{P}} KLD(q_u || p_0) + WD(P_u || Q_u) \\ &= \min_{q_u \in \mathcal{P}} \left\{ KLD(q_u || p_0) + \sup_{f_u \in \mathcal{F}_1} \left\{ \widehat{\mathbb{E}}_+ [f_u(i)] - \mathbb{E}_{q_u(i)} [f_u(i)] \right\} \right\}, \end{aligned} \quad (25)$$

where  $q_u \in \mathcal{P}$  and  $f_u \in \mathcal{F}_1$  are the generator and discriminator for  $u$ , respectively. We assume that  $\mathcal{P}$  is the set of probability measures supported on  $\mathcal{I}$ , and  $\mathcal{F}_1$  is a convex subset of 1-Lipschitz functions.

We here show that IRWGAN with convex  $\mathcal{F}_1$  leads to the same ranker as the proposed approach in Section 3. To this end, by following Theorem 1 of Farnia et al. [24], we first consider the duality of the IRWGAN objective.

**THEOREM 1 (STRONG DUALITY OF IRWGAN).** *Suppose a set  $\mathcal{F}_1$  is convex. Then, the strong duality of the IRWGAN holds, namely,*

$$\begin{aligned} & \text{RHS of Eq. (25)} \\ &= \sup_{f_u \in \mathcal{F}_1} \left\{ \widehat{\mathbb{E}}_+ [f_u(i)] + \min_{q_u \in \mathcal{P}} \left\{ KLD(q_u || p_0) - \mathbb{E}_{q_u(i)} [f_u(i)] \right\} \right\}. \end{aligned} \quad (26)$$

**PROOF.** We show the applicability of von Neumann-Fan minimax theorem (Theorem 2 of Borwein's work [6]), which guarantees the strong duality of min-max objectives under several conditions. The space  $\mathcal{P}$  of probability measures on compact  $\mathcal{I}$  is convex and weakly compact [4]. The set  $\mathcal{F}_1$  of 1-Lipschitz continuous functions is convex as we assumed. The objective in Eq. (25) is linear (i.e. concave) in  $f_u$  while it is lower semi-continuous and convex in  $q_u$  owing to the KLD regularisation for  $q_u$ . Therefore, the strong duality of IRWGAN holds by applying the minimax theorem.  $\square$

By using Theorem 1, we obtain the risk for  $f$  with the optimal generators  $q_u^*$  based on the max-min objective (i.e. the RHS of Eq. (26)) as follows:

$$P_{\text{wgan}}^*(f) := \mathbb{E}_u \left[ \widehat{\mathbb{E}}_+ [-f_u(i)] + \mathbb{E}_{q_u^*(i)} [f_u(i)] \right]. \quad (27)$$

Here, based on Eq. (26), the optimal distribution  $q_u^*(i)$  for  $u$  is the one that maximises objective  $L_u(q_u) := \mathbb{E}_{q_u(i)} [f_u(i)] - KLD(q_u || p_0)$ . To obtain  $q_u^*(i)$ , we solve the following maximisation problem with respect to  $q_u$ :

$$\begin{aligned} & \max_{q_u \in \mathcal{P}} \mathbb{E}_{q_u} [f_u(i)] - KLD(q_u || p_0) \\ &= \max_{q_u \in \mathcal{P}} \int q_u(i) f_u(i) di - \int q_u(i) (\log q_u(i) - \log p_0) di \quad (28) \\ &= \max_{q_u \in \mathcal{P}} \int q_u(i) (f_u(i) + \log p_0 - \log q_u(i)) di. \end{aligned} \quad (29)$$

By considering the differentiation with respect to density  $q_u(i)$ ,  $\nabla_{q_u(i)} L_u(q_u) = 0$  holds for each  $i$  at the optimal  $q_u^*$  owing to the strong concavity of  $L_u(\cdot)$ . Thus, we can obtain the optimal density  $q_u^*(i)$  as the following closed form.

$$\nabla_{q_u(i)} L_u(q_u^*) = 0 \Leftrightarrow f_u(i) + \log p_0 - \log q_u^*(i) - 1 = 0 \quad (30)$$

$$\Leftrightarrow \log q_u^*(i) = f_u(i) + \log p_0 - 1 \quad (31)$$

$$\Leftrightarrow q_u^*(i) \propto p_0 \exp(f_u(i)). \quad (32)$$

Therefore,  $q_u^*(i)$  is exactly equivalent to  $p_f(i|u)$  defined in Eq. (2). By substituting  $q_u^*(i) = p_f(i|u)$  into Eq. (27), we immediately obtain the same risk as  $R(f)$  in Eq. (12).

Based on this result, if we aim to obtain discriminator  $f$  in IRWGAN with convex  $\mathcal{F}_1$  as the ranker, we can omit the generator model by obtaining the optimal sampling distributions based on  $f_u$  as a closed form. It allows us to keep the sampler up-to-date at

every training step while reducing computational costs and hyper-parameters. We can utilise the norm clipping technique to obtain the  $\tilde{L}$ -Lipschitz continuity of  $f_u$  (the RHS of Eq. (19). In other words, with the norm clipping technique, our approach may minimise  $\tilde{L} \cdot WD(P_u || Q_u)$  with KLD regularisation. It should be noted that the feasible set of  $f_u$  (i.e.  $\mathcal{F}_1$ ) is generally *not* convex when using complex model architectures (probably including LightGCN), and it may lead to a duality gap in yielding the max-min form in Theorem 1 to some extent.

## 4.2 Estimating Densities on Unobserved Items

In the ranking evaluation, we do not use the training items (i.e. observed positive items) to create a ranked list for a user. Considering this, we propose a variant of our approach which estimates  $p(\cdot | u, y = +1)$  on  $\mathcal{I} \setminus \mathcal{I}_u^+$  rather than  $p(\cdot | u, y = +1)$  on  $\mathcal{I}$ . The problem here is that the empirical distribution of positive items has a support  $\mathcal{I}_u^+$  which is disjoint with that of the distribution to be estimated; namely,  $\mathcal{I} \setminus \mathcal{I}_u^+$ . However, based on the discussion in Section 4.1, our approach may minimise the WD, which can be defined even between two distributions with disjoint supports [3]. Therefore, our approach can safely estimate the density function by minimising the WD between two disjoint distributions over observed and unobserved items, whereas many divergences (e.g. KLD) are inappropriate for such distributions.

Assuming that our approach minimises the WD, we can obtain the risk for the density estimation on unobserved items by defining  $p_f(\cdot | u)$  and  $p_0(\cdot)$  on  $\mathcal{I} \setminus \mathcal{I}_u^+$ . Thus, we can also obtain the risk estimator by simply replacing  $\mathcal{I}_B$  with  $\mathcal{I}_B \setminus \mathcal{I}_u^+$  in the RHS of Eq. (16) as follows:

$$\begin{aligned} \hat{R}_{WD}(f, \mathcal{U}_B, \mathcal{I}_B) \\ := \frac{1}{|\mathcal{U}_B|} \sum_{u \in \mathcal{U}_B} \left( -\frac{1}{|\mathcal{I}_u^+|} \sum_{i \in \mathcal{I}_u^+} f_u(i) + \frac{\sum_{i \in \mathcal{I}_B \setminus \mathcal{I}_u^+} \exp(f_u(i)) f_u(i)}{\sum_{i \in \mathcal{I}_B \setminus \mathcal{I}_u^+} \exp(f_u(i))} \right). \end{aligned} \quad (33)$$

## 4.3 Relationship to Pairwise Approach

We shall discuss how our proposed risk relates to that of pairwise approaches. The user-conditional empirical risk of a pairwise approach with negative item sampler  $Q_u$  can be defined as follows:

$$R_{\text{pair}}(f_u) := \mathbb{E}_{i \sim P_u} \mathbb{E}_{j \sim Q_u} [s(f_u(j) - f_u(i))] \quad (34)$$

where  $s(\cdot) = \log(1 + \exp(\cdot))$  is the softplus function; here,  $s(f_u(j) - f_u(i))$  is a widely adopted differentiable surrogate of the 0-1 loss,  $\mathbb{1}\{f_u(j) > f_u(i)\}$  [16, 50]. Using the convexity of the softplus function, we can evaluate the following lower bound of the pairwise risk by applying Jensen's inequality.

$$R_{\text{pair}}(f_u) \geq s \left( \mathbb{E}_{i \sim P_u} [-f_u(i)] + \mathbb{E}_{i \sim Q_u} [f_u(i)] \right). \quad (35)$$

Therefore, when using  $p_f(i|u)$  on  $\mathcal{I} \setminus \mathcal{I}_u^+$  for  $Q_u$ , the RHS in Eq. (35), the lower bound of the pairwise risk corresponds to the proposed WD-based risk for a given user. Because  $p_f(i|u)$  maximises the RHS of Eq. (35) under the KLD regularisation as shown in Section 4.1, our approach minimises the upper bound of the lower bound of the pairwise risk. Here, we let  $\mu$  be the short-hand for

$\mathbb{E}_{i \sim P_u} \mathbb{E}_{j \sim Q_u} [f_u(j) - f_u(i)]$ . By assuming the  $\tilde{L}$ -Lipschitz continuity of  $f_u$ , we can obtain the upper bound of the gap between the RHS and LHS of Eq. (35), i.e., the worst-case deviation between our approach and a pairwise counterpart, as follows:

$$|R_{\text{pair}}(f_u) - s(\mu)| \leq \mathbb{E}_{i \sim P_u} \mathbb{E}_{j \sim Q_u} [|s(f_u(j) - f_u(i)) - s(\mu)|] \quad (36)$$

$$\leq \mathbb{E}_{i \sim P_u} \mathbb{E}_{j \sim Q_u} [|f_u(j) - f_u(i) - \mu|] \quad (37)$$

$$\leq \tilde{L} \quad (38)$$

In Eq. (37), we used the 1-Lipschitz continuity of the softplus function;  $|s(f_u(j) - f_u(i)) - s(\mu)| \leq |f_u(j) - f_u(i) - \mu|$ . In Eq. (38), we use  $-\tilde{L} \leq f_u(j) - f_u(i) \leq \tilde{L}$ . This result indicates that, by introducing norm clipping, we can bound the gap between our approach and pairwise counterpart with the same negative sampler  $p_f(i|u)$ . As a summary, we can expect that our approach performs similarly to the pairwise counterpart with the top-weighted adaptive sampler.

## 5 EXPERIMENTS

### 5.1 Experimental Settings

**5.1.1 Datasets.** To conduct a fair comparison, we closely follow the settings of the experiments in LightGCN [28] and DREGN-CF [57]. We utilise the same datasets and train/test splits that are available for these projects on GitHub<sup>1</sup>; (a) Gowalla [37]; (b) Yelp2018<sup>2</sup>; and (c) Amazon-Book [27]. The statistics are shown in Table 1.

**5.1.2 Recommenders and Model Settings.** We list state-of-the-art methods based on neural networks as the baselines.

- **Neural Graph Collaborative Filtering (NGCF)** [63] This is a GNN-based method that inherits the design of a graph convolutional network [33].
- **Mult-VAE** [38] This is a VAE-based collaborative filtering method. This method can be considered as a DE approach.
- **Efficient Neural Matrix Factorisation (ENMF)** [12] This method is based on the non-sampling CPE approach based on Neural Collaborative Filtering (NCF) [29].
- **LightGCN** [28] This is a state-of-the-art GNN-based method. It adopts the BPR loss [50] with uniform negative sampling.
- **DREGN-CF** [57] This is a state-of-the-art LightGCN-based method with ranking DRE-based risk. It uses the same sampling strategy we discussed in Section 3.3 and a ranker-based weighting strategy.

We utilise the implementation of each method released by the authors in GitHub for these four methods: NGCF<sup>3</sup>, Mult-VAE<sup>4</sup>, ENMF<sup>5</sup>, and LightGCN<sup>6,7</sup>. For DREGN-CF, we implemented the ranking uLSIF risk without importance sampling and with non-negative risk correction, which empirically showed better ranking performance in their experiments. For NGCF, ENMF, LightGCN, and DREGN-CF, we utilise the best parameters reported by the authors for each dataset, when available; as we implemented an

<sup>1</sup><https://github.com/kuandeng/LightGCN>

<sup>2</sup><https://www.yelp.com/dataset/challenge>

<sup>3</sup>[https://github.com/xiangwang1223/neural\\_graph\\_collaborative\\_filtering.git](https://github.com/xiangwang1223/neural_graph_collaborative_filtering.git)

<sup>4</sup>[https://github.com/dawenl/vae\\_cf](https://github.com/dawenl/vae_cf)

<sup>5</sup><https://github.com/chenchongthu/ENMF>

<sup>6</sup><https://github.com/kuandeng/LightGCN.git>

<sup>7</sup><https://github.com/gusye1234/LightGCN-PyTorch>

**Table 1: Statistics of the four datasets: Gowalla, Yelp2018, Amazon-Book.**

Dataset	User #	Item #	Interaction #	Density
Gowalla	29, 858	40, 981	1, 027, 370	0.00084
Yelp2018	31, 668	38, 048	1, 561, 406	0.00130
Amazon-Book	52, 643	91, 599	2, 984, 108	0.00062

ablated variant of DREGN-CF, we set  $|\mathcal{U}_B|$  to 2500 and tuned hyper-parameters including the learning rate, coefficient of  $L_2$  regularisation, and upper bound of density ratios. For Mult-VAE, we utilised the model architecture suggested in the paper.

Our proposed method is based on LightGCN but trained with our proposed risk and norm clipping technique; we call our method the *parametric density estimation LightGCN (PDE-LGCN)*. We also examine a method discussed in Section 4.2, namely, *Wasserstein distance LightGCN (WD-LGCN)*. For the methods based on LightGCN (i.e. LightGCN, DREGN-CF, PDE-LGCN, and WD-LGCN), we used the same network architecture, and we set the number of light-weight graph convolution layers and the dimensionality of embeddings to 3 and 64, respectively. For our PDE-LGCN and WD-LGCN, we have two hyper-parameters, namely, the weight for the  $L_2$  regularisation  $\lambda$  and the upper bound of  $L_2$  norms  $\bar{n}$ . We tuned  $\lambda$  and  $\bar{n}$  based on validation within the ranges of  $\{0.01, 0.05, 0.1\}$  and  $\{1.0, 2.0, \dots, 9.0\}$ . We set  $|\mathcal{U}_B|$  to 2500 for Gowalla, Yelp2018, and Amazon-Book. For all datasets, the learning rate was tuned in  $\{0.01, 0.02, \dots, 0.1\}$ .

## 5.2 Comparison with Baselines

We first evaluate our proposed method, PDE-LGCN, and baselines in terms of overall ranking effectiveness. Following previous works [28, 57, 63], we utilise Recall@20 and nDCG@20 [32] as the evaluation measures throughout this study. Table 2 lists the R@20 and nDCG@20 of each method in the three datasets. We report the statistical significance between our proposed methods and DREGN-CF (the best-performing baseline) by a paired Tukey HSD test with 95% CI and Hedge’s  $g$  as the effect size [52] in terms of nDCG@20.

With all the datasets, our PDE-LGCN and WD-LGCN outperform all the baselines in terms of both R@20 and nDCG@20. The gain from DREGN-CF is statistically significant in Yelp2018 and Amazon-Book despite using the same model architecture. For PDE-LGCN, the p-value and effect size are  $p < 1e-3$ ,  $g = 0.0415$  in Yelp2018 and  $p < 1e-3$ ,  $g = 0.0520$  in Amazon-Book. For WD-LGCN,  $p < 1e-3$ ,  $g = 0.0689$  in Yelp2018 and  $p < 1e-3$ ,  $g = 0.1194$  in Amazon-Book. On the other hand, the difference is not significant in Gowalla;  $p = 0.244$ ,  $g = 0.0301$  for PDE-LGCN and  $p = 0.528$ ,  $g = 0.0197$  for WD-LGCN. The gap between DREGN-CF and our proposed methods is substantial particularly in the Amazon-Book dataset, which has the largest number of items in the datasets (see also Table 1). Because the quality of the negative sampling/weighting strategy has a large impact on the ranking performance for such a dataset, this result suggests that the derived negative sampler/weighting of our approach is more effective than the weighting strategy of DREGN-CF, which is heuristically designed.

By comparing PDE-LGCN and WD-LGCN, WD-LGCN outperforms PDE-LGCN with statistical significance in Amazon-Book

**Table 2: Comparison of methods in terms of ranking effectiveness. The best-performing baseline is DREGN-CF.**

Method	Gowalla		Yelp2018		Amazon-Book	
	R@20	nDCG@20	R@20	nDCG@20	R@20	nDCG@20
NGCF	0.1567	0.1325	0.0575	0.0474	0.0342	0.0262
Mult-VAE	0.1644	0.1340	0.0589	0.0458	0.0413	0.0306
ENMF	0.1512	0.1306	0.0628	0.0515	0.0344	0.0272
LightGCN	0.1828	0.1551	0.0651	0.0532	0.0421	0.0324
DREGN-CF	0.1828	0.1551	0.0685	0.0564	0.0506	0.0395
PDE-LGCN	0.1871	0.1575	0.0706	0.0582	0.0542	0.0422
WD-LGCN	0.1859	0.1567	0.0719	0.0594	0.0580	0.0466

( $p < 1e-3$ ,  $g = 0.0954$ ), whereas the difference is not significant in Gowalla and Yelp2018;  $p = 0.841$ ,  $g = 0.0121$  for Gowalla and  $p = 0.209$ ,  $g = 0.0341$  for Yelp2018. The performance gain of WD-LGCN in Amazon-Book is remarkable. This suggests that WD-LGCN is effective for a sparse dataset; it is probably because, by estimating densities on unobserved items (i.e.  $\mathcal{I} \setminus \mathcal{I}_u^+$ ), WD-LGCN can avoid the concentration of the probability mass (i.e.  $p_f(i|u)$ ) assigned to a handful of observed items. Since a large-scale dataset inevitably experiences the sparsity of observed interactions, the WD-based approach may be beneficial for large-scale settings.

## 5.3 Ablation Study

In this section, we compare PDE-LGCN and WD-LGCN with ablated variants of them. In particular, based on the discussion in Section 4.3, we hypothesised that our approach performs similarly to the pairwise approach with an adaptive negative sampler. To examine the hypothesis, we implemented an adaptive negative sampling (ANS) technique for the variants based on the pairwise approach. We first sample negative candidates based on the same sampling strategy for  $|\mathcal{I}_B|$  discussed in Section 3.3, and then, for each user, re-sample five unobserved items according to the softmax probabilities in a mini-batch (i.e.  $\exp(f_u(i)) / \sum_{i \in \mathcal{I}_B} \exp(f_u(i))$ ) based on the predicted scores.

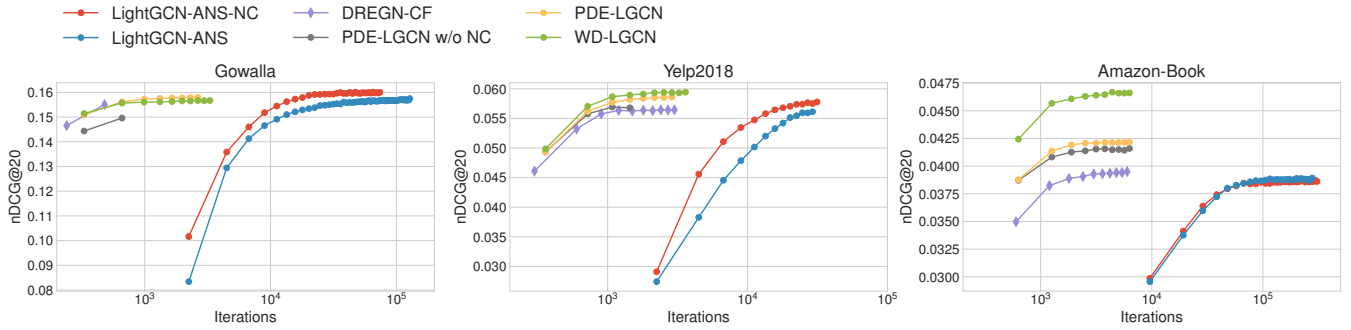
We list six ablated variants: (a) MF with the PDE risk without norm clipping (PDE-MF w/o NC); (b) MF with the PDE risk with norm clipping (PDE-MF); (c) MF with the WD risk with norm clipping (WD-MF); (d) LightGCN with the pairwise risk and ANS without norm clipping (LightGCN-ANS); (e) LightGCN with the pairwise risk and ANS with norm clipping (LightGCN-ANS-NC); and (f) LightGCN with the PDE risk without norm clipping (PDE-LGCN w/o NC). We omitted the WD-LGCN and WD-MF without norm clipping because it often fails the model training.

Table 3 lists the R@20 and nDCG@20 of each method. Our proposed methods are labelled (g) PDE-LGCN and (h) WD-LGCN. We report statistical significance and effect sizes as in Section 5.2. For the variants with the pairwise risk, we increase  $|\mathcal{U}_B|$  to 4096 for speed and ranking performance.

**5.3.1 Effectiveness of the PDE and WD risks.** From the comparison between (e) LightGCN-ANS-NC and each of (g) PDE-LGCN and (h) WD-LGCN, PDE-LGCN and WD-LGCN show comparable or substantially better performance than LightGCN-ANS-NC. The improvement is particularly clear in the Amazon-Book dataset ( $p < 1e-3$ ,  $g = 0.0781$  for PDE-LGCN and  $p < 1e-3$ ,  $g = 0.1306$  for WD-LGCN). Our methods also outperform LightGCN-ANS-NC in

**Table 3: Comparison of overall ranking effectiveness among ablated methods.**

Method	Model	Risk	norm clipping	Gowalla		Yelp2018		Amazon-Book	
				R@20	nDCG@20	R@20	nDCG@20	R@20	nDCG@20
(a) PDE-MF w/o NC	MF	PDE		0.1377	0.1097	0.0514	0.0431	0.0443	0.0349
(b) PDE-MF	MF	PDE	✓	0.1512	0.1224	0.0647	0.0525	0.0504	0.0398
(c) WD-MF	MF	WD	✓	0.1465	0.1161	0.0607	0.0513	0.0476	0.0401
(d) LightGCN-ANS	LGCN	Pairwise		0.1868	0.1568	0.0677	0.0561	0.0504	0.0389
(e) LightGCN-ANS-NC	LGCN	Pairwise	✓	0.1895	0.1601	0.0698	0.0578	0.0499	0.0386
(f) PDE-LGCN w/o NC	LGCN	PDE		0.1763	0.1493	0.0693	0.0569	0.0526	0.0414
(g) PDE-LGCN	LGCN	PDE	✓	0.1871	0.1575	0.0706	0.0582	0.0542	0.0421
(h) WD-LGCN	LGCN	WD	✓	0.1859	0.1567	0.0719	0.0594	0.0580	0.0466

**Figure 1: Comparison between LightGCN, DREGN-CF, PDE-LGCN, and WD-LGCN in terms of training efficiency.**

Yelp2018 although the differences are not significant ( $p = 0.900$ ,  $g = 0.0165$  for PDE-LGCN and  $p = 0.0576$ ,  $g = 0.0388$  for WD-LGCN). On the other hand, our methods underperform LightGCN-ANS-NC by a relatively small margin in the Gowalla dataset. However, the differences are not statistically significant ( $p = 0.597$ ,  $g = 0.0359$  for PDE-LGCN and  $p = 0.259$ ,  $g = 0.0447$  for WD-LGCN).

As PDE-LGCN, WD-LGCN and LightGCN-ANS-NC have the similar negative sampling distributions and mini-batch sampling strategy, this result suggests that our approach is particularly beneficial for use with large datasets. For a small dataset, our approach might underperform the pairwise loss with an appropriate sampling distribution and norm clipping technique.

Remarkably, in the Amazon-Book dataset, (b) PDE-MF and (c) WD-MF outperform most of the conventional neural network-based methods despite it being based on a simple MF (see Table 2). In particular, PDE-MF and WD-MF clearly outperform Mult-VAE, which is based on a DE approach. This is a surprising result because (1) PDE-MF and WD-MF are based on simple MF; and (2) for scalability, our risk estimator defined in Eq. (16) and Eq. (33) introduces the approximation of  $\exp A(f_u)$  which is exactly computed in Mult-VAE. This result further emphasises the effectiveness of our approach.

By comparing (b) PDE-MF and (c) WD-MF, PDE-MF outperforms WD-MF in Gowalla and Yelp2018;  $p < 1e-3$ ,  $g = 0.0597$  for Gowalla and  $p = 0.626$ ,  $g = 0.0174$  for Yelp2018. PDE-MF underperforms WD-MF in Amazon-Book;  $p = 0.9$ ,  $g = 0.0050$ . Considering that the difference is not significant in Yelp2018 and Amazon-Book, we recommend the use of the PDE risk for MF-based methods.

**5.3.2 Efficiency of the PDE and WD risks.** Figure 1 shows the comparison between PDE-LGCN, WD-LGCN, DREGN-CF and LightGCN-ANS with and without norm clipping in each dataset. The x- and y-axes in each figure indicate the number of training iterations in a logarithmic scale and the testing nDCG@20, respectively. We run the experiments with one NVIDIA Tesla P100 GPU, 16 Intel(R) Xeon(R) CPU @ 2.20GHz, and 60GB of RAM. For all the datasets, PDE-LGCN and WD-LGCN showed faster convergence than the LightGCN-ANS variants while achieving competitive ranking performance. PDE-LGCN converges after about 3,000 iterations and this takes 225, 261 and 522 seconds for Gowalla, Yelp2018 and Amazon-Book, respectively. WD-LGCN converges after about 4,000 iterations and this takes 301, 350 and 696 seconds for Gowalla, Yelp2018 and Amazon-Book, respectively. DREGN-CF converges with approximately 480, 2,000, and 2,000 iterations in 110, 462, and 982 seconds for Gowalla, Yelp2018, and Amazon-Book, respectively. Particularly in the Amazon-Book dataset, WD-LGCN and PDE-LGCN outperform DREGN-CF in terms of convergence time. This is because DREGN-CF takes much longer inference time than our risk estimators; the ranking uLSIF risk of DREGN-CF requires six terms involving a self-normalised weighted average, whereas our risk estimators have two average terms for each user. LightGCN-ANS-NC converges with approximately 50,000, 29,120 and 150,000 iterations in 4,843, 3,214, and 33,229 seconds for Gowalla, Yelp2018 and Amazon-Book, respectively. Our PDE-LGCN and WD-LGCN converge substantially faster than LightGCN with the pairwise loss and an adaptive negative sampler, and demonstrate comparative training efficiency when compared with DREGN-CF.



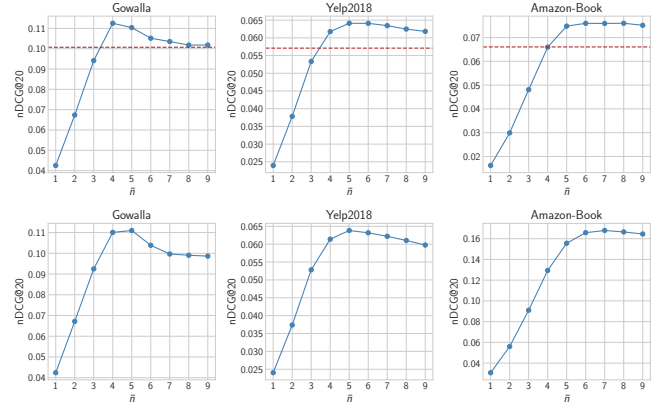
As a summary, with considering also the results in Section 5.3.1, our approach achieves comparative performance and faster convergence than the pairwise counterpart. Furthermore, it clearly outperforms the DRE-based approach in terms of effectiveness with comparative or faster model training.

**5.3.3 Effectiveness of the norm clipping Technique.** Comparing (f) and (g), PDE-LGCN consistently deteriorates without the norm clipping technique, whereas the difference is not significant in Yelp2018 and Amazon-Book.  $p < 1e-3$ ,  $g = 0.106$  for Gowalla,  $p = 0.636$ ,  $g = 0.0335$  for Yelp2018, and  $p = 0.874$ ,  $g = 0.0204$  for Amazon-Book. On the other hand, by comparing (a) PDE-MF without NC and (b) PDE-MF, PDE-MF significantly outperforms that without norm clipping in all datasets;  $p < 1e-3$ ,  $g = 0.116$  for Gowalla,  $p < 1e-3$ ,  $g = 0.0639$  for Yelp2018, and  $p < 1e-3$ ,  $g = 0.0747$  for Amazon-Book. From these results, the norm clipping technique is beneficial for our proposed approach, whereas the improvement is relatively small with LightGCN.

By introducing LightGCN, our approach further improves the ranking performance from PDE-MF for all datasets; the results of the comparison between (b) PDE-MF and (g) PDE-LGCN are  $p < 1e-3$ ,  $g = 0.283$  for Gowalla,  $p < 1e-3$ ,  $g = 0.0639$  for Yelp2018, and  $p < 1e-3$ ,  $g = 0.0366$  for Amazon-Book. These results may imply that LightGCN leads to the smoothness of predicted scores by sharing the latent vectors over the user-item graph and plays a role of inducing the Lipschitz continuity of  $f_u$ . However, the effect of graph-based locality modelling may also be a cause of the significant improvement in Gowalla, which contains the user-venue check-in data collected on a location-based social network [37].

For WD-LGCN and WD-MF, we found that the model training is quite challenging without norm clipping; this may be because the objective based on the WD in Eq. (25) is not continuous and differentiable for the distributions with disjoint supports without the Lipschitz continuity of  $f_u$  [3]. From these results, our norm clipping technique is vital for the proposed WD approach. Furthermore, due to its applicability, it allows us to take an advantage of various models such as GNNs.

To further investigate the effect of  $\bar{n}$ , we demonstrate the hyperparameter sensitivity of PDE-LGCN and WD-LGCN by varying  $\bar{n}$  from 1 to 9 while keeping other parameters fixed. Figure 2 shows the effect of  $\bar{n}$  on the validation performance of PDE-LGCN and WD-LGCN for each dataset. The x- and y-axes in each figure indicate  $\bar{n}$  and validation performance, respectively. The figures in the top and bottom demonstrate the results of PDE-LGCN and WD-LGCN, respectively. The red broken line in each figure indicates the performance with  $\bar{n} = \infty$  (i.e. PDE-LGCN w/o NC); we omit it for WD-LGCN due to unstable model training. For all the datasets, a small value ( $\bar{n} = 1, 2, 3$ ) leads to significant deterioration in terms of nDCG@20 for both PDE-LGCN and WD-LGCN. We can observe improvement with an appropriate value of  $\bar{n}$  in each dataset, whereas the performance deteriorates with a larger  $\bar{n}$  and approaches the performance seen with  $\bar{n} = \infty$  for PDE-LGCN. These results also indicate the effectiveness of the norm clipping technique, and therefore, we recommend that  $\bar{n}$  be tuned when using our approach; the appropriate value of  $\bar{n}$  tends to be large when the number of items in a dataset is large.



**Figure 2: Effect of  $\bar{n}$  on nDCG@20 of PDE-LGCN and WD-LGCN on the validation split. The figures in the top and bottom demonstrate the results of PDE-LGCN and WD-LGCN, respectively. The red broken lines indicate the performance without the norm clipping ( $\bar{n} = \infty$ ); we omit it for WD-LGCN due to unstable model training.**

## 6 CONCLUSION

We proposed a scalable learning-to-rank approach for implicit feedback. We formulated the problem as the estimation of density functions within an exponential family and developed a loss function based on the penalised MLE of generative distributions of users' positive items. Our approach includes the following features: (a) at every training step, the ranker can act as the optimal sampler for itself, thereby omitting extra sampler models; and (b) the risk estimator efficiently approximates the expectations over ranker-dependent distributions by exploiting the explicit parametric density function. Furthermore, we proposed a simple regularisation technique for MF and GNNs, namely, norm clipping, which smooths the model predictions and stabilises training. By deriving the proposed risk from a WD minimisation perspective, we addressed the essential problem in the ranking learning/evaluation; we need to know the ranking of only unobserved items. Using empirical analyses on three real-world datasets, we demonstrated the ranking effectiveness and training efficiency of PDE-LGCN and WD-LGCN. Our methods achieve comparable or substantially better ranking performance than the conventional methods, while drastically reducing the training time.

As future work, we will further explore the regularisation technique and model architectures based on our proposed risk; the discussion in Section 4.1 and Section 5.3.3 suggest that the smoothness of model prediction is essential for our approach. Based on the discussion in Section 4.1, we will theoretically investigate the duality gap between IRWGAN and our approach due to the non-convex function spaces when using complex models [7]. Since we adopted the static mini-batch sampling, it would be interesting to examine advanced algorithms [21, 67] for our approach.

## REFERENCES

- [1] Shivani Agarwal. 2011. The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In *Proceedings of the 2011 SIAM International Conference on Data Mining*.
- [2] Michael Arbel and Arthur Gretton. 2018. Kernel conditional exponential family. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1337–1346.
- [3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).
- [4] Patrick Billingsley. 2013. *Convergence of probability measures*. John Wiley & Sons.
- [5] Guy Blanc and Steffen Rendle. 2018. Adaptive sampled softmax with kernel based sampling. In *International Conference on Machine Learning*. PMLR, 590–599.
- [6] Jonathan M Borwein. 2016. A very complicated proof of the minimax theorem. *Minimax Theory and its Applications* 1, 1 (2016), 21–27.
- [7] Joey Bose, Gauthier Gidel, Hugo Berard, Andre Cianflone, Pascal Vincent, Simon Lacoste-Julien, and Will Hamilton. 2020. Adversarial Example Games. *Advances in Neural Information Processing Systems* 33 (2020).
- [8] Stephen Boyd, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic. 2012. Accuracy at the top. *Advances in neural information processing systems* 25 (2012), 953–961.
- [9] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*.
- [10] Stéphane Canu and Alex Smola. 2006. Kernel methods and the exponential family. *Neurocomputing* 69, 7–9 (2006), 714–720.
- [11] Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Jointly Non-Sampling Learning for Knowledge Graph Enhanced Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [12] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [13] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient Neural Matrix Factorization without Sampling for Recommendation. *ACM Transactions on Information Systems (TOIS)*.
- [14] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. 2020. Fast adaptively weighted matrix factorization for recommendation with implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3470–3477.
- [15] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. 2019. Samwalker: Social recommendation with informative sampling strategy. In *The World Wide Web Conference*. 228–239.
- [16] Konstantina Christakopoulou and Arindam Banerjee. 2015. Collaborative ranking with a push at the top. In *Proceedings of the 24th International Conference on World Wide Web*.
- [17] Stéphane Cléménçon, Igor Colin, and Aurélien Bellet. 2016. Scaling-up empirical risk minimization: optimization of incomplete U-statistics. *The Journal of Machine Learning Research* 17, 1 (2016), 2682–2717.
- [18] Bo Dai, Hanjun Dai, Arthur Gretton, Le Song, Dale Schuurmans, and Niao He. 2019. Kernel exponential family estimation via doubly dual embedding. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2321–2330.
- [19] Bo Dai, Zhen Liu, Hanjun Dai, Niao He, Arthur Gretton, Le Song, and Dale Schuurmans. 2019. Exponential family estimation via adversarial dynamics embedding. In *Advances in Neural Information Processing Systems*. 10979–10990.
- [20] Jingtao Ding, Yuhuan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data. In *IJCAI*.
- [21] Jingtao Ding, Yuhuan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and Robustify Negative Sampling for Implicit Collaborative Filtering. *Advances in Neural Information Processing Systems* (2020).
- [22] Jingtao Ding, Guanghui Yu, Xiangnan He, Fuli Feng, Yong Li, and Depeng Jin. 2019. Sampler design for bayesian personalized ranking by leveraging view data. *IEEE Transactions on Knowledge and Data Engineering* (2019).
- [23] Wenqi Fan, Tyler Derr, Yao Ma, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep adversarial social recommendation. *arXiv preprint arXiv:1905.13160* (2019).
- [24] Farzan Farnia and David Tse. 2018. A Convex Duality Framework for GANs. In *Advances in Neural Information Processing Systems*, Vol. 31. 5248–5258.
- [25] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NeurIPS*.
- [26] Guibing Guo, Huan Zhou, Bowei Chen, Zhirong Liu, Xiao Xu, Xu Chen, Zhenhua Dong, and Xiuqiang He. 2020. IPGAN: Generating Informative Item Pairs by Adversarial Sampling. *IEEE Transactions on Neural Networks and Learning Systems* (2020).
- [27] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*.
- [28] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [29] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*.
- [30] Xiangnan He, Jinhui Tang, Xiaoyu Du, Richang Hong, Tongwei Ren, and Tat-Seng Chua. 2019. Fast matrix factorization with nonuniform weights on missing data. *IEEE transactions on neural networks and learning systems*.
- [31] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*.
- [32] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 4 (2002).
- [33] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.
- [34] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009).
- [35] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*.
- [36] Defu Lian, Qi Liu, and Enhong Chen. 2020. Personalized ranking with importance sampling. In *Proceedings of The Web Conference 2020*. 1093–1103.
- [37] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*.
- [38] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *The World Wide Web Conference*.
- [39] Huafeng Liu, Jingxuan Wen, Liping Jing, and Jian Yu. 2019. Deep generative ranking for personalized recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 34–42.
- [40] Jixiong Liu, Weike Pan, and Zhong Ming. 2020. CoFiGAN: Collaborative filtering by generative and discriminative training for one-class recommendation. *Knowledge-Based Systems* 191 (2020), 105255.
- [41] Ting Liu, Andrew W Moore, Ke Yang, and Alexander G Gray. 2005. An investigation of practical approximate nearest neighbor algorithms. In *Advances in neural information processing systems*. 825–832.
- [42] Charles F Manski. 1975. Maximum score estimation of the stochastic utility model of choice. *Journal of econometrics* 3, 3 (1975), 205–228.
- [43] Daniel McFadden et al. 1973. Conditional logit analysis of qualitative choice behavior. (1973).
- [44] Yitong Meng, Xiao Yan, Weiwen Liu, Huanhuan Wu, and James Cheng. 2020. Wasserstein Collaborative Filtering for Item Cold-start Recommendation. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. 318–322.
- [45] Aditya Menon and Cheng Soon Ong. 2016. Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*.
- [46] Rong Pan and Martin Scholz. 2009. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*.
- [47] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*.
- [48] Guillaume Papa, Stéphane Cléménçon, and Aurélien Bellet. 2015. SGD algorithms based on incomplete U-statistics: large-scale minimization of empirical risk. *Advances in Neural Information Processing Systems* 28 (2015), 1027–1035.
- [49] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*.
- [50] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*.
- [51] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *Fourteenth ACM Conference on Recommender Systems*.
- [52] Tetsuya Sakai. 2018. Laboratory Experiments in Information Retrieval: Sample Sizes, Effect Sizes, and Statistical Power. Springer.
- [53] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*.
- [54] Bharath Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Aapo Hyvärinen, and Revant Kumar. 2017. Density estimation in infinite dimensional exponential families. *The Journal of Machine Learning Research* 18, 1 (2017), 1830–1888.
- [55] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2012. Density-ratio matching under the Bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics* (2012).

- [56] Ugo Tanielian and Flavian Vasile. 2019. Relaxed softmax for PU learning. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 119–127.
- [57] Riku Togashi, Masahiro Kato, Mayu Otani, and Shin'ichi Satoh. 2021. Density-Ratio Based Personalised Ranking from Implicit Feedback. *arXiv preprint arXiv:2101.07481* (2021).
- [58] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. 2017. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558* (2017).
- [59] Cédric Villani. 2008. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.
- [60] Haoyu Wang, Nan Shao, and Defu Lian. 2019. Adversarial binary collaborative filtering for implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5248–5255.
- [61] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *SIGIR*.
- [62] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural Memory Streaming Recommender Networks with Adversarial Training. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- [63] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*.
- [64] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. In *The World Wide Web Conference*.
- [65] Jason Weston, Chong Wang, Ron Weiss, and Adam Berenzeug. 2012. Latent collaborative retrieval. In *Proceedings of the 29th International Conference on Machine Learning*. 443–450.
- [66] Jason Weston, Hector Yee, and Ron J Weiss. 2013. Learning to rank recommendations with the k-order statistic loss. In *Proceedings of the 7th ACM conference on Recommender systems*. 245–248.
- [67] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).
- [68] Shuang-Hong Yang, Bo Long, Alexander J Smola, Hongyuan Zha, and Zhao-hui Zheng. 2011. Collaborative competitive filtering: learning recommender using context of user choice. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 295–304.
- [69] Hai-Tao Yu, Adam Jatowt, Hideo Joho, Joemon M Jose, Xiao Yang, and Long Chen. 2019. WASSRANK: Listwise document ranking using optimal transport theory. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 24–32.
- [70] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*.
- [71] Xiaofeng Zhang, Jingbin Zhong, and Kai Liu. 2020. Wasserstein autoencoders for collaborative filtering. *Neural Computing and Applications* (2020), 1–10.